Aus dem Institut für Signalverarbeitung
der Universität zu Lübeck

Direktor:
Prof. Dr.-Ing. Alfred Mertins

Synaptic Plasticity and Timing-Dependent Reinforcement in Spiking Neural Networks

Inauguraldissertation
zur
Erlangung der Doktorwürde
der Universität zu Lübeck
— Aus der Sektion Informatik —

Vorgelegt von Simon Matthew Vogt
geboren in Berlin, Deutschland

Lübeck, Februar 2015

Simon Vogt – Research Associate
Institute for Signal Processing
Universität zu Lübeck
Ratzeburger Allee 160
23538 Lübeck, Germany

1. Berichterstatter:	Prof. Dr. rer. nat. Ulrich Hofmann
2. Berichterstatter:	Prof. Dr. rer. nat. Stefan Rotter

Tag der mündlichen Prüfung: 31. Oktober 2016

Zum Druck genehmigt. Lübeck, den 1. November 2016

Abstract

When we learn new skills, our brain stores these procedural memories in a region called the basal ganglia, in order to autonomously perform movements and habits without requiring our focused attention. Many neurological illnesses such as Parkinson's Disorder (PD), Huntington's Disease (HD), and Tourette Syndrome or Obsessive Compulsive Disorder (OCD) result from a malfunction of the basal ganglia, and we are still far from understanding how this highly dynamic system can be repaired.

Current teaching models still frequently reduce the basal ganglia down to two competing pathways that either encourage (direct pathway) or discourage (indirect pathway) the selection of actions to be executed during the recall of procedural memories. The fast initiation of movements also depends on the presence of the modulatory neurotransmitter Dopamine (DA) within this area of the brain, but while fast changes of dopamine level are widely acknowledged within the neuroscience community, current medical treatments of hypokinetic disorders such as PD usually involve the broad application of dopamine-increasing drugs that ignore dopamine dynamics. For more invasive treatments such as Deep Brain Stimulation (DBS), an explanation of why they seem to temporally remove debilitating motor symptoms is even further away, and anecdotal evidence on the side effects of both dopamine-altering medication and DBS is slowly growing.

But dopamine does not only vary during the recall of procedural memories. The key to understanding the function of dopamine and basal ganglia disorders may be to closely examine the role of dopamine during the acquisition phase of procedural memories. The basal ganglia's striatum receives inputs from most parts of the brain and uses synaptic plasticity to form new procedural memories. It has to both learn and recall these memories in a highly time-critical manner in order to reduce reaction times of the animal or person responding to the physical world.

One of the two main parts of this work is therefore devoted to the thorough exploration of time-critical neural codes that allow fast and robust pickup and recall of memories within the striatum or any other brain area that has similar requirements. In particular, temporal coding is used in conjunction with spike timing dependent plasticity (STDP) to uncover the full power of anti-symmetric plasticity rules in time-critical contexts, both with and without the involvement of a dynamically changing neuromodulator that represents dopamine. Computer simulations are carried out to evaluate the best fit between a spatiotemporal neural spike code and timing-dependent synaptic plasticity, and result in a paradigm that produces highly selective receptive fields of single neurons without the explicit need for decorrelation-inducing mutual inhibition. I thereby expose important yet previously largely ignored features of STDP on synaptic drift, weight dependence, synaptic and homeostatic stability, and the interaction of multiple families of neural code with multiple plasticity mechanisms. I show that precisely-timed spatiotemporal spike codes with STDP not only reproduce the power of rate-based codes with Hebbian plasticity, but outperform them categorically. I also show how spatiotemporal codes may be passed on between groups or layers of neurons in a feed-forward manner without the need for exact synchronicity as is commonly assumed in e.g. synfire chains. I also show that the detection of and tuning to spatiotemporal spike patterns does not require the existence of an oscillatory local field potential for synchronisation, and that information transmission via spatiotemporal spike patterns can be fully independent from any rate-based fluctuations of population activity. I also develop new distance measures for three subclasses of spatiotemporal spike patterns and find a simple way to allow multiple neurons that receive

the same input spike data to form divergent, highly selective receptive fields that cover the broad range of inputs even in the absence of recurrent connections. The findings of this work may lead to the development of new experimental methods for analysing and dynamically responding to recorded biological spike data.

The second main part of this work concerns the application of dopamine into a spiking neural network (SNN) that uses STDP. While most other computational neuroscience work models the function of dopamine as affecting only direct changes to synaptic connection strengths, this work is the first to unify neuromodulated plasticity with experimentally observed instantaneous effects of dopamine on synaptic transmission. The solution we choose is biologically more plausible than previous approaches, and shows promising results for understanding dopaminergic feedback and self-regulation. Specifically, the neuromodulator here acts to dynamically change the contrast of incoming spatiotemporal spike patterns and thereby affects their ability to evoke robust postsynaptic responses. A pattern that is repeatedly combined with low levels of simulated dopamine will thereby have a low probability of being tuned to by a postsynaptic neuron, while a pattern that regularly occurs together with high concentrations of dopamine has a higher chance of being picked up by the receptive field of the postsynaptic neuron. This new method of neuromodulation is then also used to implement dopaminergic self-regulation in the basal ganglia's direct pathway in the event of over-presentation of a single input pattern, and proves promising for future biologically more realistic models of the full basal ganglia.

As we gain a greater knowledge on how the healthy basal ganglia acquire and recall new memories and how dopamine and other neuromodulators influence this function, we will be able to improve current and new treatments and maybe even heal the underlying causes of many neurodynamical diseases of the brain.

Zusammenfassung

Wenn wir neue Fähigkeiten erlernen, speichert unser Gehirn diese prozeduralen Erinnerungen in einer Hirnregion namens *Basalganglien* um in Zukunft autonom (ohne fokussierte Aufmerksamkeit) Bewegungen und angewöhnte Tätigkeiten durchzuführen. Eine Fehlfunktion der Basalganglien kann zu neurologischen Erkrankungen wie der Parkinsonkrankheit (PD), Chorea Huntington (HD), oder dem Tourettesyndrom und anderen Zwangsstörungen (OCD) führen. Leider ist unser Wissen über dieses sehr dynamische System noch zu begrenzt um eine anhaltende und nebenwirkungsfreie Heilbehandlung oder gar Reparatur der betroffenen Hirnareale durchzuführen.

Aktuelle Lehrmodelle reduzieren die Funktion der Basalganglien noch oft auf zwei konkurrierende Pfade die entweder erregend (Direkter Pfad) oder hemmend (Indirekter Pfad) auf die Selektion von Aktionen wirken. Eine schnelle Initiierung von Bewegungen erfordert auch das Vorhandensein des modulatorischen Neurotransmitters Dopamin (DA) in diesem Hirnbereich. Doch während dynamische Veränderungen von Dopaminkonzentrationen in den Basalganglien bekannt sind, basiert die klinische Behandlung neurodegenerativer Bewegungsmangelerkrankungen wie PD meist auf einer allgemeinen Erhöhung von Dopaminkonzentrationen ohne Beachtung dynamischer Effekte. Invasive Eingriffe zur Symptomunterdrückung wie die inzwischen weitläufig durchgeführte Tiefenhirnstimulation (DBS) entziehen sich bisher jeder abschließenden Erklärung über deren Wirkweise, und anekdotische Anhaltspunkte über teils schwerwiegende Nebenwirkungen von sowohl dopaminverändernden Medikamenten als auch der tiefen Hirnstimulation nehmen zu.

Allerdings verändert sich die Konzentration von Dopamin und anderen Neuromodulatoren nicht nur während des Abrufens von prozeduralen Erinnerungen. Ein wichtiger Ansatz zur Entschlüsselung des Zusammenhangs zwischen Dopamin und Funktionsstörungen der Basalganglien mag die Betrachtung der Rolle von Dopamin während der Bildung von neuen prozeduralen Erinnerungen sein. Der größte Bereich der Basalganglien, das *Striatum*, erhält Eingaben aus einer Vielzahl von Hirnbereichen und verwendet synaptische Plastizität um neue prozedurale Erinnerungen zu erzeugen und zu vertiefen. Das Striatum muss diese Engramme unter starkem Zeitdruck sowohl lernen als auch wiedergeben können, um Reaktionszeiten eines Tieres oder Menschen als Antwort auf Geschehnisse in der physikalischen Welt zu minimieren.

Der erste der zwei Hauptteile dieser Arbeit ist daher der gründlichen Erkundung zeitkritischer neuronaler Codes gewidmet, die eine schnelle und robuste Aufnahme und Wiedergabe von Speicherinhalten im Striatum und ähnlichen Hirnbereichen erlauben. Im Besonderen wird die Kombination von Zeitcodierung mit zeitabhängiger Plastizität (STDP) verwendet um in Computersimulationen die Entstehung von stark selektiven rezeptiven Feldern durch zeitlich in präziser Abfolge ankommende Aktionspotentiale zu untersuchen. Hierzu zeige ich wichtige, jedoch bisher wenig beachtete, Eigenschaften von STDP und synaptischem Drift, Abhängigkeiten von Verbindungsstärke, synaptischer und homeostatischer Stabilität, und der Interaktion von verschiedenen Klassen zeitlicher neuronaler Codes mit verschiedenen Plastizitätsmechanismen. Ich demonstriere, dass präzise zeitliche Codes in Verbindung mit STDP die Eigenschaften von Hebb'scher Plastizität nicht nur reproduzieren, sondern diesen kategorisch überlegen sind. Ebenfalls zeige ich die prinzipielle Möglichkeit der Rekonstruktion zeitlicher Abfolgen von Aktionspotentialen in nachgelagerten Gruppen bzw. Schichten von Neuronen in einer Feedforward-Anordnung, ohne eine exakte Synchronizität zu benötigen wie sie etwa im Bereich der Synchronen Ketten oft angenommen wird. Das Erlernen der Detektion von präzisen zeitlichen Codes

erfolgt auch unabhängig von etwaigen Oszillationen der Populationsfeuerrate einer Gruppe von Nervenzellen, welche somit als Taktgeber nicht benötigt wird. Tatsächlich lassen sich zeitliche Codes unabhängig von jeglichen Fluktuationen der Populationsfeuerrate in der Abfolge von Aktionspotentialen mehrerer Eingabeneurone kodieren. Ich entwickle außerdem ein neues Entfernungsmaß um zwischen zeitabhängigen neuronalen Mustern zu unterscheiden, bei denen früh eintreffende Aktionspotentiale einen höheren Einfluss auf die Detektion eines Musters haben als spät eintreffende. Des weiteren verwende ich einen erstaunlich simplen Lösungsansatz um eine Gruppe von Neuronen mit identischen Eingängen aber ohne rekurrente Verbindungen die Detektion von unterschiedlichen zeitlich präzisen Mustern von Aktionspotentialen erlernen zu lassen. Die Beiträge dieser Arbeit zum Verständnis von präzisen neuronalen Codes mögen zur Entwicklung neuer experimenteller Methoden zur (Echtzeit-)Analyse von biologischen Aufzeichnungen und der dynamischen Steuerung und Rückkopplung neuraler Systeme beitragen.

Der zweite Hauptteil dieses Textes beschäftigt sich mit dem Einfluss von Neuromodulatoren wie Dopamin auf spikende neuronale Netze (SNN) die STDP verwenden. Der überwiegende Teil existierender Arbeiten der Theoretischen Neurologie reduziert die Rolle von Dopamin in Verstärktem Lernen auf dessen direkte Wirkung für neurale Plastizität. Demgegenüber vereinigt diese Arbeit als Erste die modulatorische Wirkung von Dopamin auf synaptische Plastizität mit den experimentell ebenfalls beobachteten instantanen Effekten welche Dopamin auf synaptische Übertragung hat. Durch die Verbindung zweier experimentell beobachtbarer Prozesse ist dieser Ansatz womöglich näher an der biologischen Realität, und erlaubt hochinteressante Einblicke in die Wirkweise von dopaminergem Feedback und der Selbstregulation neuromodulatorischer Vorgänge. Im Speziellen agiert der Neuromodulator in unserem Modell als dynamischer Kontrastverstärker während der Übertragung von Mustern zeitlich präzise ankommender Aktionspotentiale unterschiedlicher räumlicher Herkunft, und beeinflusst so deren Detektion durch postsynaptische Neurone. Ein Muster welches wiederholt bei hoher simulierter Dopaminkonzentration eintrifft wird daher mit höherer Wahrscheinlichkeit von diesem postsynaptischen Neuron erlernt, während ein Muster das öfters bei niedriger Dopaminkonzentration auftritt mit nur geringer Wahrscheinlichkeit vom rezeptiven Feld des postsynaptischen Neurons erfasst wird. Wir verwenden diese Methode der Neuromodulation daraufhin zur Konstruktion einer selbstregulierenden Schleife der Dopaminausschüttung, die unter anderem eine Überanpassung einer Gruppe simulierter striataler Neurone an ein häufig wiederkehrendes Muster verhindert.

Indem wir unser Wissen darüber vergrößern, wie die gesunden Basalganglien sich neue prozedurale Erinnerungen in einem zeitkritischen Kontext aneignen und abrufen, werden wir in der Lage sein existierende medizinische Behandlungsverfahren zu verbessern sowie neue Lösungsansätze zur Heilung neurodynamischer Erkrankungen des Gehirns zu entwickeln.

Contents

I Background **1**

1 Introduction **2**
1.1 Motivation . . . 3
1.1.1 Why the Basal Ganglia? . . . 3
1.1.2 Dopamine: Teaching Signal or Contrast Enhancer? . . . 4
1.1.3 Why all the Spiking? . . . 5
1.2 Scope & Structure . . . 6

2 Biological Neurons and Membrane Dynamics **8**
2.1 Neuron, Membrane and Synapse Anatomy . . . 8
2.1.1 Neuron Anatomy . . . 8
2.1.2 Ion Channels . . . 10
2.2 Neural Membrane Dynamics . . . 11
2.2.1 Neuron Models . . . 11
2.2.2 Membrane Leak, Integrators, and Coincidence Detectors . . . 14
2.3 Types of activity-dependent long-term Synaptic Plasticity . . . 16
2.3.1 Correlations: Traditional Hebbian Plasticity . . . 16
2.3.2 Frequencies: BCM rule . . . 17
2.3.3 Temporal Order: Spike timing dependent plasticity . . . 18
2.3.4 Combined accounts of long-term plasticity . . . 19
2.4 Other forms of Neural Plasticity . . . 20

3 The Basal Ganglia **21**
3.1 Scales of Neural Modelling . . . 21
3.2 Anatomy of the Basal Ganglia . . . 23
3.3 Function of the Basal Ganglia . . . 25
3.3.1 Abstract and Rate-based Models . . . 26
3.3.2 Spike-based Models . . . 27
3.3.3 Working Hypothesis . . . 29

4 Information Theory for Synaptic Transmission **32**
4.1 Shannon's Channel Coding Paradigm . . . 32
4.2 Measuring Information . . . 33
4.3 Efficient Coding . . . 34
4.4 Signal Detection Theory . . . 35

II Synaptic Transmission and Plasticity in spiking networks **37**

5 Spike Timing, Weight Bounds, and Distributions **38**

5.1 Synaptic Drift and spike timings . . . 38
5.1.1 Bias or no bias? . . . 39
5.1.2 Distributions of pairing probability . . . 40
5.2 Weight Dependence of synaptic updates . . . 42
5.2.1 Additive vs. multiplicative STDP weight dependence . . . 42
5.2.2 Interpolating between traditional bounding rules . . . 45
5.2.3 Performance and Features (Set 1) . . . 46
5.2.4 Revisiting existing data . . . 56
5.2.5 A new approach to balanced, weight-dependent scaling of STDP . . . 59
5.2.6 Practical weight-dependent, attractor-less weight bounding rules . . . 61
5.2.7 Performance and Features (Set 2) . . . 62
5.3 Synaptic weight distributions & Homeostasis . . . 65
5.3.1 Initial Synaptic Growth . . . 67
5.3.2 Homeostatic Effects and STDP . . . 68
5.3.3 Inconclusive Experimental Data . . . 69
5.4 Summary . . . 69

6 Spatiotemporal Coding, Correlations, and Noise 70
6.1 Detection Uncertainty and Detection Speed . . . 71
6.1.1 Detection Performance for Synchronous Inputs . . . 71
6.1.2 Detection Performance for Correlated Inputs . . . 76
6.1.3 Detection Performance for Polychronous Inputs . . . 77
6.1.4 Multiple Detector Neurons recreate a Spatiotemporal Code . . . 81
6.2 Synaptic Plasticity, Pruning and Sparseness . . . 83
6.2.1 STDP finds the earliest reliable predictors of a message . . . 84
6.2.2 Non-polychronous patterns fail to harness the power of STDP . . . 87
6.2.3 Attractor-based STDP hinders learning of polychronous patterns . . . 94
6.2.4 Traditional Hebbian plasticity fails on polychronous patterns . . . 96
6.3 Multiple Overlapping Polychronous Patterns . . . 98
6.3.1 Forming Multiple Patterns . . . 100
6.3.2 Learning with Multiple Patterns . . . 100
6.4 Rephrasing the Question: From STDP to Neural Coding . . . 101
6.4.1 Neural Signal Detection as a Decision Problem . . . 101
6.4.2 Comparison to Entropy Coding . . . 101
6.4.3 Matching Pursuit and Sparse Filters . . . 102
6.4.4 A Continuum of Pattern Families . . . 102
6.5 Summary . . . 103

7 Exploring Robust Network Parameters for STDP 104
7.1 Generating plausible polychronous input patterns . . . 104
7.1.1 Pattern Types . . . 104
7.1.2 Hiding Patterns in Fluctuating Rates . . . 108
7.1.3 Hiding Patterns through fractional participation . . . 108
7.1.4 Hiding Patterns through random participation . . . 108
7.1.5 Frozen random permutations of spike order . . . 111
7.1.6 Steady-state correlations between spike trains . . . 111
7.1.7 Pattern Similarity and Unit Overlap . . . 112
7.2 Effects of Network Size, Pattern Duration, etc. . . . 115
7.2.1 Network Size and Projection Scaling . . . 115
7.2.2 Pattern-presenting vs. Background Units . . . 116

7.2.3 Pattern Duration and Interpattern Spacing . . . 116
7.3 Multiple Postsynaptic Neurons & Input Space Coverage . . . 116
7.3.1 Avoiding Mutual Inhibition . . . 119
7.3.2 Smooth Coverage of Input Space through Delayed Onset . . . 119
7.4 Noise: A feature, not a bug? . . . 120
7.5 Biasing Learning Direction . . . 120
7.5.1 Non-uniform pattern presentation counts . . . 121
7.5.2 Background rate modification with certain patterns . . . 121
7.5.3 Pattern-dependent changes to noise level . . . 121
7.5.4 Contrast enhancement or worsening . . . 121
7.5.5 Multiplication by scalar value . . . 121
7.6 Summary . . . 122

III Neuromodulation in spiking networks 123

8 Influencing Plasticity through Modulation of Synaptic Transmission 124
8.1 Introduction . . . 125
8.2 Model and Methods . . . 127
8.2.1 Network Structure . . . 127
8.2.2 Modulation Mechanics . . . 129
8.3 Results . . . 132
8.3.1 Instant Effects . . . 132
8.3.2 Effects with Synaptic Plasticity . . . 136
8.4 Conclusion . . . 141

9 Reinforcing specific spatiotemporal patterns 143
9.1 Introduction . . . 143
9.2 Methods . . . 144
9.3 Results . . . 146
9.3.1 Distorting the receptive map towards a given pattern range . . . 146
9.3.2 Distorting the receptive map away from a given pattern range . . . 148
9.4 Conclusion . . . 148

10 Limiting representational monopolies 151
10.1 Introduction . . . 151
10.2 Methods . . . 151
10.3 Results . . . 153
10.3.1 Timing effects of dopaminergic change . . . 153
10.3.2 Scaling effects of dopaminergic change . . . 157
10.3.3 Dopaminergic Feedback for Self-Regulation . . . 159
10.4 Conclusion . . . 162

IV Epilogue 163

11 Summary and Conclusion 164
11.1 Contributions to the Field . . . 164
11.2 Outlook and Future Work . . . 168
11.3 Conclusion . . . 171

V Appendices **172**

A Supplementary Data and Simulation Parameters **173**
A.1 Supplementary Figures . . . 173
A.1.1 Additive STDP (Section 6.2) . . . 173
A.1.2 One-sided and randomly permuted patterns (Section 6.3) . . . 173
A.1.3 Learning multiple randomly permuted patterns (Section 7.1.5) . . . 173
A.1.4 More examples of effective weights (Section 8.2) . . . 177
A.2 Supplementary Tables . . . 177
A.2.1 STDP rules in the literature . . . 177
A.2.2 Biased and Unbiased STDP . . . 177
A.3 Figure Settings . . . 178
A.3.1 Effect of Leak (Figures 2.5 and 2.6) . . . 178
A.3.2 Synaptic Drift (Sketch 5.2) . . . 178
A.3.3 Comparison Charts (Figures 5.6, 5.7, 5.14, 5.15) . . . 178
A.3.4 Detection Features (Figures 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8) . . . 182
A.3.5 STDP and pattern families (Figures 6.12 - 6.22, 6.25, A.1, A.2) . . . 182
A.3.6 Multiple Patterns (Figure 6.27 and supplementary Figures A.3, A.4, A.5) . . . 187
A.3.7 Pattern Generation (Figures 7.1, 7.2, 7.3 and 7.4) . . . 187
A.3.8 Polychronous Distance Metric (Figure 7.7) . . . 189
A.3.9 Multiple Neurons (Figures 7.8, 7.10 and 7.11) . . . 189
A.3.10 Reinforcing Specific Pattern Ranges (Figures 9.3 and 9.4) . . . 189
A.3.11 Exploring DA timing effects (Figures 10.1, 10.2, 10.3 and 10.4) . . . 189
A.3.12 Self-regulation of DA (Figure 10.6) . . . 189
A.4 Other Settings . . . 189
A.4.1 Activity-independent synaptic growth . . . 189
A.4.2 Simulation time step size . . . 194

B Simulator Development **195**
B.1 Simulator Structure . . . 195
B.2 Performance . . . 196
B.3 Other Simulator Software . . . 196

C Parameter & Results Tracking **198**
C.1 Versioning-based Simulation Tracking . . . 202
C.2 Results Viewer . . . 202
C.3 Comparison to other software . . . 202

D DeepBrainRecorder Software **203**
D.1 Motivation . . . 203
D.2 Structure and Efficiency . . . 204
D.3 Recording, Analysis, and Storage Modules . . . 205

E Embedded Systems Development **206**
E.1 Hardware and Development Environments . . . 207
E.2 Linux Kernel Driver System . . . 207
E.3 ADC driver development . . . 208

Bibliography **209**

Part I

Background

Chapter 1

Introduction

A few thousand years ago, a chap named Alcmaeon of Croton had a hunch that the big lumps of grey mushy stuff we have inside our heads might actually be useful for *doing stuff* like *thinking* (Oeser, 2010), which was quickly denied by many important people saying that the brain is obviously just part of the body's cooling system (Barnes, 2000). This continued until less than 200 years ago, when someone else (Flourens, 1842) discovered that this first hunch may in fact have been true after all. Since then, more and more people have come to the conclusion that our brain might even be important for survival, both to help us actively look for food and to avoid becoming food for others. It is still a great mystery, though, how exactly it does that.

Until comparably recently, brain science (or *neuroscience*, as some like to call it) consisted largely of experimental methods on how to pinch and prod certain areas of the brain into doing or not doing certain things. This trial-and-error approach quickly yielded a map of general brain areas (see Figure 1.1) where prodding certain positions always led to similar responses or sensory impressions in most humans and animals. Examining each brain area separately actually worked extremely well for some time, and people learned a lot about how to make someone's arm or leg twitch, or how the early parts of our visual system work (Hubel and Wiesel, 1959). By then, people had already noticed that the brain actually uses electricity to send quick spikes of information around between brain cells (or *neurons*), and two fellows from the seaside town of Plymouth in south England had been pinching and prodding single neurons of the squid to try and describe the fluctuations of electrical currents through mathematical equations (Hodgkin and Huxley, 1952). From today's viewpoint, one can say that they *nailed it*.

Speaking of sending around information, a completely unrelated development happened at roughly the same time. It was based on the invention of telegraph and telephone networks. More specifically, the need arose to measure the amount of information that was being passed between carriers over a noisy communication channel within a finite time. Claude Shannon (1948) came up with a couple of good ideas on how a message of information may be passed on, be compressed, reconstructed and generally even defined. *Information Theory*, as we call it today, has proven to be indispensable in the development of modern technology over the last decades, including especially computer science and machine learning.

Neuroscience, meanwhile, seems to have reached a bit of a barrier. As neuroscientists have become interested in the deeper areas of the brain, they have had to notice that the previously well-established pinching and prodding approach that worked so well on outer parts of the brain is starting to become unfeasible as a single pinch or prod in deeper areas can lead to highly chaotic outcomes that also depend on everything else the brain

is currently doing (Aihara, 2008). But not all is bad. At least the brain does appear to have some internal organizational structure that can be used here. And although it may be hard to track the exact inputs and outputs of every neuron in a network within a living brain, the above-mentioned assumption that the brain may actually be trying to *do something* may help us here.

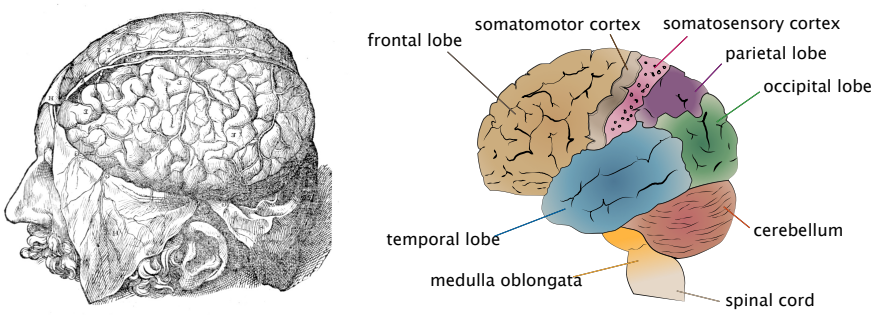


Figure 1.1: Overview of the human brain with the *dura* removed (left) and general categorisation of brain regions (right). (Left image: Vesalii Bruxellensis, 1543; Right image: Creative Commons, `http://commons.wikimedia.org/wiki/File:Cerebrum_lobes.svg`)

This is where *Computational Neuroscience* has the potential to shine. Using mathematical rules that have previously been discovered through experimental work on single building blocks of the brain (e.g. neurons, synapses, ion channels, neurotransmitters), computer simulations and theoretical modelling have the freedom to recombine these building blocks in unforeseen ways and test whether any meaningful tasks can be neatly solved in this way. This will help uncover new previously unnoticed functions that the brain may be performing and also provide answers on why the building blocks we know today look the way they do. Why does the brain use weird-looking plasticity rules to change the strength of synaptic connections between neurons? Why does the brain tolerate the seemingly noisy transmission of information that is happening via spikes nearly everywhere? Why even spikes? Aren't continuous variables in the form of rates much better, or at least equally powerful? Or could the brain actually be less binary (spike or no spike) than common simplifications/assumptions suggest?

The new questions in neuroscience are becoming more about *why* and *how* something is happening or self-organises in a certain way out of multiple options, and not so much about *what* is going on within the single building blocks anymore.

1.1 Motivation

1.1.1 Why the Basal Ganglia?

A large part of the deeper areas of the brain that have proven resilient to exposing their functional secrets through traditional pinch-and-prod methods within neuroscience are the basal ganglia. This collection of midbrain cell clusters (*nuclei*) has been connected to claims of nearly every type of task in the traditional division of brain functions in the past decades, ranging from sensory or motor relay via motivational, mood-related, and addictive properties to some involvement in broadly-defined *action selection* and procedural learning of motor skills and habits (reviewed in Redgrave, 2007). As the basal ganglia receive a large number of inputs from nearly all parts of the cortex and thalamus and also form many outgoing connections with the cerebellum and thalamus, it is extremely difficult to control all parameters for a thorough pinch-and-prod analysis without artificially slicing

off these connections and their chemical dynamics, thus changing the expected results (see also Figure 3.3, p. 24).

However, it would be very useful to know more about how the basal ganglia actually work as they are directly involved in a number of neurological illnesses. The best-known may be Parkinson's Disorder (PD), which leaves patients with motor symptoms such as tremor, rigidity, and bradykinesia, but also induces many mental problems that are difficult to quantify and so are described less often (Frank et al., 2007, 2004). But other diseases like Huntington's Disease, Obsessive Compulsive Disorder (OCD), proneness to addiction and more are also closely related to malfunctions in the basal ganglia.

It would therefore be great to know what exactly the basal ganglia are supposed to be doing in a healthy person, and how the slow loss of healthy dopamine-producing brain cells in one of its nuclei, the *substantia nigra pars compacta* (SNc), actually causes Parkinson's Disorder, for example. For some time now, people have known that dopamine seems to play an important role in the brain (Bertler and Rosengren, 1959; Carlsson, 1959), and that the loss of dopamine-producing neurons correlated with some kind of imbalance of dopamine concentrations throughout the brain (Drui et al., 2014; Janezic et al., 2013). So a seemingly easy solution was to artificially boost dopamine levels everywhere in the brain by giving PD patients dopamine-boosting drugs (Godwin-Austen et al., 1969; Hornykiewicz, 1974; Lloyd et al., 1975). This usually worked fine for a few years, depending on the individual patient, but then began to lead to a whole new set of problems, with people starting to require increasing dosages of these drugs and experiencing problems in motor control and other strong side effects (Fabbrini et al., 1987; Merims and Giladi, 2008). A big question here is: Why do the side effects of dopamine lifting drugs only start after a few years? Could the brain be slowly adapting to the drug, causing the new symptoms? If it is adapting, something is definitely going wrong in the process!

1.1.2 Dopamine: Teaching Signal or Contrast Enhancer?

In 1997, Wolfram Schultz and colleagues (Schultz et al., 1997; Suri and Schultz, 1998) noticed that the activity of dopamine-releasing (*dopaminergic*) neurons in SNc not only changes with behavioral conditions, but that the spiking activity of many of those cells is actually very similar to the so-called *reward prediction error* that is known from the behaviorally inspired field of *reinforcement learning* (Sutton and Barto, 1998). In reinforcement learning, a reward prediction error is often used as a teaching signal for the rest of the system/organism/agent to decide whether a certain choice of action in a given situation is likely to lead to more reward and should therefore be chosen again with higher chance. The finding that the activity of many dopamine neurons is so similar to a reward prediction error and that the basal ganglia are known to have some involvement in procedural learning let us wonder whether a loss of dopaminergic neurons may be somehow messing with learning processes there. Could many of the neurological illnesses originating in the basal ganglia actually be due to some plasticity process that is going wrong? And if a broad increase of dopamine levels through drugs doesn't help, could this be because the dynamic changes of dopamine concentrations are important?

Apart from having some yet to be understood involvement with plasticity in the basal ganglia, increased dopamine levels also seem to have instant effects on the spiking activity of neurons throughout the brain (Hernández-López et al., 1997; Kroener et al., 2009; Lee et al., 2004a; Nicola et al., 2000; Nicola and Malenka, 1997; Rotaru et al., 2007; Thurley et al., 2008; Waters and Helmchen, 2006). It has been hypothesized that dopamine may somehow be changing the contrast of other incoming stimuli as they arrive at a neuron (Nicola et al., 2004), causing a more precise response for higher levels of dopamine while

lower levels of dopamine might be worsening the gain and signal-to-noise ratio of those stimuli and the postsynaptic response. It is unknown, though, how exactly this works on a single spiking neuron level. Also, it is a completely open question why this effect of dopamine makes any sense under the assumption that the brain is indeed trying to get anything done: If a higher signal-to-noise ratio helps in processing, why isn't the dopamine level always fixed at a high degree? Why does dopamine vary as much as it does, and why could it be good to sometimes weaken the signal-to-noise ratio of dopamine-receiving neurons? A major goal that motivates this work is to understand why dopamine acts as it does, what the computational benefits of its multiple facets may be, and how the brain may implement dopamine-dependent plasticity processes on a spike-based level.

1.1.3 Why all the Spiking?

In order to understand learning in the basal ganglia, we need to be able to closely reconstruct plasticity processes in its largest and main input area, the *striatum*, and explain how dopamine could logically be influencing the formation of new receptive fields there while also enabling changes in the instant response to incoming stimuli.

Most theories and models about the basal ganglia nowadays live out their lives on the level of firing rates and oscillations thereof (Chapter 3). This might be fine if just the mean number of spikes per second were enough for getting stuff done in the basal ganglia. Unfortunately, it probably isn't. As the basal ganglia are by now known to be involved in fast action selection and skill execution (Redgrave, 2007; Redgrave and Gurney, 2006; Redgrave et al., 2008; Schultz, 2000, 2007), they have a strong influence on how quickly you can respond to things happening around you. If the brain always had to count the number of spikes over a given time, time-critical skills like running away from some predator or jumping from tree to tree would be impossible. In a rate-coded network, decreasing the time it waits for inputs (*integration time*) also vastly decreases the bandwidth of information that can be transmitted (Chapter 4), making us wonder why the brain would use spikes in the first place if just the number of them per second were all that mattered. The amount of transmitted information that is passed on per second can theoretically be rescued by using large pools of neurons that all represent the same message of information. However, this would require large groups of completely redundant neurons in the brain, costing lots of energy to keep alive and taking up space.

So how could the basal ganglia (and many other brain areas) be fast and use little energy? Can a single neuron be used to encode a continuous variable? And can it do so by firing only a minimum number of spikes to save energy? The idea of temporal coding proposes that the relative latency of spikes arriving from different inputs encodes valuable information (Ahissar and Arieli, 2001; Amarasingham et al., 2006; Bair and Koch, 1996; DeCharms and Merzenich, 1996; Fries et al., 2001; Gawne et al., 1996; Gerstner et al., 1997; Heil et al., 1997; Hopfield, 1995; Mainen and Sejnowski, 1995; Mehta et al., 2002; O'Keefe and Recce, 1993; Stein et al., 2005). While the idea of precise spike timing theoretically allows continuous variables to be encoded in the timespan between two spike events of a single neuron, it has seen much opposition due to the fact that spike times of single neurons are often little reliable within *in vivo* experiments (de Boer and Kuyper, 1968; Sakai et al., 1987). This has led experimenters to mostly use average responses to precisely timed inputs as the basic unit of measurement for cartographing our brains. Unfortunately, the unreliable nature of spike timings has given rise to the widespread simplification that the arrival times of spikes contain no meaningful information at all (Aertsen and Gerstein, 1985; Dorrscheidt, 1981; Foffani and Moxon, 2004; Gerstein and Kiang, 1960; Herrmann and Gerstner, 2001; Ushiba et al., 2002).

Another large goal of this work is therefore to highlight possible ways in which our brain may be using imprecisely timed spikes to reliably transmit and process information.

1.2 Scope & Structure

This work begins with a collection of four background chapters that introduce the reader to important concepts of computational neuroscience that will be needed in the subsequent chapters. After having provided an introduction to the topic and motivated my approach in the current chapter, I explain basic neural anatomy and widely used phenomenological neuron models in Chapter 2. I especially provide some early knowledge on synaptic plasticity as well as the effect of membrane leak on spike timing here. In Chapter 3, I then take a more systems-level approach to explaining brain function, and introduce the reader to the known anatomy and functional models of the region in question, the basal ganglia. I then close the background part of this work with a brief overview of important concepts in information theory in Chapter 4.

The second part of this text consists of three chapters that closely examine possible practical approaches to using imprecise temporal spike codes and *spike timing dependent plasticity* (STDP) for fast and energy efficient processing in the brain, without explicit involvement of dopamine or other neuromodulators yet. Chapter 5 therefore answers a recurring question about whether spike timing dependent updates to synaptic strength should depend on the current strength of a synapse (often called *multiplicative STDP*) or not (*additive STDP*). I will show that the dividing line between different forms of STDP has been drawn at an unfortunate position and suggest a new naming convention for STDP rules while introducing a simpler form of weight-dependent STDP that is closer to biological data than most multiplicative rules while remaining as computationally powerful as additive rules and is easy to use. In Chapter 6, I first discuss the benefits of (imprecisely timed) spatiotemporal codes over synchronous and unorderly correlated spikes from a signal detection theory standpoint. I then examine the possible advantages of using a spatiotemporal spike code together with anti-symmetric plasticity rules like STDP, and contrast them to traditional, unjustly named, *Hebbian* learning rules where only correlation and not spike order is used for updating the synaptic strength. Due to the many benefits of combining imprecise spatiotemporal spike codes with anti-symmetric STDP, I suggest a new set of functions that the brain may easily perform, which were considered costly or impossible before. Chapter 7 finally ends part two of this text with answers to practical questions arising for computer simulations of spatiotemporal codes and anti-symmetrical plasticity rules. I explain the generation of spatiotemporal input patterns within a stream of noisy background spikes and discuss the implications of network size and pattern duration. I then look at the unbiased formation of a map of receptive fields for many independent neurons and the effect of noise. I also show examples of how spatiotemporally structured spike patterns may be hidden within seemingly random background activity while still allowing detection and training by an appropriately equipped postsynaptic neuron.

Part three of this work then combines plastic spiking networks with simulated dopamine as the prototypical neuromodulator. Chapter 8 proposes a new method of dopaminergic reinforcement in spiking networks that influences plasticity through modulating synaptic transmission instead of simply scaling some assumed STDP rule. My approach is the first to cover both the instantaneous contrast enhancing effects of dopamine that have been experimentally observed, as well as dopamine's influence on spike timing dependent plasticity. This mechanism is then evaluated in the following two chapters, where Chapter 9

explores the possibility of reinforcing specific (groups of) spatiotemporally structured spike patterns over others, through fast changes of dopamine level during spike arrival. Chapter 10 then continues the exploration of this new paradigm to form a self-regulatory feedback loop for dopamine, which reduces the dopaminergic response to repeated patterns as these become more familiar to a group of simulated striatal neurons. I show that while this method of neuromodulatory action affects synaptic plasticity only very indirectly, it nevertheless is able to guide the formation of receptive fields in a controlled manner, nominating it as a valid candidate for the biological mechanism of neuromodulatory reinforcement.

I summarise and conclude my work in Chapter 11, and give ideas for future related research projects for the future of this field. The appendices provide all simulation parameters used in this work, provide some supplementary figures, and detail the construction of both the simulation software code and a tracking framework that I implemented.

Chapter 2

Biological Neurons and Membrane Dynamics

In this chapter, I will give a short introduction to the most important building blocks of biological brains, namely neurons, synapses, and ion channels and models that try to describe their dynamics. As the connections between neurons are thought to be the main method of how the brain stores information, I will also introduce some ideas that people have had on how these connections might change over time, which is generally considered a necessary mechanism for high-level learning.

2.1 Neuron, Membrane and Synapse Anatomy

I now present an overview of biological neurons, synapses, and the ion channels that enable the dynamic behaviour of neurons, before moving on to models of neural membrane dynamics in Section 2.2.

2.1.1 Neuron Anatomy

Nerve cells, or neurons, come in a wide range of morphological shapes and sizes (Figure 2.1 A-C). What they do have in common is the existence of a main cell body containing the cell nucleus (the *soma*), a highly branching tree of receiving branches (the *dendrites*), and a single *axon* that transmits information to other cells and may or may not branch out widely at its end. Neuronal axons can also be multiple centimetres long and are generally considered the main method of transmitting fast information over long distances in the brain. The *white matter* within the brain largely consists of axons.

Neurons use fast spikes of electricity to transmit information from incoming connections of the dendrites via the soma to the axon, where output connections pass on the information to the next neuron. The neuron's cell membrane thereby acts as an electric insulator between the outside and the inside of the cell, and small highly specialised proteins within this membrane (called *ion channels*) can dynamically influence both the electrical potential across and chemical concentrations of ions on each side of the membrane (Section 2.1.2). In specific conditions, the dynamic interaction between ion channels can cause a sudden increase of electrical potential within any part of the neuron, causing an *action potential*, or simply *spike*.

Transmission of information along a neuron is usually described as being unidirectional, travelling from input connections of the dendritic tree via the soma to the output connections of the axon. In reality, however, the direction of information flow within a

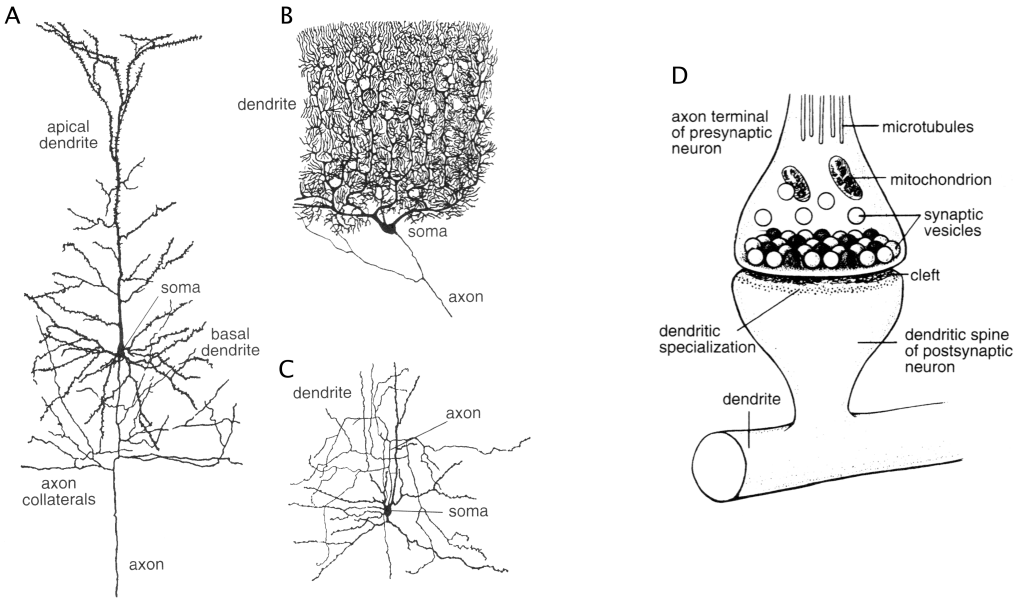


Figure 2.1: A selection of neuron types (A-C) and example diagram of a synapse (D). (A) Pyramidal neuron of the cortex. (B) Purkinje neuron from the cerebellum. (C) Stellate cells are inhibitory interneurons in the cerebral cortex. (Drawings A-C by Ramón y Cajal, 1911; D from Kandel et al., 1991; Images from Dayan and Abbott, 2001)

single neuron can be more flexible, and especially somatic spikes can produce strong feedback effects that travel back to the dendritic input connections. Still, a back-propagation of spikes *across* neural connections has not been observed in biology (Buzsáki and Kandel, 1998).

The connection between neurons is called a *synapse*, as shown in Figure 2.1D. Electrical impulses arriving at the end of the axon of one neuron usually evoke the release of stored molecules called *neurotransmitters* into the space between the first and a second neuron, which then in turn activate specialised ion channels (neurotransmitter *receptors*) that can cause a new electrical potential in the second neuron (*chemical synapses*). Each chemical synapse can hence be described as consisting of a *presynaptic* side that is part of the first neuron and a *postsynaptic* side that is part of the second neuron in this unidirectional connection. In extension, the first and second (or transmitting and receiving) neuron are also often called the pre- and postsynaptic neurons, respectively. The whole process is called *synaptic transmission*.

The type of neurotransmitter(s) that is released by the presynaptic neuron at a given synapse is thought to be predefined by the type of this neuron. This is known as *Dale's Law* or *Dale's Principle* (Dale, 1935). The family of receptors that is activated by a given neurotransmitter can either increase (excitatory connection) or decrease (inhibitory connection) the membrane potential, or otherwise influence chemical concentrations within the postsynaptic cell. The most common neurotransmitters within the cortex, *glutamate* (glutamic acid) and *GABA* (γ-Aminobutyric acid), have receptors that produce specific changes to the membrane potential of the postsynaptic neuron. Glutamate receptors generally increase postsynaptic membrane potential while GABA receptors generally decrease it. For the two most common neurotransmitters in the brain, Dale's Law therefore implies that all neurons that release one of these two neurotransmitters are either fully excitatory or fully inhibitory on all postsynaptic neurons they are connected to. This has strong implications for computational modelling. However, any (postsynaptic) neuron can receive

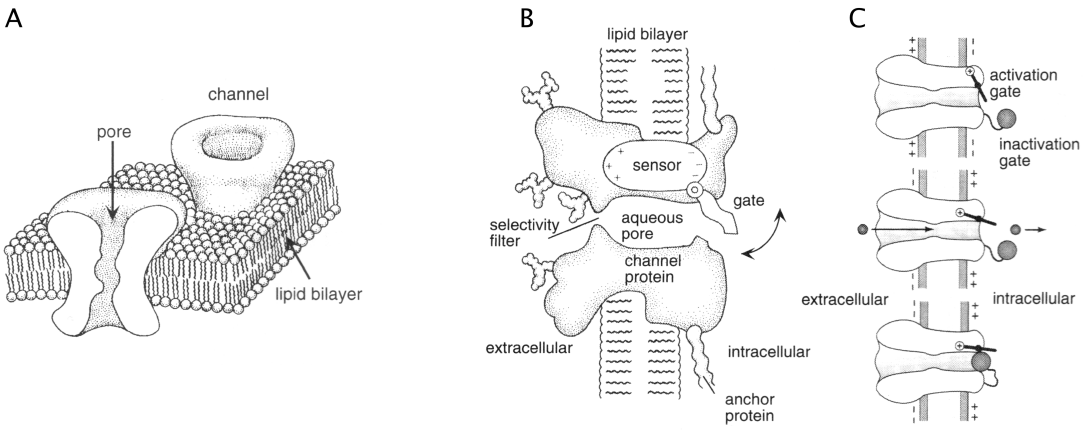


Figure 2.2: Ion channels enable the dynamic changes of membrane potentials through cascades of fast biochemical processes. (A) Principle overview of two unspecified ion channels as they bridge the neuron's cell membrane (lipid bilayer). (B) Closeup layout plan of an ion channel. (C) Dynamic functions of a voltage-gated ion channel. Notice also the changes of membrane potential on either side of the lipid bilayer. The opening of an *activation gate* allows ions to flow across the cell membrane. A second *inactivation gate* may then block ion flow on different conditions than the activation gate would need in order to close. This allows a single ion channel type to show highly complex gating dynamics. (Drawings A,B from Hille, 1991; C from Kandel et al., 1991; Images taken from Dayan and Abbott, 2001)

inputs from both excitatory and inhibitory neurons. Other neurotransmitters such as dopamine or acetylcholine have receptor families with various effects on the postsynaptic neuron, depending on which exact receptor is being expressed at a given synapse by the postsynaptic neuron. As such neurotransmitters can have both excitatory and inhibitory effects under different contexts, they are usually referred to as *neuromodulators*.

In addition to chemical synapses, the brain also uses synapses with a more direct electrical coupling (*gap junctions*). These are found in brain areas that require extremely fast processing of information, but seem to be less pronounced as a fraction of all synapses.

2.1.2 Ion Channels

Ion channels are the main mechanism for producing dynamic behaviours within neurons. As such, we now take a closer look at some of their anatomy and basic functions. The previous section has already mentioned ion channels in the context of (chemical) synaptic transmission, but they also play an important role throughout the whole neuron as a facilitator for maintaining action potentials as they travel along the neuron, as well as regulating homeostatic parameters to keep the neuron in some optimal regime of excitability.

A basic overview of how to imagine an ion channel is shown in Figure 2.2A. Ion channels are complex proteins that act as a tunnel through the otherwise impermeable cell membrane of neurons, and allow charged ions of specific chemical elements to pass. The cell membrane itself consists of a double layer of lipids that form a strong two-dimensional surface around the interior of each cell. The lipids and ion channel proteins move fairly freely within this surface. However, different types of ion channels have different probabilities of being found on different segments of a neuron.

When ions of different chemical elements are allowed to move freely between the two sides of a cell membrane, the chemical concentration of each element distributes in a gradient that is counteracted by the electrical field that many electrically charged ions generate. The *Nernst Potential* is the equilibrium potential at which differences in electrical charges of the ions on both sides of the membrane balance the chemical tendency for physical

flow of these ions. By either passively gating some chemical elements between the interior and exterior of the neuron or actively pumping them through investment of energy, the cell can produce robust and predictable dynamic changes to its membrane potential and chemical concentrations. These in turn can affect further chemical cascades as well as gene expression within the cell.

Figure 2.2B shows a more complex visualisation of a typical ion channel, complete with anchor proteins and a gate that opens and closes the channel in relation to a voltage sensor. Figure 2.2C shows an ion channel with two separate gating mechanisms, here called the *activation gate* and the *inactivation gate*. The time constants for opening and closing of gates are often different, and two or more gate types per channel can produce highly complex gating dynamics. The Hodgkin-Huxley model described in the next section aims to reproduce the dynamics of an ion channel type that has both voltage-dependent activation gates and voltage-dependent inactivation gates. In the three phases of channel activation seen in Figure 2.2C, the ion channel starts out closed, as the membrane is (in relation) negatively charged on the inside and positively charged on the outside. When the potential across the membrane changes, the activation gate opens, enabling charged ions to cross the channel. But as the potential increases further, the inactivation gate closes the channel again. After some time has passed in which the voltage has dropped back to normal, both gates relax back to their initial state.

2.2 Neural Membrane Dynamics

As we have now established the basic anatomy and function of neurons and their synapses, this section gives an introduction to the most important dynamical properties of the electric field across a neuron's membrane.

2.2.1 Neuron Models

The dynamical properties of neurons can be modelled on different levels of complexity, and I will present a short overview ranging from highly detailed to more abstract models of neurons.

Hodgkin-Huxley Model

The original model for a small compartment of the giant axon of a squid (Hodgkin and Huxley, 1952) uses two voltage-dependent ion channels as current sources I_K and I_{Na} together with a leak current I_L to describe the rise and fall of action potentials within a neuron. The potassium ion channel (I_K) uses four voltage-dependent activation gates n (n^4) for the biologically fitted equation 2.1, while the sodium ion channel (I_{Na}) uses three voltage-dependent activation gates m (m^3) and a single voltage-dependent inactivation gate h. The fit to biological data in (Hodgkin and Huxley, 1952) was done through the voltage-dependent parameters $\alpha_{\{n,m,h\}}$ and $\beta_{\{n,m,h\}}$ and a shift of the membrane's resting potential from approximately -65mV to 0mV (Izhikevich, 2007a). The complete equation is

$$\begin{aligned} C\dot{V} &= I - \overbrace{\bar{g}_K n^4 (V - E_K)}^{I_K} - \overbrace{\bar{g}_{Na} m^3 h (V - E_{Na})}^{I_{Na}} - \overbrace{\bar{g}_l (V - E_L)}^{I_L} \\ \dot{n} &= \alpha_n(V)(1-n) - \beta_n(V) n \\ \dot{m} &= \alpha_m(V)(1-m) - \beta_m(V) m \\ \dot{h} &= \alpha_h(V)(1-h) - \beta_h(V) h \end{aligned} \tag{2.1}$$

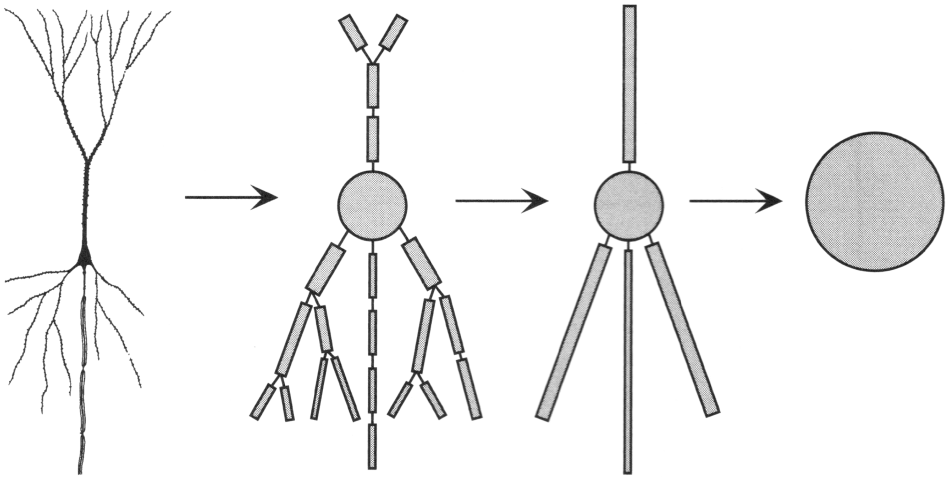


Figure 2.3: Reducing complexity of biological neurons for use in computer simulations. The continuous flow of current through a neuron's dendritic tree and across its axon can be compartmentalised in a trade-off between biological realism and computational complexity. When this results in a single compartment, we speak of a *point neuron*. Models for the dynamics governing the flow of current in each compartment are described in Section 2.2.1 (Image from Dayan and Abbott, 2001).

where I is the external input, $\bar{g}_K = 36mS/cm^2$, $\bar{g}_{Na} = 120mS/cm^2$ and $\bar{g}_L = 0.3mS/cm^2$ are the maximum conductances of each channel, and $E_K = -12mV$, $E_{Na} = 120mV$ and $E_L = 10.6mV$ are their Nernst equilibrium potentials. A dot (e.g. $\dot{V}$) over a variable indicates its derivative (e.g. $\frac{dV}{dt}$). The parameters $\alpha_{\{n,m,h\}}$ and $\beta_{\{n,m,h\}}$ are

$$\begin{aligned}
\alpha_n(V) &= 0.01\ \frac{10-V}{exp(\frac{10-V}{10})-1}, \\
\beta_n(V) &= 0.125\ exp\left(\frac{-V}{80}\right), \\
\alpha_m(V) &= 0.1\ \frac{25-V}{exp(\frac{25-V}{10})-1}, \\
\beta_m(V) &= 4\ exp\left(\frac{-V}{18}\right), \\
\alpha_h(V) &= 0.07\ exp\left(\frac{-V}{20}\right), \\
\beta_h(V) &= \frac{1}{exp(\frac{30-V}{10})+1}.
\end{aligned} \tag{2.2}$$

As it needs to track the state of the membrane potential V, the K^+ activation gate m, the Na^+ activation gate n, and the Na^+ inactivation gate h, the Hodgkin-Huxley model is called a four-dimensional equation.

Izhikevich Models

While the Hodgkin-Huxley model for the generation of neural action potentials has repeatedly been shown to be very accurate in describing a wide range of biological neurons through biologically meaningful parameters (Bower and Beeman, 1995), its many state variables mean that it is computationally very complex to simulate. Izhikevich (2003) therefore successfully used bifurcation analysis to represent the sub-threshold fluctuations

of biological neurons in a range of two-dimensional models. The standard equations of the Izhikevich neuron are

$$C\dot{v} = k(v - v_r)(v - v_t) - u + I \tag{2.3}$$
$$\dot{u} = a(bv - u) \tag{2.4}$$

with an after-spike reset condition

$$\text{if } v \geq v_{peak}, \text{then} \begin{cases} v \leftarrow c \\ u \leftarrow u + d \end{cases} \tag{2.5}$$

where v is a quickly changing variable for the membrane voltage, and u is a slowly changing variable that represents the neuron's after-spike recovery. The parameters a and b define the neuron's sub-threshold fluctuations before a spike, and c and d model the after-spike reset conditions (Izhikevich, 2007a). I is the external input current, k scales the influence of v versus u and I, v_r is the resting potential and v_t the instantaneous threshold potential.

Izhikevich (2003, 2007a) says that the parameters can be tuned to represent all known biological neurons. Typical values used in (Izhikevich, 2003) are $a = 0.02$, $b = 0.2$, $c = -65\text{mV}$ and $d = 2$.

Equation 2.3 implements the quadratic upstroke of spikes (v^2) and the inclusion of inputs, while equation 2.4 allows the model to show resonating and bursting behaviour, depending on parameter values (Izhikevich, 2007a). In normalized form, the Izhikevich neuron model (Equations 2.3, 2.4) can also be written as

$$\begin{aligned} \dot{v} &= I + v^2 - u \\ \dot{u} &= a(bv - u) \end{aligned} \tag{2.6}$$

which better visualizes the non-linear nature (v^2) of the model.

Integrate-and-Fire Models

If resonating or bursting behaviour is not required in a model neuron's membrane, the Izhikevich neuron model can be further reduced to the one-dimensional non-linear *Quadratic integrate-and-fire* model:

$$\begin{aligned} &\dot{v} = b + v^2 \\ &\quad \text{if } v \geq v_{thresh}, \text{ then } v \leftarrow v_{reset} \end{aligned} \tag{2.7}$$

This is the simplest neuron model that still shows true spiking, as linear models cannot express the upstroke (see Figure 2.4). However, linear *integrate-and-fire* neurons are often used in analytical treatments of networks of neurons, and are therefore shown below:

$$\begin{aligned} &\dot{v} = b - v \\ &\quad \text{if } v \geq v_{thresh}, \text{ then } v \leftarrow v_{reset} \end{aligned} \tag{2.8}$$

A direct result of the linear integrate-and-fire neuron missing a non-linear upstroke is that as its membrane potential approaches the firing threshold, it becomes very reactive to tiny fluctuations among its inputs, which then have a very strong influence on the actual time the membrane potential crosses the predefined threshold. This can be seen in Figure 2.4 (left), where the membrane potential spends a long time just below the threshold

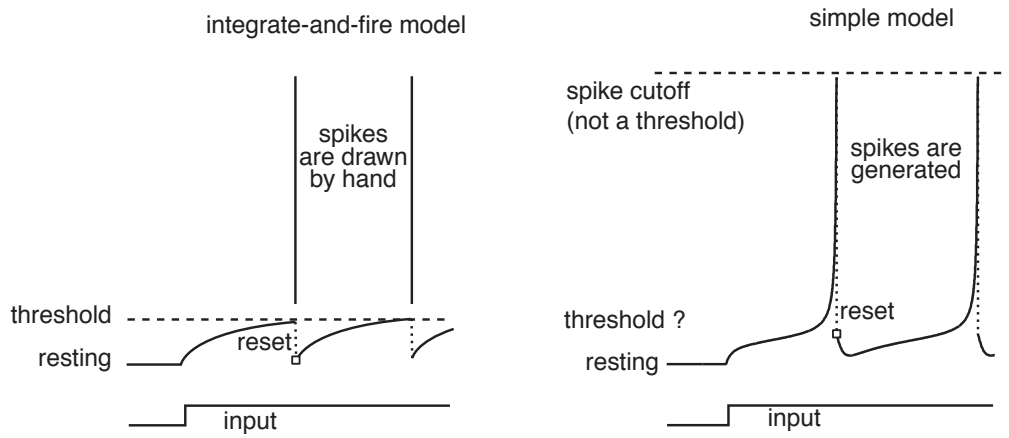


Figure 2.4: Linear integrate-and-fire neurons (*Left*) cannot actually generate spikes themselves. Instead, spikes are drawn by hand whenever the model's membrane potential reaches a predefined threshold. In contrast, non-linear models (*Right*) such as the Quadric integrate-and-fire model, the Izhikevich model, and the Hodgkin-Huxley model implement a self-amplifying spike upstroke that requires no explicit threshold value for spike declaration. (Image taken from Izhikevich, 2007a, p. 276)

before actually reaching it. As I need a phenomenological model with precise spike timing in spite of noisy inputs and low computational complexity in the following chapters, I use a default Izhikevich neuron that is reduced to a quadratic integrate-and-fire model by fixing u to its initial value.

Non-spiking Models

There also exist even more abstract models for neurons that only capture the amount of *activation* of a neuron that depends on some weighted function (e.g. the sum) of its inputs.

$$y = \sum_i w_i x_i \tag{2.9}$$

where y is the amount of activation of the postsynaptic neuron, x_i is the activation (or even just binary output) of one of many presynaptic neurons with index i, and w_i is the synaptic weight (=connection strength) between neurons. Time is here often only represented in the form of progressing steps during processing, instead of being continuous and showing dynamical properties as in the models above.

Such highly reduced models are hardly used for neuroscientific modelling today, due to their failure to capture the dynamics of neural membranes and the resulting low comparability with biological data. However, applications in machine learning successfully continue to widely use highly abstracted model neurons as feature extraction and classification algorithms, but have little in common with biological neurons apart from naming.

2.2.2 Membrane Leak, Integrators, and Coincidence Detectors

One common feature of all spiking neuron models described above is the tendency of their membrane potential to approach some equilibrium state in the absence of external input due to leak currents. But why do neural membranes even leak at all? A perfect integrator would simply wait as long as it takes to charge the membrane enough to evoke a spike, then return to some reset or resting potential, and begin the whole integration process again (compare Figure 2.5 right). The addition of a leak current causes the neuron to require a certain amount of input within limited time for evoking a spike (Figure 2.5 center), while

a very strong leak requires very strong input within a short time or the neuron will not fire at all (Figure 2.5 left). The leak current therefore can be seen as an indicator for how long the membrane will retain some form of memory about the occurrence of previous inputs.

Let us say that the maximum effect of any incoming excitatory postsynaptic current (EPSC) is smaller than the input current required to evoke a spike. In this common case, a postsynaptic neuron will need to receive multiple incoming EPSCs in order to reach a high enough membrane potential to fire a spike. Without leak, the time in which these inputs arrive can be arbitrary long (Figure 2.6 right). But if there is some form of leak current in the neuron, any EPSCs need to arrive within some maximum timespan for them to cause a spike (Figure 2.6 center). If the leak is very strong, the membrane does not remember previous inputs for very long at all, and only inputs that arrive near-simultaneously can evoke a postsynaptic response. This can be used to make a neuron specifically respond to only coincident arrivals of inputs, as shown in Figure 2.6 left.

Phenomenological neuron models are sometimes classified as being either *integrators*, i.e. expressing little leak and a long time constant for returning back to resting potential, or being *coincidence detectors* with a strong leak and thereby a fast time constant (König et al., 1996). Perfect integrators may be abstracted to rate-based neurons with just a (sigmoid) activation function as their output, while perfect coincidence detectors may be abstracted to binary artificial neurons in non-continuous applications.

When a neuron has an intermediate amount of leak (*intermediate* here strongly depends on the exact circumstances), it may show a combination of both behaviours. This allows the neuron to respond to inputs in a much more variable manner than if it were forced to be either only integrator or only coincidence detector. In the first section of chapter 6 I will show how an intermediate leak (or membrane time constant) allows precisely-timed detection of correlated inputs while increasing robustness to synaptic noise.

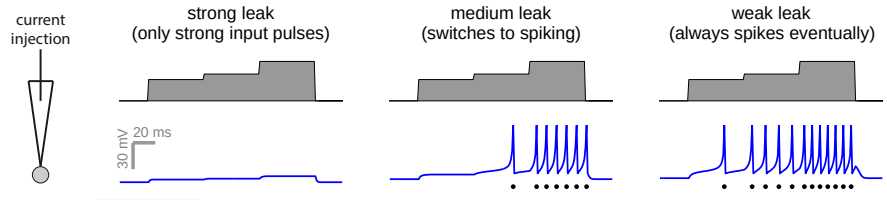


Figure 2.5: Effect of Leak on direct current injection. For some given range of piecewise constant input currents, the injected neuron may remain quiet (strong leak), begin to spike only for strong injected currents (medium leak), or respond to all inputs within the example range (weak leak). The firing frequency per current level (f-I curve) is usually nonlinear in biological neurons. *Top row:* example injected current ramps. *Bottom row:* Membrane potential of a target model neuron (see Appendix A.3.1 for parameters).

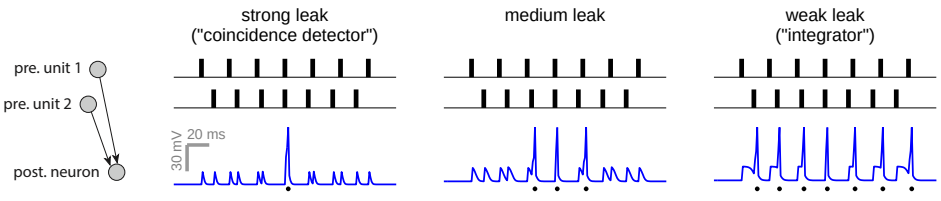


Figure 2.6: Effect of Leak on paired pulse inputs from two presynaptic spiking input units. Given example connection strengths that require two inputs for a postsynaptic response, the postsynaptic neuron responds either only for exactly coincident inputs (strong leak), inputs that occur within some maximal timespan (medium leak), or responds to most or all inputs by integrating over a long time (weak leak). *Top row:* example spike times of two presynaptic input units firing regularly with different frequencies. *Bottom row:* Membrane potential of a postsynaptic model neuron (see Appendix A.3.1 for parameters).

2.3 Types of activity-dependent long-term Synaptic Plasticity

The dynamics of electrical potentials across a neuron's membrane are not the only important factors for neural computation, however. A neuron usually receives inputs from many other neurons, which themselves receive more inputs from more neurons, up to the first sensory neurons in e.g. your eyes or ears that transform physical impressions of the outside world (*stimuli*) into neural spikes. The type of stimulus a neuron responds to strongly depends on which other neurons it is connected to. This allows neurons that are otherwise identical in genetic makeup to fulfill very different tasks, just by receiving inputs from different connections. It would also not nearly be possible to connect all neurons in the brain, as the number of required fibers (dendrites and axons) would take up much more space and require too much energy to sustain (Attwell and Laughlin, 2001). The brain, therefore, needs to choose carefully which neurons should be connected and which need not or must not be. So how does the brain decide which neurons to connect and which not to connect?

Gladly, this does not need to be decided all at once. Synaptic connections between neurons change gradually over the neurons' lifetime (mostly at the beginning) and depend on their spiking activity (Hebb, 1949) as well as on *homeostatic* factors that regulate the brain's (computational) stability (Turrigiano et al., 1998; Turrigiano and Nelson, 2004). As the synapses change over time (they can be "reshaped" or "formed"), they are called *plastic*.

2.3.1 Correlations: Traditional Hebbian Plasticity

Donald Hebb (1949, p. 62) suggested that the connection strength (or *synaptic weight*) between two neurons may be increased when a first neuron causes a second neuron to fire.

> "When an axon of cell A is near enough to excite B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased"

Unfortunately, this was soon paraphrased by many into stating that any two neurons simply need only to fire at roughly the same time, no matter in which order:

> "Neurons that fire together, wire together"

This paraphrasing of Hebb's original statement has in the last decades become known as *Hebbian Plasticity* or *Hebbian Learning*. Although the ordering was no more taken into account, traditional Hebbian plasticity has been used very successfully to model learning processes to solve many technical tasks in the field of machine learning (Bishop, 1995). A typical update rule for the change of synaptic connections $\dot{w}_i$ in Hebbian plasticity is

$$\dot{w}_i = \eta \, x_i \, y \tag{2.10}$$

where x_i is the firing rate of the ith presynaptic neuron, y is the firing rate (or degree of activation) of the postsynaptic neuron (see also Equation 2.9), and η is a factor controlling the speed of learning. When the pre- and postsynaptic firing rates tend to increase/decrease together (correlated activity), this leads to a strengthening of the synaptic connection. But when pre- and postsynaptic firing rates are not correlated, the synaptic connection strength does not increase, because the product of the two terms will usually be small.

Hebbian plasticity does not include a mechanism to also decrease weights, and so the synaptic connection strengths would increase to infinity without further care. For practical uses, therefore, a way of decreasing synaptic weights must also be included. Possible implementations include *synaptic scaling* where the sum of all synaptic strengths is made equal to some constant after each update step. Experimental work on neuronal homeostasis (Turrigiano et al., 1998; Turrigiano and Nelson, 2004) is often presented as a justification for doing this in computer simulations. However, while there is little doubt of the biological existence of activity-dependent rescaling of synaptic strengths or neural excitability, short- to mid-term homeostatic mechanisms are usually observed on different time scales than long-term synaptic plasticity. Also, the idea that all synapses of a neuron constantly change as one synaptic connection is altered seems illogical from an information transmission point of view. Still, when relaxing the requirement of biological plausibility, extensions to Hebbian learning such as Oja's Rule (Oja, 1982) have allowed artificial neural networks to become a powerful tool in machine learning.

Biological synaptic plasticity likely depends on a number of biochemical processes which are still not fully uncovered. Computational Neuroscience has continued to integrate biological experimental data into phenomenological models in order to explain how the real brain may be solving its computational tasks.

However, the idea that correlation between inputs is the main factor governing synaptic plasticity has prevailed throughout both computational neuroscience and the artificial neural network community over the last decades.

2.3.2 Frequencies: BCM rule

A highly cited work on self-regulating rate-based plasticity is that of Bienenstock et al. (1982). The rule presented by **B**ienenstock, **C**ooper, and **M**unro (BCM rule) uses the difference between the postsynaptic firing rate y and some target rate θ_M to induce either *potentiation* (strengthening) or *depression* (weakening) of synapses (see Figure 2.7).

$$\dot{w}_i = y(y - \theta_M)\, x_i - \epsilon w_i \tag{2.11}$$

The rate threshold θ_M itself may depend on the running average distance to some target firing rate y_0, leading to a self-regulating firing rate of the postsynaptic neuron:

$$\theta_M = E[(y/y_0)] \tag{2.12}$$

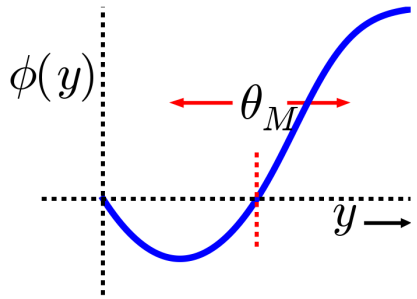


Figure 2.7: The BCM rule uses a sliding threshold on a target output rate to guide plasticity (Image from Blais and Cooper, 2008).

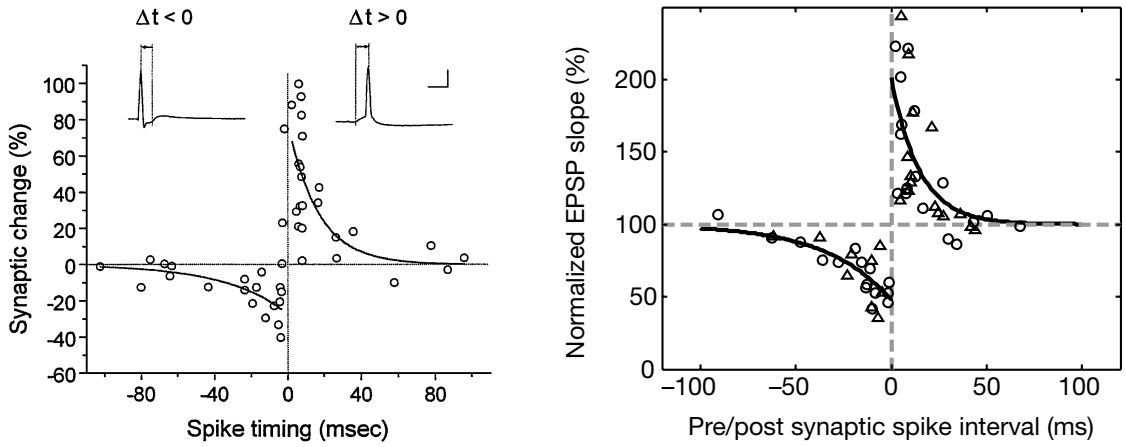


Figure 2.8: STDP experimental data. (a) Experimental data from (Bi and Poo, 1998), fitted by a curve in (Bi and Poo, 2001). (b) Data and fitted STDP curve from (Froemke and Dan, 2002). Positive spike timing differences (pre before post) lead to potentiation, while negative timings lead to depression. Note that the maximum potentiation has a larger scale than absolute maximum depression. Also, note that the exponential decay constant of potentiation is shorter than that of depression.

While rate stabilization of neurons likely is more complex than this in reality, spike frequency dependence of synaptic plasticity is seen throughout the brain (Froemke et al., 2010, 2006; Sjöström et al., 2001; Toyoizumi and Pfister, 2005).

2.3.3 Temporal Order: Spike timing dependent plasticity

Since the discovery that neurons change their spiking behaviour with behavioural context (Adrian and Zotterman, 1926), an ongoing question has been whether the single spikes that neurons fire could have importance beyond the resulting firing rate. Could the timing of spikes within a group of neurons carry additional information? An argument against this perception has long been the variability of measured spike responses to precisely timed external stimuli in experimental work (Dorrscheidt, 1981; Gerstein and Kiang, 1960). Widespread proof for temporal coding within the brain has continued to be hard to gather, possibly due to the expectation that single neurons needed to show identifiable correlation in their responses for a temporally precise response to be declared. Neural plasticity was therefore also assumed to not depend on the temporal structure of spikes, until first experimental results (Bi and Poo, 1998; Markram et al., 1997) confirmed theoretical predictions (Gerstner et al., 1996) of how spike timing may be used for learning.

Bi and Poo (1998) showed a strong dependence of neural plasticity on the timing of pairs of single spikes, with a hard switch of update direction as the order of spike arrivals on the pre- and the postsynaptic side of a synapse was reversed. Figure 2.8 shows these experimental data points together with an exponentially fitted curve (Bi and Poo, 2001; Froemke and Dan, 2002). It should be noted, though, that the observed data was collected in an experimental paradigm that involved repeated pairings over the course of multiple minutes, and that no such data could be observed for one-time pairings.

The curves that fit the experimental data in Figure 2.8 can be described as two exponential decay functions, scaled to fit each side of the time difference distribution. The general equation for timing-induced weight changes Δw as used throughout this work is

$$\Delta w = \begin{cases} A_+ \cdot \lambda \cdot e^{\frac{\Delta t}{\tau_+}} \cdot g_+(w) & \text{for } t_{pre} < t_{post} \quad \text{(LTP)} \\ -A_- \cdot \lambda \cdot e^{-\frac{\Delta t}{\tau_-}} \cdot g_-(w) & \text{for } t_{pre} > t_{post} \quad \text{(LTD)} \end{cases} \tag{2.13}$$

where A_+ and A_- are positive scaling factors, τ_+ and τ_- are the exponential decay time constants, Δt is the difference between presynaptic (t_{pre}) and postsynaptic (t_{post}) spike arrival times at a synapse ($\Delta t = t_{post} - t_{pre}$), λ is a constant that controls the learning rate, and $g(w)$ is a dynamic weight-dependent scaling parameter that will be further explained in Chapter 5.

Further theoretical work by Song et al. (2000) then coined the term *spike timing dependent plasticity* (STDP) and used randomly timed correlated inputs to show that STDP leads to a binary distribution of synaptic weights when update steps do not depend on previous synaptic strength (*additive* STDP, see Chapter 5) and the number of input units is very small. Song et al. (2000) also used a slightly more abstracted STDP rule from that fitted to the data of Bi and Poo (1998), in that both long-term potentiation (LTP) and long-term depression (LTD) had equal time constants of exponential decay in their model. As a reason for this abstraction, Song et al. (2000) state that the exact shape of an STDP rule has little effect when correlated input data is used. Much theoretical work has henceforth used STDP rules with equal decay time constants ($\tau_+ = \tau_-$) for both LTP and LTD and only controlled synaptic drift through the scaling of A_+ and A_- (e.g. Rubin et al., 2001). An overview table of typical STDP rule parameters can be found in Appendix A (Table A.1, p. 177).

I will further demonstrate the effects of synaptic drift in chapters 5 and 6 (e.g. Sketch 5.2, p. 41), where STDP without drift ($A_+ = A_-$, $\tau_+ = \tau_-$) is compared to STDP with depressing drift (hinted at by $|A_+| < |A_-|$ or $|\tau_+| < |\tau_-|$). A further review of STDP specifics is given in (Morrison et al., 2008).

2.3.4 Combined accounts of long-term plasticity

Froemke and Dan (2002) noted a dependence of timing-dependent updates to a synapse's connection strength on previous activity of the involved pre- and postsynaptic neurons. Their spike suppression rule (Froemke and Dan, 2002) stated a decreasing effect of later spikes within a burst on changing synaptic strength.

Pfister and Gerstner (2006) explained this and similar experimental results through a rule that used triplets of spikes instead of spike pairs to modify synaptic connection strength.

Clopath et al. (2010) then extended this into a voltage-based model of STDP, where previous activation of a postsynaptic neuron needed to be reflected in a low-pass filtered copy of the membrane voltage in order to allow timing-dependent weight changes to take effect. This effectively constrained STDP to a medium regime of postsynaptic firing rate, as seems to be the case in real biological neurons (Bi and Poo, 1998; Clopath et al., 2010; Froemke and Dan, 2002). For very low postsynaptic firing rates, no changes occur in the Clopath et al. (2010) model, while for very high postsynaptic firing rates, the model looses its dependence on spike timing and becomes more similar to a rate-based Hebbian model Clopath et al. (2010); Toyoizumi and Pfister (2005).

As the current work aims to uncover possible uses for fast information transmission within the basal ganglia, and as most existing work has previously focused on rate-based Hebbian modifications to synaptic strengths, I concentrate specifically on the temporal aspects of spike timings in this work by keeping pre- and postsynaptic activity within normal regimes for the observation of STDP. In the following, I will therefore be using standard STDP models as in (Bi and Poo, 1998; Billings and van Rossum, 2009; Gerstner et al., 1996; Gilson et al., 2010; Gütig et al., 2003; Guyonneau et al., 2005; Izhikevich, 2006, 2007b; Izhikevich et al., 2004; Kistler and van Hemmen, 2000; Masquelier et al., 2008; Morrison et al., 2007, 2008; Rubin et al., 2001; Song et al., 2000; van Rossum

et al., 2000; Vogt and Hofmann, 2012), without adjustments for special cases such as spike suppression, spike triplets, or voltage dependence.

2.4 Other forms of Neural Plasticity

Other forms of neural plasticity include short term plasticity, where a synapse may temporarily change its throughput according to recent neural events or slow chemical messengers, including neuromodulators. Dendritic growth during (initial) neuronal cell development is often referred to as *structural plasticity*. Random changes to synaptic connections are also likely possible, due to the statistical nature of biological systems.

Chapter 3

The Basal Ganglia

3.1 Scales of Neural Modelling

We need to establish an idea of the different scales of abstraction at which neuroscientists tend to speak about the brain.

As the complexity of the human brain has eluded scientists' understanding on so many levels from behaviour down to single cells and biochemistry, people usually need to form an abstract description in order to be able to talk about the brain at all.

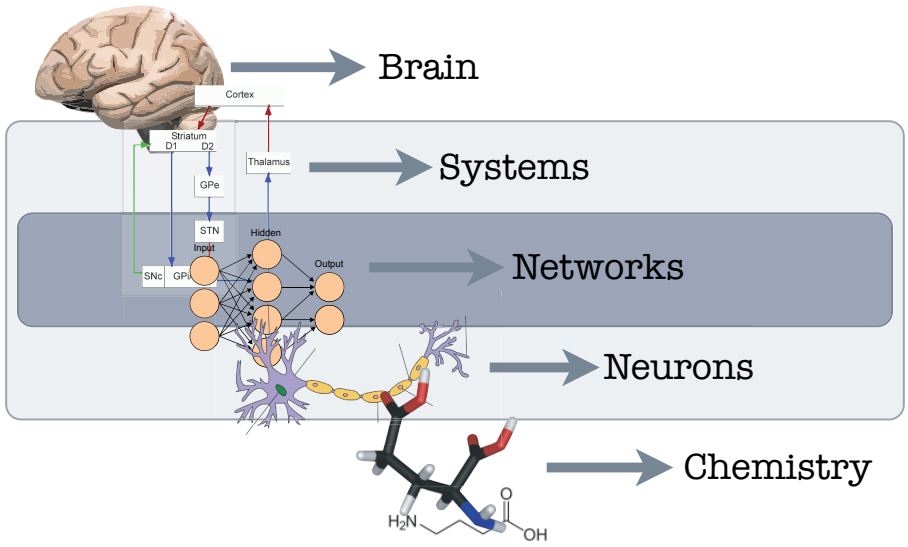


Figure 3.1: Scales of neural description. Each recording technique can only record data from a subset of these scales, and explanatory models also tend to abstract from more detailed levels in order to explain observed features. This work is largely concerned with the network level of descriptions, while bridging the gap from system-level to neuron-level explanations. (Source images: Creative Commons)

Whole Brain

Models of the whole brain are usually very abstract descriptions about general features of large assemblies of neurons. They tend to reproduce only statistical observations of EEG or fMRI recordings (Izhikevich and Edelman, 2008) without being able to solve actual tasks on any level similar to real brains. Diffusion tensor imaging (Jones and Leemans, 2011) has allowed scientists to track the propagation of high activity as it moves through

the brain, yielding a functional map of layered processing stages that fits with anatomical data.

Systems

On the systems level, research has progressed very differently for different parts of the brain. Many systems are still commonly described only through rate-based interactions between the nuclei or layers they consist of, and many models reduce the complexity of whole constitutive cell clusters to a single "activity" variable. A typical example for this are the basal ganglia, which are still described in a box-and-arrow style in most anatomy textbooks (see Section 3.2).

A contrary example for multi-scale descriptions of a functional unit is the visual system, which has been described in high detail also on functional network (Olshausen and Field, 1997) and neuron (Shlens et al., 2006) levels. However, even in the visual system there is still much left to be understood.

Networks

Network level explanations of the brain include all cases where the interaction between separate neurons is used as the main defining feature. This is also the scale of activity that is the hardest to measure experimentally, as all inputs and outputs of a large number of neurons would need to be recorded simultaneously and isolating parts of a large neural network may change its behaviour in unforeseen ways. Network-level simulations often use simple *point neurons* without modeling the full tree of anatomical branches while also using reduced membrane dynamics that capture only the most necessary features of biological neurons.

A variety of neuron models can be used as basic elements for network-level models of the brain, and the choice of neuron model is usually based on a compromise between simulation speed and essential biological realism (see Chapter 2).

Neurons & Other cells

Understanding the brain on the level of single neurons and supporting cells is a necessary requirement for reconstructing functional networks of interacting neurons. However, the dynamics within a single neuron are also highly complex and it is still not fully understood which features of neuronal dynamics are highly important for higher-level functions and which may simply be side effects that can be ignored or replaced by other features for neuronal processing. The answer likely always depends on the type of task that is to be solved.

Apart from many different types of neurons, the brain also consists of a large variety of other cells that may also have some involvement in neural information processing. Astrocytes, for example, have been shown to propagate non-electrical spikes in their internal calcium concentration to neighbouring astrocytes and neurons (Volterra and Meldolesi, 2005).

Biochemistry

Biological neurons use a large zoo of ion channels to regulate their membrane potential (Chapter 2). These ion channels and internal (metabolical) processes also depend on a large variety of neurotransmitters and other chemicals to enable and modulate their outcome, and a lot of biochemical research is currently underway to decipher statistical and

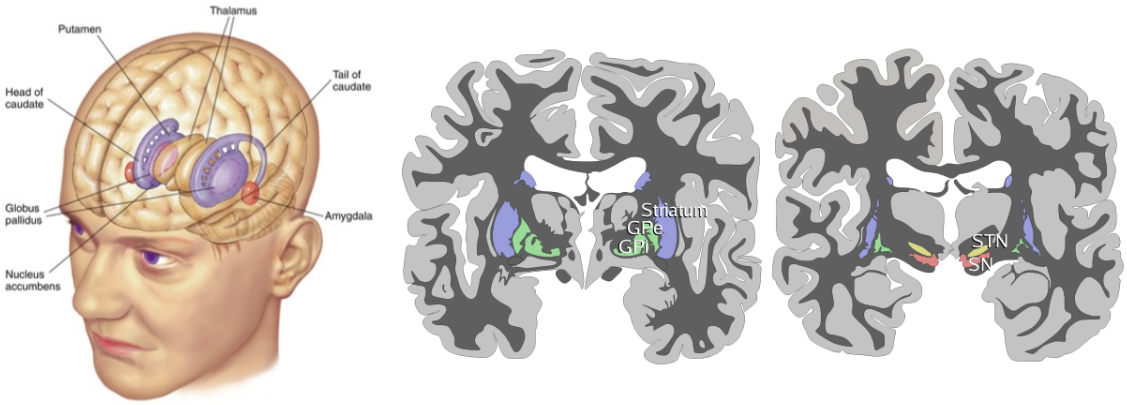


Figure 3.2: Anatomical structure of the Basal Ganglia. *Left:* Location of the basal ganglia, thalamus, and amygdala within the human brain. *Right:* Coronal (frontal) slice view of basal ganglia nuclei at two different depths. (Images: mikeclaffey.com & Creative Commons)

dynamical effects of the large group of chemical compounds that the brain uses regularly. While the sheer number of biochemical interactions within a single neuron can be discomforting, people have shown that the complexity of biological neurons can be reduced without sacrificing their most important features for higher-level neural behaviours (Izhikevich, 2007a).

3.2 Anatomy of the Basal Ganglia

The basal ganglia are a collection of midbrain nuclei that are involved in selecting the next action to perform without requiring explicit cortical attention. This is important for the autonomous execution of skills such as riding a bike or playing a musical instrument. The main anatomical parts of the basal ganglia are described below, followed by a short overview of system-level models that aim to describe their function. The actual interactions between basal ganglia nuclei remains a topic of active research, while scientists try to figure out the functional implications of the high number of inter-nuclei connections (see Figure 3.3).

As with most brain areas, each half (*hemisphere*) of the brain has its own set of basal ganglia, with each set showing a dominance in controlling one side of the body. In the following, when talking about any part of the basal ganglia, its counterpart on the opposite side of the brain is also implicitly meant unless stated otherwise.

Striatum

The largest functional area of the basal ganglia is the striatum, which itself consists of the *caudate nucleus*, the *putamen*, and the *nucleus accumbens* (shown in blueish purple in Figure 3.2). The nucleus accumbens is sometimes further subdivided into its central area and the shell due to functional division (Voorn et al., 2004).

The dominant type of neurons in the striatum are GABA-releasing (=inhibitory) *medium spiny neurons* (MSNs), which represent more than 90% of neurons there. MSNs, also known as *spiny projection neurons*, receive glutamatergic (=excitatory) inputs from nearly all areas of the cortex and thalamus, but also receive inputs from other basal ganglia nuclei as well as a number of striatal interneurons, including acetylcholine-releasing tonically active *cholinergic interneurons* (TANs) and GABA-releasing *fast-spiking interneurons* (FSIs). Medium spiny projection neurons have a large number of receptors for

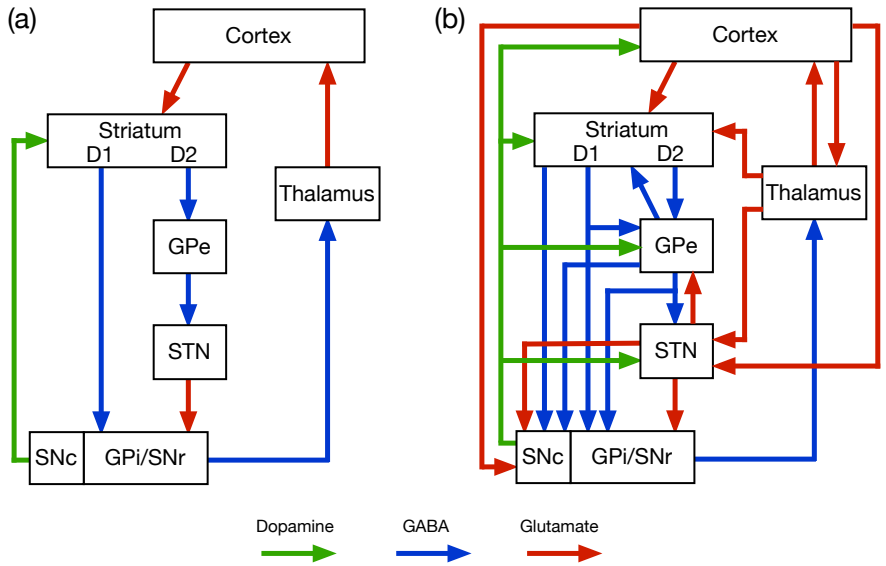


Figure 3.3: Wiring diagrams of the basal ganglia. (a) Since the direct/indirect pathway model was proposed (Albin et al., 1989), a large number of additional connections has been discovered. (b) Basal ganglia connections discovered since the Albin model (Bevan et al., 1998; Nambu et al., 2002; Parent et al., 2000; Smith et al., 1998; Smith and Kieval, 2000; Wilson, 1998), redrawn from Redgrave (2007) with additional GPe-to-SNc connection as in (Mastro et al., 2014; Paladini et al., 1999).

dopamine, which reaches the brain's highest concentration in the striatum (Redgrave, 2007). There are currently five well-known types of dopamine receptor, but these can be grouped by similarity of the effects they have on the cell that expresses them. The first group consists of receptors that are generally thought to increase excitability of (the part of) the membrane they are in when receiving high concentrations of dopamine (D1R and D5R, commonly called D1-type receptors). The second group of dopamine receptors seem to decrease excitability for high dopamine concentrations but therefore increase excitability for low concentrations (D2R, D3R, D4R, commonly called D2-type receptors). However, the exact mechanisms of how dopamine receptors affect the cells that express them is still not fully understood. The striatum contains mostly D1R and D2R dopamine receptors, and MSNs that project to the GPi and SNr (see below) tend to express more D1-type dopamine receptors, while MSNs projecting to the GPe tend to express more D2-type receptors. However, this division is not exclusive, and many striatal neurons have been found to express both types of receptors (Aizman et al., 2000; Fauchey et al., 2000; Hasbi et al., 2009; Lee et al., 2006, 2004b; Perreault et al., 2012, 2010; Rashid et al., 2007; So et al., 2005; Thompson et al., 2010).

Globus Pallidus (GPe & GPi)

The globus pallidus is the second-largest part of the basal ganglia, and is located between the putamen and thalamus, or medial to the putamen and ventro-lateral to the thalamus, to be more precise. It is anatomically divided into two functionally very separated subregions, the *external globus pallidus* (GPe) and *internal globus pallidus* (GPi), which sits slightly more ventral and medial than the GPe. Both GPe and GPi receive strong inhibitory connections from the striatum, but while GPi is considered an output nucleus of the basal ganglia, GPe projects to many other parts within the basal ganglia including the subthalamic nucleus, the striatum, and both parts of the substantia nigra (see below).

The GPe itself also appears to be further subdivided (Mallet et al., 2012), but the exact function of this is yet to be understood.

Subthalamic Nucleus (STN)

The *subthalamic nucleus* (STN) is located beneath the thalamus, and receives inhibitory connections from the GPe as well as excitatory connections directly from the cortex. It is the only basal ganglia nucleus that forms excitatory outgoing connections. Its outputs end in the GPi and GPe, thereby forming a feedback loop with the GPe. During deep brain stimulation treatments for Parkinson's Disorder, the STN has become the most targeted region of the basal ganglia due to trial-and-error testing.

Substantia Nigra (SNc & SNr)

The substantia nigra is a smaller region of the basal ganglia that is also strictly divided into two subregions. In this case, the *substantia nigra pars compacta* (SNc) contains mostly neurons that form long branches into the striatum, where they release dopamine that influences activity and plasticity there. The *substantia nigra pars reticulata* (SNr) is instead considered another output area of the basal ganglia and forms inhibitory GABAergic connections to the thalamus.

3.3 Function of the Basal Ganglia

While the basal ganglia's known anatomy has been discovered largely through pinch-and-prod approaches that include histological mapping and *in vitro* analysis of biochemistry, the exact function that this complex anatomy provides is far less clear. As seen in Section 3.2, the high number of anatomical connections and different cell types in the basal ganglia make it impossible to directly infer the exact function of each subregion from mere anatomical or in vitro electrophysiological observations. We need to find out which connections might be more important than others in which behavioural situations, and how the neural dynamics of all subregions interact to produce the ascribed functions that the basal ganglia are thought to fulfil. To complicate things, it appears impossible to research

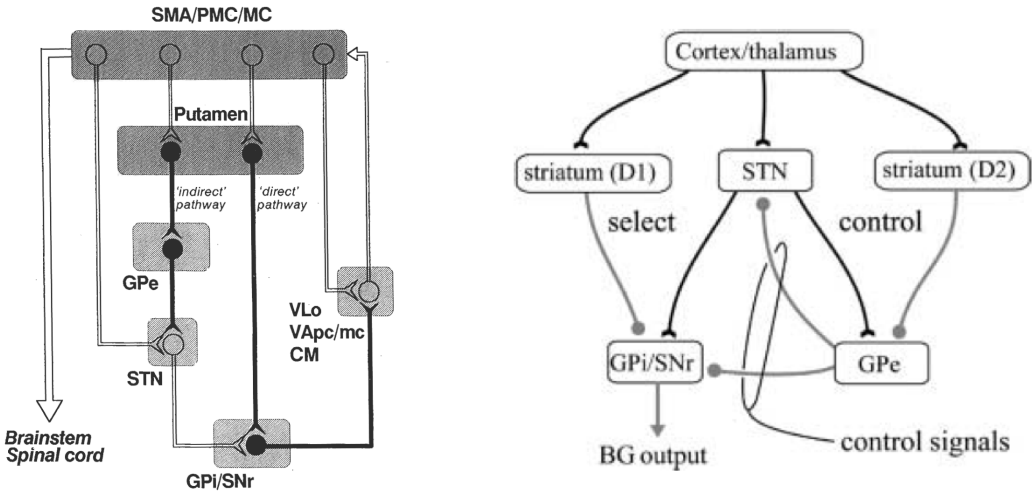


Figure 3.4: Existing models of basal ganglia function. (left) Alexander and Crutcher (1990) model of direct/indirect pathways. (right) Dynamic threshold model of Gurney et al. (2001).

the basal ganglia outside of the body, as they are so highly interconnected with nearly all other areas of the brain.

Existing models that fit the basal ganglia can be broadly categorised into two groups. The first is a collection of more abstract models that use at most rate-based neurons to try and explain how the basal ganglia might work (Section 3.3.1). The second category is a collection of models that try to implement reinforcement learning in a neural network, some of which use spiking neurons to do so (Section 3.3.2). However, neither has yet come close to solving the mystery of how the basal ganglia actually work on the level of observable spiking networks, and there is still much work to be done on the way to solving this task. This is why we can only define a temporary *working hypothesis* (Section 3.3.3) of the basal ganglia's functions that is based on the knowledge gathered *so far*, instead of relying on precise blueprints, which do not exist.

As the basal ganglia are considered to perform fast action selection, I will demonstrate in Part II of this work how the dynamics of spiking models that use temporal coding are likely very different from models that only rely on rate-based coding of information. A model that describes experimental observations on a rate-based scale often does not scale to a more detailed, spike-based implementation, and instead often produces completely different behaviour in such a changed context. At best, a formally rate-coded model may be converted into a spike-based implementation without taking any advantage of the benefits that can arise when proper temporal codes are used (see e.g. Savin et al., 2010) for the fast decision system that is the basal ganglia.

A second problem with many models presented below is that they often only decide between a predefined set of "selection channels" without showing how these channels may be formed through procedural learning. My approach here is to directly address the spike-based level of basal ganglia modelling through use of temporal coding, and include a plasticity process that automatically forms a map of receptive fields to replace the predefined selection channels that are commonly referenced in other publications (Frank, 2006; Gurney et al., 2001; Potjans et al., 2009).

3.3.1 Abstract and Rate-based Models

The standard teaching model of basal ganglia function for medical students is since the late 1980s still that of two feed forward pathways running through the basal ganglia to control motor activity (Albin et al., 1989; Alexander and Crutcher, 1990). The postulated

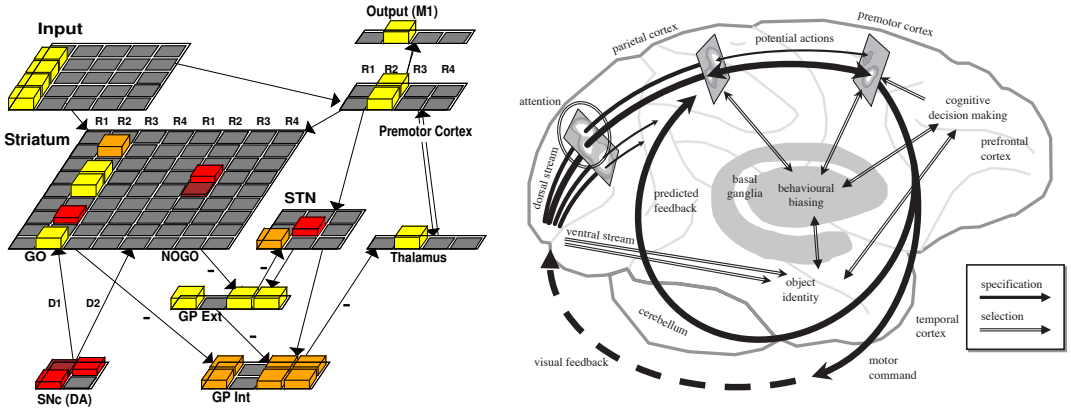


Figure 3.5: Existing models of basal ganglia function. (left) "Hold your horses" model of Frank (2006) (right) Hypothesised function of basal ganglia as a main control centre within the brain (Cisek, 2007).

direct pathway thereby runs from the striatum directly to the basal ganglia's output nuclei SNr and GPi (see Figure 3.4 left). The *indirect pathway*, in contrast, runs from a second population of neurons in the striatum to the GPe, then STN, and then also to the output nuclei SNr and GPi. A third pathway of the basal ganglia (later named the *hyperdirect pathway*) can also be seen to connect cortical inputs to the basal ganglia's STN, thereby bypassing the striatum (Nambu et al., 2002). In this model, the direct pathway is said to enable movements, while the indirect pathway acts to block the selection of movements, which is often understood as a race between the two pathways. The function of the third, hyperdirect, pathway is in this context not fully explained, but has been described as a means of direct control by cortex of thalamus over the basal ganglia's output. However, the feedforward direct/indirect pathway model in (Alexander and Crutcher, 1990) ignores many more anatomical connections that exist in the basal ganglia (see Figure 3.3) and may be too simple to explain many of the functions, disorders, and side effects of treatment that can be observed in the basal ganglia.

Gurney et al. (2001) came up with an explanation for the anatomical connections from STN to GPe and from GPe to GPi/SNr (Figure 3.4 right). In their model, the GPe is a control hub that interacts with the STN to set a dynamic threshold that potential signals passing through the direct pathway must pass in order to become selected. This is further extended by Holgado et al. (2010), who explore the conditions under which beta oscillations as observed in Parkinson's Disorder arise through frequent interactions between GPe and STN in an analytical study. Other authors have rather looked at possible interactions between the dopamine signal and the hyperdirect pathway, suggesting that the interaction between STN and GPe may act to postpone action selection in order to gather more data for making a sufficiently informed decision (Frank, 2006). See Figure 3.5 (left) for the main model diagram of Frank (2006). Again others question the role of the dopamine signal as purely signalling reward error, and propose that the release of dopamine may (also) be a way of signalling novelty within the basal ganglia and many other parts of the brain (Redgrave and Gurney, 2006).

Finally, the *affordance competition hypothesis* of Cisek (2007) sets the basal ganglia's function into a broader context as a control center for the rest of the brain, where the classical division of labour into *perception*, *cognition* and *action* between brain regions is broken into the two main functions *action specification* and *action selection*, under which all other brain functions are subsumed (Figure 3.5 right).

3.3.2 Spike-based Models

A number of publications have also tried to combine spiking neurons with reinforcement learning (Sutton and Barto, 1998) to approach the assumed function of the dopamine signal (Ljungberg et al., 1992; Schultz, 2000, 2007; Schultz et al., 1997; Suri and Schultz, 2001, 1998, 1999; Waelti et al., 2001) on a computational level.

Florian (2005, 2007) modulates the prospective weight changes that result from STDP with a scalar reinforcement value on each update step. The author assumes that the agent (in this case a worm) lives in an environment with a prearranged terrain that contains a chemical gradient signalling how far the worm's mouth is located from some food reward. As the favourable environment conveniently signals approaches towards the reward as a change of chemical concentration, no internal value function needs to be maintained and the task is highly simplified. The simple multiplication of a spike timing dependent weight change with some scalar value also evokes questions on the biological implementation of this assumed mechanism. However, this seems to be one of the earliest works that combine spiking neurons and STDP with reinforcing feedback.

Izhikevich (2007b) then goes a step further and claims to solve the distal reward problem of reinforcement learning in spiking networks by not directly affecting immediate weight changes through external reward, but using an intermediate variable. STDP here does not directly change the synaptic weight, but instead excites a so-called eligibility trace variable (known from the field of reinforcement learning) that stores the memory of previous spike pairings for a while as it slowly relaxes back to zero. If a (dopaminergic) reinforcement feedback arrives while the eligibility trace in non-zero, it is in a second step only then applied to changing the actual synaptic connection strength. Unfortunately, the problem of distal rewards is not solved in this paper in spite of contrary claims, as any biological mechanism (e.g. biochemical process) acting in a similar way could at most retain such a memory over a timespan of seconds, and not hours, days or years as is seen in humans and many animals. Other mechanisms will therefore (also) be necessary to explain biological implementations of reinforced learning.

Farries and Fairhall (2007) also directly multiply the prospective weight change arising from STDP with a scalar reinforcement signal. While the paper is well-written, addresses many useful side questions that arise when using STDP, and even correctly mentions the predictive learning feature (Abbott and Blum, 1996; Blum and Abbott, 1996) of anti-symmetric STDP which always finds the start of repeating patterns (see also Section 6.2.1), it tries to fight this through the learning task it sets itself. As the task that the paper tries to solve through reward-modulated STDP is to perfectly reproduce a predefined target spike train, it needs to fight STDP's predictive learning behaviour throughout instead of embracing it. A similar task is also tried in (Frémaux et al., 2010), which also leads to the result that STDP's inherent functionality must be counteracted in order to learn a specific set of inputs. In Chapters 8, 9 and 10 of this work I introduce a way of modulating STDP that has less need to compete with the inherent features of STDP.

Potjans et al. (2009) start with an abstract reinforcement learning model and aim to implement it using spiking neurons. However, they represent each discrete state through a group of 40 randomly spiking neurons where only the average firing rate of the group represents the state's value. Temporal coding is therefore *not* to be expected here. Also, the number of possible states is predefined and non-continuous. But the model does tackle the problem of when and how to perform automatic state value updates within its own context, leading to an interesting plasticity rule that alternates synapses between three regimes of nonplastic low and high activity, and activated plasticity. The biological evidence for the derived plasticity rule here is, at best, sparse. Still, it is one of the more bold models that try to bridge from classical reinforcement learning to spiking networks, which deserves credit.

Vasilaki et al. (2009) create a continuous time, continuous state and continuous action model that uses population coding to choose the direction an agent should move across a space of states to reach the location of a hidden reward. While the model still uses rate coding to convey state information, we consider the population code used for action selection an improvement over the discrete states used previously. The model also includes lateral interactions between output (action) units in order to improve the uniqueness of population responses. Dopamine is still included through a three-factor plasticity rule here, which results in a scalar multiplication of the reinforcement variable with presynaptic and postsynaptic activity.

Urbanczik and Senn (2009) note that the reinforced learning speed of a population of neurons drops as the size of the population increases. As the reinforcement feedback depends only on the population response and not on the individual contributions of any given neuron, neurons that happen to respond differently from the final population response on

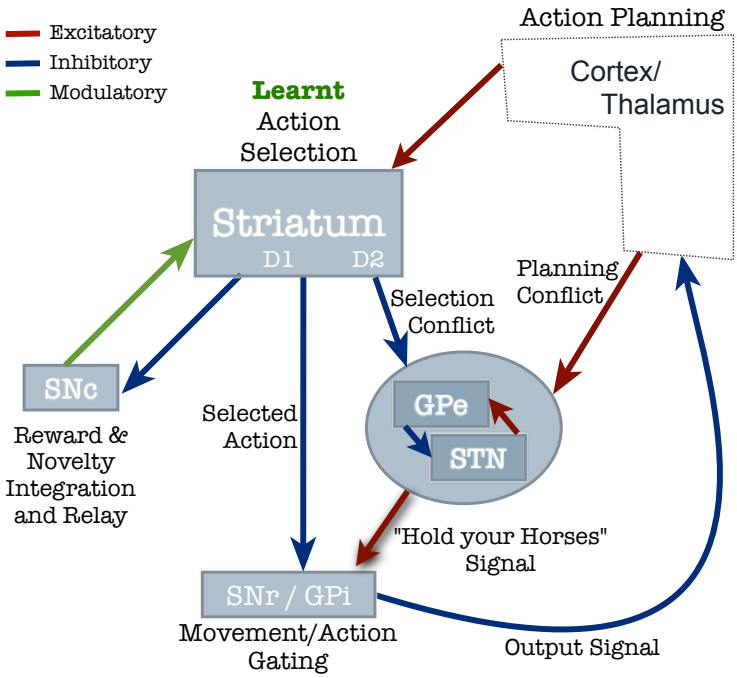


Figure 3.6: Working hypothesis of the basal ganglia as used for this work. Preliminary functions have been assigned to nuclei and a reduced set of connections based on publications noted in Section 3.3 for allowing us to work with the basal ganglia in a model. The majority of this text is concerned with plasticity processes in the Striatum that use dopaminergic reinforcement to learn to select cortical inputs correctly.

any given selection trial may be punished or rewarded wrongly, slowing down the overall speed of learning. As a solution, the paper suggests the existence of some variable that specifies how close each specific neuron's response was to the overall population response, so that the resulting reinforcement feedback can be weighted and even change sign if the neuron in question responded very differently from the whole population. The introduction of this additional feedback variable is shown to speed up learning, but requires that each neuron tracks its own history of activity, has access to the global population response, and also responds to neuromodulators present in the surrounding tissue. While the model again uses scalar multiplication to signal reinforcement in a stochastic model without any temporal coding, the presented mechanism may be useful in some cases.

3.3.3 Working Hypothesis

The current work was inspired by the wish to increase our understanding of the basal ganglia's neural spike code in order to one day improve neural implants for the treatment of neurological diseases and an overall better understanding of how the brain works. On the way to decipher the neural code of this integral part of the brain, we need to form a working hypothesis that makes preliminary assumptions about the internal functionality of all subregions of the basal ganglia. I therefore chose to assign preliminary labels to each subregion, based on the published knowledge available. From the literature review carried out during the conception stages of this work, selected publications were represented in Sections 3.3.1 and 3.3.2. The refined essence of my working hypothesis for the functions of each sub-nucleus is shown in Figure 3.6 and described below.

Action Planners and External World Evaluation

We assume that the cortex, hippocampus, and thalamus have two main roles as inputs to the basal ganglia. The first is that cortical areas plan prospective movements and more high-level actions in concert with feedback from muscle tension and other internal status signals arriving through the thalamus. The second role is the evaluation and interpretation of external world stimuli by the cortex together with the hippocampus, and the rapid comparison with new stimuli passing through the thalamus. This provides a mixture of external world state and internal action feedback to the input areas of the basal ganglia, namely the striatum and the STN.

Procedural Learner for Conflict Resolution

The inputs are then thought to be recognised and selected within the striatum, that is also constantly attempting to learn new combinations of input patterns in order to faster select them or avoid selecting them in the future. As during evolution most decisions will have required a fast response, we assume that the striatum is optimised for fast decision making. However, given enough time, the basal ganglia should also have the power to delay decisions in order to improve selection, and both speeds of decision making should be governed by the same general process to allow for a continuum of speed versus accuracy trade-off.

Dynamic threshold computation

While the STN receives direct inputs from the cortex and GPe, the GPe itself receives inputs from the STN and striatal D2R-neurons. Following Frank (2006), we assume that the GPe-STN feedback loop computes a dynamic selection threshold that it provides to the output nuclei of the basal ganglia. We hypothesise that this feedback loop has the power to allow or delay decisions in accordance with the amount of planning uncertainty in the cortex or the amount of selection uncertainty in the striatum.

Output nuclei

The output nuclei of the basal ganglia are widely believed to consist of the GPi and the SNr. They receive inhibitory projections from the D1R-type neurons of the striatum and both excitatory and inhibitory projections from the STN-GPe feedback loop, which they use to disinhibit motor areas of the thalamus and in extension the cerebellum. Due to the chain-like arrangement of "indirect pathway" nuclei, it seems likely that strong direct pathway (striatal D1R-neurons) input happens together with a higher activity in the GPe and thereby lower activity of STN, while weak activity in the direct pathway co-occurs with a lower activity in GPe and higher activity in STN. However, the exact dynamics are still far from being understood, so we cannot relay on these assumptions at the current time.

Reinforcement Relay Hub

The SNc and ventral tegmental area (VTA) are commonly named as containing dopamine-releasing neurons that form axonal projections into the striatum and other areas in the rest of the brain. But the small relative size of both SNc and VTA limits the complexity of computations that can be performed there. It is therefore likely that these areas do not take part in reward evaluation, but simply compute some difference between expected

reward and actual perceived reward that arrives from many different areas of the brain. The SNc receives collateral projections from D1R-type striatal neurons on their way to the SNr, and also receives some projections from parts of the GPe. It also receives inputs from the superior colliculus (SC) and the habenula, which may or may not indicate some information about the novelty and expectation of a stimulus, respectively.

Chapter 4

Information Theory for Synaptic Transmission

At the beginning of this text, I said that our brains may actually be trying to *do something*. More precisely, our brains seem to be important for processing the vast amount of information available in the world around us, extracting from it some form of meaningful knowledge about that world, deciding how to react to it, and telling our muscles how and when to respond. We should therefore not just look at the anatomical and electrobiochemical structure of the brain, but also ask ourselves how it might be interpreted as a biological information processing machine. *Information Theory* gives us the tools to look at the brain from this angle.

4.1 Shannon's Channel Coding Paradigm

Before we talk about the processing of information, we should first define what it is and how we can transmit information from one place in space (or time) to another.

As information theory started as a tool to better understand and build telephone networks, the standard example for an information transmission system is that of an old, unreliable telephone line. However, the system being described can be anything ranging from satellite communication or computer memory over biological cell reproduction and protein biosynthesis to the communication between (groups of) single neurons within our brain. Even the engraving of ancient texts in long-forgotten languages and their modern-day deciphering by archaeologists can be viewed as a signal transmission problem.

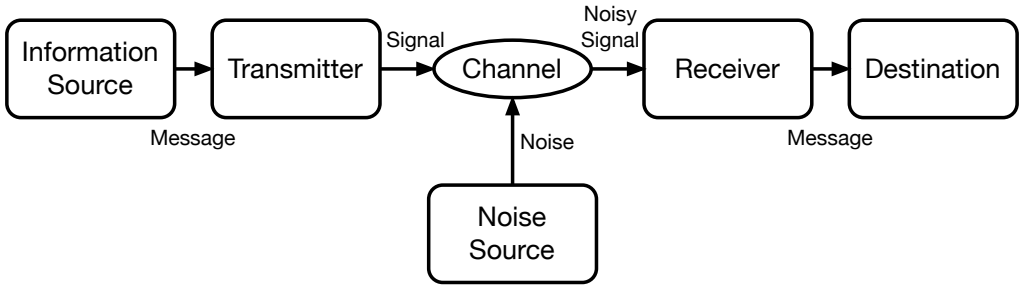


Figure 4.1: General communication system. Some *information source* produces a *message* that is to be transmitted to a *destination*. For this to happen, a *transmitter* needs to encode the message into a *signal* that can travel across some *noisy* medium (the *channel*) and be decoded by a *receiver* on the other side. (Image based on Shannon, 1948)

Claude Shannon (1948) presented a framework for describing any transmission of a message of information over a noisy channel (see Figure 4.1). Within the standard example of telephone lines, the source and destination could be two people talking over an old landline phone, the transmitter would be the microphone that converts (encodes) air pressure audio waves into electrical currents, and the receiver would be the loud speaker on the other side that converts back (decodes) the message for the destination. As any physical system is noisy, the connecting electrical phone line adds many disturbances to the transmitted signal. A standard question of information theory is how to encode a message of information into a signal such that it can be successfully decoded back into the original message on the other side of a given noisy channel. Good codes tend to use only little resources for transporting maximal information (*efficient coding*) and may sometimes be made purposely hard to decode without prior knowledge on the receiving side (*cryptography*, not further discussed here). A code that was commonly used to encode written human language well into the 1960s is the Morse code (Figure 4.2). Within the brain, it seems likely that evolutionary pressure has caused neurons to use some efficient form of communication to reach maximally possible speed and robustness to disturbances (Lewicki, 2002; Olshausen and Field, 1996a, 1997, 1996b).

4.2 Measuring Information

Not all messages that are transmitted through a communication system need to occur equally often. Examples are the alphabet of the english language, where some characters (e.g. "e") are more common than others (e.g. "q"), or sensory systems where some stimuli tend to occur more often than others in natural surroundings. The less expected a message is in relation to others, the more new information it provides at the receiving side when is does occur. Common messages are less informative as they resolve less uncertainty. If a transmitter were to always send the same message forever, the result on the receiving side would be perfectly predictable, and each new message would add no new information to the destination at all.

Information Content In order to describe this mathematically, lets say that we want to send and receive a message a_i out of some set (called *alphabet*) $\mathcal{A}_X = \{a_1, a_2, ..., a_n\}$. Not all messages tend to be needed/encountered equally often, so we need to define a probability p_i for each. The transmission outcome x can be any of the messages a_i from our alphabet, with probability $P(x == a_i) = p_i$. As always when dealing with multiple options in probability theory, the sum of all probabilities is 1. The *information content of a given transmission outcome* is then measured as a function of the message's probability to occur, with less common messages providing more information when they do happen:

$$h(x) = \log_2 \frac{1}{P(x)} = -\log_2 P(x) \tag{4.1}$$

where $P(x)$ is the probability of the transmission outcome x being some message a_i, $\log_2$ is the binary logarithm as suggested by Shannon (1948) for mathematical convenience, and $h(x)$ is the resulting information content of this new outcome in bits[1], or binary digits.

Entropy Now that we have a way of measuring and talking about the information content of new stimuli (or sensor readings, or neuronal inputs, or outcomes of a random variable),

[1] Yes, this is actually where the word *bit* comes from, which is used by nearly everyone these days!

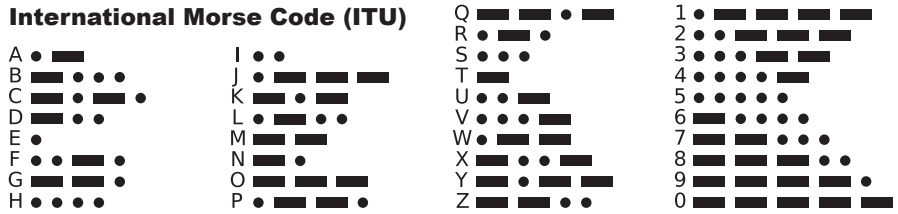


Figure 4.2: Example of a code. While the Morse Code is not an entropy code, it has some similarities in that the common letter "E" is represented by a short codeword while less common characters of the english alphabet such as "J" or "Q" are represented by longer codewords. In a different language, the frequency of characters may vary, so the efficiency (e.g. speed of manual entry) of the Morse code to transmit a given message varies with language (Image Source: Creative Commons).

we can use this to describe the average amount of information we can expect from the ongoing transmission of messages. This average amount of information or expected surprise is called the *Entropy* of a set of transmission outcomes:

$$H = \sum_{x \in \mathcal{A}_X} P(x)h(x) = -\sum_{x \in \mathcal{A}_X} P(x)\ \log_2 P(x) \tag{4.2}$$

To complete the mathematical notation, we could also group together all probabilities into the set $\mathcal{P}_X = \{p_1, p_2, ..., p_n\}$ and bundle everything into an *ensemble* $X = \{x, \mathcal{A}_X, \mathcal{P}_X\}$, as used in (MacKay, 2003). The entropy H then is actually the entropy of the ensemble, and can also be called $H(X)$.

4.3 Efficient Coding

The definition of entropy above told us that an ensemble with maximum entropy can convey a maximum average amount of information per transmitted message. So if we want to efficiently transmit information, we should use a high-entropy encoding scheme to do so.

Entropy Coding uses a varying message length to encode some other alphabet into messages where the length of each message is closely related to its information content. A typical example for an entropy code is *Huffman coding*, where exactly this happens. An ad hoc approximation of a code like this may be seen in the Morse code (Figure 4.2), where the length of common letters (e.g. "e") is encoded by a very short codeword (one short beep) and the length of less common letters uses longer codewords. An improvement over Huffman coding is *arithmetic coding*, which uses n^{th}-guesses of message occurrences and is used in many lossless compression algorithms in computers today (MacKay, 2003). Unfortunately, pure entropy codes assume a perfect connection or storage medium to function correctly, so are implausible in a biological implementation without further modification. However, the idea of encoding some messages with shorter length than others (and thus allowing different response times for different messages) may provide a useful analogy in chapter 6.

Dense Coding in neuroscience represents the idea that in a large group of receiver neurons, each neuron represents part of some collection of inputs, where the activity of all neurons needs to be known when some knowledge about the source is to be perfectly reconstructed.

While this holographic style of information storage allows for a high amount of information to be stored, decoding is rather hard (Foldiak and Endres, 2008).

Local Coding follows the opposite thought of dense codes, where within a group of neurons, each neuron only represents a single piece/message of information. The read-out of this information is very easy because of this, but the number of possible messages is restricted to the number of neurons in the network. And if one neuron dies, the information that it represents can no more be recovered (Foldiak and Endres, 2008).

Sparse Coding represents a good compromise between dense and local neural codes, as a small fraction of multiple neurons become active to signal a given message of information. This adds some redundancy against failure and has higher capacity than local codes, while also allowing much easier decoding than dense codes (Foldiak and Endres, 2008). Sparse codes may be formed when aiming to increase the entropy of neural outputs while minimizing their mutual information (Bell and Sejnowski, 1995; Olshausen and Field, 1997, 1996b)

4.4 Signal Detection Theory

When a message is sent over a noisy channel, the receiver might have some difficulty deciding which message was actually received. It often needs to make a best guess about which of a set of messages is being received at any given time. For example, two possible messages being transmitted over a noisy channel may be whether a dangerous predator within some radius around an animal is present (message 1) or not (message 2). During most of the time, message 2 may be transmitted by the environment. But when message 1 is transmitted, the animal needs to respond or it may be eaten. The task of detecting a signal is very common among all living things, as errors may have huge implications for the animal in question.

When deciding about the presence or non-presence of something within some noisy environment, we speak of *detecting* or *not detecting* something that is in reality either *present* or *not present* (Green and Swets, 1966; Heeger, 1997; Macmillan, 2002). The scientific field of *signal detection theory* was originally formed during the second world war, when humans were listening to sound or radio wave transmissions in order to decide whether enemy forces were approaching or not. Due to their independent inception, Signal Detection Theory and Information Theory have a slight notational incompatibility,

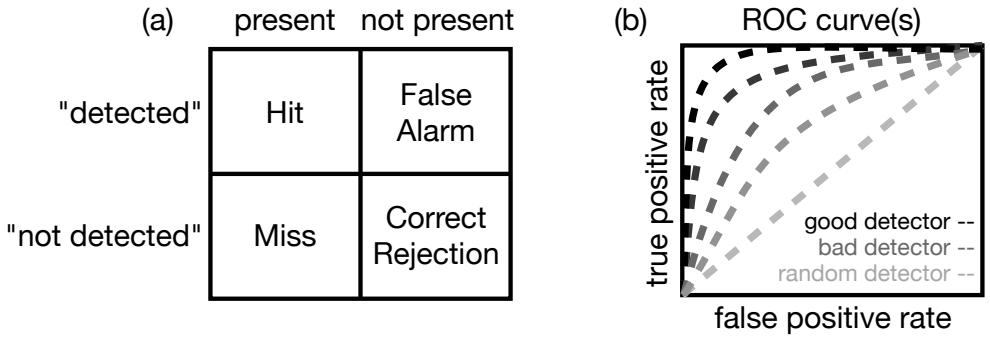


Figure 4.3: (a) The definition of hits and misses, but also of false alarms and correct rejections. (b) Typical view of multiple ROC curves that allow an intuitive comparison between multiple two-class classifiers (=detectors).

as a “signal” in signal detection theory can have different states, which is more similar to a “message” in Shannon’s classical information theory. As the decision between two alternatives can also be seen as a two-class classification problem (between message 1 and message 2), we stick to the information theoretic definition of signals and messages whenever possible. For the explanation of signal detection theory, I substitute the misleading word “signal” by “stimulus”.

Generally, there are four possible cases when trying to detect the presence of some stimulus among noise (see Figure 4.3a). If the stimulus was present and was also detected, we call this a *hit* (= true positive). If the stimulus was not present and none was detected, this is a *correct rejection* (= true negative). But if there was a stimulus present that the receiver failed to detect, this constitutes a *miss* (= false negative). And if the receiver overreacted and detected the stimulus although it was not present within the noisy environment, this is called a *false alarm* (= false positive).

The cost of possible errors may influence the detection bias of the detector. If the cost of false negatives is higher than the cost of false positives, the detector may choose to rather accept a higher fraction of false alarms than missing an event. This inherent tendency of classification can be visualised by plotting the *receiver operating characteristic* (Figure 4.3b), which shows the rate of hits over the rate of false alarms in a curve for every possible detection bias of each detector (or *classifier*). The visual inspection of a ROC plot allows a quick and intuitive comparison between multiple classifiers on a given test set of detection tasks. A large area-under-curve indicates a good classifier, while a diagonal line (half area) indicates a random and thereby worst classifier. Besides finding the best balance between hits and false alarms for a given application, ROC analysis is thereby also useful to compare the performance of different classifiers on a two-class problem.

Part II

Synaptic Transmission and Plasticity in spiking networks

Chapter 5

Spike Timing, Weight Bounds, and Distributions

In the previous chapters (Part I of this work), we touched on some background information on the basal ganglia, neural anatomy, synaptic plasticity, and visited a quick introduction to some important concepts of information theory. Chapters 5, 6 and 7 (Part II) are devoted to thoroughly understanding the mechanisms of (im)precisely timed spikes and their effect on spike timing dependent plasticity (STDP) before we include dopamine-like neuromodulation in Part III.

While chapter 6 examines the benefits that temporal coding may have on signal detection and plasticity, chapter 7 deals with practical issues of modelling temporal codes and STDP within a computer simulation. Before we can proceed to answering these questions, however, we should first spend some time on comparing established (pair-based) STDP models and choose the best suited model based on currently known biology and computational power. In this process I suggest a new weight dependent update rule that forms a unimodal distribution of synaptic weights for unstructured inputs while forming a bimodal distribution for spatiotemporally structured inputs.

5.1 Synaptic Drift and spike timings

Within the last 15 years (Markram et al., 2012) the first choice for modelling a dependence on the specific timing of spikes within Hebbian synaptic plasticity rules has become *spike timing dependent plasticity* (STDP) (Bi and Poo, 1998; Gerstner et al., 1996; Markram et al., 1997), which has by now been found to exist throughout the animal kingdom within most areas of the brain that were observed with *in vivo* methods (Froemke and Dan, 2002; Froemke et al., 2005, 2010, 2006) and *in vitro* slices (Froemke and Dan, 2002; Markram et al., 1997) and in specifically prepared cultures of neurons (Bi and Poo, 2001, 1998; Shouval et al., 2010; Sjöström et al., 2001; Wang et al., 2005; Wittenberg and Wang, 2006).

While the underlying biochemical mechanisms are still being debated (Sjöström and Gerstner, 2010) and there are occasionally voiced challenges of single aspects of STDP (Lisman and Spruston, 2005, 2010), there is by now general consensus that the strength of connection between two neurons can be influenced by the precise timing of action potentials arriving at that synaptic connection. Specifically, the arrival time of action potentials in STDP has a profound influence on the *direction* of change to the synapse's connection strength. This is in opposition to traditional Hebbian learning (as the term is used today), where the exact arrival time of pre- and postsynaptic action potentials does

not matter as long as they happen within a window of a few milliseconds from one another. Correlated activity always leads to potentiation within Hebbian models, whereas in STDP the order of firing determines whether the synapse will be potentiated or depressed. Figure 5.1 visualises two typical STDP rules with a jump from depression to potentiation as the time difference between postsynaptic and presynaptic action potential arrivals at a synapse becomes positive at $\Delta t = t_{post} - t_{pre} = 0$ ms.

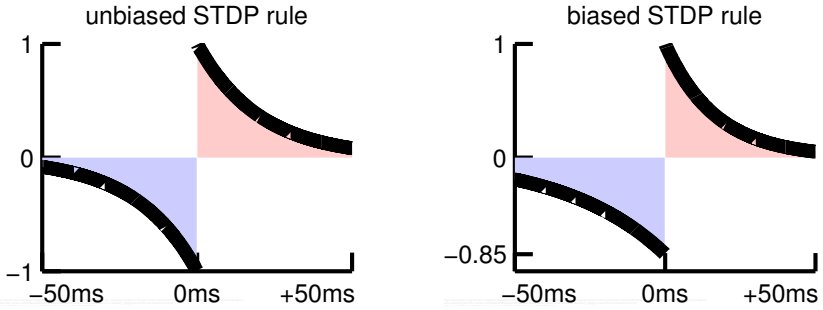


Figure 5.1: *Left:* A simplified unbiased STDP rule with $A_+ = A_- = 1$ and $\tau_+ = \tau_- = 20$ ms. This rule is used in e.g. Rubin et al. (2001); Song et al. (2000). *Right:* A slightly more realistic biased STDP rule with $A_+ = 1$, $A_- = 0.85$, $\tau_+ = 16.8$ ms , and $\tau_- = 33.7$ ms. This rule is similar to the one used in Guyonneau et al. (2005) and based on Bi and Poo (1998). Showing update step sizes for time differences $\Delta t \in [-50, 50]$ ms.

Equation 5.1 shows the mathematical definition of STDP as mentioned before (Section 2.3.3), with A_+, A_- as a constant scale of potentiation and depression, respectively, τ_+ and τ_- as time constants, $\Delta t = t_{post} - t_{pre}$ the timing difference between pre- and postsynaptic spike arrivals at a synapse, λ is a constant that controls the learning rate, Δw the change of synaptic weight, and functions $g_+(w)$, $g_-(w)$ (see Section 5.2) that may or may not scale this change as a function of the previous synaptic weight:

$$\Delta w = \begin{cases} A_+ \cdot \lambda \cdot e^{\frac{\Delta t}{\tau_+}} \cdot g_+(w) & \text{for } t_{pre} < t_{post} \quad \text{(LTP)} \\ -A_- \cdot \lambda \cdot e^{-\frac{\Delta t}{\tau_-}} \cdot g_-(w) & \text{for } t_{pre} > t_{post} \quad \text{(LTD)} \end{cases} \tag{5.1}$$

5.1.1 Bias or no bias?

STDP rules with $A_+ = A_-$ and $\tau_+ = \tau_-$ are useful for demonstrating the principal effects of spike timing on synaptic strength as was done in (Rubin et al., 2001). As both the positive and negative sides of the rule are completely identical in size, this type of STDP shows no bias towards neither potentiation nor depression. In the following, I therefore refer to this group of STDP rules as unbiased STDP. If a given synapse receives many pairs of pre- and postsynaptic spikes with equally probable (=uniformly distributed) time differences, the synaptic strength will fluctuate but prefer no direction of change. It will be performing an unbiased random walk. While this behaviour is nice for demonstration purposes and some less realistic supervised learning tasks (Frémaux et al., 2010), the random walk of synaptic weights for uncorrelated pre/post neurons (see Section 5.2) leads to a problem when trying to perform unsupervised learning as usually found in the brain: The main task of unsupervised learning methods, both in neurologically plausible as well as in abstract machine learning algorithms, is usually described as aiming to learn to recognise structured, reliable information among a haystack of meaningless data (Bishop, 1995; Hinton and Sejnowski, 1999). If a synaptic update rule fails to filter out meaningless inputs that are at all times completely unrelated to postsynaptic activity, the synapse is

effectively just introducing noise into the system and counteracting the goal of making sense of the world around it.

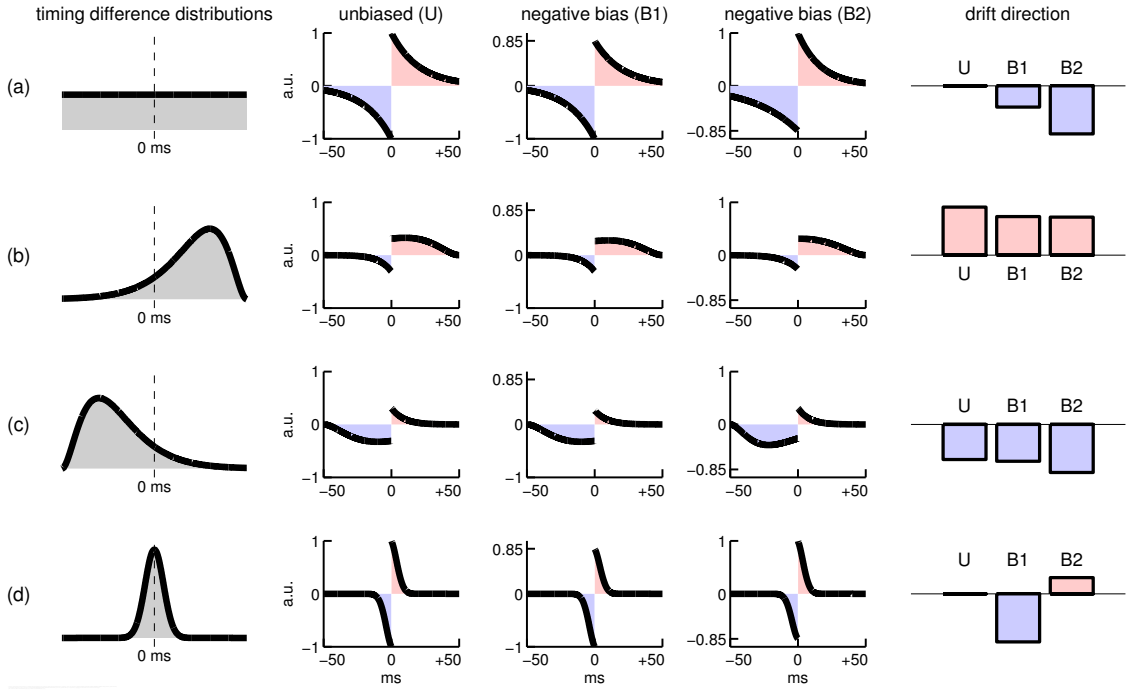
By biased STDP, I refer to the group of STDP rules where the time constants for potentiation τ_+ and depression τ_- or the maximum (normalised) weight update sizes A_+ and A_- are *not* equal ($\tau_+ \neq \tau_-$ or $A_+ \neq A_-$). A rule with a negative drift can be formed by either making the negative time constant larger than the positive time constant or by increasing the negative maximum update size vs. positive, as both increase the size of the negative area of the STDP rule, hence decreasing the STDP rule's integral below zero. Biased STDP rules with positive drift are seldomly used, because a broad increase of synaptic weights without basis in correlated activity can rightfully be considered implausible.

5.1.2 Distributions of pairing probability

As the synaptic drift caused by an STDP implementation has important consequences for the overall success of learning, I close this subsection by taking a closer look at how synaptic drift depends on pairing probabilities for *non*-uniform distributions of spike timing differences. In this broader case, the distribution of spike timing differences becomes an important hint in finding the overall direction of synaptic updates, and the integral of the STDP rule alone is no more sufficient to describe the direction of synaptic drift. Sketch 5.2 shows an overview of how various spike timing difference distributions influence the overall direction of synaptic update steps. While the location of the peak of spike timing differences has an obvious influence on the direction of synaptic updates (Sketch 5.2 b,c), the width of a peak can also affect the direction of synaptic drift if the STDP rule is biased as in Figure 5.1 (see Sketch 5.2 d).

When independent neurons are given random spikes to elicit, i.e. the occurring spike timing differences are uniformly distributed (see Sketch 5.2 a), a directed drift of synaptic strengths occurs for STDP rules that are biased towards either depression or potentiation. In this case, the direction of this drift only depends on the full integral of the (all-to-all) STDP rule. Keep in mind that this integral *can* be negative even if either $|A_+| > |A_-|$ or $|\tau_+| > |\tau_-|$, but not both. This is the case for the negatively biased STDP rule in Figure 5.1, where the rule's integral is negative despite A_+ being larger than A_- (see also Table A.4, p. 181). This rule is also used as the third STDP rule shown in Sketch 5.2 (rule B2).

Throughout this text, we assume that each spike is allowed to interact with every other existing spike (all-to-all spike pairing). If, however, some combinations of spikes are given a stronger influence in changing the synaptic strength (e.g. nearest neighbour pairing (van Rossum et al., 2000), spike suppression (Froemke and Dan, 2002; Wang et al., 2005), or triplet rules (Pfister et al., 2006)), the effective time window and windowed integral of the STDP rule may in addition also change with the firing rates of the neurons that the synapse connects. As an example, see row *d* of Sketch 5.2, where all spike timing differences happen within close proximity (e.g. $\Delta t \in [-5, 5]$ ms). The distribution of spike timing differences is no more uniform. Within this short interval, the windowed STDP integral of the biased rule B2 is now positive. So when spike pairs often arrive within very short intervals, the synaptic drift can become positive for this rule. This can easily happen when nearest-neighbour spike selection is used together with high firing rates. But it can also happen for all-to-all STDP when pre- and postsynaptic neurons always fire within very short intervals of each other. A typical case in which this happens is when a postsynaptic neuron is repeatedly presented with a highly correlated, but still slightly jittered, pattern of inputs from multiple presynaptic neurons. The effect of presenting near-synchronous inputs to a freely responding postsynaptic neuron will be explored in Chapter 6.



Sketch 5.2: **Influence of spike timing difference distributions and selected STDP rules on overall drift direction of the synaptic weight.** Windowing the STDP rule by the probability distribution of spike timing differences hints at whether a synapse will be overall strengthened or weakened or whether it performs a random walk with no tendency towards either direction. Here, the effect of four types of timing difference distributions (rows *a* - *d*) is tested on an unbiased (U) and two biased (B1, B2) STDP rules, as described in Appendix A.3.2, Table A.6 (p. 181). (a) For uniform pairing probability distributions (uncorrelated poisson spikes), the sign of the STDP rule's integral directly gives the mean direction of synaptic updates, or *synaptic drift*. The windowed STDP rules have the same shape as the original rules. (b) Positive-leaning timing difference distributions (correlated poisson spike pairs with mostly right-shifted, positive lag) tend to have spikes of the postsynaptic neuron shortly after those of the presynaptic neuron, thereby pronouncing the effect of the right side of the STDP rule. This may indicate a *causal* relationship between the two neurons (standard causality warnings apply). For the typical STDP rules examined here (sketched in row *a*), such a "causal" relationship between neurons produces an overall potentiating synaptic drift, even when the integral of the STDP rule is negative (rules B1, B2). A sketch of the combined influence of each STDP rule windowed by the pairing distribution is shown in the three central columns. (c) Negative-leaning timing difference distributions (correlated poisson spike pairs with mostly left-shifted, negative lag) tend to instead promote the left side of the selected STDP rule. This produces an overall depressing synaptic drift for all STDP rules shown here, including the otherwise unbiased STDP rule. This type of correlation between two neurons *may* respectively indicate an *anti-causal* relationship between the two neurons. (d) For strongly correlated neurons with very narrow zero-lag correlation, differences around the centre of an STDP rule have a stronger effect on the direction of change than the overall integral has. While the unbiased rule here again shows no specific drift direction tendency, the two negatively biased rules behave very differently (column *"drift direction"*): Although rule B1 maintains its depressing drift even when most pairings only happen in a very small window around the rule's centre point (0 ms), rule B2 now produces potentiating synaptic drift even though its *full* integral is negative. Isolated consideration of an STDP rule's integral is therefore a bad predictor for the overall direction of synaptic change if the distribution of pairings is not uniform, as in any real-world scenario. The (typically varying) shape of correlation between STDP-equipped neurons must also be accounted for when assessing whether a specific STDP rule will lead to depression or potentiation within a spiking network. As we will see in Chapter 6, it is also not sufficient to reduce this complexity to steady-state correlations while working with more realistic, precisely timed spike patterns.

5.2 Weight Dependence of synaptic updates

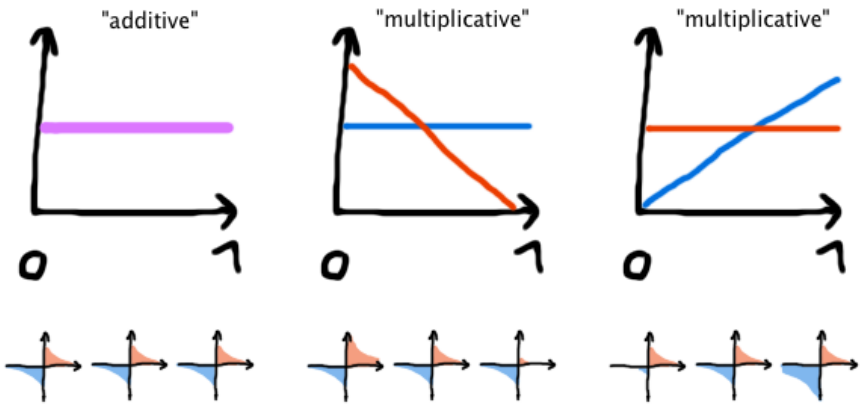
The direction of synaptic drift need not be fixed to a constant value. Many models of STDP include a dynamic term ($g_{\pm}$) that scales potentiating and/or depressing updates with the current strength of a synaptic connection. This term behaves similar to A_+ and A_- in that it multiplies all changes by some (now dynamic) factor. It never affects the time constants τ_+ and τ_- in any of the standard STDP models reviewed below.

One recurring aim of weight-dependent STDP rules is to keep simulated connection strengths within some predefined biophysical boundary. While the lowest possible strength of a synaptic connection is logically zero, it is still unclear what the maximum strength of any given synapse is. It seems likely that different types of neurons in different brain areas and with different morphology have different maximal strengths, so the only reliable statement that can be made today is that no infinitely strong synaptic strength can likely exist (i.e. there *exists* some upper bound per synapse).

I will therefore first give a review of existing weight-dependent STDP rules, and then suggest my own additions to this set. I also suggest a new classification of STDP rules that avoids the terms *additive* and *multiplicative* as described below.

5.2.1 Additive vs. multiplicative STDP weight dependence

The first ad hoc idea for applying a series of synaptic updates to changes of the connection strength between two neurons – i.e. the synaptic weight – would be to simply sum up all of these updates and add them to the previous weight of the synapse. This works well at first, but soon the synaptic connection strength may accidentally drift below zero or above some biophysically plausible maximum value. To avoid this, the ad hoc solution is to keep the synaptic weights within a defined range through clipping: After each change to a synapse's strength, its value is checked for whether it has left the defined range and, if needed, artificially cut to remain within that range. This rule for updating synaptic strengths and keeping them within a defined range is called *additive* STDP because synaptic updates are summed up linearly. Additive rules look as if their updates don't depend on the current



Sketch 5.3: Typical "additive" STDP rule (Eq. 5.2) vs. two types of so-called "multiplicative" STDP rules (Eqs. 5.4 and 5.5). Red: $g_+(w)$, Blue: $g_-(w)$, Purple: $g_+(w) = g_-(w)$. The second row shows sketches of how the effective STDP rule changes due to weight-dependent scaling of its timing-dependent update steps. More information in Figures 5.6 and 5.7

weight of a synapse (see Sketch 5.3, left):

$$\begin{aligned} g_+(w) &= const, \\ g_-(w) &= const \end{aligned} \tag{5.2}$$

This is not entirely true, as the clipping effectively introduces this dependence as a rectangular window over the rule's defined range, with strong steps to zero at the hard boundaries. The horizontal line in Figure 5.3 (left) can therefore also be seen as the visible part of a rectangular window function with hard jump from 1 to 0 at the edges of the defined range. Equation 5.3 describes this a little better[1], with $rect(w) = 1$ for $w \in [0, 1]$ and $rect(w) = 0$ otherwise.

$$\begin{aligned} g_+(w) &\propto rect(w), \\ g_-(w) &\propto rect(w) \end{aligned} \tag{5.3}$$

Obviously, such a hard bound on synaptic strengths can be called into question as being biologically not very plausible. In a biological synapse, one would expect a more gradual approach towards the edges of some range of synaptic weights. And indeed, early work on STDP by Bi and Poo (1998) in hippocampal neuron cultures has found that the change to synaptic strength correlates with the electric current that a synapse initially evoked in the postsynaptic cell. This initial excitatory postsynaptic current (EPSC) was measured before execution of 60 forced pairings of pre- and postsynaptic spikes in a patch clamp setup of neurons grown in a petri dish (*in vitro*) and compared to the EPSC after pairing. The authors plot the changes of EPSC amplitude in a potentially misleading style, though, as is also pointed out by (Billings and van Rossum, 2009). By displaying only the percentages of EPSC change relative to initial EPSC (see Figure 5.4), they make the data look at first sight as though depressing update steps did not depend on initial synaptic strength, while in reality they strongly depend on this initial strength in a linear relationship. The presentation style also makes it look as if the relationship between potentiating updates and initial synaptic current were inversely linear, while in reality only the relative change as a percentage of initial synaptic strength decreases. The absolute step size of potentiating synaptic updates still keeps increasing with initial EPSC, as will be further explained in section 5.2.4.

A first, maybe naive, weight-dependent bounding rule may therefore have an inverse linear relationship of potentiation to the initial weight, while the amount of depression may not depend on the present synaptic strength (see also Sketch 5.3, centre):

$$\begin{aligned} g_+(w) &= 1 - w, \\ g_-(w) &= const \end{aligned} \tag{5.4}$$

The potentiation bound $g_+(w)$ of this rule is the same as used in the work of (Kistler and van Hemmen, 2000) and (Rubin et al., 2001). The depression bound $g_-(w)$ of Equation 5.4 is independent of synaptic weight as a contrast to Equations 5.5 and 5.6 (see below). Section 5.2.3 will examine the effects of this bounding rule on synaptic drift, resulting weight distributions, and implications for performance in learning of structured data for both unbiased and biased STDP, and show simulation results for each in Figures 5.6 and 5.7 (column B).

[1] Keep in mind that an implementation within a digital computer may require an additional implementation detail for handling synaptic update steps that happen very close to the edges of the defined range. This is only required due to the discrete nature of synaptic updates in computer simulations.

An alternative multiplicative rule was used by van Rossum et al. (2000) and Billings and van Rossum (2009). While noticing the misleading presentation of data in (Bi and Poo, 1998), the authors here simply switch the single-sided weight dependence from potentiating to depressing updates, thereby forming the following bounding rule (see also Sketch 5.3, right):

$$\begin{aligned} g_+(w) &= const, \\ g_-(w) &= w \end{aligned} \tag{5.5}$$

While the linear dependence of depressing weight updates is indeed captured by this rule, it fails to correctly represent potentiating updates according to the data of (Bi and Poo, 1998), as will be discussed in section 5.2.4. Van Rossum et al. (2000) argue that this rule is valid because it can reproduce unimodal weight distributions measured in neurons grown in vitro, when it is given Poisson-distributed random presynaptic inputs to learn. The learning performance is, however, called into question by the same group in (Billings and van Rossum, 2009), and reasserted through careful parameter tuning in (van Rossum et al., 2012). I will present a few thoughts on plausible learning performance in section 5.2.3. It is also questionable whether their definition of what constitutes a useful pattern for STDP learning is indeed a valid assumption, as I will show in the next chapter (Chapter 6).

Of course the exact shape and time dependence of any experimentally found STDP rule depends on the experimental conditions under which the measurements of spike timing dependent plasticity take place. Any general rule of STDP will always be an abstraction from reality. It does seem physically plausible, though, that the size of synaptic modification steps indeed depends in some way on present synaptic strength as an indicator of past activity of the synapse. In addition to synaptic strength, updates to a synapse likely also depend on a combination of many other variables such as concentrations of chemical messengers in the surrounding tissue. While the effect of one neuromodulatory chemical messenger on STDP outcome and network behaviour (i.e. dopamine) will be investigated in Part III of this text, I continue to concentrate here on the local effect that recent history of synaptic activity may have on local timing-dependent plasticity.

In spite of the claimed biologically plausible weight distributions for unstructured input data and perhaps because of the bad learning performance of "multiplicative" weight

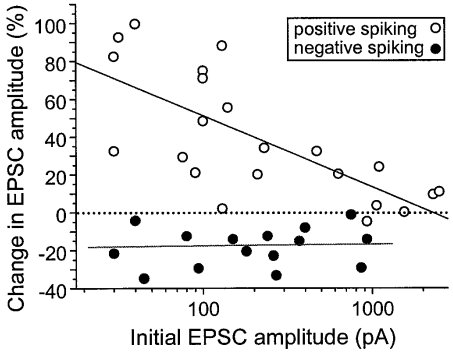


Figure 5.4: Relative change of excitatory postsynaptic current (EPSC) over initial EPSC size. Taken from Bi and Poo (1998, Fig. 5). Plotting relative change instead of absolute change in EPSC amplitude has caused some confusion according to (Billings and van Rossum, 2009; Morrison et al., 2007; van Rossum et al., 2000). Also, the horizontal axis is logarithmic while the vertical axis is not. Compare also Figures 5.8, 5.10, and 5.11.

bounding rules (Billings and van Rossum, 2009) in comparison to "additive" rules, much of the STDP modelling literature keeps using weight-independent synaptic update rules, and uses hard clipping to keep synaptic strengths within the borders of their weight ranges. This has led Gütig et al. (2003) to approach the problem by interpolating between additive and multiplicative rules through a power law controlling switch.

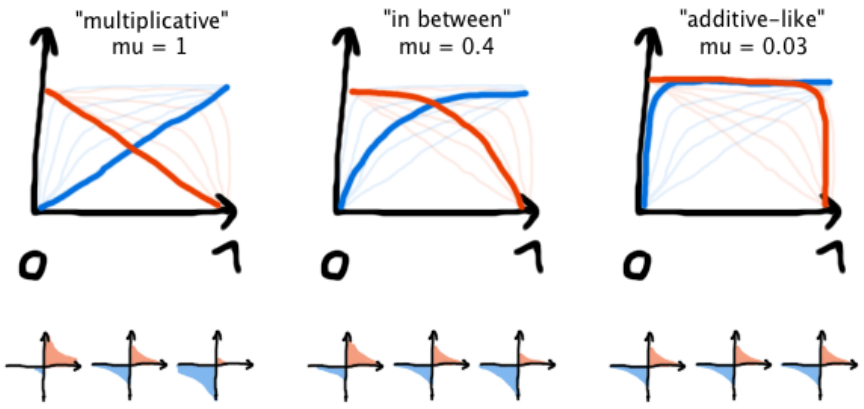
5.2.2 Interpolating between traditional bounding rules

As a consolidating action for combining the ascribed biological validity of multiplicative rules with the superior learning performance of additive STDP, Gütig et al. (2003) had the idea of interpolating between these two general types of weight update rules through an exponential parameter that controls the update rule's dependence on synaptic weight. By aiming at the update rules of Kistler and van Hemmen (2000) and Rubin et al. (2001), Gütig et al. (2003) could smoothly transition from "additive" to "multiplicative" behaviours by using an exponent $\mu \in [0, 1]$ (see also Sketch 5.5):

$$\begin{aligned} g_+(w) &\propto (1-w)^\mu, \\ g_-(w) &\propto w^\mu \end{aligned} \tag{5.6}$$

In their publication, (Gütig et al., 2003) also used a parameter α to scale the maximal amount of possible synaptic depression in relation to potentiation, effectively causing the unbiased STDP rule they used to become biased through constant scaling and cause a (negatively) directed synaptic drift. As multiplicative rules form a peak in the distribution of synaptic weights around some central value (as also seen for Equations 5.4 and 5.5 in Figure 5.6), the α parameter effectively controls the location of this peak. This behaviour is captured within the ratio between A_+ and A_- in the STDP rules used here, and therefore we do not require a parameter α to exist separately.

The Gütig et al. (2003) rule shows many interesting behaviours for random Poisson inputs, but the main statement is that with careful tuning of μ, one can let a group of synaptic weights form a bimodal distribution without the need for hard bounds at the edges of the range of possible synaptic weights. Gilson et al. (2010) successfully used this rule with $\mu = 0.03$ to obtain an "additive-like" rule that allows synaptic differentiation



Sketch 5.5: Transition from so-called "multiplicative" STDP rules to "additive" rules by tuning a single parameter μ. Red: $g_+(w)$, Blue: $g_-(w)$. The second row shows sketches of how the effective STDP rule changes due to weight-dependent scaling of its timing-dependent update steps. More information in Figures 5.6 and 5.7

across most of the allowed weight range, which never reaches the hard edges as long as some form of noise is also present in the inputs. As the authors use correlated inputs (but without any spatiotemporal structure), the inputs inherently contain noise which is reflected in the distribution of response times by any postsynaptic neuron responding to these correlated inputs (see Section 6.1, p. 71).

A motivation for avoiding pure additive STDP is given in the form of correlation-dependent learning success (see also Rubin et al., 2001). This states that additive rules are bad because they tend to form binary distributions even for minimal amounts of correlation, where the identities of strong synaptic weights are randomly selected. And that multiplicative rules are better because they represent the level of input correlation as a shift in the peak of the resulting weight distribution as compared to the peak for uncorrelated inputs. In multiplicative rules, the amount of correlation within a set of input units can indeed be tracked by measuring how far the resulting connection strengths can escape the counteracting synaptic drift towards some stable attractor (Morrison et al., 2008). I will demonstrate in chapters 6 and 7 (and Figures 5.6, 5.7) that this may not be the best performance metric for a synaptic learning rule, and learning may actually be more successful if a plasticity rule tries to detect even the smallest correlations or recurring structure among input data that it can still find among very noisy surroundings. In sections 5.2.7 (p. 62) and 5.3 (p. 65) I will look at how experimentally observed weight distributions may alternatively be formed.

Another interesting note by (Gütig et al., 2003) is made on how two groups of inputs that show some amount of inner-group correlation compete for control over a postsynaptic neuron. As correlation increases, one group of inputs may win over the other. This is expressed by the input units of one group forming stronger synaptic connections to the postsynaptic cell than the input units of the other group. But as correlation within each group is further increased, competition seems to be reduced again as both groups then form strong connections to the postsynaptic neuron. The neuron learns to react to both groups. The correlation values at which these splits occur are influenced by the chosen value for μ, among a few other parameters.

I will now take a look at some features of additive and multiplicative STDP, and their performance for a set of input patterns that will itself be further explored in chapter 6.

5.2.3 Performance and Features (Set 1)

Let us examine the effects of the previously introduced bounding rules on unbiased and biased STDP and how they work in keeping the synaptic weight within some predefined range of minimum and maximum values. The minimum value of connection strength between two neurons can easily be suggested to be zero, or *not connected*. The maximum connection strength, however, may be a bit harder to define, as the factors that determine a synapse's maximum strength are not yet fully known. They most likely depend on chemical signalling, homeostasis, energy availability, spinal proximity due to random structural growth, and only partially on the synapse's electrical activity. Each synapse of a given neuron may therefore have an individual maximum strength, which becomes important to keep in mind when measuring existing distributions of synaptic connection strengths. Such distributions were measured through visual inspection of the size of spines by Turrigiano et al. (1998) or O'Brien et al. (1998), but neither publication claims that the measured unimodal weight distributions were due to timing-dependent synaptic plasticity. However, van Rossum et al. (2000) uses this as proof that spike timing dependent plasticity should result in unimodal distributions of synaptic weights. I will give an explanation for why I object to this view in section 5.3.

For the following comparison, synaptic weights are normalised to a common minimum of 0 and maximum of 1, as in the sketches above (Sketches 5.3 and 5.5), yielding a common range for possible synaptic strengths $w \in [0, 1]$.

Visualisation of Effects

In order to provide a comparative overview of the functional differences between the reviewed STDP rules, I plot a number of their features in Figures 5.6 and 5.7.

1. I first show the normalised scaling $g_{\pm}(w)$ of weight-dependent updates for long-term potentiation (LTP, *red*) and long-term depression (LTD, *blue*) in the first row of those figures. This presentation style is the same as used in sketches 5.3 and 5.5, with *purple* colour indicating rules in which the weight-dependent scaling of LTP and LTD is equal ($g_+ = g_-$). Some plots in the first row may also include narrow dashed lines in addition to continuous lines. This is done to visualise the effects of different parameter settings on the shape of g_+ and g_-, and only the parameter settings that produce the continuous lines are used for the following rows.

2. As the actual shape of most STDP rules changes with the strength of the synapse in question, I show five snapshot sketches as examples of scaled rules for five different values of the synaptic weight w, respectively ($w \in \{1, 2, 3, 4, 5\}/6$). This is done for each of the four main columns (second row in the figures).

3. Although the visualisation in the second row gives us some idea on how the weight-dependent scaling factors $g_+(w)$ and $g_-(w)$ change the effective shape of an STDP rule, we can also re-plot the rules to obtain some intuition on how they change continuously with the synaptic strength. The third row plots the timespan $[-50, 50]$ ms from the second row as single columns (vertical axis), over a continuous gradient from $w = 0$ to $w = 1$ (horizontal axis).

4. The fourth row contains randomly selected synaptic updates for each STDP rule, taken from a uniform distribution of timing differences, and repeated 100 times (vertical axis). As the shape of most STDP rules changes with w, these random pairings may have a tendency for potentiation or depression, depending on the initial synaptic strength w (horizontal axis). This row demonstrates how, while the actual step size and direction is grainy and stochastic (potentiating updates shown in *red*, depressing updates in *blue*), synaptic drift can be seen by blurring (locally averaging) the view. This may help the reader to intuitively grasp the nature of synaptic drift.

5. The fifth row gives a closer view of overall drift tendency, or averaged weight updates from the fourth row (black curve), together with standard deviation (yellow curve, scaled by 0.2). Potentiating drift (positive sign of mean step size) is shown as red area and depressing drift (negative sign of mean step size) is shown as blue area. Mean values were computed for each of 200 weight bins from 20000 repetitions per bin.

6. While the first five rows show didactic examples of the reviewed STDP rules for supporting the reader's understanding, the sixth row tracks the development of the distribution of synaptic weights of a single postsynaptic neuron in a computer simulation. Simulation details are given in Appendix A.3.3 (p. 178). Postsynaptic membrane potential here does *not* yet depend on presynaptic inputs (as it will in later chapters), but is set such that the postsynaptic neuron fires at a fixed rate of

1 Hz. This row makes the synaptic drift of the connections of 2000 uncorrelated poisson-distributed presynaptic input units visible (grey). The postsynaptic neuron is also connected to a further 1200 presynaptic input units, of which 600 units present a spatiotemporally structured pattern just *before* each postsynaptic spike (selected synapse trajectories shown in red), and the other 600 input units present a spatiotemporal pattern just *after* each postsynaptic spike (shown in blue), in addition to random spikes as in the first 2000 units. A discussion of why any STDP rule should also be tested with spatiotemporally structured patterns is the main topic of Chapter 6. The exact structure of spatiotemporal inputs used here can be seen in Figure A.7 (p. 181).

7. The last row plots the resulting weight distribution of all synapses after 20min of simulation. The distribution of only the 2000 uncorrelated input units is shown in grey, while the distributions of the two groups presenting spatiotemporal patterns are shown in red and blue. Depending on the STDP rule and weight-dependent scaling used, these three distributions can be very different in shape and location.

Unbiased STDP as used in Figure 5.6 has no inherent directed drift, as $A_+ = A_- = 1$ and $\tau_+ = \tau_- = 20$ms. Any drift is therefore only due to $g_+(w)$ and $g_-(w)$, which are defined by one of Eqs. 5.3, 5.4, 5.5, 5.6. Biased STDP as used in Figure 5.7 has an asymmetry both in $A_\pm$ and in $\tau_\pm$ ($A_+ = 1 > A_- = 0.85$ and $\tau_+ = 16.8\text{ms} < \tau_- = 33.7\text{ms}$). The asymmetry in the time constants $\tau_\pm$ has a stronger effect here (compare Sketch 5.2, top row), so the overall drift direction of this STDP rule is negative, leading to a depression of all connections to uncorrelated inputs. The effects of $g_+(w)$ and $g_-(w)$ on biased STDP will also be discussed below.

Comparison of scaling rules for unbiased STDP

The first row of Figure 5.6 shows that for additive STDP (column A) and additive-like STDP (column D) the weight-dependent scaling factors $g_+(w)$ and $g_-(w)$ are identical or at least very similar throughout most of the range of weights, respectively. This is in strong contrast to typical multiplicative STDP rules (columns B,C), where the weight-dependent scaling factors are very different for LTP (scaled by g_+) and LTD (scaled by g_-) for any given synaptic strength w, except for a single crossover point (here at $w = 0.5$). Below this point within the range of synaptic weights ($w < 0.5$, left side within each plot), the scale for potentiating updates is larger than the scale for depressing updates. Above this point ($w > 0.5$, right side within each plot), this relation is inverted, and depression dominates. When the distribution of times at which input units fire a spike in relation to the time of postsynaptic response is sufficiently uniform, this difference in weight-dependent step sizes can be shown to form a *stable fixed point attractor* (Morrison et al., 2008) that keeps the strength of synaptic connections w close to this crossover point. The existence of a strong stable fixed point is a defining feature of multiplicative STDP rules, while the absence of a strong stable fixed point is said to indicate additive STDP rules (Morrison et al., 2008). In additive-like STDP (column D), there still exists a very weak fixed point as there still is a crossover between g_+ and g_-, but it is so weak that it has little effect. In theory, it would also be possible to create an *unstable* fixed point that pushes all synaptic strengths towards the edges of the weight range by inverting the ratio between g_+ and g_-. But as the resulting distribution of synaptic weights would not represent any meaningful information relative to a neuron's inputs, this approach rightfully has not been explored in the literature.

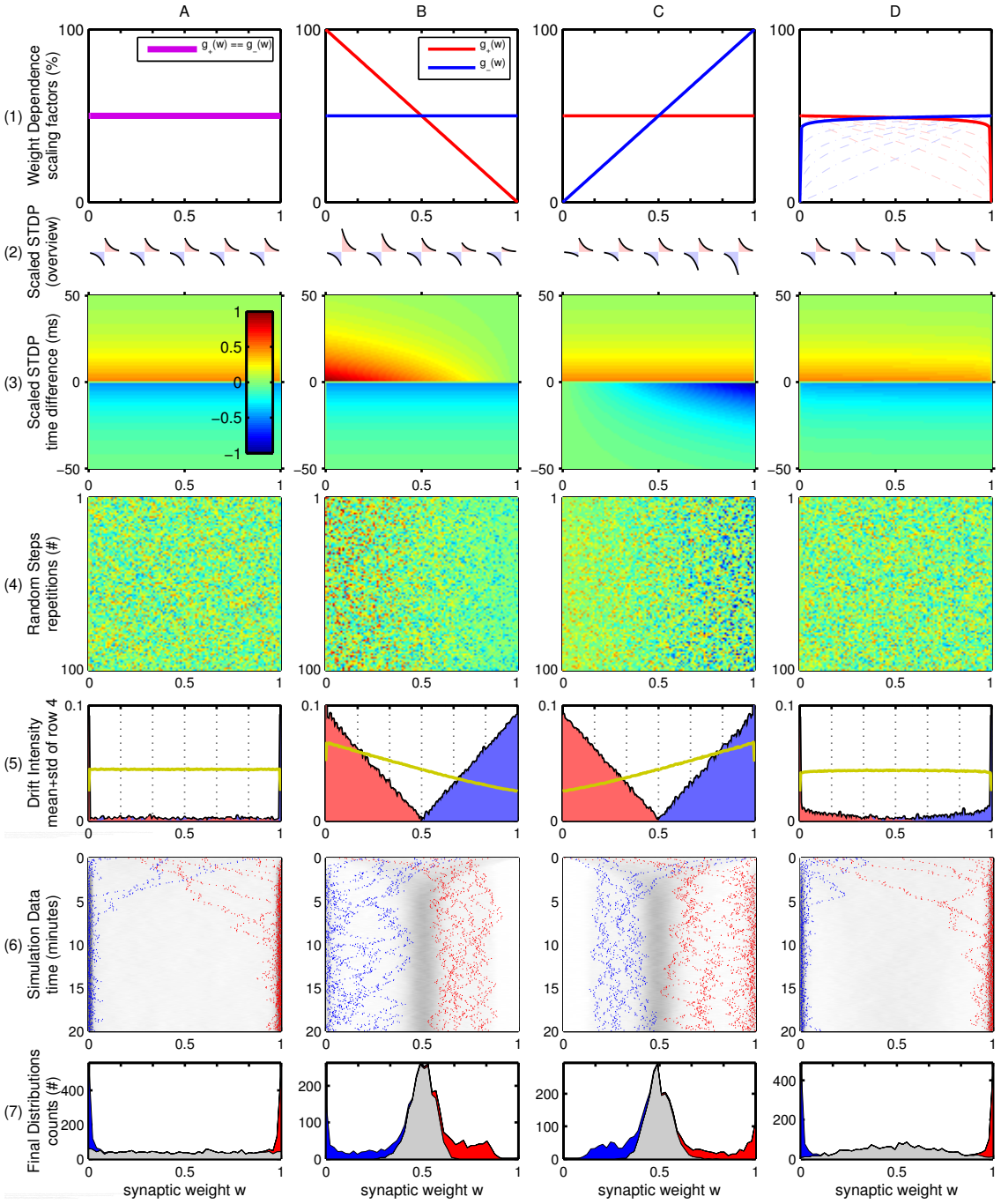


Figure 5.6: Comparison of effects of weight-dependent scaling rules on unbiased STDP. *Column A*: Additive STDP (Eq. 5.3). *Column B*: Weight-Dependent LTP (Eq. 5.4). *Column C*: Weight-Dependent LTD (Eq. 5.5). *Column D*: Interpolated power law (Eq. 5.6). – *continued on next page...*

Figure 5.6 *(previous page)*: Description of rows (see also *Visualisation of Effects*, p. 47): *Row 1:* Visualisation of weight-dependent scaling rules. Red lines: scale factor $g_+(w)$ on potentiating steps (LTP). Blue lines: scale factor $g_-(w)$ on depressing steps (LTD). Magenta lines: identical (balanced) dependence on synaptic weights for both LTP and LTD ($g_+ == g_-$). *Row 2:* Example effects of weight-dependence on the shape of STDP rules for weights $\frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}$. Note how the actual shape of the STDP rule changes with the synaptic weight. *Row 3:* Full overview of STDP weight changes across the continuous range of possible synaptic weights, within a timing window of [-50,50] ms. *Row 4:* Random pairings of spikes arriving at a given synapse reflect the relation of large to small weight update steps. The weight-dependent tendency for potentiating vs. depressing weight update steps also becomes graspable here. *Row 5:* Mean weight update from row 4 (black line) together with scaled ($\frac{1}{5}$) standard deviation of step size (yellow line) for defined range of weights. Potentiating drift (positive sign of mean step size) shown as red area. Depressing drift (negative sign of mean step size) shown as blue area. *Row 6:* Example simulation with the given STDP rule and given weight bounding rule. Weight distribution estimates for each second of simulation shown as rows of pixels in grey (background). Example random walks and biased drift of causally (red) and anti-causally (blue) firing input units overlaid for clarity. See Figure A.7 for simulation details. *Row 7:* Resulting distribution histograms after 20 minutes of simulation. Synaptic weights to uncorrelated inputs (relative to postsynaptic firing) are shown in grey, weights to positively shifted correlated inputs ("usually before") shown in red, and weights to inputs with negatively shifted correlations to postsynaptic firing ("usually after") shown in blue. Histograms stacked in upward order: grey → red → blue.

In the second row, one can see how multiplicative rules (columns B,C) cause the overall shape of the full STDP rule (especially the ratio between potentiation and depression) to change remarkably with synaptic weight, while this shape does not change or hardly changes for additive and additive-like rules, respectively (columns A,D).

This is further visualised in row three, where the full scaled STDP rules are re-plotted to show not just the effect of the weight-dependent scaling factors g_+ and g_-, but also the influence of spike timing differences on the resulting change to a synaptic weight. In many publications on STDP (Gütig et al., 2003; Rubin et al., 2001; van Rossum et al., 2000), the effect of timing difference is averaged out by using (uncorrelated) poisson-distributed inputs without spatiotemporal structure. Even when unstructured correlation between input units is used (Gütig et al., 2003; Morrison et al., 2007; Rubin et al., 2001; van Rossum et al., 2000), some of the temporal discriminative power of STDP remains unused, as I will show in Chapter 6. There, I will also address the ongoing debate on whether spatiotemporal structure within biological neuronal networks should be ignored due to the unreliable predictability of single spikes or be included due to the possible computational power and simple existence of time-discrete spikes. But for now, the third row should simply give the reader an intuitive idea of how synaptic weight changes in classical (pair-based) STDP rules always also depend on spike timing differences, whether there is weight-dependent scaling (columns B,C) or not (columns A,D).

The fourth row demonstrates the stochastic nature of spike timing dependent updates to synaptic weight. While the ratio between potentiating and depressing updates seems largely uniform for additive and additive-like STDP (columns A,D), one can already see a slight dominance of potentiating updates for weak weights (column B) and a slight dominance of depressing updates for strong weights (column C) for multiplicative STDP.

This tendency is further visible by taking the mean of a large number of updates in row five, where multiplicative STDP (columns B,C) shows a strong positive drift for weak weights and a strong negative drift for strong weights. As the synaptic connection strength becomes closer to the crossover point described above, the drift intensity weakens, and only reaches zero at one weight position. This is an indication of the stable fixed point attractor mentioned above. The interested reader is referred to Morrison et al. (2008) for an analytical approach to computing the fixed point for inputs that do not contain any

spatiotemporal structure. It should also be noted that the standard deviation of update step sizes changes, depending on which multiplicative STDP scaling rule is used (compare columns B,C, row 5, yellow lines). This is a direct result of the shape of g_+ and g_- as plotted in the first row.

Rows six and seven show the results of example simulations with the weight-dependent scaling used in columns A-D. As mentioned above, the exact simulation parameters can be found in Appendix A.3.3 (p. 178). The main point here is that all simulation conditions are exactly equal, except for the different weight-dependent scaling rules used in each column. Both of rows six and seven show a very different behaviour between the additive and additive-like rules (columns A,D) on one hand and the multiplicative rules (columns B,C) on the other.

The distribution of synaptic connection strengths that connect the postsynaptic neuron to uncorrelated, poisson-type input units (grey area) is uniform for additive STDP (column A), while connections to the same units form a definite unimodal distribution for multiplicative STDP (columns B,C). The uniform distribution of synaptic weights for uncorrelated inputs (column A) is achieved when the postsynaptic neuron is not allowed to become correlated with presynaptic units, i.e. when the time at which the postsynaptic neuron fires does not depend on presynaptic activity. If this were different, and the postsynaptic neuron were allowed to respond directly to presynaptic inputs, the postsynaptic neuron would begin to tune to a randomly chosen, minimal subset of inputs, the size of which depends on maximum projection strength and size of the input population (Gütig et al., 2003; Rubin et al., 2001). As stated in Gütig et al. (2003), the choice of a subset of inputs only works reliably for very small populations of input units, where each unit of the input population has a comparably high influence on causing a postsynaptic depolarisation. As this effect vanishes for input populations larger than a few hundred units, I consider it an artefact of too small network sizes and at best only one part of the full story. In an alternative view, the formation of strongly bipolar weight distributions in these publications can be interpreted as a desperate attempt of STDP to identify small groups of recurring spatiotemporal patterns within its inputs (see Chapter 6).

Additive-like STDP (column D) here shows a behaviour that is similar to additive STDP (column A), but differences in the distribution of connection strengths to uncorrelated inputs (grey area) can be seen. As additive-like STDP still has a very weak stable fixed point, the final distribution (grey area, row 7, column D) is not perfectly uniform, but shows a slight peak near this attractor. However, this peak is much less pronounced than the grey peaks in columns B and C. Additive-like STDP also allows bimodal peaks near the edges of the defined range (red and blue areas, row 7, columns A,D), making it indeed more similar to additive STDP than to multiplicative STDP.

We should now discuss what produces the red and blue areas in the histograms of row 7 (and the red and blue dotted curves in row 6). As noted in *Visualisation of Effects* (p. 47), red dotted lines (row 6) and areas (row 7) denote the development (row 6) and final distribution (row 7) of connections to a group of 600 input units that always fire *before* a postsynaptic spike, in addition to random activity. Similarly, blue dotted lines (row 6) and areas (row 7) denote connections to input units that always fire at least *after* the postsynaptic spike. A visualisation of all inputs is shown in Figure A.7 (p. 181). We first look at additive (column A) and additive-like (column D) STDP. While inputs that are uncorrelated with postsynaptic activity form a uniform or near-uniform distribution of synaptic weights (grey areas), the connections to input units that have a higher probability of firing before each postsynaptic spike become strong and gather near the maximal edge ($w = 1$) of the defined weight range. Similarly, connections to input units that tend to fire

after a postsynaptic spike become weak and gather near the minimal edge ($w = 0$). This is expected behaviour for additive STDP, and can be understood by looking at Sketch 5.2, rows *b* and *c*. For multiplicative STDP (columns B,C), the story is quite different. The synaptic weights of the *before* and *after* groups do not gather at the edges of the defined weight range, but also don't take part in forming the unimodal distribution as seen by uncorrelated inputs. Instead, the weights of these two groups (red and blue areas) seem to disperse more smoothly on both sides of the grey area. Also, in column B, no synaptic connection of the *before* group (red area) comes near to the maximum weight ($w = 1$) and, in column C, no synaptic connection of the *after* group (blue area) reaches the minimum weight ($w = 0$). So why is this happening?

It of course has something to do with the firing times of each input unit in the two groups, in relation to the timing of postsynaptic spikes. As can be seen in Figure A.7 (p. 181), some input units in each group reliably fire in close proximity to each postsynaptic spike, while others fire at an absolute time difference of up to 55ms from the postsynaptic spike time. This means that due to the two exponential terms in standard STDP rules (one for potentiation, one for depression) the size of reliably occurring directed weight changes is larger for input units that tend to fire in close timing proximity to postsynaptic spikes, while the size of reliable weight changes is smaller for input units that only produce reliable spikes more distant in time. However, input units don't only fire precise, reliable spikes. As also shown in Figure A.7 (p. 181), they are also noisy. The random spikes of these units induce the same drift as those of fully uncorrelated input units (rows 1-5), which in the case of multiplicative STDP (columns B,C) causes an attraction towards the central fixed point. Reliably timed spikes of a single input unit hence compete with randomly timed spikes over control of the synaptic strength.

Synaptic weights of input units that fire reliably either before or after a postsynaptic spike do not gather at the edges of the defined weight range because the counteracting weight-dependent drift (see row 5) becomes increasingly intense as the weight moves away from the stable fixed point. The increasing pull towards the stable attractor begins to outweigh any directed weight changes coming from precise, recurring timing differences. If the average effect of precisely-timed spikes is weaker than the maximal counteracting drift intensity on the way to the edges, the weight of this connection remains at an intermediate level between the stable attractor and the edge of the defined range with some amount of fluctuation. Depending on the standard deviation of random update steps (row 5, yellow line) at this intermediate level, the synaptic weight may fluctuate more or less.

Due to the pull of multiplicative STDP rules, small update steps induced by precise timing are made undone by the intermittent random pairings of uncorrelated noise. A number of publications that advocate multiplicative STDP (Gilson et al., 2010; Gütig et al., 2003; Morrison et al., 2007; van Rossum et al., 2000) say that this competition is good because the distribution of weights represents the level of correlation in the input data (see also Section 5.3, p. 65). The question here is, though, whether a smooth representation of input correlations is really the main or only goal of biological neurons or whether they may rather aim to tune themselves to the smallest amount of structured information they can possibly identify. From the perspective of unsupervised (machine) learning, it would appear to make more sense for a synapse to be able to form maximally strong connections to repeatedly firing inputs that contain some recurring structured pattern in order to improve detection quality and detection time instead of trading this in for incomplete representations of input correlation. In a noisy environment, it will always be difficult enough to filter out "meaningless" noise among the input data, and even for an STDP rule without stable attractor such as additive STDP (column A), the resulting

distribution in a real-world application will be less binary as the signal-to-noise ratio is less than optimal in difficult real world pattern detection problems.

There is another issue with the weights never reaching their possible maximum value (column B, row 7) or never reaching zero (column C, row 7) in multiplicative STDP. While the avoidance of maximum values may not be such a problem and is even referred to as one of the main features of multiplicative STDP rules by van Rossum et al. (2000), the inability to reach zero synaptic strength if a synaptic connection has repeatedly shown to be detrimental for high quality or fast detection of inputs appears far more critical. As has been pointed out repeatedly by Olshausen and Field (1996b) and others, the brain seems to use the idea of sparseness ubiquitously, and many new technical advancements have only been possible by copying the brain's ascribed search for sparseness. When synaptic weights are never allowed to become (close to) zero, a large number of random inputs may still evoke random postsynaptic responses while the neuron will never be able to learn to ignore unrelated noise among its inputs. It will therefore never be able to learn to reject false positives while learning to detect a specific set of inputs. A neuron without the ability to form sparse connections will never be able to achieve long pattern retention times as it keeps responding more randomly than necessary. A neuron with a sparse distribution of incoming weights will instead be able to remain quiet throughout most of its (post-training) lifetime and only fire a spike when the inputs it receives are sufficiently similar to those it was trained on. Sparseness in a neuron's weight distribution therefore also allows sparseness in its responses. The bad pattern retention times of multiplicative STDP was also noticed in (Billings and van Rossum, 2009), and the benefits of sparse weight distributions will be further explored in chapters 6 and 7.

This drawback in unsupervised learning performance of multiplicative STDP is likely the reason for why Gilson et al. (2010) only chose to use the multiplicative STDP rule of Gütig et al. (2003) with a very small exponent $\mu = 0.03$. The interesting behaviour of automatically keeping synaptic weights within a predefined range while allowing good detection performance through bimodal weight distributions comes into effect only for very small values of μ, where the rule becomes very close, but not completely equal to, additive STDP rules.

As multiplicative STDP rules have a built-in synaptic drift towards the stable attractor, the connections of purely randomly spiking input units (noisy inputs) gather around this attractor while reliable input units that always fire shortly before a postsynaptic spike (e.g. the proximal units of the *before* group from above) form connections that are overall stronger. While I explained above why an ever-increasing drift may be counterproductive when trying to form a reliable memory and pattern detector, some level of (constant) depressing synaptic drift may indeed be very helpful for separating repetitive spatiotemporal (or *polychronous*[2]) incoming spike patterns from random noise. However, this negative drift need not be implemented by weight-dependent scaling of synaptic updates as in multiplicative STDP, but can simply be made an inherent feature of any STDP rule by using biased STDP.

Comparison of scaling rules for biased STDP

Figure 5.7 shows the same weight bounding rules as just discussed, but with a biased STDP rule that was successfully used to learn temporally structured spike patterns among noisy data. Biased STDP rules have a constant (typically negative) synaptic drift as shown in Figure 5.1 (right) and are more similar to what has been observed experimentally (Figure

[2]as coined by Izhikevich (2006); Izhikevich et al. (2004), see also Chapter 6.

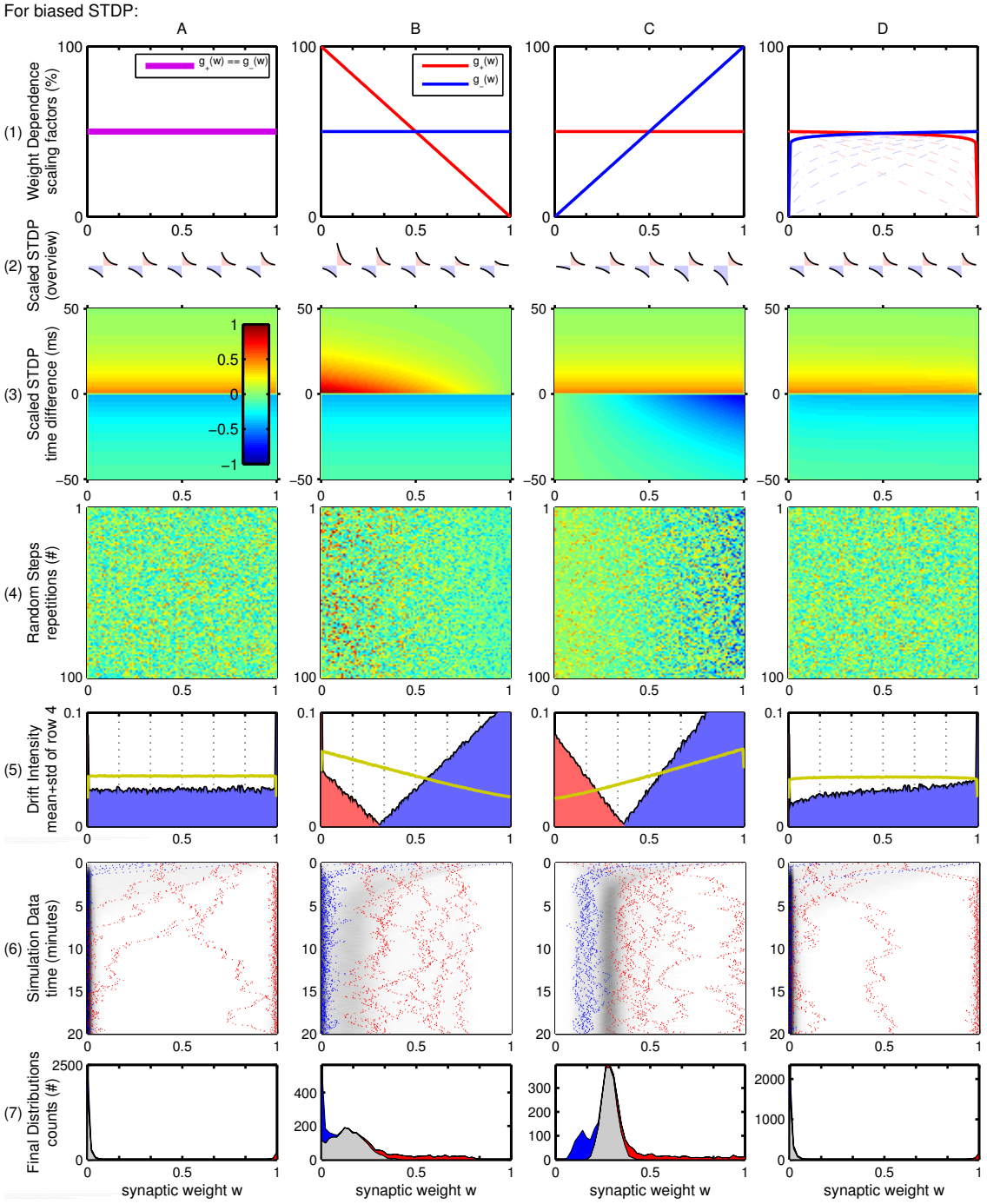


Figure 5.7: Comparison of effects of weight-dependent scaling rules on biased STDP. *Column A*: Additive STDP (Eq. 5.3). *Column B*: Weight-Dependent LTP (Eq. 5.4). *Column C*: Weight-Dependent LTD (Eq. 5.5). *Column D*: Interpolated power law (Eq. 5.6). – *see Figure 5.6 for description of rows. . .*

2.8, p. 18). While row 1 (columns A-D) is identical to Figure 5.6, rows 2 and 3 already show differences of the biased STDP rule in weight-dependent step sizes. The negative drift of biased STDP is fully visible in row 5, where both the additive (column A) and additive-like (column D) rules now have a constant negative drift added for all weights, causing depression for uncorrelated inputs. The built-in negative drift of the biased STDP rule I use here also causes the stable fixed point attractor of multiplicative rules to shift a little further into depression, although some potentiating drift is preserved for weak synapses (columns B and C, row 5). While the depressing drift for additive-like STDP (column D) is a bit less for weak weights, the drift remains overall negative throughout nearly the full weight range here.

The figure becomes even more interesting when observing the simulation results of rows 6 and 7. For both additive and additive-like STDP bounding (columns A and D), the previously broadly distributed weights of uncorrelated inputs drift towards zero connection strength, while only some connections to the *before* group (shown in red) remain strong. Not all inputs within this group maintain strong inputs, though, as the inputs that tend to fire very early before postsynaptic activity are lost to the negative drift. Only inputs that repeatedly fire just before a postsynaptic spike have the ability to remain strong, because their potentiating steps are large enough to successfully compete with the many random steps with negative drift that happen in between two potentiating pairings. The size of the negative drift bias can therefore also be seen as a cut-off value below which all inputs to a postsynaptic neuron are interpreted as noise. Closely examining this bias in real neurons might therefore give a hint on the levels of noise that a measured neuron expects.

For multiplicative bounding rules (columns B,C and rows 6,7) the weights of input units within the *before* group (red) are spread out over a larger range. It is also even less common for units of this group to form very strong connections near the maximum weight (column C). Indeed, for these settings, increased negative drift has caused the red peak at the maximum weight that was observed in the previous figure (row 7, column C of Figure 5.6) to disappear here. As all units in the *before* group have by definition the same correlation with the postsynaptic cell, albeit with different amount of shift in time, it becomes clear that the claimed mapping of correlation strength to resulting connection strength (Rubin et al., 2001; van Rossum et al., 2000) is not fully correct. The mean time between presynaptic firing and postsynaptic response also has a very strong effect on the resulting weight distribution (compare Sketch 5.2, p. 41).

The *after* group shows even more interesting behaviour. While the distribution of synaptic weights of this group (blue area) looked simply like an extension of the unimodal distribution of uncorrelated weights for unbiased STDP (rows 6 and 7, column C in Figure 5.6), this group of input weights now forms an own peak. And still, the van Rossum et al. (2000) rule (column C, row 7) does not allow any weights to become zero, with all the implications on learning and sparseness as described above. Chapter 7 will describe learning performance as a measure of hits and misses, as well as false alarms and correct rejections of a set of training patterns.

But what about the seemingly superior biological validity of multiplicative rules? The neat idea of keeping weights within a given range without the need for a hard boundary, just by using a central attractor for synaptic connection weights? We will now see that multiplicative rules with inverse proportion of potentiation step size to synaptic weight are in fact a very bad fit to experimental data.

5.2.4 Revisiting existing data

Morrison et al. (2007) re-examined the biological data gathered by Bi and Poo (1998) and re-fitted an exponential weight-dependent update rule to this existing data. While the general idea of making the weight dependence non-linear by use of a real-valued exponent is similar to (and inspired by) Gütig et al. (2003), Morrison et al. (2007) noticed that the update step size of potentiation in the data actually increased with increasing weight as it does for depression. The flipped sign of weight dependence for potentiation can not be described within the rule of Gütig et al. (2003). Also, to fit an exponential each to potentiating and to depressing steps, different exponents had to be used for potentiation and depression.

The data in Figure 5.8 was extracted from Figure 1A in Morrison et al. (2007) and originally recorded by Bi and Poo (1998). It shows the change in postsynaptic current (PSC) over the initial PSC measured before commencement of an STDP pairing protocol. As in Morrison et al. (2007), the figure here shows absolute changes in pA, in contrast to the relative percentage changes displayed in Figure 5.4 (p. 44) (from Bi and Poo, 1998, Fig. 5). In addition to the logarithmic plot in Morrison et al. (2007), Figure 5.8 here also shows the same data on linear axes on the right. This helps to better visualise a bounding rule's match with strong initial weights and/or large absolute step sizes and allows an easier visual comparison of all figures in this chapter.

The distinct-exponential fit by Morrison et al. (2007) resulted in an exponent $\mu_1 = 0.4$ on the synaptic weight for potentiation and an exponent $\mu_2 = 1$ for depression (Equation 5.7):

$$\begin{aligned} g_+(w) &\propto (c_1 w)^{\mu_1}, \\ g_-(w) &\propto (c_2 w)^{\mu_2} \end{aligned} \tag{5.7}$$

where c_1 and c_2 are some constants.

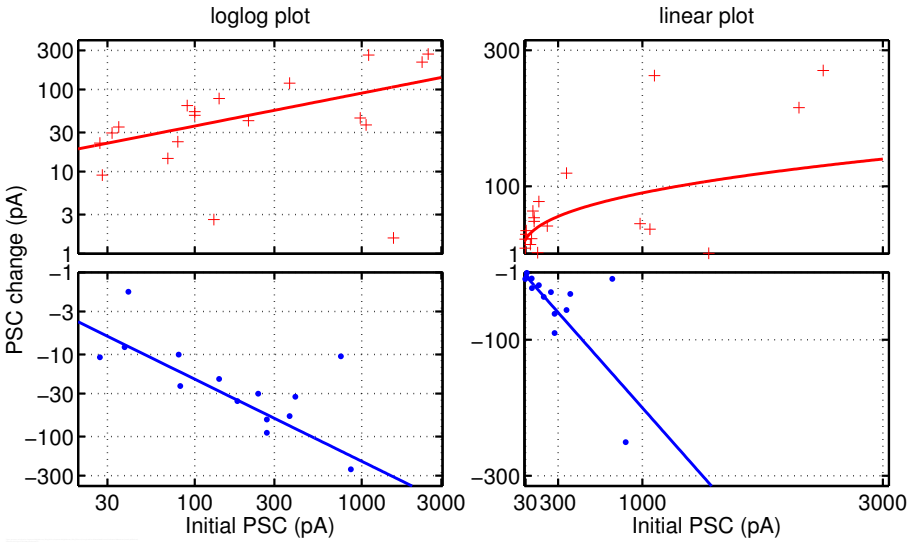


Figure 5.8: Reproduced from Morrison et al. (2007). An exponential fit of potentiating updates to a continuously increasing function $g_+(w)$ (Eq. 5.7) represents experimentally measured data far better than constant (Eq. 5.2) or continuously decreasing functions (Eq. 5.4). Red plus sign markers: Potentiating update steps. Blue filled circle markers: Depressing update steps. Left: Logarithmic plot as in Morrison et al. (2007). Right: Linear plot for easier visual comparison to other figures. Data from Bi and Poo (1998), extracted from Morrison et al. (2007).

The shape of $g_+(w)$ makes the scale of potentiating weight updates increase steeper than $g_-(w)$ at first (for weak weights), but is soon overtaken by $g_-(w)$ (scale of depressing weight updates) for higher initial weights because $\mu_1 < \mu_2$. This leads to a slight dominance of LTP for weak weights and a dominance of LTD for stronger weights (see also Sketch 5.9).

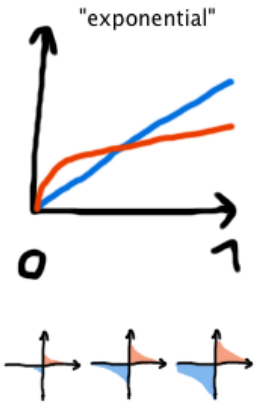
Although both potentiating and depressing weight updates increase with initial synaptic weight, this rule still forms a stable fixed point that attracts the weights at the intersection between $g_+(w)$ and $g_-(w)$. It therefore behaves similar to the rule used by van Rossum et al. (2000), as can be seen in Figures 5.6 and 5.14 (columns C and E) as well as Figures 5.7 and 5.15 (columns C and E).

Morrison et al. (2007) also exemplarily fit weight-dependent update scaling rules with additive ($g_+(w) \propto 1$) and multiplicative ($g_+(w) \propto 1 - w$) potentiation to the same data. Figure 5.10 shows that an additive fit for potentiation as in van Rossum et al. (2000) (Equation 5.5) does not describe the data well. In extension, purely additive weight dependence as in Song et al. (2000) (Equation 5.2) also describes the data for depressing update steps badly (no figure shown, but imagine Figure 5.10 with a horizontal blue line instead of diagonal).

Multiplicative rules with inverse dependence of potentiation on weight $g_+(w)$, as used by Kistler and van Hemmen (2000) and Rubin et al. (2001) and by extension also Gütig et al. (2003) (Equations 5.4 and 5.6) can now be seen to fit the existing potentiation data even worse. Figure 5.11 shows an attempted fit (based on Morrison et al., 2007), which cannot describe the potentiation data points satisfactorily.

Among the previously described weight-dependent update scaling rules, the rule by Morrison et al. (2007) shows the best fit to existing data, while still exhibiting a stable fixed point as previous multiplicative rules do. Figures 5.6 and 5.15 (columns C vs. E) show that its behaviour is closely related to that of van Rossum et al. (2000) and very different from additive rules (column A). Sections 5.2.5 and 5.2.7 and chapters 6 and 7 will discuss benefits and shortcomings of this rule for efficient pattern learning.

There remains, however, the small question of how the weights would act in the absence of noise (also noted by Morrison et al., 2007). If all synaptic updates were purely potentiating, the connection weights would theoretically grow to infinity. I am currently aware of no data which settles this, and so will discuss a simple solution in the next section.



Sketch 5.9: Exponential absolute weight dependence of synaptic updates as in Morrison et al. (2007) (Eq. 5.7) for $g_+(w) \propto w^{0.4}$ (red) and $g_-(w) \propto w$ (blue). The second row shows sketches of how the effective STDP rule changes due to weight-dependent scaling of its timing-dependent update steps. More information in Figures 5.14 and 5.15

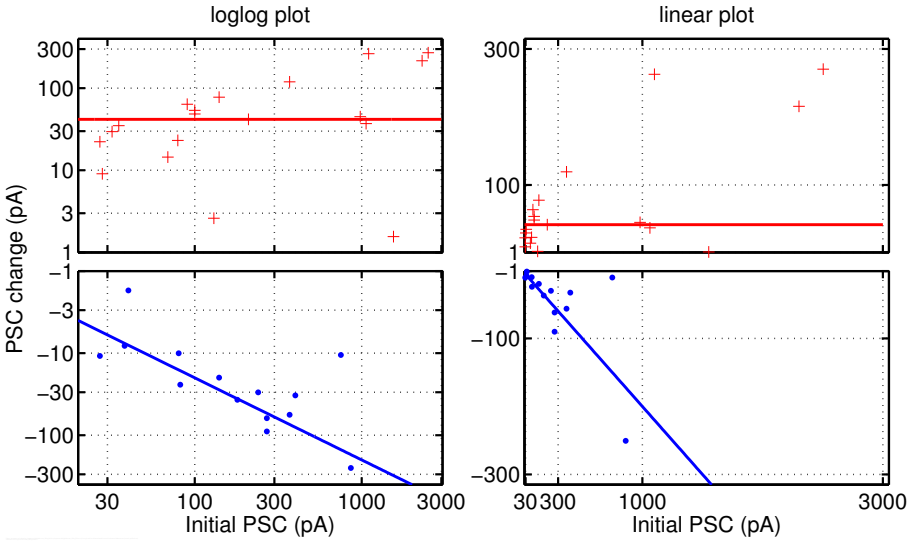


Figure 5.10: A multiplicative rule as in van Rossum et al. (2000) (Eq. 5.5) does not fit the experimental data as well as Eq. 5.7. Red plus sign markers: Potentiating update steps. Blue filled circle markers: Depressing update steps. Left: Logarithmic plot as in Morrison et al. (2007). Right: Linear plot for easier visual comparison to other figures. Data from Bi and Poo (1998), extracted from Morrison et al. (2007).

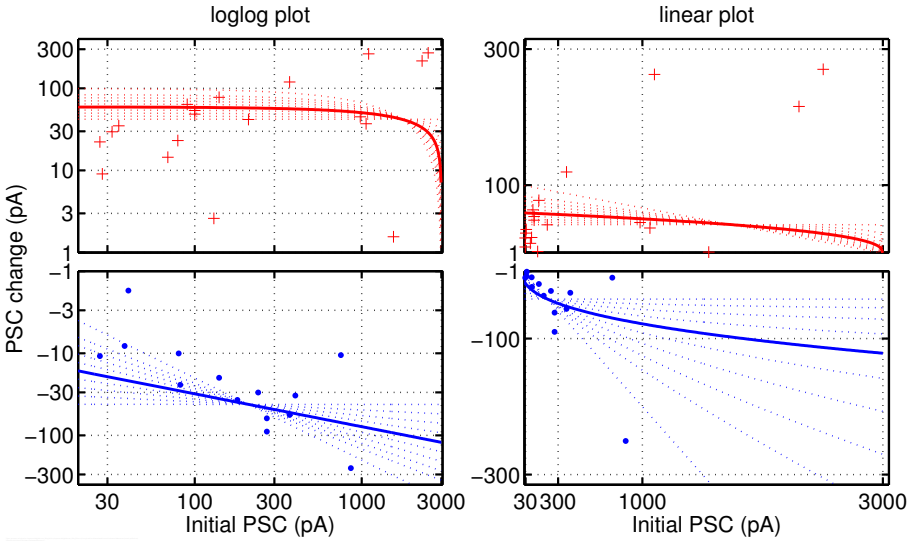


Figure 5.11: A two-sided multiplicative interpolated rule as in Gütig et al. (2003) (Eq. 5.6) fits the experimental potentiation data (in *red*) much worse than even in Figure 5.10. Red plus sign markers: Potentiating update steps. Blue filled circle markers: Depressing update steps. Left: Logarithmic plot as in Morrison et al. (2007). Right: Linear plot for easier visual comparison to other figures. Data from Bi and Poo (1998), extracted from Morrison et al. (2007).

5.2.5 A new approach to balanced, weight-dependent scaling of STDP

While fitting their weight-dependent STDP scaling rule to biological data, Morrison et al. (2007) found that the data does not fit previous conceptions of how potentiating weight update steps should depend on initial synaptic strength. Instead, they found that *both* potentiating and depressing steps increase in amplitude with initial PSC. Although the exponents μ_1,μ_2 used in Equation 5.7 are not equal, the weight-dependent scaling of synaptic update steps is a lot more similar than previously assumed.

As discussed in the previous section (and p. 48), a stable fixed point attractor for the (one-dimensional) synaptic drift only exists inside the defined weight range for $\mu_1 < \mu_2$. If this relationship were inverted ($\mu_1 > \mu_2$), the fixed point within the weight drift dynamics would become unstable, and repel any weight updates towards the extremes of the defined weight range. Such an artificially forced creation of a bipolar weight distribution is not desirable because it is not based on structured information in the inputs (see Chapter 6), but solely on the initial strength of a synapse when it originally became plastic. Structured inputs could only make their synaptic weight cross the repelling fixed point if the fixed point was very weak, that is, if the rule were nearly additive.

But also for a stable fixed point that strongly attracts synaptic drift ($\mu_1 < \mu_2$), the representation of structured information within the distribution of synaptic weights becomes less informative, as all synaptic strengths cluster around this fixed point. So-called multiplicative rules that expose a strong fixed point attractor within the weight range have within the last 15 years not satisfyingly managed to show sufficient performance in learning structured data (Billings and van Rossum, 2009; Gilson et al., 2010). The main repeated argument that is given in favour of multiplicative rules with strong fixed-point attraction has been that their weight distribution looks unimodal (Gilson et al., 2011; Morrison et al., 2007; Rubin et al., 2001; van Rossum et al., 2000), but this can also be achieved by other means (see Section 5.3).

I therefore pose the question whether the stable fixed point inside the borders of a synaptic weight range as caused by multiplicative STDP should indeed be given the

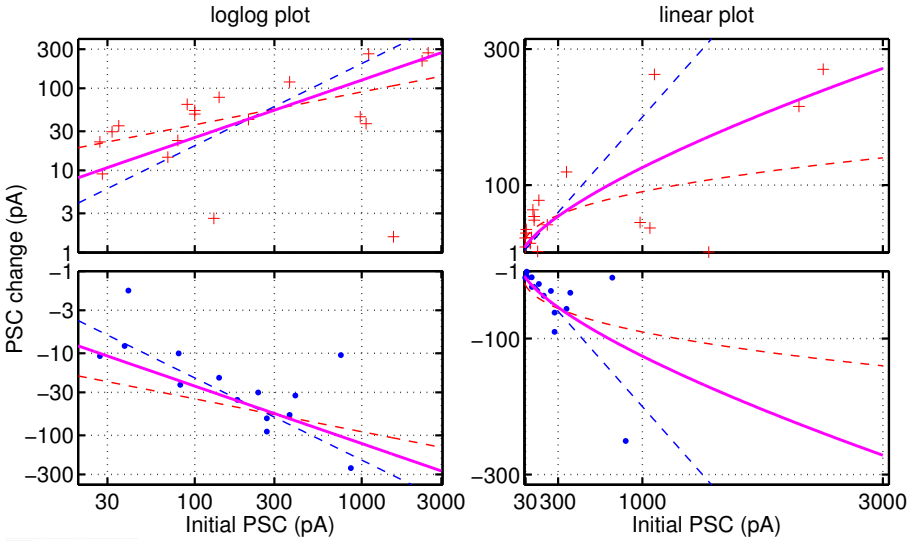


Figure 5.12: The weight-dependent scaling rule by Morrison et al. (2007) without a stable fixed point for $\mu_1 = \mu_2 = 0.7$ (magenta curve). Dashed curves (red and blue) show original fit as in Figure 5.8. All markers and scalings as in Figures 5.8, 5.10, 5.11. Data extracted from Morrison et al. (2007), originally from Bi and Poo (1998).

amount of attention it has seen in the past, or if it should rather be noted as just a marginal influence on synaptic drift that is *not* a main enabling factor for stable synaptic plasticity. While its existence can not be questioned from the data at this point, its origin in divergent weight-dependent scaling of LTP and LTD can. Its effect on robust learning of structured information is more detrimental than helpful when modelled as an interaction between two different weight-dependent scaling factors g_+ and g_-, and needs to be overcome by any successful STDP learning rule.

If we set $\mu_1 == \mu_2$ in Equation 5.7, the fixed point disappears while the size of both LTP and LTD remains dependent on synaptic weight. Figure 5.12 shows that while such a weight-dependent STDP scaling rule may not fit the data as well as using $\mu_1 = 0.4$ and $\mu_2 = 1$ as in Morrison et al. (2007), it would still be a much better fit than either of Equations 5.2 through 5.6.

In fact, when viewing the dashed red and blue curves in the linear plot (right side) of Figure 5.12, one can see that the crossover point where potentiating drift turns into depressing drift is fairly close to the lower border of the defined range of synaptic weights (around initial PSC $\approx$ 200 pA in Morrison et al., 2007). A positive synaptic drift (where potentiating steps tend to be larger than depressing steps) only exists for very small synaptic connection strengths, and most of the synaptic weight range encountered in experiment has a strong dominance of depression, causing a negative drift there. The weight-dependent update scaling rule found by Morrison et al. (2007) therefore displays a strong negative drift over most of its range, while increasing update step sizes indefinitely (compare Figure 5.12 with Figures 5.14 and 5.15 column E). The fit to data by Morrison et al. (2007) puts the stable fixed point attractor very close to zero, leading to negative synaptic drift throughout most of the defined range of synaptic weights, and a positive drift only for the weakest weights. The existence of a stable fixed point when fitting the recorded data to different weight-dependent scaling functions for LTP and LTD need not be the only conclusion to draw from the data. Instead, the resulting fixed point may also be created by a number of other biological factors, including possibly a combination of (homeostatic) synaptic growth in combination with a biased STDP rule that has constant negative drift.

As the data leaves open how a maximum synaptic weight is approached, I assume that at some point within the weight range, the step size of weight updates begins to gradually decrease again before reaching an upper bound of weights. Equation 5.8 shows a weight-dependent update scaling rule where $g_+(w) = g_-(w)$, that also includes a mirrored progression to that seen in Figure 5.12 as the synaptic weight approaches a yet unknown upper border (see also Figure 5.13 left):

$$g_\pm(w) \propto \begin{cases} w^\mu & \text{for } \; w < 0.5 \\ (1-w)^\mu & \text{for } \; w \geq 0.5 \end{cases} \tag{5.8}$$

This weight bounding rule succeeds in keeping synaptic weights within a defined range $w \in [0, 1]$, without the need for hard clipping at the borders of that range. Although spike-timing dependent synaptic updates are multiplied with a weight-dependent function $g_\pm(w)$, the rule in Equation 5.8 does not cause a drift in either direction because $g_+(w)$ and $g_-(w)$ do not intersect but are instead identical. It introduces no additional synaptic drift that changes with synaptic strength. While it is different from additive rules in its dependence on synaptic weight, it does not have any stable fixed point attractors inside the defined range of weights as the previously described multiplicative rules do.

This shows that the previous classification of weight-dependent update scaling rules

into "additive" rules and "multiplicative" rules is unfortunately named, because it is not the multiplication of weight updates with some weight-dependent function $g_{\pm}(w)$ in Equations 5.4, 5.5, 5.6 and 5.7 that causes their resulting weight distributions to become unimodal, but the existence of a stable fixed point attractor in their synaptic drift dynamics. A better term for identifying classes of STDP rules may therefore be found by identifying the stable attractor as the distinguishing feature. From now on I will therefore use the term *attractor-based STDP* to describe STDP rules that form a stable attractor somewhere within the defined range of possible weights, and *attractor-less STDP* to describe rules where the direction of an STDP rule's synaptic drift does *not* depend on synaptic weight, even though the absolute step size for both LTP and LTD may.

After introducing two more attractor-less STDP rules in the following section, I will compare these to all previously described STDP scaling rules in Section 5.2.7.

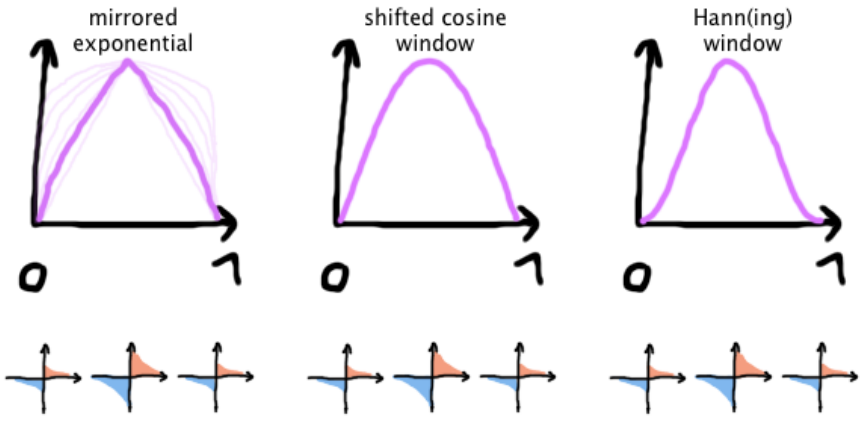
5.2.6 Practical weight-dependent, attractor-less weight bounding rules

The weight-dependent update scaling rule that was introduced in the last section is just one of many possible attractor-less bounding rules imaginable. I will now show two more instances of weight-dependent bounding rules that are continuous while keeping synaptic weights within a desired range.

A first possibility could be to use a simple shifted cosine window function as the weight bound on a normalised range ($w \in [0, 1]$), as is used in many standard signal processing algorithms today. The benefit is that this continuous, parabola-like function also approaches the edges of the defined weight range with decreasing steps while allowing a broader area in mid-range where step sizes remain large. Equation 5.9 describes this window function, which is also shown in Sketch 5.13 (center):

$$\begin{aligned} g_+(w) &\propto cos(\pi\, w - \frac{\pi}{2}) = sin(\pi\, w), \\ g_-(w) &\propto cos(\pi\, w - \frac{\pi}{2}) = sin(\pi\, w) \end{aligned} \tag{5.9}$$

The use of a shifted cosine window for weight bounding is partly inspired by Guyonneau et al. (2005), where an unclipped additive rule was projected to a sinusoidal range before



Sketch 5.13: Attractor-less weight-dependent update scaling rules (Eqs. 5.8, 5.9, 5.10). The step size of STDP updates varies equally for LTP and LTD. *Purple lines:* $g_+(w) = g_-(w)$. Step size is largest for central weights, while diminishing as weights become closer to the extremes. *Left:* The exact shape of the weight-dependent scaling rule of Eq. 5.8 depends on the chosen setting for the exponential μ. Other settings for $\mu < 1$ are shown in thin lines in the background.

computation of each synapse's effect on the postsynaptic membrane. However, the weight-dependent scaling rule presented here in Equation 5.9 is different in that weight updates are directly applied to the actual synaptic weight, and no intermediate mathematical transformation step is needed before synaptic transmission can take place.

While this function scales weight update steps by the present synaptic weight of a synapse and is continuous, its curve is still similar to that of Equation 5.8. It will therefore be mainly used in the subsequent chapters. A second possible soft weight bounding rule is to use a Hann(ing) window function for weight-dependent update scaling, as in Equation 5.10 (see Figure 5.13 right):

$$\begin{aligned} g_+(w) &\propto Hann(w), \\ g_-(w) &\propto Hann(w) \end{aligned} \tag{5.10}$$

$$Hann(w) = \frac{1 - cos(2\pi\, w)}{2} \tag{5.11}$$

This window function decreases the step size of synaptic updates long before they reach the edge of the defined range, so that uncorrelated spike pairings cause synaptic weights to gather further away from the boundaries of the defined weight range. This will be further explored in the following section.

5.2.7 Performance and Features (Set 2)

I now compare all previous STDP scaling rules (Sections 5.2.1-5.2.3) with the attractor-based power-law rule with equal sign of slopes of g_+ and g_- as reviewed in Section 5.2.4, and with the three new attractor-less weight-dependent STDP scaling rules I proposed in Sections 5.2.5 and 5.2.6. The previous rules were evaluated in columns A-D of Figures 5.6 (p. 49) and 5.7 (p. 54), while the four additional rules are described in columns E-H of Figures 5.14 (p. 63) and 5.15 (p. 66).

Comparison of scaling rules for unbiased STDP

As noted in *Visualisation of Effects* (p. 47), the first row of Figure 5.14 plots only the weight-dependent scaling factors $g_+(w)$ and $g_-(w)$. Column E here shows Equation 5.7 for exponents $\mu_1 = 0.4$ and $\mu_2 = 1$ as in Morrison et al. (2007). Constants c_1 and c_2 were chosen to be $c_1 = c_2 = 0.5$ for improved visual clarity, as the original constants of Morrison et al. (2007) move the crossover point very close to $w = 0$, making it hard to see the effects of the stable fixed point attractor in this comparison. While at first sight the Morrison et al. (2007) rule of column E looks quite different than the multiplicative rule of van Rossum et al. (2000) in column C (Figure 5.6), a closer inspection reveals that, just as in multiplicative STDP, the scale of potentiating updates (LTP) dominates for weak weights while the scale of depressing updates (LTD) dominates for strong weights. The crossover point again indicates the existence of a stable fixed point attractor at this location. As noted before, if μ_1 were larger than μ_2, this fixed point would stop being an attractor and instead repel any synaptic drift.

Column F (row 1) shows the power law rule with $\mu_1 = \mu_2$ for $w \in [0, 0.5)$, and a mirrored progression for $w \in [0.5, 1]$ (Equation 5.8). Different possible values for μ_1, μ_2 are visualised by the dashed lines, but we choose $\mu_1 = \mu_2 = 0.7$ for this comparison (continuous line) as it is the mean of 0.4 and 1 (compare Figure 5.12). Columns G and H

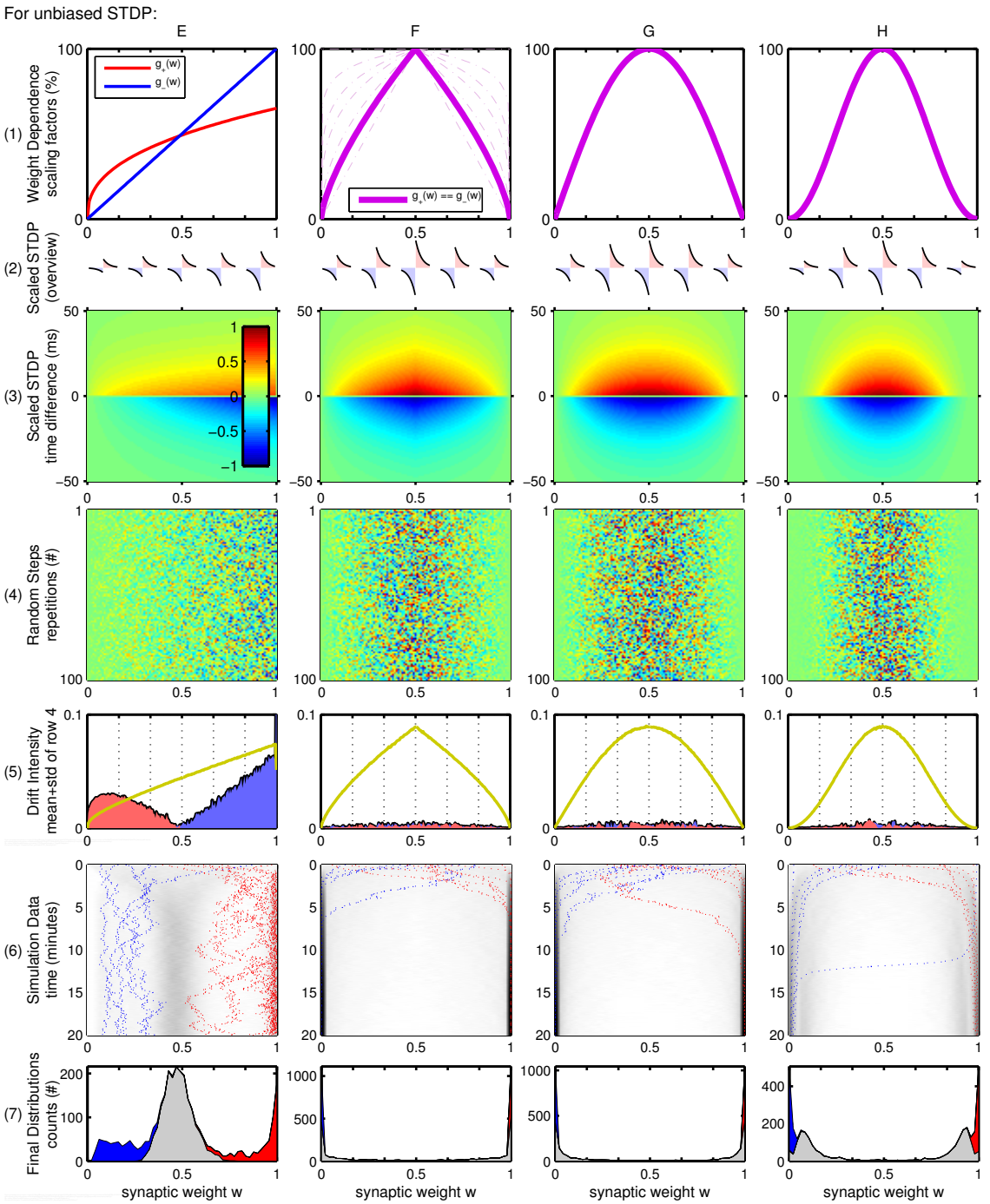


Figure 5.14: Comparison of effects of weight-dependent scaling rules on unbiased STDP. *Column E*: Power law scaling of STDP with weight dependent increase of LTD *and* LTP scale (Eq. 5.7). *Columns F-G*: Balanced, weight-dependent soft bound. *Column F*: Power law scaling with equal exponents and mirrored right half (Eq. 5.8). *Column G*: Shifted cosine window (Eq. 5.9). *Column H*: Hann(ing) window (Eq. 5.10). – *see Figure 5.6 for description of rows...*

show the (shifted) cosine window of Equation 5.9 and the Hann(ing) window of Equation 5.10.

The second and third row again visualise the effect of weight-dependent scaling on the full STDP rule as in Figure 5.6. In column E the total amount of potentiation outweighs the amount of depression for weak weights, but is overtaken by increased depression for stronger weights as expected for multiplicative rules. In columns F-H, however, the ratio between potentiation and depression remains constant throughout the full range of possible weights. Due to the weight dependent scaling, the step size of updates also varies here, but equally for both LTP and LTD. The unbiased STDP rules in rows two and three retain their symmetry.

This directly affects rows four and five, where columns F-H show no bias in synaptic drift. Synaptic weights truly perform a random walk. This is in spite of the fact that update step sizes are weight dependent, as shown by the green areas on either side of the weight range in row four, and the course of standard deviation in update step size across the weight range in row five (yellow curve, scaled by 0.2 for display again).

Rows six and seven of column E show the typical development of synaptic weights in a computer simulation (see Appendix A.3.3) as would be expected for multiplicative STDP. The synaptic weights of uncorrelated inputs (grey area) gather around the stable fixed point attractor, as this type of weight-dependent scaling keeps random weight updates near the crossover point. As in column C (Figure 5.6), synaptic connections from input units in the *after* group (blue dotted lines and blue area) can never reach $w = 0$ when the presynaptic input units are also noisy. Synaptic connections to units in the *before* group (red dotted lines and red area) form a stronger peak near $w = 1$ than in column C, as the scale of potentiating updates kept increasing and is higher here for $w \approx 1$ than in column C.

Columns F and G of rows 6 and 7 show a very similar behaviour in that all synaptic weights gather near the edges of the defined range. Not just the strongly correlated inputs of the *before* and *after* groups do this (as in additive STDP, column A, Figure 5.6), but also the connection weights of uncorrelated inputs. However, one must *not* conclude from this that there must be an unstable fixed point somewhere in the weight range. As there is no crossover between $g_+(w)$ and $g_-(w)$, no fixed point exists. Instead, what we see is simply an effect of the larger (symmetric) step size for central weight values compared to the step size of weight updates around the edges of the weight range. For a more intuitive understanding, the reader may compare this to viewing a crowd through a fish-eye lens or a convex mirror: While single movements are truly random, the step size appears faster near the center of the lens/mirror. Uncorrelated inputs still cause their connection weights to perform a true random walk, except that the time they spend at intermediate values ($w \approx 0.5$) is short.

In column H (row 7), the strength of synaptic connections to randomly firing input units (grey area) forms clusters near the edges of the defined weight range without seemingly reaching these edges. This is simply a transient effect that will result in a similar distribution as in columns F and G (row 7). But as the approach towards the extreme edges of the range is so much slower for uncorrelated inputs than when structured inputs (red and blue areas) are used, future learning tasks may still benefit from using an STDP scaling rule as in this column (column H, row 1). Again, the division of the weight distribution for uncorrelated inputs is not due to any (unstable) fixed point. Instead, each synapse of randomly spiking inputs here may at any point spontaneously switch sides as per the random walk described above.

Comparison of scaling rules for biased STDP

The biased nature of the STDP rule used in Figure 5.15 can be seen in row three, where the amount of LTP and LTD is no more symmetric in columns F-H. As the discrepancy between potentiation and depression is not caused by the weight-dependent scaling factors g_+ and g_- but by the inherent differences in the main STDP rule ($A_+ \neq A_-$ and $\tau_+ \neq \tau_-$), the graphs in the first row show no differences in weight-dependent scaling here (purple curves, columns F-H, row 1).

The constant negative synaptic drift of the biased STDP rule also strongly affects the average drift intensity in row five, where the red area (=potentiating drift) in column E (row 5) is decreased while the blue area (=depressing drift) is increased. This change also indicates a slight shift of the stable fixed point towards weaker weights in comparison to Figure 5.14. Columns F-H (row 5) now show a depressing drift throughout the complete weight range, which is strongest around central weights and becomes weaker near the edges of the defined weight range. However, in opposition to column E, the sign of synaptic drift still does not change as there is still no crossover between g_+ and g_-.

The simulation data (rows 6 and 7) of Figure 5.15 shows the influence of negative synaptic drift in extension of Figure 5.7 (p. 54). Attractor-based STDP scaling rules (columns B-E) see the peak of weights for uncorrelated inputs (grey areas) shift further into depression. This indicates that the stable fixed point attractor has also moved in this direction. However, synaptic connection weights of the *after* group (blue dotted curves and blue area, rows 6 and 7) again never reach zero when using the Morrison et al. (2007) STDP scaling rule (column E) as in the van Rossum et al. (2000) rule (column C, Figure 5.7), because the scaling function of LTD (g_-) becomes zero as the weight moves towards zero. If it didn't, an attractor-based STDP scaling rule could theoretically reach zero synaptic strength (see column B), but any successful synapse would still need to overcome the competing drift of the stable fixed point attractor.

For attractor-less STDP scaling rules (columns A,F-H), a negatively biased STDP rule lets all weights of uncorrelated inputs move towards the lower bound ($w \to 0$) instead of evenly distributing those weights as under unbiased STDP (Figures 5.6 and 5.14). As in Figure 5.7, this reduction of synaptic weights to randomly firing inputs lets only some connections of the *before* group (red dotted curves and red areas, row 6 and 7) remain strong. Again, not all synapses in this group manage to escape the negative synaptic drift, and those input units in the *before* group that fire a long time before each postsynaptic spike loose this competition and are also depressed. This can be seen as some red dotted curves in row 6 also move towards zero, together with all blue dotted curves and the synapses of uncorrelated background inputs (grey area). See also Figure A.7 (p. 181) for an overview which input units fired at which timing differences from each postsynaptic spike.

5.3 Synaptic weight distributions & Homeostasis

The reader should now be convinced that the classification of STDP rules into *multiplicative* ("weight-dependent") and *additive* ("not weight-dependent") STDP is badly chosen, and that the terms *attractor-based* and *attractor-less* may rather be used. This is motivated by the construction of a weight-dependent, attractor-less STDP rule in the previous section. Attractor-less STDP seems a good choice for maximising representational difference between causally correlated (often *before*) input units and those that are either anti-causally correlated with (often *after*) postsynaptic spikes or which are not correlated.

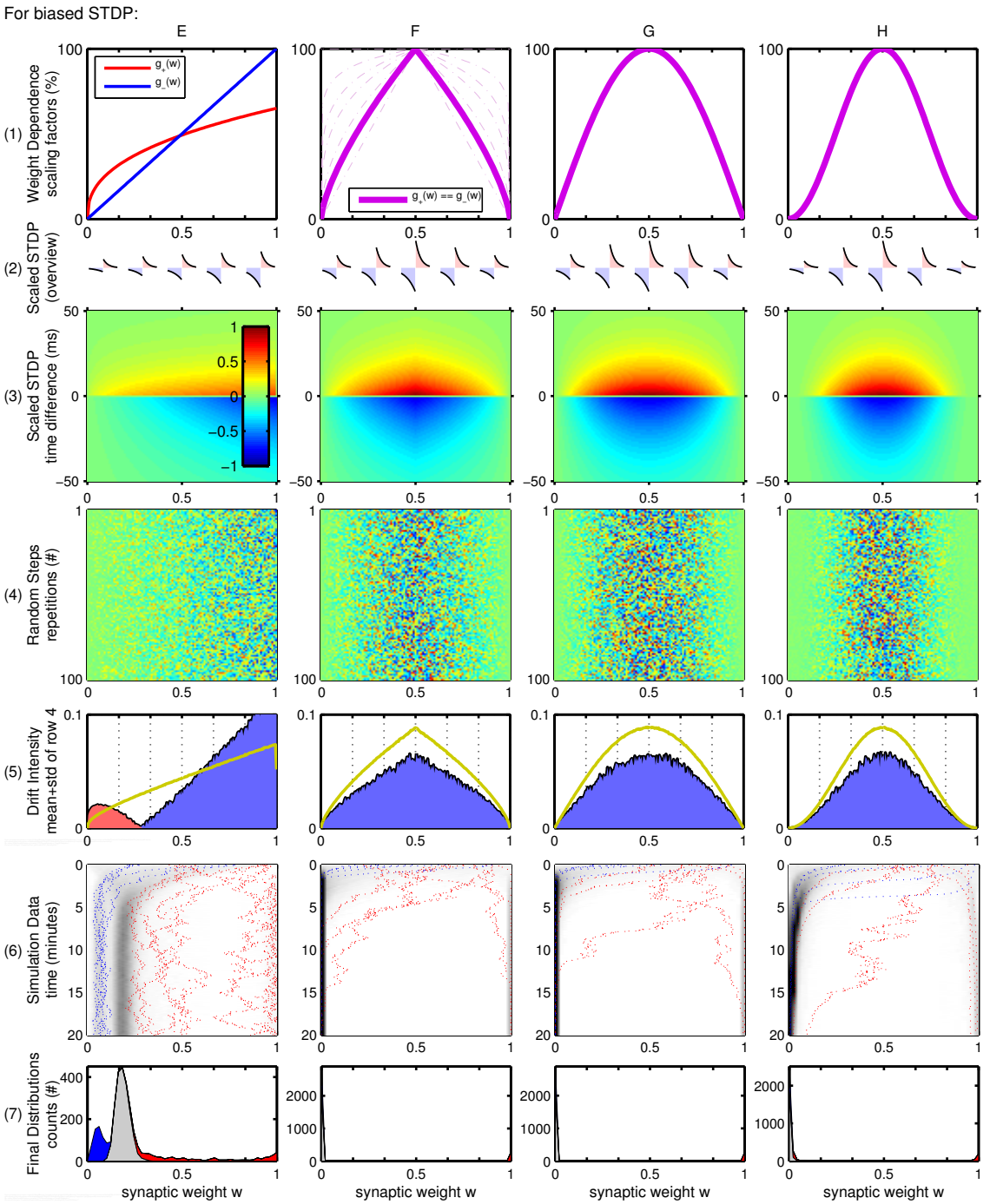


Figure 5.15: Comparison of effects of weight-dependent scaling rules on biased STDP. *Column E*: Power law scaling of STDP with weight dependent increase of LTD *and* LTP scale (Eq. 5.7). *Columns F-G*: Balanced, weight-dependent soft bound. *Column F*: Power law scaling with equal exponents and mirrored right half (Eq. 5.8). *Column G*: Shifted cosine window (Eq. 5.9). *Column H*: Hann(ing) window (Eq. 5.10). – *see Figure 5.6 for description of rows...*

I have also discussed how attractor-less STDP can lead to sparsification of synaptic weight distributions, while attractor-based STDP typically keeps most synaptic connections non-zero.

In the simulations shown in Figures 5.6, 5.7, 5.14, and 5.15 (pp. 49, 54, 63, and 66), postsynaptic spike timing was manually fixed to exactly 1 Hz regular firing (see Appendix A.3.3) to visualise STDP effects in a clear manner. However, in practice, the timing of postsynaptic spikes is thought to instead depend on some weighted combination of its inputs (see also Chapter 6). When the spike timing of a postsynaptic neuron is allowed to depend solely on its presynaptic inputs, long periods of uncorrelated input spikes lead either to a randomly selected small subgroup of inputs becoming strong while all other weights move close to zero (attractor-less unbiased STDP), or to a common decrease of all synaptic weights until the postsynaptic neuron becomes quiet forever before it can form a precise representation of any group of inputs (attractor-less biased STDP, and attractor-based STDP where the attractor is very close to zero), assuming sufficiently large network size (see also Gütig et al., 2003; Rubin et al., 2001).

Proponents of attractor-based ("multiplicative") STDP argue that keeping most synaptic connections far away from zero helps in keeping the postsynaptic neuron active and thereby plastic. This is indeed important during the initial training phase of a (new) neuron, when it has not yet formed a sufficient number of strong connections to allow it to respond with high selectivity to a small group of concurrently active inputs. If a neuron that has not yet formed a reliable receptive field is allowed to become quiet too soon, it may never properly form a selective receptive field. This is because, by definition, STDP needs postsynaptic spikes for synaptic changes to occur, just as traditional Hebbian plasticity needs a non-zero postsynaptic activation term (Equation 2.10, p. 16).

It is, though, highly doubtful that most adult neurons recordable in *in vivo* electrophysiological recordings still show high spiking activity throughout their lifetime. Histological cell counting methods have revealed a far higher number of neurons in recording range around implanted electrodes than are usually recorded (Kerr et al., 2005; Shoham et al., 2006; Vann et al., 2000; Wan et al., 2001). It may be more likely that well-trained neurons remain quiet for most inputs and only become active if they detect a trained pattern that fits their receptive field. As a neuron becomes more selective to a distinct group of inputs (sparse weight distribution), its responses to incoming (random) activity also become more sparse (sparse spike timing). The period in which a neuron responds to all inputs in general without being specifically tuned to a subset of inputs and without being quiet has been called the *critical period* (Crair and Malenka, 1995; Hensch, 2005).

A good model for spike timing dependent plasticity should capture these two different phases of neural development, and keep most synaptic weights non-zero only for as long as necessary before that neuron can form a highly selective receptive field. In Section 7.3 (p. 116) I will further investigate how a group of highly selective neurons can be combined to form a sparse code, but for now I concentrate only on the receptive fields of single neurons.

5.3.1 Initial Synaptic Growth

All STDP rules presented in Section 5.2 initially require a minimum number of strong synaptic connections to allow a newly formed neuron to become active. When initial synaptic weights are too weak to induce spiking in the postsynaptic cell, no spike timing dependent changes to synaptic connections can happen (Section 2.3), and no spike timing dependent formation of receptive fields can begin.

Luckily, activity-dependent plasticity is not the only form of changes to synaptic connections that can occur. During the initial phases of neuronal formation, newly formed

neurons guide nearby axons towards their own dendrites through a variation of neurotrophic factors and other chemical gradients (Lindsay et al., 1994). The identity of attracted input axons is, if at all, only very loosely guided by neuron identity, and seems to create rather random synaptic connections. The newly formed postsynaptic neuron uses this to form some minimum number of strong synaptic inputs, after which axonal guidance subsides. The point at which the postsynaptic neuron stops attracting further incoming axons likely also depends on the energy consumption of sustaining strong synapses (Fonseca et al., 2004).

In a computer simulation of synaptic plasticity in point neurons, it may therefore make sense to include some initial synaptic growth factor that simply increases synaptic weights, and subsides as the neuron forms sufficiently many strong connections that keep it responsive to a select group of inputs. While it is yet unknown whether axonal guidance can be reenabled when a postsynaptic neuron looses all its strong synapses, or if a neuron that has lost all inputs is always simply replaced by a new neuron with a new randomly formed dendritic tree, both cases make little difference from a computational implementation perspective. In order to push initially silent neurons into a critical regime in which they respond to their synaptic inputs, I assume the existence of some (random) synaptic growth factor that is present only for neurons with an insufficient number of strong inputs.

The size of this growth factor only depends on the number of strong synaptic input connections and can be used to control activity-independent synaptic growth in the absence of postsynaptic firing. Activity-independent synaptic growth then counteracts the negative synaptic drift by biased STDP rules and keeps an untrained neuron responsive to inputs. While it keeps the neuron within a critical regime of activity, it does *not* interfere with spike timing dependent changes to synaptic strength, because changes are applied either equally to all synapses or randomly to all synapses with equal probability. When the number of strong input connections reaches some target given by the total maintenance cost of existing strong synapses, this unfocused synaptic growth factor subsides (see Appendix A.4.1 for implementation details).

Synaptic growth must not be confused with attractor-based STDP, as it is more powerful (attractor-based STDP can not initialise postsynaptic responses for newly formed neurons as it requires existing postsynaptic activity to take effect) and less interfering with spike timing dependent changes. Also, synaptic growth is not needed when the connections of a postsynaptic neuron are initially strong, as this usually gives the receptive field enough time to form in spite of negatively biased STDP. In fact, it is only required when new simulated neurons are formed that initially possess no strong connections as is the case in Chapters 9 and 10. Otherwise, attractor-less STDP is alone capable of forming highly selective receptive fields, as will be demonstrated in Chapters 6, 7 and 8.

5.3.2 Homeostatic Effects and STDP

In fact, STDP itself is likely the product of many different mechanisms interacting and should probably not be sought after in a single biochemical mechanism (Shouval et al., 2010). As noted in Chapter 2, the functional description of STDP has been constantly expanded throughout the past years, adding dependences on postsynaptic membrane states and chemical messengers into the equation. The observations that *in vitro* cell cultures of neurons grown outside the body produce certain types of weight distributions (Turrigiano et al., 1998) should therefore not be taken as indication that an STDP rule must always include the formation of a given distribution (e.g. unimodal) of synaptic weights *in vivo* (van Rossum et al., 2000). A simple counter-example is the idea that a combination of STDP and homeostatic mechanisms may produce a unimodal distribution of synap-

tic weights only in the absence of meaningful (e.g. random, unstructured) input data and produce a bimodal distribution of synaptic weights when meaningful input data (e.g. polychronous patterns, see Chapter 6) is given (Toyoizumi et al., 2007). When combining activity-independent synaptic growth as discussed in Section 5.3.1 with a negatively biased STDP rule, this is exactly what happens.

5.3.3 Inconclusive Experimental Data

Contrary to what is argued by van Rossum et al. (2000) and others, experimental evidence is not yet conclusive on the conditions under which certain distributions of synaptic strengths occur for any given postsynaptic neuron. While Turrigiano et al. (1998) find a unimodal distribution of synaptic weights for neurons grown in a petri dish, O'Brien et al. (1998) find a more multimodal distribution of synaptic strengths, albeit with a different approach to measuring synaptic strength as well as using a different preparation method and tissue from different brain areas.

A combination of homeostatic mechanisms that shows different types of synaptic weight distributions for different periods in a neurons developmental cycle therefore appears no less likely than a strict fixation upon any specific distribution measured in a subset of experiments.

5.4 Summary

In this chapter I examined a fundamental building block that will be used throughout the remainder of this text. After having introduced basic spike timing dependent plasticity (STDP) in Chapter 2, I explored the more subtle dependence of synaptic drift on the statistics of pre- and postsynaptic spike arrival pairings and defined and explored the effects of biased and unbiased STDP. I reviewed widespread models for weight-dependent scaling of STDP updates and demonstrated how they affect synaptic drift. I showed how *attractor-based* STDP interferes with spike timing dependent changes of synaptic connection strength and introduced a new family of weight-dependent STDP rules that are *attractor-less* and easy to implement in a computational model (Vogt and Hofmann, 2015a). Finally, I suggest that observed unimodal distributions of synaptic weight may have other origins than requiring an attractor-based STDP rule, and present an alternative approach that uses activity-independent synaptic growth to produce both unimodal and multimodal weight distributions as a function of the existence of spatiotemporal structure within presynaptic input spike trains. The embedding of spatiotemporal structure within presynaptic spike trains and the effect that this may have on pattern detection as well as plasticity will be the topic of the following two chapters.

Chapter 6

Spatiotemporal Coding, Correlations, and Noise

How does the brain encode important information? How is this information transported and processed by the electrical pulses of spiking neurons that we can observe in all animals that have a brain? We will not be able to observe the ground truth of every neuron in the complete brain for the foreseeable future. Even within a small area, the dynamics of each single neuron remain very hard to observe, due to the complexity of interactions between brain cells and the distorting effect of inserting electrode probes or certain chemicals into our brain. Today it is still not clear which (electrical, chemical, or physical) parameters in an area of brain tissue we can safely ignore and which may prove incredibly important for accomplishing some function that is yet to be discovered.

But we are getting a bit closer. It is by now accepted that the brain indeed uses little pulses of electrical currents to transmit ostensibly urgent information between neurons. For less urgent information and homeostatic/supporting processes, the brain likely uses chemical messengers such as calcium pulses of astrocite glia cells or hormones of the endocrine system. As I am mainly interested in fast information processing between neurons in this work, I will now take a closer look at how the brain may use fast electrical pulses between neurons to allow it to quickly respond to anything happening in the environment, be it as prey or as predator.

While biological experiments fail to provide us with all the answers on how spiking neurons may interact to produce behaviour, the field of information theory (Chapter 4) may give us some answers here. If a single neuron in the brain can be seen as a basic classification unit that has learnt to filter out specific features of its inputs and respond to others, it effectively works as a pattern detector by firing an action potential for known inputs when they are discernible from the huge number of unrelated background spikes, and remaining quiet when no known inputs are discernible. This discernibility may also be good to pass on as a measure of certainty that the detecting neuron has in its response, thereby extending a simple detector into a feature extractor that can give real-valued feedback about the world of inputs it receives. A likely candidate for this real-valued feedback about how certain or uncertain a neuron is in detecting a pattern of spikes in its inputs may be the time its membrane potential takes to reach the neuron's firing threshold for a given set of inputs. If spikes arrive through previously strengthened connections within a short time span, the neuron may respond faster than if those spikes only arrive occasionally, or through less strong connections, or if not all of those connections are equally strong. If not enough spikes arrive through sufficiently many sufficiently strong connections within a short time, the receiving neuron's response spike may be postponed

indefinitely.

In this chapter, I first take a look at the response characteristics of a spiking model neuron to three categories of input patterns without the inclusion of synaptic plasticity (Section 6.1). I then examine the effect of these three pattern categories on synaptic plasticity, especially STDP (Section 6.2). The results obtained here then lead to the question whether we should reconsider current assumptions and experimental practices when handling electrophysiological recordings because synchronous bursts of activity, variations in firing rates, and statistics on correlations and inter-spike intervals of single spike trains may be distracting us from searching for information encoded in a spatiotemporal neural code (Section 6.4).

6.1 Detection Uncertainty and Detection Speed

Before we look at the requirements that fast and reliable information processing poses on plasticity and learning, we first consider the task of actually detecting an environmental feature or situation within a noisy spiking neural network and how response time and certainty may be encoded.

Traditionally, recordings of the electrical activity of brain cells have only been able to observe a mixture of the action potentials of many co-located neurons in a given area. These local field potentials (LFPs) already showed some relation to external stimuli or motor effects, but soon single neurons were found to use electrical spikes of their membrane potential as a basic element of fast information transfer. These travelling action potentials were observed to change in rate when controlling muscles or signalling sensory input in the peripheral nervous system, and until this day the description of neural activity by its rate of firing is seen as a valid abstraction from the underlying dynamical complexity of biological neurons. In particular, even when spiking networks are observed in an experiment or used in a model, the exact timing of spikes is still often seen as unimportant while only the correlated activity among many neurons in a population or among many repetitions of some evoked response is taken into account. The exception to this rule are modelling studies where an exact neuronal spike train is tried to be perfectly reproduced through supervised learning (Frémaux et al., 2010; Gütig and Sompolinsky, 2006), leading to the opposite extreme of saying that the exact timing of every single spike were important.

In this section, I acknowledge the existence of randomness in timing or even occurrence of spikes in a biological neural network while examining how a few precisely-timed spikes may introduce structure into neural activity that may be used to encode meaningful information.

6.1.1 Detection Performance for Synchronous Inputs

Largely independent from the neuron model being used, a biologically realistic neuron tends to respond to its inputs by producing an action potential after receiving enough input to make its membrane potential cross some threshold (Chapter 2). Synchronously arriving input spikes evoke a quick response from a postsynaptic neuron if the synapses connecting the input units to the neuron are strong enough. If they are not, the postsynaptic neuron does not fire at all, as visualised in Figures 6.1 and 6.2.

As the strength of all incoming connections to a given postsynaptic neuron is increased, the neuron begins to respond to the presentation of a group of synchronous spikes. However, the response shows no information about the strength of connections (Figure 6.1) or how many strong connections have been formed (Figure 6.2) with a given group of

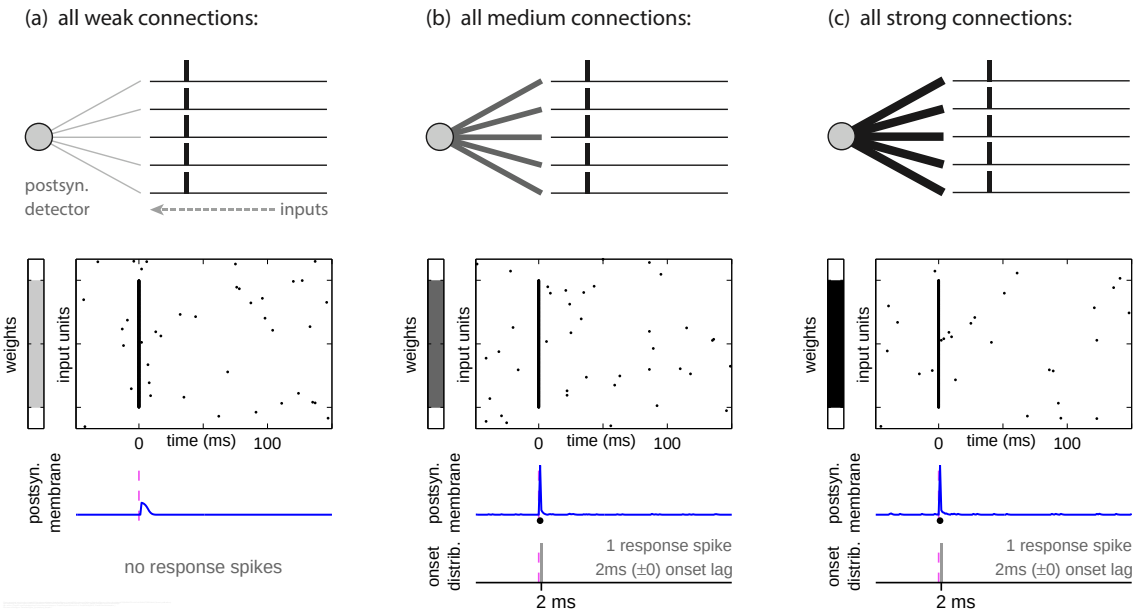


Figure 6.1: **Synchronous** input spike patterns allow only **binary classification** when all synaptic connections have a common **real-valued strength**. The postsynaptic output only relays minimal information about the receptive field and decision confidence. See Appendix A.3.4 for simulation details and Figure 6.2 for the description of rows. *Column (a):* A large number of weak connections is unable to elicit a postsynaptic spike. *Column (b):* A large number of medium-strength connections produces an immediate (2ms) postsynaptic response, without notable timing variability. The timing delay is equal to column (c) and to Figure 6.2 (column b). *Column (c):* A large number of strong connections produces an immediate (2ms) postsynaptic response as would be expected.

synchronously firing input units. If the neuron has a soft threshold for spiking (Chapter 2), it may show some slight difference between strong and medium inputs in the time the neuron takes to respond. However, this delay is negligible in a noisy system, as was discussed in Section 2.2.

Synchronous inputs hence lead to a purely binary response of a postsynaptic neuron, and no (or hardly any) information can be extracted from the neuron's response about how certain it is about its classification decision (to fire or not to fire). For later processing stages, this binary classification hampers further valuation of this neuron's response as it transports no information about possible uncertainty in the classification result. If one were to repeat this detection phase for a large number of postsynaptic neurons that all had slightly different sets of strong input connections (*receptive fields*), a third-stage observer that received all detector neurons' outputs would not be able to put more value on those detector responses that were made with certainty over those that are rather uncertain.

A large number of synchronously arriving inputs may therefore not be the best way of transporting meaningful data within a multi-layered spiking neural network, and indeed it has proven hard to find such perfectly synchronous spikes within the brain (Abeles and Gerstein, 1988; Gerstein et al., 2012). What has been seen, however, are groups of neurons that seem to become active together but are less than perfectly synchronous. Groups of correlated inputs with a correlation coefficient less than one (perfectly synchronous) but above zero (uncorrelated) are discussed next.

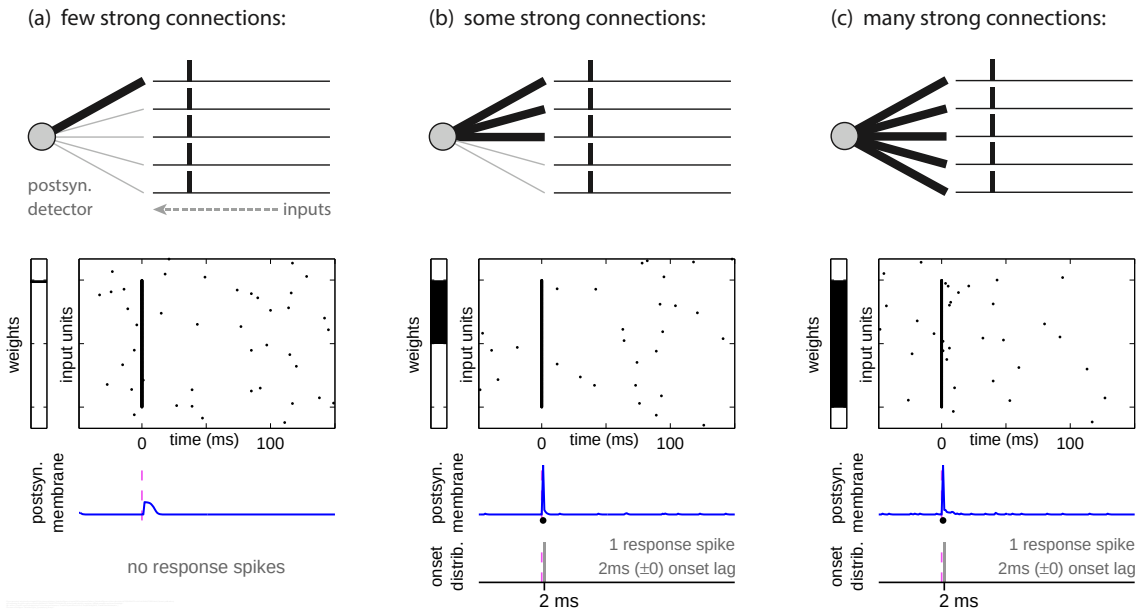


Figure 6.2: **Synchronous** input spike patterns allow only **binary classification** when a **fraction of binary synaptic connections** is strong. The postsynaptic output only relays minimal information about the receptive field and decision confidence. *Column (a):* A small number of binary strong connections is unable to elicit a postsynaptic spike. *Column (b):* A medium number of binary strong connections produces an immediate (2ms) postsynaptic response, without notable timing variability. The timing delay is equal to column (c) and to Figure 6.1 (column b). *Column (c):* A high number of binary strong connections produces an immediate (2ms) postsynaptic response as would be expected. *Row 1:* Overview diagrams of input spike pattern and the strength of synaptic connections to the single postsynaptic neuron. Synaptic connection strength is exemplified by thin grey lines for weak connections and wide black lines for strong connections. *Row 2:* Main plot: Actual input spike scatterplots used in the computer simulations. Left plot (within each column): Manually defined synaptic connection strengths (=weights) that were used to project the input spike pattern onto the single postsynaptic neuron. Nonexistent connections shown in white, weak connections shown in (light) grey, strong connections shown in black as in the overview diagrams. *Row 3:* Membrane potential of the single postsynaptic neuron (arbitrary units). A black dot below the membrane trace indicates the first postsynaptic response spike, while any grey dots (see following figures) indicate subsequent response spikes. The dashed magenta line indicates the time of pattern onset. *Row 4:* Distribution of timing variability of the first response. Onset distribution histogram was computed from 200 repetitions of the simulation. See Appendix A.3.4 (page 182) for simulation details.

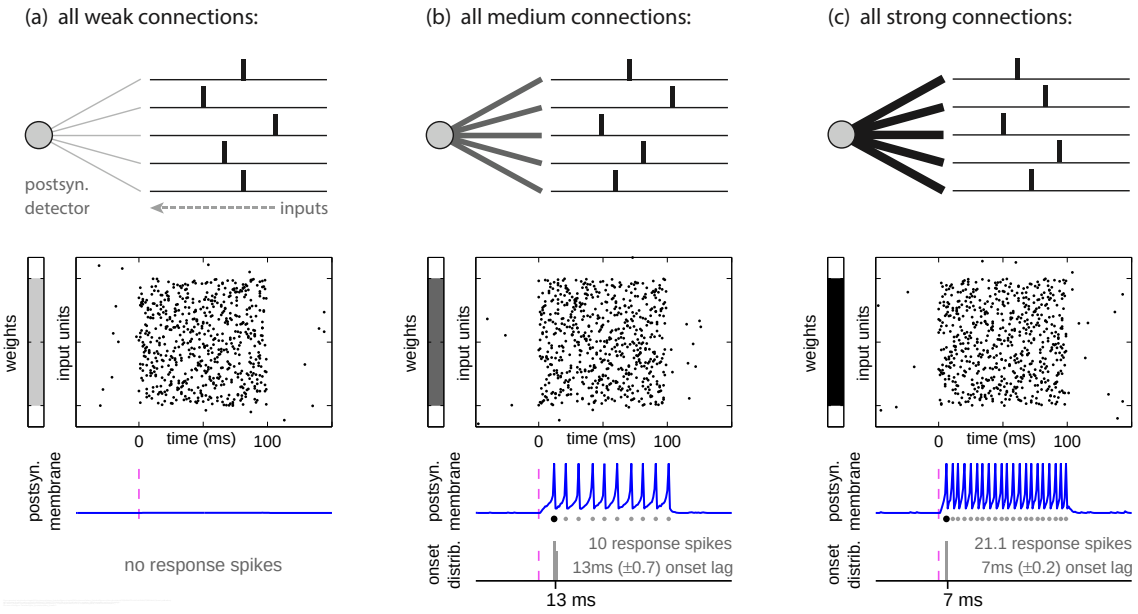


Figure 6.3: **Correlated** inputs allow the response onset delay to **encode the mean strength** of synaptic connections when all synaptic connections have a common **real-valued** strength. As the timing order of individual spikes from correlated input units is by definition random (undefined), many repetitions are required to find the average response delay. Correlated inputs allow more information to be extracted from responses of a postsynaptic neuron than if perfectly synchronous inputs were used (see Figures 6.1 and 6.2), but less than when truly polychronous inputs are used (see Figures 6.5 and 6.6). See Appendix A.3.4 for simulation details and Figure 6.2 for the description of rows. *Column (a):* A large number of weak connections is unable to elicit a postsynaptic spike. *Column (b):* A large number of medium-strength connections produces a slower (13ms) postsynaptic response, with some timing variability. The small variability in response times here is because we provide a near-constant number of input spikes to the postsynaptic neuron per time step, only distorted by little background activity (0.2 Hz). The response onset delay is equal to Figure 6.4 (column b) but slower than column (c) of this figure. *Column (c):* A large number of strong connections produces a fast (7ms) postsynaptic response after a sufficient number of inputs has arrived.

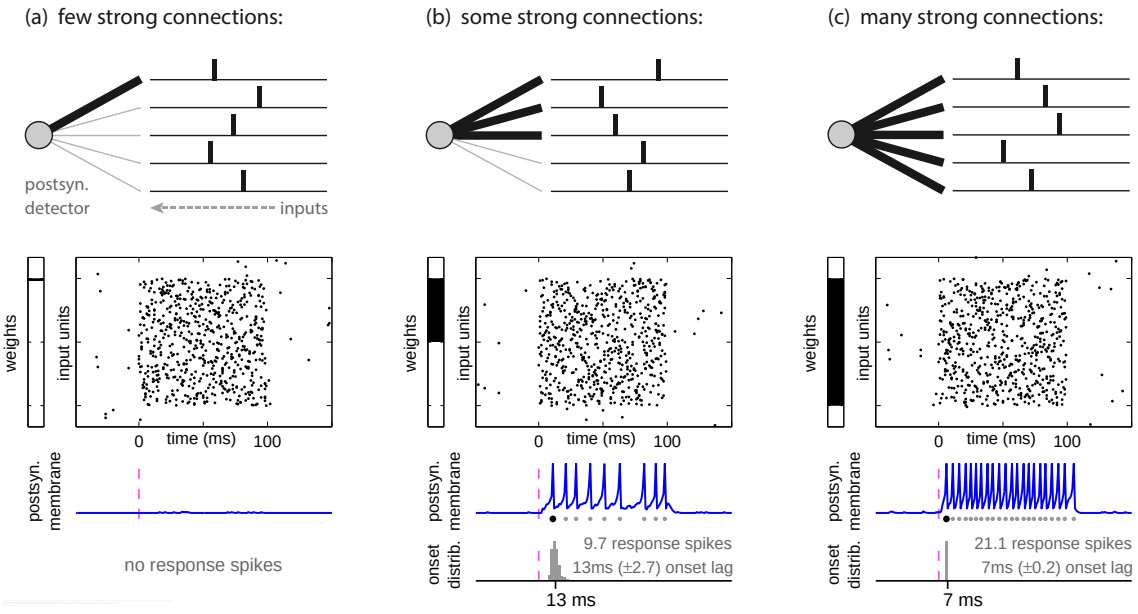


Figure 6.4: **Correlated** inputs allow the response onset delay to **encode the mean strength** of synaptic connections when a **fraction of binary synaptic connections** is strong. The membrane responses here are slightly more erratic than in Figure 6.3, but produce the same mean response times. When only some weights are strong (column b), the timing of first responses is more variable than in any other case shown in this section. See Appendix A.3.4 for simulation details and Figure 6.2 for the description of rows. *Column (a):* A small number of binary strong connections is unable to elicit a postsynaptic spike. *Column (b):* A medium number of binary strong connections produces a slower (13ms rounded mean) postsynaptic response, with much timing variability. This is because only a fraction of arriving spikes actually increase the postsynaptic membrane potential, and the time until the postsynaptic membrane reaches spike threshold therefore depends on which input units randomly fire first on a given pattern presentation. We still provide a near-constant number of input spikes to the postsynaptic neuron per time step, as in Figure 6.3. While there is more variance, the mean of the response onset delay is equal to Figure 6.3 (column b) but slower than column (c) of this figure. *Column (c):* A high number of binary strong connections produces a fast (7ms) postsynaptic response after a sufficient number of inputs has arrived.

6.1.2 Detection Performance for Correlated Inputs

Correlated inputs are those spike trains that arrive from input units which have some tendency of being active together. While the amount of correlation between two spike trains can be easily calculated (Chapter 4), it may not describe the complete relationship between two spike trains sufficiently. However, as it is easy to detect correlated activity in many biological recordings of spiking neurons, this approach to modelling synaptic inputs of a spiking neural network is currently the most commonly used method for testing the performance of spike timing dependent plasticity (STDP).

If we present a detector neuron with correlated inputs through strong connections, it will have a tendency to quickly respond to the peaks of correlated activity (Figure 6.3 right). If the connections were very weak, the postsynaptic detector neuron may not respond at all (Figure 6.3 left). But if the correlated inputs arrive through synaptic connections with some medium strength, we see a different behaviour than in the synchronous case:

As the inputs are no longer perfectly synchronous, a smaller number of input spikes is arriving at the postsynaptic neuron at any moment in time. During a peak of input activity, the membrane of a postsynaptic cell is drawn towards its firing threshold only slowly, and may remain beneath this threshold for the time being. Without further inputs, leak currents would cause the neuron's membrane potential to return back to resting potential. But if the heightened input activity continues, the postsynaptic neuron's membrane potential can further increase above this threshold, causing a response spike (Chapter 2). The slope of this comparably slow ramp of membrane potential during a peak of input activity depends on the strength of connections between the spiking input units and the postsynaptic neuron. Therefore, for a fixed number of correlated inputs, the time that a detector neuron takes to respond to a peak of correlated inputs contains information about how strong the synaptic connections to this correlated input group are (Figure 6.3). In theory, the delay that a detector neuron takes to respond to the onset of a peak in correlated activity may pass on a hint on how certain it is that a given pattern of correlated inputs that the detector is tuned to is present.

However, in practice, all synapses connecting a group of correlated inputs to a postsynaptic neuron rarely have exactly equal strength. Also, the mean number of inputs arriving at each point in time is only sufficiently smooth for very large groups of input units. During a peak of correlated input activity, some inputs may coincidentally fire early and at the same time be connected to the postsynaptic detector neuron through strong synapses, causing it to respond a lot sooner. Similarly, if coincidentally all early firing input units of an activity peak happen to be connected through less strong synapses, they may not excite the postsynaptic neuron enough for it to respond quickly. Just as randomly, it may happen that most inputs that have strong connections with a postsynaptic neuron fire late during a peak in correlated input activity, causing the detector neuron to respond late, too. Therefore, in a more realistic setting, correlated inputs lead to a high variance of response timings in the postsynaptic detector neuron (Figure 6.4 center).

In this case, it is hardly possible to infer any statement about a detector neuron's certainty in reaching a decision by observing the time it took to respond in a single instance. As we will see in section 6.2, this random response timing for correlated inputs is a main factor for sub-optimal STDP performance when using merely correlated spike trains as input data.

I will therefore now examine the idea of temporal coding (Thorpe et al., 2001) for detecting structured information within a stream of incoming spikes, and explain how any uncertainty in a pattern detector neuron's classification decision can thereby be passed on

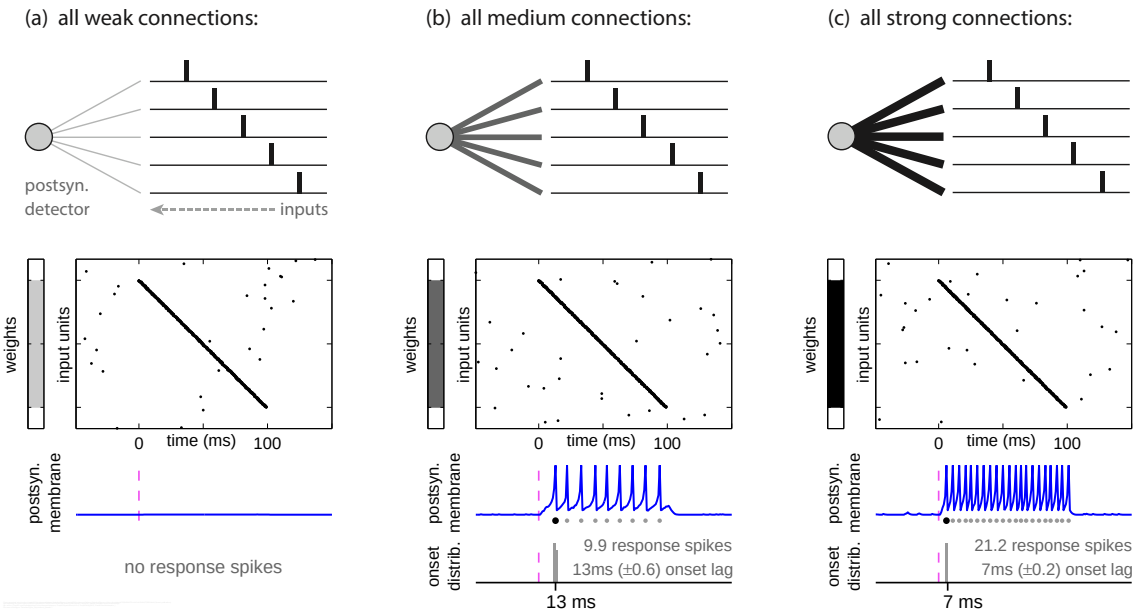


Figure 6.5: **Polychronous** inputs allow the response onset delay to **encode the strength of a subset** of synaptic connections. When all incoming synaptic connections have a common **real-valued** strength, the response is similar to that of correlated inputs when the same average number of spikes arrives on each time step/bin (Figure 6.3). See Appendix A.3.4 for simulation details and Figure 6.2 for the description of rows. *Column (a):* A large number of weak connections is unable to elicit a postsynaptic spike. *Column (b):* A large number of medium-strength connections produces a slower (13ms) postsynaptic response, with little timing variability. The response onset delay here is similar to Figure 6.3 (column b) but *slower* than Figure 6.6 (column b) because the earliest set of inputs arrives through a stronger subset of connections there. Only polychronous patterns take account of which subset of connections transmits the earliest inputs. For synchronous input patterns (and zero-lag correlation) as in the previous figures, only the mean connection strength counts. *Column (c):* A large number of strong connections produces a fast (7ms) postsynaptic response after a sufficient number of inputs has arrived.

to later processing stages.

6.1.3 Detection Performance for Polychronous Inputs

Spatiotemporal coding of synaptic inputs to biological neural networks have long been acknowledged to exist in many parts of the brain (Bair and Koch, 1996; Gawne et al., 1996; Hopfield, 1995; Mainen and Sejnowski, 1995; O'Keefe and Recce, 1993). A strong proponent argument for temporal coding was made in (Thorpe et al., 2001), but the general neuroscience community has continued to model synaptic inputs to spiking networks as merely correlated groups of input units (with random spike order) and thereby avoids the question of how such structured inputs may look. Izhikevich et al. (2006; 2004) found that a recurrent plastic network tends to produce repeating groups of spatiotemporally structured spikes and coined the term of *poly*chronous patterns which I use here for describing spatiotemporal inputs in a feed-forward setup. While it has been hard to detect repeating patterns of exact spatiotemporal ordering within biological recordings so far (Berger et al., 2010; Gerstein et al., 2012; Schrader and Grün, 2008), there are many hints that point to the existence of such a spatiotemporal neural code, albeit not in a perfectly precise manner. Starting with the transmission of detection uncertainty as described above, it seems logical that the brain as a whole needs some mechanism of passing on each neuron's "confidence" in its own classification decision. While a neuron's spike delay is a good candidate for this kind of information, it is of course not conclusive evidence for the existence

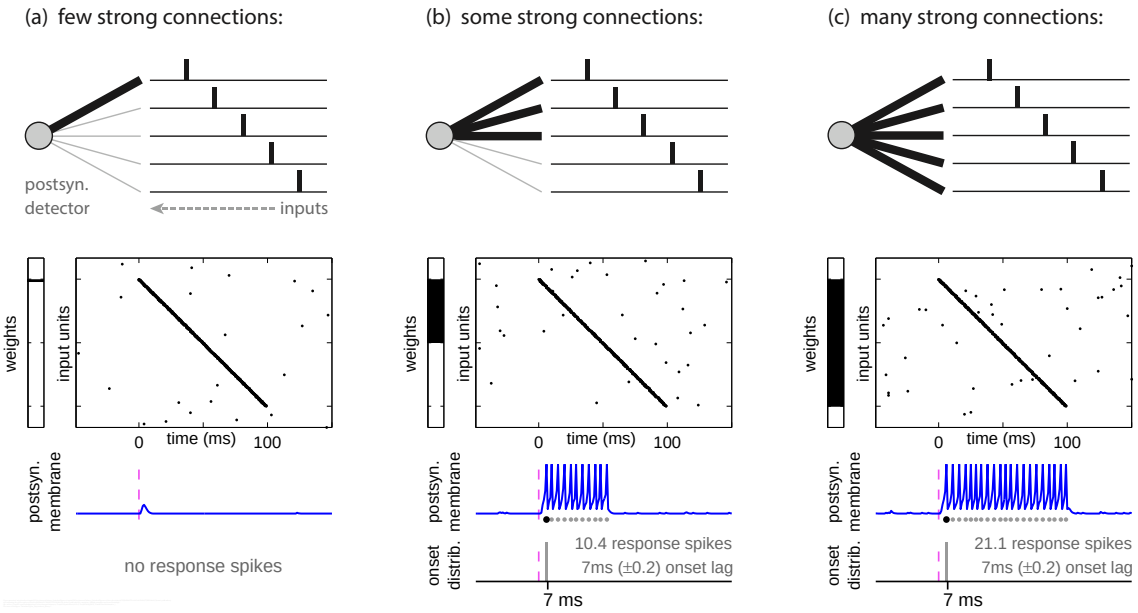


Figure 6.6: **Polychronous** inputs allow the response onset delay to **encode the strength of a subset** of synaptic connections. When only a **fraction of binary synaptic connections** is strong, the delay until first response depends strongly on how many of the earliest inputs arrive through synaptic connections that happen to be strong. As in polychronous patterns the order of spike arrival is by definition *not* fully random, the response delay for a given set of weights allows inferences on how well the incoming pattern fits the detector neuron's receptive field. If the pattern itself is unreliable (Sections 7.1.2 to 7.1.4), response delay may also encode certainty about pattern presence within background noise. See Appendix A.3.4 for simulation details and Figure 6.2 for the description of rows. *Column (a):* A small number of binary strong connections is unable to elicit a postsynaptic spike. *Column (b):* A medium number of binary strong connections produces a fast (7ms) postsynaptic response, iff the subset matches the earliest inputs. Only polychronous patterns take account of which subset of connections transmits the earliest inputs. *Column (c):* A high number of binary strong connections produces a fast (7ms) postsynaptic response after a sufficient number of inputs has arrived.

of spatiotemporal coding in deep, fast-responding parts of the brain. A second hint for the existence of spatiotemporal coding though comes from the shape of typical spike timing dependent plasticity rules and will be the subject of the remainder of this chapter (see Section 6.2). First, however, I will now describe the effects of spatiotemporal coding on detection performance and detection speed, and explain a useful scenario for delaying a neuron's response.

Figures 6.5 and 6.6 show an example of a perfectly timed, precisely ordered spatiotemporal code that is used as input to a single detector neuron in a feed-forward setting. While very weak inputs (Figure 6.5 left) still evoke no postsynaptic spike as any deviations in the membrane potential are too weak to pass the neuron's firing threshold, the neuron does respond to slightly stronger (Figure 6.5 center) and strong (Figure 6.5 right) inputs. In opposition to a synchronous burst of input activity (Figures 6.1 and 6.2), the postsynaptic detector neuron now shows a response time delay that is somewhat inversely proportional to the synaptic strength of connecting synapses. In opposition to the case of merely correlated inputs (Figure 6.4), the response delay for a mixed group of strong and weak synapses is now reliable (Figure 6.6 center) and depends only on the identity of which connections are strong at the time of transmission. For a given set of synaptic weights, the response delay of a spiking neuron that receives such a spatiotemporal input pattern is much more predictable than if it received randomly firing correlated inputs. This predictability is what allows later processing stages to read out information from the firing delay of the detector neuron. The onset time for measuring this delay can be signaled through multiple events, and I introduce possible candidate events together with an option for solving the onset problem in section 6.1.4.

We have just established that the detector neuron's response delay can convey the average strength of synapses it has formed to input units that fire early in a precise spatiotemporal order. It signals the match between its receptive field and the spatiotemporal pattern being presented to it. This information is transmitted in a single spike by a single detector neuron, while a correlation code requires either a large postsynaptic population or many repetitions to access this information. For synchronous inputs, this information is simply not available without testing a range of synaptic connection strengths for each occurrence of synchronous inputs.

The goodness of fit between a detector neuron's receptive field and an incoming pattern is, however, not the only information that can be passed on through the delay of a detector neuron's response after some onset event. When a given set of inputs tends to fire in some spatiotemporal order to represent a message of information (Chapter 4), a corrupted or partial message may be represented by some of the inputs remaining quiet or firing out of order. If more early-firing inputs remain quiet than usually would if the message were uncorrupted, the response of the postsynaptic detector neuron is also delayed. If all or most inputs that would normally take part in presenting a spatiotemporally ordered pattern of spikes to a detector neuron remain quiet, the detector neuron's response may be delayed infinitely. The response delay therefore also contains information about how clear the pattern of spatiotemporal spikes is present within the stream of inputs. Similarly, if some input units show earlier or multiple spikes during pattern presentation, this may decrease the response time of a postsynaptic detector neuron if no adaptive countermeasures are in effect. The response delay therefore can also signal the presence, or amount of detectability, of a pattern within a stream of continuous inputs.

How much of the detector neuron's response delay is caused by a pattern's inherent detectability or contrast, and how much is caused by how well the detector is tuned to that pattern? This may not be easily extracted through observation of a single spike from

the detector neuron alone. If, however, two (or more) detector neurons were to receive the same inputs and one detector responded earlier than the other, the detector neuron with the earlier response must be better tuned to the occurring pattern. This can be easily tested by giving two detector neurons identical inputs so that the different response timings must be due to different receptive fields. The total delay observed in the second detector neuron is then not just due to a weak contrast or corruption of the presented message/pattern, but at least in part due to a worse match between the second detector's receptive field and the pattern that was received. While a pattern's inherent detectability or contrast may cause similar response delay in all neurons, divergent receptive fields cause a relative difference in response timing. In this way, later processing stages can extract valuable information about the relative tuning quality of all detector neurons even if the absolute influence of a pattern's inherent detectability within a stream of input spikes remains unknown. This will be further established in section 6.1.4. The shortcomings of unreliable transmission of detection uncertainty to downstream targets will play an important role in Section 6.2, when STDP is used to tune a neuron to reliable sources.

A major cause for bad detectability of a pattern that encodes a given message of information is *noise* and the pattern's *signal to noise ratio*. As discussed in Chapter 4, noise can be introduced into a system through multiple sources, and a main goal of any classifier or pattern detection system is to filter out reliable information from a haystack of random background data. For a neuron, this goal translates to the task of deciding which of its inputs to ignore and which to pay attention to, via the distribution of strengths of its synaptic connections. While it is easy to set up a model neuron that has maximal connections to always-reliable inputs and has no or very weak connections to always-random inputs, the task becomes more difficult when input units transmit a mixture of random noise and meaningful spikes. The default solution to this is to require multiple excitatory inputs to become active together, where a concerted increase in activity on strong synapses signals the postsynaptic neuron to fire. However, this does not mean that the inputs must be randomly correlated or even synchronous, but also covers polychronous codes. How sudden and how far this activity must be increased for successful detection is defined by the detector neuron's leak current (Section 2.2.2) and/or a combination of other (homeostatic) factors. A too weak leak current leads to many false alarm detections (=false positives), while a leak current that is too strong may make the detector neuron miss (=false negatives) all but the most synchronous inputs.

Equally important as being able to respond to the existence of a given message within a stream of input data (=true positives) is the ability to remain quiet if the message to be detected is *not* present. Correct rejections (=true negatives) of distractor patterns are rarely explicitly tested when describing the response of a biological neuron to some real-world stimulus in experimental neuroscience publications. This is owed to the fact that instantaneous rates and LFPs are often used as basic form of measurement, while the response properties of single neurons are considered to be highly stochastic. Responses of larger groups of neurons are only considered if they show a population response that is measurable in the LFP, while pattern-specific ordering of the single spikes within population responses are hardly ever tracked. In section 6.1.4 I show how information may be contained in the spatiotemporal structure of population responses while the LFP shows no difference between responses.

In the input layer, if the increase in activity of a few inputs is not masked by a similar-sized decrease of activity of some or all other inputs during that time, the mean rate over all inputs increases slightly and can be detected in the local field potential (LFP) of the input layer. This is usually the case when modeling perfectly synchronous or slightly

correlated input codes, as it is hard to tell how many of the other inputs should be used for compensation. When using a spatiotemporal code, however, the occurrence of a spatiotemporally structured pattern can be easily hidden from the LFP while postsynaptic detector neurons are still able to detect it. This will be demonstrated in Chapter 7 with the pattern types introduced in Section 7.1.1 and shows that much more information can be encoded/hidden in spatiotemporal patterns within the spike train than is extractable from a recorded LFP alone.

As the delay of responses to polychronous inputs encodes a neuron's "confidence" in its own detection decision, decisions that are made with high confidence are transmitted earlier. This ensures that the minimum time for reaching a safe classification decision is taken by each detector neuron, while waiting for more inputs in favor of supporting evidence when the earlier inputs have been inconclusive. This *shortest-time-to-decision* code lets the single postsynaptic neuron respond earlier to easy decision tasks while it spends more time gathering evidence on more difficult decision problems.

6.1.4 Multiple Detector Neurons recreate a Spatiotemporal Code

As proposed above, a detector neuron's firing delay signals both the match of its receptive field to a given stimulus and the degree to which that spatiotemporal stimulus is present within some noisy background activity. Any later processing stage can use this information to decide how much it trusts the detector neuron's decision to fire. As the best detectors for a given pattern respond first, early responses should be given the most consideration by later processing stages.

Figures 6.7 and 6.8 show the responses of 100 postsynaptic (detector) neurons with different receptive fields as they detect a given spatiotemporally structured input pattern. As the pattern in Figure 6.7 is clearly visible within the background activity, it is detected by all postsynaptic neurons, but across a timespan of many tens of milliseconds due to the different receptive field of each detector neuron. When the pattern is less clear within a

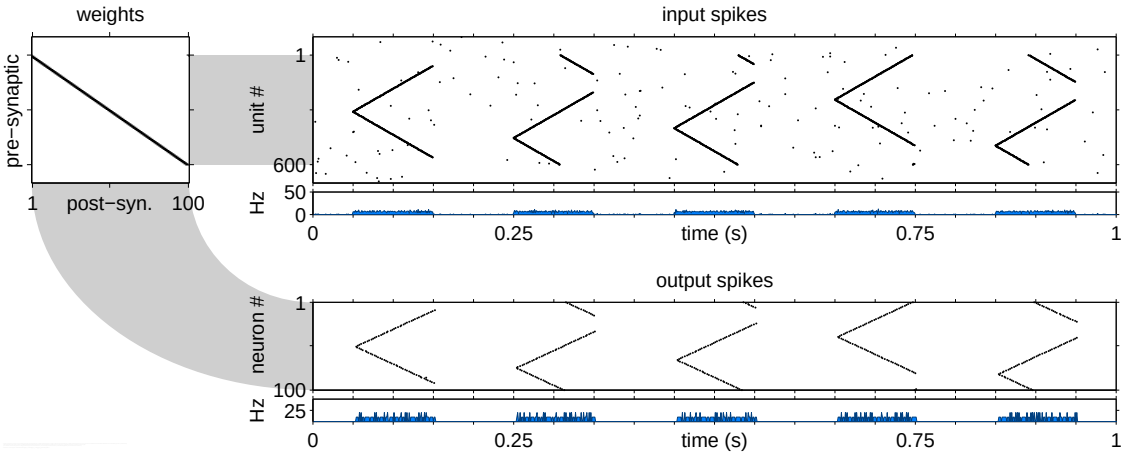


Figure 6.7: **Multiple Detectors** can recreate a polychronous spike pattern when patterns preserve neighbourhood (Sections 6.3 and 7.1.7) and connecting weights are sparse and evenly map the full space of inputs to a group of output neurons. The group response gives a hint on the relative tuning of each detector neuron when all receive the same input. The polychronous input patterns seen here were generated with method A of Section 7.1.1 (p. 104), and overlaid onto a background of random spikes with 0.2 Hz firing rate. Pattern presentation can therefore also be seen in changes to the population firing rate of the group of input units. See Appendix A.3.4 and Table A.13 (p. 184) for further simulation settings. In the spirit of the diagrams in Figures 6.6 and 6.9, grey connecting ribbons between the panels here visually connect the input and output spike trains via the weight matrix.

noisy background (Figure 6.8), the response of all detectors becomes slightly delayed and more jittery due to random noise spikes among the background inputs. Still, the general ordering of responses is kept equal to that of input spiking, thanks to the diagonal weight matrix that was set by hand here.

While the relative tuning of each detector neuron to each pattern is encoded in its relative response delay, the overall pattern detectability within the stream of noisy input spikes leads to a constant delay of detection. If later processing stages wanted to measure a detector's response delay since pattern onset, they would require at least one direct connection from an input unit that fires early within that pattern, in addition to receiving the outputs of the detector neurons. This information would be needed for each different possible pattern, so would require connections from many input neurons in a network. On top of this, the activity of single noisy input neurons is inherently unreliable, which is why the detector neurons are needed in the first place. Therefore, it will usually be impossible for later processing stages to measure a detector neuron's response delay since stimulus onset time.

However, this may not be much of a problem. We have established above that postsynaptic neurons that work to detect the earliest part of a given spatiotemporally ordered (polychronous) pattern should only fire when they have gathered enough evidence to make their decision. Therefore the earliest detector response signals the earliest point in time when any later processing stage can reliably assume the existence of a given stimulus within the environment. So although the ground truth of when the stimulus appeared in the environment is not available to most of the brain, the relative delay of detectors may contain enough information as it represents the earliest time this stimulus can be *reliably* detected. Candidates for onset events from which the delay of detector responses should be measured are therefore the earliest responses that arrive from the group of detectors. From the point of view of a later processing stage, the earliest arriving response would have a delay of 0 ms, while the delay of any later responses is measured from the arrival time of the first response only.

In a multi-layered network like the brain, if a small group of neurons proves to be reliable in signaling the start time of some stimulus, synaptic connections may be formed

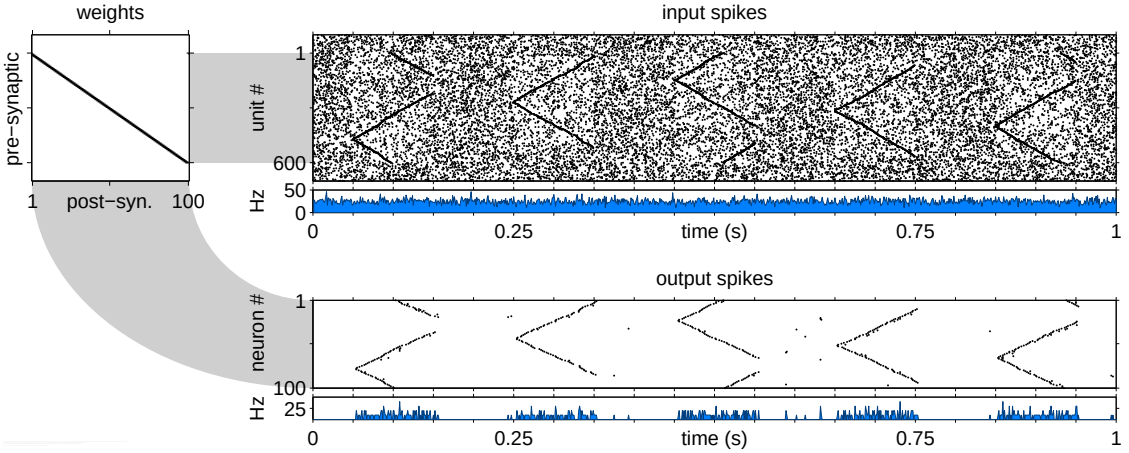


Figure 6.8: **Multiple Detectors** can recreate a polychronous code even when the inputs are very noisy. Here, patterns were created directly from 25 Hz background noise (pattern type D, Section 7.1.1, p. 104), thereby preserving a homogeneous input firing rate that contains no information about pattern presence. While the weight matrix was set by hand here, we will see in the next section and chapter 7 that STDP can achieve a similar sparse map of receptive fields in some cases without requiring recurrent inhibition (esp. Figure 7.11, p. 119). See Appendix A.3.4 and Table A.14 (p. 184) for simulation settings.

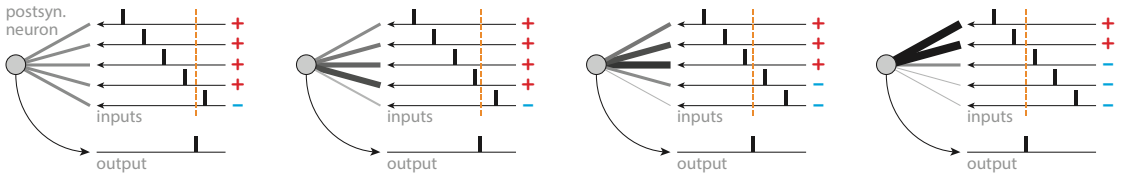


Figure 6.9: On **polychronous inputs**, STDP gradually increases connections to early-spiking inputs and gradually weakens connections to late-spiking inputs. The response of the postsynaptic neuron happens while the incoming pattern is still being presented (output train and orange dashed line). As learning progresses, the response time is reduced until only the minimally required set of connections that still evokes a response remains strong. This represents a sparsification of the synaptic weight distribution. Light grey thin lines indicate weak synaptic connections, dark grey thick lines indicate strong connections. Red plus signs indicate a potentiation of synaptic connection strength and blue minus signs indicate depression.

that skip a few layers. But as early processing stages also tend to be less reliable in their detection performance, it is probable that most of those connections will decay over time as Hebbian synaptic plasticity that has negative synaptic drift only supports reliable synaptic connections, as will be now demonstrated.

6.2 Synaptic Plasticity, Pruning and Sparseness

In Section 6.1 we established that polychronous patterns allow the responses of a postsynaptic neuron to carry more reliable information in response timings than when synchronous or correlated (=random spike order) input spikes are used. When polychronous patterns are used, more information can be transmitted, including some notion of *detection (un)certainty*. This has direct implications when a synaptic plasticity rule can actually take advantage of such precise timings.

In this section, I point out the advantages that spike timing dependent plasticity (STDP) has over traditional Hebbian plasticity when dealing with a precise spatiotemporal spike code, and how this advantage is lost when stochastically spiking or synchronous inputs are used. I argue that STDP has been undervalued as being comparably as powerful as traditional Hebbian learning and that some behaviours of STDP that have been treated as detrimental to learning success are actually very useful features when the learning problem is rephrased. By examining the type of tasks that STDP is good in solving, inferences can be made as to the typical learning tasks that neurons in our brains face, and typical spike codes that may prevail in brain areas that feature STDP.

I now use the high reliability of the precise response delay to explain how STDP finds the start of spatiotemporally structured input patterns in continuous input streams (Masquelier et al., 2008), and point out some additional observations that have so far seen little attention (Section 6.2.1). After combining various versions of STDP (as introduced in Chapter 5) with various families of input patterns (partially introduced in the previous section) and comparing their success in forming selective receptive fields, I then discuss similarities of the code that STDP promotes to efficient codes from information theory and explain how automatic pruning in STDP circumvents the need for accessing some original stimulus for computing a reconstruction error as needed by more abstract gradient descent algorithms to function (Section 6.4).

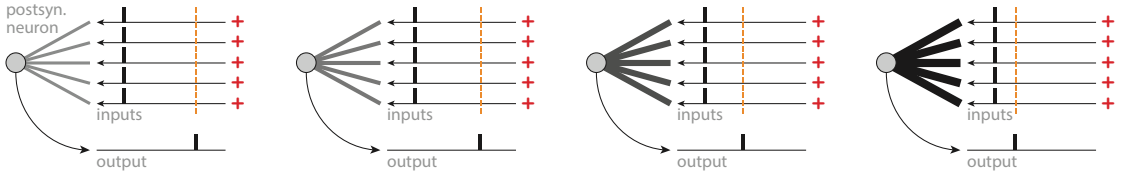


Figure 6.10: On **synchronous inputs**, STDP increases all connections until all are strong. The response of the postsynaptic neuron always happens after the incoming pattern (output train and orange dashed line), so all connections are strengthened (red plus signs) to their maximum. The response time of the postsynaptic neuron is reduced as far as membrane dynamics permit. No sparsification takes place. Line indicators and colours as in Figure 6.9.

6.2.1 STDP finds the earliest reliable predictors of a message

Spike timing dependent plasticity has been shown (Guyonneau et al., 2005; Masquelier et al., 2008) to tune a postsynaptic neuron to those input units that tend to fire a spike during the start of a repeating polychronous pattern. This has been described with some puzzlement in Masquelier et al. (2008) and has been tried to be counteracted by suppressing simultaneous responses through mutual inhibition between multiple postsynaptic (detector) neurons. A second feature of STDP that has seen little attention is that it not only strengthens connections to early inputs but also actively weakens connections to late firing inputs. This is visualised in Figure 6.9, where a connection that was initially strengthened (because it initially always fired before a postsynaptic spike) is later weakened when the postsynaptic response starts to happen earlier than inputs through this initially strengthened connection. This feature will prove to be very useful in all following sections and chapters and only works if the patterns are polychronous (compare Figures 6.9, 6.10, 6.11).

The general function of STDP when working with spatiotemporally structured patterns can be understood by imagining the following sequence of events:

1. At first, the postsynaptic neuron just fires randomly. Therefore it also happens to fire occasionally while a polychronous pattern is being presented.

2. Whenever the postsynaptic neuron fires, this causes a strengthening of all connections to input units that have recently fired a spike. It will also cause a decrease of connections to all input units that happen to fire after the postsynaptic neuron (assuming negligible axonal and dendritic conduction delays).

3. Over time and many spike pairings, connections to input units that fire together become stronger than connections to randomly firing input units. This is because inputs that tend to fire together have a higher chance of evoking a postsynaptic response and therefore are potentiated more often than they are depressed. Randomly firing inputs, on the other hand, are depressed equally often as they are potentiated, causing them to perform a random walk. See also Figure 5.2 (p. 41). As a result, the receptive field of the postsynaptic neuron is shaped towards the presented pattern.

4. The next time the pattern is presented, the postsynaptic neuron has a slightly higher probability to respond. Also, as it reaches its firing threshold a little earlier, it will respond a little sooner, too. (compare Figure 6.6 center and Figure 6.9)

5. Connections to those early inputs that fired before the postsynaptic response are further strengthened. Connections to inputs that (now) fire after the postsynaptic neuron are weakened.

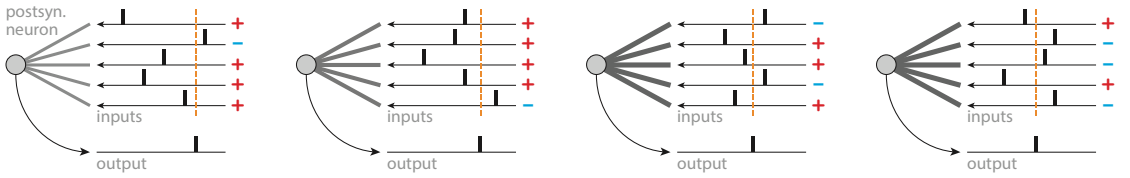


Figure 6.11: On **correlated inputs**, STDP increases all connections until the postsynaptic neuron regularly fires early, but the mean of connection strength remains at intermediate levels. The response of the postsynaptic neuron happens while the incoming pattern is still being presented (output train and orange dashed line), but the identity of early inputs vs. late inputs constantly changes. Therefore, the mean of connections neither moves towards its maximum (as for synchronous inputs) nor is there any structural specification (as for polychronous inputs). Each connection strength underlies a random drift that depends on the specifics of the STDP rule and pairing statistics as seen in Sketch 5.2. Line indicators and colours as in Figure 6.9.

6. Continue this (go to list item 4) until the minimal group of earliest inputs that is still large enough to elicit postsynaptic firing has been found. (compare Figure 6.9)

Figures 6.12 and 6.13 combine an attractor-less weight-dependent STDP rule with polychronous input patterns. A subset of 2000 presynaptic input units repeatedly present this pattern to a single postsynaptic output model neuron, while also taking part in random background spiking. While the simulation shown in Figure 6.12 uses negatively biased STDP, Figure 6.13 shows the results of the same simulation setup when unbiased STDP is used (for STDP parameters see Table A.4, p. 181).

Let us first look at the similarities between the two simulation results. Both figures show how the single postsynaptic neuron quickly succeeds in forming strong connections to the earliest spiking units that present the pattern (ca. units 700-750) within the first ten seconds of simulation. This causes the postsynaptic neuron to respond early after the start of each pattern presentation. In Figure 6.12, we can see that initially the neuron does not quite tune to the earliest inputs but rather chooses inputs near units 750-800 to form connections with, resulting in an initial response delay of ca. 20ms. However, within the first ten seconds of simulation, the strong connections have changed to input units 700-750 and remain there, allowing the postsynaptic neuron to reliably respond to pattern onset with only 5ms delay. This is the mentioned predictive learning feature of STDP (Farries and Fairhall, 2007; Guyonneau et al., 2005) and can also be seen very clearly in supplementary Figure A.1 (p. 174).

As was announced above and indicated in the diagrams of Figure 6.9, connections to pattern-presenting input units that regularly fire *after* the postsynaptic response (units 800-1300) are actively weakened. That is, they decrease faster than connections to units that only fire random spikes and do not take part in pattern presentation (units 1-700 and 1301-2000). The fact that connections to randomly firing units decrease at all is because Figure 6.12 uses negatively biased STDP (see Chapter 5). Taken together, predictive learning of early inputs and active weakening of late inputs allow the combination of STDP and polychronous spike patterns to produce a sparse distribution of synaptic weights that makes the postsynaptic neuron highly selective, without requiring any external teaching signal. For brain regions that need to minimise the response time to incoming stimuli including the basal ganglia, such a plasticity paradigm is likely vital for allowing an animal to learn fast and precise reactions to the earliest reliable indicators of events happening in the physical world.

While both biased and unbiased STDP (Chapter 5) strengthen early inputs and weaken late inputs, they each show a different behaviour for those input units that fire randomly

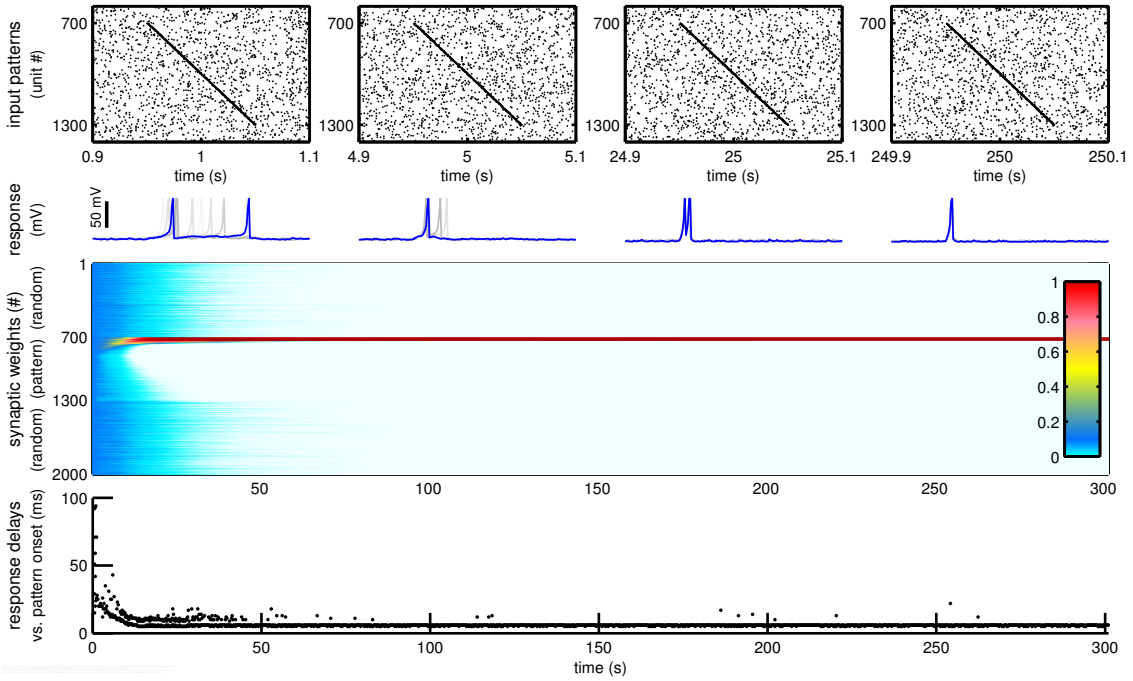


Figure 6.12: **Polychronous input patterns** with **attractor-free biased STDP** reliably form a sparse receptive field that detects the earliest parts of a polychronous spike pattern and **nothing else**. Simulation settings of this and all following similar figures are given in Appendix A.3.5. *Row 1:* Input patterns at four different times during the simulation. The pattern is presented by 600 input units (#701-#1300) and all other 1400 input units only present random poisson-distributed spikes at 10 Hz. *Row 2:* Membrane responses of the single postsynaptic neuron (blue). X-axis is equal to row 1. Responses to two immediately preceding and two immediately succeeding pattern presentations shown in grey as an indicator of response variability. *Row 3:* Weight development of incoming connections to the single postsynaptic neuron over the course of simulation. $w_{min} = 0$ and $w_{max} = 1$. *Row 4:* Response delay plot over the course of the simulation. X-axis shows simulation time. Y-axis shows the delay, if any, of postsynaptic spikes after the start of each pattern presentation. The polychronous pattern was presented every 200ms, and lasted for 100ms, allowing for 100ms of unstructured (poisson-distributed) noise between patterns.

and therefore cause random spike timing differences between pre- and postsynaptic neurons. When timing differences occur randomly in a uniform manner (pre- and postsynaptic neurons are uncorrelated), the sign of the STDP rule's integral gives the direction in which the strength of a synapse drifts. To take advantage of a depressing synaptic drift for random inputs, most STDP publications in the literature use negative-integral STDP rules (see Table A.1, p. 177) or argue towards the existence of some weak attractor that is close to the minimum weight and has a similar effect (see Table A.2 and Chapter 5).

In Figure 6.13 an unbiased STDP rule is used for learning polychronous patterns. Here, many synaptic connections that do not take part in pattern presentation (units 1-700 and 1301-2000) become strong in contrast to Figure 6.12. However, the postsynaptic neuron still forms strong connections to the earliest pattern-presenting inputs and decreases connections to late-firing pattern-presenting units as before. The formation of many randomly strong connections here is due to the condition that the postsynaptic neuron is allowed to respond freely to its inputs (compare Gütig et al., 2003; Rubin et al., 2001; Song et al., 2000). In Chapter 5, this was not yet allowed, and an unbiased STDP rule there instead lead to a uniform distribution of weights for randomly occurring spike pairings (see Figure 5.6, column A, rows 6 and 7, grey area, p. 49). This means that some connections to randomly firing inputs become strong although they do not contain any causal relationship with postsynaptic spikes. In turn, this produces far more noise in the responses of the

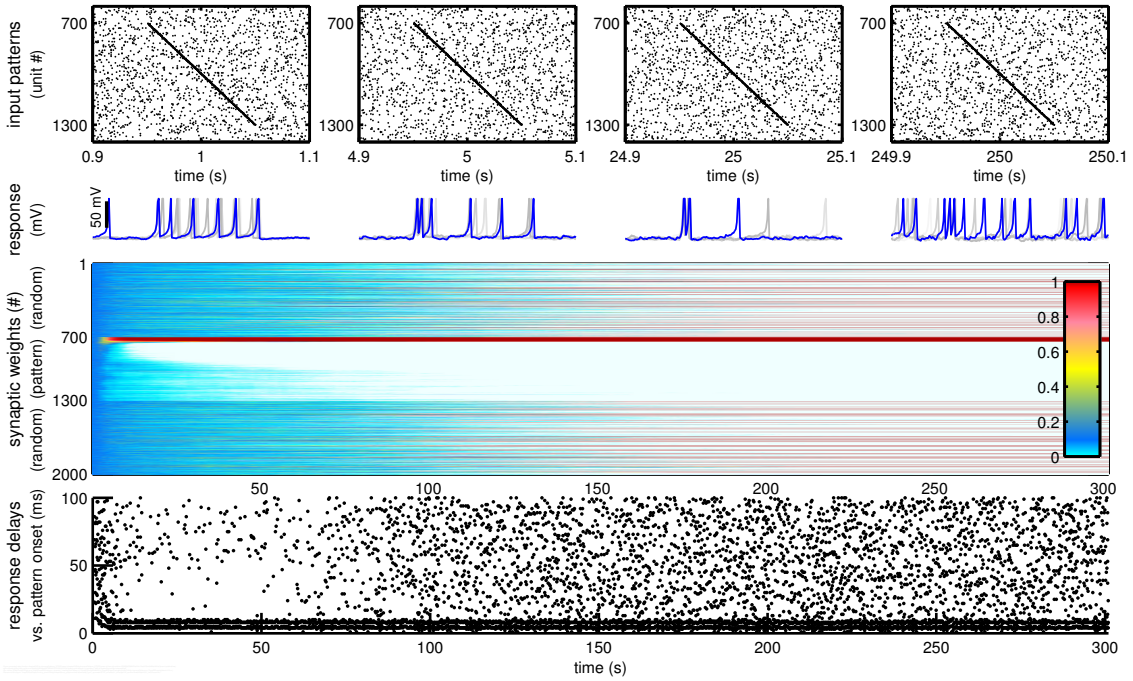


Figure 6.13: **Polychronous input patterns** with **attractor-free unbiased STDP** form a sparse receptive field within the set of pattern-presenting units, but also allow many randomly strong connections to form within the group of non-presenting input units (#1-#700 and #1301-#2000). The behaviour of non-presenting input units has also been seen in (Gütig et al., 2003; Rubin et al., 2001; Song et al., 2000), where exclusively non-polychronous inputs were used. Note also how due to the exponential shape of STDP rules the synaptic weights of very late-firing pattern-presenting input units (900-1300) are reduced more slowly than those of input units that fire just shortly after a postsynaptic spike (units 750-900). However, the weights of all pattern-presenting input units that fire later than postsynaptic spikes (750-1300) are reduced faster than the mean weight of non-presenting units. In principle, a neuron that uses unbiased STDP can even form multiple stable groups of strong inputs within the range of pattern-presenting inputs if it coincidentally tunes to two parts of the pattern at the start. Such a conflict can then not be resolved because both sides of the STDP rule have equal magnitude in unbiased STDP (example figure not shown). See Figure 6.12 for a description of rows and Appendix A.3.5 for simulation settings.

postsynaptic neuron (Figure 6.13, *response delays*). Using unbiased STDP hence has a detrimental effect on the formation of good receptive fields for robust pattern detection.

The response delay of a neuron that learns polychronous patterns through negatively biased STDP is much more precise than if unbiased STDP is used.

6.2.2 Non-polychronous patterns fail to harness the power of STDP

I now compare the use of polychronous patterns to some other families of (correlated) inputs while again using unbiased and biased STDP rules with attractor-less weight-dependent scaling.

The motivation for this is to examine if only polychronous patterns can reliably allow a single postsynaptic STDP neuron to form a predictable sparse receptive field without additional external influences, or if this may also be possible when other pattern families are used. The reader should keep in mind here that only one single postsynaptic neuron is used, and therefore no mutual inhibition between multiple postsynaptic neurons is possible. This is different from traditional Hebbian plasticity, where the mutual information between multiple postsynaptic neurons can be minimised to help form a sparse code (Bell and Sejnowski, 1995). The aim here is to uncover categorical differences between STDP and

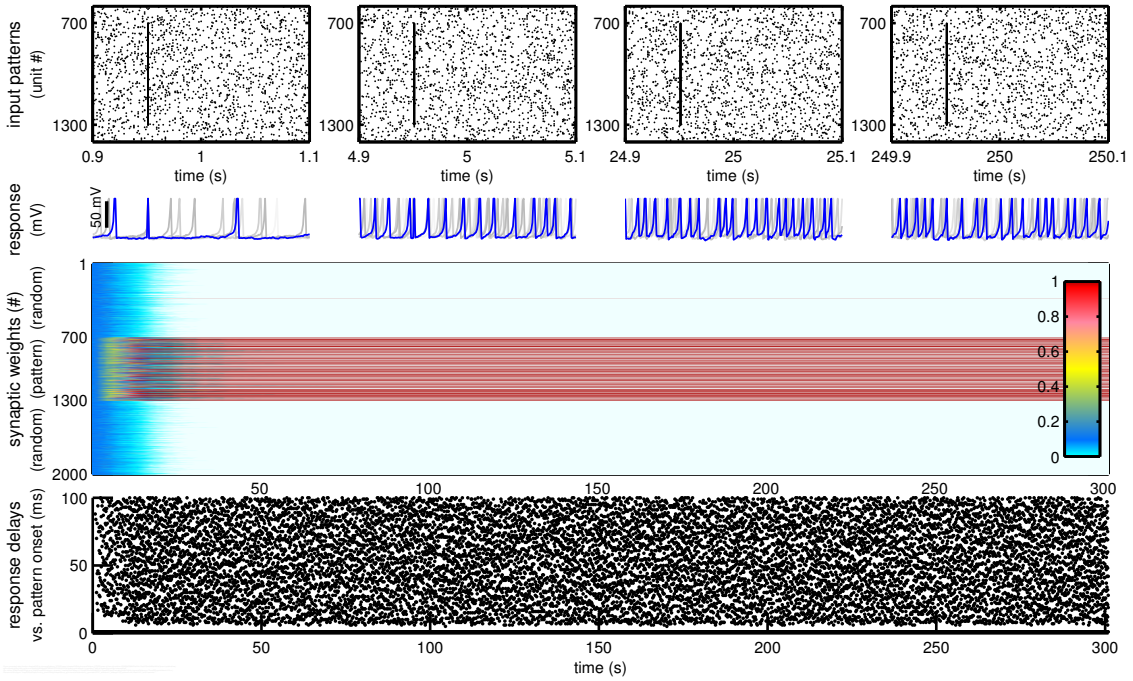


Figure 6.14: **Synchronous input patterns** with attractor-free **unbiased STDP** form more strong connections to pattern-presenting input units than they do to non-presenting units. However, strong connections are formed to both groups of inputs. This unguided formation of strongly bimodal weight distributions was seen as a disadvantage of attractor-less STDP by (Gütig et al., 2003; Rubin et al., 2001) and others, leading to the preference for attractor-based STDP by many (Chapter 5).

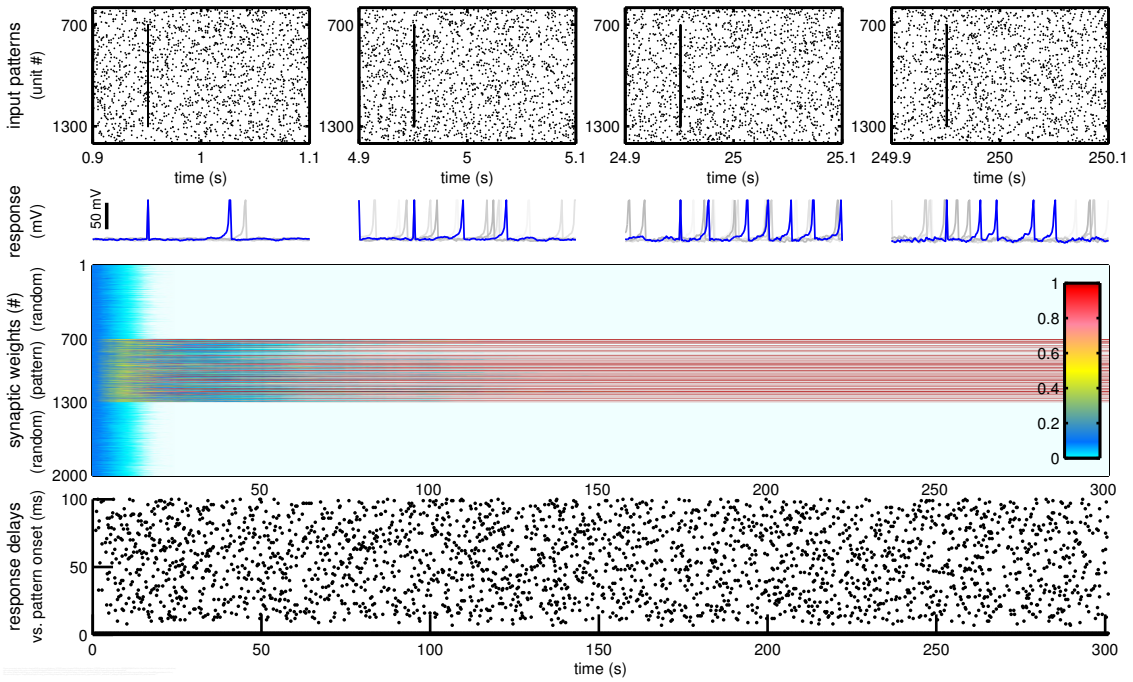


Figure 6.15: **Synchronous input patterns** with attractor-free **biased STDP** form a large number of strong connections to pattern-presenting input units, allowing the postsynaptic neuron to respond with minimal latency. However, the set of strong synapses is not minimised as for polychronous patterns and, instead, the identity of which connections become strong is random within the pattern-presenting group of inputs. Inter-pattern noise then produces additional random responses. Without inter-pattern noise, all connections to pattern-presenting input units would become strong. No connections to non-presenting inputs (units 1-700 and 1301-2000) become strong due to the use of negatively biased STDP.

traditional Hebbial learning. The manifestation of these differences, however, depends on which type of inputs STDP is treated to. The simulations displayed in Figures 6.16 - 6.21 use the exact same parameter settings as in Figures 6.13 and 6.12, except for different types of input pattern family. The effects of non-polychronous arrangement of inputs versus polychronous inputs will become highly visible.

Synchronous Inputs

Figures 6.14 (unbiased STDP) and 6.15 (biased STDP) show the development and effects of synaptic weights when synchronous inputs are used. In both figures, the single postsynaptic neuron quickly begins to form many strong connections to the pattern-presenting input units (#701-#1300) while strong connections to other inputs are rare. As the synchronous firing of all pattern-presenting input units produces a fast peak in the instantaneous firing rate of the input population, the postsynaptic neuron is likely to respond to this causally, so the dominance of pattern-presenting input units in the set of weights is to be expected. However, not all connections to pattern-presenting input units become strong, and not all connections to non-presenting input units become weak (at least for unbiased STDP). A slight difference between unbiased and biased STDP can be seen in the number of strong connections that are formed with pattern-presenting input units. Unbiased STDP (Figure 6.14) forms more strong connections while negatively biased STDP (Figure 6.15) forms slightly less. Similarly, the number of strong connections within the group of non-presenting inputs is also reduced for biased STDP, in agreement with Section 6.2.1 and Chapter 5.

The selection of which connections to pattern-presenting units become strong appears random, as no inner structure that could control this has been defined in the simulations.

In summary of Figures 6.14 and 6.15, we can say that synchronous input patterns form a less sparse distribution of synaptic weights, and that the identity of which synaptic connections win and become strong is only randomly selected.

Correlated Inputs

Within the present example, correlation between input spike trains is introduced by changing the firing rate of 600 poisson-distributed input units in unison. At pattern onset, the firing rate of these units is increased for a limited time span, and reduced again after a fixed number of milliseconds (here 50 ms). The resulting pattern can be seen in the first row of Figures 6.16 (unbiased STDP) and 6.17 (biased STDP).

Again, unbiased STDP causes a random selection of synaptic weights to become strong, with more strong connections forming with input units #701-#1300 than with background units that do not change their underlying firing rate. The reason for a higher tendency of pattern-presenting units to form strong connections again lies in the higher probability of postsynaptic spikes happening shortly after a positive jump in input firing rate, just as for synchronous input patterns. The two groups of input units (pattern-presenting vs. background) here show a less clear distinction than for synchronous inputs, though.

Biased STDP here causes a decrease of both background synapses and pattern presenting synapses, but the synapses to pattern presenting input units decrease slower here. In fact, whether we see a decrease of weights to pattern presenting units or an increase depends on the interplay of two factors. As first factor, the sudden increase of firing rate at the start of pattern presentation tends to evoke a postsynaptic response and thereby would increase synaptic weights to the pattern-presenting group. This is comparable to what happens for synchronous inputs, as the sudden increase in presynaptic population

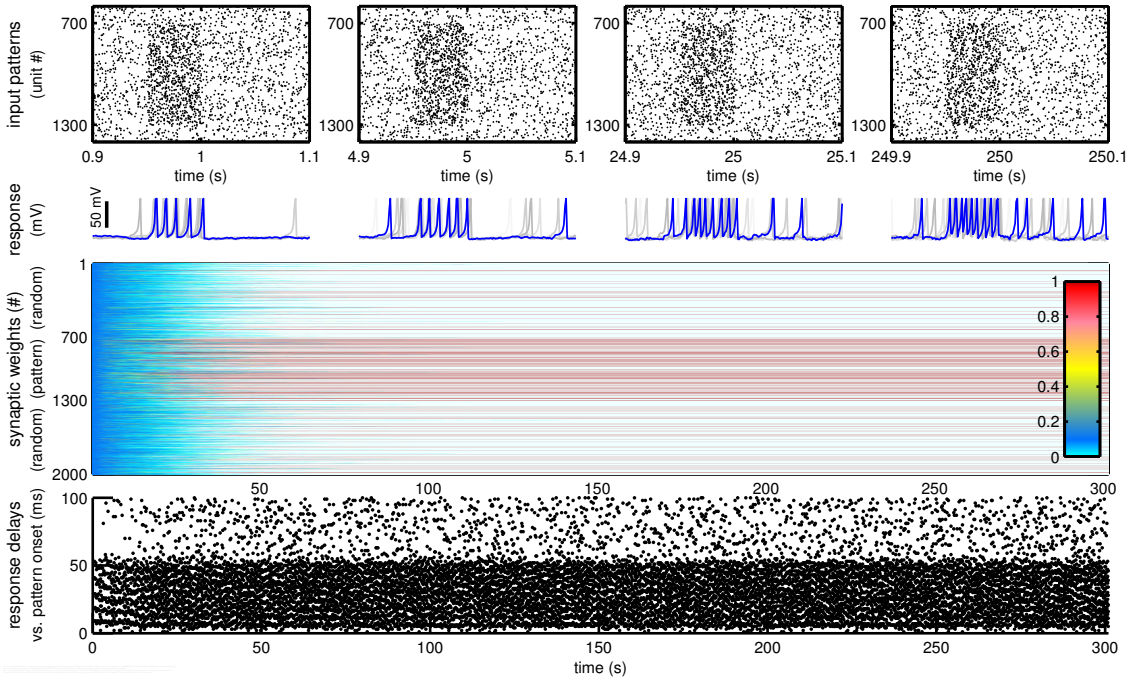


Figure 6.16: **Correlated input patterns** with attractor-free **unbiased STDP** form many strong input connections to correlated inputs (units 701-1300) but also form some strong connections to uncorrelated inputs (units 1-700 and 1301-2000). The identity of strong connections is random as in (Gütig et al., 2003; Rubin et al., 2001). The slightly higher ratio of strong connections in the pattern-presenting group is merely due to pattern duration, as patterns are generated as an intermediate between Figures 6.14 and 6.18 See Figure 6.12 for a description of rows and Appendix A.3.5 for simulation settings.

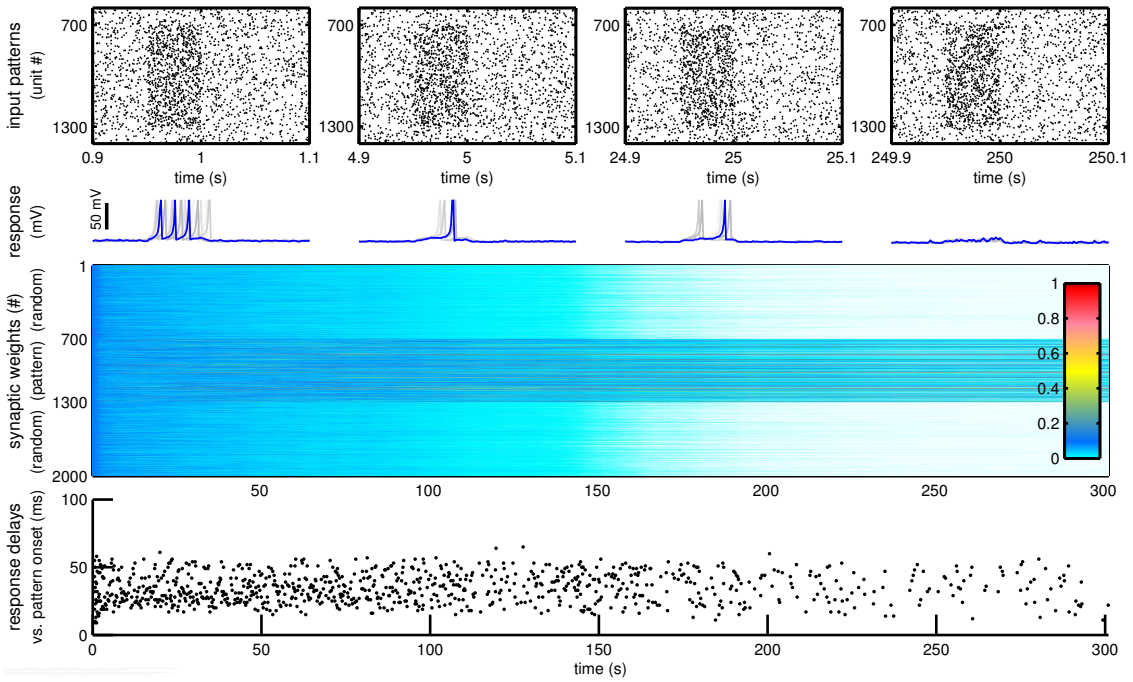


Figure 6.17: **Correlated input patterns** with attractor-free **biased STDP** form some stronger-than-average connections to pattern-presenting input units and reduce all other connections to non-presenting inputs. This figure shows an intermediate case between those of Figures 6.15 and 6.19: The input patterns consist of a brief period (50ms) of higher random input activity, after which activity of "pattern-presenting" input units falls back to normal (10 Hz). The correlation here comes from the condition that all pattern-presenting input units increase and decrease their firing rate together.

firing rate facilitates a postsynaptic response. The second factor, however, is the spread of increased firing rate over a longer time span than for synchronous inputs. The higher input firing rate during pattern presentation does not help synaptic growth because the time at which each of the pattern-presenting units fires is sometimes before and sometimes after the postsynaptic response to the whole group (see diagram in Figure 6.11). As the sign of spike timing differences at any given synapse is random, the higher presynaptic rate merely produces a faster negatively biased random walk. This is further explored in Figures 6.18 and 6.19.

The tendency for potentiation vs. depression of synapses to the pattern-presenting group can be controlled by changing these two factors. In the current example (Figure 6.17), the relationship between pattern onset effect and pattern duration lead to a slow decrease of weights. This also slowly decreases the number of responses of the postsynaptic neuron.

Higher-Rate Inputs

In Figures 6.18 and 6.19, I further show the effect of different firing rates on synaptic drift for unbiased and biased STDP, respectively. As the only difference between the "pattern-presenting" group of input units (units #701-#1300) and the background group (units #1-#700,#1301-#2000) is now just a difference in constant target rate, we no longer have a pattern onset time that could be used to measure postsynaptic response delay. Therefore, in Figures 6.18, 6.19, 6.20 and 6.21, the response delay subplot in each figure's lowest row is replaced by simply the firing rate of the postsynaptic neuron.

The second factor controlling synaptic drift direction that was mentioned in *Correlated Inputs* is here the only factor affecting synaptic drift direction. No synchronous or correlated events happen here, and any concurrent occurrence of presynaptic input spikes is by construction purely random. The increased frequency of presynaptic spike arrivals (input units #701-#1300) produces more frequent spike timing dependent weight update steps, which leads to more STDP updates within a given timespan.

For unbiased STDP, this leads to a faster differentiation of the weights into randomly selected strong synapses. In agreement with (Gütig et al., 2003; Rubin et al., 2001), the higher input rate also produces a smaller number of strong connections in that input group (units #701-#1300). As the simulation progresses, the group of background inputs (input units #1-#700,#1301-#2000) catches up in synaptic differentiation, but settles at a higher ratio of strong synapses (also in agreement with Gütig et al., 2003; Rubin et al., 2001).

For biased STDP, the depressing synaptic drift is also stronger for synapses connecting input units #701-#1300 than for background units. As the attractor-less but weight-dependent STDP rule used here slows synaptic step sizes as they approach the minimum weight ($w \to 0$), the difference in synaptic drift intensity remains visible.

Inputs driven by an Ornstein-Uhlenbeck process

A possibly more realistic (but also less controlled) way of implementing correlation between input spike trains is to use an Ornstein-Uhlenbeck (O-U) process (Uhlenbeck and Ornstein, 1930) to control a dynamic target spike rate for a subset of input units. The variability and mean reversion speed of an O-U process can be adjusted to either produce frequent changes in the target spike rate or instead allow more lengthy phases of just slowly changing target spike rates, while generally reversing back to some preset goal. Figures 6.20 and 6.21 show

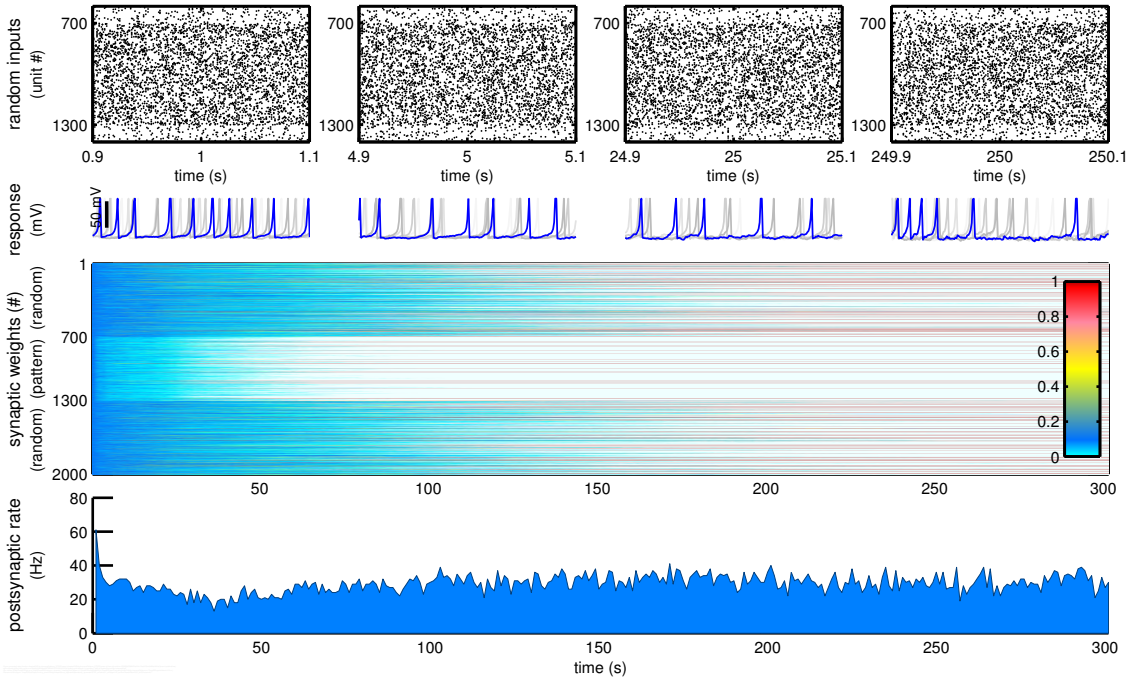


Figure 6.18: **Higher-rate random inputs** with attractor-free **unbiased STDP** randomly form strong connections with some of the input units. The identity of which connections become strong does not reflect any meaningful information as all inputs fire random poison-distributed spikes. Differentiation happens slightly faster for higher-rate inputs, as more spike pairings then happen within a given timespan. See Figures 6.12 and 6.19 for a description of rows and Appendix A.3.5 for simulation settings.

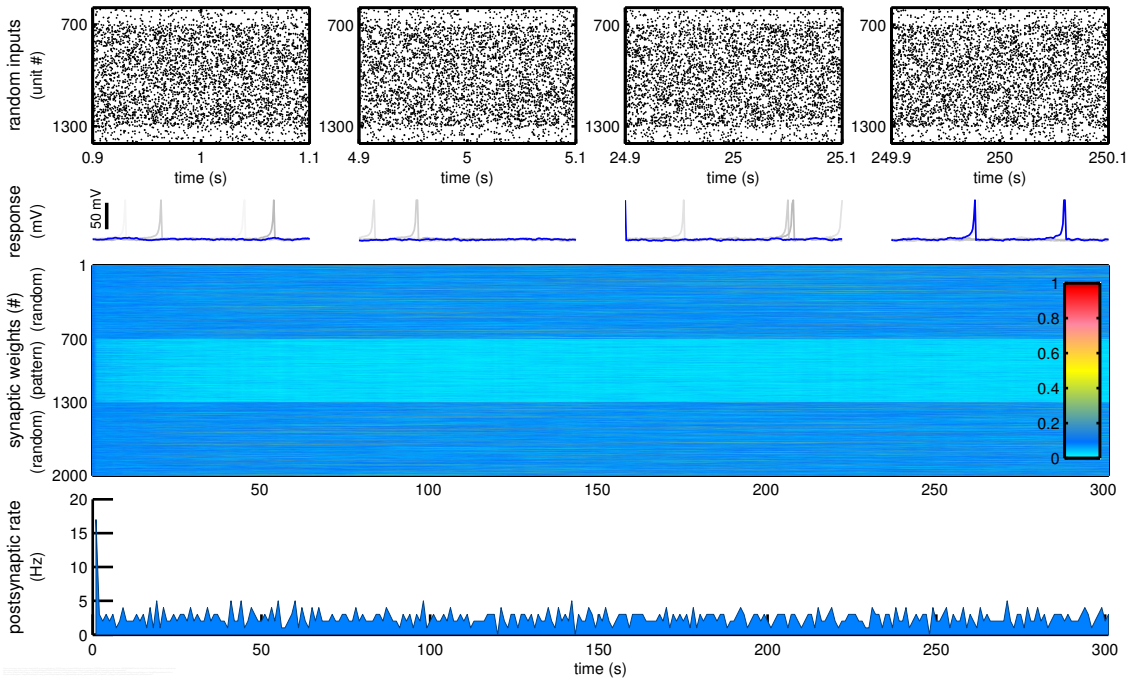


Figure 6.19: **Higher-rate random inputs** with attractor-free **biased STDP** decrease synaptic connections faster than normal-rate inputs due to accelerated depressing drift when spike pairings happen more often. The synaptic weights are here kept from reaching zero through a growth parameter that activates when the neuron has no strong connections (see Section 5.3.1), as the neuron would otherwise quickly become quiet. Both groups of input units now are uncorrelated, so we can no more speak of presenting "patterns" to the postsynaptic neuron. The last row therefore only shows the postsynaptic firing rate instead of a response delay plot. See Figure 6.12 for a description of rows 1-3.

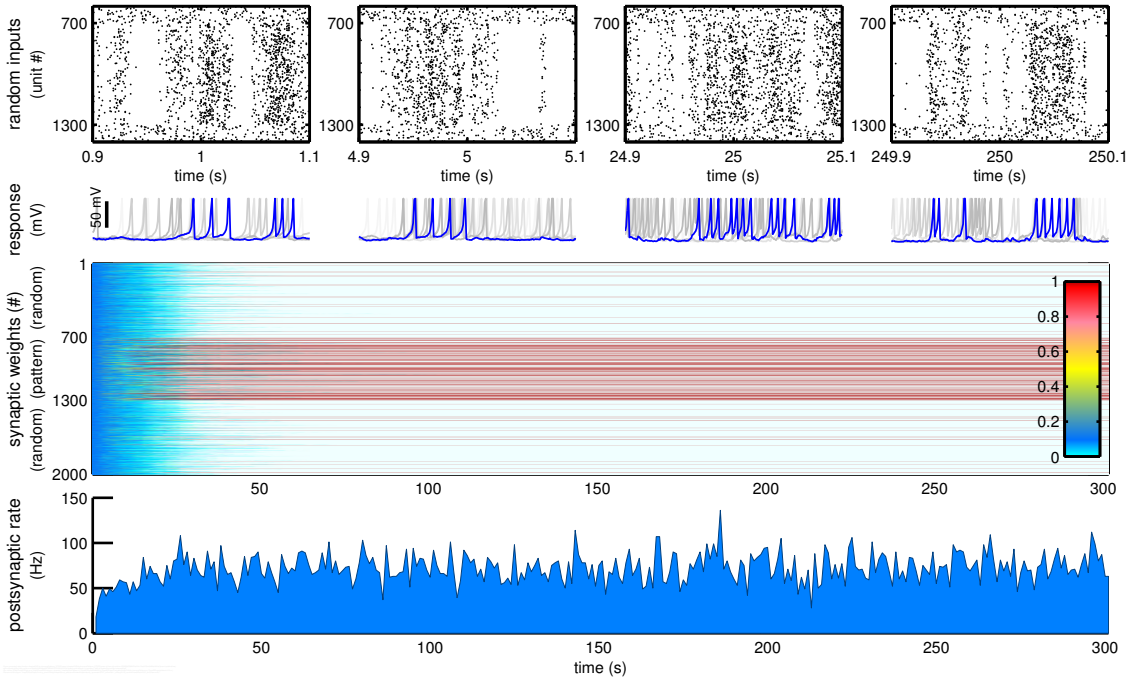


Figure 6.20: **Random inputs driven by a single Ornstein-Uhlenbeck process** with attractor-free **unbiased STDP** allow more strong connections to form with correlated inputs (units 701-1300) than with uncorrelated inputs. However, not all correlated inputs become strong and not all uncorrelated inputs become weak. The identity of strong connections remains random, placing this figure (at given O-U settings) between Figures 6.14 and 6.16. See Figures 6.12 and 6.19 for a description of rows.

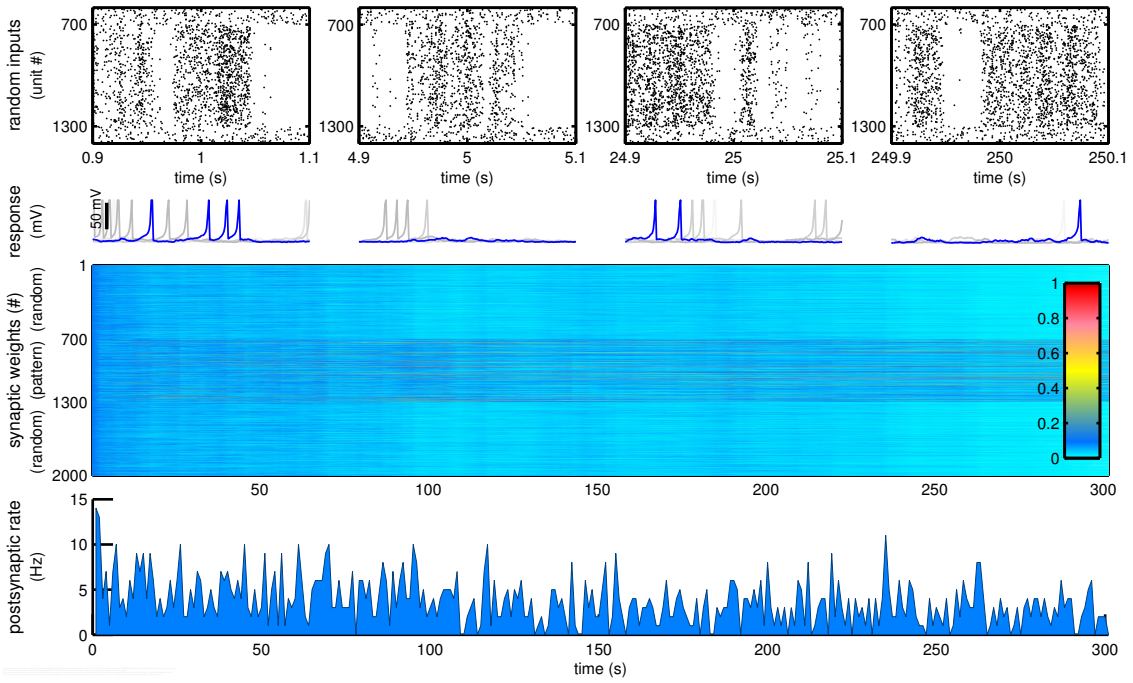


Figure 6.21: **Random inputs driven by a single Ornstein-Uhlenbeck process** with attractor-free **biased STDP** show no clear winners yet after 5 minutes of simulation. The O-U process introduces correlation between inputs (units 701-1300), but can also lead to periods of increased input firing rate. The formation of strong synaptic connections therefore depends on the parameters of the O-U process and here puts this resulting figure at an intermediate between Figures 6.17 and 6.19. The postsynaptic neuron is again kept active through a growth parameter as in Figure 6.19 (Section 5.3.1). See Figures 6.12 and 6.19 for a description of rows and Appendix A.3.5 for simulation settings.

the effect of an O-U controlled, dynamically changing target spike rate for input units #701-#1300.

As would be expected due to the described similarities, the development of synaptic weights for unbiased STDP (Figure 6.20) looks very similar to that shown in Figure 6.16. There is a higher relative number of strong synaptic connections for the group of correlated inputs (units #701-#1300) and a smaller ratio of strong connections for the background of uncorrelated inputs (#1-#700,#1301-#2000). The exact relation between strong and weak connections depends on the chosen O-U parameters, and shall be of no further concern here (but see Appendix A.3.5). What should be noted, is that again the identity of which units become strong is *randomly* selected, as by construction the simulation setup does not contain any structure that causes a particular arrangement of strong synapses.

Biased STDP also behaves similar to Figure 6.17, where background units have their synaptic weights decreased slightly faster than the O-U-correlated group. As mentioned above, the decision whether the correlated group or the background group decreases faster results from an interplay between the degree of correlation and the firing rate during more constant phases of presynaptic activity. If correlation were reduced while firing rate were increased for input units #701-#1300, the speed of synaptic decay of this group would be faster than that of the background group and become more comparable to Figure 6.19.

In summary, we can say that a highly selective receptive field that represents actual inputs can only be formed without an external teaching signal when polychronous input patterns are combined with a negatively biased STDP rule that is anti-symmetric and has no strong weight-dependent attractor.

Otherwise, a random subset of synapses may become strong that does not represent anything but random fluctuations in the input data. While the resulting weight distribution may possibly be called sparse because it is often binary with only few strong connections, the random permutations of strong synaptic connections do not carry any specific information that could be used to represent different messages of information.

If STDP is not combined with polychronous input patterns, a given group of presynaptic input units can therefore transmit only a single boolean or, when encoded in firing rate, one scalar (one-dimensional) value. When polychronous input patterns are used, the (continuous) transmitted value need not depend on firing rate (see Section 6.3), while firing rate could additionally be used for additional information throughput.

6.2.3 Attractor-based STDP hinders learning of polychronous patterns

As seen in Chapter 5, a strong stable fixed point attractor in "multiplicative" STDP rules makes it difficult for synaptic weights to escape the attraction even when they are constantly being increased by repetitive pre-before-post or depressed through repetitive post-before-pre spike pairings. While attractor-based STDP may allow the formation of stable receptive fields if the attractor is either very weak (Section 5.2.2) and/or very close to $w = 0$ (Section 5.2.4), these adjustments to attractor-based STDP only work because they make it more similar to attractor-free STDP.

Figures 6.22 and 6.23 show an example of the attractor-based rule of van Rossum et al. (2000) (see also Equation 5.5, p. 44) when postsynaptic activity is no more fixed to regular firing as it was in Figures 5.6 (p. 49) and 5.7 (p. 54), but depends solely on presynaptic inputs.

Figure 6.22 (unbiased STDP with attractor-based scaling) shows how the strong fixed point attractor keeps most weights close to central values ($w \approx 0.5$). Of the pattern-presenting units, only those synapses that connect to units near #700 become slightly

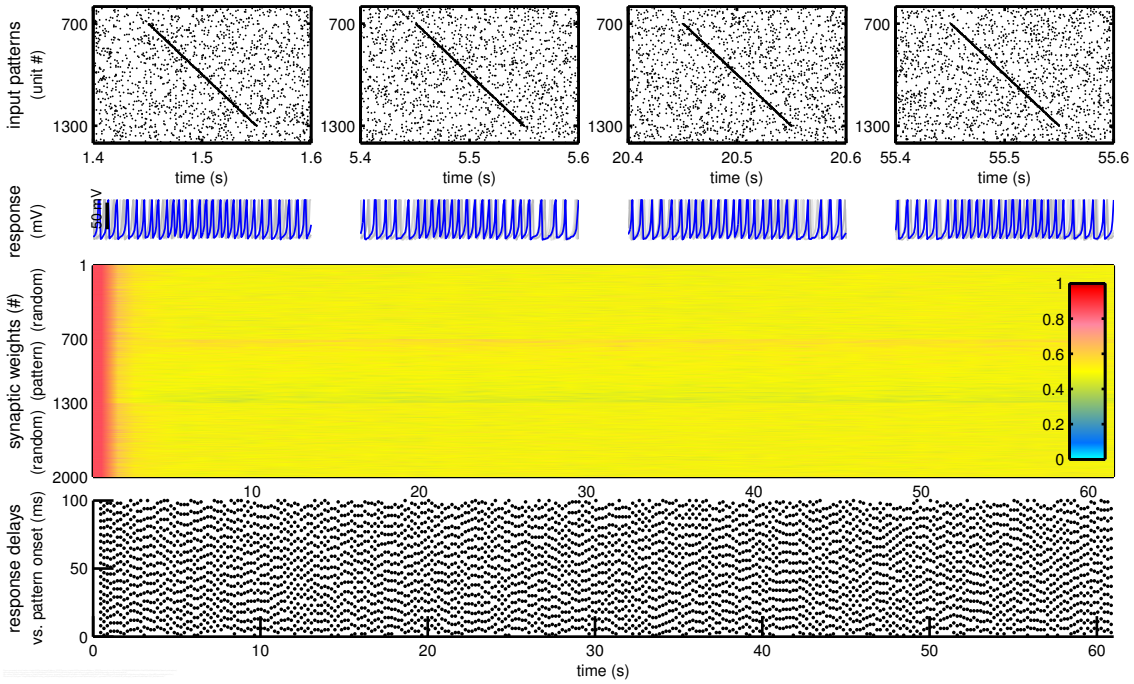


Figure 6.22: **Polychronous inputs** with **attractor-based** unbiased STDP as in Van Rossum et al. (2000) (Equation 5.5, p. 44). With no directed synaptic drift, the weight-based attractor keeps all synaptic weights near $w = 0.5$ here. STDP produces slightly stronger weights near the start of the polychronous pattern and slightly weaker weights near the end. But this is a minor effect compared to that of the attractor, and the postsynaptic neuron just fires constantly, with a frequency that depends only on input rate. See Figure 6.12 for a description of rows and Appendix A.3.5 for simulation settings.

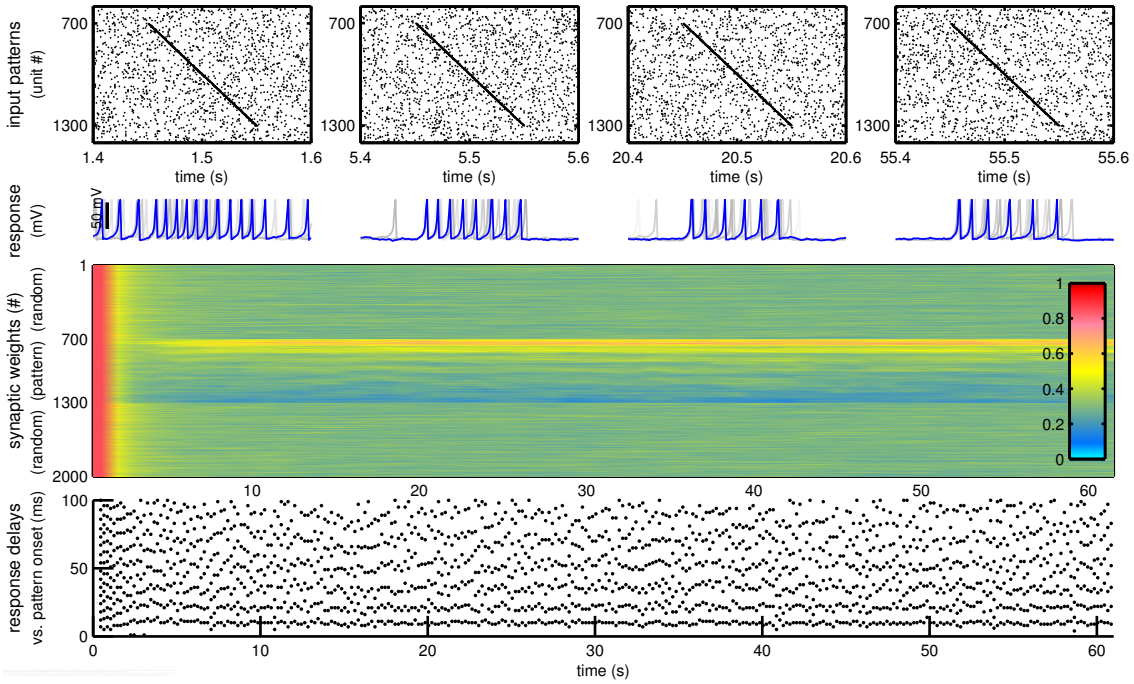


Figure 6.23: **Polychronous inputs** with **attractor-based** biased STDP as in Van Rossum et al. (2000) (Equation 5.5, p. 44). The postsynaptic neuron is unable to form a selective receptive field that would allow it to reliably respond only to the start of the polychronous pattern. Instead, it reacts to the increased input rate due to the inclusion of polychronous spikes. STDP does form a slight preference for the early part of the pattern, but this is insufficient for yielding robust output responses. See also Chapter 5. See Figure 6.12 for a description of rows and Appendix A.3.5 for simulation settings.

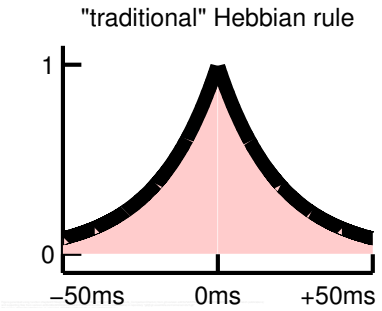


Figure 6.24: Traditional Hebbian plasticity rule. While Hebbian learning is often defined on the level of firing rates (see Section 2.3.1), it can easily be emulated in a spike timing context by making both sides of an STDP rule positive and mirror-symmetric ($A_+ = -A_- = 1$ and $\tau_+ = \tau_-$ in Equation 5.1). As all plasticity is potentiating, some additional form of "homeostatic" change to synaptic weights is necessary to keep synaptic weights from increasing to infinity. I therefore add a constant synaptic decay at each time step in Figure 6.25 (see also Appendix A.3.5).

stronger than $w = 0.5$, while those that are near #1300 become slightly weaker than the value that all weights of uncorrelated input are pulled towards ($w = 0.5$). However, this slight deviation from central values is not sufficient to allow the pattern-presenting input units to take control over postsynaptic firing, which continues to be controlled by a broad excitation from all input units together.

Figure 6.23 (biased STDP with attractor-based scaling) shows a clearer distinction between those input units at the start of the polychronous pattern (units ca. #701 - #801), which become stronger than average, and input units that present the end of the pattern (units ca. #1200 - #1300), which become weaker than average. The overall negative drift of the biased STDP rule also moves the fixed point attractor slightly further towards lower weights $w \approx 0.3$ (compare Figure 5.7 vs. Figure 5.6, column C), just enough to allow the neuron to become quiet when no pattern is being presented to it (Figure 6.23, 2nd row). Apart from the initial response to pattern onset (Figure 6.23, 4th row), postsynaptic spike times during pattern presentation appear increasingly irregular as pattern presentation progresses. The response timing seems to depend only on the change of firing rate of the input layer during pattern presentation (see also Section 7.1 for generating patterns without affecting the firing rate). While very close tuning of network parameters may help in improving detection of patterns (van Rossum et al., 2012), the unstable nature of this required tuning disqualifies attractor-based STDP with strong central attractors from being useful for practical applications to pattern detection.

6.2.4 Traditional Hebbian plasticity fails on polychronous patterns

We have now seen that only polychronous input patterns allow anti-symmetric STDP to form a robust and predictable sparse receptive field (Sections 6.2.1 and 6.2.2), and that additionally any weight-dependent scaling of STDP should have a nonexistent (or at least only weak) fixed point attractor for the formation of sparse receptive fields to be successful (Section 6.2.3). I now examine whether the anti-symmetric shape of standard STDP rules is indeed necessary for successful formation of sparse receptive fields of a single postsynaptic neuron.

An alternative mirror-symmetric version of STDP is shown in Figure 6.24, where the change of synaptic connection strength does *not* depend on causal order of pre- and postsynaptic spikes. In continuation of this idea, any weight-dependent scaling of synaptic updates is attractor-free, so does not change the ratio between potentiation and depres-

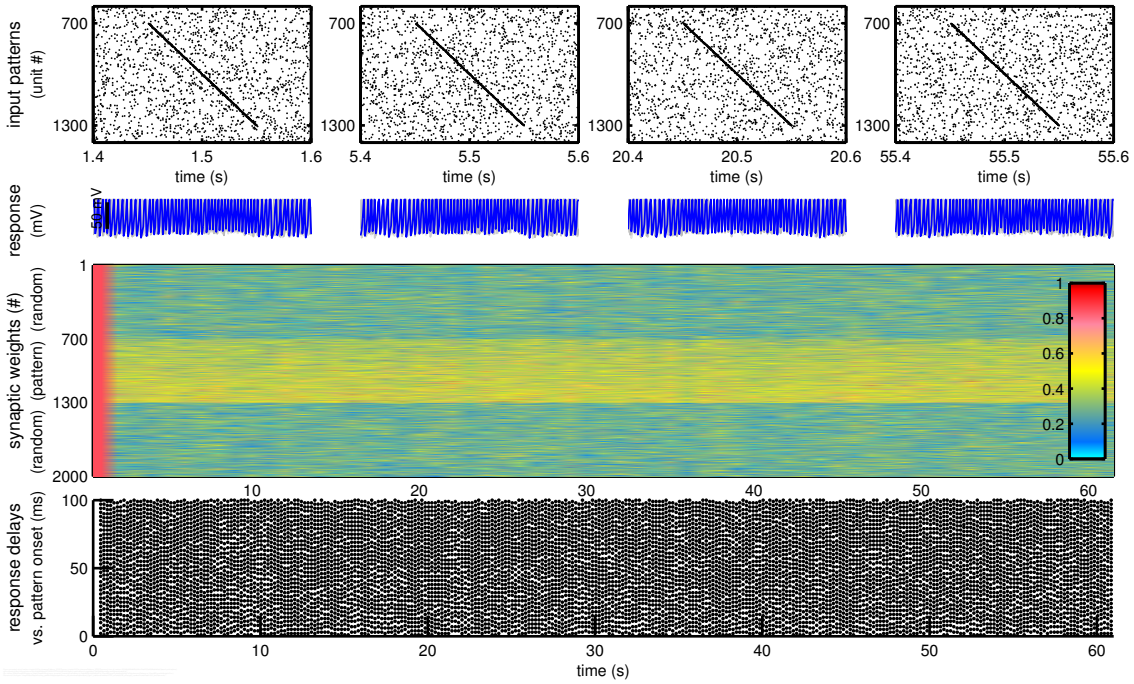


Figure 6.25: Traditional **Hebbian plasticity** is not able to take advantage of **polychronous inputs**. Every spike pairing adds to the synaptic weight, so the STDP rule is inherently unstable. Close tuning of a constant synaptic decay term can keep the weights from reaching the maximum ($w_{max} = 1$) or decaying to the minimum ($w_{min} = 0$), but this still produces a neuron that is very volatile and can hardly detect polychronous pattern presentations, apart from possible co-occurring increases in input firing rate. In order to detect polychronous patterns, they would either need to be presented with far less background noise so that the neuron could detect the pattern's rate increase, or only a subset of inputs could be used for detection with the help of some external teaching signal. While the synaptic connections to the pattern-presenting input units do appear slightly stronger than connections to non-presenting background units, this is merely a transient effect due to the added input firing rate during pattern presentation. Soon after each pattern-presenting input unit fires its additional spike as part of pattern presentation, its weight decays back to the level of background units. The increased mean synaptic strength that can be observed for pattern-presenting input units in row 3 is a direct consequence of the time at which each pixel-column of row 3 was recorded: At the start of each second of simulation, the end of the previous polychronous pattern presentation is always exactly 50ms past, so synaptic decay has not yet fully equalised all synapses. This also explains why late-firing input units (50ms ago at the time of sampling) in the plot (row 3) are seen as having a slightly stronger synaptic weights than early-firing input units (150ms ago at the time of sampling). See Figure 6.12 for a description of rows and Appendix A.3.5 for simulation settings.

sion for any synaptic weight. This makes the rule presented in Figure 6.24 a spike timing dependent implementation of traditional Hebbian rules, where a synapse's connection strength increases whenever its two neurons repeatedly fire together (are correlated). As in rate-based Hebbian plasticity, the purely potentiating rule shown in Figure 6.24 requires some additional synaptic decay to be implemented to prevent all synapses from increasing to infinity (or the predefined upper bound $w = 1$). As mentioned in Section 2.3.1 (p. 16), this can be done through a variety of homeostatic mechanisms, including short-term synaptic scaling to obtain either a target output firing rate or a target sum of all weights. While synaptic scaling can be implemented with only minimal tuning of parameters, I avoid using synaptic scaling because it requires the assumption of existence of either a target firing rate or a target sum of all incoming synaptic connections to a postsynaptic neuron. Instead, I use a constant synaptic decay of all synapses on each time step to counteract the correlation-based increase of synaptic strengths, even though it does require finding a useful step size of decay (see methods in Appendix A.3.5).

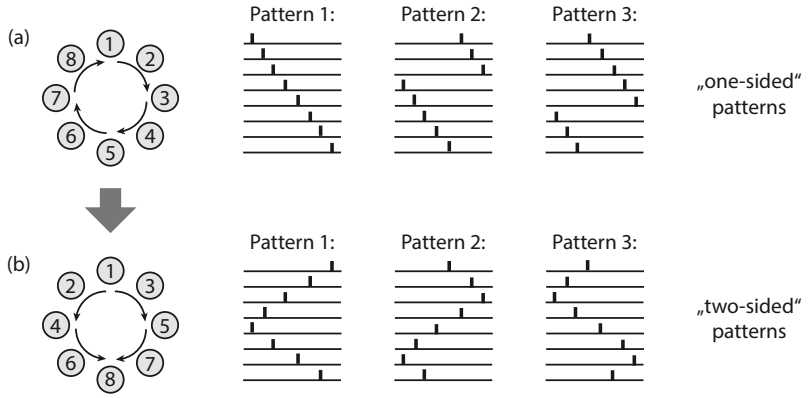


Figure 6.26: When multiple patterns are used in conjunction with STDP, two-sided polychronous patterns should be chosen instead of one-sided polychronous patterns. (a) One-sided patterns have been used until now for didactic purposes. They are formed by simply letting all neurons fire in an ordered fashion. (b) In order to use multiple patterns in a plastic network, we need an alternative way of generating patterns that allows for stable receptive fields to be formed. From now on, two-sided polychronous patterns will be used whenever pattern similarity should reflect stimulus similarity. This also resembles biological spike wave fronts more closely than one-sided patterns. See also Section 7.1.7.

Figure 6.25 shows the simulation of a single postsynaptic neuron that receives polychronous inputs just as before. The time constant of synaptic decay was chosen high enough to keep synapses from reaching the upper bound, but not too high as to avoid all weights becoming zero. The 2nd and 4th rows of Figure 6.25 show that the postsynaptic neuron remains constantly active for this setting. The synapses connecting the postsynaptic neuron to the pattern-presenting inputs (input units #700 - #1300) are slightly stronger than other synapses. On closer inspection, one can see that the synapses just below #1300 are also slightly stronger than those just above #700. Unfortunately, instead of representing some stable receptive field, this increased strength of synapses that transmit spikes towards the end of each pattern presentation is merely an artefact of the time at which the state of all weights is tracked. As the STDP rule is not able to form sparse receptive fields, the random background activity of all 2000 input units continues to strongly affect postsynaptic response times. The highly random spike times of the postsynaptic neuron in turn produce highly random spike pairings for STDP, which has a negative drift. So what we see is in fact a quickly decaying set of weights since the last pattern presentation that had temporally increased the firing rate of pattern-presenting input units. If the presentation of input patterns was halted, this memory of pattern presentation would decay in a matter of seconds.

This shows that without biologically questionable practices such as constant synaptic scaling or the introduction of a target firing rate, traditional Hebbian learning fails to learn sparse codes from polychronous patterns. Without these supplementary assumptions, only anti-symmetric STDP can form sparse receptive fields when polychronous input patterns are used.

6.3 Multiple Overlapping Polychronous Patterns

After having demonstrated how the repeated presentation of a single polychronous pattern can lead to a sparse or near-sparse distribution of incoming connections of a single neuron, let us now think about the general usefulness of this way of presenting information. Depending on the order of input spike arrival, the postsynaptic neuron begins to respond

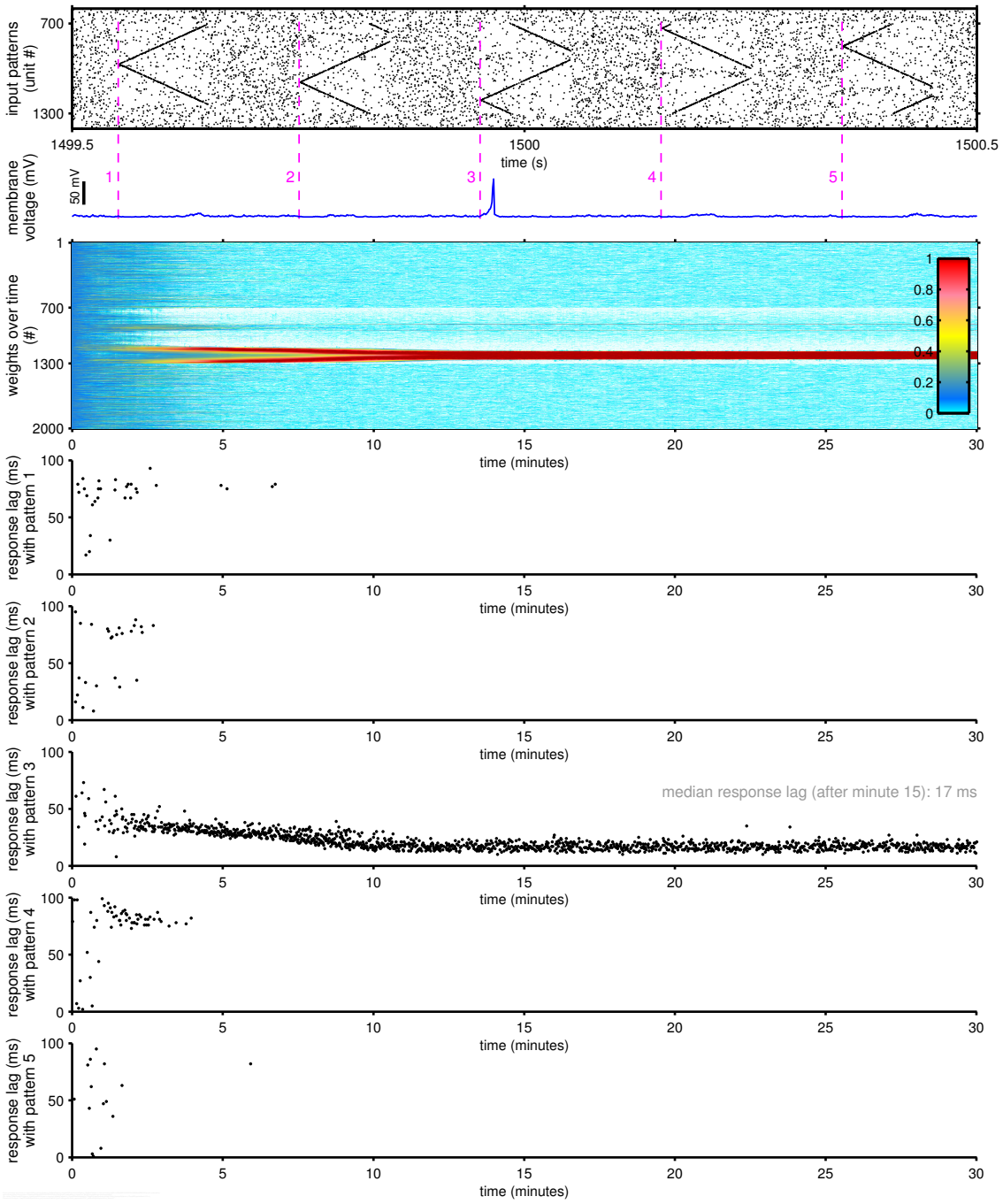


Figure 6.27: Multiple polychronous patterns presented by the same input units allow a single postsynaptic neuron to tune to one of those patterns and ignore the others, as long as two-sided patterns are used instead of one-sided. This allows the formation of sparse receptive fields *without the need for mutual inhibition between multiple postsynaptic neurons*. Simulation details are given in Appendix A.3.6. Compare also to Figures A.3 (p. 175) and A.4 (p. 176).

to only a minimal set of inputs while ignoring all others. If other input units repeatedly fired first, the postsynaptic neuron might tune to those, instead. Hence the same group of input units might be used for presenting a large number of polychronous patterns, where the difference between patterns is the order in which the involved input units tend to fire.

6.3.1 Forming Multiple Patterns

A first way of producing multiple polychronous input patterns may be to shift the spike times of all pattern-presenting input units in a circular manner. This first example can be seen in Figure 6.26a.

However, if we want a single group of input units to present multiple polychronous spike patterns to a postsynaptic neuron, we need to allow for stability of the resulting receptive fields. If the polychronous input patterns are too similar, the postsynaptic neuron may end up constantly retuning to every pattern that is presented, causing highly unstable receptive fields. When multiple patterns are generated through a circular shift from a one-sided polychronous pattern as in Figure 6.26a, the first three units to fire in the first pattern 1 also fire together in patterns 2 and 3. In a plastic network, this prevents the receptive field of a postsynaptic neuron from tuning to precisely one pattern (see supplementary Figure A.3, page 175, as a negative example). This fails because the group of earliest firing units that fire at the beginning of one pattern also fire in close temporal proximity within each other pattern. See Section 7.1.7 for an evaluation of pattern distances.

Instead, we need to find a way to prevent all early-firing units of one pattern from firing together at later stages of any other pattern. It would, however, be nice if we could retain similarity between patterns representing similar stimuli, as there might be cases in which this is necessary. The solution then is to arrange the spike order of the pattern-presenting units in a two-sided manner, as shown in Figure 6.26b. Again, Section 7.1.7 explains why this simple rearrangement allows the single postsynaptic neuron to succeed in forming stable receptive fields that are selective to exactly one of the presented patterns. The aim of the current section is instead to first introduce the reader to the idea of presenting multiple distinct polychronous patterns through the same set of input units.

6.3.2 Learning with Multiple Patterns

Figure 6.27 shows the successful case of using two-sided polychronous input patterns as inputs to a plastic neuron that uses STDP. In the beginning, the postsynaptic neuron responds randomly to all patterns, but then soon starts to respond primarily to pattern 3, while responding much less to patterns 1, 2, 4 and 5 and may eventually completely stop responding to those other patterns. The start of each pattern is marked by dashed vertical magenta lines in the first and second panel rows (*input patterns* and *membrane voltage*). The patterns compete with each other over control of the postsynaptic neuron. When the neuron begins to preferentially respond to the presentation of one of the patterns, any other units that may coincidentally be active due to random noise during the postsynaptic response will have their connections slowly decreased due to the negative drift of STDP. ***No mutual inhibition*** between multiple postsynaptic neurons is necessary for learning multiple polychronous patterns as long as the STDP rule has a negative drift and all input units occasionally fire random spikes. This also shows how important noise is within the neural system, as without noise there would be much less competition between polychronous patterns. The system may likely be called non-ergodic, as it randomly settles on one of many presented patterns and then forms a robust receptive field to detect the start of exactly this pattern. However, I leave a formal proof of this to future work. The choice of input patterns by multiple postsynaptic neurons will be further explored in Section 7.3, and the generation of multiple polychronous patterns for simulations will be handled in Section 7.1.1.

For completeness, a third variant of forming multiple polychronous patterns (see Sec-

tion 7.1.5) was also used for training a single postsynaptic neuron. The results can be seen in supplementary Figure A.4 (page 176).

6.4 Rephrasing the Question: From STDP to Neural Coding

We have now established that STDP can form highly selective receptive fields from polychronous inputs without the need for artificial scaling or mutual inhibition. The automatic sparsification of receptive fields comes for free when STDP is paired with polychronous input spike patterns.

In light of this seemingly perfect match between attractor-free STDP rules and spatiotemporally structured input patterns, a certain question arises. Has neuroscience been oversimplifying the expected inputs to STDP-capable neurons for too long by using merely correlated or synchronous inputs? Has this led us to miss powerful features of STDP that are only seen when polychronous patterns are presented? Should the mere existence of STDP in many brain areas then not point us towards the idea that all those areas use polychronous coding of neural information to some extent?

Also, the uses of the observed features of general STDP should be further explored. If STDP automatically leads to highly selective receptive fields for polychronous patterns under good conditions, could this help to form a sparse network as in (Olshausen and Field, 1996b) without the need for precise mutual inhibition and biologically implausible approaches that require the availability of some original stimulus for calculation of a reconstruction error (as in gradient descent / matching pursuit algorithms)? What could the benefits still be under bad conditions, when a perfectly sparse representation is not achievable due to insufficient or noisy data?

What are the implications of having a detector that makes a decision about a message before the complete message has even been received. How can we "be sure" that the remainder of the message did not also contain some important information? How does the transmitting side "know" when to stop transmitting a message or if more data is needed?

In the following I discuss some more questions and possible new directions of research that arise from using polychronous patterns together with STDP for neural computation.

6.4.1 Neural Signal Detection as a Decision Problem

The classical distinction between "coincidence detector" and "integrator" neurons (Section 2.2.2) appears artificial when polychronous inputs are used, as a minimum number of distinct coincident inputs need to be sustained over some time for the postsynaptic neuron to respond. As is commonly modelled in more abstract rate-based models, the decision whether to fire or not to fire can likely also be described as a decision process with two alternatives for a single neuron receiving polychronous inputs.

6.4.2 Comparison to Entropy Coding

A postsynaptic neuron that learns to respond to an incoming polychronous pattern will begin to respond earlier after the pattern has occurred (and likely caused it to fire) multiple times (Section 6.2.1). When the postsynaptic neuron is initially tuned to input units that fire late within pattern presentation, the response delay after some amount of training will (loosely) correlate with the number of pattern presentations. Patterns that occur often will be responded to with short delay, while patterns that occur only rarely will be responded to with longer delay after pattern onset. If we liken the number of presynaptic input spikes occurring before the postsynaptic response to the length of a message that

is being transmitted, then the postsynaptic neuron is using short messages for common pieces of information, and longer messages for rare events. This then bares some general similarity to entropy coding techniques used for compression and efficient transmission in computer science.

Of course the comparison with entropy coding is not a perfect match, however, as entropy coding is commonly described as a lossless transmission method while biological systems can hardly claim this. A late response to an incoming pattern may also have many reasons, such as noise in the external world (e.g. night or fog), internal noise (fluctuations in the activity of background inputs or membrane noise), or a simple absence of any strong connections for a given stimulus. Still, the inherent shortening of decision time as a neuron becomes more tuned to a given stimulus remains intriguing and deserves to be further explored in future work.

6.4.3 Matching Pursuit and Sparse Filters

In efficient coding as presented in Olshausen and Field (1996a) or Bell and Sejnowski (1995), a large (overcomplete) set of filters is made to represent each element of a training set through a weighted combination of a minimal number of filters that can still together reconstruct the training examples to a given precision. The two constraints for using a filter here are often (1) how well the filter helps in reconstructing the training example, and (2) how different it is from other filters. In non-biological implementations, this decomposition of sources into a set of filters with a minimal set of non-zero coefficients has proven to be highly useful in solving a number of blind source separation tasks (Lulham et al., 2011). But the algorithms used to decompose a set of training examples into a set of sparsely active filters are currently very slow in rate-based networks (Savin et al., 2010) or use shortcuts that are likely not available in biological neural networks (Zhao et al., 2012).

When polychronous patterns are used together with anti-symmetric STDP, the neurons (=filters) that best represent the earliest part of incoming patterns (stimuli) will respond first, and together transmit a fast and good representation of the encountered stimulus. This implies that the earliest responses also indicate the most important part of an incoming pattern, which may compete with the possible interpretation given in Section 6.4.2. However, the interpretation of this may also depend on the use case and more research into performing blind source separation with polychronous input patterns and STDP is necessary in future work.

6.4.4 A Continuum of Pattern Families

Synchronous patterns can be seen as a subset of polychronous patterns (Figure 6.28), which themselves are a subset of patterns created through random permutations of spike order (Section 7.1.5), which again is a subset of patterns created through frozen random noise (used in Masquelier et al., 2008). However, a restriction of pattern set size appears useful for coping with dimensionality when exploring the functionality of STDP. But as has become clear in this chapter, the reduction from (seemingly) randomly occurring spikes in electrophysiological measurements to synchronous patterns or merely correlated spike trains is an oversimplification that causes the loss of most of the additional power that STDP has over traditional Hebbian plasticity. When reducing biological complexity to a computational model that uses STDP, new models should retain the possibility of using polychronous firing.

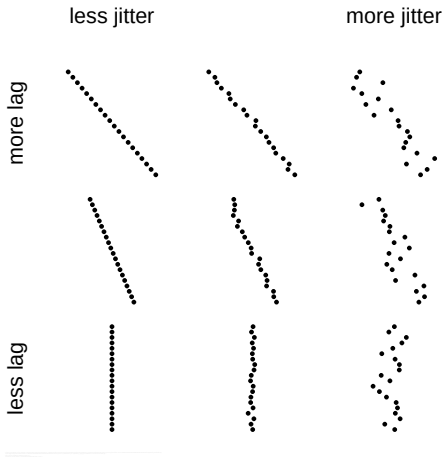


Figure 6.28: Synchronous spike patterns are merely a subset of all polychronous patterns. Similarly, zero-lag correlated inputs can be seen as the repeated presentation of jittered synchronous patterns. It seems highly questionable why one should artificially restrict the expected neural code to only (jittered) synchronous patterns. Especially as we have just seen the increased computational power of STDP when polychronous patterns are used instead of (jittered) synchronous input spike patterns. In Chapter 7 we also generate polychronous patterns with (slightly) more jitter.

6.5 Summary

In this chapter, we found that neural codes consisting of polychronous patterns are far more powerful than codes consisting of synchronous patterns or rate-based codes. This was found first in Section 6.1, where polychronous inputs allowed the response time of a non-plastic detector neuron to encode how well its receptive field matches the incoming pattern. It was then shown again for plastic neurons that used standard (anti-symmetric) STDP to form a highly selective receptive field. Neither traditional (symmetric) Hebbian plasticity nor weight-dependent STDP with strong attractors are able to do this. Also, merely rate-based inputs and inputs with highly fluctuating rates (but without polychronous patterns) failed to produce a predictable, robust, and selective receptive field (Vogt and Hofmann, 2015c).

Zero-lag correlation can be viewed as a jittered version of synchronous codes, while nonzero-lag correlation is comparable to the repeated presentation of a single polychronous pattern. However, polychronous codes may also use a given set of inputs to present a large number of polychronous patterns, as will be further discussed in the next chapter.

Chapter 7

Exploring Robust Network Parameters for STDP

In the previous two chapters, I first described how largely attractor-free STDP rules outperform attractor-based STDP rules in their sensitivity to spike timing differences (Chapter 5). I then showed how only polychronous input patterns uncover the full potential of STDP in forming highly selective receptive fields that allow fast and robust detection of recurring stimuli (Chapter 6).

In this chapter, I will now present some practical issues that arise when working with polychronous input patterns within computer simulations. This allows us to infer the type of spatiotemporally structured spike groups we should be looking for within electrophysiological multielectrode recordings when trying to decipher possible complex neural codes in the future.

Specifically, I first present a variety of easy-to-generate polychronous pattern types, and demonstrate their possible camouflaging within a stream of seemingly random inputs (Section 7.1). I then discuss general effects like network size and presentation duration (Section 7.2) before expanding the number of postsynaptic neurons from 1 to 100 in order to reveal differences in the distribution of receptive fields (Section 7.3). I then take a quick look at increasing levels of noise (Section 7.4) and discuss mechanisms for purposely biasing the learning outcome (Section 7.5).

7.1 Generating plausible polychronous input patterns

There are many ways in which spatiotemporally structured input patterns may be present within a recorded stream of spikes from multiple units, which may not appear directly visible to the naked eye. I present here some ideas I have had, together with the benefits and possible drawbacks of each.

7.1.1 Pattern Types

I will restrict my examples to four methods of pattern generation, retracing the thought process that leads to the development of well-camouflaged patterns that do not stand out in typical descriptions of electrophysiological recordings as they are used today. The random background activity is generated via a simple poisson process where the probability of a spike occurring within a time bin of 1ms is re-evaluated on each time step of the simulation. For small enough time bins and low enough firing rates, this simplified method of generating Poission spikes has been demonstrated to approach the full mathematical

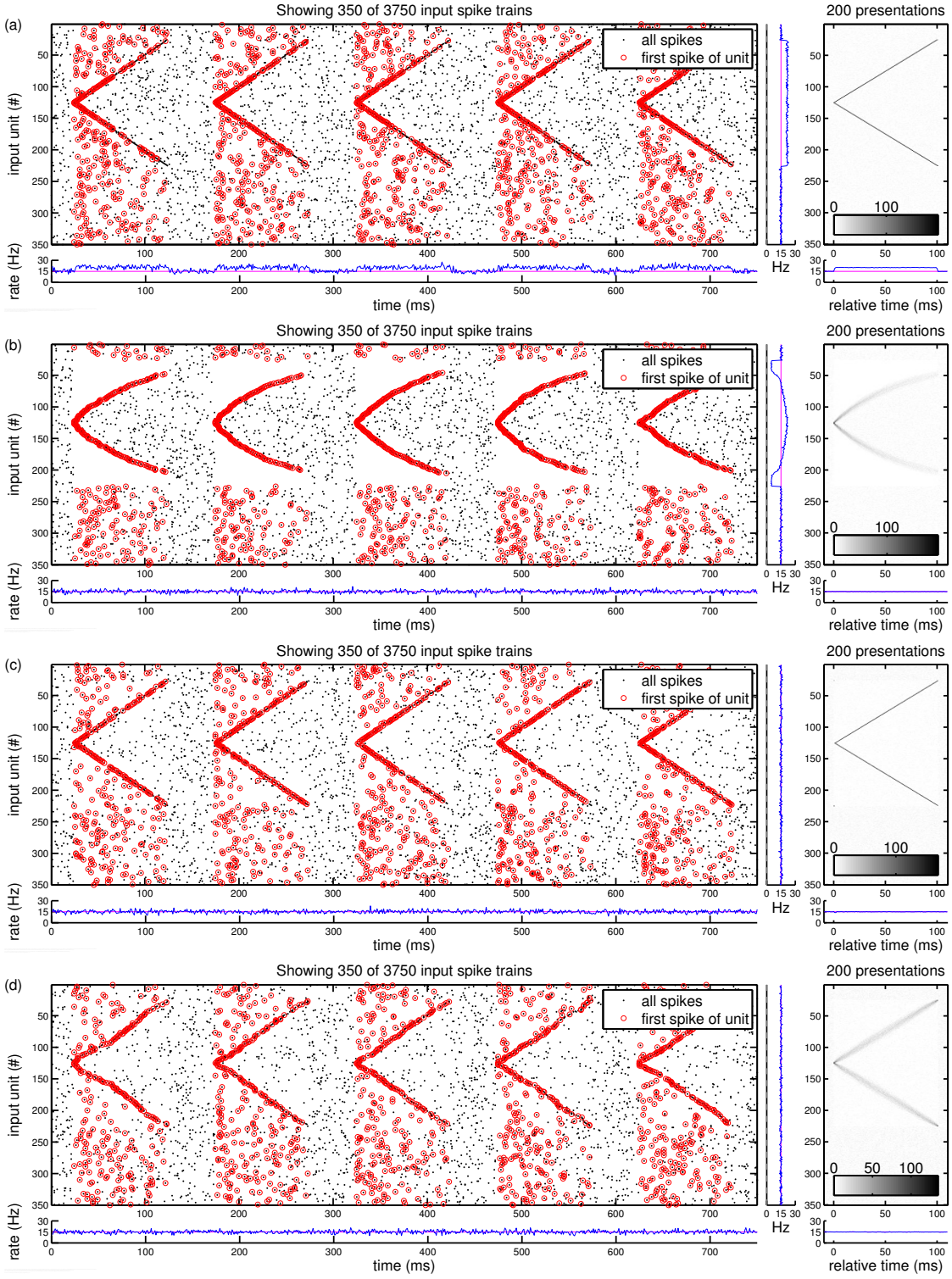


Figure 7.1: Pattern types A-D showing a **single pattern** each. (a)-(d) Each set of subplots shows one of pattern types A-D. Within each set of subplots, the largest (main) plot shows the freshly generated spike trains that may be presented to a postsynaptic neuron for detection or training. Below and to the right of each main plot are the instantaneous and unit-wise firing rates of the pattern-presenting population of units, respectively. Magenta lines represent target (background noise) firing rate while blue lines represent actual firing rate. On the far right we see an overlay of 200 pattern presentations, again together with the averaged instantaneous population rate below each overlay plot. The greyscale colour bar's maximum is set to the maximum number of times any unit fires within a given time bin during pattern presentation. This also shows that pattern generation type D produces less predictable spike times than the other three types.

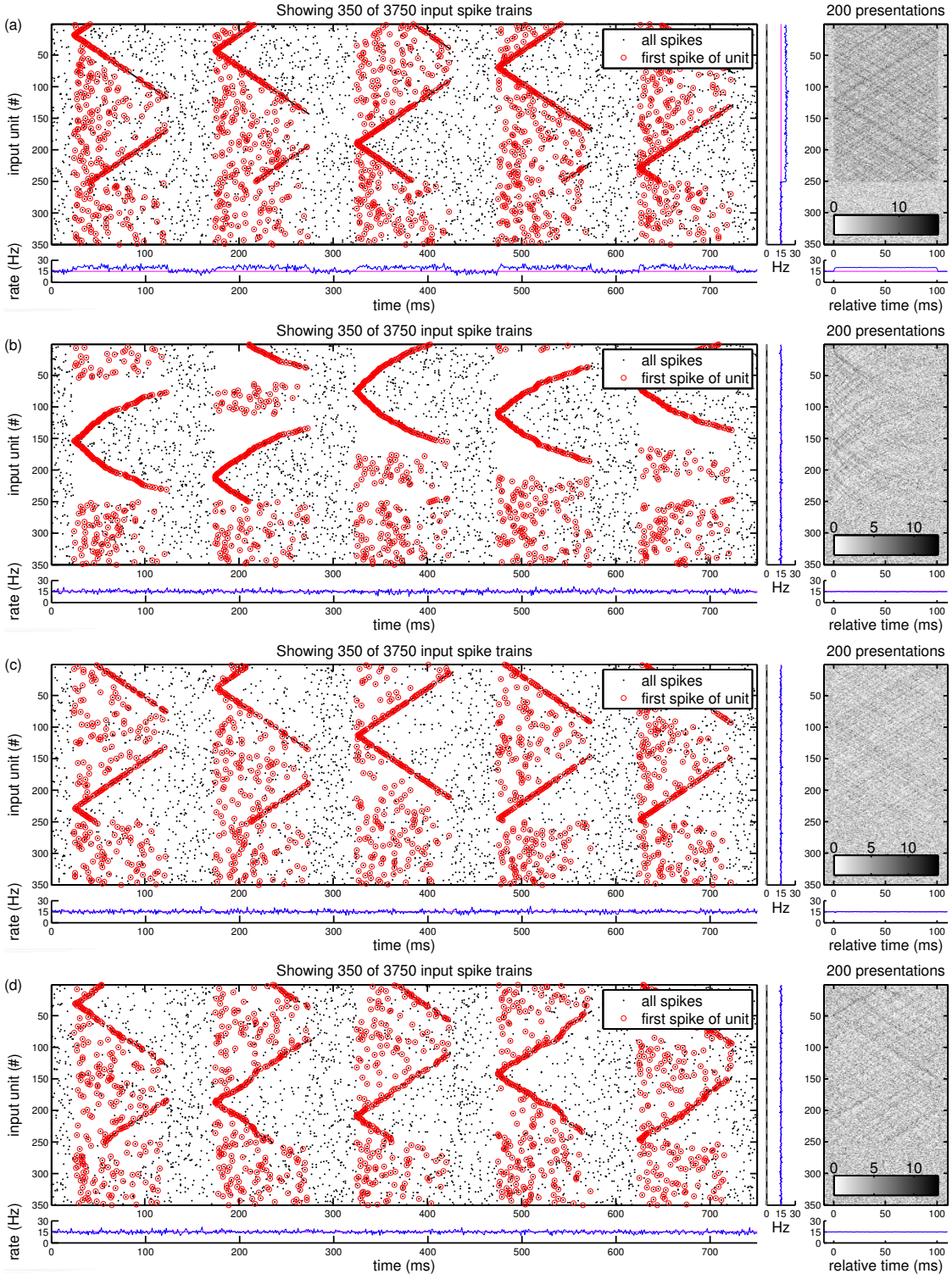


Figure 7.2: Pattern types A-D showing **multiple patterns** each. (a)-(d) Each set of subplots shows one of pattern types A-D presented by units #1-#250. Other units only fire randomly. Subplots are as in Figure 7.1. Multiple patterns are made by circularly shifting the order of firing (seen as vertical shift here). For multiple patterns, differences in time-averaged per-unit rate vanish for pattern type B here. For pattern type A, the increased instantaneous firing rate during pattern presentation is still visible here because spikes are simply added to background activity, while the instantaneous firing rate remains independent of pattern presentation for pattern types B-D. The overlay plots also show that patterns are harder to identify, but some diagonal structure is still visible. The grey scale of the overlay plots also shows that units fire far less often (ca. 15 spikes max.) within specific time bins than when only a single pattern is being presented (ca. 150-200 spikes max. in Figure 7.1).

description (Brette, 2009; Macke et al., 2009) and is more computationally efficient for large-scale simulations (Brette, 2009).

Type A: Ad-hoc Overlay

The usual first idea for embedding structured information within a stream of random input spikes is to simply add on some additional spikes that are spatiotemporally ordered. Figures 7.1a and 7.2a show such overlaid additional spikes within a stream of random background activity, for patterns that encode a single value or multiple values, respectively.

While this type of pattern is useful as an easy example for didactics, it does not hide very well among the background activity as is visible in firing rate changes whenever a pattern is present (shown below main plot). While a correlation between pattern occurrence and local field potentials may indeed exist in some brain regions, we should for now not yet rule out the possibility that structured information may be encoded without any indicators that appear within the overall instantaneous population firing rate.

Type B: Latency-based Row Sorting

A second thought is then to use a latency-based, or *rank-ordered*, rearrangement of the backround spike trains (Figures 7.1b and 7.2b). As each background spike time is independent of any previous spikes (no explicit refractory period implemented for the inputs), a simple permutation of the rows of input spike trains is enough to form detectable patterns that do *not* show in the population's instantaneous firing rate. A refractory period could also be implemented, but would not have much effect on learning outcome at the typical input firing rates of 10Hz-15Hz used here.

While the instantaneous population rate shows no indication of pattern occurrence (because the original random data is simply row-permutated), a strong difference in each input unit's overall activity is definitely observable in Figure 7.1b (vertical rate plots). At least, this difference in unit activity vanishes when multiple values are presented equally often (Figure 7.2b).

The remaining visible indicator of pattern occurrence and even value identity are the large white areas that result from permuting all rows by latency. This is also very visible in the overlay view on the right of Figure 7.1b, but at least becomes less pronounced in Figure 7.2b as more values are presented by the same group of units.

Type C: Spike Time Shifting

The next idea for avoiding large areas of white space may be to circularly shift the spike times of each unit back or forward in time to again form one or two diagonal lines within the random input data (Figures 7.1c and 7.2c). Each unit hence keeps its original spikes and only the times at which they occur are shifted. This gets rid of the large white areas of pattern type B while also hiding pattern occurrence from both instantaneous population-averaged and per-unit firing rates.

However, there remains a problem with this type of polychronous pattern generation. Consider Figure 7.3, where the underlying rate parameter that was used to generate the random spikes was not constant, but changed with time. If spike timings of each unit are shifted around in time to form a reoccurring polychronous pattern, a fluctuating population rate cannot also be adhered to (compare Figure 7.3c).

Type D: Random Choice Row Sorting

The last example for pattern generation is shown in Figures 7.1d and 7.2d, and is formed by similar methods as pattern types B and C. Specifically, as in pattern type C, a random spike is selected from each unit that happened to fire at least once. But different from pattern type C, each unit's spikes are then not shifted in time to form the pattern. Instead, as in pattern type B, the rows of spike trains are permuted to bring the selected spikes of each row in to an increasing order to form the polychronous pattern. This leads to slightly more fluctuating shapes than e.g. the perfect lines in pattern types A and C, and is still nicely detectable by STDP under most circumstances.

This pattern type is much more difficult to detect by naked eye than any of the previous types, even when units are displayed in sorted order to improve visibility. Further camouflaging is possible, as will be shown in sections 7.1.2, 7.1.3 and 7.1.4.

7.1.2 Hiding Patterns in Fluctuating Rates

In the real world, the actual population rate within a recorded group of neurons is never perfectly constant. Small fluctuations of firing rates constantly happen, and are sometimes correlated with the occurrence or even identity of information within a network. This subset of informative rate fluctuations has led to the widespread belief that the firing rate of neurons within a network is all that is necessary to understand most of the brain. In Figure 7.3 I demonstrate that information-carrying patterns of neural activity may be completely independent from fluctuations in the overall population rate, opening up a wide window of search options for deciphering our brains' neural code.

Especially pattern type D may be a good candidate to guide the search for polychronous patterns within the brain, as it shows none of the potentially revealing drawbacks of the other presented pattern types. Pattern type B also closely follows the varying target rate that drives spike probability (magenta line), while pattern types A and C warp the actual population rate (blue line) and are therefore only useful for constant or hardly changing population rates.

Besides using an Ornstein-Uhlenbeck process for controlling the fluctuating population rate target, any template target waveform such as ramps or sine waves may be useful for simulations and have been implemented.

7.1.3 Hiding Patterns through fractional participation

For a given type of pattern and underlying firing rate, the difficulty of pattern detection can be increased by simply reducing the number of units that take part in presenting a pattern. As seen in Figure 7.4a, this can be done by using a small but fixed-size set of pattern-presenting units for each value that is to be represented, potentially leading to a reduction of overlap between structured patterns.

7.1.4 Hiding Patterns through random participation

Alternatively, the decision whether a unit takes part in presenting a pattern may be randomised on each pattern presentation, leading to a less predictable but overall wider group of pattern-presenting units (Figure 7.4b). STDP is able to easily detect such patterns by simply strengthening the connections to those inputs that fire together more often than not. In a short-time recording of neural activity, however, this type of unreliable participation may indeed be hard to detect by a brute-force spike-to-spike matching data analysis algorithm.

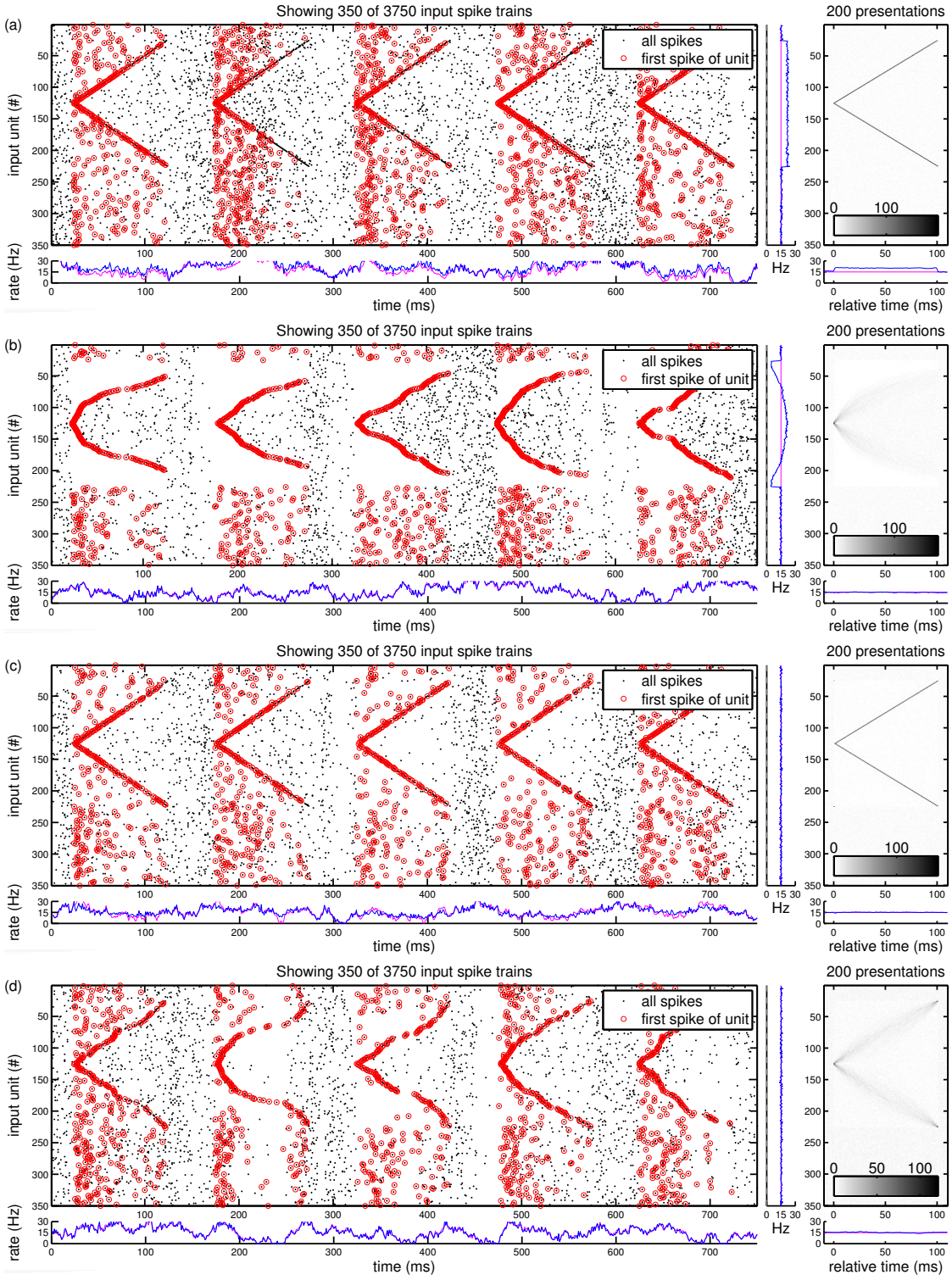


Figure 7.3: Polychronous spike patterns generated with **independently fluctuating background firing rate**. Background noise spike rate was driven by an Ornstein-Uhlenbeck process ($\theta = .05$, $\sigma = 2.55$) and polychronous patterns were generated through either overlay (a, pattern type A) or rearrangement of spikes (b-d, pattern types B-D) as described in Section 7.1.2. While in (b) and (d) the actual instantaneous spike rate (blue line below each main plot) closely follows the target spike rate (magenta line) that was given by the O-U process, the two lines mismatch in (a) and (c) during pattern presentation. Pattern type D (d) is the only one where both actual instantaneous (below main plot) and actual per-unit (right side of main plot) firing rates always match the given target rate.

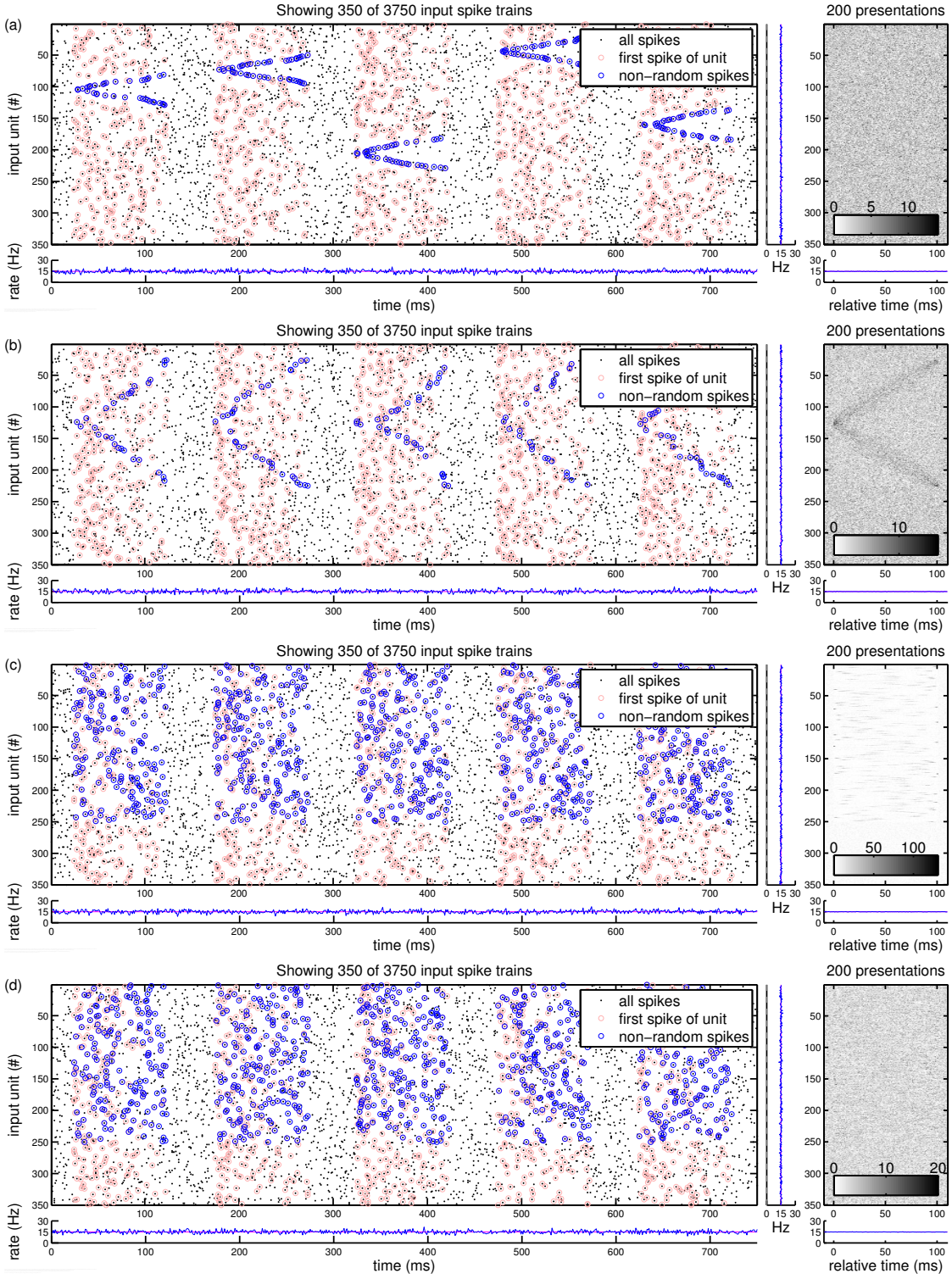


Figure 7.4: Different strategies for **hiding polychronous patterns within background noise**. Based on pattern type D. (a) Partial participation of units in a shifted pattern (Section 7.1.3). (b) Random participation on each presentation (Section 7.1.4). (c) Fixed random permutations can define a single pattern (Section 7.1.5). (d) Same as *c*, but for 40 separate recurring patterns (Section 7.1.5). When patterns do not need to conserve neighbourhood, the use of frozen random permutations can look very similar to white noise when overlaying 200 pattern presentations (overlay subplot of *d*). This may additionally be combined with e.g. random participation (Section 7.1.4) to further obfuscate the existence of structured information within a stream of noisy spike trains. However, given enough time (=samples), a postsynaptic neuron can still learn these patterns, as can be seen in supplementary Figure A.5 (p. 179).

7.1.5 Frozen random permutations of spike order

Until now, all previously presented example pattern types represent different values by shifting the order of spiking units circularly. Neighbouring input units thereby are always correlated, as can be seen in Figure 7.5a. While patterns can be more difficult to detect through cross-correlation of spike trains when some of the above measures for hiding patterns are used (Subsections 7.1.2, 7.1.3 and 7.1.4), they remain detectable by overlaying all spike trains at specific time windows (Figures 7.1, 7.2, 7.3 and 7.4b, right) and other brute-force matching methods (Gerstein et al., 2012).

But providing a strict order in which input units fire is of course not the only way in which polychronous patterns may be arranged. It would, for example, be possible to generate a random permutation of spike order whenever a new value is to be represented through a polychronous pattern for the first time. This permutation then comes to represent that specific value throughout the remainder of the simulation. Similar values are then no more represented by similar patterns, and the spike lag of single input units then can no more be used to interpolate between represented values. Two single presynaptic input units may have very different spike lags when representing neighbouring values, and may coincidentally have very similar spike lags when representing distant values. An example of this type of polychronous pattern is shown in Figures 7.4c and 7.4d for 1 and 40 different values, respectively. Any of the previously described pattern generation methods (Types A-D) can be used to generate such randomly permuted patterns, simply by choosing a distinct random permutation for each value that needs to be represented. While Subsection 7.1.6 examines the correlation coefficients of spike trains with either ordered or randomly permuted arrangement of unit order during pattern presentation, Subsection 7.1.7 further examines similarity between patterns. Appendix A.1.3 discusses learning with fixed random permutations of spike order.

7.1.6 Steady-state correlations between spike trains

A first approach when examining a set of spike trains for structured information encoded in the relative timing of individual spikes may be to compute the pairwise cross-correlation coefficients between any two spike trains in the set. If logically (e.g. topologically) adjacent units tend to fire spikes at similar times (Chapter 6 and Section 7.1.1), we receive a neural code that shows increased correlation in the spike train of neighbouring units. Figure 7.5a shows the correlation coefficients of the spike trains of 700 input units, of which 600 repeatedly present a pattern that was generated with method D from Section 7.1.1. As the patterns used here had the form of a double-sided wave (originating from a single first unit #300 and propagating in two directions from there), there is also an orthogonal line crossing the central diagonal when one pattern is repeatedly presented (column 1). When multiple polychronous patterns are repeatedly presented, more orthogonally diagonal lines of increased correlation are visible. But with increasing number of different patterns, each orthogonal correlation line becomes less pronounced.

When instead of ordering by adjacency, a new random permutation of unit order is chosen for each pattern, correlations between pattern-presenting units are far less clear (Figure 7.5b). While single pairs of units still remain highly correlated when one pattern is repeatedly presented (column 1), any structure within the spike trains becomes very hard to detect through computing the correlation coefficients when multiple patterns are presented (columns 2 and 3). For improved visibility, each plot of Figure 7.5 contains a zoomed-in version in its lower left corner. Also, the colour scale of each plot is set to the minimum and maximum of non-diagonal correlation coefficients, producing a slightly

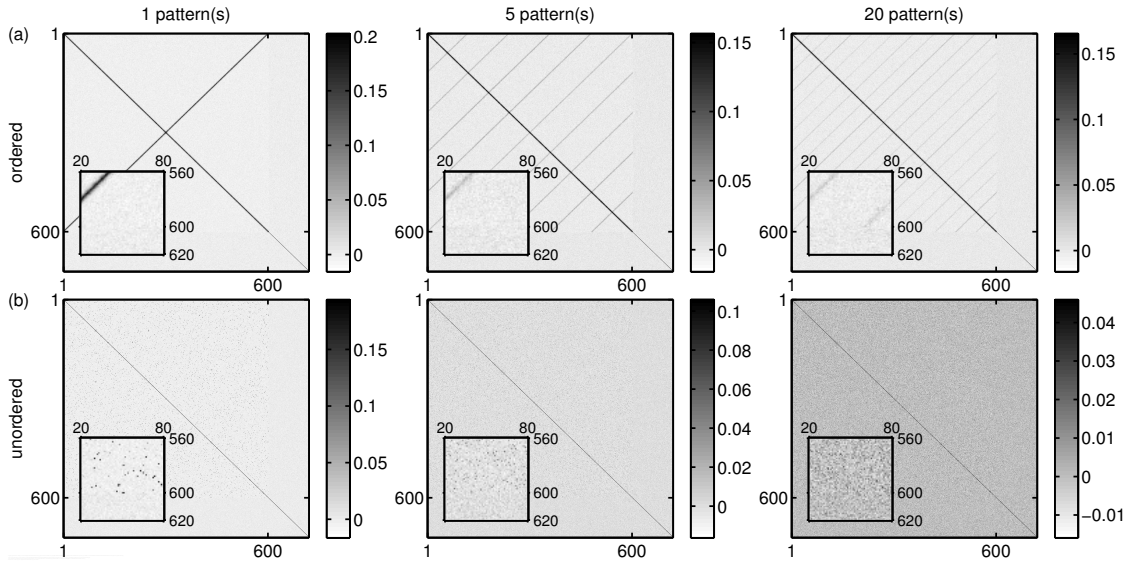


Figure 7.5: Correlation coefficients of spike trains of 600 input units, containing 200 polychronous pattern presentations. Pattern duration was 100ms, with another 100ms of noise between any two patterns, yielding a total length of 40s per spike train (1ms time bin size). Insets show zoomed-in version. The value to be represented by the patterns was either fixed (column 1) or selected randomly from 5 or 20 equidistant value options (columns 2 and 3). (a) Patterns generated via pattern type D (ordered, Section 7.1.1). The lines orthogonal to the diagonal are due to the two-sided shape of the patterns as seen e.g. in Figure 7.1. (b) Generated via pattern type D (unordered, Section 7.1.5). Correlation does not follow from neighbourhood. However, some dark speckles can be seen in column 1, indicating correlation. Colour scale is set to minimum and maximum of correlation coefficients that are not on the diagonal (because the diagonal merely represents the autocorrelation coefficients of each spike train with itself). Notice how the maximum crosscorrelation coefficient decreases as more patterns are presented (darker subplots in columns 2 and 3 than in column 1).

darker overall tone in column 3 of Figure 7.5b.

Although correlations between the spike trains of pattern-presenting (input) units are hardly visible when spike order is randomised and multiple patterns are being repeatedly presented, we found that STDP succeeds in learning to detect such obfuscated patterns within a stream of arriving spikes (see supplementary Figure A.5, p. 179).

7.1.7 Pattern Similarity and Unit Overlap

How do patterns differ when they are presented by the same set of input units? In Section 7.1.1, patterns for different values are generated by circularly shifting the center of the spike wave around the set of pattern-presenting units while maintaining neighbourhood relationships between units. In Section 7.1.5, the adjacency-preserving circular shift is replaced by a new random permutation of spike timing for each new value that is to be presented. In the first case, patterns for similar values likely also have some similarity in the timing of spikes, while patterns representing distant values may differ more. In the second case, the degree of similarity between patterns is maybe less intuitive. We can use a standard metric for comparing spike trains and measure the number of time bins each spike within one pattern would need to be shifted in order to receive any of the other patterns. This comparison is shown exemplarily in Figure 7.6, where a set of 40 patterns is compared in terms of the sum of absolute spike timing differences. While ordered (adjacency-preserving) pattern types show a small difference between patterns that represent similar values and a strong difference between distant values is a circular

arrangement (a,b), the difference between randomly permuted patterns remains largely constant around some mean value (c). As all other patterns are compared to the pattern representing the value 0.25 in this figure, the point of zero difference is at this value on the x-axis. All of the three general pattern generation techniques shown here (a,b,c) have likely identical mean pattern differences, at about two thirds of the maximum difference for ordered patterns.

There is, however, a strong drawback to this method of measuring differences between patterns. According to the metric of summing the absolute spike timing differences that is used here, it seems unimportant whether one-sided (Chapter 6) or two-sided (Sections 6.3 and 7.1.1) ordered patterns are used, as the plots in (a) and (b) of Figure 7.6 are identical. This is not the case. Even if the absolute difference between spike times of different patterns is high, a neuron using STDP may not allow the receptive field to settle on a specific pattern when the earliest units that fire together in one pattern also fire in close temporal proximity within some other pattern. If the early part of a pattern matches some other part of another pattern, a receptive field that is tuned to detect the first pattern may easily be attracted by the second pattern, thus leaving the first pattern without any neurons that detect it. This is because STDP always tunes a receptive field to only the earliest inputs, as was explained in Section 6.2.1 (p. 84). Also, how high is the probability of this happening when patterns are created through random permutations as in Section 7.1.5?

We therefore need a new metric that respects this feature of STDP. The first inputs of a polychronous pattern are more important than inputs that arrive later. This is because only the earliest inputs take part in causing the first postsynaptic response after pattern onset. Specifically, the size of this group of early-firing input units is bounded by the time at which the postsynaptic neuron tends to respond after pattern onset. We can measure this timespan and use it as a (rectangular) shifting window to see how many of these early-firing units of one pattern are also active together within close temporal proximity in any other pattern.

Let us call the size of this group m and assume that $m = 30$ early units fire before a postsynaptic neuron can respond. We also assume the total number of pattern-presenting units to be 600 and a total pattern duration of 600 time bins within this example. As each time bin then contains exactly one spike, the timespan needed for the first 30 units to spike is then exactly 30 time bins wide. We use this timespan as our shifting window

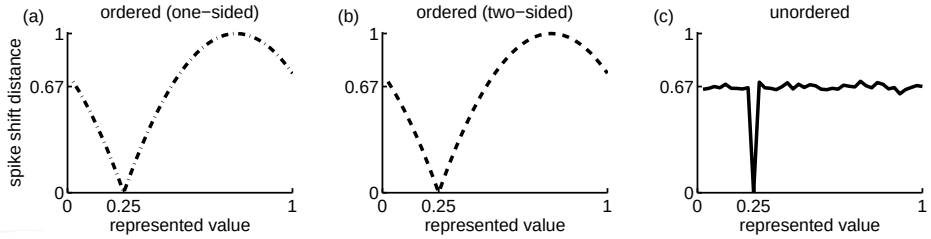


Figure 7.6: Distance between patterns when generated with predefined spike ordering (a,b) or when only random ordering is used (c). When patterns are generated through circular shift of spike order (Section 7.1.1), any patterns representing adjacent values are more similar than patterns representing distant values (a,b), while all patterns have a mean difference of two thirds of the normalised maximum range when random permutation (Section 7.1.5) is used (c). Pattern distances were calculated by counting the number of shifts required for all spikes of one pattern (here representing the value 0.25) to match all spikes of any second pattern. The differences between 40 patterns are shown here for visual clarity. The standard spike distance metric used here is insufficient to discriminate between (a) and (b), so we need an alternative way of measuring the differences between one-sided and two-sided ordered spike arrangements (see Figure 7.7).

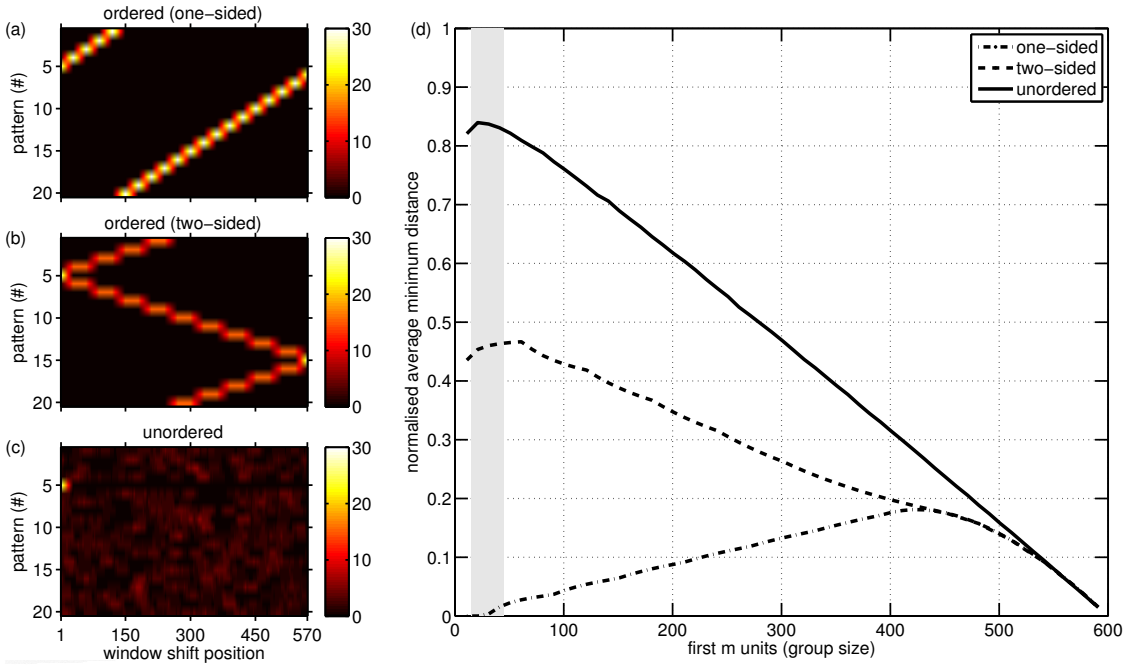


Figure 7.7: Distance between patterns for all three variants: one-sided ordered, two-sided ordered, and randomly permuted. (a-c) The timespan required for the first $m = 30$ units firing in pattern #5 is used as a shifting window within each pattern to track how many of these units fire together again at any other time or for any other pattern. Colour shows number of matches. 20 distinct patterns are used here (rows), and $600 - m = 570$ starting timepoints are used for the shifting window (columns). (*a*) one-sided pattern variant. (*b*) two-sided pattern variant. (*c*) randomly permuted (=unordered) pattern variant. (*d*) Normalised distances between each of the three pattern generation variants, for any group size m of earliest firing units out of a total of 600 pattern-presenting units. The grey area indicates typical group sizes of strong weights produced by STDP in simulations. As polychronous patterns allow STDP to form sparse receptive fields (Section 6.2.1), the differentiation to distinct selective receptive fields is ensured for randomly permuted and for two-sided ordered pattern variants. *Summary of metric algorithm:* The minimum distance (= maximum match) is taken for each pattern, and the average distance of other patterns is divided by the distance of the selected pattern for normalisation. This fraction is then computed for each of the 20 patterns, repeated ten times, and again averaged to form a datapoint in the plot. Variance is too small to display.

and count how many of the 30 earliest units of one pattern are active together within any other pattern at any shift position. This produces plots (a), (b) and (c) in Figure 7.7 for each of the three pattern generation variants, respectively. We now see a clear difference between one-sided (a) and two-sided (b) patterns. When one-sided ordered patterns are used, the first 30 units of pattern #5 also fire together in all other patterns, only shifted in time. This is indicated in Figure 7.7a as a bright yellow area in each row. A receptive field that is tuned to the earliest units of one pattern can therefore easily be attracted by any other pattern and prevent the receptive field from settling on any single pattern at all.

In contrast, two-sided ordered patterns overlap with only half of the earliest units in any other two-sided pattern (Figure 7.7b). Assuming the other network parameters are set up in such a way that half of the earliest units are insufficient to evoke a postsynaptic response, this simple construction of two-sided polychronous patterns allows the receptive fields to settle on distinct sets of early-firing input units, i.e., the beginning of different patterns. In practice, STDP will prune the set of earliest firing input units to the minimum necessary to evoke a postsynaptic response (see Section 6.2.1), so the requirement that

half of these input units should *not* evoke a postsynaptic response is easily fulfilled.

The careful observer will have seen that in Figure 7.7b the full set of 30 earliest units of pattern #5 also fire in close temporal proximity at the end of pattern 15 (out of 20 patterns). In practice, this apparent competition between opposite patterns does not pose a problem for the stability of receptive fields because any attraction by late units of pattern #15 will be matched by the early parts of pattern #5, and negative synaptic drift will therefore at most reduce synaptic weights but not succeed in snatching the receptive field away from pattern #5, as long as both patterns remain to be presented equally often.

When input units fire in a fixed random permutation (Figure 7.7c), it is very uncommon for more than a third of the earliest input units of one pattern to ever fire together in any other pattern. This is indicated in the example figure by the fact that the only bright yellow area is at the original location of the earliest units of pattern #5. However, random matches of around 10 of these input units are often encountered, causing a patchy dark red area throughout most of the plot.

The minimum group size m of strong input connections required to evoke a postsynaptic response depends on a number of factors including the maximum strength of each EPSP, somatic excitability, somatic leak, and the average timespan during which reliable inputs arrive in a non-synchronous pattern. We can therefore not restrict our new metric to a fixed group size of earliest firing input units, and need to compute similarity for a range of group sizes.

The basic idea of the new metric for comparing pattern distances is that we contrast the maximum number of units that fire within close temporal proximity with how many of these fire together within other patterns on average. We do this for a large number of group sizes m and for all three pattern generation variants to produce the three curves in Figure 7.7d. The algorithm is given in pseudocode in Appendix A.3.8.

Figure 7.7d shows that the average distance between two-sided ordered patterns is always higher than one-sided ordered patterns when only a small set of early-firing input units is used for comparison. The grey area marks typical values for m as produced by STDP. When topological structure is to be contained in similarity between patterns, then two-sided ordered patterns should be used instead of one-sided ordered patterns because only two-sided patterns allow robust receptive field formation. However, when topological structure is not of importance, randomly permuted unit order has the highest average distance.

7.2 Effects of Network Size, Pattern Duration, etc.

On top of the pattern generation method that is used, a number of other parameters affect the learning outcome and speed.

7.2.1 Network Size and Projection Scaling

The number of inputs to the postsynaptic detector neuron would strongly affect postsynaptic firing rate if it was not compensated for. The effect that each incoming spike has on the postsynaptic membrane potential therefore is divided by the total number of inputs.

It is important that a sufficiently large input network size is used, as very small input sizes (ca. < 500 units) will give each incoming spike a too strong influence on postsynaptic activity, circumventing the need for many inputs to arrive together to evoke a postsynaptic spike. If a single input unit can strongly affect the postsynaptic response, the postsynaptic neuron may tune to single input units instead of a given pattern of spatiotemporally firing

units, no matter if those units take part in presenting polychronous patterns or if they just constitute background noise spikes. The problem of too small network sizes has also been mentioned in (Gütig et al., 2003), where random winners were seen only for very small input networks.

7.2.2 Pattern-presenting vs. Background Units

The ratio between pattern-presenting input units and those that only ever fire randomly and contain no inherent meaningful information (background inputs) has a noticeable impact on initial learning success. However, the effect of background units on postsynaptic activity decreases strongly as (simulated) learning progresses, when a biased STDP rule with negative drift is used. (See e.g. the decrease of connection strength to units #1-#700 and #1301-#2000 in Figure 6.12, p. 86).

After this decay of connections to randomly firing background units, they have little effect on postsynaptic activity. That is, as long as the level of possibly existing neuromodulators remains constant (See Part 3 of this text). In Chapter 8, as I present a new method for applying dopaminergic modulation to an STDP-based network of neurons, the randomly firing background units become important once again, as a source of entropy for causing random postsynaptic responses on low levels of the neuromodulator.

7.2.3 Pattern Duration and Interpattern Spacing

The duration of polychronous patterns in all previous examples in this chapter has been 100ms, but shorter timespans of e.g. 50ms, 30ms or 15ms are also detectable by STDP within a very noisy environment, depending on network parameters. Even if a fully trained detector neuron can respond to an incoming pattern within 3-5ms, a further shortening of pattern duration has a drawback during the learning process: When the pattern duration is very short, the postsynaptic neuron has only a very small time window in which it must coincidentally fire in order to catch the pattern. With shorter pattern duration it may therefore take a long time until a postsynaptic neuron begins to tune to a given pattern. For longer pattern durations, the postsynaptic neuron may fire at any point during this time and thereby begin to respond to that pattern. Due to the STDP-inherent migration of strong synaptic connections to the earliest firing units (Section 6.2.1, pp. 84), any postsynaptic neuron that picks up a pattern at any point during its presentation will soon respond to it with minimal delay.

The minimum space between two patterns should be large enough to avoid a fusion of similar patterns that often occur in fixed order to fuse into one. As the usual STDP time constant is around 15ms-30ms, I choose an inter-pattern spacing of 100ms during which only random noise spikes are fired by all units.

7.3 Multiple Postsynaptic Neurons & Input Space Coverage

We have until now been dealing with the effects of polychronous patterns on a single postsynaptic neuron. I have shown that a single neuron chooses to tune to one of many polychronous patterns, and forms a minimal set of strong connections that allow it to robustly detect further occurrences of that pattern (Section 6.3, p. 98). The choice of which pattern a postsynaptic neuron will tune to and which final receptive field it will settle upon has seemed difficult to predict until now. In particular, it was unknown whether each pattern has the same probability of being learnt by a postsynaptic neuron, or if some yet to be uncovered additional factors influence the choice of pattern that a

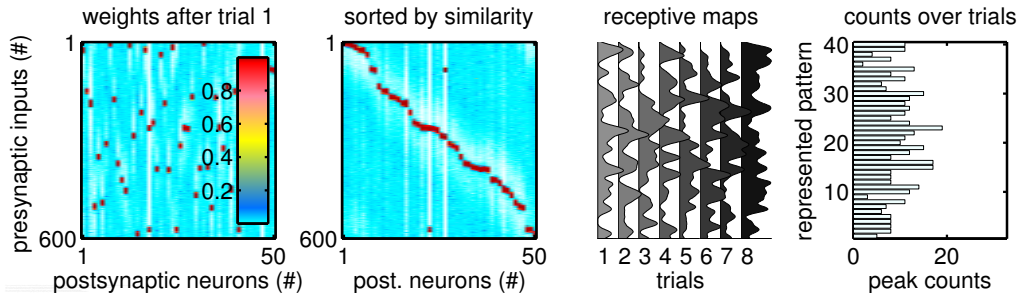
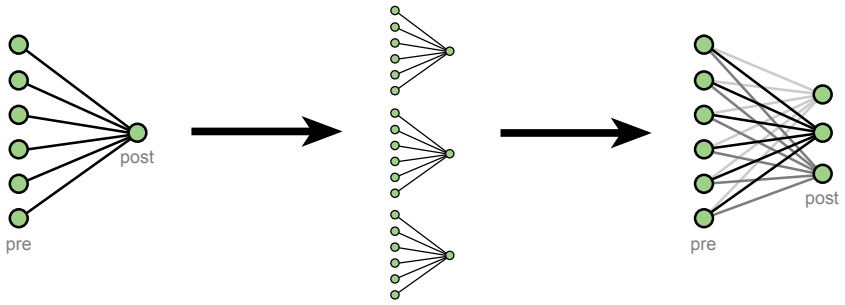


Figure 7.8: Multiple independent repetitions of the simulation experiment yield a smooth coverage of the full space of input patterns. This is an overview figure that captures the results of multiple repetitions (trials) of the same simulation (also used in Chapter 9). The first and second panels from the left show all synaptic weights between postsynaptic neurons and pattern-presenting input units (here #1 - #600) in original unsorted arrangement (panel 1) and sorted by similarity of receptive fields (panel 2). The peak of strong weights of each receptive field is found for each postsynaptic neuron, and the 600 presynaptic input connections are compressed into a histogram of peak locations, which is then convolved with a Gaussian smoothing window to form a single column (panel 3). The compressed representation of the first two panels can be seen as the first (=leftmost) column in panel 3. Panel 4 finally is a combined histogram of all eight repetitions (trials) which roughly hints at the total number of neurons that learnt each pattern.

postsynaptic neuron tunes to. To isolate the cause for this choice, I constructed and ran a set of simulation experiments that differ only in the small fluctuations of random background activity of input units, and how these presynaptic units are connected to one or more postsynaptic neurons.

First, a set of 50 simulations with a single postsynaptic neuron each (see diagram in Sketch 7.9, center) and exactly equal initial settings but random background activity was set up and run (see Appendix A.3.9 for simulation details). I found that giving each postsynaptic neuron its own set of independent inputs leads to a uniform distribution of receptive fields across the full range of presented patterns. Figure 7.8a shows the receptive field of each neuron after simulation, which when ordered according to similarity (Figure 7.8b) resembles a diagonal line across the full range of synaptic weights. Hardly any neurons share identical receptive fields, and nearly all of the space of inputs is covered by a receptive field peak of at least one neuron. Figure 7.8c shows that the distribution of receptive field peaks – or what I now call a *receptive map* – resembles a uniform distribution, and this remains true for many repetitions (*trials*) of the experiment. The first two panels of Figure 7.8 are shown in a compressed view in panel 3 (trial 1), smoothed



Sketch 7.9: We now expand our network topology from a single postsynaptic neuron (left) to multiple postsynaptic neurons without recurrent connections. In Figure 7.8, every postsynaptic neuron receives an independent set of inputs, comparable to independent repetitions of the simulation (center). In all other figures, all postsynaptic neurons receive the same inputs, including random fluctuations of the background firing rate (right).

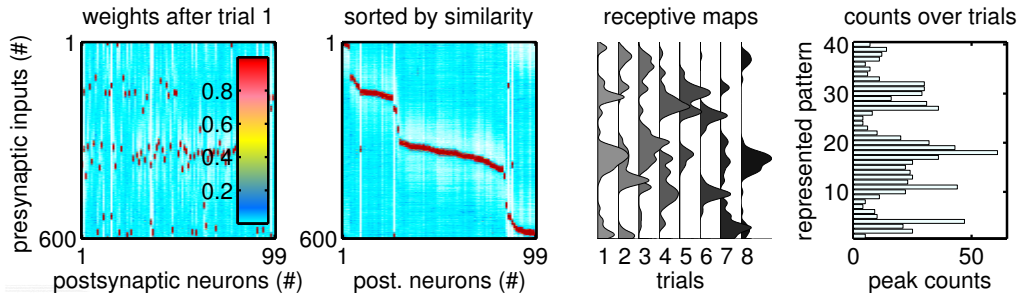


Figure 7.10: When all postsynaptic neurons receive identical inputs, any random fluctuations in the input firing rate are received by all neurons. This usually leads to a common bias towards a subset of patterns and the resulting distribution of receptive fields is no longer uniform. As this non-uniformity is not based on any meaningful information but comes from known randomness, it should be treated as an unwanted artifact in the learning process. Instead, any clusters in the distribution of receptive fields should represent non-random significant features. Figure description as in Figure 7.8.

by a Gaussian bell shape for visual clarity. The other trials in this panel are repetitions of the complete experiment. While there is variation in the distribution of receptive field peaks from trial to trial, the peaks are overall evenly dispersed across the full range of pattern-presenting inputs. The sum of all peaks at all positions in all trials is shown as a uniform-like histogram in Figure 7.8d.

However, these uniform distributions of tuning probability only hold when each postsynaptic neuron really receives individual inputs with independent background rate fluctuations. Unfortunately, this is an unrealistic case for a biological neural network where the neurons in one (cortical) layer always receive (at least partially) overlapping inputs from many neurons of some previous layer. A more biologically plausible case can be simulated by connecting the postsynaptic neurons to a single group of input units instead of connecting each neuron to a separate set of inputs (compare Sketch 7.9 right). This not only presents the same polychronous patterns to all postsynaptic neurons at the same time, but also provides them with identical fluctuations in the input firing rate. When we run this simulation and feed the generated input data into multiple postsynaptic neurons (Figure 7.10a+b), the resulting distribution of receptive field peaks is a lot more clustered (Figure 7.10c). The neurons have a high probability to tune to the same set of patterns. As each pattern was presented equally often and all fluctuations in background input firing rate are by design random, the location of clusters does *not* represent any meaningful hotspot of information.

It would be nice if it did. If clusters within the receptive field map would not happen accidentally, we could use them to signify some important event, as a cluster in the receptive field map means that many postsynaptic neurons will be responding to the same stimulus / input spike pattern. In Part 3 of this text and in Section 7.5 of this chapter I examine possible uses for receptive map clusters and how to guide their formation intentionally.

But first, if clusters in the receptive map are to signify meaningful information, we need to ensure that without any intentionally encoded information, the distribution of randomly forming receptive field peaks shows little or no variation from trial to trial. In particular, if each pattern really does have the same probability of being learnt by a postsynaptic neuron, the distribution of receptive field peaks should always approach the uniform distribution. I wondered whether it may be possible to regain a uniform distribution of receptive fields even when strongly overlapping inputs are given to multiple postsynaptic neurons. A few solutions come to mind.

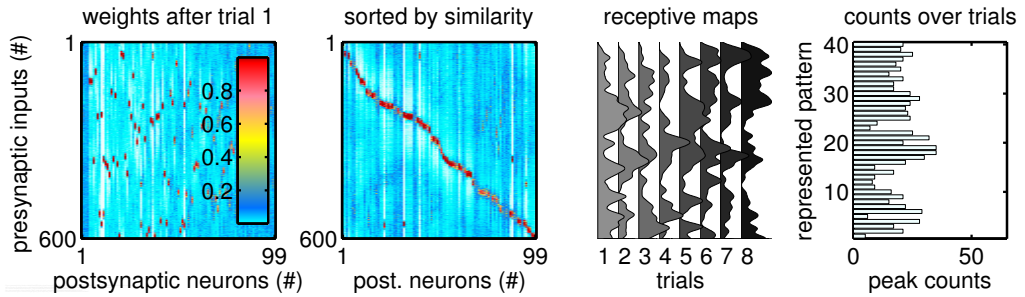


Figure 7.11: An easy solution to the problem of accidental cluster formation from rate fluctuations that by design are known to be random. By simply beginning the learning process of all postsynaptic neurons at different times, a uniform-like distribution of receptive fields is again obtained. This solves the question of how to learn a weight matrix for Figure 6.8 (p. 82). See Figure 7.8 for subpanel descriptions.

7.3.1 Avoiding Mutual Inhibition

A common approach for restricting the random formation of clusters within the receptive map of a neural network is to use mutual inhibition between postsynaptic neurons. When the first neuron begins to respond to a given pattern, (predefined) inhibitory connections may be used to inhibit the response of other neighbouring neurons to the same pattern, which keeps them from also tuning to that pattern. Depending on the maximal strength of inhibitory connections, multiple k winner neurons may be required to respond to a pattern together in order to sufficiently inhibit further neurons from responding.

There are a few problems with this approach, however. First, it assumes the existence and dedication of inhibitory mutual connections between neurons in a group that receives common inputs. It gives no hint on how a similar feature may be implemented in a part of the brain where no inhibitory mutual connections have been found anatomically. It also prematurely prescribes a fixed role to the inhibitory connections of brain regions that do contain inhibitory mutual connections, closing the mental door to a possible examination of alternative uses for those connections.

Second, if the dominance of a group of receptive fields within a group of neurons was actually intended through external circumstances, mutual inhibition that is targeted at discouraging accidental receptive map clusters will also hinder such an intended formation of clusters in the receptive map.

Instead of assuming that a large portion of mutual inhibition within a group of neurons is dedicated to decorrelating accidental clusters that arise from strong overlap between the inputs to a group of neurons, I chose to look at ways of forming unbiased receptive maps without the use of mutual inhibition.

7.3.2 Smooth Coverage of Input Space through Delayed Onset

A simple solution to this presents itself in the idea of neurogenesis. With the insight that in the brain, no two neurons are alike and new connections (and new neurons) are formed continuously, it is highly unlikely that a large number of new neurons will reach their critical period of being highly responsive to most inputs at the same moment in time. This is often not reflected in computer simulations of neural plasticity, where learning starts at the same time for all virtual neurons of the simulation.

I ran computer simulations where a postsynaptic neuron became plastic every second until all neurons had activated STDP. Already at this onset time delay, the resulting distribution of receptive field peaks approaches a uniform distribution again, as seen in

Figure 7.11. As in the two previous figures, panel (c) summarizes the results of multiple repetitions of the simulation experiment, showing the improvement over the case when all neurons became plastic at the same time (Figure 7.10). While the distribution of receptive field peaks may not be perfectly uniform yet for an onset delay of 1s, the random formation of clusters is already largely reduced. It is to be assumed that a further increase of onset delay would further allow the distribution of receptive field peaks to approach uniformity. As this would also increase the total time the simulation needs to run until the last neuron has settled into a stable receptive field, and the gain would be minimal, I henceforth keep the onset delay at 1s for the rest of this text.

7.4 Noise: A feature, not a bug?

The brain, as any physical system, is noisy. A continuous goal of any neural system, therefore, is to improve the signal-to-noise ratio of meaningful pieces of information compared to random background activity. This separation between signal and noise is not always clear, and seemingly random spikes may be found to contain valuable information for fulfilling a certain task. At the same time, an input signal that would need many bits of information to be described (has high entropy) may still contain little useful information for a task at hand (see Chapter 4).

In the previous section, I showed how the choice of a neuron's receptive field can be strongly influenced by random fluctuations in its inputs' background activity. These background spikes contain by definition no meaningful information, as they were generated for the simulation by a random number generator. Still, this background noise can have a strong influence on the choice of pattern that the neuron will eventually tune to.

The negative drift of biased STDP (Chapter 5) helps to improve the signal to noise ratio of a neuron's response through learning to ignore inputs that have in the past fired (mostly) at random and have not been seen to transmit useful information. But where should the line be drawn between a randomly firing background unit and, for example, a unit that fires random spikes only most of the time but occasionally takes part in presenting a structured pattern of spikes that could be picked up by a postsynaptic neuron over time?

While exact definition boundaries depend on the specific parameters of the (biological or simulated) neural network, the principle behaviour of STDP may be evaluated under conditions of increasing noise in future biological experiments. Noise originating in the system itself may be a detrimental factor when trying to robustly detect a pattern of spikes that represent some external stimulus. But another cause for a bad signal to noise ratio may indeed be the unknown presence of a given stimulus within the world. It is important for the neural system to not overestimate the presence of some external stimulus and not overtrain on unreliable data. Noisy inputs help to prune less informative synapses through negative synaptic drift for the benefit of more informative ones and thereby keep the set of strong synapses that a neuron needs to maintain at a minimum (see Sections 5.1 and 6.2 for pruning of synapses that convey little information).

7.5 Biasing Learning Direction

In many cases, we would like to use spike timing dependent plasticity to learn to respond to specific features of the world. So how can we transform the unsupervised learning paradigm that is STDP into supervised learning without having to work against the unsupervised nature of STDP? This section lists some approaches that may help to use normal STDP in a supervised (or reinforcement learning) task.

7.5.1 Non-uniform pattern presentation counts

A first easy approach to steering STDP is to control the relative number of pattern presentations a plastic network encounters. Without further normalisation, a pattern that is presented more often than others will have a higher probability of being picked up by a neuron's receptive field than a pattern that is presented rarely. This indeed has such a strong effect that it poses a problem for the application of neural networks to real-world data without introducing additional inhibition between neurons. If only a single postsynaptic neuron is observed that receives no additional inhibitory inputs to account for stimulus frequency, it may easily be pushed into tuning to the most frequent pattern/stimulus.

However, this feature may also be used to the experimenter's advantage in tasks where he/she has control over the number of pattern presentations and can control the relative effect of compensatory inhibition.

7.5.2 Background rate modification with certain patterns

Another approach is to take control of the background firing rate of inputs for influencing the time of postsynaptic responses (Section 7.1.2). If the background firing rate is increased reliably whenever a specific pattern is presented to the network, the postsynaptic neuron will fire more often during presentation of this pattern, which in turn will increase the probability of tuning the postsynaptic neuron's receptive field to this pattern.

However, the presynaptic background rate is in reality influenced by a large number of factors, and it seems unlikely that changes of background rates are used as the main mechanism in biology to control which pattern or stimulus becomes trained. Still, this remains an option that may occasionally be used in biology and seems useful in some computer simulation tasks.

7.5.3 Pattern-dependent changes to noise level

As mentioned in Sections 7.1.3, 7.1.4, and 7.4, it is possible to hide patterns in the background noise. When this is done for only some patterns and not for others, those that are less hidden will be responded to more often, and more postsynaptic neurons will tune to less hidden patterns. When the noise level is artificially increased for specific patterns, it may be hard for the postsynaptic neuron to tell whether this is an internal function of the brain or if a specific stimulus is less present in the environment.

As for the non-uniform pattern presentation probabilities mentioned above, this method also may lead to an over-learning of the less hidden patterns.

7.5.4 Contrast enhancement or worsening

An interesting possibility is to increase the contrast of specific pattern-presenting input units whenever they present a pattern that is to be learned. This is different from the previous point in that the contrast of some inputs versus others may be controlled during synaptic transmission and does not need to include changes to the presynaptic input layer. Indeed, we use this method for transforming the unsupervised learning function of STDP into a reinforced process in Chapters 8, 9 and 10.

7.5.5 Multiplication by scalar value

Unfortunately, a bad example for turning unsupervised STDP into supervised or reinforced learning is to multiply the prospective weight change computed by an STDP rule by

some scalar value as is often done in traditional Hebbian learning. The increased level of biological reality when moving from correlation-based learning to timing-based learning is lost when the scale of STDP is suddenly changed vastly on each learning step. Chapter 5 closely examines the importance of the scale of LTP and LTD in STDP for synaptic drift and how this effects random inputs relative to inputs that contain some inherent spatiotemporal structure.

When this method for biasing STDP is used in the literature (Frémaux et al., 2010), it often leads to the need for active suppression of the main (unsupervised) features of STDP.

7.6 Summary

In this chapter, we saw how polychronous patterns may be generated efficiently for a computer simulation and be hidden within a stream of seemingly random spikes. The fact that it is so easy to hide spatiotemporal patterns may point us towards a renewed search for polychronous patterns within electrophysiological recordings in the future (Vogt and Hofmann, 2015d). Especially the unreliable participation of units in presenting a polychronous pattern (Section 7.1.4) and the possibility that polychronous patterns may be independent of population rate (Section 7.1.2) while still being detectable by a neuron that was trained through STDP may lead us to new experimental methods in the future. For instance, the idea of spike-triggered averages may be extended into pattern-triggered averages where only the presence of a specific polychronous pattern instead of a single unit's spike activated some trigger within the recording software.

In Section 7.3 we explored what happens when multiple postsynaptic neurons without any recurrent connections are given the same inputs to learn a polychronous code. It turns out that when the background fluctuations for each postsynaptic neuron are fully independent while being given the same patterns, we receive an evenly distributed map of receptive fields. When all postsynaptic neurons receive completely identical inputs (= identical background fluctuations), the development of receptive fields becomes biased. But this can be largely restored through the simple trick of initialising neurons at different times, which seems very likely in biology as part of neurogenesis.

After having established a framework for plasticity with temporal coding though STDP and polychronous patterns in Chapters 5, 6 and 7, we can now combine this with neuromodulation in the next three chapters.

Part III

Neuromodulation in spiking networks

Chapter 8

Influencing Plasticity through Modulation of Synaptic Transmission

In order to understand procedural learning in the brain, we cannot settle for unsupervised learning methods such as STDP alone. Reinforcement learning models of the basal ganglia have been suggested, but usually only in a very abstract form. In the past, the application of a reward error or reinforcement signal into a (spiking) neural network has usually been done by multiplying some prospective weight change with some scalar value (Farries and Fairhall, 2007; Izhikevich, 2007b; Morrison et al., 2008; Potjans et al., 2009). The biological justification for doing this was taken from biological experiments (Reynolds and Wickens, 2000, 2002) in which dopamine was found to influence the outcome of a plasticity process in the basal ganglia's striatum. As modellers have moved from more abstract models of plasticity to using a specific dependence on spike timing (Chapter 2), they have oftentimes continued the simple multiplication of a scalar "reinforcement" value with the otherwise biologically well documented weight changes arising from STDP (Section 3.3.2). It may be time to take a closer look at possible mechanisms of how dopamine and other neuromodulators may affect synaptic plasticity in a more realistic manner.

Another strong motivation for the existence of this chapter are biological findings that dopamine by far does not *only* influence synaptic plasticity. Instantaneous effects of dopamine on the excitability of neurons have been found in dopamine-receiving brain areas including the basal ganglia and prefrontal cortex (Kroener et al., 2009; Thurley et al., 2008). More low-level instant influences on postsynaptic currents across cell membranes have also been observed (Hernández-López et al., 1997; Lee et al., 2004a; Nicola et al., 2000; Nicola and Malenka, 1997; Rotaru et al., 2007; Waters and Helmchen, 2006).

Finally, some authors have speculated that dopamine may act as a *contrast enhancer* for glutamatergic inputs to a dopamine-receiving cell (Nicola et al., 2004), but no models (other than ours) have captured this on the level of spiking neural networks until this date. In Vogt and Hofmann (2012), we not only construct a model that captures the effects of instantaneous modulation of contract in synaptic transmission within spiking neural networks, but also demonstrate first effects that this has on synaptic plasticity.

The remainder of this chapter is a close reproduction of our 2012 paper, with some changes especially in Section 8.2 (*Model and Methods*) for improved clarity and better integration with the rest of this text. The following two chapters (Chapters 9 and 10) further explore the implications of the new paradigm for synaptic plasticity. As a consequence of the paradigm for reward-modulated synaptic transmission that is presented here and in Vogt and Hofmann (2012), it will likely be possible to build powerful reinforcement learning models with high biological realism in the future.

8.1 Introduction

Recent experiments have indicated that dopamine may directly influence the spiking activity of prefrontal neurons by increasing signal-to-noise ratio (Kroener et al., 2009) or gain (Thurley et al., 2008) during synaptic transmission. These studies suggest that an instant effect of dopamine may only be present during additional glutamatergic synaptic input and that dopamine may not be capable of eliciting a synaptic response on its own here.

While the influence of varying dopamine on synaptic learning processes has been recognised and modelled in a large number of publications (Farries and Fairhall, 2007; Izhikevich, 2007b; Morrison et al., 2008; Potjans et al., 2009; Reynolds and Wickens, 2002), this has usually been done by directly adapting the STDP rule to include an additional third factor signalling dopaminergic reinforcement, beside the inclusion of the pre- and postsynaptic activities. The requested synaptic weight change defined by STDP would often be multiplied by a reinforcement factor in the interval [-1,1], yielding no synaptic learning for baseline levels of dopamine (zero reinforcement) and inverted, or "anti-hebbian", learning for negative values of reinforcement. Postsynaptic activity would be only indirectly affected by dopamine in these models, through the gradual synaptic weight change induced by reinforced STDP.

However, there have been modelling studies incorporating experimental evidence on increased postsynaptic facilitation under dopamine exposure (Chorley and Seth, 2011), albeit by affecting the neuron-wide recovery function of the postsynaptic model neuron. To our knowledge, no previous modelling studies of dopaminergic influence on synaptic transmission with an only indirect effect on the synaptic learning process have been published.

We chose to investigate the possible network level implications of a (dopamine-like) neuromodulator purely affecting synaptic transmission on a local scale. As any postsynaptic activity is highly dependent on the received synaptic input, changes to this input directly affect the postsynaptic neuron's spiking activity as the second factor in usual STDP rules (Bi and Poo, 2001). Any neuromodulation of synaptic transmission is therefore in principle able to affect the learning outcome of STDP, even when no direct involvement of the modulator in the actual weight adjustment process is present.

There are in theory two ways of affecting the amount of synaptic input a neuron receives. The first would need sufficient control over the spiking activity of presynaptic neurons (see also Sections 7.5.1 and 7.5.2), which is difficult to provide for inputs arriving from distant brain areas such as the axonal endings arriving in the basal ganglia's striatum from all parts of the cortex. The second possibility would be to introduce a short-term, reversible effect on the actual process of synaptic transmission, which regulates the amount of input arriving at a postsynaptic neuron through the synaptic connection when a presynaptic neuron fires. If a neuromodulator were to influence the short-term transmission efficacy of synaptic connections, it would temporarily be changing the *effective weight* of those synaptic connections for all intents and purposes.

We therefore modulate only synaptic transmission in our model, and show that we can still influence the learning outcome with a standard two-factor STDP rule. Without any direct reference to existing chemical neuromodulators, we coincidentally call our neuromodulatory reinforcement factor *DA*. However, simply multiplying the actual synaptic connection strength (henceforth called *baseline weight*) with the current DA level to form a temporary effective weight would be problematic: Whenever the applied reinforcement would reach a value around zero, all synaptic transmission would stop, which is unwanted

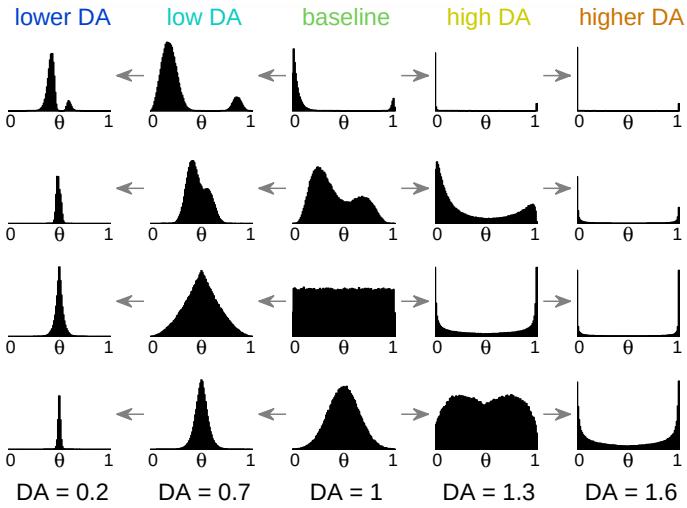


Figure 8.1: Effects of the proposed effective weight rule on four examples of actual (baseline) weight distributions. Example weight distributions at $DA = 1$ are given in the centre column (effective weights are equal to baseline weights when DA is at baseline). The threshold θ was chosen to be $\theta = 0.5$ in this figure, but additional values are shown in Supplementary Figure A.6 (p. 180). *Left:* Columns 1 and 2 show the resulting distributions of effective weights for low levels of DA. *Right:* Columns 4 and 5 show effective weight distributions for high levels of DA, in accordance with Eqs. 8.3 and 8.4. *Rows 1 and 2:* Any double-peak distribution where the two peaks are on opposite sides of θ will remain so for high DA, but with increased sparseness. For low DA levels, the two peaks move closer to θ, and thereby make discrimination of strong vs. weak weights increasingly difficult. *Rows 3 and 4:* A uniform or a normal distribution of weights around θ will act as a true bimodal distribution for high DA levels and become a thin unimodal distribution for low levels of DA. The reduced signal-to-noise ratio for low DA and increased ratio for high DA (assuming the signal is represented by the strong weights) becomes visible.

behaviour for a neuromodulated synapse. Instead, we use a power law relationship between baseline weights and the current level of DA to form the effective synaptic weights used for transmission.

We define a threshold value θ that divides the baseline weights into *strong* and *weak* weights (see Figure 8.1). For low DA, all effective weights are made to become temporarily more similar to θ than their baseline counterparts, while for high DA all effective weights temporarily move away from θ, towards the extremes of the defined weight range. Throughout this chapter, we therefore call θ the *generalisation threshold* for low levels of the reinforcement signal and the *sparsification threshold* for high levels. Strong baseline weights always produce effective weights that are above θ and weak baseline weights always produce effective weights below θ during modulated synaptic transmission. For stability, it is helpful to assume that the DA-dependent effective weights always remain within the same bounds as baseline weights, so our rule for computing effective weights from baseline weights and DA level ensures this (Figure 8.3, Eqs. 8.3 and 8.4). For a possible chemical interpretation of θ and of DA-dependent effective weights (used only for transmission), see Section 8.4 (*Conclusion*).

Synaptic plasticity (e.g. via STDP) is only ever applied to baseline weights, and the DA-dependent effective weights are then instantly updated accordingly.

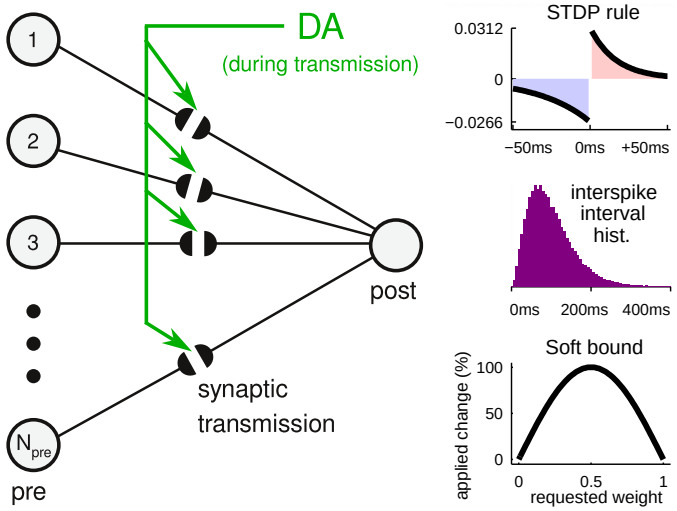


Figure 8.2: Network structure and dynamics overview. *Left:* Network structure. $N_{pre} = 1800$ presynaptic input units transmit spikes via DA-modulated synapses to one postsynaptic output neuron. The modulation influences the synaptic transmission process, and has no direct involvement in updating long term weights during STDP. *Right Top:* Asymmetric STDP rule. The integral over the full range of the curve is negative, so random firing normally leads to a slow decrease of weights. *Right Middle:* Interspike interval (ISI) of background noise. The mean ISI is 100 ms in a right-skewed gamma distribution, giving a mean background firing rate of 10 Hz. *Right Bottom:* Soft bound on weights, as described in Chapter 5, Section 5.2.6. The step size of STDP is greatest at medium weight values, and decreases towards the boundaries on each side. This also has the effect that synaptic connections are most volatile at medium weight values, while becoming more robust for more extreme weight values.

8.2 Model and Methods

8.2.1 Network Structure

The network structure is shown in Fig. 8.2 (left), where the two-layer network consists of $N_{pre} = 1800$ presynaptic units (inputs) that are fully connected to one postsynaptic model neuron (output) via DA-modulated synapses. The modulation instantly affects the excitability of each synapse, which can be perceived as a change in effective synaptic weight. The activity-dependent learning process (STDP rule) is not altered.

Neuron Model

We use the standard Izhikevich model in its one-dimensional form (Izhikevich, 2004) as postsynaptic neuron. This gives us the realism of a delayed, self-firing neuron while improving predictability and computational complexity for large-scale simulations. The predictability specifically benefits from the reduction to a one-dimensional model as the neuron's future activity is fully described by only its present membrane state and the present sum of arriving input currents. The neuron's membrane potential v is controlled by a variant of Eq. 2.3:

$$v' = 0.04v^2 + 5v + 140 - u + I \tag{8.1}$$

where u is fixed at its typical dimensionless starting value of -13, and I is the weighted sum of inputs arriving at the neuron. The amount of current arriving at the postsynaptic neuron was computed by multiplying each weighted input with $3000/N_{pre}$ to achieve some scalability to the number of input units.

Input Patterns and Background Noise

We use polychronous pattern type A (Section 7.1.1) to present inputs to the single postsynaptic neuron. Only a fraction of 200 input units is involved in presenting each of three patterns (Section 7.1.3). As each pattern is presented by a distinct set of input units, we can use one-sided patterns (compare Figure 6.26, Section 6.3). Pattern duration is 50 ms (see Figure 8.5a).

The input units here also insert random background spikes with an interspike interval (ISI) distribution that follows the gamma distribution with shape $k = 3$ and a mean firing rate of 10 Hz. As the background activity of an input unit is affected by any pattern it presents, the overall firing rate of the input layer is only slightly affected during pattern presentation, and remains in the range of random background variation (see Figure 8.5a, *input rate*).

Plasticity rule

Our weight update rule is based on that of spike-timing dependent plasticity (STDP) as used in (Masquelier et al., 2008). It is from the class of anti-symmetric rules where the sign of weight modification depends on the sequence in which the pre- and postsynaptic units fire. It is also a negatively biased rule (Chapter 5) in that the area-under-curve of the negative side (long term depression, LTD) is greater than that of the positive side (long term potentiation, LTP) for the full defined range. This property slowly decreases the overall strength of synaptic weights towards zero when input spikes occur at random times and the postsynaptic neuron is forced to keep firing. Alternatively, if the postsynaptic neuron is allowed to become quiet as all its input weights decline, this biased rule leads to a habituation to the input background activity, making the neuron highly reactive to any non-random time-structured inputs.

The negative integral of the STDP curve within [-50,50] ms does not prohibit the bounded integral from becoming positive within shorter ranges around ±5 ms, e.g. for very high firing rates or bursting, which could lead to a possibly unintended overall increase of synaptic weights. A possible solution for dealing with higher firing rates is proposed in (Pfister and Gerstner, 2006). We circumvent the problem by avoiding the occurrence of bursts through a reduction of the standard two-dimensional Izhikevich model (Izhikevich, 2003) to a one-dimensional model (Izhikevich, 2004).

The STDP rule used in this chapter is the same as was introduced in Chapter 2 (Eq. 2.13) and explored in Chapter 5 (Eq. 5.1):

$$\Delta w = \begin{cases} A_+ \cdot \lambda \cdot e^{\frac{\Delta t}{\tau_+}} \cdot g_+(w) & \text{for } t_{pre} < t_{post} \quad \text{(LTP)} \\ -A_- \cdot \lambda \cdot e^{-\frac{\Delta t}{\tau_-}} \cdot g_-(w) & \text{for } t_{pre} > t_{post} \quad \text{(LTD)} \end{cases} \tag{8.2}$$

where $A_+ = 1$ and $A_- = 0.85$ are positive scaling factors, $\tau_+ = 16.8$ ms and $\tau_- = 33.7$ ms are the exponential decay time constants, Δt is the difference between presynaptic (t_{pre}) and postsynaptic (t_{post}) spike arrival times at a synapse ($\Delta t = t_{post} - t_{pre}$), $\lambda = \frac{1}{32}$ is a constant that controls the learning rate (see Table A.1), and $g(w)$ is a dynamic weight-dependent scaling parameter that is defined below.

Useful features of this STDP rule include (Chapters 5 and 6):

- **Causal firing (pre, then post)** leads to fast potentiation.
- **Anti-causal firing (post, then pre)** leads to fast depression.
- **Acausal (random) firing** leads to slow depression because the integral of the STDP curve is negative.

We apply STDP in an all-to-all pairwise matching scheme. To keep all weights within an interval of [0,1], we apply the following soft bound on the weight change.

Weight Bounding

We use an attractor-less weight-dependent update scaling rule from Chapter 5 for $g_+(w)$ and $g_-(w)$. This reduces the step size of synaptic updates as the synaptic strengths becomes close to the lower ($w = 0$) or upper ($w = 1$) bound of the defined weight range. We choose the shifted cosine window (Eq. 5.9, p. 61) as the mapping function for this chapter, as it gives us the characteristics of a wide range of applied change around medium weight values and reduced change as weights come closer to their extremes (see Figure 8.2 *bottom right*). Different soft bound kernels with steeper slopes may be also used in future work.

8.2.2 Modulation Mechanics

We propose the existence of a neuromodulator which directly affects the process of synaptic transmission. Depending on the ratio of enabling receptors (henceforth called *D1-type*) to attenuating receptors (henceforth called *D2-type*) in a neuron's synapse, an increase of neuromodulator above baseline levels may sparsify synaptic transmission by further easing signal transmission through strong synapses and hindering transmission through weak synapses. Similarly, we suggest that declining amounts of neuromodulator in the surrounding tissue may have a generalising effect on synaptic transmission, where the efficacy of strong and weak synapses becomes more equal around some threshold ratio θ.

We further assume that the threshold θ at which the sparsification bifurcates, and to which the generalisation tends, may slowly be regulated by homeostatic (chemical) gradients within the cell. This will be explored in future chapters.

Here, we examine the implications of our proposition, and show that synaptic learning can be reliably modulated by only the given mechanisms. No direct influence in the actual spike timing dependent plasticity process is needed for modulation to succeed.

Affecting Synaptic Excitability

We define a modulatory parameter DA that contains our reinforcement information within the range $[0, 2]$, where the value 1 stands for *no specific feedback*. If it were to be mapped to the activity of dopaminergic cells in the Substantia Nigra pars compacta (SNc), $DA = 1$ would be equivalent to normal tonic firing and default levels of dopamine released into the striatum.

We simulate DA-dependent changes in perceived synaptic efficacy as changes to *effective weights* e_{ij} used for computation of synaptic transmission, in distinction from the default efficacy of synapses at baseline levels of the neuromodulator ($DA = 1$), which we call *baseline weights* w_{ij}. Assuming a range of synaptic weights where $w_{min} = 0$ and $w_{max} = 1$, we set

$$e_{ij} = \begin{cases} \theta \left(\frac{w_{ij}}{\theta}\right)^{\xi} & w_{ij} \leq \theta \\ 1 - (1-\theta)\left(\frac{1-w_{ij}}{1-\theta}\right)^{\xi} & w_{ij} > \theta \end{cases} \tag{8.3}$$

and

$$\xi = 2^{r(DA-1)} \tag{8.4}$$

where $\theta \in [0, 1]$ is the above-mentioned threshold for weight sparsification and generalisation, $DA \in [0, 2]$ is the level of dopamine currently applied to the network, and r is a range parameter for controlling the impact on sparsification or generalisation the neuromodulator can have (see Figure 8.3). For simplicity we assume $r = 5$ and $\theta = 0.5$ within most of this chapter. Slow adaptation of θ will be explored in future chapters. The double power law relationship between the baseline weight and the current DA level (introduced

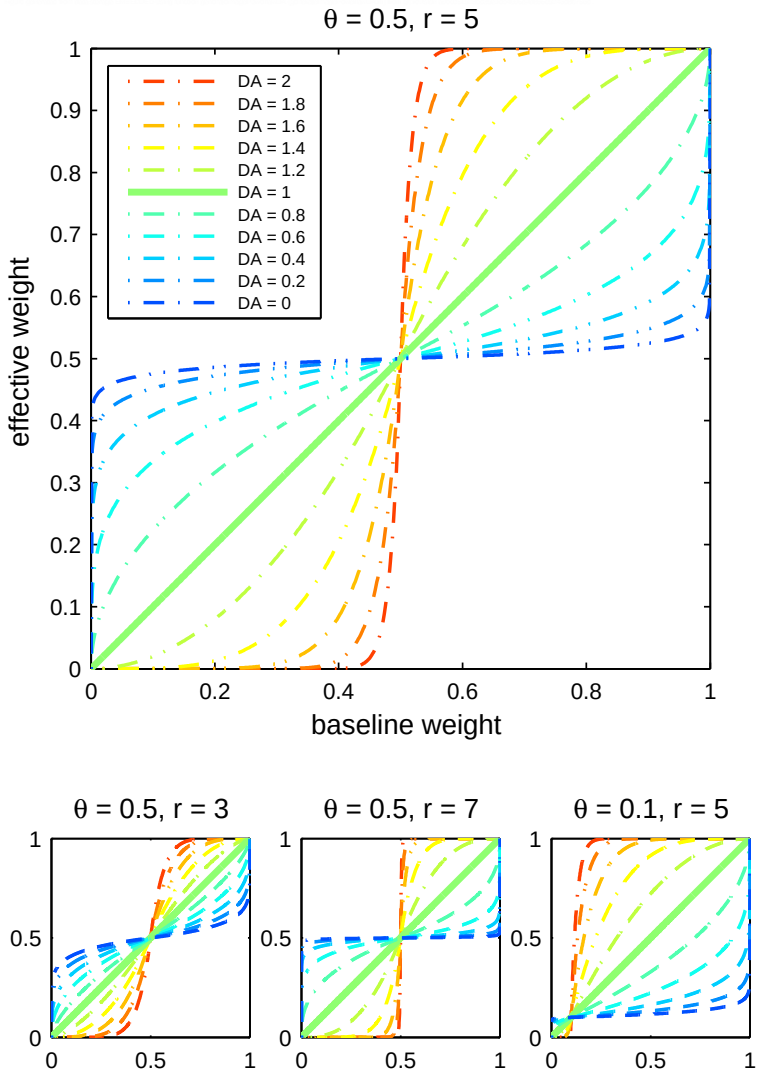


Figure 8.3: Proposed effects of a reinforcement signal DA on synaptic transmission, perceived as a DA-dependent synapse-local change of *effective weights*. As in biological dopaminergic systems, a tonic base line of the reinforcement signal ($DA = 1$) exists, which here keeps the effective weights equal to their actual baseline values. The threshold for defining strong and weak weights can be changed by varying θ.

by combining Eq. 8.3 and Eq. 8.4) allows a bijective projection between actual (=baseline) weights and effective weights, and makes the curves for high DA levels mirror those of low DA levels across the $DA = 1$ line. As this study aims to predict biological mechanisms, we should not restrict our nonlinearity too much yet. For exploring the concept, Eq. 8.3 and 8.4 merely aim to keep the range of effective weights equal to that of baseline weights, and provide some basic symmetry.

Effective Weight Distribution

The formulation of *effective weights* allows us to compare any instant changes in the perceived distribution of weights that are due to changes in neuromodulator level. In Figure 8.1 (and supplementary Figure A.6) we show a selection of baseline weight distributions in a centre column, and their DA-induced changes as effective weight distributions in the other columns. Synaptic transmission is computed using the current effective weights, while any STDP-induced weight change is applied to the baseline weights. The effective weights are updated on any change of baseline weights or DA level.

As the neurotransmitter level is increased, any broad distribution of weights becomes more bimodal, away from the sparsification threshold θ. Slightly stronger synapses (weights above the threshold) thereby become dominant in guiding postsynaptic activation, while connections with weights even slightly below the threshold loose influence on postsynaptic activation. A slightly trained network therefore acts as if it has undergone more training and acts more selective to a smaller number of inputs. Any overrepresentation of strong effective weights that would lead to excessive postsynaptic firing is then gradually reduced by our negatively biased STDP rule towards sparse coincidental firing, given random uncorrelated inputs (see Chapter 6). This competition reduces the number of strong synapses and readies the neuron for detecting more structured, non-random inputs by adapting it to mostly ignore random background input activity.

As the neuromodulator level is decreased, any distribution of effective weights becomes more centred around the generalisation threshold θ, leading towards an equalisation of effective weights. The effect of each synaptic connection on membrane activity of the postsynaptic neuron becomes less dependent on the actual (baseline) synaptic strength. Instead, all connections start to behave increasingly similar in transmission efficacy, making it harder for the neuron to discriminate strong learnt inputs from ignorable background activity. This amplified noise level leads to frequent random weight adjustments, causing existing baseline weights to perform a semi-random walk. This randomisation process causes the baseline weights to become less sparse, while their mean is also reduced due to the negative drift of the STDP rule used (Section 5.1). Even a strongly trained neuron can thereby be "reset" to a general state with unimodal distribution of weights if the decrease in DA and range parameter r are large enough and the postsynaptic neuron is kept firing. As most weights act as being close to θ for very low amounts of neuromodulator, the definition of θ directly affects the output activity of the postsynaptic neuron: A high value of θ leads to infinite firing of the postsynaptic neuron, while a low value of θ may make the postsynaptic neuron silent as soon as the strongest connections cease to be able to provoke postsynaptic firing. A slow adaptation of θ as a local variable within the postsynaptic neuron may therefore prove useful for continued activity, which may or may not be implemented as a slow chemical gradient for homeostasis.

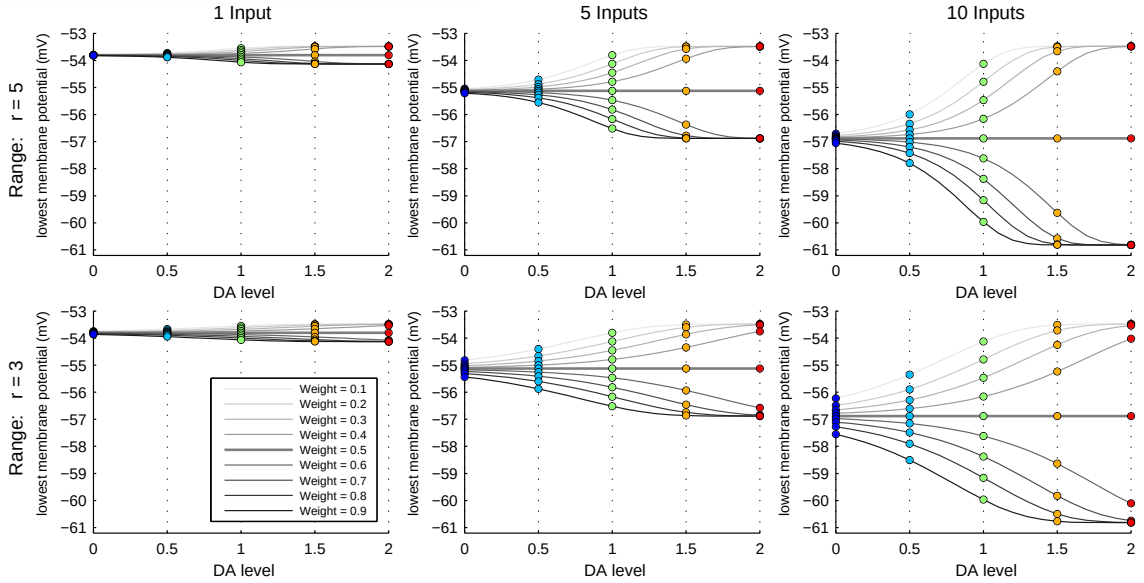


Figure 8.4: Effects of various DA levels $\in [0, 2]$ on firing onset membrane potentials of a 1D-Izhikevich neuron (Izhikevich, 2004). At baseline level (DA = 1) the onset potentials seem evenly distributed across a voltage range defined by the amount of neural input (green dots). Lower DA makes the onset potentials become less dependent on the actual weight of the input synapses and instead approach a central mean value dependent on θ as DA goes towards 0 (blue dots). As the activity of the postsynaptic neuron now depends less on the actual weights but mostly on the overall input to the network, we can argue that the firing pattern of the postsynaptic neuron becomes *less causal* (less dependent on specific inputs) compared to baseline DA levels. In the opposite case of high DA ($DA \to 2$), the effect of the weights also changes. Increasing levels of DA make inputs arriving through weak connections have an even smaller effect on the postsynaptic neuron's activity (The upper membrane boundary seen near -53.5 mV in the figures is the neuron's onset potential in absence of any inputs). The influence of inputs arriving through already stronger connections is increased up to a maximal effect when effective weights are near the maximum value of 1 (The lower membrane boundary seen in the figures is the neuron's lowest onset potential for the given number of inputs). The effect of partially trained synapses is thereby enhanced, up to a binary effect strongly depending on the synaptic strength.

8.3 Results

We performed two stages of tests with our proposed new method of modulation. During the first stage, we examined the direct effect on postsynaptic activity for a typical distribution of fixed weights, while on-line synaptic modification using spike timing dependent plasticity was incorporated during the second stage of tests.

8.3.1 Instant Effects

A central feature of our method is its instant effect on the activity of the postsynaptic cell. All later differences in learning are guided only by this alteration of postsynaptic spiking activity. No modulation whatsoever takes place within the STDP rule itself. The fact that an influence in synaptic learning processes can still be observed in our simulations points to the high importance of how exactly these instant effects in synaptic transmission change the postsynaptic neuron's instant response.

Firing Tendency

In the first test, we examined the amount of input needed to produce a spike response from the one-dimensional Izhikevich neuron. As its recent history can be summed up in

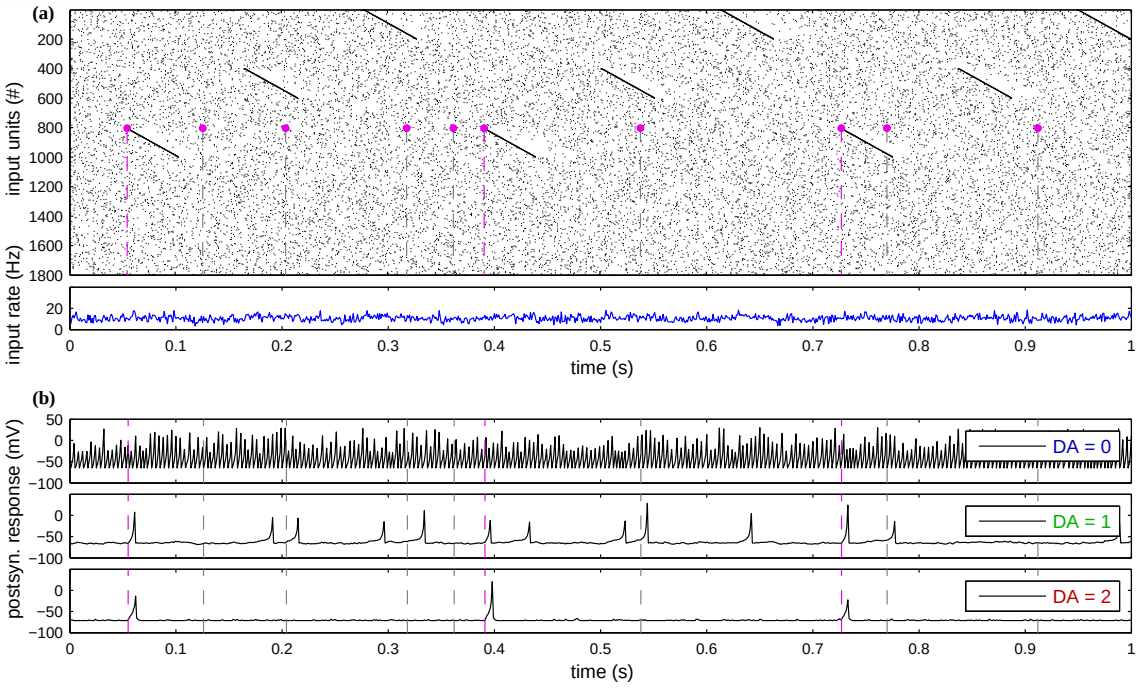


Figure 8.5: Typical response snapshot for a fixed-weights trial. (a): Input patterns created by presynaptic input units (see *Model and Methods*). Pattern 1 is repeatedly presented by units 001-200. Pattern 2 by units 401-600. Pattern 3 by units 801-1000. The first 100 units that present pattern 3 are connected to the postsynaptic neuron through strong weights around 0.7 while all other neurons are connected through weak weights around 0.1 (compare Figure 8.6a). The purple dots represent the firing activity of input unit 804 (used in Figure 8.6d as Unit 3), with striped lines representing the times of a spike for comparison with the postsynaptic response. Purple lines signal a spike within pattern presentation, and grey lines signal spiking due to random background activity. The instantaneous firing rate of the input layer is shown below. Only little variation in presynaptic rate is discernible. (b): The membrane response of the postsynaptic neuron for different levels of DA with $\theta = 0.5$. The full data from which this image is a snapshot was used for the results shown in Figure 8.6.

the model neuron's current membrane state, we can ask the question differently: At which preset level of membrane depolarisation does the model neuron still fire, given a specific number of synchronous unit inputs?

The answer is plotted in Figure 8.4 for 1, 5 and 10 synchronously arriving inputs, and for range parameters $r = 5$ and $r = 3$. At a membrane potential above about -53.5 mV, the model neuron will fire even without inputs. This upper bound is approached when all actual weights are far below θ and the level of neurotransmitter is increased above baseline ($DA > 1$). Analogously, when all weights are above θ while neurotransmitter level is increased, the neuron's membrane threshold before inputs can be more negative as any inputs are fully transmitted to the postsynaptic neuron.

For decreasing levels of neurotransmitter ($DA < 1$), synaptic transmission always approaches that of weights around the current value of θ. The effect of this may be imagined as a neuron-level reduction in signal to noise ratio, as the effect of strong synapses (possibly having learnt structured patterns) is decreased while that of weak synapses (possibly having learnt to ignore background activity) is increased. This tendency is shown in section 8.3.2 .

The amount of synaptic transmission depends less on the actual baseline weight of a synapse as the level of neurotransmitter moves away from baseline. The range of transmission effects is evenly distributed for $DA = 1$ (green dots in each plot), while decreasing

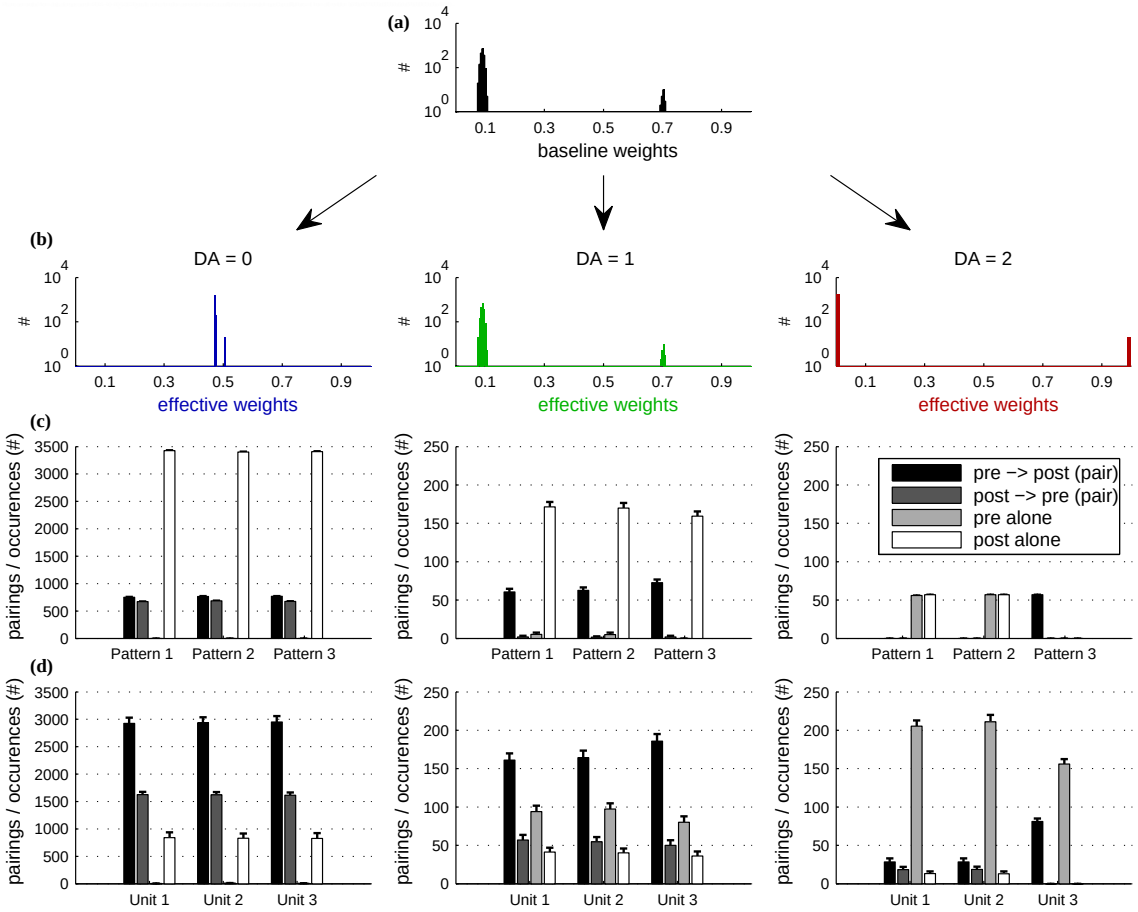


Figure 8.6: Effective weight distributions and resulting changes in relative spike event pairings. As any deviation of DA from 1 temporarily alters the effective weight of each synapse in our model, we show the effective distribution of weights for three levels of DA above. (a): Actual (baseline) weight distribution used for this test. (b): DA-dependent effective weight distributions for DA levels 0, 1, and 2. As all firing of the postsynaptic neuron is caused only by inputs from the input layer, the relative amount of spike pairs gives a hint at the causality relationship between pre- and postsynaptic events. Causal or anti-causal event pairings are counted if a presynaptic and a postsynaptic event occur within 100 ms of each other. If two events occur with longer time difference, both are counted as single presynaptic and single postsynaptic events. (c): Comparison of event times where the presynaptic event is the presentation time of each pattern, and the postsynaptic event is the time of each spike of the postsynaptic neuron. (d): Comparison of event times where the presynaptic event is the time of spike of input unit 004 for pattern 1, 404 for pattern 2, and 804 for pattern 3.

levels of DA make the transmission effect become solely dependent on θ (blue dots) and increasing levels make the effect go towards that of weights in the extremes of 0 and 1 (red dots).

Causal Postsynaptic Response

Due to its temporary changes in effective weight distribution, our proposed rule for DA-dependent modulation of synaptic efficacy leads to a change in causal relationships between presynaptic and postsynaptic activity.

Figure 8.5a shows a snapshot of typical input data generated online as described in *Model and Methods*, together with the instantaneous firing rate in 1ms bins. The purple dots exemplify the spiking behaviour of one input unit that happens to take part both in the random background activity and in representing the partially trained time-structured

input pattern 3.

The inputs are projected via DA-modulated synapses with fixed, partially trained weights to the postsynaptic model neuron, evoking an output response that strongly depends on the current neuromodulator level (Figure 8.5b). For visual clarity, we again used $\theta = 0.5$ during this test, which has the side effect that the postsynaptic firing rate is increased for low values of DA and decreased for high values of DA for any typical (right-skewed) weight distribution. However, apart from the changes in firing rate, the important difference between the three response plots is the increasing selectivity of action potentials on presentation of the partially trained pattern as the neuromodulator level increases. While near-random firing is observed for low values of DA, the output behaviour becomes more sparse at $DA = 1$, with no misses but some false positives in detecting pattern 3. The detection of the pattern becomes perfect for the highest level of $DA = 2$ in this case, as the postsynaptic neuron now fires if and only if pattern 3 is presented.

To make a statement on the general applicability of this observation, we chose a fixed distribution of weights as shown in Fig. 8.6a and simulated the response of a postsynaptic neuron for 20 s on each of three different levels of DA, repeated 100 times for each DA level. The vast majority of 1800 synapses had random weights around 0.1, while 100 connections to units coding pattern 3 were given weights around 0.7 (units 801-900). By repeatedly counting the number of occurrences of single events, of causal pairs, and anti-causal pairs, we can examine the relative change in selectivity for the values $DA = 0$, $DA = 1$, $DA = 2$. Events in this context were either a spike of a presynaptic input unit taking part in coding the beginning of a pattern, the presentation of the patterns themselves, or a spike of the postsynaptic neuron. Event pairs were either causal (pre-synaptic spike followed by post-synaptic spike within 50 ms) or anti-causal (post-synaptic spike followed by pre-synaptic spike within 50 ms).

Each bar plot in Figure 8.6c shows groups of causal and anti-causal event pairs and single events where a presynaptic event is the respective onset time of each of the three time-structured input patterns. A postsynaptic event is a spike of the postsynaptic neuron. While the distribution of events is similar for all three patterns on $DA = 0$, there is a slight increase in causal pairs and a slight decrease in single postsynaptic firing for $DA = 1$ on presentation of pattern 3 compared to presentation of the other patterns. The difference becomes obvious for $DA = 2$, as pattern 3 reliably and perfectly provokes a postsynaptic spike on each presentation, with no false positives or misses. The equally high white single event bars for patterns 1 and 2 represent the same postsynaptic activity that was counted as part of the causal pair for pattern 3, except that here it represents a single event, unrelated to neither pattern 1 nor 2.

A more noisy result is seen when comparing not the onset of pattern presentation to postsynaptic firing, but the spiking activity of a presynaptic unit that happens to take part in presenting the pattern. Figure 8.6d shows the results of this comparison, where the same tendency can be observed: The response to all three patterns seems highly similar for the lowest level of DA, while a slight difference is seen for normal neuromodulator levels. Again, a strong change in response to the unit presenting pattern 3 is observed when the neuromodulator level reaches $DA = 2$.

The differences in total number of events for different levels of DA are again due to the chosen value of $\theta = 0.5$, which increases the effective weight of the majority of synapses for low levels of neuromodulator. In a (biologically less plausible) left-skewed weight distribution with the majority of weights above θ, the opposite effect on firing rate would be observed. Automatically keeping θ within a homeostatically plausible range is therefore an important topic for widespread applicability in large scale multi-layer networks.

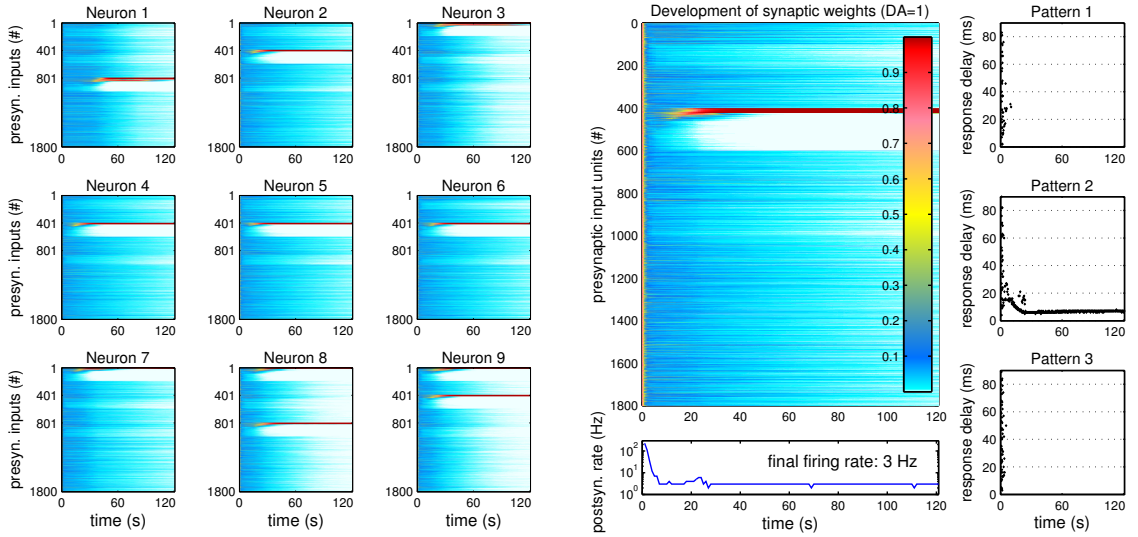


Figure 8.7: Development of baseline weights: Unsupervised learning of structured input patterns by 10 independent postsynaptic neurons at baseline DA levels ($DA = 1$). Given equal inputs (see Fig. 8.5a) and a narrow range of starting weights, the 10 neurons tune to different patterns. The slight preference for choosing pattern 2 here comes from the coincident timing of the equal background inputs, and different random background inputs lead to a different pattern preference. Here, all connections start with strong weights around 0.8, leading to an initial overall decrease due to high postsynaptic activity and the asymmetry of the STDP rule. Then, as only the recurring polychronous inputs repeatedly cause the postsynaptic neuron to fire, the weights of the connections to input units reliably firing just before postsynaptic activation begin to be strengthened. As the now stronger weights (shown in red) lead to an earlier onset of firing of the postsynaptic neuron relative to pattern presentation times, connections to even earlier firing input units are strengthened. The earliest firing units of a repeating pattern soon form the strongest connection, as seen by the rise of red lines in the weight development plots (see also supplementary Figure A.1). Also, connections to input units representing a late part of a pattern are now weakened, because they repeatedly fire *after* a postsynaptic spike. *Far Right:* Input response delay plots for each of the three patterns show an initially decreasing and then constant delay of the postsynaptic neuron's response to learnt pattern 2, and an extinction of responses to patterns 1 and 3, to which the neuron did not tune. *Bottom Right:* Instantaneous firing rate of postsynaptic model neuron.

8.3.2 Effects with Synaptic Plasticity

After testing our proposed neuromodulation approach on fixed-weight networks, we now examine the modulatory effects of our transmission rule on freely acting synaptic plasticity. As described in *Model and Methods*, no modulation whatsoever is factored directly into the STDP rule we use. The only adjustment in how STDP-induced synaptic plasticity is converted to actual weight changes is the soft bound to keep all weights within the interval $[0, 1]$. After examining unsupervised behaviour of STDP at baseline levels of DA, we test reinforced learning with fixed, non-baseline levels of neuromodulator and examine the effects of sudden DA level changes on synaptic plasticity characteristics.

Fast variation of neuromodulator gradients for large-scale reinforcement learning will need automatic adjustment of θ (see Chapters 9 and 10).

Learning with Baseline Modulator Levels

When the level of neuromodulator remains around the baseline of $DA = 1$, the network performs unsupervised learning, depending only on the structure of arriving inputs. Figure 8.7 shows a test where 10 independent postsynaptic neurons were trained in parallel to the same inputs. Each neuron's weights were initialised randomly around 0.8 within the range

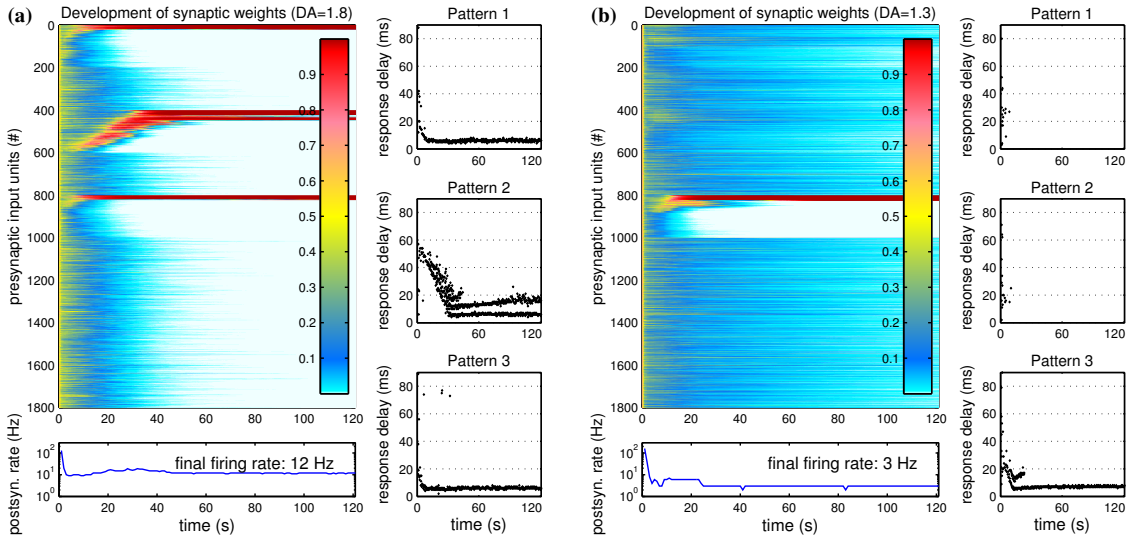


Figure 8.8: Modulated learning of polychronous input patterns by each of two independent postsynaptic neurons (a,b) at high levels of DA. Each postsynaptic neuron is more likely to tune to any of the patterns, where just slightly increased baseline weights act as high effective weights, enabling further strengthening of those connections. *(a):* For $DA = 1.8$, this neuron quickly tunes to all three patterns, but initially only with a late response to presentations of pattern 2. Because of natural STDP behaviour (see Chapter 6), the neuron slowly re-tunes to input units representing the start of pattern 2, while connections to late-firing input units in pattern 2 are weakened. The shortest response delay for pattern 2 is reached after about 30s of simulation, with a seemingly stable double spike response to pattern 2 presentations. *(b):* For $DA = 1.3$, this neuron happens to only tune to one input pattern, but the effect of high DA levels on weights to background inputs is nicely visible (compare Figure 8.7): While the weights to input units taking part in pattern 3 quickly go towards either 0 (blue) or 1 (red), all other weights are only slowly weakened when background activity happens to coincide with postsynaptic firing. Although the baseline weights to background inputs are still around 0.1 and would usually induce postsynaptic firing for normal DA levels, the DA-dependent effective weights to input units that present only background spikes have become low enough to have no chance in activating the postsynaptic neuron.

[-0.025,0.025] in a uniform distribution. No random growth (Section 5.3.1) was used in this chapter. Spike timing dependent plasticity was allowed to change the weights of synaptic connections, but no modulation signal was given ($DA = 1$). At the start of the simulation, the postsynaptic neurons begin to fire excessively for a short time due to the high mean of inputs arriving at each simulation step (thin red vertical line at t=0 in each plot). This is then reduced by the negatively biased all-to-all STDP rule which adapts the weights to account for the random background activity arriving through the input units (light blue). Without structured patterns occurring within the input stream, all postsynaptic firing would stop at this point (data not shown). However, after about 3 s, the first postsynaptic neurons begin to increase the weights of synaptic connections to pattern presenting input units. After usually no more than 10 s, all postsynaptic neurons have tuned to at least one structured pattern (yellow to orange), and start developing strong connections to the first input units of each pattern (dark red). Shortly after this, synaptic weights to any input units that fire repeatedly at a later stage in pattern presentation are reduced to near zero (bright cyan). This fast LTD is due to the repeated (anti-causal) post-pre pairing of spikes in opposition to the slower LTD induced by uncorrelated background activity.

The decision which of the patterns is learnt depends here both on the random starting distribution of weights and on coincidental peaks in the background activity (noise). Because of the soft bound on weights we use, narrow initialisation ranges near the extremes can have a similar exploratory effect on tuning preference as a wide initialisation range

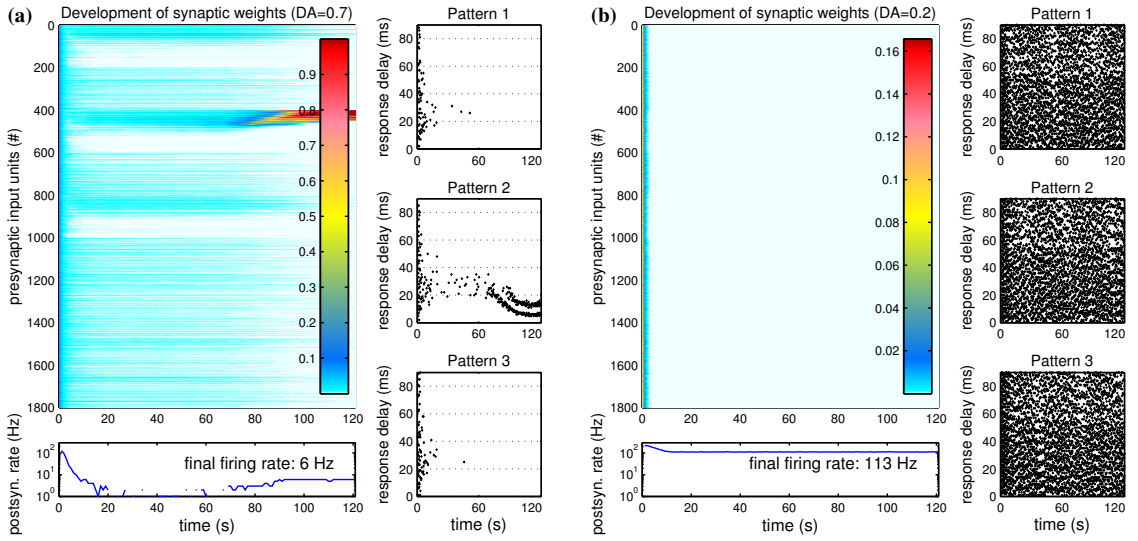


Figure 8.9: Modulated learning of polychronous input patterns by two postsynaptic neurons at low levels of DA. Each postsynaptic neuron is less likely to tune to patterns, as any initial increase in baseline weights is masked by the high similarity of all effective weights. Any polychronous inputs are increasingly difficult to discriminate from background activity as the DA level decreases. *(a):* For $DA = 0.7$, this neuron does finally manage to reliably tune to pattern 2 after about 70 s, and even forms a double spike response shortly before 90 s of simulation have passed. Note that here we were able to start with very low baseline weights around $w = 0.1$, because the high value of $\theta = 0.5$ keeps the initial effective weights high enough to produce a postsynaptic response. *(b):* For $DA = 0.2$, no more learning is possible. Most effective weights come very close to θ, completely blocking out any baseline weight variation. In this case, as θ is fixed at $\theta = 0.5$, the DA-dependent grouping of effective weights around this value also leads to continuous, pathological firing of the postsynaptic neuron. While this high activity could be reduced by (automatically) lowering θ, the masking of trained vs. untrained connections can not.

has around the centre weight value of 0.5.

In absence of dopamine or other strong modulatory factors, previous approaches stopped all form of learning (Izhikevich, 2007b). In our proposed method, learning simply switches from reinforced to unsupervised learning when the modulatory signal remains fixed at baseline level.

Learning with Non-Baseline Modulator Levels

We now add some permanent reinforcement into the simulation by changing the applied level of neuromodulator. Figures 8.8 and 8.9 show the typical development of weights for $DA = 1.8$, $DA = 1.3$, $DA = 0.7$ and $DA = 0.2$.

For $DA = 1.8$ (Figure 8.8a), the postsynaptic neuron has the highest tendency to quickly tune to multiple (non-overlapping) polychronous input patterns. The probability of tuning to new patterns is highest during the first few seconds of simulation and diminishes in absence of any homeostatic weight adjustment due to random background activity while simulation progresses. Here, the postsynaptic neuron starts responding to all three input patterns quickly, but initially has a high response delay (~ 50 ms) when detecting pattern 2 because it happens to initially tune to late input units of this pattern. It then slowly re-tunes to the first input units repeatedly firing within pattern 2. As the postsynaptic neuron continues to tune to the first spiking input units of the pattern, connections to units representing late parts of the pattern are again actively decreased, as indicated by the gradual upwards shift of the middle red line.

For $DA = 1.3$ (Figure 8.8b), the postsynaptic neuron tends to tune to less input pat-

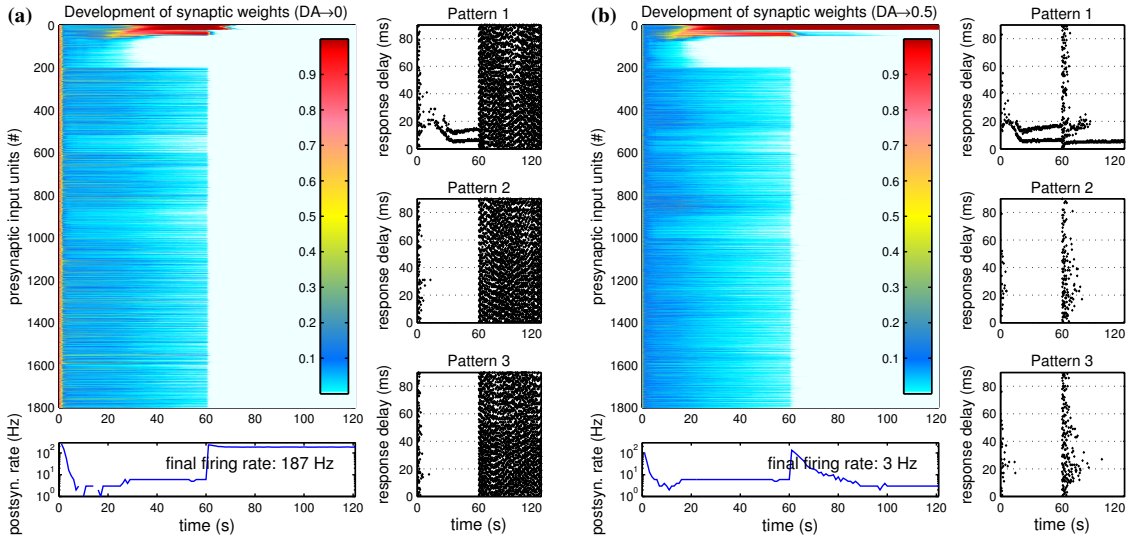


Figure 8.10: Modulated learning behaviour for baseline levels of DA, dropping to low levels after 60 s. During the first 60 s, the weight development is similar to Fig. 8.7, and both (a) and (b) happen to tune to pattern 1 with a double spike response. *(a):* When DA drops to zero, the neuron instantly starts firing quickly as all effective weights move close to $\theta = 0.5$. The effect on trained weights is slower, as it takes about 3 s after DA drop for the first group of weights to be decreased. After about 8 s, all weights have been decreased to values close to zero, as the high spiking activity of the postsynaptic neuron is continued. The drop of DA hereby led to a deletion of trained weights, or *unlearning* of previously learnt patterns. If weights of this neuron were randomly increased in the future, it would be ready to learn completely new patterns without relation to its previous identity. *(b):* When DA only drops to 0.5, the neuron still instantly increases its firing rate, but manages to recover by further reduction of synaptic weights to background inputs. The group of weights causing the second spike response on each presentation of pattern 1 is reduced towards zero during the initial phase of high postsynaptic firing. But the group of weights causing the repeating first spike response to pattern 1 survives here. After the neuron recovers to normal firing, the first-spike response to pattern 1 is still intact. A sudden decrease of DA may therefore be useful as a pruning measure to sparsify trained neural responses.

terns. Once it has started to fire regularly to one of the patterns, the random background activity continues to diminish all weights to other (randomly active) input units not taking part in the tuned pattern. As the weights to units taking part in other patterns are hereby also slowly reduced, the postsynaptic neuron slowly looses its ability to further tune to more patterns and remains highly specialised. The reduction of weights to background inputs is slower than in Figure 8.7, as the increased DA level here decreases transmission by below-θ (weak) weights earlier and the lower resulting firing rate of the postsynaptic neuron produces slower LTD on background activity.

For $DA = 0.7$ (Figure 8.9a), we still see the neuron tune to one of the structured patterns, albeit only after a long time of uncertainty (here $\sim$ 75s). Coincidentally, it also repeatedly fires twice on each pattern presentation for the remaining duration of the test. Note that in this test we were able to start with a very low initial range of baseline weights, because our low DA level lets the effective weights act as being closer to our generalisation threshold $\theta = 0.5$.

For $DA = 0.2$ (Figure 8.9b), no more tuning is observed, and the postsynaptic neuron reaches a pathological state of relentless firing. While this high postsynaptic activity could be controlled by lowering θ, the failure in tuning can not be compensated as DA goes towards zero. The structured inputs vanish in the random background activity that is transmitted to the neuron with equal efficacy. From the neuron's perspective, the signal-to-noise ratio between structured and random inputs is strongly reduced and can no more

be used for successful learning.

In the next test we reduce the neuromodulator level suddenly after 60 s of simulation time. Figure 8.10a shows results of a trial that starts with $DA = 1$ and drops to $DA = 0$ after some initial training has occurred. While the neuron now instantly enters the pathological state of excessive firing due to the high θ value (all effective weights approach $e_{ij} = 0.5$ because $\theta = 0.5$), a delayed influence on the learnt weights becomes visible. The response delay plot for pattern 1 shows a repeating two-spike response before the DA level drop. This is due to strong learnt connections to input units $\sim$ 1-50 as indicated by the wide red bar at the top of the main plot. About 2-3 s after DA drop, the weights to input units 27-47 are quickly reduced as they move away from the maximum value defined by the soft bound. In a normally firing neuron, this would remove the second response spike to pattern 1, and may be used for pruning a neuron's response. At about 10-12 s after DA drop, the last existing strong weights (1-23) break down and all weights of the neuron go towards zero. This complete formatting of weights resets the neuron into an unspecialised state. For allowing the neuron to tune to new patterns, some homeostatic form of re-enabling spiking activity would need to be added to the neuron. This may be a combination of either random weight growth or automatic adjustment of θ together with low DA levels. Moderate baseline weight increase can then allow new tuning to correlated inputs as used in Figure 8.9a.

Figure 8.10b shows a reduction to $DA = 0.5$ after 60 s, which again initially leads to fast postsynaptic firing. However, in this case the neuron is able to recover normal operation after a few seconds by further reducing weak connections to a level low enough to not be pulled up to high effective weights near θ by the given DA level anymore. This sudden reduction of DA still allows pruning of double spiking to take place, but preserves the single-spike response to trained patterns 1 and 3.

Synaptic Competition

With the described effects of our proposed rule for DA-dependent signal transmission, we can affect the network learning process without directly changing the STDP rule. While we can push the network or single neuron into sparsely fitting an active input signal for high levels of DA, we can induce a randomisation process through low levels of DA, thereby resetting the neuron into a less selective state, or "forgetting" the learnt patterns. Dopamine, or any combination of neurotransmitters signalling reinforcement, can in this way be simulated to either increase the probability of learning a given input pattern or to reduce the probability of learning and even forgetting learnt weights to active inputs. The inputs must be active at least occasionally in order for any change to occur, so completely silent inputs would always remain unchanged.

Apart from fast LTP and LTD through repetitive causal or anti-causal event pairs, we induce slow LTD by taking advantage of the fact that random firing causes weight decrease for negatively biased STDP rules. So by equalising the effective synaptic weights towards θ on low DA, we are allowing the random background activity to induce slow LTD.

If the inputs through strong synapses are strong enough to produce spikes in the postsynaptic neuron and repeatedly happen in close temporal proximity as is the case for polychronous patterns, the causal relation between the repeated presentation of inputs and postsynaptic firing leads to a further strengthening of these weights (see also Masquelier et al., 2008). If, on the other hand, the number of strong weights is high and the presynaptic neurons fire mostly independently, causing the postsynaptic neuron to fire at random, the noisy input leads to an overall decrease of even these strong weights as synapses compete for control over postsynaptic activation. The network behaviour for

high levels of DA is therefore the following: If many synapses are strong at the beginning of DA application, an overall reduction of weights takes place, until most synaptic weights have passed the threshold θ and are effectively close to zero. When only a small number of strong weights remain, the competition between synapses for control over postsynaptic firing that caused the overall decrease is weaker, which allows a small number of weights to remain strong and even be reinforced again up to maximal selectivity when some inputs are correlated in time. This allows a sparse distribution of synaptic weights to develop.

8.4 Conclusion

In this chapter, I demonstrated how we can influence the learning outcome of a spiking network simply by applying some global reinforcement during synaptic transmission. All synaptic modification is only dependent on the pre- and postsynaptic spiking activity, and no third (modulatory) factor is used during spike-timing dependent plasticity. Through controlling a global level of neuromodulator concentration, we are able to influence the effective range of synaptic efficacies, and thereby the discriminability of trained vs. untrained inputs arriving at a postsynaptic neuron. This change in synaptic efficacy is computed locally in each synapse, using only the current synaptic strength and the current global neuromodulator concentration.

A variable neuron-wide threshold θ will be used in the next chapters as a homeostatic slow parameter that automatically updates to retain normal excitability on varying neuromodulator levels for non-uniform weight distributions. The size of θ would likely come to be far below 0.5 in an automatically adopting implementation.

Applying modulation by locally affecting synaptic transmission instead of direct manipulation of the STDP rule gives the advantage of direct control over the causal firing relationship between selected presynaptic and postsynaptic neurons, which can instantly be observed as the modulation factor changes. In terms of network learning, the reinforcement signal does not directly increase or decrease active synapses, but instead leads to a temporary sparsification of effective weights for high reinforcement and a generalisation around θ for low reinforcement.

As the modulatory factor needs to be present during the arrival of inputs, we do not approach the *distal reward problem* (Izhikevich, 2007b) through our model, but assume for the case of delayed reward an involvement of hippocampus and cortical working memory instead of direct application of delayed reward into an STDP rule. Instead, we hope to provide a possible explanation for experimentally observed (Kroener et al., 2009; Thurley et al., 2008) instantaneous effects during neuromodulator application. Assuming the process of novelty detection by subcortical sensory nuclei performs faster than or equally fast as the semantic processing of some signals in the cortex (Trimmer et al., 2008), our model may also be useful for learning the short-latency novelty portion (Redgrave and Gurney, 2006) of the nigral reinforcement signal (Schultz et al., 1997).

Although we have until now only been studying pairwise rules of STDP, there is no reason to assume that the proposed modulation rule should not be combinable with STDP learning based on triplets of spikes (Pfister and Gerstner, 2006) or voltages (Clopath et al., 2010). Specific examination of this combination is not focus of the current work.

Our model presents some interesting questions for biological validation: It is currently unclear how exactly dopamine affects signal transmission locally at single synapses. Little is known about the exact local concentrations of dopaminergic receptors across a neuron's membrane (Reynolds and Wickens, 2002; Shen et al., 2008; Surmeier et al., 2007). Also, it might be useful to look for a biological analogy to our theoretical sparsification and

generalisation threshold θ, as this may explain many of the observed instant effects of dopamine or related substances. Heteromeric co-expression of D1+D2 receptors comes to mind (Aizman et al., 2000; Fauchey et al., 2000; Hasbi et al., 2009; Lee et al., 2006, 2004b; Perreault et al., 2012, 2010; Rashid et al., 2007; So et al., 2005; Thompson et al., 2010).

A chemical prediction by our proposed rule may start at the ratio between D1-type and D2-type receptors on a dopamine-modulated synapse of a D1-dominant postsynaptic neuron. While a neuron-wide baseline ratio of dopaminergic receptors may represent a homeostatic default configuration similar to θ in our model, any strengthening synaptic connection may be found to also increase the local concentration of D1-type receptors towards a higher excitability on raised levels of dopamine. Similarly, a weakening synaptic connection may reduce the local concentration of D1-type receptors, allowing the existing D2-type receptors to become locally dominant in controlling the synapse's reaction to drops in global dopamine concentration. Although the actual curve of θ-dependent neuromodulation of synaptic efficacy would be up for experimental refinement, such a weight-dependent dynamic reconfiguration of D1-type/D2-type receptor ratio might allow for fast dopamine-dependent modulation of synaptic transmission to take place. Similarly, on D2-dominant neurons, the concentration of D2-type dopaminergic receptors may be locally increased with a strengthening of synapses, leading to a supposed opposite behaviour on application of dopamine.

Chapter 9

Reinforcing specific spatiotemporal patterns

9.1 Introduction

In the previous chapter I established that a dopamine-like neuromodulator may affect a neuron's learning outcome by solely influencing the contrast of synaptic transmission. I thereby showed that a chemical modulator need not act strictly through explicit scaling of a spike timing dependent plasticity rule, but that a network effect that quickly influences postsynaptic responses to spatiotemporal inputs can also account for observed dopamine-dependent changes in synaptic connection strengths. In addition, my presented mechanism also qualitatively replicates the instantaneous effects that dopamine has on neuronal gain and signal to noise ratio in many brain areas, which is not captured by dopamine-dependent rules that simply scale STDP.

My proposed mechanism of dopaminergic reinforcement works by changing the difficulty of an unsupervised learning task. It is therefore not in competition with the unsupervised nature of spike timing dependent updates to the synaptic strength (e.g. by trying to undo weight changes coming from the unsupervised learning rule), but rather enhances or inhibits the ability of a postsynaptic neuron to pick up and tune to a recurring spatiotemporal pattern that it may tune to anyway in a completely unsupervised setting. The definition of what constitutes learning success therefore changes here in comparison to a simpler case where some Hebbian update is multiplied by some scalar reinforcement value. As above-baseline dopamine in my model increases the probability of fast pickup of a certain pattern while below-baseline dopamine delays (possibly infinitely) the pickup of tuning to a presented pattern, learning success may be measured by the number of neurons that reliably respond quickly to a rewarded pattern vs. unrewarded. The detection of a positively reinforced pattern may hence be seen as a faster and stronger population response, while negatively reinforced patterns would show a weaker population response on presentation. As seen in Part II of this text, the population response may itself be spatiotemporally structured (Figure 6.8, p. 82), allowing a detailed recognition by later layers in a feed-forward setting.

In this chapter, I therefore test whether it is possible to use my proposed neuromodulation framework to specifically reinforce some spatiotemporal patterns over others and read out this information by looking at the strength of a population response to a given set of inputs. A high fraction of instantaneously responding postsynaptic neurons would signal a previously reinforced, high-value stimulus, and a low number of quickly or reliably responding neurons would signal a stimulus that has not been previously rewarded or has not yet been encountered. The biological analogue to this within the context of basal ganglia circuits may be the striatal part of the direct pathway (D1R-MSNs), while

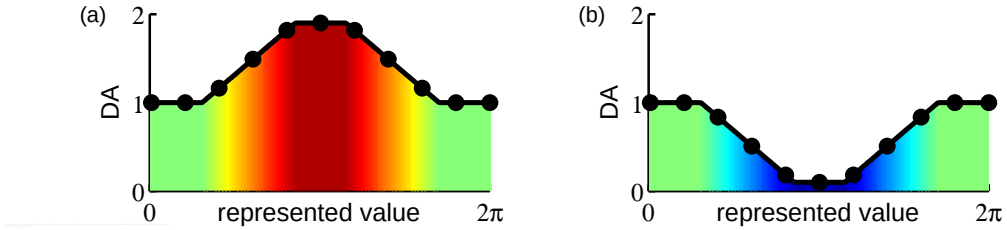


Figure 9.1: Neuromodulator levels now change as a function of the value associated with specific polychronous patterns. Neighbourhood is preserved here through the choice of ordered pattern variants (see Figure 6.26, p. 98) and similar patterns are assigned similar values. The large black filled circles here represent example values that will be assigned to each of ten example patterns, where $0 \equiv 2\pi$. Colours (and Y-axis) indicate neuromodulator level (DA) as in the previous chapter, and will be reencountered in the following figures. While the nearly full range of $DA \in [0.1, 1.9]$ is shown here as an example, the actual range of neuromodulator levels applied in simulations may differ. (a) Pattern-specific increases of neuromodulator level (above-baseline DA) will be explored in Section 9.3.1. (b) Pattern-specific decreases of neuromodulator level (below-baseline DA) will be explored in Section 9.3.2.

active detection of negative stimuli for avoidance may be performed by the basal ganglia's indirect pathway.

9.2 Methods

Input data

Spatiotemporal input patterns were generated by the type B pattern generation method (see Section 7.1.1) without random permutations (Section 7.1.5) from the poisson distributed spike trains of 600 out of 2000 input units. The remaining 1400 input units continued random poisson distributed firing without any further structure imposed on their outputs. No information about pattern identity could be extracted from the firing rate of input units, and steady-state correlations between input unit spike trains were also kept uninformative (see Section 7.1.6, p. 111).

The circular input space of the 600 pattern-presenting units was used to present 40 different equidistant patterns of 100ms length (see also Section 7.1.7). Inter-pattern noise also lasted for 100ms, leading to a new pattern being presented every 200ms. The choice of which pattern to present was drawn uniformly from all 40 options on each new presentation. Neighbouring patterns were similar in that they shared more units with similar spike delay (relative to pattern onset) than non-neighbouring patterns.

Output neurons

The group of postsynaptic neurons consisted of 100 Izhikevich-1D model neurons (Izhikevich, 2003, 2004), without any mutual inhibition (no recurrent connections). Plasticity was allowed with the start of simulation, but input connection strengths were initially too small to allow the postsynaptic neurons to fire. Learning onset was delayed by 1s per neuron, leading to a maximum learning onset delay of 99s for the last neuron. Apart from the onset delay, all postsynaptic neurons were initialised to exactly identical values.

Connectivity

Connections were formed from each input unit to each output neuron in an all-to-all feedforward setup and initialised to a synaptic connection strength of $w = 0.05$ for every

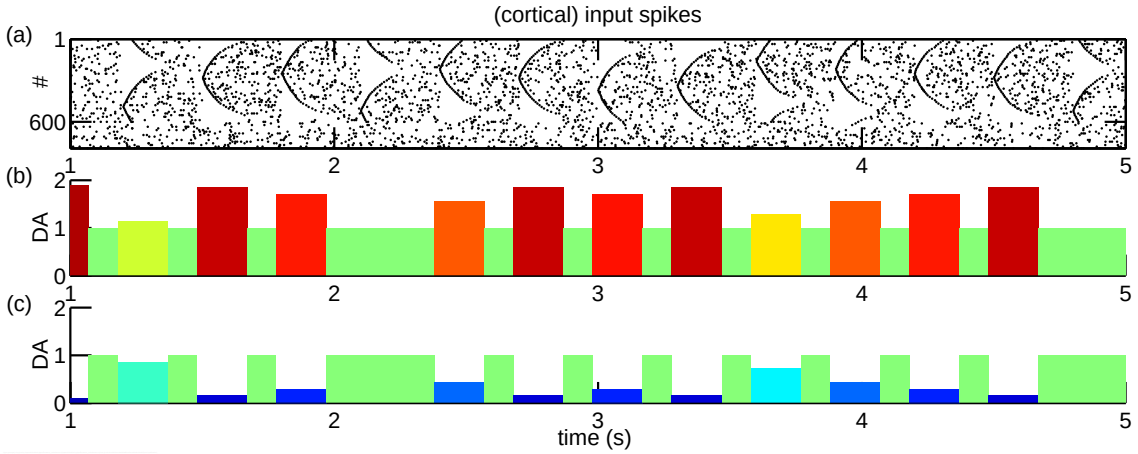


Figure 9.2: Example variations of simulated dopamine during pattern presentations. The neuromodulator concentration leaves baseline levels shortly before pattern onset and returns to baseline shortly after the end of each pattern (see main text). The temporary level of non-baseline dopamine is specified by the pattern that is to be presented (Figure 9.1). Actual DA ranges may vary. (a) Incoming spike trains of 800 out of 2000 input units. The first 600 units present polychronous patterns (type B) while the remaining 1400 units produce purely random spike trains. Only $\frac{1}{12}$ of spike trains are shown to avoid visual clutter. (b) Positive reinforcement as used in Section 9.3.1. (c) Negative reinforcement as used in Section 9.3.2.

connection. No recurrent connections were formed. The initial connection strength did not allow postsynaptic membranes to reach spiking threshold. Instead, we implemented a small normally distributed weight jitter where the synaptic weights of untrained postsynaptic neurons would randomly change in tiny steps until some synapses randomly became strong enough to allow postsynaptic spiking. See *Homeostatic Parameters*. This caused postsynaptic neurons to only begin regular spiking after about 20-30 seconds of simulation time after activation of plasticity for any given neuron. The time at which a postsynaptic neuron begins to respond to any inputs marks the begin of its critical period in this case, as STDP requires both pre- and postsynaptic spikes to take effect (see also Section 5.3).

Homeostatic Parameters

In order to allow neurons to begin to respond to inputs even when connections were initialised very low, a small growth parameter was applied to all weights whenever the sum of all strong synaptic connections of a neuron (we call this the neuron's *trainedness*) was below a given threshold. This homeostatic parameter is not necessary when synaptic connections are initialised to higher values, as done in Chapter 8. See Chapter 5 (Section 5.3.2, p. 68) for a further discussion of activity-independent synaptic growth and other homeostatic effects.

Neuromodulation

Synaptic transmission was modulated by the method introduced in Chapter 8. The neuromodulatory parameter DA was allowed to leave baseline ($DA = 1$) and vary according to the pattern currently being presented. As neighbouring patterns were similar, changes to the postsynaptic representation of one pattern would also affect its neighbouring patterns. This was reflected in pattern dependent neuromodulator levels by using a wider pattern-dependent DA basin as shown in Figure 9.1. Whenever a polychronous pattern was randomly selected for presentation, DA levels were adjusted 10 ms before pattern

onset and set back to baseline exactly at the end of each pattern. The effect of DA timing is further explored in the next chapter.

Sparsification/Generalisation Weight Threshold

The threshold for neuromodulator-dependent contrast enhancement θ was chosen to always represent the median of the current baseline weight distribution of each untrained postsynaptic neuron. For trained neurons (neurons with a sufficient number of strong connections), θ was kept between the two peaks of the bimodal weight distribution that results from attractor-less STDP. This was done to allow the simulated dopamine to indeed act as a contrast enhancer, where high levels of neuromodulation ($DA > 1$) would sharpen the distribution of effective weights into a situation where above-median synapses would "over-confidently" act as having an increased effect on the postsynaptic membrane, and below-median synapses would "under-confidently" act as having a decreased effect on the postsynaptic membrane potential. For low levels of neuromodulation, the synaptic contrast was instead decreased, by gathering the effective weights of all synapses around the threshold θ, as demonstrated in Chapter 8.

Further Simulation Parameters

The network was repeatedly simulated for 180 seconds, after which the 600 weights to pattern-presenting input units were plotted for each postsynaptic neuron (Figures 9.3 and 9.4).

9.3 Results

The learning task here can be described as a modulation of unsupervised learning by dynamically increasing or decreasing the contrast within the learning domain as learning progresses. This means that the formally unsupervised learning mechanism of STDP becomes controlled by making the learning problem harder or easier, depending on the type of pattern currently being presented to each postsynaptic neuron.

9.3.1 Distorting the receptive map towards a given pattern range

Unmodulated unsupervised learning already forms slight differences in the receptive fields of the postsynaptic neurons due to variations within background noise and the nature of STDP combined with polychronous inputs (as presented in Chapter 6). These slight differences in receptive fields are pronounced by a phasic increase in neuromodulator level ($DA > 1$) such that only those postsynaptic neurons tend to respond to concurrently spiking presynaptic input units that already have (by chance) slightly stronger connections to those active units. Postsynaptic neurons that only have strong connections to other, not currently active, presynaptic units have a lower probability to respond. Therefore, increasing the simulated dopamine level together with the presentation of specific patterns increases the tuning probability of yet hardly selective neurons to this pattern. Neurons that are already reliably tuned to any pattern are less affected by this form of neuromodulation, because receptive fields that are already strongly distinctive do not much change the neuron's response when there is a temporary increase of dopamine. As an example, imagine a postsynaptic neuron that has a very different receptive field from that required to respond to some presented incoming pattern. As STDP with polychronous patterns produces sparse receptive fields, a neuron with such a different receptive field will likely

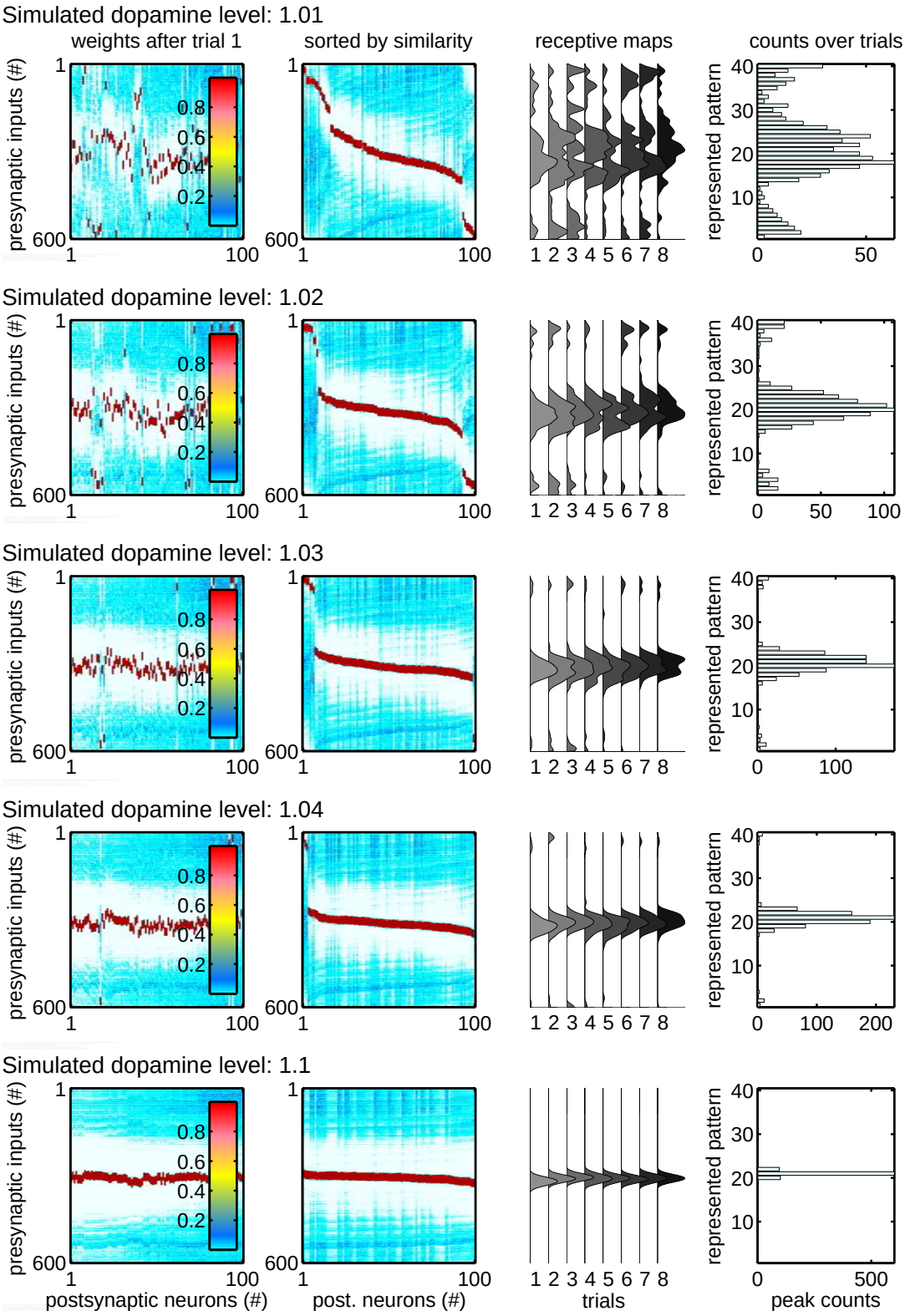


Figure 9.3: Overview images of repeated simulations where a subset of polychronous patterns was presented together with above-baseline dopamine. Levels of simulated dopamine ranged from $DA = 1.01$ to $DA = 1.1$ within the paradigm described in Chapter 8. Higher levels of dopamine paired with pattern presentation lead to more postsynaptic neurons tuning to this range of patterns. The exact location of the peak of receptive fields also becomes more predictable. Each simulation was repeated eight times (8 trials), as seen in the third column. Subplot descriptions as in Figure 7.8 (p. 117). Simulation parameters given in Appendix A.3.10.

have very weak connections to the input units active together during presentation of the given pattern. Likewise, most strong incoming synapses of this postsynaptic neuron will be connected to input units that do not fire or fire late during pattern presentation.

As pattern-coupled neuromodulator increases continue to reinforce the pickup of selected patterns, the number of postsynaptic neurons responding to these patterns rises. This can be seen in the weight matrices after multiple seconds of simulation in Figure 9.3. When the increase of DA is only slightly above baseline (Figure 9.3, top), a slight tendency to tune to a reinforced pattern (first units of pattern around unit #300) can already be seen. However, some of the 100 postsynaptic neurons still tune to other patterns.

When the same simulation experiment is repeated with slightly higher increases above baseline DA, a larger fraction of postsynaptic neurons tunes to the reinforced pattern. At $DA = 1.10$, all postsynaptic neurons reliably tune to the reinforced pattern, thereby forming a strong population response as soon as the reinforced pattern begins to be presented, and not responding at all when other patterns are presented.

Pattern-dependent increases in simulated dopamine level hence have a strong effect on receptive field formation. Keep in mind that no direct scaling of synaptic plasticity is allowed here, and only changes in the postsynaptic neurons' activity due to modulated synaptic transmission can influence the formation of receptive fields.

9.3.2 Distorting the receptive map away from a given pattern range

To find out if it is possible to specifically *avoid* the pickup of a negatively reinforced pattern, we combine below-baseline DA with the presentation of specific patterns. As visualised in Figure 9.1b, the simulated concentration of dopamine (or similar neuromodulator) was dropped whenever a given range of patterns was presented to a group of 100 postsynaptic neurons. Again, all neurons were initialised with exactly equal synaptic weights and membrane dynamics, and the only difference was the time at which plasticity was switched on in each neuron (1s offset), emulating different times of neurogenesis. Differences between receptive fields were therefore solely due to variations of the background noise at the time when neurons reached criticality (began to respond).

The near-uniform coverage of the input space without modulation (at $DA = 1$, see Figure 7.11) became less uniform when some patterns were paired with lower-than-baseline dopamine ($DA < 1$). For $DA = 0.99$ (Figure 9.4, top row), patterns centred around input unit #300 had a slightly smaller tendency of being learnt. When simulations were repeated with stronger drops of DA, the avoidance of affected patterns became even more pronounced (Figure 9.4, rows 2-5).

However, with the current setup, it did not seem possible to completely stop negatively reinforced patterns from being learnt. The bottom row of Figure 9.4 still shows rare occasions where postsynaptic neurons do tune to negatively reinforced patterns, albeit more rarely. The next chapter therefore further explores ways in which negative reinforcement may be used in a D1R-type network to avoid the pickup of specific patterns.

9.4 Conclusion

In this chapter, I showed that my approach to neuromodulation which was introduced in Chapter 8 can indeed be used to shape the map of receptive fields without directly interfering with an unsupervised STDP rule. Instead, the way in which neuromodulation is applied only to synaptic transmission (instead of directly influencing plasticity) changes the difficulty of the learning problem from a postsynaptic neuron's perspective. This is

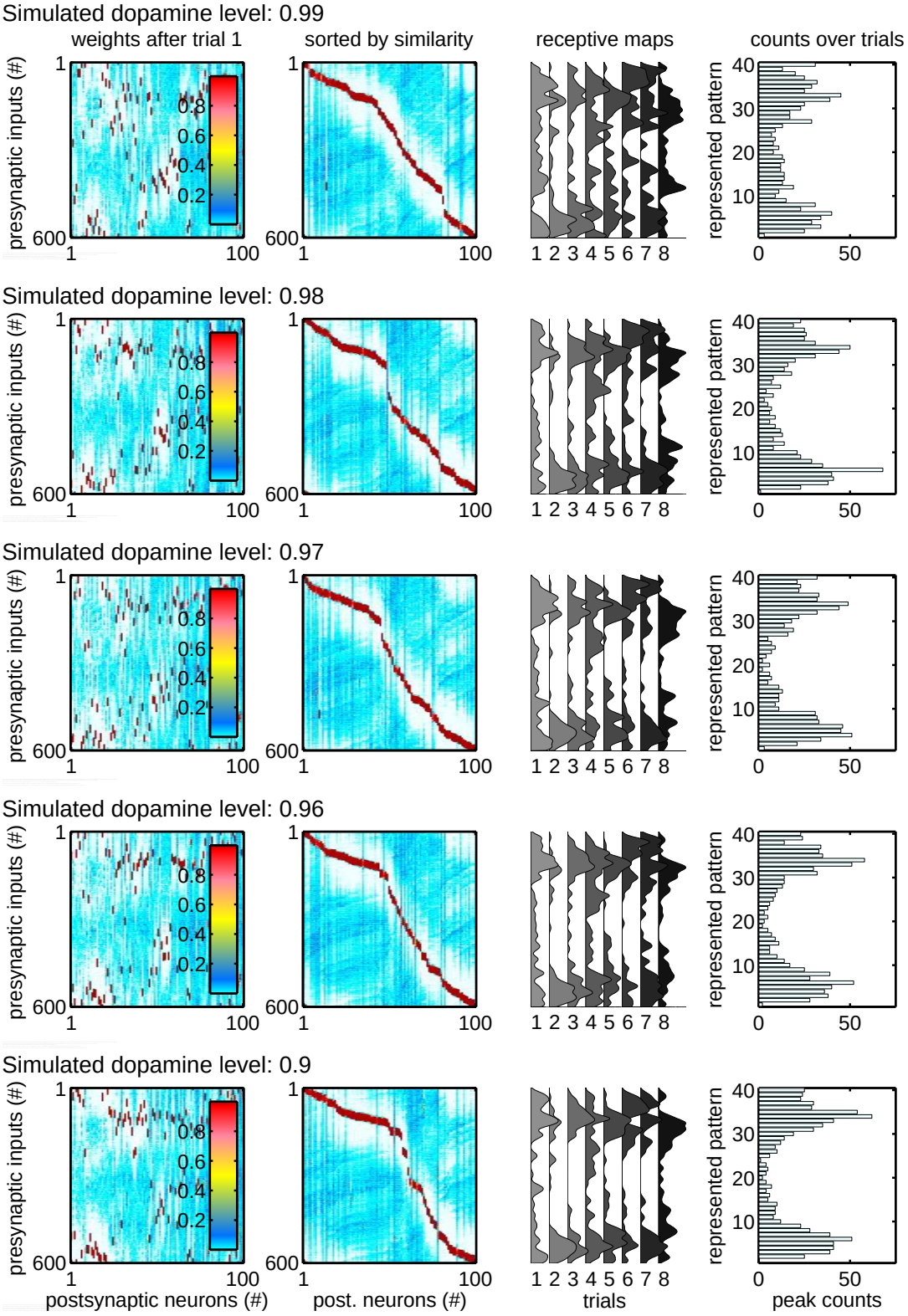


Figure 9.4: Overview images of repeated simulations where a subset of polychronous patterns was presented together with below-baseline dopamine. Levels of simulated dopamine ranged from $DA = 0.90$ to $DA = 0.99$ within the paradigm described in Chapter 8. Lower levels of dopamine paired with pattern presentation lead to less postsynaptic neurons tuning to this range of patterns. However, avoidance of synaptic tuning to these patterns could not be completely stopped within these tests, so Chapter 10 further explores the low-DA case. Each simulation was repeated eight times (8 trials), as seen in the third column. Subplot descriptions as in Figure 7.8 (p. 117). Simulation parameters given in Appendix A.3.10.

done through a dynamic increase or decrease of contrast of polychronous patterns within a stream of noisy background spikes. Only this change of contrast in synaptic transmission then influences possible spike responses of the postsynaptic neuron(s). And only due to these changed response characteristics does dopaminergic modulation influence the outcome of plasticity in this candidate model.

Instead of working against the unsupervised nature of spike timing dependent plasticity (Frémaux et al., 2010), unsupervised learning is allowed to take place and only sped up or slowed down through changes of contrast. It can, for example, only learn to detect patterns that are actually present within its inputs. And if the modulatory factor remains constant on any level, the underlying unsupervised learning mechanism will always try to form the best representation of inputs it can achieve for the given fixed level of contrast.

The possibility that dopamine may act as a contrast enhancer has been suggested by Nicola et al. (2004) and others, but only on a broad scale without specific ideas of how a temporary dopamine-dependent variation in contrast may be useful for neural information processing. With this chapter, I have now shown that a dynamic variation of contrast can indeed be made useful within a functional implementation of spiking neurons that learn to preferably detect some patterns over others. The new mechanism tested here (and proposed in Chapter 8) may also be useful for understanding neuromodulatory effects in brain areas other than the basal ganglia (Thurley et al., 2008), and may also apply to neuromodulators other than just dopamine (Calabresi et al., 2000; Daw et al., 2002; Delgado et al., 2008; Hasselmo and McGaughy, 2004; Wang et al., 2006).

The definition of reinforced learning I use here aims to either *learn* positively reinforced patterns or *not learn* negatively reinforced patterns. This definition of learning success is different from some reinforcement learning tasks that aim to detect either some option A or some option B. However, learning *not to select* a negatively reinforced option B may be achieved by *not learning* to choose it. The avoidance of negative options need not be performed by the same mechanisms/networks that learn to choose positively reinforced options in the brain, and there is some evidence that this is indeed the case (Centonze et al., 2002; Daw et al., 2002; Delgado et al., 2008; Frank et al., 2007).

In the current chapter, I only used a class of polychronous input patterns that involved a fixed ordering of neighbouring input units. That is, no randomly permuted input patterns were used (Section 7.1.5). This allowed the presented patterns to yield neat clusters of strong weights within each neuron's receptive field. Future examinations need to test the contrast-altering method of modulation while using randomly permuted patterns.

The choice of pattern class also influenced reinforcement of neighbouring patterns. As neighbouring patterns shared many common presenting input units with only slightly shifted spike timing offsets, any reinforcement (positive or negative) of one pattern also changed the degree to which neighbouring patterns were favoured or avoided. This is the main reason why it was difficult to further narrow down the trough of pattern avoidance for negatively reinforced patterns (see right-most histograms in Figure 9.4).

To further explore this method of neuromodulatory control on STDP-based learning of polychronous patterns, more work is necessary. Especially the tendency of a group of postsynaptic neurons to all learn identical receptive fields as soon as a subset of polychronous input patterns are paired with slightly enhanced dopamine need to be understood if we wish to establish this method as a possible explanation of how dopamine may affect fast information processing, action selection, and plasticity processes within the basal ganglia and other parts of the brain. The next chapter therefore explores a possible mechanism of self-regulation that could prove useful in preventing over-representation of reinforced patterns within a dopaminergic reinforcement paradigm for the basal ganglia.

Chapter 10

Limiting representational monopolies

10.1 Introduction

We saw in Chapter 9 that the training of groups of polychronous input patterns can indeed be affected by the new method of dopaminergic modulation that I proposed in Chapter 8. However, it is still far away from being declared a universal plasticity mechanism for procedural learning in the basal ganglia. One large issue that became apparent in Chapter 9 and needs to be solved first is that of how to avoid the formation of identical receptive fields when positively reinforcing a pattern.

For baseline levels of dopamine, I showed in Chapter 7 (Section 7.3, p. 116) that a simple delayed onset of plasticity and a uniform probability of pattern occurrence forms a uniform distribution of receptive fields across the space of input patterns. If we wish to distort this uniform dispersion of receptive fields to approximate (some smooth representation of) some continuous value function as known from the field of reinforcement learning, we need to be able to control the amount of distortion to precisely represent received feedback.

The next step therefore is to test if the over-representation of partially reinforced patterns can be controlled. For this, we make use of basal ganglia anatomy and known behaviours of dopaminergic outputs of the substantia nigra pars compacta (SNc, p. 25).

10.2 Methods

Input data

As in the previous chapter, spatiotemporal input patterns were generated by a method that leaves no trace of pattern identity or even pattern presence in the firing rate of input units. Two-legged (Section 6.3) type D patterns (Section 7.1.1) without random permutations (Section 7.1.5) were used as input data for 600 out of $N_{pre} = 2000$ presynaptic input units.

In opposition to the previous chapter, only a single pattern was repeatedly presented (see 1st row of Figures 10.1 - 10.3). Pattern duration was again 100 ms, with 200 ms (Sections 10.3.1 and 10.3.2) or 400ms (Section 10.3.3) between the start of two presentations.

Output neurons

As in the previous chapter, $N_{post} = 100$ Izhikevich-1D model neurons (Izhikevich, 2003, 2004) were used as postsynaptic receivers. Plasticity was activated (or reset once per simulation) with a one second delay per neuron, so that the last neuron began to learn

99s after the first (see Section 7.3.2). Apart from the onset delay, all postsynaptic neurons were initialised to exactly identical values.

Connectivity

No recurrent connections between postsynaptic neurons were allowed. Any differences between the receptive fields of two postsynaptic neurons can therefore not arise through mutual inhibition.

Connections from the group of presynaptic inputs to each postsynaptic neuron were initialised in an all-to-all feed-forward setup with initial synaptic strength of $w = 0.05$ for all connections. Differences between the receptive fields of any two postsynaptic neurons therefore need to arise from variations in input noise at the time when each postsynaptic neuron first becomes responsive.

Homeostatic Parameters

In order to allow neurons to begin to respond to inputs even when connections were initially very low, a small growth parameter was applied to all weights whenever the sum of all synaptic connections of a neuron was below a given threshold. This homeostatic parameter is not necessary when synaptic connections are initialised to higher values, as done in Chapter 8. But as Section 10.3.3 requires neurons to be quiet when they have not yet formed a selective receptive fields, initialising all synaptic connections to strong weights was not an option here. See Chapter 5 (Section 5.3.2, p. 68) for a further discussion of homeostatic effects.

Neuromodulation

Synaptic transmission was modulated by dopamine through the method introduced in Chapter 8. In Sections 10.3.1 and 10.3.2, simulated dopamine (DA) is repeatedly set to below-baseline values ($DA < 1$) for some period during each pattern presentation. This repeating pattern-related drop of neuromodulator level starts at either $t = 1s$ or $t = 90s$ in the results shown, and ends at $t = 280s$ ($20s$ before the end of simulation). These times were chosen because they allowed a good visualisation of effects (see *Results*). Outside of this timespan, the level of simulated dopamine was kept constant at "tonic" baseline ($DA = 1$).

In Section 10.3.3, dopamine is no more controlled explicitly through predefined timing, but depends on the number of postsynaptic neurons that respond to a given stimulus. Each spike of a postsynaptic neuron here decreases the level of simulated dopamine by a small constant value, while the dopamine variable recovers back to baseline exponentially with $\tau_{DA} = 100ms$. This produces a comparably smooth change of dopamine level in opposition to Sections 10.3.1 and 10.3.2.

Sparsification/Generalisation Weight Threshold

The threshold for neuromodulator-dependent contrast enhancement θ was chosen to always represent the median of the current baseline weight distribution of each untrained postsynaptic neuron. This was done to allow the effect of DA to indeed approach that of contrast enhancement, where high levels of neuromodulation ($DA > 1$) would sharpen the distribution of effective weights into a situation where above-median synapses would over-confidently act as having an increased effect on the postsynaptic membrane and below-median synapses would act as having a further decreased effect on the postsynaptic

membrane potential. For low levels of neuromodulation, the synaptic contrast would be decreased by gathering the effective weights of all synapses around the weight threshold θ, as demonstrated in Chapter 8.

Further Simulation Parameters

The network was simulated for 300s (5 minutes, Sections 10.3.1 and 10.3.2) or 1200s (20 minutes, Section 10.3.3), after which the 600 weights to pattern-presenting input units were plotted for each postsynaptic neuron. A tabular summary of simulation parameters is given in Appendix A.3.12.

10.3 Results

We now examine how the formation of receptive fields for a specific pattern can be slowed and/or avoided. This is necessary to avoid over-representation of patterns among the map of receptive fields. We therefore repeatedly present the same (polychronous) pattern to each of 100 postsynaptic neurons, and try to pause unsupervised learning of this pattern by varying the level of simulated dopamine (DA). While the goal is to test whether it is possible to set up a self-regulating feedback loop that uses dopamine for controlling the number of neurons that form selective receptive fields for a given pattern (Section 10.3.3), we first need to find out if it is at all possible to pause the unsupervised learning process through a reduction of dopamine-dependent contrast. While Section 10.3.1 examines the effect that the timing of DA change has in relation to pattern onset, Section 10.3.2 tests how far the dopamine level needs to be reduced in our model to have a noticeable effect.

For each case, we also test if the time at which the level of dopamine begins to change has any effect on the overall outcome. We test this by repeating each experiment for two cases of DA change onset (1s and 90s), shown on the left (onset 1s) and right (onset 90s) of the following figures.

10.3.1 Timing effects of dopaminergic change

In the first set of tests, the level of simulated dopamine was reduced to $DA = 0$ for the duration of each pattern and some time after that. The time at which DA is reduced was varied from 10 ms before pattern onset, to 10 ms after and 50 ms after pattern onset. Simulated dopamine always returned back to baseline exactly 50 ms after pattern completion, as will be seen in the figures. Figures 10.1, 10.2 and 10.3 show how these predefined changes of simulated dopamine level affect the formation of receptive fields.

10ms before pattern onset

As the first test, we decrease the level of simulated dopamine to zero at 10 ms *before* each pattern onset. It is kept there for 160 ms (=50 ms before next pattern), after which it is set back to baseline ($DA = 1$). In Figure 10.1a-f, this pattern-related change of dopamine level begins already after $t = 1s$ of simulation and ends only 20s before the end of the 300s simulation ($t = 280s$). While there is a slight decrease of synaptic connections to input units that take part in presenting early parts of each pattern (units around ca. #300 in d-f), no strong connections are formed at all. This is understandable as the extreme low level of simulated dopamine does not allow any reliable postsynaptic response to its inputs to take place.

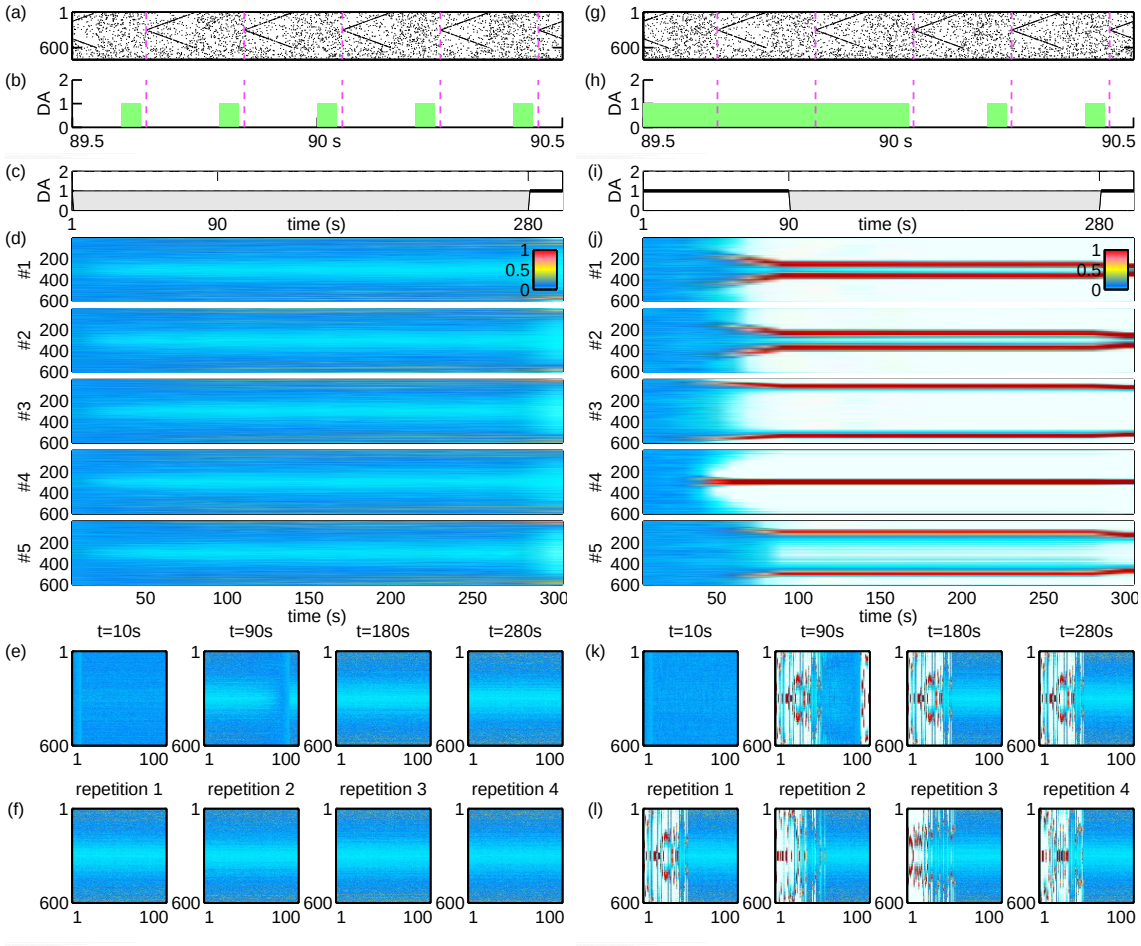


Figure 10.1: Development of synaptic weights under rapid changes of dopamine level ***10ms before*** each pattern. Activation of non-baseline DA after 1s (***left***) or after 90s (***right***). (***a,g***) Polychronous input patterns formed by 600 out of 2000 input units. Time scale as in b,h. Pattern onset marked by vertical magenta line. (***b,h***) Area plot of dopamine levels during one second of simulation. Pattern onset marked by vertical magenta line as in a,g. Here, the drop of DA level occurs slightly (10ms) before pattern onset, and lasts until 50 ms before the onset of the next pattern. (***c,i***) Zoomed-out overview of dopamine variability during full course of simulation. Grey area visualises phases of rapidly changing dopamine, with upper and lower bounds showing maximum and minimum of dopamine level per second. A single thick black line shows phases when dopamine is fixed to a single value. (***d,j***) Time course of input weight matrices of five postsynaptic neurons (out of 100) during simulation. Only the connections to the 600 pattern-presenting input units are shown (y-axis) for reduced clutter. Simulation time is shown on the x-axis. (***e,k***) Snapshots of weight matrices of all 100 postsynaptic neurons at different stages of simulation (time t shown above each plot). As postsynaptic neurons are each reset once with 1s delay, the development of initially diverse receptive fields can be noticed in subplots where $t < 100$s. (***f,l***) Repetitions of the same experiment show qualitatively similar results, while the distinct initial receptive fields are different each time. However, as only one polychronous pattern is being presented in these tests, the final receptive field will always settle on those input units that take part in presenting the initial spikes of each pattern, i.e. near input unit #300 (see Section 6.2.1).

However, we wish to not completely prevent the formation of postsynaptic receptive fields, but only pause their further development after some initial neurons have already had the chance to partially tune to the presented pattern. We therefore repeat the set of simulations while giving the postsynaptic neurons a chance to form some preliminary receptive fields before activating the rapid drop of dopamine. In the simulations shown in Figure 10.1g-l, simulated dopamine remains fixed at baseline (DA=1) for the first 90s of

simulation. As can be seen in the weight matrix subplot for $t = 90s$ (Figure 10.1k), many of the early initialised postsynaptic neurons have formed or are forming a distinctive receptive field to detect the occurrence of the polychronous input pattern. This set of receptive fields is kept stable while dopamine keeps to drop to $DA = 0$ during pattern presentation (e.g. $t = 180s$, $t = 280s$), which can also be seen in Figure 10.1j as the development of the input weight matrices of five postsynaptic neurons remains constant during the period of $t = 90$ to $t = 280$ seconds. Before and after this period, the receptive fields of the shown postsynaptic neurons develop normally, as would be expected from Chapter 6. The pausing of receptive field development is particularly visible for postsynaptic neurons #1 and #2 in Figure 10.1j, where two groups of strong weights begin to move towards central (ca. #300) presynaptic units until time $t = 90s$, at which point no further changes to the weight vectors happen until $t = 280s$. The same here can also be observed for the weight vectors of postsynaptic neurons #3 and #5, which are initially tuned to later parts of the polychronous input pattern. Postsynaptic neuron #4 is also affected by the rapid changing of DA, but as its receptive field already reaches the first pattern-presenting input units before $t = 90s$, no further changes to the receptive field of postsynaptic neuron #4 are observed after this point.

In summary, it seems as though reducing the level of simulated dopamine to zero before each pattern presentation acts as to fully block postsynaptic receptive field formation.

10ms after pattern onset

We now examine the effect of a subtle difference in dopaminergic timing on the formation of postsynaptic receptive fields. Instead of dropping to zero 10 ms *before* pattern onset, our simulated neuromodulator now drops to zero 10 ms *after* pattern onset.

This already has a noticeable effect on the left side of Figure 10.2, where simulated dopamine already begins dropping to zero at $t = 1s$ of simulation time. While in Figure 10.2d most postsynaptic neurons still remain unselective during the first 280 seconds of simulation, the fourth shown postsynaptic neuron here coincidentally forms a strong receptive field for early-firing input units (near presynaptic input unit #300) and keeps it stable during the rest of the simulation. Rows (e) and (f) show that this now happens repeatedly, but only for a minority of postsynaptic neurons during the simulated 300 seconds. Also, it should be noted that no receptive fields initially start out tuned to input units that fire later during pattern presentation as was seen in Figure 10.1k,l. The 10 ms delay of dopamine drop after pattern onset only allows the possibility of tuning to the earliest pattern-presenting input units.

When the repeated rapid drop of simulated dopamine is only started after $t = 90s$ of simulation time (right side of Figure 10.2), the result is more similar to the right side of Figure 10.1. However, Figure 10.2l now shows some additional receptive fields tuned to the early parts of the presented polychronous pattern, which were not there in Figure 10.1l. For receptive fields that have started to form before $t = 90s$ and therefore partially also represent later parts of the polychronous pattern, the pausation effect of dropping DA to zero is still present (see e.g. the fourth shown postsynaptic neuron (#8) in Figure 10.2j).

50ms after pattern onset

If a subtle delay of DA change in relation to pattern onset already shows noticeable differences in the resulting receptive fields, how does a delay of 50ms after pattern onset affect receptive field formation? Figure 10.3 shows that the formation of selective receptive

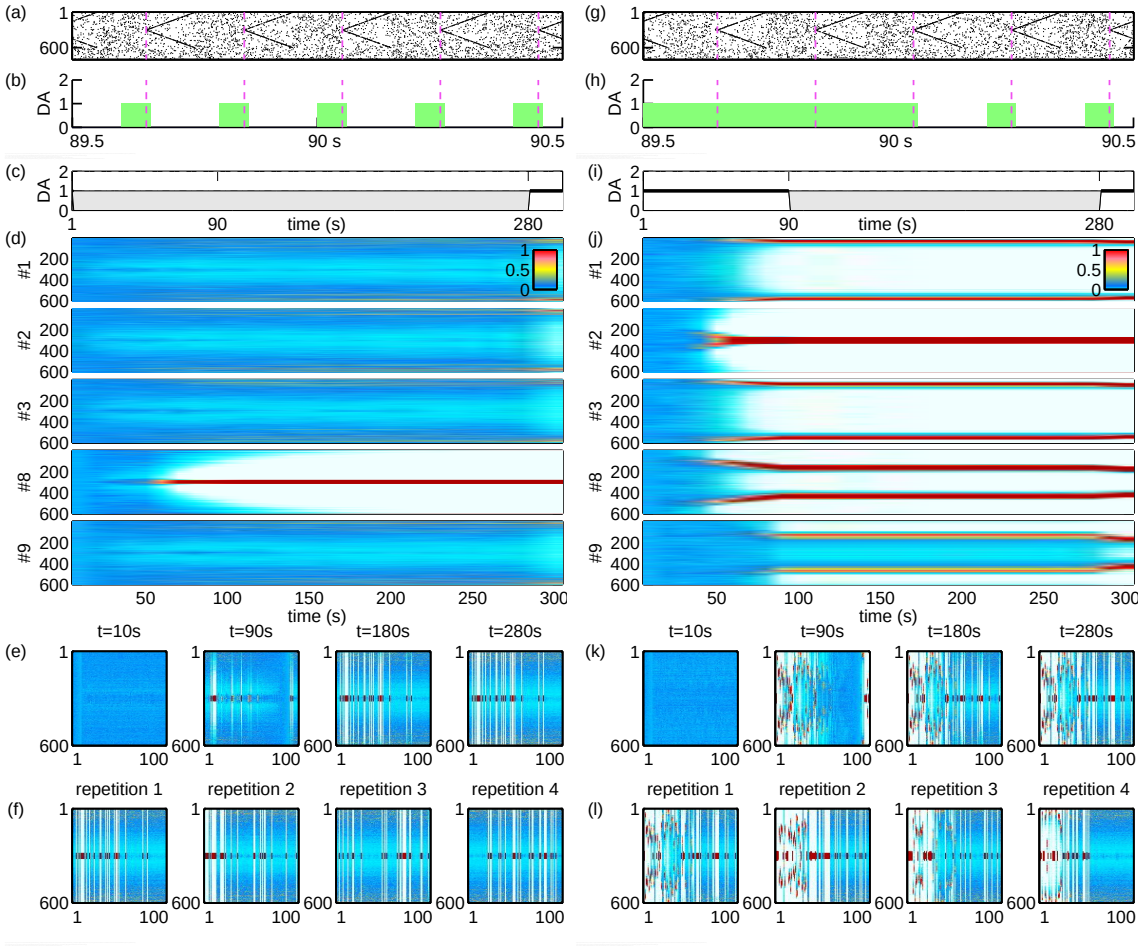


Figure 10.2: Development of synaptic weights under rapid changes of dopamine level ***10ms after*** each pattern. Activation of non-baseline DA after 1s (***left***) or after 90s (***right***). See Figure 10.1 for description of subplots.

fields is then a lot more common. Again, the left side of the figure shows the process of receptive field formation while simulated dopamine repeatedly drops to zero rapidly from the first second ($t = 1s$), while the right side shows the effects when the neuromodulator only leaves baseline between $t = 90s$ and $t = 280s$.

Figure 10.3d now shows all five displayed postsynaptic neurons form a selective receptive field to exactly those input units which take part in presenting the first milliseconds of the pattern. Indeed, all 100 postsynaptic neurons form a selective receptive field to a small range of early-firing input units (Figure 10.3f). In the development snapshots at seconds 10, 90, 180 and 280 (Figure 10.3e), the process of formation can be observed as each postsynaptic neuron is once reset during the first 100 seconds of simulation. While all postsynaptic neurons form selective receptive fields, any initial receptive fields respond to input units that fire during the first 50ms of pattern presentation, because the initial development of receptive fields that select later-firing input units is being blocked by the dopaminergic drop to zero that always happens 50ms after pattern-onset.

The right side of Figure 10.3 at first sight looks as if the tuning process can proceed normally when DA first leaves baseline at $t = 90s$. In Figure 10.3j, all of the five postsynaptic neurons show a random-looking initial position of weight matrices, which then slowly converges on the early-firing input units for the presented pattern. However, when

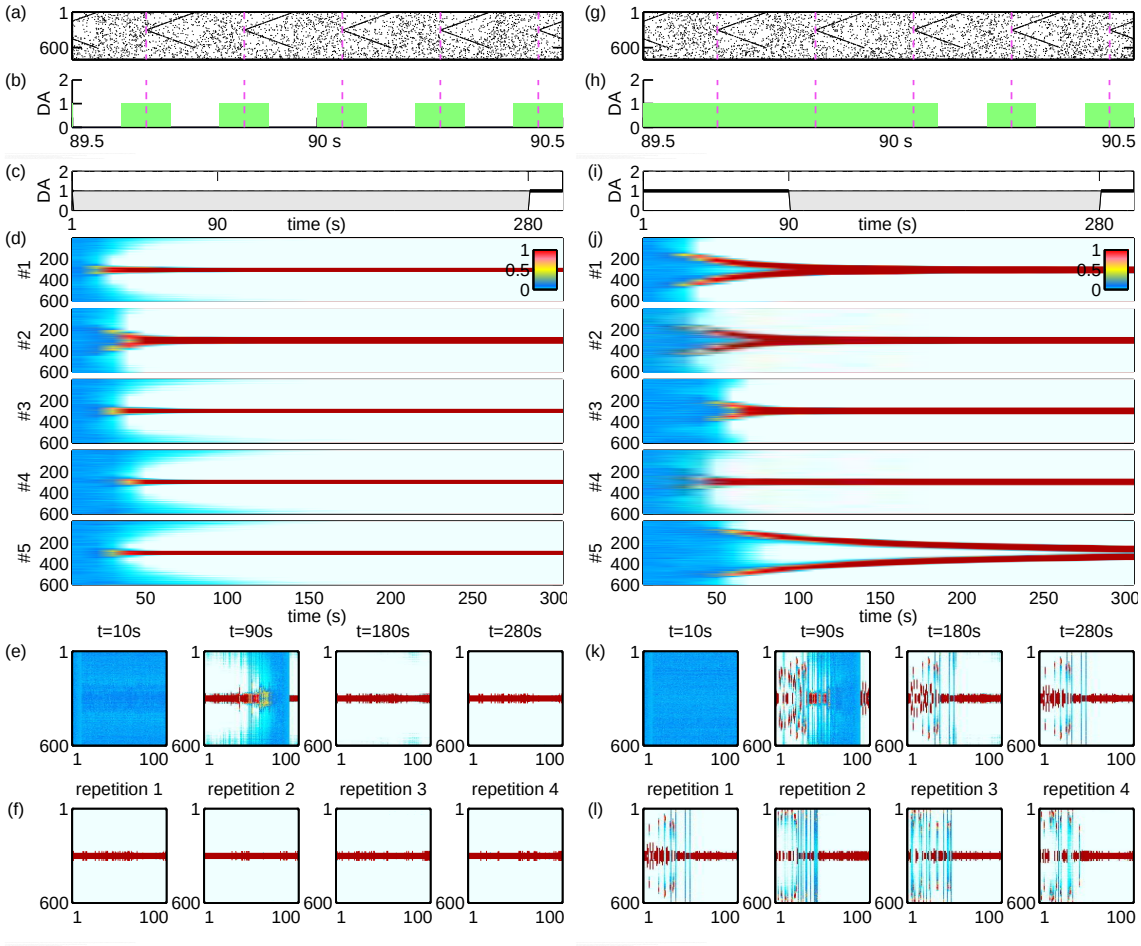


Figure 10.3: Development of synaptic weights under rapid changes of dopamine level ***50ms after*** each pattern. Activation of non-baseline DA after 1s **(*left*)** or after 90s **(*right*)**. See Figure 10.1 for description of subplots.

looking at Figure 10.3l, one can see that some receptive fields that are initially selective for very late-firing input units (near presynaptic unit #1 and #600) do see a pause of convergence. In these final weight matrices after 300s of simulation, every receptive field has either converged to the start of the polychronous pattern (input units ca. #151 - #450), or remains selective to input units that only take part in presenting parts of the polychronous pattern that happen more than 50ms after pattern onset (input units ca. #1 - #150 and #451 - #600).

The relative delay of simulated dopamine drop in relation to pattern onset can hence be used to control which postsynaptic receptive fields continue to converge on the earliest inputs, and which postsynaptic neurons have their receptive fields remain fixed to detect only late-firing input units.

10.3.2 Scaling effects of dopaminergic change

Now that we have looked at the effects that the timing of a drop in neuromodulator level can have on receptive field development, we also would like to know how far the level of simulated dopamine needs to be reduced in this framework (Chapter 8) for the effect to be visible. Of course the absolute values of simulated dopamine used here have only little in common with real biophysical concentrations of dopamine-like neurotransmitters within

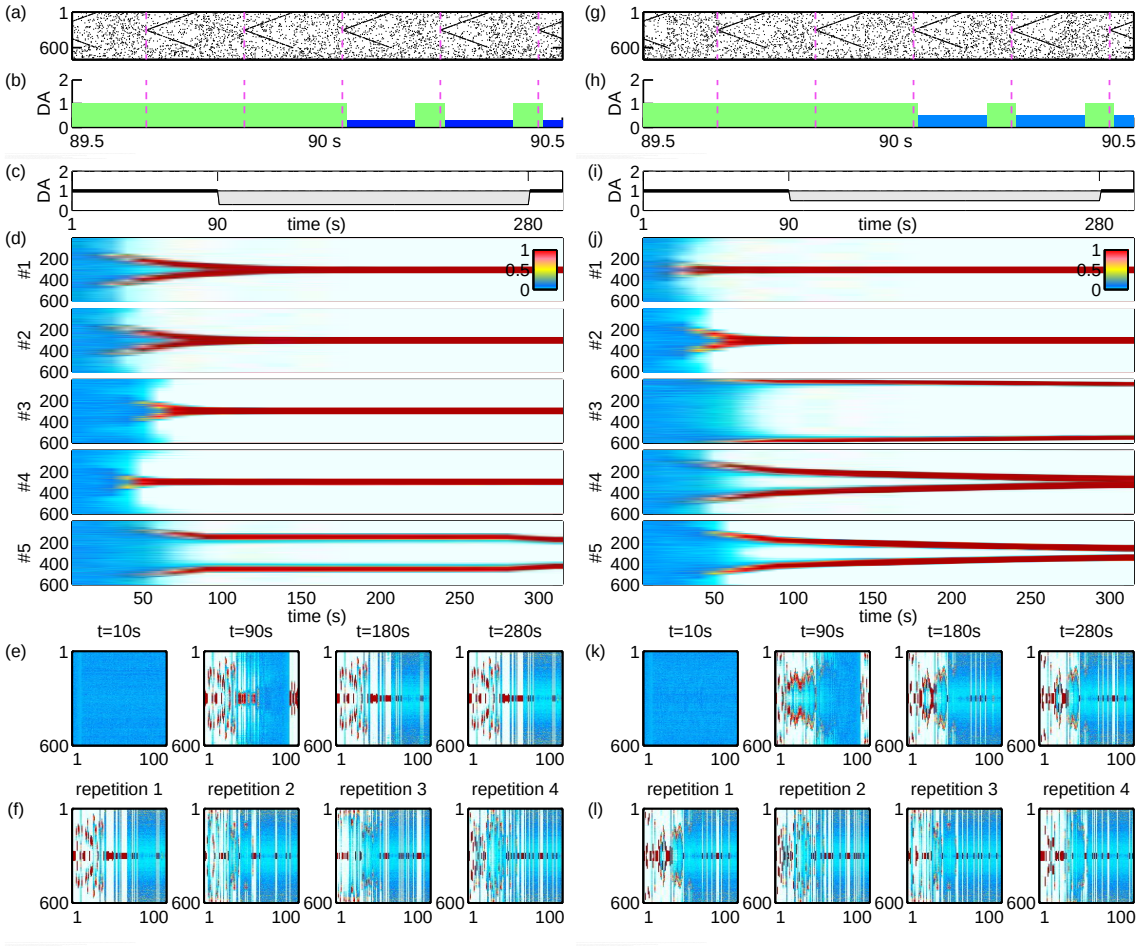


Figure 10.4: Development of synaptic weights under rapid changes of dopamine level ***10ms after*** each pattern. Non-baseline dopamine drops to $DA = 0.3$ **(*left*)** or $DA = 0.5$ **(*right*)** during pattern presentation. See Figure 10.1 for description of subplots.

neural tissue. But as the principle of influencing synaptic weight development through rapidly modifying the contrast of synaptic transmission has only just been established in Chapter 8, we first need to understand the dynamical implications in our model environment before searching for quantitive matches with (yet to be gathered) biological data.

Figure 10.4 shows the development of synaptic weights when dopamine is *not* lowered all the way to zero. The drop of simulated dopamine here happens 10ms after pattern onset, as was used in Figure 10.2. In both sets of simulations on the left and right of Figure 10.4, dopamine only leaves baseline between $t = 90s$ and $t = 280s$ as this is the more interesting case (see Section 10.3.1). The difference between the two sides of Figure 10.4 is that dopamine is dropped to $DA = 0.3$ on the left, while it is dropped to $DA = 0.5$ on the right side. The choice of displayed dopamine levels was made for the different effects on postsynaptic receptive field development that happens near $DA = 0.4$ for the current set of parameters.

Figure 10.4d shows a pause in receptive field convergence for postsynaptic neuron #5 when dopamine is dropped to $DA = 0.3$. Also, in Figure 10.4e the receptive fields of the first 50 postsynaptic neurons do not seem to change much after $t = 90s$. This is very similar to what happens when dopamine is dropped all the way to $DA = 0$ (Figure 10.2).

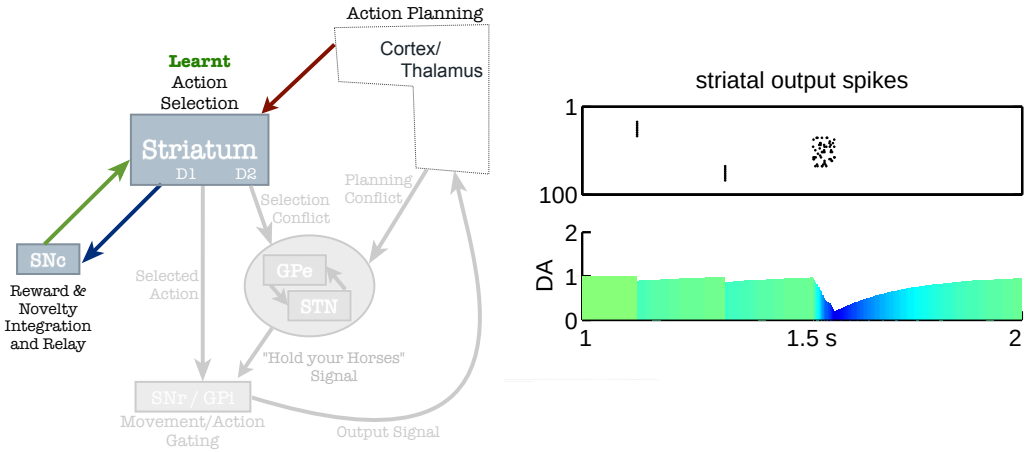


Figure 10.5: Setup of dopaminergic feedback. *Left:* Inhibitory D1R neurons of the striatum also project to SNc while mainly projecting to SNr and GPi. This forms a dopaminergic feedback loop as less dopamine is then released in the striatum. Compare Figure 3.6 (p. 29). *Right:* In our model, we chose a nonlinear relationship between the number of inhibitory striatal output spikes arriving at the SNc and dopamine (DA) concentration in the striatum. After inhibition, the level of simulated dopamine relaxes back to baseline exponentially. See Appendix A.3.12 for implementation notes.

In contrast, when simulated dopamine is only dropped to $DA = 0.5$ on the right side of Figure 10.4, convergence of receptive fields towards the earliest input units does not completely stop but carries on slowly. Figure 10.4k shows how most receptive fields of the first 50 postsynaptic neurons continues to converge towards central values (input unit #300) even while dopamine is repeatedly dropping to $DA = 0.5$ exactly 10ms after each pattern onset. In Figure 10.4j, the development of the weight vectors of postsynaptic neurons #3, #4 and #5 also continues to develop in spite of the pairing of each polychronous input pattern with low dopamine, albeit maybe slower than before $t = 90s$.

The lower value of simulated dopamine drop can hence possibly be used to control the speed at which postsynaptic neurons tune to a specific polychronous input pattern. This might help to explain the successful self-regulation of receptive field formation through dopaminergic feedback that is described in the next subsection.

10.3.3 Dopaminergic Feedback for Self-Regulation

In the previous results presented in this chapter, the level of simulated dopamine (DA) was always set manually to specific values at specific times during the simulation. We now extend the method of dopamine timing to construct a closed loop feedback system that regulates its own dopamine level to prevent the network from over-learning repeatedly presented patterns. The idea here is that the basal ganglia should habituate to recurring behavioural situations and compensate for expected rewards arriving at the SNc from other brain areas. As the striatal neurons (D1R-MSNs) begin to respond to a specific behavioural situation (represented by a polychronous pattern arriving from the cortex and/or thalamus), their GABAergic *direct pathway* projections also inhibit SNc cells (see Chapter 3). Following our working hypothesis (Section 3.3.3), this results in a temporary decrease in activity of SNc cells and therefore a temporary reduction of dopamine level in the striatum. As more striatal D1R neurons tune to a given pattern, the stronger decrease in dopamine level should prevent more striatal neurons from tuning to the same pattern. We tested this in a set of simulations shown in Figure 10.6.

The SNc and its dopaminergic projections to the striatum are here implemented as a

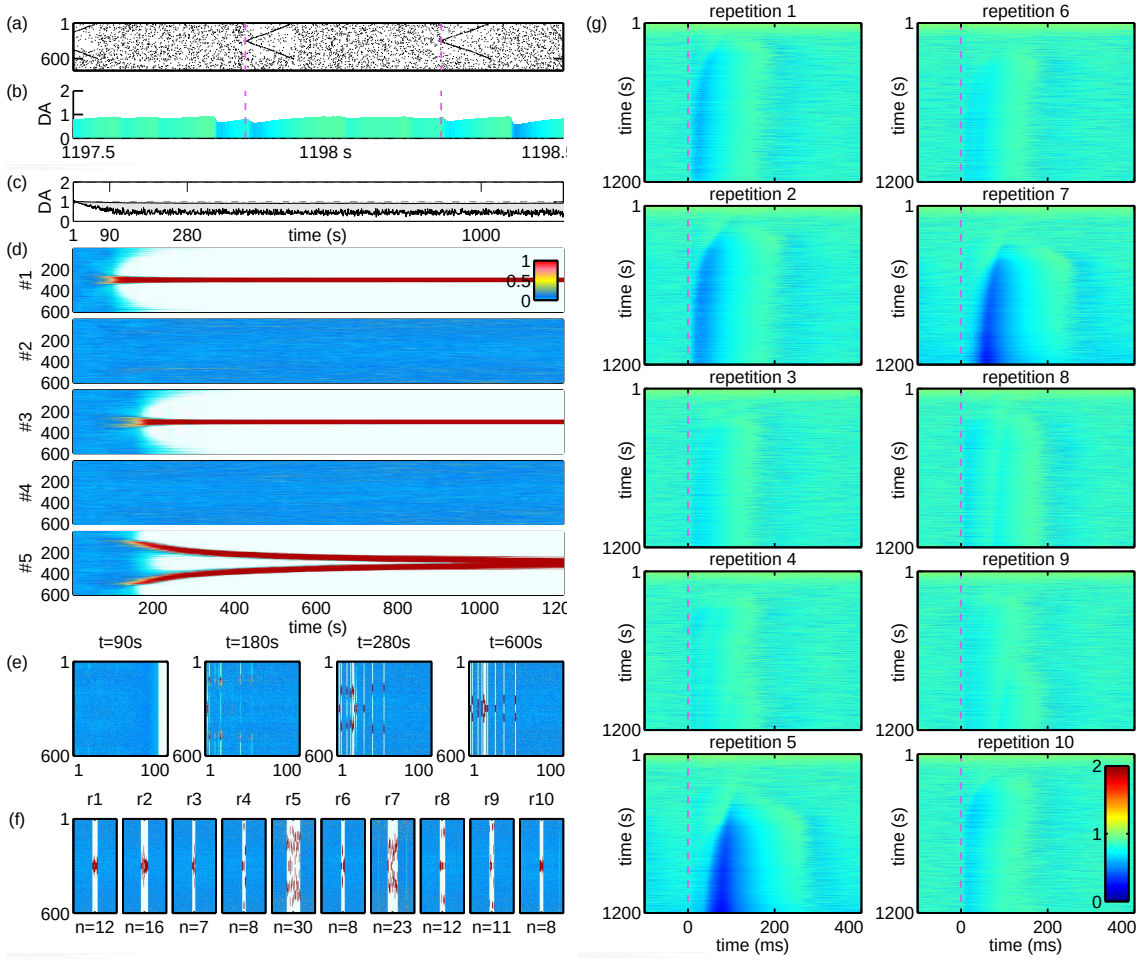


Figure 10.6: Development of synaptic weights under dynamic changes of dopamine level through feedback with every postsynaptic spike. Dopamine level relaxed back to baseline ($DA = 1$) exponentially with $\tau_{DA} = 100ms$. Example weight development of one simulation (***left***) and pattern onset dependent changes in dopamine level (***right***). See Figure 10.1 for description of subplots (a)-(f). (***g***) Neuromodulator levels from 100ms before until 400ms after polychronous pattern onset. Pattern onset marked by dashed magenta line. Colours as in (b). Shown for 10 repetitions of simulation experiment, of which resulting weight matrices are shown in (f).

single variable that controls the dopamine level in the striatum and rests by default at some baseline level of dopamine ($DA = 1$) through implicit tonic activity of dopaminergic cells. Any spike by a striatal D1R-type (inhibitory) neuron deflects the dopamine level released in the striatum, which then slowly recovers back to baseline exponentially with a time constant τ_{DA} (Figure 10.5). This view of dopamine level regulation is highly simplified, but serves the point of testing whether some of SNc activity may possibly be dynamically influenced by inhibitory projections from striatal direct pathway neurons to reduce dopaminergic responses as a stimulus ceases to be novel. As we do not model any other inputs to the SNc here, the resulting trajectory of dopamine levels can not be likened to biological data of SNc activity just yet. One reason for this is the continued lack of experimental data on precisely timed inputs to the SNc, together with the still uncertain coupling of SNc activity to the dynamic changes of dopamine concentrations in the striatum (e.g. Howe et al., 2013).

As dopamine was to be kept sufficiently below baseline for the course of each polychronous pattern (pattern duration 100ms), the recovery time constant τ_{DA} was chosen

to be $\tau_{DA} = 100$ms. This in turn necessitated an increase of pattern offset from 200ms to 400ms in order to allow the DA level to recover sufficiently towards baseline before the next pattern was presented.

Figure 10.6a shows the input patterns with an increased offset, while the dynamic dopamine level that is now controlled through the number of responding postsynaptic neurons is shown in Figure 10.6b. A slight dip of dopamine level can be seen shortly after the onset of each polychronous pattern (dashed magenta vertical lines), which indicates a reliable number of postsynaptic responses to each pattern presentation at this late stage of simulation ($t = 1198s$). However, other dips in dopamine concentration are also visible. A full display of pattern onset-related dips in dopamine concentration of this simulation is shown in Figure 10.6g (repetition 1), where the dips that occur shortly after pattern onset can be seen to be reliable while other dips in concentration are not. Figure 10.6g will be further explained below.

In Figure 10.6c, the overview of simulated dopamine levels during the full simulation shows that while the maximum of dopamine levels always remains close to $DA = 1$, the minimum decreases linearly while more postsynaptic neurons are activated during the first 100s of simulation time, and then remains noisily constant around $DA = 0.4$ thereafter. The maximum DA level settles just below baseline, which is due to the interplay between the DA recovery time constant of $\tau_{DA} = 100$ms and the pattern-to-pattern offset of 400ms, and could likely be changed by further increasing the time between pattern presentation or by using some improved model for dopamine diffusion within striatal tissue in future iterations of the model. As an optimal model for dopamine diffusion is not the main topic of this work, however, we do not further optimise the time course of dopamine concentrations here.

Figure 10.6d shows that some postsynaptic neurons (neurons #1, #3 and #5) soon form selective receptive fields within the first 200 seconds of simulation, while others (neurons #2 and #4) remain unselective throughout the full 20 minutes of simulated time. This can be further seen in the development of all 100 postsynaptic neuron's receptive fields in Figure 10.6e.

The final receptive fields of each of the 100 postsynaptic neurons is shown for ten repetitions in Figure 10.6f. Only a fraction of postsynaptic neurons have become selective to the presented input pattern in each repetition, with a slight bias towards those postsynaptic neurons that were initialised first and could therefore reach the critical period (Section 5.3) earlier than postsynaptic neurons that were initialised later during the first 100s of simulation. However, the order of initialisation does *not* ensure that a neuron indeed forms a selective receptive field, as evidenced by Figure 10.6d.

Also, the number of postsynaptic neurons that finally possess a selective receptive field is variable. For example, in repetitions 5 and 7 (Figure 10.6f), a larger number of postsynaptic neurons has become selective than e.g. in repetitions 8 to 10. The relative number of selective postsynaptic neurons can likely be made more predictable in future work when an improved self-regulation model for dopamine is conceived and used.

The development of dopamine responses for all ten repetitions of the simulation experiment is shown in Figure 10.6g. There, the dopamine levels around each pattern presentation are shown as a row each for every presentation time (y-axis), with each row showing the level of simulated dopamine from 100ms before to 400ms after pattern onset (x-axis). As the receptive fields shown in Figure 10.6f indicate, the strongest dopaminergic response to pattern onset was seen in repetitions 5 and 7. However, this strong response started later during the simulation than medium and weak dopamine responses.

In summary, we find that the method of neuromodulation for spiking networks that

was introduced in Chapter 8 can successfully be used to form a self-regulatory dopaminergic feedback loop that limits over-learning of repeated polychronous patterns in the absence of any external reinforcement signals. The group of plastic neurons modelled here represents striatal medium spiny neurons with D1-type dopamine receptors (D1R-MSNs), the input units represent either cortical or thalamic inputs, and the self-regulating DA feedback signal represents dynamic responses of SNc neurons and the subsequent diffusion of dopamine within the striatum of the brain's basal ganglia.

10.4 Conclusion

This chapter explored whether a dynamic feedback of dopamine (or similar neuromodulators) may in principle succeed in regulating learning processes in substructures of the basal ganglia. More specifically, we wanted to find out if the known anatomical pathways between the basal ganglia's striatal D1R-neurons and the dopamine-releasing cells in the SNc that project back to the striatum may interact to prevent the over-training of a network of plastic neurons to repeatedly occurring stimuli.

We showed that this interaction can indeed be used to control the formation of sparse receptive fields and thereby possible selection channels in the basal ganglia's striatum, hinting at a possible function for these anatomical connections of the so-called direct pathway in biology (Vogt and Hofmann, 2015b).

Similar mechanisms may also be in place within the indirect pathway, where an anatomical loop from striatal D2R-neurons to the SNc exists via the GPe. However, a possible indirect pathway mechanism could not yet be explored and remains for future work.

Part IV

Epilogue

Chapter 11

Summary and Conclusion

In this last chapter, I summarise the contributions made here to the field of computational neuroscience, give an outlook on future work, and conclude this text (see also Appendices).

11.1 Contributions to the Field

Clarification of Synaptic Drift The behaviour of synapses that implement spike timing dependent plasticity (STDP) and receive noisy input spikes was described. It was clarified that the synaptic drift does not only depend on the full integral of an STDP rule, but specifically on a windowed interval that depends on the distribution of spike pairing distances. For highly correlated inputs, the effective window for spike pairings becomes more narrow. This may invert the direction of synaptic drift from depression into potentiation for more realistic ($\tau_+ < \tau_-$, $A_+ > A_-$) STDP rules (Section 5.1).

Clarification of the Effects of Weight Dependent Scaling of STDP It is not the multiplication with a weight-dependent scaling factor that causes unimodal weight distributions of "multiplicative" STDP, but the existence of a stable fixed point attractor within the range of possible synaptic strengths. A fixed point is only created when potentiating and depressing weight update steps depend on synaptic weight through different rules, respectively. I show this by giving a counter-example that uses weight-dependent scaling of an STDP rule without producing an attractor. The new rule fits experimental data better than many "multiplicative" STDP rules, while behaving more similar to an "additive" rule. I therefore suggest the terms "attractor-based" and "attractor-less" instead of "multiplicative" and "additive" to refer to the weight dependent behaviour of STDP rules (Section 5.2).

A new class of weight-dependent, attractor-less STDP rules I extend the counter-example from above into a new set of useful weight-dependent STDP rules that approach the bounds of a predefined (physical) range of possible synaptic weights softly, instead of using abrupt clipping as in classical "additive" STDP. This is especially useful when synaptic connection strengths approach their minimum, because random spike pairings due to noise have a smaller chance of disrupting sparseness (Section 5.2).

Attractor-based STDP leads to Damaging Competition between Structured Inputs and Synaptic Drift In a large comparison between many attractor-based and attractor-less scaling rules for unbiased as well as negatively biased STDP, I demonstrate that the existence of an attractor prevents STDP from forming a robust selective receptive field in response

to spatiotemporally structured (polychronous) incoming spike patterns if the attractor is reasonably strong. The unimodal weight distribution that is known to result from attractor-based STDP actively impedes the creation of sparse receptive fields. In later chapters, I demonstrate that the identity of strong synaptic connections in a sparse receptive field does not need to be random, but can be precisely predicted when polychronous input spike patterns are used (Section 5.2 and Chapter 6).

Structural synaptic growth can keep a neuron with negatively biased STDP within a critical regime until it settles for a selective receptive field As introduced in Section 5.3, the critical regime is here the period of synaptic weight development during which a postsynaptic neuron is responsive to all inputs in general while not yet having formed a selective receptive field. Instead of attempting to remain in this period through attractor-based STDP, it is rather easily sustainable through a combination of structural synaptic growth and any negatively biased STDP rule. An attractor-less STDP rule with depressing synaptic drift then maintains a unimodal distribution of synaptic weights while input units spike at random, which switches into a bimodal distribution that represents a selective receptive field when inputs (begin to) contain repeating spatiotemporal patterns that can be detected by the plastic neuron. The selectivity of the neuron's receptive field can be measured and used as a slow variable that eventually turns off structural growth. In fact, structural growth is only absolutely necessary when neurons start with very weak incoming synaptic weights that are unable to elicit postsynaptic responses, because STDP requires by definition both pre- and postsynaptic spikes to take effect (Section 5.3).

Detection of polychronous patterns provides more reliable information to subsequent layers The response time of a neuron receiving polychronous inputs depends on the mean strength of synapses within the group of earliest firing inputs. The response delay will therefore contain information on how well the postsynaptic (detector) neuron's receptive field matches the incoming pattern of spikes. This may indicate either how much training a neuron has previously received on a particular stimulus that is represented by the incoming spike pattern, or the signal-to-noise ratio of this presentation of the stimulus. This conundrum may be solved by comparing the response times of multiple postsynaptic neurons (Section 6.1).

Multiple detector neurons can re-create a spatiotemporal code When the map of sparse receptive fields of multiple postsynaptic neurons evenly represents the start of a set of neighbourhood-preserving (ordered) polychronous patterns, noisy polychronous inputs can be reproduced with less noise by a group of detector neurons. This is because the response onset delay of each neuron then represents the degree of match between patterns and receptive fields. (Section 6.1.4).

Traditional Hebbian plasticity cannot take advantage of polychronous spike patterns Synaptic plasticity rules that use only the *absolute* spike pairing difference for changing synaptic weights (sign-independent spike-to-spike distance between pre- and postsynaptic units) can not produce maximally sparse receptive fields (Section 6.2.4).

STDP behaves similar to traditional Hebbian plasticity for synchronous (or zero-lag correlated) spike patterns When (jittered) synchronous spike patterns are used as repeating inputs, attractor-less STDP may randomly select any subset of synapses to become strong. It will not succeed in reliably selecting a predictable subset of synapses (Section 6.2.2). The

resulting spike timing of the postsynaptc neuron will therefore be erratic, allowing only rate-coded readouts. Attractor-based STDP is even worse, as receptive fields never become sparse (see above and Section 6.2.3). Standard zero-lag correlation between input units represents a jittered version of a synchronous code, which therefore also does not allow the development of receptive fields to be predictable.

Only polychronous input patterns combined with STDP allow a neuron to form highly selective receptive fields Polychronous input spike patterns can be used by STDP to find the earliest set of reliably firing input units, also known as predictive learning. What has yet seen little attention is that the anti-symmetric nature of standard STDP rules also actively decreases all connections to other inputs that reliably fire later in a (e.g. synfire) chain. This actively produces a sparse receptive field when all inputs regularly fire in a fixed order, and succeeds even for higher levels of background noise. Inputs that fire without any repeating order (random spiking) reliably reduce connection strength when the STDP rule is negatively biased (= has a depressing synaptic drift). In combination, standard STDP rules that have a negative depressing synaptic drift produce sparse receptive field where only the earliest of the sufficiently reliable inputs form strong synapses with the postsynaptic neuron (Section 6.2, esp. 6.2.1).

Multiple polychronous patterns can signal multiple messages A given set of input units can repeatedly present many distinct polychronous patterns. The postsynaptic neuron uses STDP to tune to the start of exactly one pattern if all patterns are presented by each unit in the group (Section 6.3). The choice of pattern depends on (random or controlled) fluctuations in input population rate at the time of repeated pattern presentation (Sections 7.1.2 and 7.5.2). If fluctuations are random, then the choice of selected pattern is also random (see also Section 7.3).

Polychronous patterns can be generated from any random input noise I show practical ways of generating polychronous patterns without this being reflected in firing rates or simple cross-correlations. I also present various ways of further embedding polychronous patterns within background noise, hence hiding them from the naked eye (Section 7.1).

Multiple polychronous patterns may not show up in cross-correlation analysis of spike trains A time-shifted (nonzero-lag) correlation code can be interpreted as the repeated presentation of a *single* highly jittered polychronous pattern (Section 6.4.4). However, the presentation of multiple polychronous patterns produces varying time shifts depending on which pattern is being presented. A simple cross-correlation analysis between spike trains may therefore be unable to detect multiple patterns when they are being presented during a limited-time (electrophysiological) recording (Section 7.1.6).

Implications of polychronous patterns for new experimental methods Jitter and occasional omission of spikes by units that take part in presenting polychronous patterns is of little importance for the overall detection robustness of polychronous codes by postsynaptic detector neurons. While not synchronous, the code is still redundant, or rather holographic (each unit represents similar, but slightly different information). Also, each unit may fire randomly outside of pattern presentation at any time. This provokes the question whether established experimental techniques such as *evoked responses* or *spike-triggered averages* could be extended into *evoked multiunit responses* or *pattern-triggered averages*.

Multiple neurons can form a sparse receptive map without requiring mutual inhibition As each neuron individually becomes highly selective when STDP is used together with polychronous patterns, the choice of pattern the neuron's receptive field tunes to is randomly selected from a probability distribution that depends on fluctuations in background population rate of all inputs together (Section 7.1.2), other forms of noise (Section 7.4), and relative pattern presentation counts among other factors (Section 7.5). When multiple neurons are given the same input patterns and initialised with identical weights and membrane parameters, tiny background fluctuations in the input population rate decide which pattern a neuron tunes to. When plasticity is activated at a different time for each neuron, the resulting map of receptive fields approaches a broad representation of all presented inputs (Section 7.3). This formation of a sparse code requires no mutual inhibition between neurons. Instead, it depends only on the probability distribution of tuning to each pattern. If the probability distribution is uniform, then the resulting map also represents all patterns uniformly.

Neuromodulation of synaptic transmission can affect learning outcome I show that when neuromodulator level is allowed to affect the contrast of synaptic transmission, the altered behaviour of postsynaptic neurons also affects plasticity. This accounts both for observed instantaneous effects of dopamine on neural excitability as well as for neuromodulator-dependent changes to synaptic strength. Neuromodulation does not need to be directly integrated into an STDP rule to take effect, in contrast to a widespread practice in computational neuroscience and biological reinforcement learning (Chapter 8).

The new method allows for fast pattern-dependent changes through dopamine that produce reinforcement of specific patterns Plasticity need not just be influenced in a general, time-independent manner through modulating synaptic transmission. When neuromodulator levels are changed together with specific (groups of) patterns, this changes the probability of certain patterns being learnt. The resulting receptive map then reflects positive or negative reinforcement that was delivered together with specific (groups of) polychronous patterns (Chapter 9).

Learning can be slowed and paused through precisely-timed dopamine Over-representation of a specific repetitively incoming pattern can be avoided by dropping the level of dopamine below baseline (negative reinforcement) at specific times relative to pattern onset. Both the relative time (Section 10.3.1) and the amount of neuromodulator concentration drop (Section 10.3.2) affect plasticity and the resulting receptive map.

Self-regulatory control of a dopaminergic feedback loop can limit overtraining The precisely timed drop of dopamine concentration relative to the presentation of specific patterns can be autonomously controlled through a feedback loop that is found in the basal ganglia. After some neurons have tuned to a single repeating polychronous pattern in the computer simulation, they begin to inhibit dopamine-releasing cells, which in turn decreases the probability of further neurons tuning to this pattern (Section 10.3.3). This may give a biological agent more robustness to vastly different presentation counts of distinct stimuli.

11.2 Outlook and Future Work

By recombining the biological building blocks of temporal coding, spike timing dependent plasticity and dopamine-like neuromodulation in new models and testing them in a number of computer simulations, we came a little closer to understanding how the brain, and especially the basal ganglia, may work. Besides further improving and inventing future computer models of brain function, the next step is also to apply our new knowledge to experimental paradigms and biological data analysis. I now present some main open questions and future directions of research.

On Unsupervised Learning in Spiking Networks

Learning Polychronous Patterns with Inhibition Although we have seen that STDP can use polychronous patterns to form a sparse receptive field without mutual inhibition, recurrent connections do exist in the brain. Why could this still be useful to a group of neurons that use STDP to learn a polychronous code? Multiple answers come to mind. If two otherwise distinct patterns share the same group of early firing units, the standard behaviour of STDP is to tune any postsynaptic neuron's receptive field to the earliest part of both patterns. If a first neuron that tunes to the common set of inputs were to inhibit all other potential candidates until the input patterns diverge, any additional postsynaptic neurons may each remain tuned to the later parts of each pattern. This was partly approached by Masquelier et al. (2009), albeit assuming that mutual inhibition was necessary for any development of distinct receptive fields. A second but similar purpose may be to use mutual inhibitory connections to indeed maximise independence when polychronous patterns are otherwise very similar.

New analysis methods for biological data The insight that STDP only reveals its full power when polychronous patterns are used as inputs has some implications for the analysis of real biological data. When (constrained) spike timing dependent plasticity is found to exist in any brain area, we should consider the strong possibility that temporal coding through relative spike times of a large group of neurons may play a role here. This assumption may prove helpful for the analysis of experimental recordings. For example, we may want to consider extending the well-known method of spike-triggered averages, which finds the average stimulus that happens before each spike of a specific recorded neuron. This could be extended into a paradigm that first uses artificial neurons with STDP to learn to detect noisily repeating spatiotemporal patterns within the stream of spikes, and then use the occurrence of these patterns as a trigger to recreate the average stimulus that was present whenever a given polychronous pattern of spikes passes through the buffer of (freshly) recorded data. A similar extension could be made to the practice of evoked responses, where a stimulus is repeated multiple times and the average responses of single units are tracked. When we assume that neurons respond very unreliably to inputs, with detectability only minimally above noise level, the observation of a large group of neurons may similarly allow us to detect a polychronous spike pattern even though the background population rate may be highly fluctuant (Section 7.1.2). A single recorded neuron may either represent varying parts of a repeating pattern, which would show as large jitter in recorded spike timings, or not respond at all on all but a few stimulus presentations (Section 7.1.4). It will be interesting to transform these ideas into new experimental paradigms in the future.

Embedded Implementation and Neuromorphic Hardware Artificial neurons receiving real-time electrophysiological data arriving from multi-electrode recording systems may prove useful as an online analysis method that learns repeating polychronous patterns of neural spiking activity on the fly. An embedded implementation on small neuromorphic chips may make it possible to integrate spike train analysis directly into electrode probes or build fully implantable chips that support or replace brain functions that have been lost through disease or injury. However, a number of problems still need to be solved before this can be tried, the most important of which is the formulation of a concept of neural codes throughout the brain as well as binding ethical rules on the circumstances under which interventions should be allowed.

On Reinforcement Learning in Spiking Networks

D2R-type feedback loop As we have seen in the introductory chapters of this text, the basal ganglia consist of a number of reentrant loops, many of which interact with the striatum and the release of dopamine. While a possible use for the direct pathway loop was evaluated in Chapter 10 as a mechanism for dopamine self-regulation, we have yet to explore the possible effects of an indirect pathway loop, which consists of projections from striatal D2R-neurons to the GPe and from there to the SNc, which in turn also changes the amount of dopamine being released in the striatum. As the indirect pathway (D2R-type) neurons of the striatum are often described as reacting to dopamine in a way that is opposite to direct pathway (D1R-type) neurons, the existence of a second inhibitory projection in this loop (via the GPe) may have a similar self-regulatory effect on D2R neurons of the indirect pathway as the direct inhibitory projections of D1R-type neurons to SNc have for the direct pathway. The stability of such an indirect pathway dopaminergic feedback loop though needs to be evaluated both alone and in combination with other known basal ganglia interactions, especially that between GPe and STN.

Build a reliable act-or-delay network With the knowledge gained from dopamine self-regulation in the direct and indirect pathways and assuming that the feedback loop between STN and GPe indeed acts to integrate uncertainty in the cortex and the striatum, we may build a spiking neural network model of the basal ganglia that decides to either act quickly or delay action in response to behavioural contexts. Depending on incoming stimuli, a biologically realistic implementation of such functionality will wait for sufficient information to arrive in order to make the best decision when enough time is available, or make quick (impulsive) decisions when decision time is constrained. The switch between slow good decisions and fast impulsive decisions should be gradual and depend on an evaluation of the current world context.

Implement skill learning in a biologically more plausible manner With the knowledge of how to construct a network that only transitions between behavioural states when there is reason to do so, we may be able to implement a true biologically realistic reinforcement learner that takes meaningful advantage of structures and dynamics of the basal ganglia. The biological implementation of multiple skills, the swift switching between them, and easy generalisation towards performing completely new tasks may be understood through this approach. A first reinforcement learning task that could be performed with this new method could be the widespread Morris watermaze task (Morris et al., 1982) as used in Vasilaki et al. (2009). Extensions could then involve the acquisition of multiple skills in one environment and their application to solving tasks in a second environment (Barto et al., 2004; Singh et al., 2005; Sutton et al., 1998). While aiming to reproduce the full range

of basal ganglia functionality through biologically realistic mechanisms, we will be vastly extending our knowledge on why the basal ganglia and connected brain structures are formed the way they are, and may also find new efficient forms of reinforcement learning for use in autonomous agents.

More realistic implementation of neuromodulator-dependent contrast adjustment While the observations of dopamine affecting instant excitability and contrast have led to the construction of the dopamine-dependent synaptic transmission paradigm that was introduced in Chapter 8, we have in this work concentrated on introducing and evaluating the basic principle of reward-modulated synaptic transmission. However, it seems promising to further increase the biological realism of neuromodulated transmission of electric membrane potentials from axon via synapse to dendrite in improved models. Apart from demonstrating the effects of this on synaptic plasticity, more realistic models may also provide a sub-cellular explanation for why a dynamic modulation of contrast through dopamine may be a viable solution to including external reward into dynamical processes in the brain.

Roles for striatal interneurons, other subpopulations, other neurotransmitters It is still unclear which functions the other neuron types in the basal ganglia fulfil. Acetylcholine is released by tonically active neurons, and has strong but yet hardly explored effects on dopamine release from nigral terminals in the striatum (Calabresi et al., 2000). Similarly, the interaction between dopamine and serotonine is still unclear, as is the effect of fast-spiking interneurons and newly isolated projections from a subpopulation of GPe neurons back to the striatum (Mallet et al., 2012). We need to understand the various elements of this highly dynamic system if we want to separate functionally significant features from replaceable implementation details. A true description of the functionality of the striatum and the rest of the basal ganglia can only be agreed upon when at least all dynamical effects and anatomical substructures are accounted for.

11.3 Conclusion

In the present work, I have explored and recombined experimentally found building blocks of brain function on a level of abstraction that seems the most likely candidate for linking higher cognitive functions to biochemical implementations of supporting capacity. I used plasticity in spiking neural networks to examine the effects and implications of temporal codes that use unreliable spikes to reliably transmit and process messages in high speed. Within the model framework of the basal ganglia, I then found an alternative way of modulating the outcome of spike timing dependent plasticity through controlling the contrast of synaptic transmission and showed that this can be used to reinforce specific actions / patterns / stimuli in relation to others. Future work includes the construction of a plastic spiking neural network that models the full basal ganglia and is able to perform typical reinforcement learning tasks as seen in behavioural neuroscience and machine learning. While many questions have been answered here, many more have been and will be spawned and resolved as a consequence of this work.

Part V

Appendices

Appendix A

Supplementary Data and Simulation Parameters

A.1 Supplementary Figures

A.1.1 Additive STDP (Section 6.2)

In addition to the attractor-less weight-dependent scaling rule used in Section 6.2.1 (Equation 5.9), similar figures can be produced by using attractor-less STDP that is mostly not dependent on synaptic weight except for a hard clipping at the bounds of the defined range ("additive" STDP, Equation 5.3). See Figures A.1 and A.2 for examples using purely "additive" STDP with a biased and unbiased shape, respectively. Simulation settings are listed and further explained in Appendix A.3.5.

A.1.2 One-sided and randomly permuted patterns (Section 6.3)

When multiple polychronous patterns are used together with STDP, it needs to be ensured that the start of each pattern is not equally part of some other pattern (Sections 6.3 and 7.1.7). If in Figure 6.27 (p. 99) we had used the special case of one-sided polychronous patterns instead of two-sided patterns, the set of strong weights would not settle and keep shifting (Figure A.3). However, the probability of such similar patterns happening by chance can be controlled by increasing the number of input units. Figure A.4 shows polychronous patterns created through fixed random permutations (Section 7.1.5) and the single postsynaptic neuron successfully tunes to exactly one of the presented polychronous patterns. The patterns are invisible to the naked eye (row 1), but the response plots show the definitive preference.

A.1.3 Learning multiple randomly permuted patterns (Section 7.1.5)

Pattern generation through fixed random permutations was also tested (see Figures A.4 and A.5). Learning progresses slower because single input units that fire at different times for different patterns may initially contradict each other. However, given sufficient time, a postsynaptic neuron with a negatively biased STDP rule will learn any repeatedly presented polychronous pattern that is sufficiently different from other repeatedly presented patterns (Section 7.1.7).

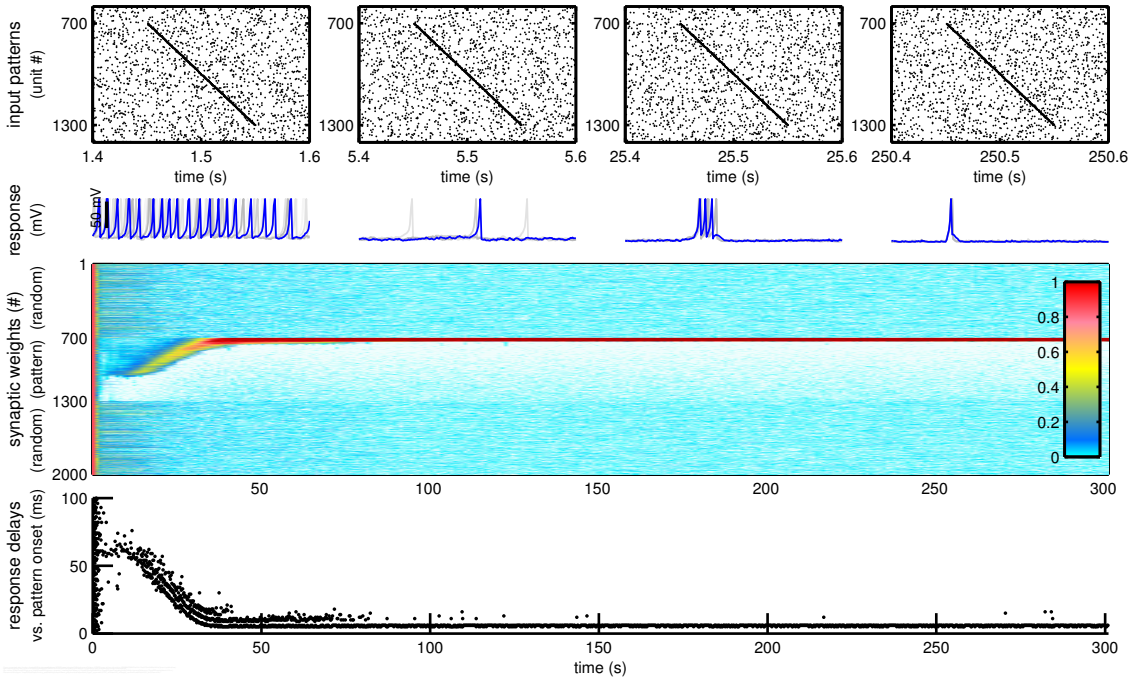


Figure A.1: **Polychronous inputs** with **attractor-free** biased STDP as in Equation 5.3. See Figure 6.12 for description of subplots and Appendix A.3.5 and A.1.1 for simulation settings. The formation of a highly selective receptive field is successful. By chance, the typical behaviour of STDP is also nicely visible in this figure: The postsynaptic neuron here coincidentally began to respond to input units near #1100, and then proceeded towards the start of the polychronous pattern as mentioned by Guyonneau et al. (2005) and in Section 6.2.1.

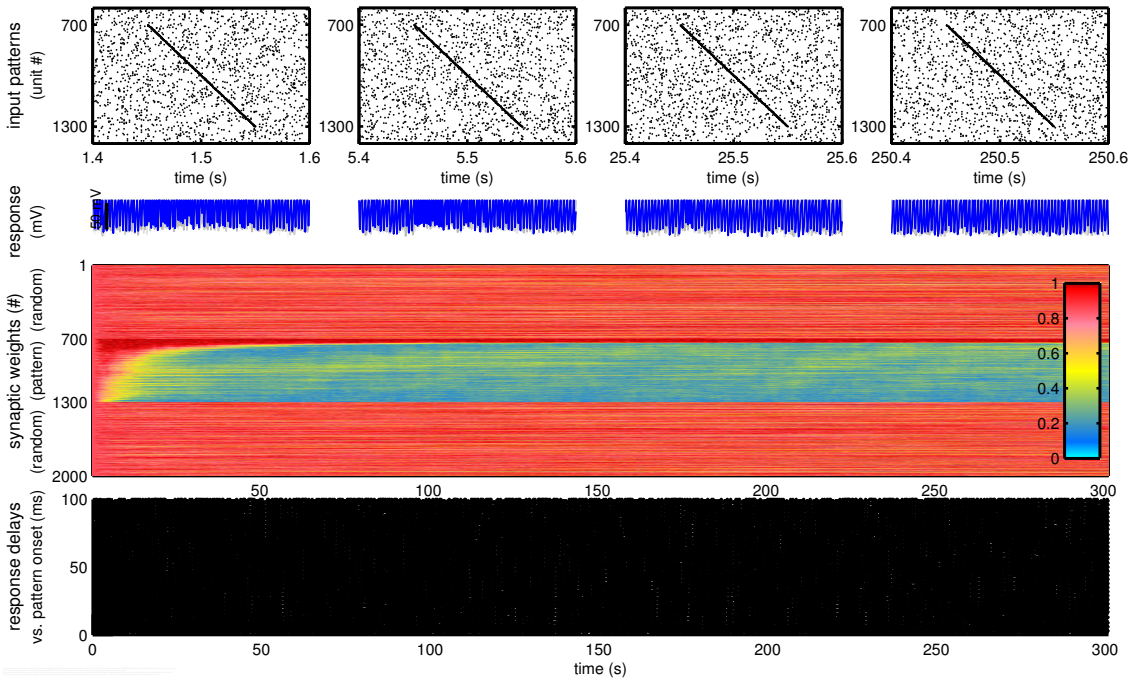


Figure A.2: **Polychronous inputs** with **attractor-free** unbiased STDP as in Equation 5.3. See Figure 6.13 for description of subplots and Appendix A.3.5 and A.1.1 for simulation settings. While the connections that receive polychronous inputs aim to become somewhat selective, the lowest weights remain around 0.25 and never come close to zero. Random inputs perform even worse. No selective receptive field is achieved.

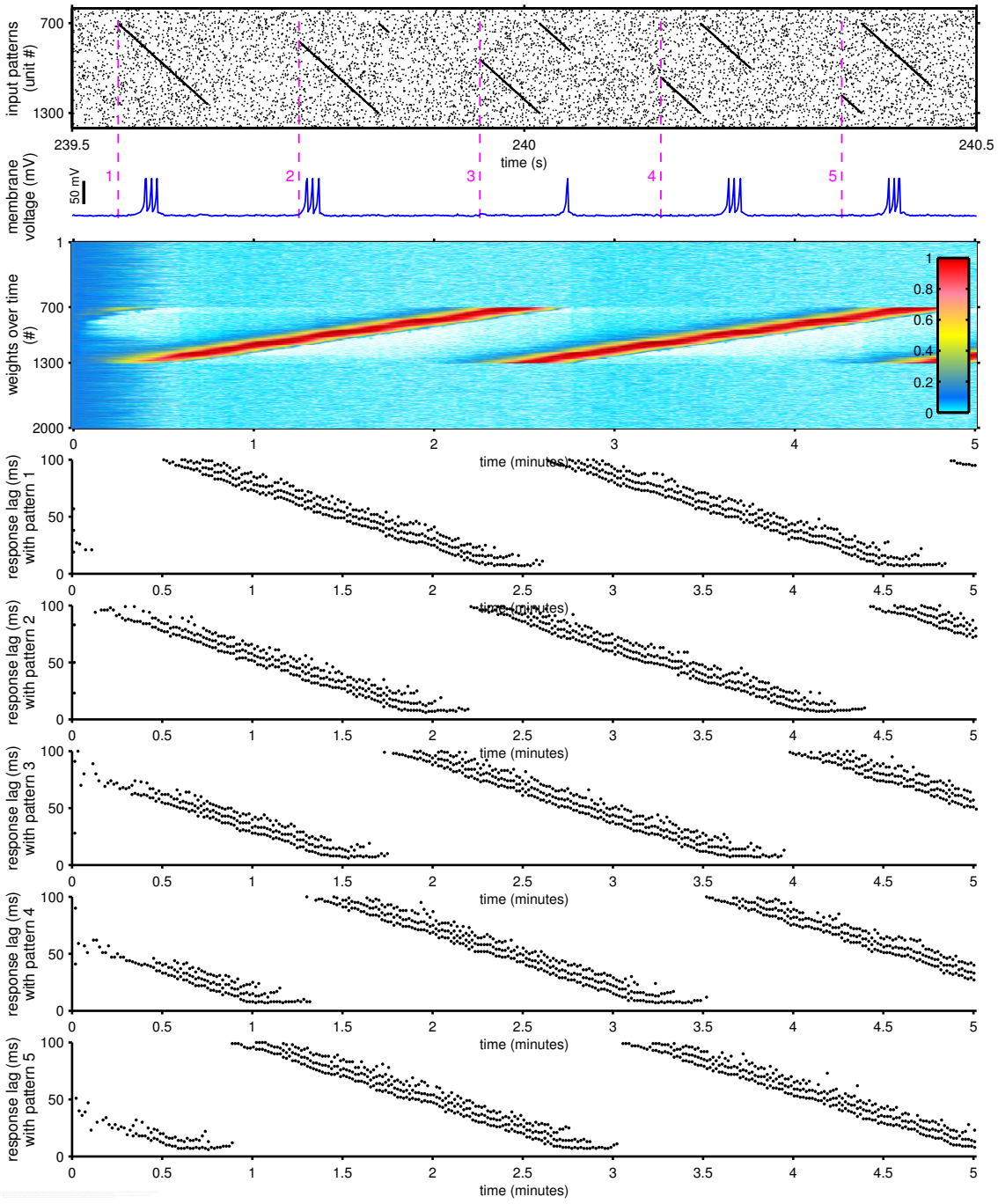


Figure A.3: Polychronous patterns generated in the one-sided ordered variant do not allow a postsynaptic neuron to form a stable receptive field (compare Figure 6.27, p. 99). This is because the early-firing units of each pattern also fire together at later stages of each other pattern. See Sections 6.3 and 7.1.5 for explanation. See also Appendix A.3.6 (p. 187) for simulation details.

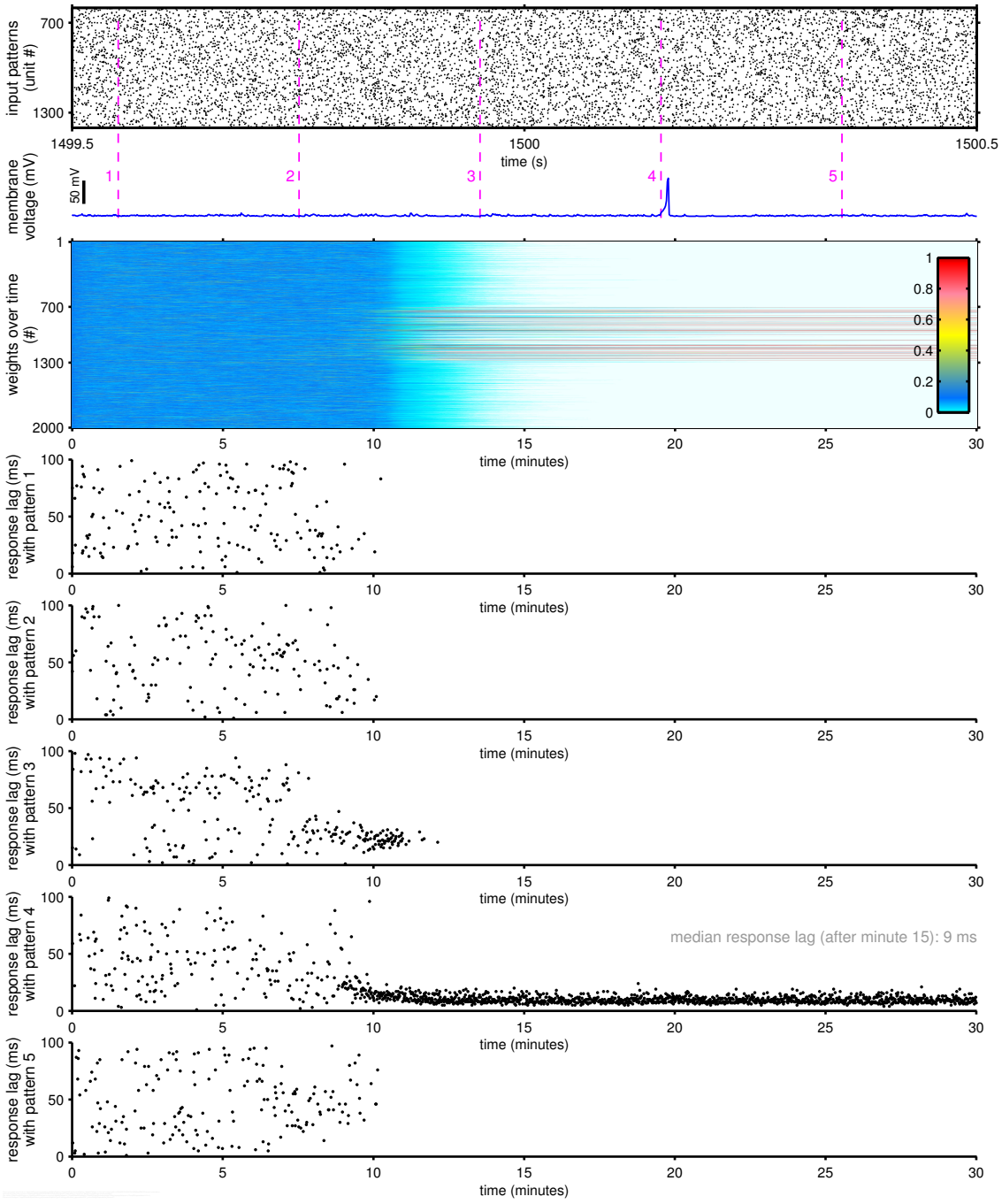


Figure A.4: Five polychronous patterns generated with fixed random permutations (Section 7.1.5) are presented to a single postsynaptic neuron (compare Figure 6.27). See Appendix A.3.6 (p. 187) for simulation details. The neuron chooses one pattern (pattern 4) and begins to respond robustly and selectively only to this one pattern when STDP is used to form the receptive field. (Around minute 10, some close competition between patterns 3 and 4 can be observed, which pattern 4 wins.)

A.1.4 More examples of effective weights (Section 8.2)

The example effective weight distributions shown in Figure 8.1 were computed for $\theta = 0.5$. Figure A.6 shows the same transformations for different values of θ.

A.2 Supplementary Tables

A.2.1 STDP rules in the literature

The tables A.1, A.2 and A.3 list the STDP settings of a number of publications for reference. Negative synaptic drift is often invoked by scaling $A_+ < A_-$ and less often by using $\tau_+ < \tau_-$. A combination of $A_+ > A_-$ with $\tau_+ < \tau_-$ is hardly ever used, likely due to the added complexity of STDP behaviour.

A.2.2 Biased and Unbiased STDP

Independent of any weight-dependent scaling terms g_+ and g_-, the interaction between A_+, A_-, τ_+ and τ_- already leads to synaptic drift. Throughout this work, I use the terms unbiased STDP and biased STDP to refer to two sets of configurations as shown in table A.4.

	λ	$A+$	$A-$	τ_+	τ_-	drift
(Bi and Poo, 2001, 1998)						
(Froemke and Dan, 2002)		101%	52%	14.8 ms	33.8 ms	–
(Kistler and van Hemmen, 2000)		1 ?	1 ?	20 ms	20 ms	0 (?)
(Song et al., 2000)	0.005	1	1.05	20 ms	20 ms	0
(van Rossum et al., 2000)				20 ms	20 ms	0 (?)
(Rubin et al., 2001)a	0.005	1	1.05	10 ms	10 ms	–
(Rubin et al., 2001)m						
(Billings and van Rossum, 2009)n	0.005	1	1.05	20 ms	20 ms	–
(Billings and van Rossum, 2009)w	0.005	1	2.28	20 ms	20 ms	–
(Izhikevich et al., 2004)	0.004	1	1	15 ms	20 ms	–
(Izhikevich, 2006)	0.1	1	1.2	20 ms	20 ms	–
(Izhikevich, 2007b)	1	1	1.5	12.5 ms	12.5 ms	–
(Gütig et al., 2003)		1 (?)	1.05 (?)	?	?	– (?)
(Gilson et al., 2010)						
(Morrison et al., 2007)	0.1	1	0.11	20 ms	20 ms	+
(Guyonneau et al., 2005)		1	1	20 ms	22 ms	–
(Masquelier et al., 2008)	0.03125	1	0.85	16.8 ms	33.7 ms	–
(Vogt and Hofmann, 2012)	0.03125	1	0.85	16.8 ms	33.7 ms	–

Table A.1: STDP model parameters in the literature. λ scales $A+$ and $A-$ equally, so is a learning rate. See also Tables A.2 and A.3 and Section A.2.1.

A.3 Figure Settings

A.3.1 Effect of Leak (Figures 2.5 and 2.6)

The model neuron used here was the one-dimensional Izhikevich neuron with default parameter settings. See Table A.5. Synaptic weights and input currents were chosen to produce good example behaviour for didactics.

A.3.2 Synaptic Drift (Sketch 5.2)

The settings used in Sketch 5.2 can be seen in Table A.6, which includes a third set of STDP settings for the central column in the figure.

A.3.3 Comparison Charts (Figures 5.6, 5.7, 5.14, 5.15)

Figures 5.6 and 5.7 show the effects of weight-dependent update bounding rules on unbiased and biased STDP, respectively. While the weight dependence is only seen at the range borders for additive bounding (column A), its effects are highly visible for one-sided multiplicative (columns B,C) and interpolated, nearly-additive update rules (column D).

The *first row* of each figure plots each weight bounding rule from Equations 5.3 to 5.6. Red lines are used for the weight-dependent bound $g_+(w)$ on potentiating steps while blue lines show $g_-(w)$ on depressing steps. Purple lines indicate balanced bounding rules where $g_+(w) = g_-(w)$. Underneath this, *row 2* shows five realisations of effective STDP rules under influence of weight dependent bounding for weights $[\frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}, \frac{5}{6}]$. Note how the effective shape of the STDP rule and especially the ratio between LTP and LTD (for multiplicative rules) changes with the synaptic weight. *Row 3* shows the gradient of weight-dependent STDP updates within a timing window of [-50,50] ms over the full range of possible synaptic weights. The colors show the size of update steps for a given weight and

	window	pairing	$g_+(..)$	$g_-(..)$	attractor
(Bi and Poo, 2001, 1998)	$[-50, 50]$ ms (?)	nn ?	1	w	yes
(Froemke and Dan, 2002)	$[-100, 100]$ ms	other	?	?	?
(Kistler and van Hemmen, 2000)		nn	1 - w	w	yes
(Song et al., 2000)		a2a			
(van Rossum et al., 2000)		nn	1	w	yes
(Rubin et al., 2001)a		?	1 - w	w	yes
(Rubin et al., 2001)m					
(Billings and van Rossum, 2009)n	$[-5\tau_-, 5\tau_+]$ ms	?	1	1	no
(Billings and van Rossum, 2009)w	$[-5\tau_-, 5\tau_+]$ ms	?	1	w	yes
(Izhikevich et al., 2004)	$[-50, 50]$ ms	other	1	1	no
(Izhikevich, 2006)	$[-50, 50]$ ms		1	1	no
(Izhikevich, 2007b)	$[-50, 50]$ ms		1	1	no
(Gütig et al., 2003)			$(1-w)^\mu$	w^μ	yes
(Gilson et al., 2010)			$(1-w)^\mu$	w^μ	weak
(Morrison et al., 2007)	$[-\infty, \infty]$ ms (?)	a2a	w^{μ_1}	w^{μ_2}	yes
(Guyonneau et al., 2005)	$[-\infty, \infty]$ ms (?)	a2a ?	1	1	no
(Masquelier et al., 2008)	$[-7\tau_-, 7\tau_+]$ ms	nn	1	1	no
(Vogt and Hofmann, 2012)	$[-50, 50]$ ms	a2a	$\sin(\pi w)$	$\sin(\pi w)$	no

Table A.2: STDP model parameters in the literature. Spike consideration rules ("pairing") are all-to-all (a2a), nearst neighbour (nn), or "other". See also Tables A.1 and A.3 and Section A.2.1.

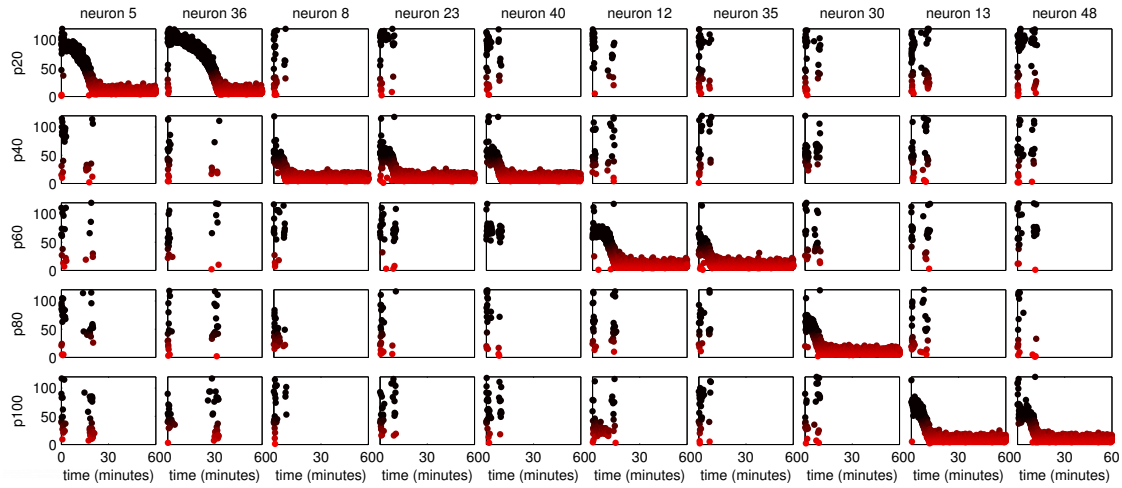


Figure A.5: Response delay plots of 50 neurons that have each learnt to respond to exactly one of five patterns. Y-axis signals response delay in milliseconds. Short latencies are indicated in *red*. Only ten neurons were shown for avoiding clutter, but every neuron became responsive to exactly one pattern. No mutual inhibition was used. Patterns were generated with method D, while the order of firing was randomly permuted for each of the five patterns (Section 7.1.5). Simulation parameters were tracked in –local:*parsim2014-8-26_20.19.55_results*– and discussed in Appendix A.1.3 and A.3.6.

	remarks
(Bi and Poo, 2001, 1998)	experimental paper ;
(Froemke and Dan, 2002)	experimental paper ; pairing: “spike suppression”
(Kistler and van Hemmen, 2000)	
(Song et al., 2000)	
(van Rossum et al., 2000)	
(Rubin et al., 2001)a	
(Rubin et al., 2001)m	
(Billings and van Rossum, 2009)n	
(Billings and van Rossum, 2009)w	attractor is just below $w = 0.5$
(Izhikevich et al., 2004)	pairing: “last opposite spike”
(Izhikevich, 2006)	
(Izhikevich, 2007b)	low (1Hz) input rate, so a2a $\approx$ nn.
(Gütig et al., 2003)	μ scales between "additive" and "multiplicative" STDP
(Gilson et al., 2010)	attractor is very weak because $\mu << 1$
(Morrison et al., 2007)	$\mu_1 = 0.4$ and $\mu_2 = 1$. Attractor vanishes for $\mu_1 == \mu_2$
(Guyonneau et al., 2005)	no clipping, but sigmoid mapping to [0,1] for projection
(Masquelier et al., 2008)	standard additive STDP with clipping to [0,1]
(Vogt and Hofmann, 2012)	own publication

Table A.3: STDP model parameters in the literature. Additional remarks. See also Tables A.1 and A.2 and Section A.2.1.

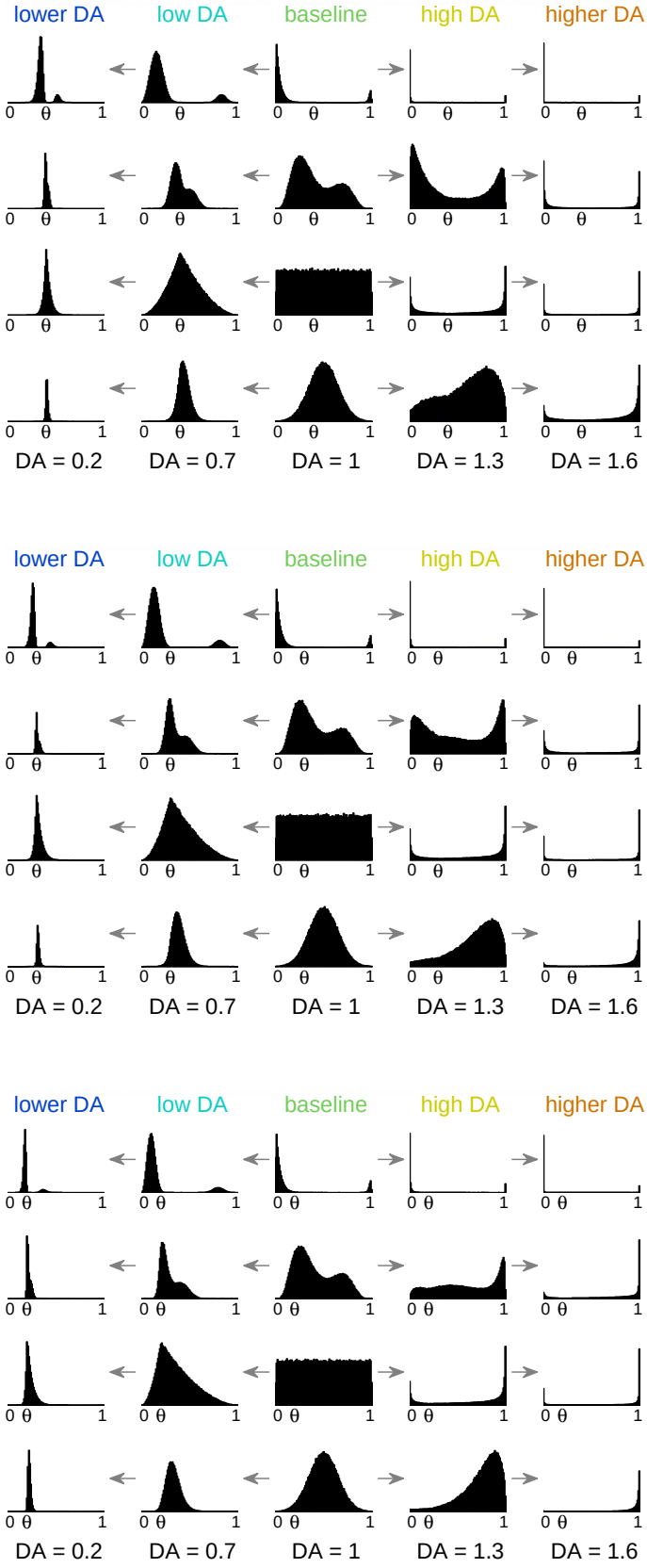


Figure A.6: Effective weights for different values of θ than in Figure 8.1 (p. 126). Range is always $r = 5$ (Equations 8.3 and 8.4). *Top:* $\theta = 0.4$. *Middle:* $\theta = 0.3$. *Bottom:* $\theta = 0.2$. Note that at baseline levels of DA ($DA = 1$), effective weights always remain unchanged from baseline weights.

variable	value
λ	1/32
A_+	1
A_-	1
τ_+	20 ms
τ_-	20 ms
$g_+(w)$	Equation 5.9
$g_-(w)$	Equation 5.9

(a) Unbiased STDP

variable	value
λ	1/32
A_+	1
A_-	0.85
τ_+	16.8 ms
τ_-	33.7 ms
$g_+(w)$	Equation 5.9
$g_-(w)$	Equation 5.9

(b) Biased STDP

Table A.4: Default settings when referring to unbiased and biased STDP. Weight-dependent scaling functions $g_+(w)$ and $g_-(w)$ only apply when not otherwise noted in the text.

variable	value
a	0.02
b	0.2
c	-65
d	2
V_{init}	-65
U_{init}	-13

(a) Standard (two-dimensional)

variable	value
a	n.a.
b	n.a.
c	-65
d	n.a.
V_{init}	-65
U	-13 (constant)

(b) One-dimensional

Table A.5: Default parameters for the Izhikevich neuron (Izhikevich, 2003, 2004) used throughout this work. If not otherwise mentioned, the one-dimensional version is used.

variable	value
λ	1/32
A_+	1
A_-	1
τ_+	20 ms
τ_-	20 ms

(a) Unbiased (U)

variable	value
λ	1/32
A_+	0.85
A_-	1
τ_+	20 ms
τ_-	20 ms

(b) Negative bias (B1)

variable	value
λ	1/32
A_+	1
A_-	0.85
τ_+	16.8 ms
τ_-	33.7 ms

(c) Negative bias (B2)

Table A.6: Settings for the three STDP rules shown in Sketch 5.2.

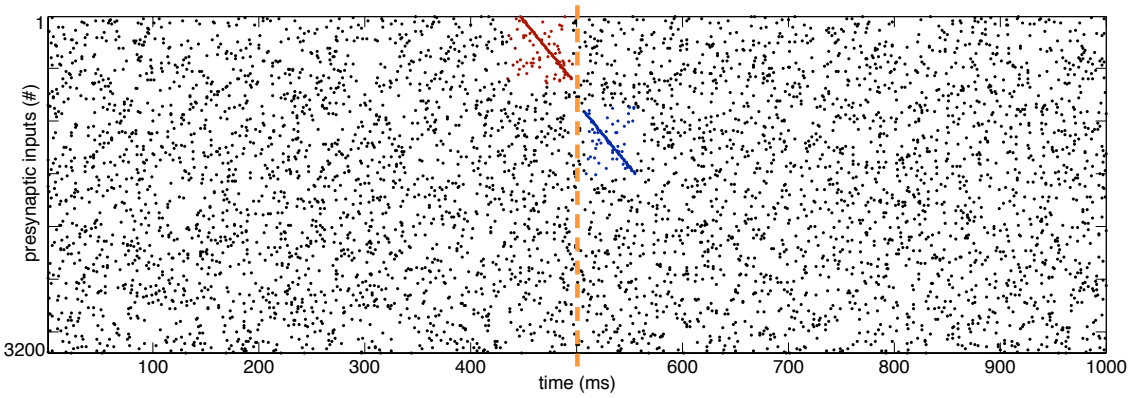


Figure A.7: Input data for weight bounding test simulations. Most input units fire uncorrelated poisson-distributed spikes, but two small groups of units also present polychronous patterns either shortly before (first group) or shortly after (second group) each regularly occurring postsynaptic spike (indicated by orange dashed line). For more information see Appendix A.3.3. Data taken from simulation, colours added manually.

pairing difference, where red/yellow is used for potentiating steps, blue/cyan for depressing steps, and green for very small or zero-size update steps. Note how "multiplicative" rules are much less balanced than additive or additive-like rules. In *row 4*, random pairings (uniformly distributed) of spikes arriving at a given synapse visualise the relation of large to small weight update steps, and the grainy nature of stochastic pairings for stochastically firing input or output neurons. The weight-dependent tendency for overall potentiation vs. depression (synaptic drift) can also be vaguely noticed here. *Row 5* gives a closer view of overall drift tendency, or mean weight update, from row 4 (black line) together with standard deviation (yellow line, scaled by $\frac{1}{5}$). Potentiating drift (positive sign of mean step size) is shown as red area and depressing drift (negative sign of mean step size) is shown as blue area. Mean values were computed for each of 200 weight bins from 20000 random pairings per bin.

I ran example simulations for each bounding rule that show the development of synaptic weights over the time course of 1200 seconds (20 minutes). 3200 presynaptic inputs produced stochastic, poisson-distributed spikes while one postsynaptic unit was controlled to regularly fire at exactly 1 Hz. In addition, two groups of 600 inputs each were overlaid with additional spikes happening exactly before (55ms to 5ms) and exactly after (-5ms to -55ms) each postsynaptic spike. The arrangement of inputs can be seen in Figure A.7. *Row 6* of Figures 5.6 and 5.7 shows an example simulation with the given STDP rule (unbiased or biased) for each weight bounding rule and a learning rate around 0.03 as used in (Guyonneau et al., 2005) and 1ms time step. Weight distribution histograms of all 3200 weights for each second of simulation are shown as rows of grey pixels (background). Example biased random walks/drifts of ten inputs per group from causally (red) and anti-causally (blue) firing unit groups are overlaid to show the divergence of synapses for structured inputs relative to unstructured inputs. Note that the two groups with fixed-order pairings also contain mostly random events, hence there is a competition between random pairings and ordered pairings in these synapses. *Row 7* shows the resulting distribution histograms after 20 minutes of simulation. Synaptic weights to uncorrelated inputs (relative to postsynaptic firing) that do not belong to any of the two input groups are shown in grey, weights to units in groups with positively shifted correlated inputs ("usually before") shown in red, and weights to inputs with a dominance in negatively shifted correlations to postsynaptic firing ("usually after") shown in blue. Histograms stacked in upward order: grey → red → blue.

A.3.4 Detection Features (Figures 6.1, 6.2, 6.3, 6.4, 6.5, 6.6, 6.7, 6.8)

Simulation parameters for Figures 6.1, 6.2, 6.3, 6.4, 6.5 and 6.6 are given in Tables A.9 - A.12. The parameters of Figures 6.7 and 6.8 are given in Table A.13 and A.14, respectively. The source code function and git revision that produced the figures can be seen in each figure's lower left corner in light grey (zoom in via electronic version of this document).

A.3.5 STDP and pattern families (Figures 6.12 - 6.22, 6.25, A.1, A.2)

This appendix subsection lists the simulation settings used to generate Figures 6.12, 6.13, 6.15, 6.14, 6.17, 6.16, 6.19, 6.18, 6.21, 6.20, 6.23, 6.22, 6.25, A.1, and A.2.

Common Settings

Most simulations described here use identical settings. Table A.15 shows the settings used in Figures 6.13 and 6.12, which also serve as the default values for all other figures

variable	value
N_{smooth_total}	600 units
w_{weak}	0.021
w_{medium}	0.5
w_{strong}	1.0

Table A.7: Settings for simulations of Figures 6.1, 6.3 and 6.5. More settings in Tables A.9, A.10 and A.11.

variable	value
N_{binary_total}	600 units
N_{few}	13 units
N_{some}	300 units
N_{many}	600 units

Table A.8: Settings for simulations of Figures 6.2, 6.4 and 6.6. More settings in Tables A.9, A.10 and A.11.

variable	value
pattern family	synchronous
pattern type	n.a.
pattern shape	n.a.
pattern duration	1 ms
inter-pattern spacing	199 ms

Table A.9: Additional settings for simulations of Figures 6.1 and 6.2. More settings in Table A.12.

variable	value
pattern family	correlated
pattern type	n.a.
pattern shape	n.a.
pattern duration	100 ms
inter-pattern spacing	100 ms

Table A.10: Additional settings for simulations of Figures 6.3 and 6.4. More settings in Table A.12.

variable	value
pattern family	polychronous
pattern type	A
pattern shape	ordered (one-sided)
pattern duration	100 ms
inter-pattern spacing	100 ms

Table A.11: Additional settings for simulations of Figures 6.5 and 6.6. More settings in Table A.12.

variable	value
N_{inputs}	2000 units
pattern-presenting input group size	600 units
background noise source	homogeneous poisson 0.2 Hz
number of distinct patterns	1
projection multiplier	$3000/N_{inputs}$
$w_{nonpresenting}$	0.0
$N_{outputs}$	1 model neuron
output neuron model	default Izhikevich-1D

Table A.12: Common settings for all simulations of Figures 6.1, 6.2, 6.3, 6.4, 6.5, and 6.6.

variable	value
N_{inputs}	2000 units
pattern-presenting input group size	600 units
background noise source	homogeneous poisson 0.2 Hz
number of distinct patterns	5
pattern family	polychronous
pattern type	A
pattern shape	ordered (two-sided)
pattern duration	100 ms
inter-pattern spacing	100 ms
projection multiplier	$6000/N_{inputs}$
w_{other}	0.0
$w_{notconnected}$	0.0
$w_{connected}$	1.0
$N_{connected}$	18 units
$N_{outputs}$	100 model neurons
output neuron model	default Izhikevich-1D

Table A.13: Parameter settings for the simulation of Figure 6.7.

variable	value
N_{inputs}	2000 units
pattern-presenting input group size	600 units
background noise source	homogeneous poisson 25 Hz
number of distinct patterns	5
pattern family	polychronous
pattern type	D
pattern shape	ordered (two-sided)
pattern duration	100 ms
inter-pattern spacing	100 ms
projection multiplier	$6000/N_{inputs}$
w_{other}	0.0
$w_{notconnected}$	0.0
$w_{connected}$	1.0
$N_{connected}$	18 units
$N_{outputs}$	100 model neurons
output neuron model	default Izhikevich-1D

Table A.14: Parameter settings for the simulation of Figure 6.8.

described here.

The *projection multiplier* regulates the effect that each arriving presynaptic spike can have on the postsynaptic membrane potential. This accounts for the fact that synaptic weights are normalised to the range $w \in [0, 1]$ throughout this work. The projection multiplier can also be used as a static handle to manually down-regulate the maximum effect incoming spikes can have on the postsynaptic membrane when the STDP scaling rule being used produces a large number of strong synapses. Without adjustments to the projection multiplier in Tables A.20, A.21, and A.22, the STDP variants presented there would do even worse.

Settings for Figures 6.13 and 6.12

As mentioned above, Figures 6.13 and 6.12 directly use the settings shown in Table A.15. The creation of pattern type A is described in Section 7.1.1 (p. 104). During the 100 ms of pattern presentation, each of the 600 pattern-presenting input units (group A) fires exactly once, in addition to the background noise already produced by the homogeneous poisson process. This increases the firing rate of the input layer by 3 Hz during pattern presentation. Other pattern generation types that do not affect firing rate are also shown in Section 7.1.1, but here a pattern-dependent change of firing rate is more close to the following methods of input generation.

Settings for Figures 6.14 and 6.15

Synchronous input pattern are created by simply decreasing the duration of patterns to the simulation step size of 1 ms. This produces a synchronous "action potential" in 600 of the 2000 total input units, leading to a strong spike in the input group's firing rate. Inter-pattern spacing is adjusted accordingly. See Table A.16 for non-default parameters.

Settings for Figures 6.16 and 6.17

Correlated input patterns can be generated by a common increase of firing rate at the beginning of each pattern and a common decrease of firing rate at the end of each pattern. While each input unit of group A still fires exactly once, the timing is now random within each presentation of a pattern. The only reliable timing that remains in the simulation is therefore the start and end of each pattern presentation. See Table A.17 for non-default parameters.

Settings for Figures 6.18 and 6.19

We can also remove the last piece of reliable timing information from the inputs. By removing pattern timing completely, STDP can only use differences in firing rate to guide plasticity. See Table A.18 for non-default parameters.

Settings for Figures 6.20 and 6.21

An alternative way of implementing correlated inputs is to use a inhomogeneous poisson process. The Ornstein-Uhlenbeck is a simple biased random walk process that returns to some predefined central value. As the drifting value here is the target firing rate of the spike-generating poisson process, we want it to always return to some goal (10 Hz) instead of performing a truly random walk. See Table A.19 for non-default parameters.

variable	value
N_{inputs}	2000 units
input unit group A	600 units
input unit group B	1400 units
group A noise source	homogeneous poisson 10 Hz
group B noise source	homogeneous poisson 10 Hz
pattern onset time	450 ms
pattern family	polychronous
pattern type	A
pattern shape	ordered (one-sided)
pattern duration	100 ms
inter-pattern spacing	100 ms
number of distinct patterns	1
projection multiplier	$3000/N_{inputs}$
$N_{outputs}$	1
output neuron model	Izhikevich-1D
$w_{initCenter}$	0.15
$w_{initRange}$	0
$\tau_{hebbianDecay}$	0 ms

Table A.15: Default settings for all simulations shown in the figures described in A.3.5, unless otherwise noted. Common STDP parameters are shown in Table A.4.

variable	value
pattern family	synchronous
pattern type	n.a.
pattern duration	1 ms
inter-pattern spacing	199 ms

Table A.16: Non-default settings for the simulations of attractor-less STDP with synchronous inputs shown in Figures 6.14 and 6.15.

variable	value
pattern family	correlated
pattern type	n.a.
pattern duration	50 ms
inter-pattern spacing	150 ms

Table A.17: Non-default settings for the simulations of attractor-less STDP with correlated inputs shown in Figures 6.16 and 6.17.

variable	value
group A noise source	homogeneous poisson 30 Hz
group B noise source	homogeneous poisson 10 Hz
pattern onset time	inf
pattern family	n.a.
pattern type	n.a.
pattern shape	n.a.
pattern duration	n.a.
inter-pattern spacing	n.a.

Table A.18: Non-default settings for the simulations of attractor-less STDP with correlated inputs shown in Figures 6.18 and 6.19.

Settings for Figures 6.22 and 6.23

The previous use of attractor-less STDP should be contrasted to the use of attractor-based STDP. As attractor-based STDP with very weak attractors or with attractors very close to the minimum weight ($w = 0$) act very similar to attractor-less STDP, they would provide a weak contrast here. I therefore choose a commonly used weight-dependent scaling rule for STDP that has a strong centrally located attractor (see Chapter 5). See Table A.20 for non-default parameters.

Settings for Figure 6.25

Traditional Hebbian plasticity does not care about spike order. I emulate this by making A_- negative, setting $A_+ = -A_-$ in Equation 5.1. As all activity-related plasticity is now potentiating, I use decaying synapses that decay towards $w = 0$ with a time constant $\tau_{hebbianDecay}$. The reason for using decaying synapses is explained in the main text (Section 6.2.4). The time constant was manually chosen to sufficiently counteract the timing-dependent potentiation, but could likely be automatically computed through consideration of network size, projection multiplier, firing rate of input units, and membrane excitability of the postsynaptic neuron. As this would likely not much change the principal message of Figure 6.25, I did not fine-tune this time constant. See Table A.21 for non-default parameters.

Settings for Figures A.2 and A.1

On a side note, I also compare attractor-less weight-dependent STDP rules as in Equation 5.9 to the common "additive" STDP rule shown in Equation 5.3. The behaviour of "additive" STDP when given polychronous input patterns as shown in Figures A.2 and A.1 is qualitatively similar to that of Figures 6.13 and 6.12. All figures show some degree of tuning to the start of repeating patterns (Section 6.2). While unbiased STDP also shows an increase of many (Figure 6.13) or most (Figure A.2) connections to background units, biased STDP is successful in producing a robust sparse receptive field that can reliably detect pattern occurrence in both figures 6.12 and A.1. See Table A.22 for non-default parameters.

A.3.6 Multiple Patterns (Figure 6.27 and supplementary Figures A.3, A.4, A.5)

Figure 6.27 uses the same settings as noted in Table A.15, except for the changes shown in Table A.23. Figure A.3 also uses the same settings as noted in Table A.15, except for the changes shown in Table A.24. Figure A.4 uses the same settings as defined in Table A.15, except for using pattern type D for 5 patterns and presenting randomly permuted patterns. See Table A.25.

Figure A.5 uses the same settings as Figure A.4, except for using 50 postsynaptic neurons instead of 1, and using Section A.4.1 for randomising early synaptic weights.

A.3.7 Pattern Generation (Figures 7.1, 7.2, 7.3 and 7.4)

The source code for generating input patterns can be found in the class file `ClassBGLayerFluctuatingNoiseToPolychronousPatterns.m`.

variable	value
group A noise source	inhomogeneous (ornstein-uhlenbeck) poisson 10 Hz
group B noise source	homogeneous poisson 10 Hz
pattern onset time	inf
pattern family	n.a.
pattern type	n.a.
pattern shape	n.a.
pattern duration	n.a.
inter-pattern spacing	n.a.
θ_{OU}	0.05
σ_{OU}	4.5

Table A.19: Non-default settings for the simulations of attractor-less STDP with correlated inputs shown in Figures 6.20 and 6.21.

variable	value
$g_+(w)$	Equation 5.5
$g_-(w)$	Equation 5.5
projection multiplier	$1000/N_{inputs}$

Table A.20: Non-default settings for the simulations of attractor-less STDP with correlated inputs shown in Figures 6.22 and 6.23.

variable	value
λ	1/32
A_+	1
A_-	-1
τ_+	20 ms
τ_-	20 ms
$g_+(w)$	Equation 5.3
$g_-(w)$	Equation 5.3
projection multiplier	$1000/N_{inputs}$
$\tau_{hebbianDecay}$	160 ms

Table A.21: Non-default settings for the simulations of attractor-less STDP with correlated inputs shown in Figure 6.25.

variable	value
$g_+(w)$	Equation 5.3
$g_-(w)$	Equation 5.3
projection multiplier	$1000/N_{inputs}$

Table A.22: Non-default settings for the simulations of attractor-less STDP with correlated inputs shown in Figures A.1 and A.2.

variable	value
pattern type	C
pattern shape	ordered (two-sided)
number of distinct patterns	5
STDP rule	Table A.6c
$g_+(w)$	Equation 5.3
$g_-(w)$	Equation 5.3

Table A.23: Settings of Figure 6.27 that are different from those used in Table A.15.

A.3.8 Polychronous Distance Metric (Figure 7.7)

The pseudocode for the distance metric that respects the higher importance of early parts of polychronous patterns is shown in Algorithm 1 (p. 190). For the actual implementation, see the MATLAB function `chapter203RobustParams_PatternSimilarity.m` in the git repository.

A.3.9 Multiple Neurons (Figures 7.8, 7.10 and 7.11)

The overview figure in which each postsynaptic neuron received completely independent inputs (Figure 7.8) was formed from simulation results in which each of 50 postsynaptic neurons received 2000 inputs. It was implemented here as an input layer of 100000 input units with a connection matrix such that each postsynaptic neuron received non-overlapping inputs from exactly 2000 units. Due to the slow simulation speed, only 50 neurons were used here. Alternatively, a simulation with 2000 input units and one output neuron could have been repeated 50 times to produce the same figure. See Table A.26 for a summary of parameters. Figure 7.10 also used zero onset lag as in Figure 7.8. But different from Figure 7.8, Figure 7.10 postsynaptic neurons did not receive independent inputs. Instead, all postsynaptic neurons receive the same inputs, including random fluctuations of input population background rate. See Table A.27 for parameters. Figure 7.11 uses nearly equal settings to Figure 7.10, except that postsynaptic neurons began to be plastic with 1 second offset each. See Table A.28.

A.3.10 Reinforcing Specific Pattern Ranges (Figures 9.3 and 9.4)

Each row of Figure *9.3* summarises multiple simulations of *above*-baseline dopamine. Apart from the changed dopamine settings, all other simulation settings are identical. See Table A.29 for parameters that differ from the defaults of Table A.15.

Each row of Figure *9.4* summarises multiple simulations of *below*-baseline dopamine. Apart from the changed dopamine settings, all other simulation settings are identical. See Table A.30 for parameters that differ from the defaults of Table A.15.

A.3.11 Exploring DA timing effects (Figures 10.1, 10.2, 10.3 and 10.4)

Simulation settings for Figure 10.1 are shown in Tables A.31 and A.32. Simulation settings for Figure 10.2 are shown in Tables A.33 and A.34. Simulation settings for Figure 10.3 are shown in Tables A.35 and A.36. Simulation settings for Figure 10.4 are shown in Tables A.37 and A.38.

A.3.12 Self-regulation of DA (Figure 10.6)

In Figure 10.6, DA level is finally controlled by the number of spikes arriving at the SNc at any given time. See Table A.39 for special simulation parameters. The implementation of the dopamine-defining nonlinearity can be seen in source code file `ClassBGLayerSpikesToReinforcementFeedback.m` .

A.4 Other Settings

A.4.1 Activity-independent synaptic growth

See source code in the following functions for implementation details:
`ClassBGWeights.randomwalkWeightIncrease()`

variable	value
pattern type	A
pattern shape	ordered (one-sided)
number of distinct patterns	5
STDP rule	Table A.6c
$g_+(w)$	Equation 5.3
$g_-(w)$	Equation 5.3

Table A.24: Settings of Figure A.3 that are different from those used in Table A.15.

variable	value
pattern type	D
pattern shape	unordered
number of distinct patterns	5

Table A.25: Settings of Figure A.4 that are different from those used in Table A.15.

Algorithm 1: Computation of Distance Metric for Polychronous Patterns

Input: window size m
foreach *pid* ← *pattern* **do**
  Data: theEarlySet ← set of units that fire within first m timebins in pid
  foreach *oid* ← *other pattern* **do**
    foreach *wpos* ← *shifting window postition* **do**
      count how many units of theEarlySet are active within the shifted window in pattern oid;
    end
    Data: maxMatch ← maximum over window shift positions
  end
  Data: avgMaxMatch ← average maximum match over all other patterns
  (normalise via ratio between avgMaxMatch and number of units in theEarlySet)
end
Result: avgAvgMaxMatch ← average over all patterns
(repeat for more averaging);

variable	value
group A noise source	homogeneous poisson 15 Hz
group B noise source	homogeneous poisson 15 Hz
pattern type	B
pattern shape	ordered (two-sided)
number of distinct patterns	40
$N_{outputs}$	50
independent inputs	yes
plasticity onset lag	0 s
weight random walk rate	0.001 (A.4.1)

Table A.26: Non-default settings for all simulations shown in Figure 7.8. Compare default parameters given in Table A.15.

variable	value
group A noise source	homogeneous poisson 15 Hz
group B noise source	homogeneous poisson 15 Hz
pattern type	B
pattern shape	ordered (two-sided)
number of distinct patterns	40
$N_{outputs}$	99
independent inputs	no
plasticity onset lag	0 s
weight random walk rate	0.001 (A.4.1)

Table A.27: Non-default settings for all simulations shown in Figure 7.10. Compare default parameters given in Table A.15.

variable	value
group A noise source	homogeneous poisson 15 Hz
group B noise source	homogeneous poisson 15 Hz
pattern type	B
pattern shape	ordered (two-sided)
number of distinct patterns	40
$N_{outputs}$	99
independent inputs	no
plasticity onset lag	1 s
weight random walk rate	0.001 (A.4.1)

Table A.28: Non-default settings for all simulations shown in Figure 7.11. Compare default parameters given in Table A.15.

variable	value
group A noise source	homogeneous poisson 15 Hz
group B noise source	homogeneous poisson 15 Hz
pattern type	B
pattern shape	ordered (two-sided)
number of distinct patterns	40
$N_{outputs}$	100
independent inputs	no
plasticity onset lag	1 s
inner DA range	0.1
outer DA range	0.2

Table A.29: Non-default settings for all simulations shown in Figure 9.3. Compare default parameters given in Table A.15.

variable	value
group A noise source	homogeneous poisson 15 Hz
group B noise source	homogeneous poisson 15 Hz
pattern type	B
pattern shape	ordered (two-sided)
number of distinct patterns	40
$N_{outputs}$	100
independent inputs	no
plasticity onset lag	1 s
inner DA range	0.1
outer DA range	0.2

Table A.30: Non-default settings for all simulations shown in Figure 9.4. Compare default parameters given in Table A.15.

variable	value
group A noise source	homogeneous poisson 15 Hz
group B noise source	homogeneous poisson 15 Hz
pattern type	C
pattern shape	ordered (two-sided)
number of distinct patterns	1
$N_{outputs}$	100
independent inputs	no
plasticity onset lag	1 s
τ_{DA}	0 ms
start of DA variations	1^{st} second
end of DA variations	280^{th} second
DA timing at start of pattern	-10 ms
DA timing at end of pattern	+50 ms
non-baseline DA level	0.0

Table A.31: Non-default settings for all simulations shown in Figure 10.1a-f. Compare default parameters given in Table A.15.

variable	value
start of DA variations	90^{th} second
end of DA variations	280^{th} second
DA timing at start of pattern	-10 ms
DA timing at end of pattern	+50 ms

Table A.32: Non-default settings for all simulations shown in Figure 10.1g-l that are not already defined in Table A.31. Compare default parameters given in Table A.15.

variable	value
start of DA variations	1^{st} second
end of DA variations	280^{th} second
DA timing at start of pattern	+10 ms
DA timing at end of pattern	+50 ms

Table A.33: Non-default settings for all simulations shown in Figure 10.2a-f that are not already defined in Table A.31. Compare default parameters given in Table A.15.

variable	value
start of DA variations	90^{th} second
end of DA variations	280^{th} second
DA timing at start of pattern	+10 ms
DA timing at end of pattern	+50 ms

Table A.34: Non-default settings for all simulations shown in Figure 10.2g-l that are not already defined in Table A.31. Compare default parameters given in Table A.15.

variable	value
start of DA variations	1^{st} second
end of DA variations	280^{th} second
DA timing at start of pattern	+50 ms
DA timing at end of pattern	+50 ms

Table A.35: Non-default settings for all simulations shown in Figure 10.3a-f that are not already defined in Table A.31. Compare default parameters given in Table A.15.

variable	value
start of DA variations	90^{th} second
end of DA variations	280^{th} second
DA timing at start of pattern	+50 ms
DA timing at end of pattern	+50 ms

Table A.36: Non-default settings for all simulations shown in Figure 10.3g-l that are not already defined in Table A.31. Compare default parameters given in Table A.15.

variable	value
start of DA variations	90^{th} second
end of DA variations	280^{th} second
DA timing at start of pattern	+10 ms
DA timing at end of pattern	+50 ms
non-baseline DA level	0.3

Table A.37: Non-default settings for all simulations shown in Figure 10.4a-f that are not already defined in Table A.31. Compare default parameters given in Table A.15.

variable	value
start of DA variations	90^{th} second
end of DA variations	280^{th} second
DA timing at start of pattern	+10 ms
DA timing at end of pattern	+50 ms
non-baseline DA level	0.5

Table A.38: Non-default settings for all simulations shown in Figure 10.4g-l that are not already defined in Table A.31. Compare default parameters given in Table A.15.

```
ClassBGWeights.newbornWeightIncrease()
ClassBGWeights.updateTrainedness()
```

A.4.2 Simulation time step size

All computer simulations were performed with 1ms time step. Izhikevich neuron models were updated twice per timestep with half scale, as suggested by original example code of (Izhikevich, 2003).

variable	value
τ_{DA}	100 ms
start of DA variations	1^{st} second
end of DA variations	280^{th} second
DA timing at start of pattern	feedback-dependent
DA timing at end of pattern	feedback-dependent
non-baseline DA level	feedback-dependent

Table A.39: Non-default settings for all simulations shown in Figure 10.6a-f that are not already defined in Table A.31. Compare default parameters given in Table A.15.

Appendix B

Simulator Development

As the presented work explored many new concepts or new combinations of known concepts, a high flexibility was required of any neural simulation environment that was to be used in the process. After evaluation of some neural simulation software packages, and comparing the overhead work needed to adjust them to my needs, I decided to use a prototyping language for my own software development instead. I chose to implement my own simulation code in MATLAB, as this environment presents a tolerable balance between flexibility and predefined library functions to quickly implement new ideas at the expense of execution time.

B.1 Simulator Structure

My simulator code consists of the three main classes

- `AbstractClassBGLayer.m`
- `ClassBGWeights.m`
- `ClassBGNucleiNetwork`

which all inherit from MATLAB's `handle` class to allow their objects to be passed around within MATLAB without copying. `AbstractClassBGLayer.m` is then extended by any of the following classes to produce layer objects that contain state variables and functions to actually run the simulation:

- `ClassBGLayerDirect.m`
- `ClassBGLayerIzhikevich.m`
- `ClassBGLayerIzhikevich1D.m`
- `ClassBGLayerRateToRegularSpikes.m`
- `ClassBGLayerRateToSimultaneousSpikes.m`
- `ClassBGLayerRateToUncertaintyValue.m`
- `ClassBGLayerRLVariablesToChainedNoisySpikePatterns.m`
- `ClassBGLayerRLVariablesToCurvedNoisySpikeChain.m`
- `ClassBGLayerRLVariablesToShuffledNoisySpikeChain.m`

- `ClassBGLayerFluctuatingNoiseToPolychronousPatterns.m`
- `ClassBGLayerSpikesToReinforcementFeedback.m`
- `ClassBGLayerSpikesToRateThreshold.m`

Any two "layers" can then be connected through the creation of a `ClassBGWeights.m` object, and grouped in a network via `ClassBGNucleiNetwork`, which also provides housekeeping functions such as recording of membrane history for later analysis. Simulations are described in separate simulation scripts for each scientific question that is to be explored, and additional classes help in recurring visualisation tasks.

All files are tracked in a Git version control repository[1], and frequent changes are also committed to a specific branch of this repository whenever a new simulation is started on a remote server (see Appendix C). This ensures that simulation results are always in sync with a committed Git revision that produces them. The standard visualisation functions use common functions in `PublicationSettingsAndHelpers_simulationRuns.m` or similar publication-specific settings to embed a tiny watermark with the originating Git revision, figure-generating source code file name, dependent functions, and any additional data sources in vector image files they create. See the lower left corner of any simulation figure in the electronic version of this document for an example.

B.2 Performance

Simulation performance was optimised using the MATLAB profiling functionality. The execution time of simulations highly depended on network size and total simulated time. Code optimisations via vectorisation reduced the initial runtime of a typical 10-minute biological time simulation with activated STDP and dopamine effects from days down to around 30 minutes execution time.

While MATLAB proved to be highly useful for fast exploration of new ideas, further large improvements in simulation speed are likely only possible by migrating the code to a more performance-oriented programming language and/or platform.

B.3 Other Simulator Software

Other simulator software that is commonly used for neural modelling includes:

- GENESIS - **Ge**neral **Ne**ural **Si**mulator **S**ystem. This software package was written for modelling multi-compartmental neurons in a Hodgkin-Huxley-type manner. While the software has been extended to support both more complex levels of detail (e.g. ion channel mechanics) and simpler point neurons, the plasticity mechanisms of the GENESIS simulator are rather rudimentary and the source code is said to be hardly maintainable.
- Neuron - The Neuron simulator is very similar to GENESIS in its intended level of detail. As it was also developed for multi-compartmental simulations of single neurons, it also appears as unsuited for highly plastic network simulations as the GENESIS simulator.

[1] The repository can be found at git@git.assembla.com:simonscnstuff.git

- NEST - The **Ne**ural **S**imulation **T**ool is aimed at large scale simulations of many point neurons, and may therefore be an interesting candidate for extension by my new paradigms. However, it does not (yet) support the features I needed for this work (e.g. generation of polychronous input patterns or neuromodulation of synaptic transmission) and was therefore not used for this work. However, now that the mechanisms shown in this work have been established, an efficient implementation of the missing features in NEST's C++ code may be appropriate.

- Brian - The brian simulator is written in scientific Python, and is advertised as promoting the flexible exploration of ideas with a simulator that does not get in the way of the user. As mentioned above, when given the choice between a prototyping language and a tool written in another prototyping language, I chose to use the language directly. However, the Brian simulator implements some very neat ideas about neural modelling and should be given increased attention in future work.

- Emergent - The simulator software used by (Frank, 2006) and others uses a strong graphical approach to neural network simulation. It may therefore have some didactical advantage for teaching. On the other hand, it does not seem to be widely used outside of the labs that develop it.

- GeNN - A fairly new entrance to the ever-changing group of neural simulators is GeNN. This **G**PU **e**nhanced **N**eural **N**etwork simulator uses Nvidia GPUs to run neural network simulations. It may be interesting to test the performance in runtime versus programming time for this new class of neural simulators.

This list is by no means complete. More overview pages can be found online at:

- http://software.incf.org/software?getTopics=Large+scale+modeling

- http://www.cnsorg.org/software

- https://grey.colorado.edu/emergent/index.php/
  Comparison_of_Neural_Network_Simulators

- http://en.wikipedia.org/wiki/Neural_network_software#Research_simulators

Appendix C

Parameter & Results Tracking

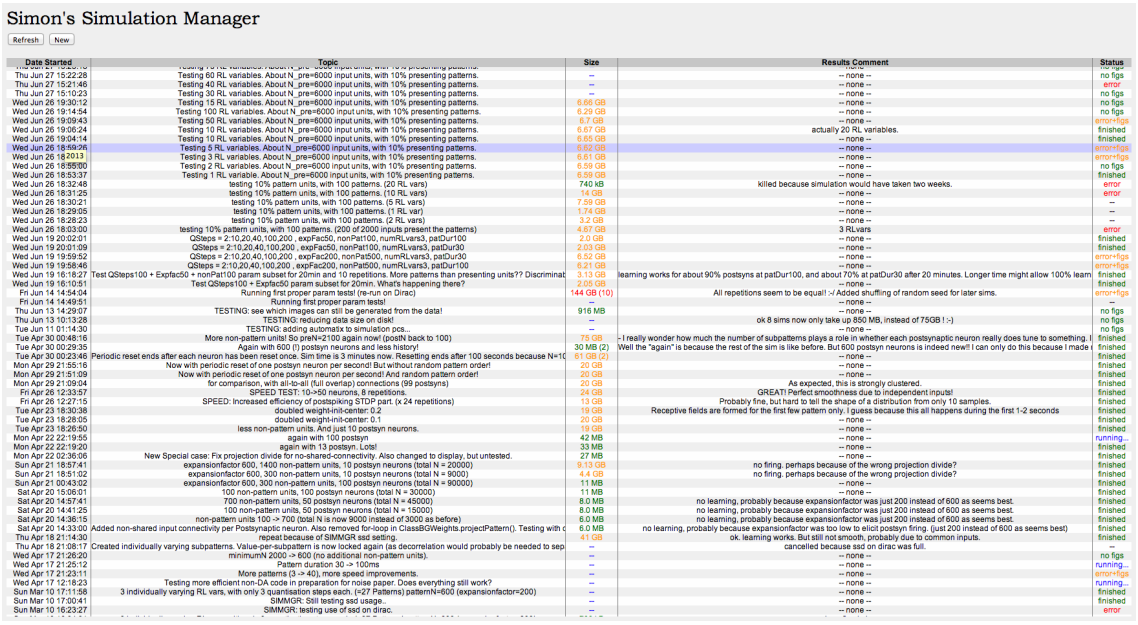


Figure C.1: Main View.

The large number of exploratory computer simulations performed during this work needed to be tracked in an organised manner. I therefore designed and implemented a "Simulation Manager" software package in the Python programming language (v2.7), with an HTML/CSS/Javascript based frontend. Features include:

- Automatic GIT version tracking of both simulation parameters and development code of the simulator software.
- Graphical user interface for entering the topic of / questions asked in new simulations with automatic highlighting of changed parameters (see Figure C.2).
- Dropdown choice of simulation server, with automatic gathering of load per server before starting new simulations (Figure C.2).
- Automatic check-in and push of GIT changes to any web service via SSH, remote login to the selected simulation server, remote GIT pull of changes to selected server, and initiation of remote execution of simulations on the remote server.

Figure C.2: "New Simulation" screen.

- Overview list of all simulation experiments, with quick view of simulation topic, result comments, simulation status, size on server disks, and time of initialisation (Figure C.1).

- Detailed view of simulation progress and simulation results, with screen output stream and error stream automatically downloaded from server, as well as automatic download and thumbnail generation of any figures generated by the simulation code. The details view also again lists the GIT differences again for later reference, and summarises collected runtime parameters of the server environment that was used for a given simulation (Figure C.3).

- Option to enter comments and thoughts on any simulation as a memory support for future reference (Figure C.3). Implemented via the GIT notes feature. This means that all relevant information on simulation topic and result comments are also available via any standard GIT browser that supports the display of GIT notes.

- Figure View. Automatically downloaded figures can either be viewed inline with the main details view of a given simulation, or opened full-screen (e.g. in a new browser tab) for in-depth analysis of visualised simulation results. The figure view (Figures C.4 and C.5) organises downloaded figures into a large table with snapshot figures of progressing stages of simulation in the columns, and parallel repetitions in the (major) rows. Within each repetition and time snapshot, multiple visualisation figures can contain multiple views on specific observed parts of a simulation.

- Regular Expressions can be used to limit the display of snapshot figures (Figure C.4). This is especially useful when viewing multiple downloaded figures in full size.

- Keyboard control. Many hotkeys have been defined via javascript to ease quick navigation within the figures view and between multiple simulations in the web browser. This vastly speeds up the exploration of simulation results by the user, and allows an animation effect when browsing through multiple snapshots at once.

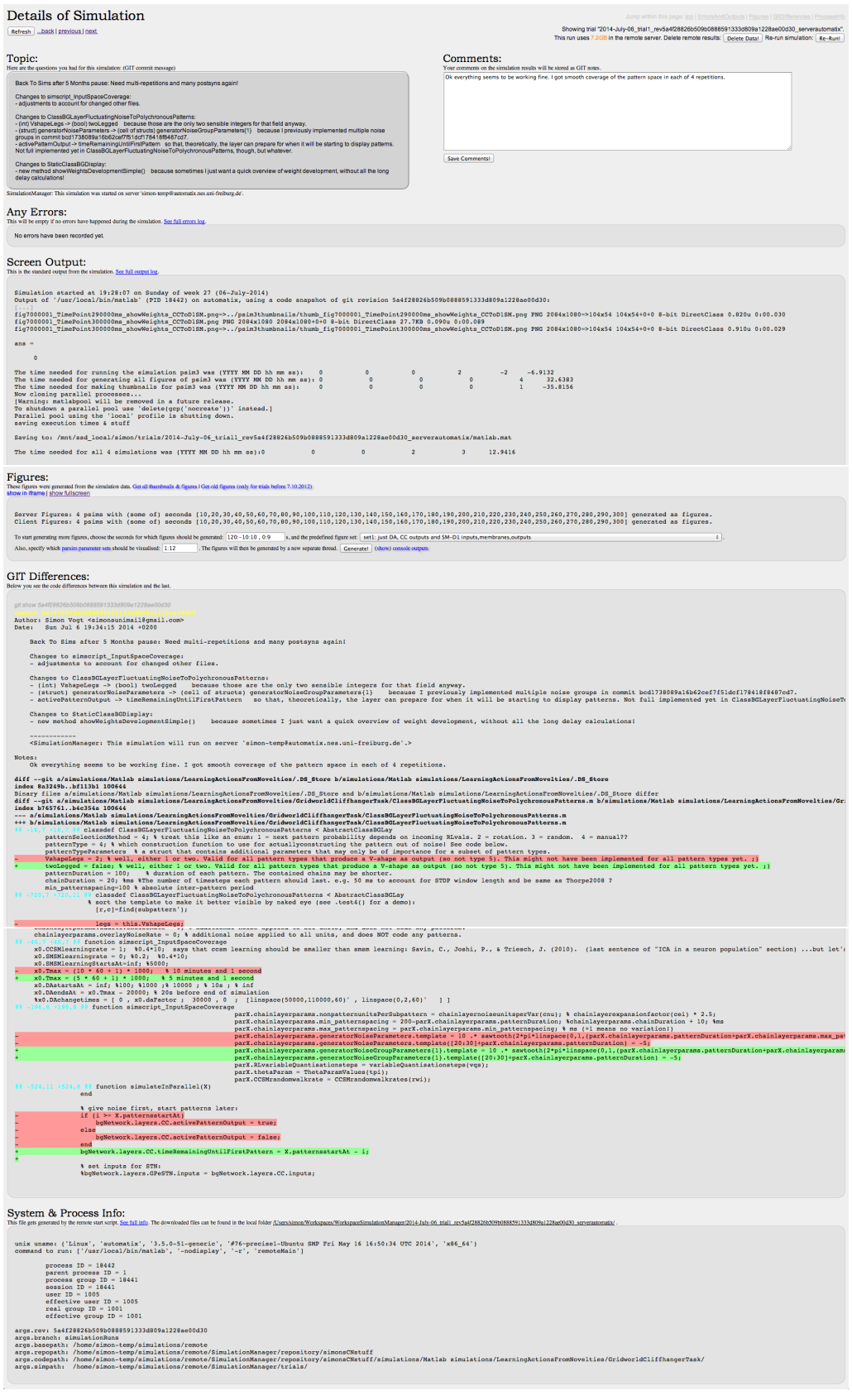


Figure C.3: "Simulation Details" screen.

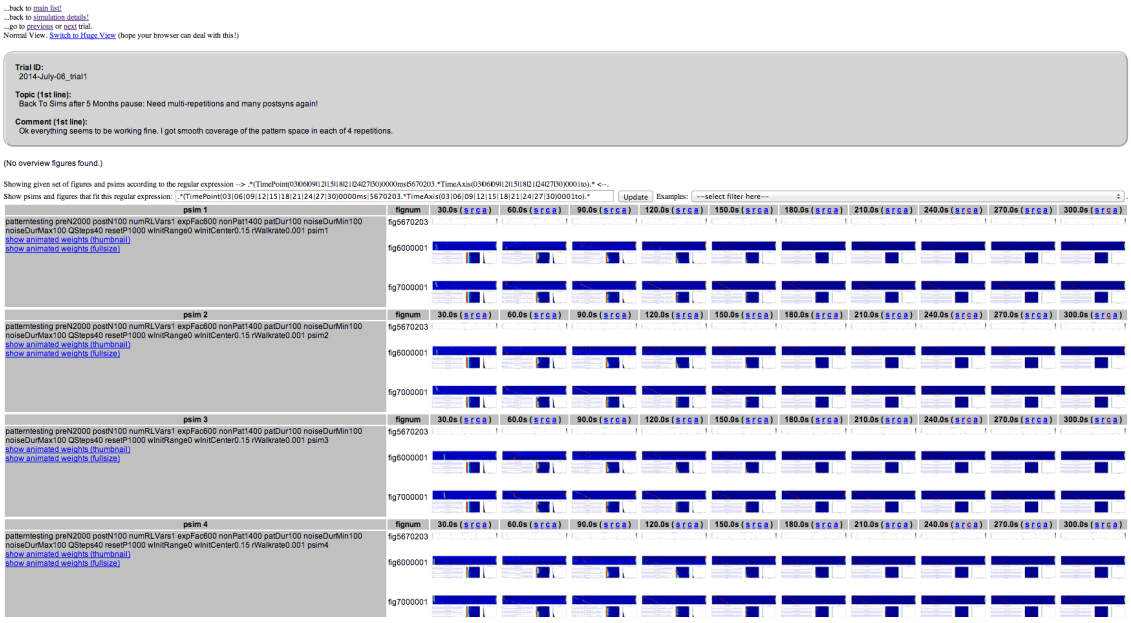


Figure C.4: “Figures” screen with regular expressions.

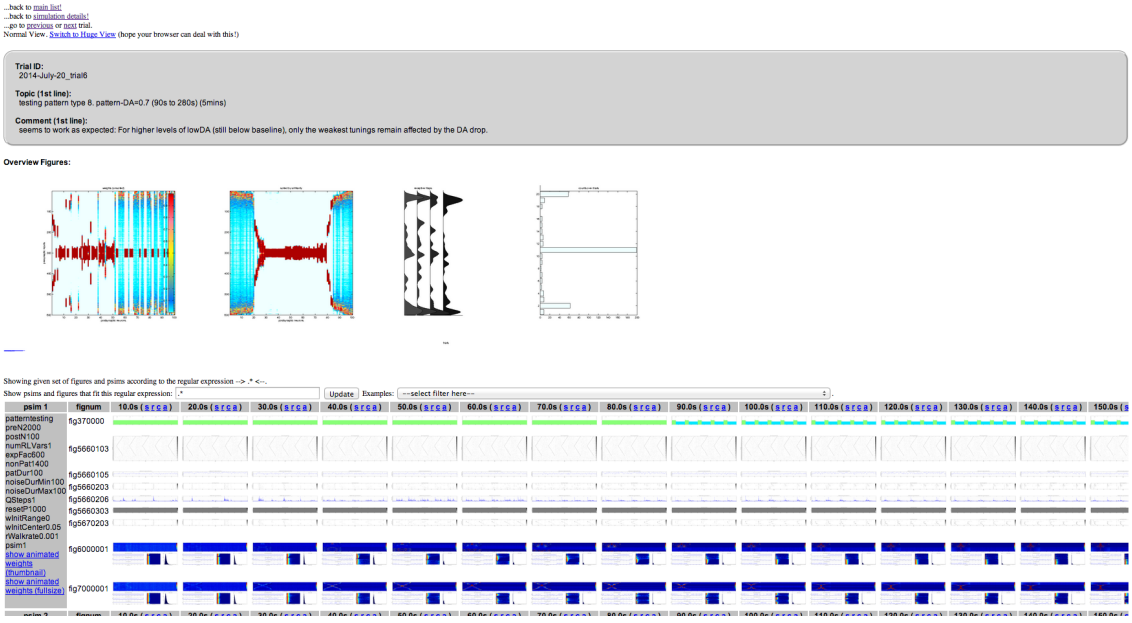


Figure C.5: “Figures” screen without regular expressions but with overview figure.

- Display of summary images. Downloaded figures that do not belong to a specific parallel run of the simulation, but rather describe an overview of all repetitions of a given simulation are displayed above the table of visualisation thumbnails in the Figure View (Figure C.5).
- Portability. Due to the minimal system requirements, the Simulation Manager is highly portable and can be accessed on many systems (see next item).
- Minimal system requirements: Only Python (e.g. v2.7), a minimal webserver (e.g. lighttpd), GIT, and some optional packages for syntax highlighting are needed to run the Simulation Manager software. Any web browser can be used for viewing, while the best results have been achieved using the cross-plattform *Google Chrome* web browser.

The source code is available as part of the software CD provided with this thesis, and anytime upon request. An open source release with accompanying publication is planned.

C.1 Versioning-based Simulation Tracking

Each different simulation is committed as its own Git revision with unique hash identifier, with a complete snapshot of the current state of the simulator code at the time of simulation. Scientific questions and result comments are notes in Git's commit messages and notes feature, respectively. This approach minimised the danger of incompatible code versions between parameter/simulation scripts and the executing simulation code.

C.2 Results Viewer

The existence of a central place to access all past simulations over multiple months and years provided an indispensable resource for scientific discovery. The interaction between the many dynamic properties involved in spatiotemporally structured spike patterns, plasticity, and instantaneous neuromodulatory effects could not have been evaluated without a clean way of planning, executing, viewing, annotating, and organising simulation results.

C.3 Comparison to other software

Recently, the development of a second simulation tracking framework has become public, to be found at `https://pythonhosted.org/Sumatra/`.

While the Sumatra package shares some features with our software and implements a direct Python interface, it does not provide a graphical interface for initiating new simulations and instead relies on a terminal command for this. However, talks with people close to Sumatra development have opened the possibility of integrating the two projects in the future.

Appendix D

DeepBrainRecorder Software

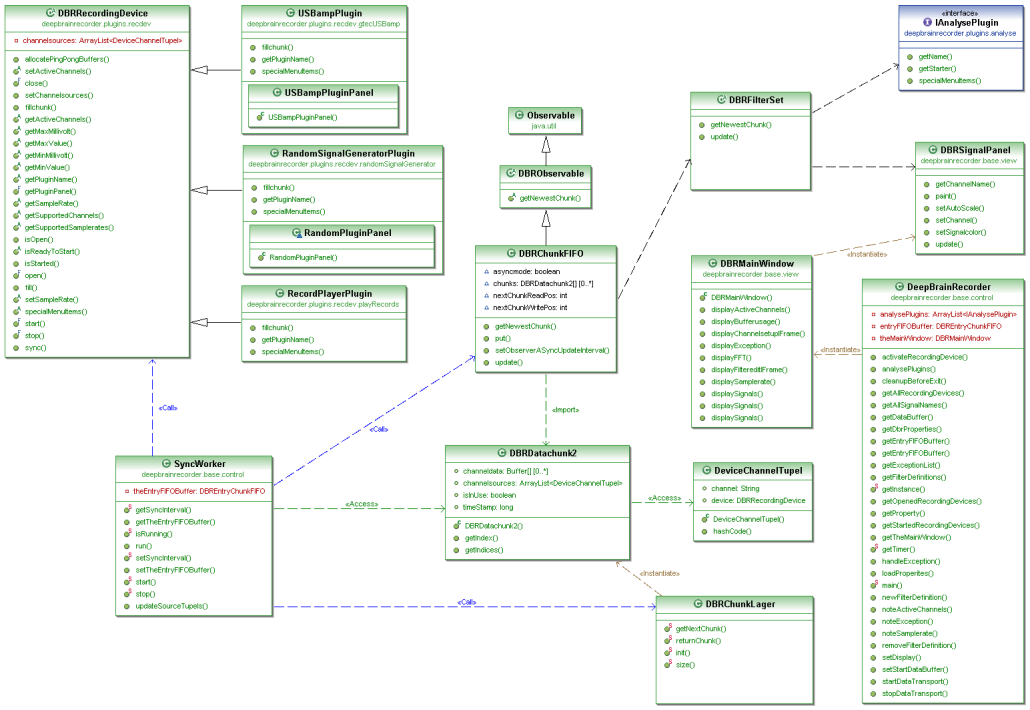


Figure D.1: UML Model of the DeepBrainRecorder.

For a cooperation with the Department of Neurology, University of Lübeck (Löffler et al., 2008; Ramrath et al., 2009), I wrote a general-use biosignal recording and online processing software for interfacing with various hardware for deep brain electrophysiological recordings. The software was later also extended to incorporate input modules for a number of self-designed biological-data acquisition systems that were being designed at the Institute for Signal Processing, University of Lübeck (Vogt et al., 2010a). The software was also demonstrated in a live setup at the INCF Conference for Neuroinformatics (Vogt and Hofmann, 2009).

D.1 Motivation

The institute for signal processing acquired four *USBAmp* biosignal amplifiers from *g.tec* (`http://www.gtec.at`) for use both in single-cell electrophysiological recordings of spikes

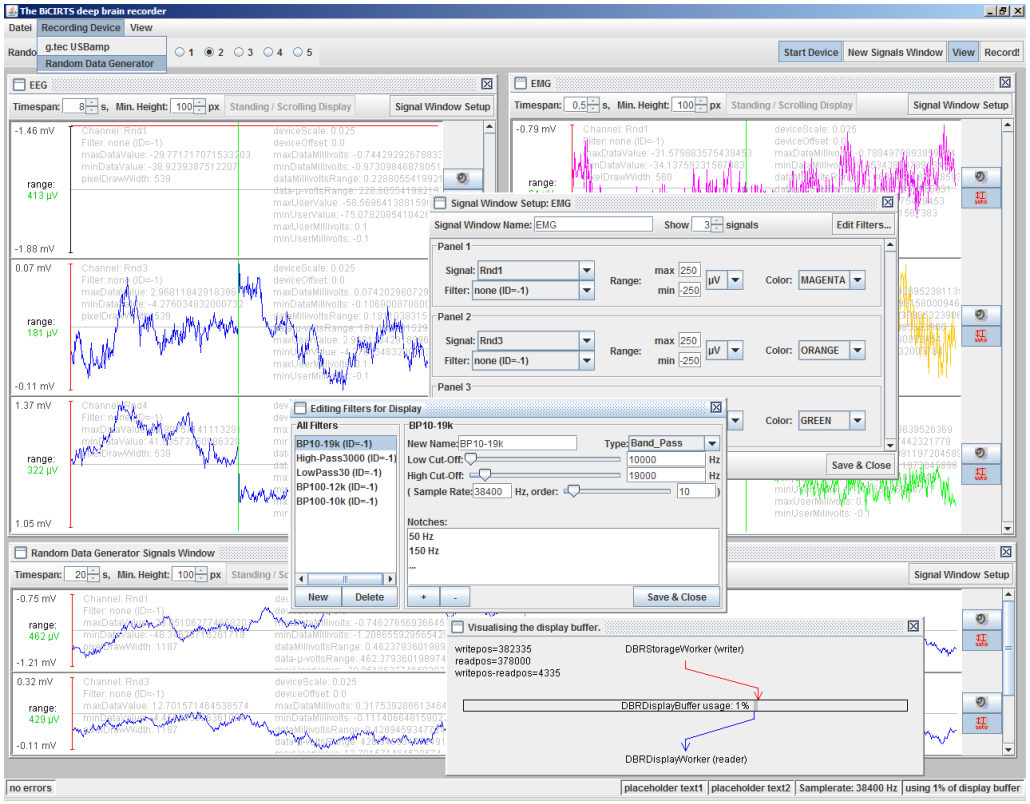


Figure D.2: Screenshot (Windows) of the DeepBrainRecorder.

as well as for EEG and EKG recordings in the context of multimodal patient monitoring (Klostermann et al., 2009; Mankodiya et al., 2010a,b,c, 2009; Vogt et al., 2010b). While the early versions of the USBAmp software were well-suited for EEG-type data acquisition, the much higher sampling rate and data throughput of electrophysiological spike recordings could not be handled robustly by the shipped software that was bundled with the device, although the hardware itself is capable of producing high data rates. Specifically, the shipped software crashed after a few minutes whenever data acquisition was performed at very high sampling rates.

A reduction of the sampling rate was not an option for use in single-cell spike recordings, and so we searched for alternative ways of using the hardware. Fortunately, the USBAmp's drivers also supported lower-level API access, and so Christ et al. (2010) wrote a C-based Win32 driver interface and enabled direct communication with the driver. A JNI interface to the Java programming language was also written, enabling a new user interface to be written in a more high-level language with less proneness to memory leaks.

Input plugins for the new software quickly extended possible data sources beyond *g.tec*'s USBAmp, and the software was henceforth used also as data acquisition and online analysis and visualisation tool for many self-built data acquisition devices around the lab, including the embedded biosignal acquisition prototype introduced in Appendix E.

D.2 Structure and Efficiency

The software was written in the Java programming language in order to use the features of modern programming languages to avoid typical C-style errors that may lead of hard to debug instabilities as was seen in the original, unreliable, software. Performance was

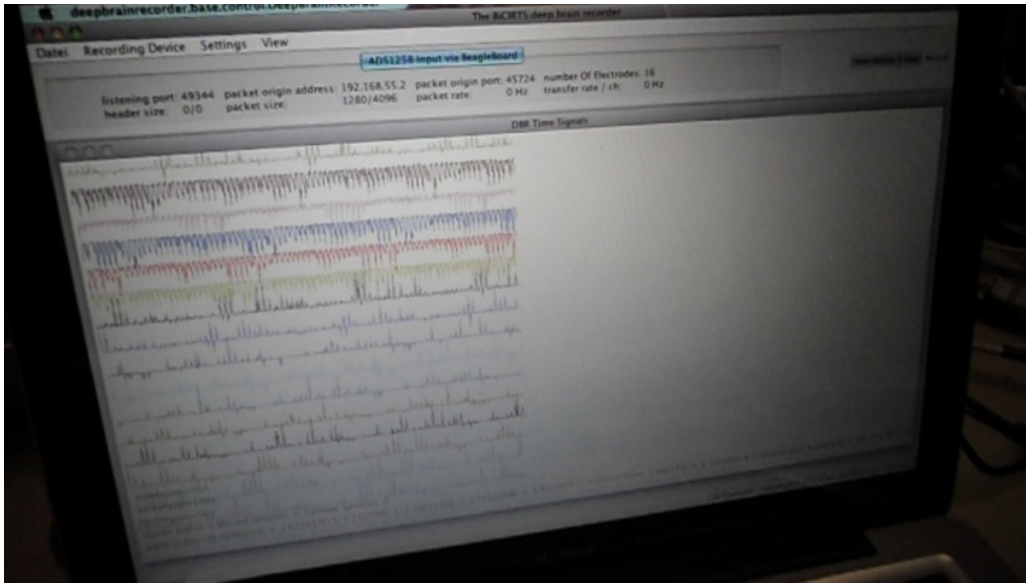


Figure D.3: The DeepBrainRecorder in practice (Mac OSX).

monitored through the use of profiling tools, and garbage collection was kept at a minimum through economical use of memory, especially by avoidance of repeated creation of java objects. At full throughput of 14.7 Mbit/s maximum USBAmp sampling rate, the implemented java software never used more than 2% of CPU load on an old 2003 Dell Laptop to receive, analyse, display and store the data stream.

The UML class diagram of the software can be seen in Figure D.1. Screenshots of the software running under windows and under Mac OSX are shown in Figures D.2 and D.3.

D.3 Recording, Analysis, and Storage Modules

The software uses a modular design to ease the creation of new recording, analysis, or storage modules. Due to this feature, the software's flexibility could be extended towards providing a general online signal analysis and interpretation tool for the use in real-time experiments.

The source code is located on the Hofmann Group's revision control server and is available upon request.

Appendix E

Embedded Systems Development

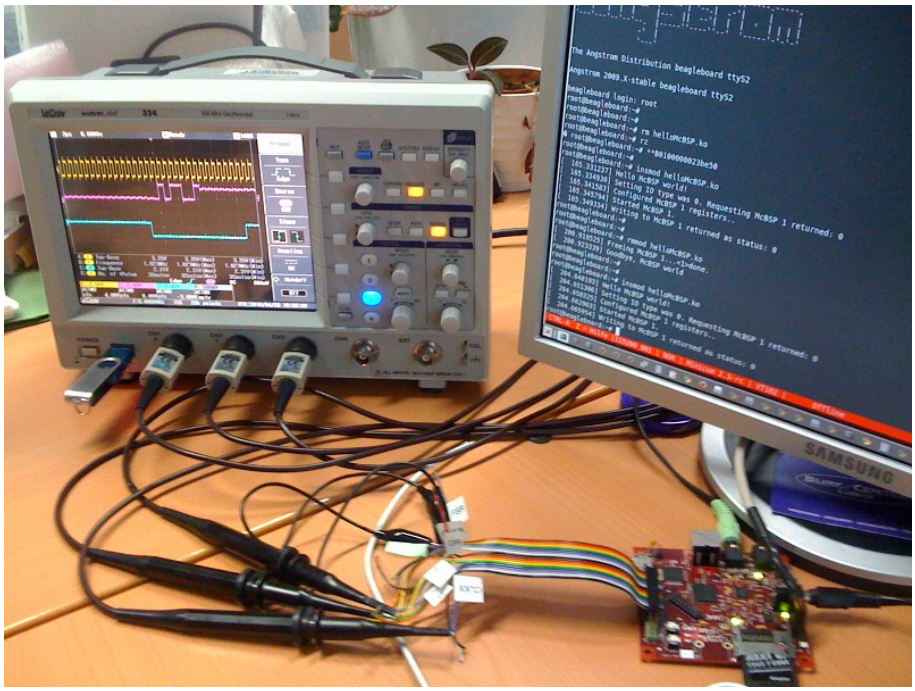


Figure E.1: Development environment while testing SPI signal timing via DMA transfer over Texas Instruments' OMAP3530 McBSP bus.

As part of a multiparty cooperative project, I implemented various embedded prototype systems that perform signal acquisition and online analysis within a low-power, low-maintenance context. The target aim of the project was to produce the technology for a future implantable intelligent deep brain stimulator device that would allow closed-loop control of brain regions affected by neurodynamical disorders such as Parkinson's Disease. While the processing hardware available for low-power execution of computationally complex online analysis algorithms has seen a surge due to the rise of fast smart phone processors in recent years, the bottleneck for building an intelligent deep brain stimulator is not the processing hardware but rather the lack of existing knowledge about neurodynamics of the brain regions in question. Still, I will present a very short overview of some embedded hardware projects I have performed during the early developmental stages of this text.

E.1 Hardware and Development Environments

Development hardware I actively used includes the following development boards:

- DSK6713 - A stand-alone digital signal processing board containing a single floating-point C6713 DSP core.
- OSK5912 - The first OMAP development board that was widely available to universities and the general public. The OMAP family of processors consists mostly of multi-core designs where a DSP is combined with one or more ARM cores for division of labour between real-time signal processing on the DSP core and preemptive multitasking and asynchronous operation on the ARM core.
- LDK5912 - A proprietary OMAP board that uses the same processor as the OSK5912 but includes a touchscreen display. We reverse-engineered undocumented ports of the LDK5912 to access the processor's known McBSP signals to control and receive data from an analog-to-digital converter that we connected to the LDK5912.
- BeagleBoard - Open-source hardware containing the 3rd-generation OMAP3530 processor by Texas Instruments. This was the main development board of the lab for some time, and can be seen in Figure E.1.
- Overo Gumstix - This proprietary hardware used the same processor as the open-source BeagleBoard, but is much smaller in size.
- BeagleBone - Based on the ideas of the BeagleBoard, this open source hardware development board is a low-cost alternative to many of the expensive boards above, but does not contain a dedicated DSP processor. However, it does contain a GPU and a dedicated ISP (image signal processing unit) as most current embedded application processors now do.

The ADS1258 is a 24 Bit, fast channel cycling Delta-Sigma analog-to-digital converter. Notable features include:

- 24 Bits, No Missing Codes
- Fixed-Channel or Automatic Channel Scan
- Fixed-Channel Data Rate: 125kSPS
- Auto-Scan Data Rate: 23.7kSPS/Channel
- Single-Conversion Settled Data

The software toolchain used during hardware development was based on the OpenEmbedded open source project and is documented in the Hofmann Group's internal wiki.

E.2 Linux Kernel Driver System

As only basic SPI ports without interrupt generating functionality were exposed to the user space of the default embedded linux distribution at the time, I wrote a linux kernel module to access the more powerful and flexible McBSP ports of the OMAP3530 processor. The module sets up two DMA transfer channels for the McBSP port and exposes

them to the linux user space as a user-accessible device. A user-space daemon or embedded DSP core can then access the data in memory for further analysis or transmission. I also implemented network code to pass on the (buffered) stream of incoming electrophysiological data to any ethernet client on the local network. This data could be received by the DeepBrainRecorder software of Appendix D for display and further online analysis or storage.

The linux kernel driver code is located on the Hofmann Group's revision control server and is available upon request.

E.3 ADC driver development

The ADS1258 data acquisition chip is a register-controlled analog-to-digital converter, which means that it needs to be set up programmatically before it can be used in a controlled manner. Communication with the ADS1258 chip was established from the DSK6713, the OSK5912, the LDK5912, and the BeagleBoard (including BeagleBoard-xM). However, only on the BeagleBoards was the communication via the ADS1258 established from within a linux environment, while the proprietary Code Composer Studio software by Texas Instruments was used for communication with the DSK6713, OSK5912, and LDK5912.

The hardware system was then tested via a hardware-in-the-loop testing environment created in our lab (Haid et al., 2010; Vogt et al., 2009).

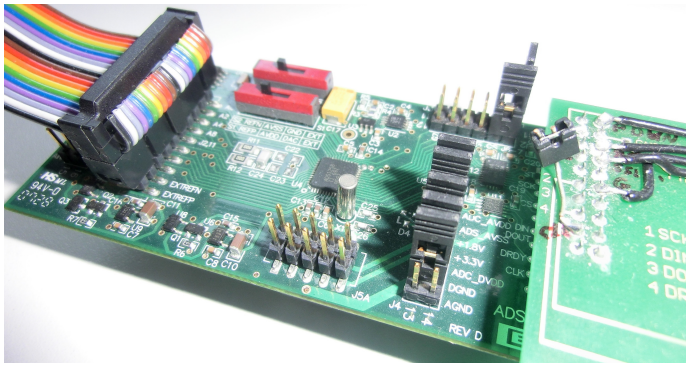


Figure E.2: ADS1258 analog-to-digital converter for which the Linux driver prototype was written. The image shows the ADS1258EVM fast prototyping board, with the ADS1258 chip located in the center.

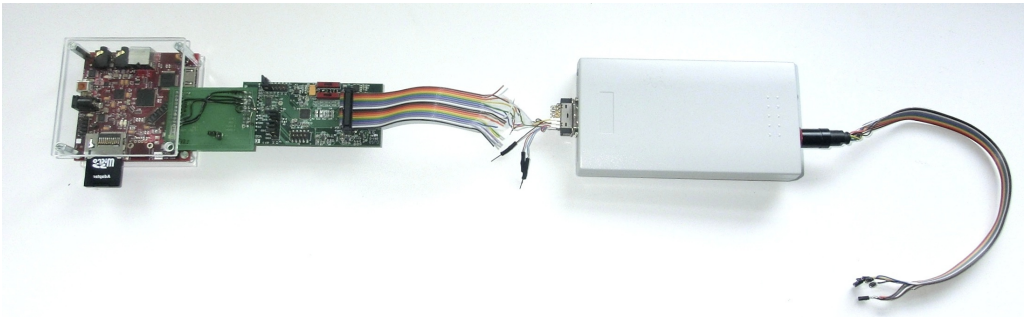


Figure E.3: BeagleBoard embedded development board connected to ADS1258 and a pre-amplifier for electrophysiological signal recording. The minimisation process still leaves some space for optimisation. ;)

Bibliography

Abbott, L. and K. Blum
1996. Functional significance of long-term potentiation for sequence learning and prediction. *Cereb. Cortex*.

Abeles, M. and G. Gerstein
1988. Detecting spatiotemporal firing patterns among simultaneously recorded single neurons. *J. Neurophysiol.*, 60(3).

Adrian, E. D. and Y. Zotterman
1926. The impulses produced by sensory nerve endings: Part 3. Impulses set up by Touch and Pressure. *J. Physiol.*, 61(4):465–83.

Aertsen, A. M. H. J. and G. L. Gerstein
1985. Evaluation of neuronal connectivity: Sensitivity of cross-correlation. *Brain Res.*, 340:341–354.

Ahissar, E. and A. Arieli
2001. Figuring space by time. *Neuron*, 32:185–201.

Aihara, K.
2008. Chaos in neurons. *Scholarpedia*, 3(5):1786.

Aizman, O., H. Brismar, P. Uhlén, E. Zettergren, a. I. Levey, H. Forssberg, P. Greengard, and a. Aperia
2000. Anatomical and physiological evidence for D1 and D2 dopamine receptor colocalization in neostriatal neurons. *Nat. Neurosci.*, 3(3):226–230.

Albin, R., A. Young, and J. Penney
1989. The functional anatomy of basal ganglia disorders. *Trends Neurosci.*, 12(10).

Alexander, G. E. and M. D. Crutcher
1990. Functional architecture of basal ganglia circuits: neural substrates of parallel processing. *Trends Neurosci.*, 13(7):266–271.

Amarasingham, A., T.-L. Chen, S. Geman, M. T. Harrison, and D. L. Sheinberg
2006. Spike count reliability and the Poisson hypothesis. *J. Neurosci.*, 26(3):801–809.

Attwell, D. and S. B. Laughlin
2001. An energy budget for signaling in the grey matter of the brain. *J. Cereb. Blood Flow Metab.*, 21(10):1133–45.

Bair, W. and C. Koch
1996. Temporal precision of spike trains in extrastriate cortex of the behaving macaque monkey. *Neural Comput.*, 8:1185–1202.

Barnes, J.
2000. *Aristotle: A Very Short Introduction.* Oxford Paperbacks.

Barto, A., S. Singh, and N. Chentanez
2004. Intrinsically motivated learning of hierarchical collections of skills. *Proc. 3rd Int. Conf. Dev. Learn. (ICDL 2004), Salk Institute, San Diego.*

Bell, A. and T. Sejnowski
1995. An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.*, 1159(1994):1129–1159.

Berger, D., C. Borgelt, and S. Louis
2010. Efficient identification of assembly neurons within massively parallel spike trains. *Comput. . . .*, 2010.

Bertler, A. and E. Rosengren
1959. Occurrence and distribution of dopamine in brain and other tissues. *Experientia*, 15:10–11.

Bevan, M. D., P. a. Booth, S. a. Eaton, and J. P. Bolam
1998. Selective innervation of neostriatal interneurons by a subclass of neuron in the globus pallidus of the rat. *J. Neurosci.*, 18(22):9438–9452.

Bi, G. and M. Poo
2001. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annu. Rev. Neurosci.*, 24:139–66.

Bi, G. Q. and M. M. Poo
1998. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.*, 18(24):10464–72.

Bienenstock, E., L. N. Cooper, and P. W. Munro
1982. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. . . . .*

Billings, G. and M. C. W. van Rossum
2009. Memory retention and spike-timing-dependent plasticity. *J. Neurophysiol.*, 101(6):2775–88.

Bishop, C.
1995. Neural networks for pattern recognition.

Blais, B. and L. Cooper
2008. BCM theory. *Scholarpedia*, 3(3):1570.

Blum, K. I. and L. F. Abbott
1996. A Model of Spatial Map Formation in the Hippocampus of the Rat. *Neural Comput.*, 8(1):85–93.

Bower, J. and D. Beeman
1995. *The book of GENESIS: exploring realistic neural models with the GEneral NEural SImulation System.*

Brette, R.
2009. Generation of correlated spike trains. *Neural Comput.*, 21:188–215.

Buzsáki, G. and a. Kandel
1998. Somadendritic backpropagation of action potentials in cortical pyramidal cells of the awake rat. *J. Neurophysiol.*, 79:1587–1591.

Calabresi, P., D. Centonze, P. Gubellini, A. Pisani, and G. Bernardi
2000. Acetylcholine-mediated modulation of striatal function. *Trends Neurosci.*, 23(3):120–126.

Carlsson, A.
1959. The occurrence, distribution and physiological role of catecholamines in the nervous system. *Pharmacol. Rev.*, 11:490–493.

Centonze, D., A. Usiello, and P. Gubellini
2002. Dopamine D2 receptor-mediated inhibition of dopaminergic neurons in mice lacking D2L receptors. *Neuropsychopharmacology*, (02).

Chorley, P. and A. K. Seth
2011. Dopamine-signaled reward predictions generated by competitive excitation and inhibition in a spiking neural network model. *Front. Comput. Neurosci.*, 5(May):21.

Christ, O., S. Otto, G. Radzimski, and U. G. Hofmann
2010. The Java Media Framework Extension Layer and its applications to biosignal acquisition. P. 4.

Cisek, P.
2007. Cortical mechanisms of action selection: the affordance competition hypothesis. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, 362(1485):1585–99.

Clopath, C., L. Büsing, E. Vasilaki, and W. Gerstner
2010. Connectivity reflects coding: a model of voltage-based STDP with homeostasis. *Nat. Neurosci.*, 13(3):344–52.

Crair, M. C. and R. C. Malenka
1995. A critical period for long-term potentiation at thalamocortical synapses.

Dale, H.
1935. Pharmacology and Nerve-endings (Walter Ernest Dixon Memorial Lecture): (Section of Therapeutics and Pharmacology). *Proc. R. Soc. Med.*, 28:319–332.

Daw, N. D., S. Kakade, and P. Dayan
2002. Opponent interactions between serotonin and dopamine. *Neural Networks*, 15:603–616.

Dayan, P. and L. F. Abbott
2001. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, volume 39 of *Computational neuroscience*. The MIT Press.

de Boer, R. and P. Kuyper
1968. Triggered correlation. *IEEE Trans. Biomed. Eng.*, 15(3):169–179.

DeCharms, R. C. and M. M. Merzenich
1996. Primary cortical representation of sounds by the coordination of action-potential timing.

Delgado, M. R., J. Li, D. Schiller, and E. a. Phelps
2008. The role of the striatum in aversive learning and aversive prediction errors. *Philos. Trans. R. Soc. Lond. B. Biol. Sci.*, 363(1511):3787–800.

Dorrscheidt, G. H.
1981. The statistical significance of the peristimulus histogram (PSTH). *Brain Res.*, 220:397–401.

Drui, G., S. Carnicella, C. Carcenac, M. Favier, a. Bertrand, S. Boulet, and M. Savasta
2014. Loss of dopaminergic nigrostriatal neurons accounts for the motivational and affective deficits in Parkinson's disease. *Mol Psychiatry*, 19(3):358–367.

Fabbrini, G., J. Juncos, M. M. Mouradian, C. Serrati, and T. N. Chase
1987. Levodopa pharmacokinetic mechanisms and motor fluctuations in Parkinson's disease. *Ann. Neurol.*, 21:370–376.

Farries, M. a. and A. L. Fairhall
2007. Reinforcement learning with modulated spike timing dependent synaptic plasticity. *J. Neurophysiol.*, 98(6):3648–65.

Fauchey, V., M. Jaber, M. G. Caron, B. Bloch, and C. Le Moine
2000. Differential regulation of the dopamine D1, D2 and D3 receptor gene expression and changes in the phenotype of the striatal neurons in mice lacking the dopamine transporter. *Eur. J. Neurosci.*, 12:19–26.

Florian, R.
2005. A reinforcement learning algorithm for spiking neural networks. *Seventh Int. Symp. Symb. Numer. Algorithms Sci. Comput.*, (Synasc):8 pp.

Florian, R. V.
2007. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural Comput.*, 19(6):1468–502.

Flourens, P.
1842. *Recherches expérimentales sur les propriétés et les fonctions du système nerveux dans les animaux vertébrés.* Balliere.

Foffani, G. and K. A. Moxon
2004. PSTH-based classification of sensory stimuli using ensembles of single neurons. *J. Neurosci. Methods*, 135:107–120.

Foldiak, P. and D. Endres
2008. Sparse coding. *Scholarpedia*, 3(1):2984.

Fonseca, R., U. V. Na, R. G. M. Morris, T. Bonhoeffer, and G. Square
2004. Competing for Memory : Hippocampal LTP under Regimes of Reduced Protein Synthesis. 44:1011–1020.

Frank, M. J.
2006. Hold your horses: a dynamic computational role for the subthalamic nucleus in decision making. *Neural Netw.*, 19(8):1120–36.

Frank, M. J., J. Samanta, A. a. Moustafa, and S. J. Sherman
2007. Hold your horses: impulsivity, deep brain stimulation, and medication in parkinsonism. *Science*, 318(5854):1309–12.

Frank, M. J., L. C. Seeberger, and R. C. O'reilly
2004. By carrot or by stick: cognitive reinforcement learning in parkinsonism. *Science*, 306(5703):1940–3.

Frémaux, N., H. Sprekeler, and W. Gerstner
2010. Functional requirements for reward-modulated spike-timing-dependent plasticity. *J. Neurosci.*, 30(40):13326–37.

Fries, P., S. Neuenschwander, a. K. Engel, R. Goebel, and W. Singer
2001. Rapid feature selective neuronal synchronization through correlated latency shifting. *Nat. Neurosci.*, 4(2):194–200.

Froemke, R. and Y. Dan
2002. Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature*, 416(6879):433–438.

Froemke, R., M.-m. Poo, and Y. Dan
2005. Spike-timing-dependent synaptic plasticity depends on dendritic location. *Nature*, 2033(May 2004):2032–2033.

Froemke, R. C., D. Debanne, and G.-Q. Bi
2010. Temporal modulation of spike-timing-dependent plasticity. *Front. Synaptic Neurosci.*, 2(June):19.

Froemke, R. C., I. a. Tsay, M. Raad, J. D. Long, and Y. Dan
2006. Contribution of individual spikes in burst-induced long-term synaptic modification. *J. Neurophysiol.*, 95(3):1620–9.

Gawne, T. J., T. W. Kjaer, and B. J. Richmond
1996. Latency: another potential code for feature binding in striate cortex. *J. Neurophysiol.*, 76:1356–1360.

Gerstein, G., E. Williams, and M. Diesmann
2012. Detecting synfire chains in parallel spike data. *J. Neurosci. ....*

Gerstein, G. L. and N. Y.-S. Kiang
1960. An Approach to the Quantitative Analysis of Electrophysiological Data from Single Neurons. *Biophys. J.*, 1(1):15–28.

Gerstner, W., R. Kempter, J. van Hemmen, and H. Wagner
1996. A neuronal learning rule for sub-millisecond temporal coding. *Nature*.

Gerstner, W., a. K. Kreiter, H. Markram, and a. V. Herz
1997. Neural codes: firing rates and beyond. *Proc. Natl. Acad. Sci. U. S. A.*, 94(November):12740–12741.

Gilson, M., A. N. Burkitt, D. B. Grayden, D. a. Thomas, and J. L. van Hemmen
2010. Emergence of network structure due to spike-timing-dependent plasticity in recurrent neuronal networks V: self-organization schemes and weight dependence. *Biol. Cybern.*, 103(5):365–86.

Gilson, M., T. Masquelier, and E. Hugues
2011. STDP allows fast rate-modulated coding with Poisson-like spike trains. *PLoS Comput. Biol.*, 7(10):e1002231.

Godwin-Austen, R., C. Frears, E. Tomlinson, and H. Kok
1969. Saturday 26 July 1969. *Lancet*, 294(7613):165–168.

Green, D. and J. Swets
1966. Signal detection theory and psychophysics.

Gurney, K., T. Prescott, and P. Redgrave
2001. A computational model of action selection in the basal ganglia. I. A new functional anatomy. *Biol. Cybern.*, 84(6):401–410.

Gütig, R., R. Aharonov, S. Rotter, and H. Sompolinsky
2003. Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity. *J. Neurosci.*, 23(9):3697–714.

Gütig, R. and H. Sompolinsky
2006. The tempotron: a neuron that learns spike timing-based decisions. *Nat. Neurosci.*, 9(3):420–8.

Guyonneau, R., R. VanRullen, and S. J. Thorpe
2005. Neurons tune to the earliest spikes through STDP. *Neural Comput.*, 17(4):859–79.

Haid, B., M. Klostermann, M. Epp, S. M. Vogt, and U. G. Hofmann
2010. Real-World Testing of Brain Stimulator Prototype. In *Biomed. Tech. Eng.*

Hasbi, A., T. Fan, M. Alijaniaram, T. Nguyen, M. L. Perreault, B. F. O'Dowd, and S. R. George
2009. Calcium signaling cascade links dopamine D1-D2 receptor heteromer to striatal BDNF production and neuronal growth. *Proc. Natl. Acad. Sci. U. S. A.*, 106(50):21377–21382.

Hasselmo, M. E. and J. McGaughy
2004. High acetylcholine levels set circuit dynamics for attention and encoding and low acetylcholine levels set dynamics for consolidation. *Prog. Brain Res.*, 145:207–31.

Hebb, D. O.
1949. *The organization of behavior; a neuropsychological theory.*

Heeger, D.
1997. Signal detection theory. *Dept. Psych., Stanford Univ., Stanford, CA, Teach. Handout.*

Heil, P., B. H. Scott, B. J. Malone, and M. N. Semple
1997. Auditory Cortical Onset Responses Revisited . I . First-Spike Timing Auditory Cortical Onset Responses Revisited . I . First-Spike Timing. *J. Neurophysiol.*, Pp. 2616–2641.

Hensch, T. K.
2005. Critical period plasticity in local cortical circuits. *Nat. Rev. Neurosci.*, 6(November):877–888.

Hernández-López, S., J. Bargas, D. J. Surmeier, a. Reyes, and E. Galarraga
1997. D1 receptor activation enhances evoked discharge in neostriatal medium spiny neurons by modulating an L-type Ca2+ conductance. *J. Neurosci.*, 17(9):3334–42.

Herrmann, A. and W. Gerstner
2001. Noise and the PSTH response to current transients: I. General theory and application to the integrate-and-fire neuron. *J. Comput. Neurosci.*, 11:135–151.

Hille, B.
1991. *Ion Channels of Excitable Membranes*, 2001 edition. Sinauer Associates,Inc.MA.

Hinton, G. E. and T. J. Sejnowski
1999. *Unsupervised Learning: Foundations of Neural Computation.* MIT Press.

Hodgkin, A. and A. Huxley
1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, Pp. 500–544.

Holgado, a. J. N., J. R. Terry, and R. Bogacz
2010. Conditions for the Generation of Beta Oscillations in the Subthalamic Nucleus-Globus Pallidus Network. *J. Neurosci.*, 30(37):12340–12352.

Hopfield, J. J.
1995. Pattern recognition computation using action potential timing for stimulus representation.

Hornykiewicz, O.
1974. On the mode of action of L-dopa in Parkinson's disease. *Life Sci.*, 15(7):1249–1259.

Howe, M. W., P. L. Tierney, S. G. Sandberg, P. E. M. Phillips, and A. M. Graybiel
2013. Prolonged dopamine signalling in striatum signals proximity and value of distant rewards. *Nature*, 500(7464):575–9.

Hubel, D. H. and T. N. Wiesel
1959. Receptive fields of single neurones in the cat's striate cortex. *J. Physiol.*, 148:574–91.

Izhikevich, E. M.
2003. Simple model of spiking neurons. *IEEE Trans. Neural Netw.*, 14(6):1569–72.

Izhikevich, E. M.
2004. Which model to use for cortical spiking neurons? *IEEE Trans. Neural Netw.*, 15(5):1063–70.

Izhikevich, E. M.
2006. Polychronization: computation with spikes. *Neural Comput.*, 18(2):245–82.

Izhikevich, E. M.
2007a. *Dynamical Systems in Neuroscience.*

Izhikevich, E. M.
2007b. Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cereb. Cortex*, 17(10):2443–52.

Izhikevich, E. M. and G. M. Edelman
2008. Large-scale model of mammalian thalamocortical systems. *Proc. Natl. Acad. Sci. U. S. A.*, 105:3593–3598.

Izhikevich, E. M., J. a. Gally, and G. M. Edelman
2004. Spike-timing dynamics of neuronal groups. *Cereb. Cortex*, 14(8):933–44.

Janezic, S., S. Threlfell, P. D. Dodson, M. J. Dowie, T. N. Taylor, D. Potgieter, L. Parkkinen, S. L. Senior, S. Anwar, B. Ryan, T. Deltheil, P. Kosillo, M. Cioroch, K. Wagner, O. Ansorge, D. M. Bannerman, J. P. Bolam, P. J. Magill, S. J. Cragg, and R. Wade-Martins
2013. Deficits in dopaminergic transmission precede neuron loss and dysfunction in a new Parkinson model. *Proc. Natl. Acad. Sci. U. S. A.*, 110:E4016–25.

Jones, D. K. and A. Leemans
2011. Diffusion tensor imaging. *Methods Mol. Biol.*, 711:127–44.

Kandel, E. R., J. H. Schwartz, and T. M. Jessell
1991. Principles of neural science. *Princ. neural Sci.*

Kerr, J. N. D., D. Greenberg, and F. Helmchen
2005. Imaging input and output of neocortical networks in vivo. *Proc. Natl. Acad. Sci. U. S. A.*, 102(39):14063–14068.

Kistler, W. M. and J. L. van Hemmen
2000. Modeling synaptic plasticity in conjuction with the timing of pre- and postsynaptic action potentials. *Neural Comput.*, 12(2):385–405.

Klostermann, M., O. Christ, K. Mankodiya, S. Vogt, and U. G. Hofmann
2009. OMAP3 based signal processing for biomedical engineering teaching. *eurasip.org*.

König, P., A. K. Engel, P. R. Roelfsema, and W. Singer
1996. Coincidence detection or temporal integration. The role of the cortical neuron revisited. *Trends Neurosci.*, 19(4):130–137.

Kroener, S., L. J. Chandler, P. E. M. Phillips, and J. K. Seamans
2009. Dopamine modulates persistent synaptic activity and enhances the signal-to-noise ratio in the prefrontal cortex. *PLoS One*, 4(8):e6507.

Lee, K.-W., Y. Kim, A. M. Kim, K. Helmin, A. C. Nairn, and P. Greengard
2006. Cocaine-induced dendritic spine formation in D1 and D2 dopamine receptor-containing medium spiny neurons in nucleus accumbens. *Proc. Natl. Acad. Sci. U. S. A.*, 103(9):3399–3404.

Lee, S. P., C. H. So, A. J. Rashid, G. Varghese, R. Cheng, a. J. Lança, B. F. O'Dowd, and S. R. George
2004a. Dopamine D1 and D2 receptor Co-activation generates a novel phospholipase C-mediated calcium signal. *J. Biol. Chem.*, 279(34):35671–8.

Lee, S. P., C. H. So, A. J. Rashid, G. Varghese, R. Cheng, a. J. Lança, B. F. O'Dowd, and S. R. George
2004b. Dopamine D1 and D2 receptor co-activation generates a novel phospholipase C-mediated calcium signal. *J. Biol. Chem.*, 279(34):35671–35678.

Lewicki, M. S.
2002. Efficient coding of natural sounds. *Nat. Neurosci.*, 5(4):356–63.

Lindsay, R. M., S. J. Wiegand, C. Anthony Altar, and P. S. DiStefano
1994. Neurotrophic factors: from molecule to man. *Trends Neurosci.*, 17(5):182–190.

Lisman, J. and N. Spruston
2005. Postsynaptic depolarization requirements for LTP and LTD: a critique of spike timing-dependent plasticity. *Nat. Neurosci.*, 8(7):839–41.

Lisman, J. and N. Spruston
2010. Questions about STDP as a General Model of Synaptic Plasticity. *Front. Synaptic Neurosci.*, 2(October):140.

Ljungberg, T., P. Apicella, and W. Schultz
1992. Responses of monkey dopamine neurons during learning of behavioral reactions. *J Neurophysiol*, 67(1):145–163.

Lloyd, K., L. Davidson, O. Hornykiewicz, and O. F. Pharmacology
1975. The neurochemistry of Parkinson's disease: effect of L-dopa therapy. *J. Pharmacol. Exp. Ther.*, 195(3):453–464.

Löffler, S., S. Vogt, U. G. Hofmann, and A. Moser
2008. Striatal microstimulation in awake animals depends on NMDA receptor activity. *Signal Processing*, 1:2–5.

Lulham, A., R. Bogacz, S. M. Vogt, and M. W. Brown
2011. An Infomax algorithm can perform both familiarity discrimination and feature extraction in a single network. *Neural Comput.*, 23(4):909–26.

MacKay, D. J. C.
2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press.

Macke, J. H., P. Berens, A. S. Ecker, A. S. Tolias, and M. Bethge
2009. Generating spike trains with specified correlation coefficients. *Neural Comput.*, 21:397–423.

Macmillan, N. A.
2002. Signal detection theory. *Stevens' Handb. Exp. Psychol.*

Mainen, Z. F. and T. J. Sejnowski
1995. Reliability of spike timing in neocortical neurons. *Science*, 268(June):1503–1506.

Mallet, N., B. R. Micklem, P. Henny, M. T. Brown, C. Williams, J. P. Bolam, K. C. Nakamura, and P. J. Magill
2012. Dichotomous organization of the external globus pallidus. *Neuron*, 74(6):1075–86.

Mankodiya, K., Y. Hassan, S. Vogt, and H. Gehring
2010a. Wearable ECG Module for Long-term Recordings using a Smartphone Processor. In *Work. Pap. UbiHealth. Proc. 12th ACM Int. Conf. Ubiquitous Comput.*, Pp. 2–5.

Mankodiya, K., M. Klostermann, S. Vogt, O. Christ, and U. G. Hofmann
2010b. Teaching signal processing with a dual-core embedded platform. In *4th Eur. Educ. Res. Conf. (EDERC 2010)*, number c, Pp. 15–18.

Mankodiya, K., S. Vogt, M. Klostermann, and U. G. Hofmann
2010c. Near Patient Sensor Integration with Smart Phone Processors. 55:10–13.

Mankodiya, K., S. Vogt, A. Kundu, M. Klostermann, J. Pohl, A. Ayoub, and H. Gehring
2009. Portable electrophysiologic monitoring based on the OMAP-family processor from a beginners' prospective. In *Digit. Signal Process. 2009 16th Int. Conf.*, number Mcm.

Markram, H., W. Gerstner, and P. J. Sjöström
2012. Spike-timing-dependent plasticity: a comprehensive overview. *Front. Synaptic Neurosci.*, 4(July):2.

Markram, H., J. Lübke, M. Frotscher, and B. Sakmann
1997. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297):213–5.

Masquelier, T., R. Guyonneau, and S. J. Thorpe
2008. Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. *PLoS One*, 3(1):e1377.

Masquelier, T., R. Guyonneau, and S. J. Thorpe
2009. Competitive STDP-based spike pattern learning. *Neural Comput.*, 21(5):1259–76.

Mastro, K. J., R. S. Bouchard, H. a. K. Holt, and A. H. Gittis
2014. Transgenic mouse lines subdivide external segment of the globus pallidus (GPe) neurons and reveal distinct GPe output pathways. *J. Neurosci.*, 34(6):2087–99.

Mehta, M. R., a. K. Lee, and M. a. Wilson
2002. Role of experience and oscillations in transforming a rate code into a temporal code. *Nature*, 417:741–746.

Merims, D. and N. Giladi
2008. Dopamine dysregulation syndrome, addiction and behavioral changes in Parkinson's disease. *Park. Relat. Disord.*, 14:273–280.

Morris, R. G., P. Garrud, J. N. Rawlins, and J. O'Keefe
1982. Place navigation impaired in rats with hippocampal lesions.

Morrison, A., A. Aertsen, and M. Diesmann
2007. Spike-timing-dependent plasticity in balanced random networks. *Neural Comput.*, 19(6):1437–67.

Morrison, A., M. Diesmann, and W. Gerstner
2008. Phenomenological models of synaptic plasticity based on spike timing. *Biol. Cybern.*, 98(6):459–78.

Nambu, A., H. Tokuno, and M. Takada
2002. Functional significance of the cortico–subthalamo–pallidal 'hyperdirect' pathway. *Neurosci. Res.*, 43(2):111–117.

Nicola, S., D. J. Surmeier, and R. C. Malenka
2000. Dopaminergic modulation of neuronal excitability in the striatum and nucleus accumbens. *Annu. Rev. Neurosci.*, Pp. 185–215.

Nicola, S. M., F. W. Hopf, and G. O. Hjelmstad
2004. Contrast enhancement: a physiological effect of striatal dopamine? *Cell Tissue Res.*, 318(1):93–106.

Nicola, S. M. and R. C. Malenka
1997. Dopamine depresses excitatory and inhibitory synaptic transmission by distinct mechanisms in the nucleus accumbens. *J. Neurosci.*, 17(15):5697–710.

O'Brien, R. J., S. Kamboj, M. D. Ehlers, K. R. Rosen, G. D. Fischbach, and R. L. Huganir
1998. Activity-dependent modulation of synaptic AMPA receptor accumulation. *Neuron*, 21(5):1067–78.

Oeser, E.
2010. Geschichte der Hirnforschung: Von der Antike bis zur Gegenwart.

Oja, E.
1982. Simplified neuron model as a principal component analyzer. *J. Math. Biol.*, 15(3):267–273.

O'Keefe, J. and M. L. Recce
1993. Phase relationship between hippocampal place units and the EEG theta rhythm. *Hippocampus*, 3(3):317–330.

Olshausen, B. and D. Field
1996a. Natural image statistics and efficient coding*. *Netw. Comput. neural ...*, 7(September 1995):333–339.

Olshausen, B. and D. Field
1997. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Res.*

Olshausen, B. A. and D. J. Field
1996b. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609.

Paladini, C. a., P. Celada, and J. M. Tepper
1999. Striatal, pallidal, and pars reticulata evoked inhibition of nigrostriatal dopaminergic neurons is mediated by GABA(A), receptors in vivo. *Neuroscience*, 89(3):799–812.

Parent, A., F. Sato, Y. Wu, J. Gauthier, M. Lévesque, and M. Parent
2000. Organization of the basal ganglia: the importance of axonal collateralization. *Trends Neurosci.*, 23(10):S20–S27.

Perreault, M. L., T. Fan, M. Alijaniaram, B. F. O'Dowd, and S. R. George
2012. Dopamine D1-D2 receptor heteromer in dual phenotype GABA/glutamate-coexpressing striatal medium spiny neurons: Regulation of BDNF, GAD67 and VGLUT1/2. *PLoS One*, 7(3):1–10.

Perreault, M. L., A. Hasbi, M. Alijaniaram, T. Fan, G. Varghese, P. J. Fletcher, P. Seeman, B. F. O'Dowd, and S. R. George
2010. The dopamine D1-D2 receptor heteromer localizes in dynorphin/enkephalin neurons: Increased high affinity state following amphetamineandin schizophrenia. *J. Biol. Chem.*, 285:36625–36634.

Pfister, J.-P. and W. Gerstner
2006. Triplets of spikes in a model of spike timing-dependent plasticity. *J. Neurosci.*, 26(38):9673–82.

Pfister, J.-P., T. Toyoizumi, D. Barber, and W. Gerstner
2006. Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning. *Neural Comput.*, 18(6):1318–48.

Potjans, W., A. Morrison, and M. Diesmann
2009. A spiking neural network model of an actor-critic learning agent. *Neural Comput.*, 21(2):301–39.

Ramón y Cajal, S.
1911. Histologie du Systeme Nerveux de l'homme et des Vertebres, Vol. 2. *Maloine, Paris*, Pp. 887–890.

Ramrath, L., S. Vogt, W. Jensen, U. G. Hofmann, and A. Schweikard
2009. Computer- and robot-assisted stereotaxy for high-precision small animal brain exploration. *Biomed. Tech.*, 54:8–13.

Rashid, A. J., C. H. So, M. M. C. Kong, T. Furtak, M. El-Ghundi, R. Cheng, B. F. O'Dowd, and S. R. George
2007. D1-D2 dopamine receptor heterooligomers with unique pharmacology are coupled to rapid activation of Gq/11 in the striatum. *Proc. Natl. Acad. Sci. U. S. A.*, 104:654–659.

Redgrave, P.
2007. Basal ganglia. *Scholarpedia*, 2(6):1825.

Redgrave, P. and K. Gurney
2006. The short-latency dopamine signal: a role in discovering novel actions? *Nat. Rev. Neurosci.*, 7(12):967–75.

Redgrave, P., K. N. Gurney, and J. Reynolds
2008. What is reinforced by phasic dopamine signals? *Brain Res. Rev.*, 58(2):322–339.

Reynolds, J. and J. Wickens
2000. Substantia nigra dopamine regulates synaptic plasticity and membrane potential fluctuations in the rat neostriatum, in vivo. *Neuroscience*, 99(2):199–203.

Reynolds, J. N. J. and J. R. Wickens
2002. Dopamine-dependent plasticity of corticostriatal synapses. *Neural Netw.*, 15(4-6):507–21.

Rotaru, D. C., D. a. Lewis, and G. Gonzalez-Burgos
2007. Dopamine D1 receptor activation regulates sodium channel-dependent EPSP amplification in rat prefrontal cortex pyramidal neurons. *J. Physiol.*, 581(Pt 3):981–1000.

Rubin, J., D. Lee, and H. Sompolinsky
2001. Equilibrium Properties of Temporally Asymmetric Hebbian Plasticity. *Phys. Rev. Lett.*, 86(2):364–367.

Sakai, H. M., K. Naka, and B. Biology
1987. Signal transmission in the catfish retina. V. Sensitivity and circuit. *J Neurophysiol*, 58(6):1329–1350.

Savin, C., P. Joshi, and J. Triesch
2010. Independent component analysis in spiking neurons. *PLoS Comput. Biol.*, 6(4):e1000757.

Schrader, S. and S. Grün
2008. Detecting synfire chain activity using massively parallel spike train recording. *J. Neurophysiol.*, Pp. 2165–2176.

Schultz, W.
2000. Multiple reward signals in the brain. *Nat. Rev. Neurosci.*, 1(3):199–207.

Schultz, W.
2007. Multiple dopamine functions at different time courses. *Annu. Rev. Neurosci.*, 30:259–288.

Schultz, W., P. Dayan, and P. Montague
1997. A neural substrate of prediction and reward. *Science (80-.).*, 275(5306):1593.

Shannon, C.
1948. A mathematical theory of communication. *ACM SIGMOBILE Mob. Comput. . . .*, XXVII(3).

Shen, W., M. Flajolet, P. Greengard, and D. J. Surmeier
2008. Dichotomous dopaminergic control of striatal synaptic plasticity. *Science*, 321(5890):848–51.

Shlens, J., G. D. Field, J. L. Gauthier, M. I. Grivich, D. Petrusca, A. Sher, A. M. Litke, and E. J. Chichilnisky
2006. The structure of multi-neuron firing patterns in primate retina. *J. Neurosci.*, 26(32):8254–8266.

Shoham, S., D. H. O'Connor, and R. Segev
2006. How silent is the brain: Is there a "dark matter" problem in neuroscience? *J. Comp. Physiol. A Neuroethol. Sensory, Neural, Behav. Physiol.*, 192:777–784.

Shouval, H. Z., S. S.-H. Wang, and G. M. Wittenberg
2010. Spike timing dependent plasticity: a consequence of more fundamental learning rules. *Front. Comput. Neurosci.*, 4(July):1–13.

Singh, S., A. Barto, and N. Chentanez
2005. Intrinsically motivated reinforcement learning. *Adv. Neural Inf. Process. Syst.*, 17:1281–1288.

Sjöström, J. and W. Gerstner
2010. Spike-timing dependent plasticity. *Scholarpedia*, 5(2):1362.

Sjöström, P. J., G. G. Turrigiano, and S. B. Nelson
2001. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron*, 32(6):1149–64.

Smith, Y., M. D. Bevan, E. Shink, and J. P. Bolam
1998. Microcircuitry of the direct and indirect pathways of the basal ganglia. *Neuroscience*, 86(2):353–387.

Smith, Y. and J. Z. Kieval
2000. Anatomy of the dopamine system in the basal ganglia. *Trends Neurosci.*, 23(00):S28–S33.

So, C. H., G. Varghese, K. J. Curley, M. M. C. Kong, M. Alijaniaram, X. Ji, T. Nguyen, B. F. O'dowd, and S. R. George
2005. D1 and D2 dopamine receptors form heterooligomers and cointernalize after selective activation of either receptor. *Mol. Pharmacol.*, 68(3):568–578.

Song, S., K. D. Miller, and L. F. Abbott
2000. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nat. Neurosci.*, 3(9):919–26.

Stein, R. B., E. R. Gossen, and K. E. Jones
2005. Neuronal variability: noise or part of the signal? *Nat. Rev. Neurosci.*, 6(May):389–397.

Suri, R. and W. Schultz
2001. Temporal difference model reproduces anticipatory neural activity. *Neural Comput.*, 13(4):841–862.

Suri, R. E. and W. Schultz
1998. Learning of sequential movements by neural network model with dopamine-like reinforcement signal. *Exp. Brain Res.*, 121(3):350–4.

Suri, R. E. and W. Schultz
1999. A neural network model with dopamine-like reinforcement signal that learns a spatial delayed response task. *Neuroscience*, 91(3):871–90.

Surmeier, D. J., J. Ding, M. Day, Z. Wang, and W. Shen
2007. D1 and D2 dopamine-receptor modulation of striatal glutamatergic signaling in striatal medium spiny neurons. *Trends Neurosci.*, 30(5):228–235.

Sutton, R. and A. Barto
1998. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Networks*, 9(5):1054–1054.

Sutton, R., D. Precup, and S. Singh
1998. Intra-option learning about temporally abstract actions. *Proc. Fifteenth Int. Conf. Mach. Learn.*, Pp. 556–564.

Thompson, D., L. Martini, and J. L. Whistler
2010. Altered ratio of D1 and D2 dopamine receptors in mouse striatum is associated with behavioral sensitization to cocaine. *PLoS One*, 5(6).

Thorpe, S., a. Delorme, and R. Van Rullen
2001. Spike-based strategies for rapid processing. *Neural Netw.*, 14(6-7):715–25.

Thurley, K., W. Senn, and H.-R. Lüscher
2008. Dopamine increases the gain of the input-output response of rat prefrontal pyramidal neurons. *J. Neurophysiol.*, 99(6):2985–97.

Toyoizumi, T. and J. Pfister
2005. Generalized Bienenstock–Cooper–Munro rule for spiking neurons that maximizes information transmission. *Proc. ...*, 102(14):5239–5244.

Toyoizumi, T., J.-P. Pfister, K. Aihara, and W. Gerstner
2007. Optimality model of unsupervised spike-timing-dependent plasticity: synaptic memory and weight distribution. *Neural Comput.*, 19(3):639–71.

Trimmer, P. C., A. I. Houston, J. A. R. Marshall, R. Bogacz, E. S. Paul, M. T. Mendl, and J. M. McNamara
2008. Mammalian choices: combining fast-but-inaccurate and slow-but-accurate decision-making systems. *Proc. Biol. Sci.*, 275(1649):2353–61.

Turrigiano, G. G., K. R. Leslie, N. S. Desai, L. C. Rutherford, and S. B. Nelson
1998. Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391(6670):892–6.

Turrigiano, G. G. and S. B. Nelson
2004. Homeostatic plasticity in the developing nervous system. *Nat. Rev. Neurosci.*, 5(2):97–107.

Uhlenbeck, G. E. and L. S. Ornstein
1930. On the Theory of the Brownian Motion. *Phys. Rev.*, 36(5):823–841.

Urbanczik, R. and W. Senn
2009. Reinforcement learning in populations of spiking neurons. *Nat. Neurosci.*, 12(3):250–2.

Ushiba, J., Y. Tomita, Y. Masakado, and Y. Komune
2002. A cumulative sum test for a peri-stimulus time histogram using the Monte Carlo method. *J. Neurosci. Methods*, 118:207–214.

van Rossum, M. C. W., G. Q. Bi, and G. G. Turrigiano
2000. Stable Hebbian learning from spike timing-dependent plasticity. *J. Neurosci.*, 20(23):8812–21.

van Rossum, M. C. W., M. Shippi, and A. B. Barrett
2012. Soft-bound synaptic plasticity increases storage capacity. *PLoS Comput. Biol.*, 8(12):e1002836.

Vann, S. D., M. W. Brown, J. T. Erichsen, and J. P. Aggleton
2000. Fos imaging reveals differential patterns of hippocampal and parahippocampal subfield activation in rats in response to different spatial memory tests. *J. Neurosci.*, 20(7):2711–2718.

Vasilaki, E., N. Frémaux, R. Urbanczik, W. Senn, and W. Gerstner
2009. Spike-based reinforcement learning in continuous state and action space: when policy gradient methods fail. *PLoS Comput. Biol.*, 5(12):e1000586.

Vesalii Bruxellensis, A.
1543. *scholae medicorum Patavinae professoris, de Humani corporis fabrica Libri septem.* Bruxelles: [publisher unknown].

Vogt, S. M., O. Christ, G. Zeplin, and U. G. Hofmann
2010a. Open control and online analysis software for your biomedical recording hardware. In *Biomed. Tech.*, volume 55, Pp. 2–5.

Vogt, S. M. and U. G. Hofmann
2009. A unified open-source recording and online analysis software for deep brain recordings. In *Front. Neuroinform.*

Vogt, S. M. and U. G. Hofmann
2012. Neuromodulation of STDP through short-term changes in firing causality. *Cogn. Neurodyn.*, 6(4):353–366.

Vogt, S. M. and U. G. Hofmann
2015a. Prolonging a spiking neuron's critical period through attractorless STDP and random growth. (in preparation).

Vogt, S. M. and U. G. Hofmann
2015b. Reinforcing STDP through contrast modulation of synaptic transmission: Biological mechanisms. (in preparation).

Vogt, S. M. and U. G. Hofmann
2015c. STDP: Garbage In, Garbage Out! (in preparation).

Vogt, S. M. and U. G. Hofmann
2015d. The pattern-triggered average: A robust spike train analysis method for imprecise neural spike codes. (in preparation).

Vogt, S. M., M. Klostermann, A. Kundu, S. Andruschenko, and U. G. Hofmann
2009. Hardware-in-the-Loop Testing for closed-loop Brain Stimulators. In *4th Eur. Conf. Int. Fed. Med. Biol. Eng.*, J. Vander Sloten, P. Verdonck, M. Nyssen, and J. Haueisen, eds., volume 22 of *IFMBE Proceedings*, Pp. 1128–1132, Berlin, Heidelberg. Springer Berlin Heidelberg.

Vogt, S. M., K. Mankodiya, M. Klostermann, and U. G. Hofmann
2010b. Programming Smart Phone Processors in Medical Applications. P. 965124.

Volterra, A. and J. Meldolesi
2005. Astrocytes, from brain glue to communication elements: the revolution continues. *Nat. Rev. Neurosci.*, 6(July):626–640.

Voorn, P., L. Vanderschuren, H. Groenewegen, T. Robbins, and C. Pennartz
2004. Putting a spin on the dorsal–ventral divide of the striatum. *Trends Neurosci.*, 27(8):468–474.

Waelti, P., A. Dickinson, and W. Schultz
2001. Dopamine responses comply with basic assumptions of formal learning theory. *Nature*, 412(6842):43–48.

Wan, H., E. C. Warburton, P. Kusmierek, J. P. Aggleton, D. M. Kowalska, and M. W. Brown
2001. Fos imaging reveals differential neuronal activation of areas of rat temporal cortex by novel and familiar sounds. *Eur. J. Neurosci.*, 14:118–124.

Wang, H.-X., R. C. Gerkin, D. W. Nauen, and G.-Q. Bi
2005. Coactivation and timing-dependent integration of synaptic potentiation and depression. *Nat. Neurosci.*, 8(2):187–93.

Wang, Z., L. Kai, M. Day, J. Ronesi, H. H. Yin, J. Ding, T. Tkatch, D. M. Lovinger, and D. J. Surmeier
2006. Dopaminergic control of corticostriatal long-term synaptic depression in medium spiny neurons is mediated by cholinergic interneurons. *Neuron*, 50(3):443–52.

Waters, J. and F. Helmchen
2006. Background synaptic activity is sparse in neocortex. *J. Neurosci.*, 26(32):8267–77.

Wilson, C. J.
1998. *The Synaptic Organization of the Brain, Fourth Edition.* Oxford University Press, USA.

Wittenberg, G. M. and S. S.-H. Wang
2006. Malleability of spike-timing-dependent plasticity at the CA3-CA1 synapse. *J. Neurosci.*, 26(24):6610–7.

Zhao, Y., C. A. Rothkopf, J. Triesch, and B. E. Shi
2012. A unified model of the joint development of disparity selectivity and vergence control. In *2012 IEEE Int. Conf. Dev. Learn. Epigenetic Robot.*, Pp. 1–6. IEEE.