VILNIUS UNIVERSITY

Dalius Krunglevičius

STDP LEARNING OF SPATIAL AND SPATIOTEMPORAL PATTERNS

Doctoral Dissertation

Physical Sciences, Informatics (09P)

Vilnius, 2016

Dissertation work was carried out at the Faculty of Mathematics and Informatics of Vilnius University from 2011 to 2015.

Scientific Supervisor

Prof. Dr. habil. Šarūnas Raudys (Vilnius University, Physical Sciences, Informatics - 09P)

VILNIAUS UNIVERSITETAS

Dalius Krunglevičius

STDP MOKYMO TAIKYMAS ERDVINĖMS BEI ERDVINĖMS-LAIKINĖMS STRUKTŪROMS ATPAŽINTI

Daktaro disertacija

Fiziniai mokslai, informatika (09P)

Vilnius, 2016

Disertacija rengta 2011-2015 metais Vilniaus universiteto Matematikos ir informatikos fakultete.

Mokslinis vadovas

prof. habil. dr. Šarūnas Raudys (Vilniaus universitetas, fiziniai mokslai, informatika - 09P)

Acknowledgements

I’m sincerely grateful to professor Šarunas Raudys for his support, help and belief in the significance of my work. I would like to thank professor Adam Czajka for providing me with data-sets for my experiments and Geoff Vasil for helping with the editing.

And my deepest gratitude goes to my wife, Gražina, without whose patience and support this dissertation would have never been written.

Table of Contents

Notation

ANN – artificial neural network.

STDP – spike-timing-dependent plasticity.

LTD – long term depression.

LTP – long term potentiation.

PSP – postsynaptic potential.

EPSP – excitatory postsynaptic potential.

IPSP – inhibitory postsynaptic potential.

SRM – spike response model.

PDF – probability distribution function.

WTA – winner-take-all.

ϑ – neuron threshold.

w – synaptic strength/weight.

ϕ – synaptic strength factor.

u – membrane potential.

ϵ – SRM response kernel function which describes the response over time to an incoming spike.

η – response kernel function which describes the action potential and after-hyperpolarization.

μ – mean.

Introduction

Motivation and the Field of Research

The large goal in the scientific field of machine learning is to achieve human-level cognition and eventually artificial intelligence, a thinking machine capable of human-level reasoning and beyond. This aspiration remains for now largely in the realm of science fiction, but it is the direction of and motivation for continuing scientific research. Spin-offs from these efforts include a large variety of machines which are applicable to the practical tasks of heuristic optimization, pattern recognition, prediction, data clustering, dimensionality reduction and other jobs. Machine learning is applied in multiple fields of human endeavor, from predicting financial trends to medical diagnosis, from entertainment to industrial engineering.

There are multiple approaches for building systems which are capable of learning, and most of them are based on mimicking nature, such as Darwinian evolution, the behavior of swarms, etc. In the emulation of human-level cognition, there are of course attempts to mimic the behavior of the central nervous system. There are two distinct approaches, one being the top-down approach, where principles of machine learning are based, for example, on assumptions about "what the brain *does*," and researchers rely upon human or animal psychology. The most prominent example of such an approach is reinforcement learning, based on the behaviorist paradigm within the field of psychology (Sutton & Barto, 1998). Another approach is the bottom-up one where principles of learning are based on "how the brain *works*," the physiology of the brain. Mimicking the behavior of neurons and artificial neural networks are the best example of this approach. Multiple families of artificial neural networks were developed which have been applied widely in practical machine learning, including the Perceptron, the Self-Organizing-Map and others. Wolfgang Maass distinguishes three generations of artificial neural networks (ANNs): the first generation of ANNs was based on McCulloch-Pitts neurons;

the second generation has a continuous activation function, such as a sigmoid function; and the third generation is spiking neural networks (Maass, 1997).

On the subject of artificial neural networks, there are two related but very distinct fields of scientific inquiry, one being computational neuroscience, which is a branch of theoretical neuroscience which studies the function of the brain and the processes of structures in the brain and biological neurons; and the other being machine learning, which is a branch of computer science. The aim of neuroscience is to explain the brain, while the aim of machine learning is to build a practical machine. Because these are very different goals, the two fields are not always in agreement. As an example, the Perceptron was constructed based on contemporary knowledge and assumptions about the behavior of biological neurons, but we know now with a high degree of certainty that biological neurons behave quite differently. For instance, the multilayer Perceptron network is commonly trained by back-propagation learning. No such process exists in biological neurons; there is no supervised learning at all. Despite that, Perceptron neural networks have been developed and researched for decades because they are a very useful tool for classification and prediction. On the other hand, a number of even prominent theories of neural learning developed in the field of computational neuroscience are almost completely ignored by machine learning because they never find a practical application. The most prominent examples are the Bienenstock-Cooper-Munro (BCM) theory of synaptic plasticity (Bienenstock et al.,1982) and Adaptive Resonance Theory (ART) (Grossberg, 2013).

My dissertation belongs rightly in the field of connectionism (Elman, 1998). I combine the fields of neuroscience, knowledge of the plasticity of biological neurons and models emanating from computational neuroscience for attacking the problem of pattern recognition. The term connectionism was coined by Donald Hebb in the 1940s, and it is therefore an old and established field of scientific inquiry. The subject of my research is spiking neural networks and spike-timing-dependent plasticity (STDP) from the perspective of the learning

of spatial and spatiotemporal patterns. STDP is a form of Hebbian learning discovered in biological neurons. Although models of neurons developed in the field of computational neuroscience can reproduce some aspects of biological neurons with astonishing accuracy, one cannot just take an existing model and apply it to real-world data: there are multiple problems associated with this, for example, the seemingly simple problem of how to encode data turns out to be far from trivial. Other problems are connected with the stability of the network and optima of parameters. There is a phenomenological variety of STDP implementations and each has its own distinct behavior. At the present time it remains unclear which is the most accurate, and which one is the best choice for a given task. Put simply, the goal of my research was to master spiking neural networks with STDP learning for the task of pattern recognition. In the process I identified a number of problems, and solved several of them.

Objectives of Research and Problems

The primary objective of my research was to apply STDP learning to spatiotemporal and spatial pattern recognition. I identified and solved problems as my research progressed.

When I began my research, there were very little works done in the field. It was known that a specific STDP rule can act as a coincidence detector, that it can detect the beginning of a spatiotemporal pattern injected into Poisson noise (Masquelier et al., 2009; Song et al., 2000; Guyonneau et al., 2005; Gerstner & Kistler, 2002). Several authors have used STDP learning for character recognition (Gupta & Long , 2007; Nessler et al., 2009). Interesting work has been done focusing on evolving networks as well (Kasabov et al., 2012; 2013).

I began my research by repeating the experiments of Masquelier and colleagues (Masquelier et al. 2008; 2009). Masquelier and colleagues concentrated their research on spatiotemporal patterns. STDP training is incapable of creating any spatiotemporal memory, however, and is only capable of spatial pattern memory. I solved this problem by designing a multilayer neural circuit capable of learning

actual spatiotemporal patterns (Krunglevicius, 2011). I concentrated my research on the problems which I identified in implementing my circuit: the parameters of the neuron model can be optimized only for a certain range of stimulation. The amount of stimulation results from the intensity or spatial size of the pattern itself and from different levels of background noise. Moreover, spatial patterns (which can be components of spatiotemporal patterns) can be inclusive of one another, and STDP learning is unable to discriminate one from another. I solved the latter problem by designing a network circuit with distance-dependent synaptic strength factors (Krunglevicius, 2015a). I also studied the problem of variable background noise and solved it with limited success by introducing dynamic adaptive inhibition based on an inverted STDP training window (Krunglevicius 2014).

I attempted to come up with a method for predicting STDP behavior more accurate than those based on Poisson distribution (Izhikevich & Desai, 2003). The problem is that the spike-response-model (SRM) which I use in my work cannot produce Poisson distributed postsynaptic spike trains. Since the SRM reproduces the behavior of the biological neuron quite accurately, it is safe to assume that this is the case in biological neurons as well. Non-Poisson distribution of postsynaptic spikes potentially may have completely different outcomes from Poisson distributed ones when dealing with STDP training. I attempted to use bivariate Azalini distributions for modeling the postsynaptic membrane potential process, but had limited success.

In addition, by applying heuristic optimization for benchmarking the performance of STDP training, I discovered that the STDP triplet interaction function under certain conditions can perform far better than other STDP implementations (Krunglevicius, 2015b). This discovery implies that the triplet interaction rule needs more research than has been done to date.

Relevance

If human-level cognition in machine learning is even achievable, it is impossible to predict its effect across the spectrum of human endeavors; one can only surmise the impact would be immense. My work is only a tiny fraction of a fraction of the whole problem, but does belong to the field generally.

For now, existing computer power is insufficient to use spiking neural networks in practice, but it seems safe to assume this is a temporary obstacle. Understanding the limitations and benefits of spiking neural networks with respect to STDP learning is necessary before attempting to use STDP learning for practical applications in the future.

Practical Value of the Research

Although it is still premature to use STDP for practical applications, I did solve a few important problems associated with STDP learning. In other words, this work is a step forward on the way to building a practical machine. At this time, however, it is not possible to determine how efficient such a machine would be.

Aim of the Research

The main aim of the research was to improve existing competitive neural networks with STDP learning (Guyonneau et al., 2004; Masquelier et al., 2009) by implementing novel neural circuits which could learn sequences of spatial patterns and differentiate similar spatial patterns and to find a way to control the process of training so that neural network could function at different levels of stimulation.

Tasks of the Research

To achieve the main objectives of the research, I completed the following tasks:

- Designed, implemented and tested a novel artificial neural network for learning long-lasting sequences of spatio-temporal patterns.
- Designed, implemented and tested a novel artificial neural network for learning mutually inclusive spatial patterns.

- Designed, implemented and tested a novel artificial neural network for learning spatial patterns in a variable noise environment.
- Experimentally compared different phenomenological models of STDP and determined that under certain conditions the modified triplet interaction model can be more much robust than competitors.

Methods

In my research for modeling spiking neurons I used the spike-response-model (Gerstner & Kistler, 2002) and a few different phenomenological models of STDP, including nearest-neighbor interaction, all-to-all interaction (Morrison et al., 2008) and triplet interaction (Pfister & Gerstner, 2006). I implemented and simulated neural circuits in Matlab and C/C++ environments.

Scientific Novelty

I have designed a neural circuit which is capable of learning long sequences of spatial codes, in other words, spatiotemporal coding. It is not the first model of neural learning and memory based on STDP learning (Szatmáry & Izhikevich, 2010), but I introduced a novel architecture of a multilayer WTA network which is capable of aggregating sequences of events.

I designed a neural circuit for learning overlapping spatial patterns of different size. This is a completely new approach in spiking neural networks based on the assumption that length-dependent conductivity in dendrites plays a crucial role in controlling neural activity and learning.

I used an inverted STDP window for dynamic inhibition when learning took place in conditions of variable background noise. This is a novel method.

I discovered that under certain conditions triplet STDP interaction can perform better by far than a two-spike-based interaction.

Thesis statements

1. There are no theoretical obstacles to using STDP training for spatiotemporal patterns. This supports the hypothesis of spatiotemporal coding in living neural systems.
2. When applying distance-dependent factors to synaptic strengths in a circuit which combines vertical excitation and inhibition, it is possible to build a neural network which is capable of discriminating overlapping spatial patterns and patterns of different spatial size. This approach overcomes the problem of parameters for the model only being capable of optimization over a specific range of stimulation.
3. It is possible to build an STDP-based neural network with adaptive inhibition, and thus it is possible to train it even if the amount of input stimulation varies over time.
4. The STDP triplet interaction rule can lead to a more rapid and more stable training than usual all-to-all and nearest-neighbor interactions.

Approbation

The results of this thesis have been presented at a number of international conferences:

- NCTA 2011 – International Conference on Neural Computation Theory and Applications, Paris, France;
- BISIP 2014 - The 3rd IEEE Workshop on Bio-Inspired Signal and Image Processing, Vilnius, Lithuania;
- NCTA 2014 – International Conference on Neural Computation Theory and Applications, Rome, Italy.

The main results of this thesis were published in the following peer-reviewed journals:

1. Krunglevicius, D. (2011). Neural Processing of Long Lasting Sequences of Temporal Codes - Model of Artificial Neural Network Based on a

Spike Timing-dependant Learning Rule. IJCCI (NCTA), 196-204. SciTePress, (2011).

2. Krunglevicius, D. (2014). STDP Learning under Variable Noise Levels. IJCCI (NCTA), 165-171. SciTePress, (2014).
3. Krunglevicius, D. (2015). Competitive STDP Learning of Overlapping Spatial Patterns. Neural Comput, 27(8):1673-85.
4. Krunglevicius, D. (2016). Modified STDP Triplet Rule Significantly Increases Neuron Training Stability in the Learning of Spatial Patterns. Advances in Artificial Neural Systems, 2016, 1.

Structure of the Dissertation

This dissertation consists of an introduction, three major sections, a conclusion and references. In section 1 I introduce readers to the fundamental basics of neuron biology relevant to my research: the physiology of the neural spike, known kinds of neural plasticity and concepts of neural coding. Section 2 is dedicated to computational neuroscience and is an overview of well-known computational models of the neuron and phenomenological models of STDP. Section 3 details my research and contains five subsections, four of them dedicated to corresponding thesis statements.

My initial work is described in subsection 3.1 ”Neural Processing of Long-Lasting Sequences of Temporal Codes.” There I indicate the problems associated with competitive learning in the spiking WTA circuit, in particular, how to adapt the training neuron to different levels of stimulation, and how to discriminate spatial patterns when a significant portion of these patterns overlap. I address these problems in subsections 3.2 “STDP Learning under Variable Noise Levels” and 3.3 “Competitive STDP Learning of Overlapping Spatial Patterns.” In subsection 3.4 I analyze some statistical properties of the spike-response model in an attempt to predict STDP behavior. This section describes problems associated with predicting STDP behavior and serves both as an explanation and the motivation for the final part of research, subsection 3.5. “Modified STDP Triplet Rule Significantly Increases Neuron Training Stability

in the Learning of Spatial Patterns," where I compare experimentally several models of STDP in the pursuit of the best candidate for improving the neural circuits I introduced earlier in the dissertation.

1. The Physiology of the Neuron and Synaptic Plasticity

1.1. The Spike

Action potential is an event in excitable cells such as neurons and muscle cells where membrane potential rises and falls rapidly. The action potential of neurons usually takes the form of a spike and consequently the terms spike and action potential of neurons are used interchangeably in the literature. In this section I will explain briefly the basic physiological mechanisms of the neuron spike.

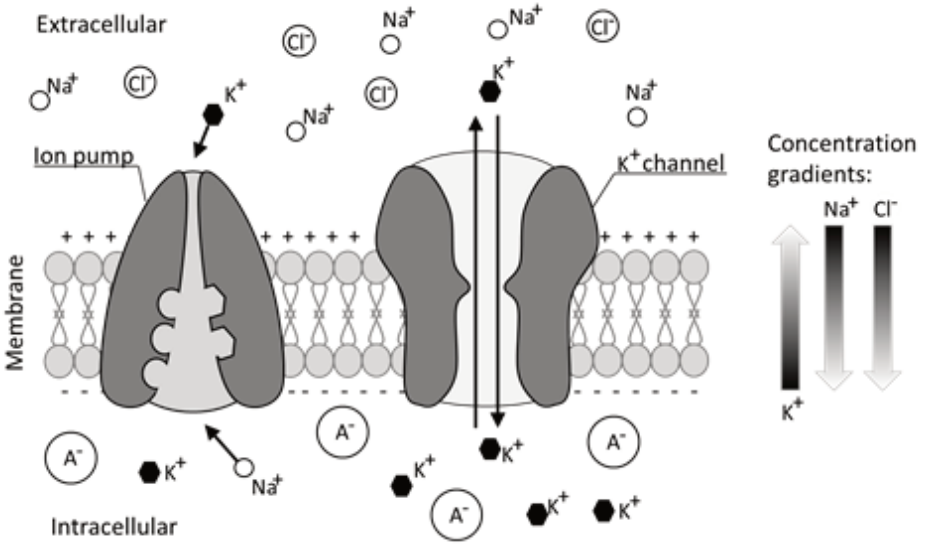

Figure 1.1. In this figure potassium (K^+) can emerge from and enter into the cell through the K^+ channel. Currents of potassium (K^+) ions are caused by concentration and electrical gradients. Diffusion pushes K^+ ions from the cell while the electrical field pushes K^+ ions back into the cell. If these two K^+ currents are equal, the cell is at potassium equilibrium potential. The sodium-potassium pump pumps sodium out of cells and potassium into cells in opposition to their concentration gradients.

Membrane potential is the difference in electric potential between the interior and exterior of a eukaryotic cell. Eukaryotic cells are surrounded by a membrane composed of a lipid bilayer and various embedded proteins. Membrane potential is caused by different concentrations of positive and negative ions throughout the membrane. Inside the cell there is a number of large negative ions called anions which are incapable of passing through the cell membrane. Anions originate from a variety of sources, for example, from amino acids and proteins.

The membrane allows the passage of small ions. The most important ions are potassium (K^+), sodium (Na^+) and chloride (Cl^-) ions. There is a higher concentration of potassium ions on the inside of the membrane than on the outside. Likewise, there is a higher concentration of sodium and chloride on the outside (Kevelaitis et al. 2006). The difference in potassium (K^+) and sodium (Na^+) concentrations is caused mostly by the action of a sodium-potassium pump. The sodium-potassium pump is an enzyme which pumps sodium out of and potassium into cells (Figure 1.1). Ion currents emanating from the cell and entering into the cell occur due to two forces: diffusion, caused by the gradient of concentration; and the electric field.

Different concentrations of ions inside and outside the cell cause concentration gradients, that is, ions tend to move towards lower concentrations due to diffusion. When the current of a particular ion caused by diffusion is equal to the current caused by the electric field, the cell is at the equilibrium potential for that particular ion. Equilibrium potentials can be derived from the Nernst equation:

$$E_x = \frac{R \cdot T}{z \cdot F} \cdot ln \frac{[X_{out}]}{[X_{in}]} \tag{1.1}$$

- R - the ideal gas constant (8.3 J/K· mol);
- T –temperature in Kelvin;
- z - number of moles of electrons;
- F- the Faraday constant (96,500 C/mol);
- $[X_{out}]$ – concentration of ion X outside the cell;
- $[X_{in}]$ – concentration of ion X inside the cell;

The potassium equilibrium potential for cells of warm-blooded animals is approximately E_K=-100 mV and that for sodium is approximately E_{Na}=+65 mV.

The mechanism of action potential has been explained by Hodgkin and Huxley (Hodgkin & Huxley, 1952). Action potential happens due to the activation of voltage-gated ion channels which allow ions to cross the membrane. Ion

channels are mostly selective to a single type of ion. Voltage-gated Na+ ion channels and similar gates have three states: deactivated (not conductive), activated (conductive) and inactived (not conductive). During action potential, these voltage-gated channels progress through the cycle deactivated→activated→inactivated→deactivated (Figure 1.2).

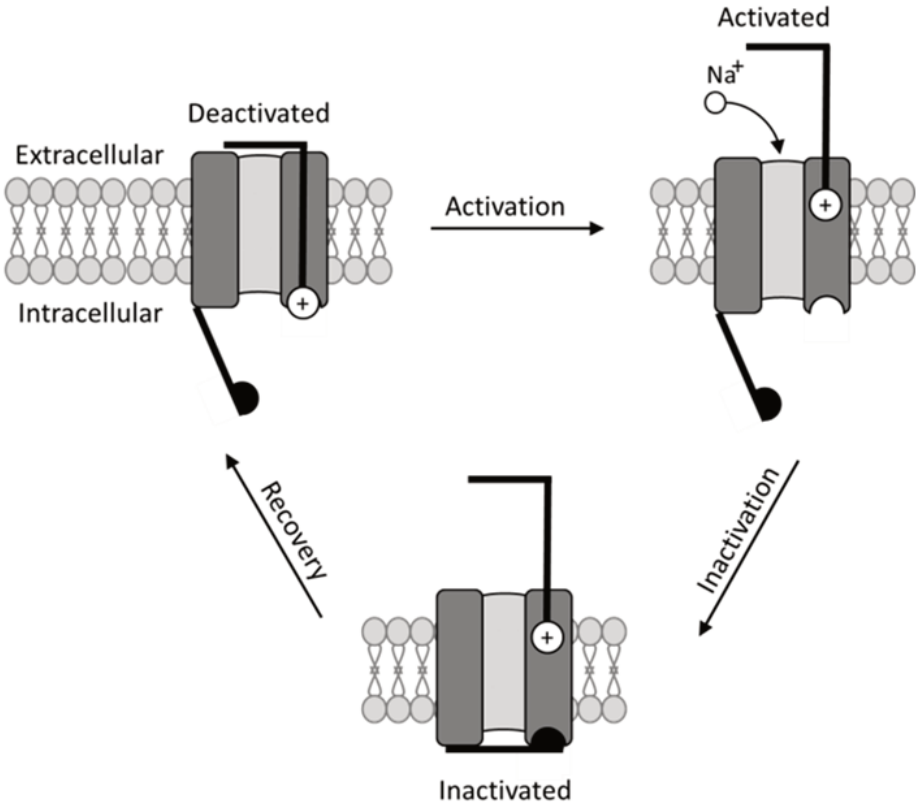

Figure 1.2. States of the voltage-gated sodium channel during action potential. Drawing based on Lehmann-Horn & Jurkat-Rott, 1999.

Neuron membrane resting potential typically is -70 mV (Kevelaitis et al. 2006). If the membrane is stimulated and potential reaches the threshold value which is above resting potential, it causes action potential to fire (see Figure 1.3). Action potential follows the all-or-nothing rule, that is, if the threshold is reached it causes action potential, and if the threshold is not reached there is no action potential. The threshold value of membrane potential activates voltage-gated Na^+ ion channels. This causes Na^+ ions to flow into the cell. This changes the concentration of positively charged cations in the cell and increases membrane potential. This phase of action potential is called depolarization. Potassium

current pushes membrane potential towards E_{Na} equilibrium potential. At the beginning of depolarization K^+ channels also open and this causes K^+ ions to flow outward from the cell. At the peak of depolarization Na^+ and K^+ are equal and therefore total ionic current is 0. Right after depolarization Na^+ current decreases and stops, due to voltage-gated Na^+ ion channels entering the inactive state, while K^+ current continues and increases. The K^+ current decreases membrane potential, and this phase of action potential is called repolarization. It is typical for most neurons that after repolarization membrane potential falls below resting potential (the membrane is hyperpolarized) towards sodium equilibrium potential E_K, and only then gradually returns to normal. This phase is called afterhyperpolarization (Purves et al. 2012).

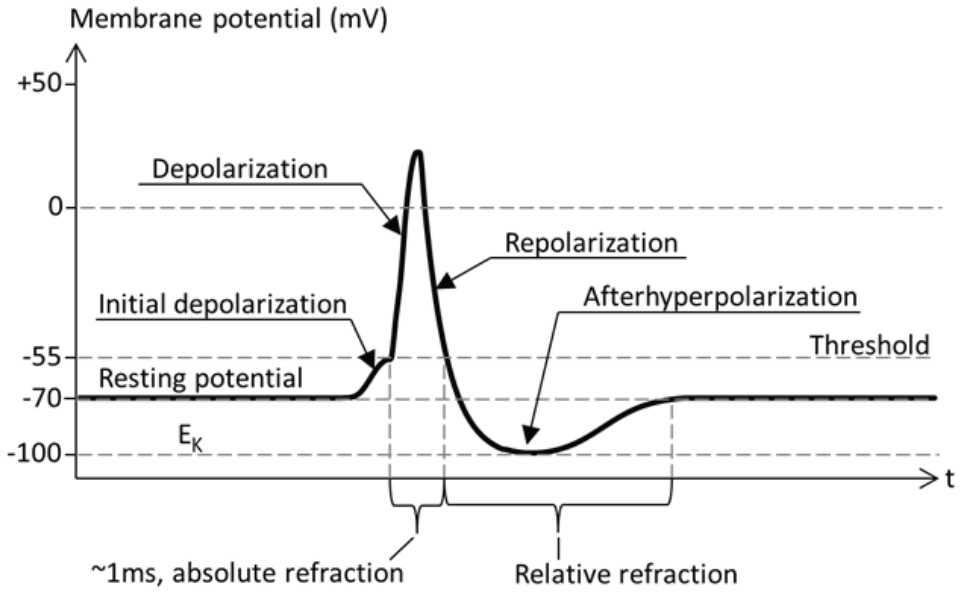

Figure 1.3. Phases of action potential. Drawing based on Kevelaitis et al. 2006.

From the moment of depolarization and until voltage-gated Na^+ ion channels return to the deactivated state, the neuron is in the refractory period. This means that no additional increase of membrane potential can cause action potential. During the afterhyperpolarization phase the neuron is in the relative refractory period, when greater stimulus is required to cause action potential.

1.2. Postsynaptic Potentials and Synapses

The neuron membrane is an excitable medium for action potential to propagate: it occurs locally and then propagates across the entire membrane. When the action potential reaches chemical synapses at the axon terminals of the neuron, it triggers the release of a quantity of chemical messengers called neurotransmitters into the synaptic gap. Neurotransmitters interact with receptors located on the receiving (postsynaptic) side of the synapse. This interaction opens selective ion channels for a brief moment of time and consequently ionic currents occur in the membrane of the postsynaptic neuron. What sort of ion channels open depends on the type of neurotransmitter and the type of receptor. Whether the ionic current is negative or positive depends upon which kinds of ions are allowed to flow and the type of ion. If the ionic current depolarizes the membrane this depolarization is excitatory because it facilitates the firing of action potential. Such a brief increase of membrane potential is called excitatory postsynaptic potential (EPSP). If ionic current hyperpolarizes the membrane, the result is the opposite: this makes it harder for the neuron to fire. Such brief decreases in membrane potential are called inhibitory postsynaptic potentials (IPSPs). Put simply, EPSPs and IPSPs have opposite polarity and may cancel each other out. A neuron always releases the same kind of neurotransmitters, whether just one or a mixture of several. Both presynaptic and postsynaptic neurons define the type of the synapse. Synapses may be inhibitory or excitatory but can never change their function (Kevelaitis et al. 2006).

The amplitude and duration of individual postsynaptic potential (PSP) can be defined by the synaptic strength or the synaptic weight. In the literature strength and weight are often used interchangeably. Synaptic strength depends on the amount of neurotransmitters being released, the density of postsynaptic receptors and other factors such as neurotransmitter reuptake. Reuptake is the job of the presynaptic neuron which pumps back some of the neurotransmitters it has released (Kandel et al., 2000).

It is interesting to note that not all neurons produce action potentials and release quantities of neurotransmitters. Neurons of *Ascaris*, a nematode, for example, seem to release neurotransmitters topically and do not fire spikes (Davis & Stretton, 1989).

Individual PSPs propagate through the membrane and are summed when they meet. There are two important factors which are caused by membrane electrical conductivity and capacity. One is that PSP typically decays exponentially when moving further away from the place of origin. It is a function of the length constant λ (Kandel et al., 2000). The length constants of different neurons typically range from 0.1 to 5 mm (Kevelaitis et al. 2006). In the literature λ is also called the electronic length constant:

$$\Delta V(x) = \Delta V_0 \cdot e^{-x/\lambda} \tag{1.2}$$

where x is the distance and ΔV_0 is the change of membrane potential at the point x=0.

Another important factor is membrane time constant τ which defines the time necessary for the membrane to charge or discharge during the propagation of PSPs. The time constant τ corresponds to the time taken to reach 63% of its final voltage. The time constants of different neurons typically range from 20 to 50 ms (Kandel et al., 2000).

The implication of the length constant is that the closer synapses are, the larger the summed value is, and therefore the proximity of synapses facilitates action potential and *vice versa*. The implication of the time constant is that the closer PSPs are correlated in time, the higher the probability for causing action potential to fire.

It must be noted, however, that propagation of postsynaptic potential in dendrites is a highly complex process; there are multiple non-linearities caused by the morphology of dendrites and likely other properties of the cell as well (van Elburg & van Ooyen, 2010).

Chemical synapses can by classified by their location. Axosomatic synapses are located directly on the soma of the postsynaptic neuron; axodendritic synapses on postsynaptic neuron dendrites and axo-axonic synapses are located at the terminus of the axon of the postsynaptic neuron (Figure 1.4).

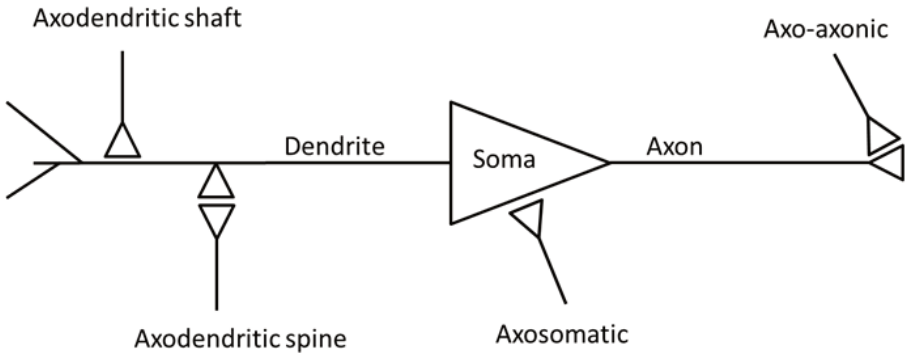

Figure 1.4. Synapses classified by location on the postsynaptic neuron. This schematic does not include all types of synapses, for example, there are axo-extracellular synapses which emit neurotransmitters into extracellular fluid and axo-secretory synapses which emit neurotransmitters into the bloodstream.

Besides chemical synapses, there are electrical synapses called gap junctions. Gap junctions directly connect the cytoplasm of two cells (Lampe & Lau, 2004). Besides many other known functions, gap junctions enable electrical coupling between cells, that is, PSPs and action potential can travel from one cell to another, enabling synchronous firing (Kandel et al., 2000).

1.3. Synaptic Plasticity

Synaptic plasticity is the ability of synapses to change synaptic strength over time. It is a postulate of Hebbian theory that memories are represented by networks of synapses in the brain, and thus synaptic plasticity is the main function involved in learning and memory. In this section I briefly introduce different kinds of synaptic plasticity. This section emphasizes spike-timing-dependent plasticity (STDP), which is the primary subject of my research.

Synaptic plasticity may be short-term, persisting from several milliseconds to a few minutes, and long-term, lasting from minutes to hours or longer. It is

important to note that the properties of plasticity itself may change over time; this kind of plasticity is referred to as metaplasticity (Abraham and Bear, 1996).

Another important kind of synaptic plasticity is homeostatic plasticity. Homeostatic plasticity is a compensatory adjustment of synaptic strengths which allows the neuron to regulate its own activity relative to the activity of the surrounding network. Homeostatic plasticity operates on a timescale of days (Turrigiano & Nelson, 2004; Surmeier & Foehring, 2004).

Short-term plasticity may increase the probability of the release of neurotransmitters in response to a presynaptic spike. Such increases in synaptic strength may occur due to an increase in the quantity of packaged neurotransmitters or an increase in the size of the pool of readily releasable packaged neurotransmitters (Stevens & Wesseling, 1999).

A short-term decrease in synaptic strength is called synaptic fatigue. It is usually caused by depletion of neurotransmitters in vesicles, but can be caused by other presynaptic or postsynaptic processes (Zucker et al., 2002).

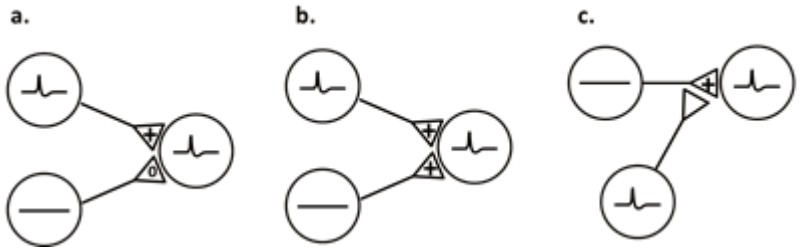

Figure 1.5. Homosynaptic and heterosynaptic plasticity. a: Homosynaptic plasticity. Activity of the presynaptic neuron causes synaptic strength to change. b: Heterosynaptic plasticity. Activity by one presynaptic neuron alters the synaptic strength of the synapse of an idle neuron. c: Heterosynaptic plasticity in an axo-axonic synapse.

Long-term plasticity either increases or decreases synaptic strength. Long-term increases of synaptic strength are referred to as long-term potentiation (LTP), while long-term decreases are referred to as long-term depression (LTD). LTP and LTD can result from an increased or decreased density of postsynaptic glutamate receptors (Malinow & Malenka, 2002), changes in glutamate receptor

function itself (Benke et al., 1998) or changes in the quantity of neurotransmitter released (Weisskopf & Nicoll, 1995). Long-term plasticity can be either homosynaptic or heterosynaptic (Figure 1.5). Homosynaptic plasticity requires activity by a presynaptic neuron (Figure 1.5a), whereas in heterosynaptic plasticity a presynaptic neuron may be idle with plasticity triggered by the activity of another presynaptic neuron (Figure 1.5b). Heterosynaptic plasticity is known to be triggered by a number of neurotransmitters, with serotonin and dopamine notable among them (Bailey et al., 2000). Dopamine plays an important role in learning, motivation and the reward system (Wise, 1996). Thus it is reasonable to assume heterosynaptic plasticity facilitates reinforcement in learning. The heterosynaptic plasticity of axo-axonic neurons (Figure 1.5c) was the first kind of synaptic plasticity discovered, in the sea slug *Aplysia californica* (Purves et al. 2012).

1.4. Spike-Timing Dependent Plasticity

Spike-timing-dependent plasticity is the main subject in my research. STDP is a kind of long-term homosynaptic plasticity. It is a function of the time difference between presynaptic and postsynaptic spikes which regulates the amount of change in synaptic strength. STDP learning windows typical for excitatory-to-excitatory synapses (Figure 1.6) were discovered relatively recently, and the first experiments with precisely timed pre- and postsynaptic spikes at a one-millisecond temporal resolution were performed by Markram and colleagues (Markram et al., 1995, 1997), followed later by other researchers (Bi & Poo 1998, Debanne et al. 1998, Magee & Johnston 1997, Zhang et al. 1998). Later still (Caporale & Dan, 2008) the rule (Figure 6) was observed in a wide variety of biological neural systems (Boettiger & Doupe, 2001; Cassenaer & Laurent, 2007; Egger et al., 1999; Feldman, 2000; Froemke & Dan, 2002; Sjostrom et al., 2001, Tzounopoulos et al., 2004).

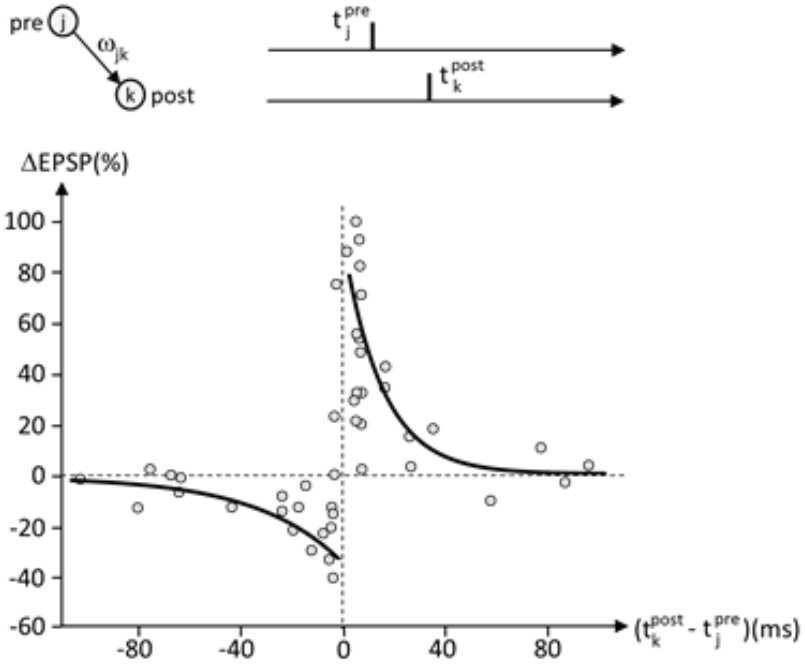

Figure 1.6. STDP function. Markers represent the change in synaptic efficiency in a rat hippocampal neuron. Here EPSP is an excitatory postsynaptic current. Data points redrawn from Bi and Poo (1998). Lines are only an approximation of the experimental data.

A variety of other STDP rules have been discovered for different types of synapses. Figure 1.7 is a comparison of two known STDP rules for excitatory-to-excitatory synapses. Figure 1.7a is the same rule as in Figure 1.6. Figure 1.7b is an STDP rule observed in CA1 neurons (Nishiyama et al., 2000; Wittenberg & Wang, 2006).

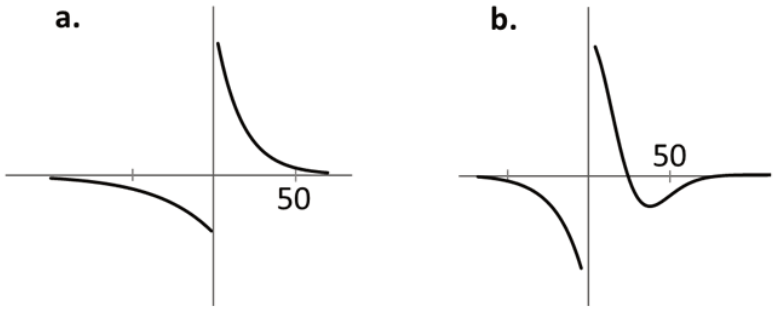

Figure 1.7. Known STDP rules for excitatory-to-excitatory synapses. Drawing based on Caporale and Dan (2008).

Excitatory-to-inhibitory synapses seem to show an inverted STDP window when compared to excitatory-to-excitatory synapses. Figure 1.8a is an STDP rule

found in Purkinje-like GABAergic neurons of *Mormyridae* electric fish. A similar rule, but with LTP absent, has been observed in mouse cartwheel neurons (Tzounopoulos et al., 2004, 2007).

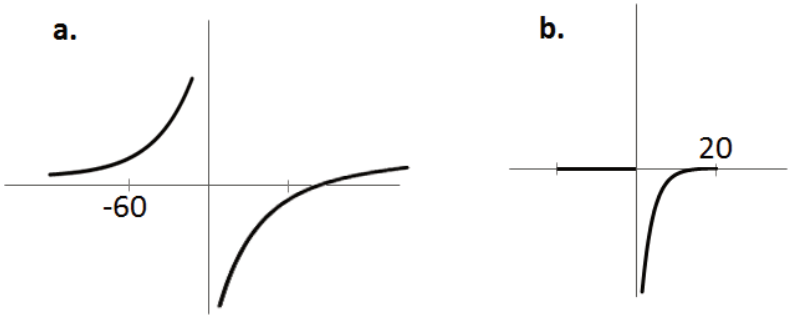

Figure 1.8. Known STDP rules for excitatory-to-inhibitory synapses. Drawing based on Caporale and Dan (2008).

There are several STDP rules for GABAergic inhibitory synapses which are quite dissimilar to one another. This likely indicates a variety of neural inhibition functions (Figure 1.9). Figure 1.9a. illustrates an STDP rule for inputs of neocortical L2/3 pyramidal neurons (Holmgren & Zilberter, 2001). Figure 1.9b is a symmetric STDP window observed in the hippocampus, GABAergic synapses to CA1 pyramidal neurons (Woodin et al., 2003). Figure 1.9c is a rule observed in the entorhinal cortex (Haas et al., 2006).

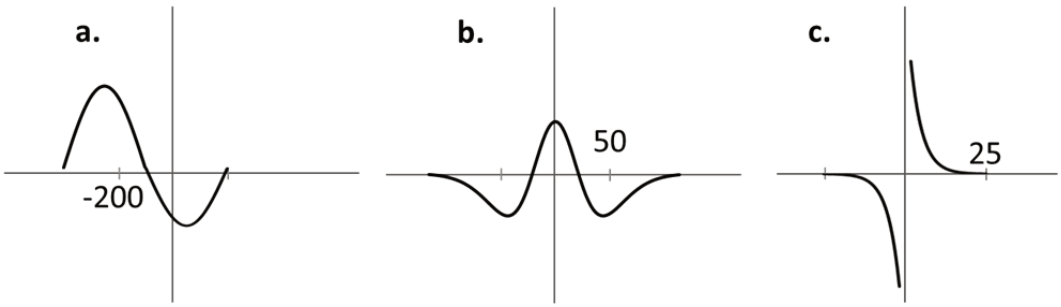

Figure 1.9. Known STDP rules for inhibitory-to-excitatory synapses. Drawing based on Caporale and Dan (2008).

It must be emphasized that the STDP learning windows discussed briefly above are just approximations of experimental data. There are multiple observed non-linearities (Sjostrom et al., 2001; Wang et al., 2005). Some STDP rules depend not only on the type of the synapse, but also on the location on the dendrite, it

seems (Caporale & Dan, 2008) (Figure 1.10). Time constants for STDP decay and amplitude can vary along the dendrite (Froemke et al. 2005) (Figure 1.10a). It was also observed that in some synapses located away from the soma some neurons had a different STDP window polarity than in synapses closer to the soma. This has been observed in the visual cortex (Figure 1.10b) (Sjostrom & Hausser 2006) and the barrel cortex (Figure 1.10c) (Letzkus et al. 2006).

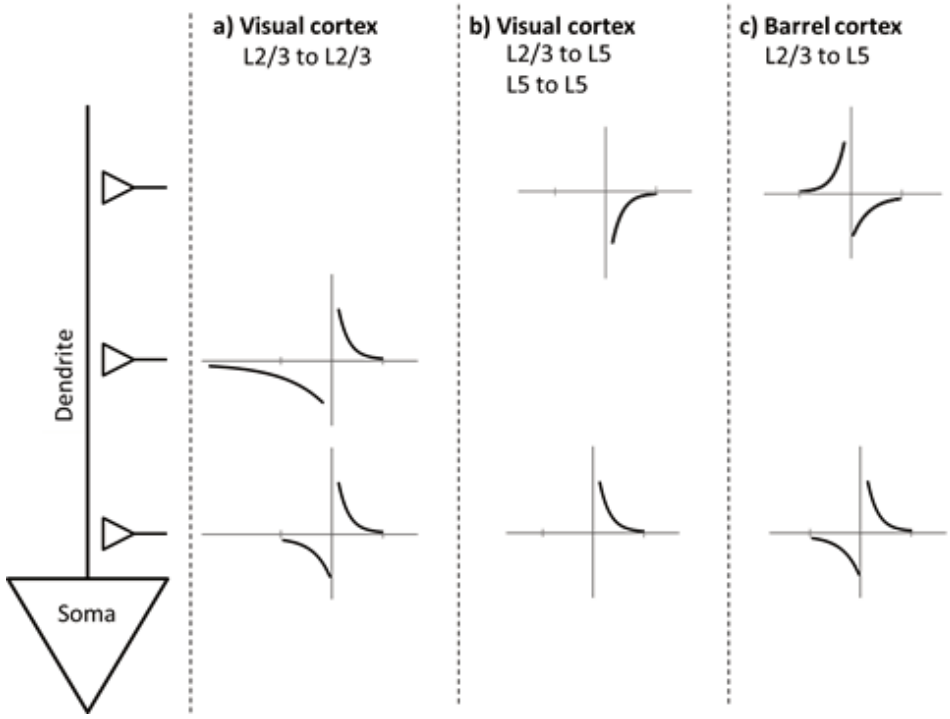

Figure 1.10. STDP dependence on dendrite location. Drawing based on Caporale and Dan (2008).

1.5. Neural Coding

The central nervous system receives input stimuli via the sensory systems and processes the data further. The data are encoded in sequences of action potentials, also referred to as spike trains. Some sensory systems, touch for example, have been studied intensively, and we understand the meaning of the spike trains, that is, we understand how sensory data are encoded. This is not the case in general, however, especially when we look at neural communications inside the brain. Neuroscientists are only just beginning to understand how to decode spike trains. At this time the actual neural "language" is not understood.

In this section I discuss two major paradigms of neural coding: rate coding and temporal coding. I will omit discussion of neural coding schemes which are not directly part of my research.

1.5.1. Rate Coding

The rate-coding model assumes that the frequency of spikes or count of spikes in a specific temporal window encodes information. Since a single spike typically takes approximately one millisecond, the rate code is typically a value in the range of from 0 to 1 kHz. In rate coding, any temporal structure in the spike train is ignored. This makes rate coding very robust to noise.

Rate coding was discovered almost a century ago by Adrian and Zotterman (Adrian & Zotterman, 1926). Since then it has been confirmed in most sensory systems (Kandel et al., 2000). For decades rate coding was a standard tool for describing neural communications. Rate coding coincides perfectly with many concepts of artificial neural networks in machine learning such as the Perceptron, the Self-Organizing Map and etc., where neuron input and output values can simply be a real number.

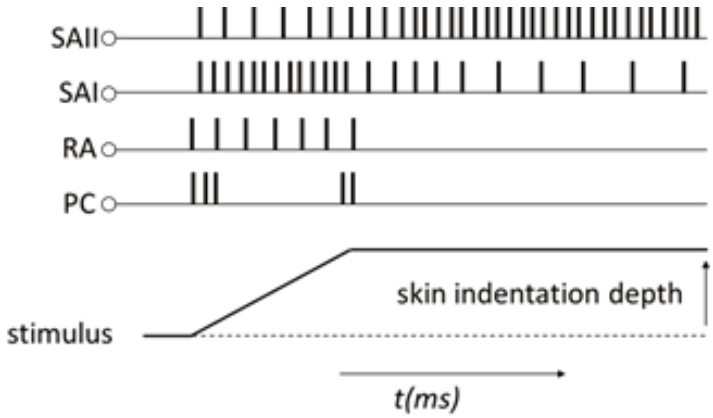

Figure 1.11. Mechanosensory coding of touch stimulus in the skin. The bottom curve represents the depth of the indentation of the skin, the spike patterns represent responses of different mechanoreceptors located in the skin. SAII is the Ruffini ending, SAI is Merkel's disk, RA is Meissner's corpuscle and PC is the Pacinian corpuscle. Adapted from Kevelaitis et al. (2006).

One of the best examples of rate coding is the output from mechanosensory receptors in the skin (Figure 1.11). Mechanoreceptors of the skin are classified

by their adaptation speed. There are receptors which are slowly adapting (SA) and rapidly adapting (RA). Slowly adapting receptors are Merkel's disks (SAI) and Ruffini endings (SAII). In Figure 1.11 we see Merkel's disks and Ruffini endings are active over the duration of the stimulus. Meissner's corpuscles are rapidly adapting receptors and they respond to the velocity of the stimulus. Acceleration of stimulus is detected by Pacinian corpuscles, classified as very rapidly adapting receptors (Kevelaitis et al. 2006).

In recent years multiple experimental findings have suggested the rate coding approach might be too simplistic to explain data processing in the brain (Stein et al., 2005). There is evidence, for example, that rate coding alone cannot account for the efficiency of information transmission in some biological neural systems (Gerstner et al., 1996; van Rullen & Thorpe, 2001).

1.5.2. Temporal Coding

When information is encoded in the precise timing of the spike or in high frequency fluctuations of firing rate, this is often referred to as a temporal coding (Dayan & Abbott, 2001). A number of studies have discovered evidence that precise spike timing is a significant element of neural coding (Butts et al., 2007; Kayser et al. 2009; Thorpe, 1990).

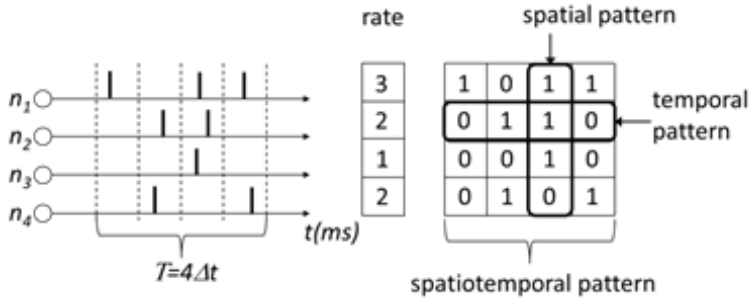

Figure 1.12. Examples of neural coding schemes.

In Figure 1.12 four neurons in parallel produce four spike trains. Spikes may be measured across a specific temporal window, resulting in rate coding. On the other hand, the temporal window may be converted to logical bits by dividing the temporal window into smaller units resulting in a 1 if a spike occurred or a 0

if the spike has not occurred. This kind of interpretation is called temporal coding. In the case of rate coding, spike trains 0110 and 0101 have the same value, 2, but in temporal coding these two spike trains can encode two different values. When multiple temporal patterns are combined into a single pattern, it is called a spatiotemporal pattern. A single frame of the spatiotemporal pattern is called a spatial pattern.

A special case of temporal coding is rank-order coding, where information is encoded in the order of spikes arriving from ranked neurons. In this case, the exact latency at which a neuron fires is not critical, and only the rank order of each neuron is important (Gautrais & Thorpe, 1998).

1.5.3. Phase-of-Firing Code

Phase-of-firing code is a neural coding scheme which combines oscillations in the neural system with spike firing times or firing rate. Oscillations could be waves in the central nervous system such as alpha waves, or oscillations in external stimuli. Phase-of-firing code takes into account the timing of the spike in relation to the phase of oscillation by labeling spikes according to phase (Montemurro et al. 2008; Fries et al., 2007; Kayser et al. 2009).

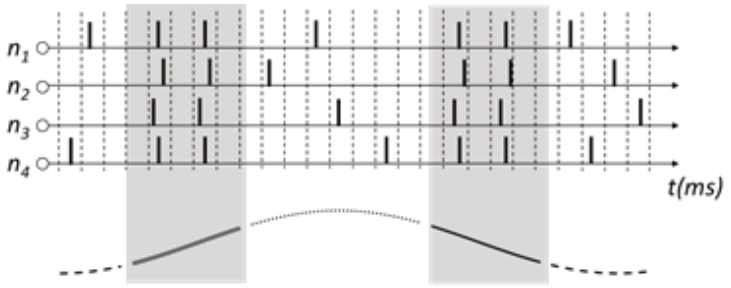

Figure 1.13. Schematic representation of phase-of-firing code. Drawing based on study of the auditory cortex of primates, adapted from Kayser et al. (2009). At the phases of stimuli denoted by solid lines neurons produce spatial patterns, but at phases denoted by dashed and dotted lines neurons do not produce a persistent pattern although the firing rate is persistent.

Strong evidence has been found of phase coding in the visual cortex (Havenith et al., 2011) and auditory cortex (Kayser et al. 2009).

1.5.4. Population Coding

Under the concept of population coding, the brain encodes information in joint activity by populations of neurons. Yet again population coding may involve a combination of firing rates in a population, or it may be a temporal code, where information is encoded in correlations between spike trains of different neurons. In such cases spatial or spatiotemporal patterns are examples of population coding.

At the moment, population coding is the most accurate method for decoding information from neural activity. Pasley and colleagues (Pasley et al., 2012), for example, successfully reconstructed human speech from neuron activity in the non-primary auditory cortex. Researchers used 15 patients who were undergoing neurological procedures for epilepsy or brain tumor treatment. This allowed placing multi-electrode arrays over the lateral temporal cortex and recording the activity of populations of neurons.

Temporal population coding, or spatial and spatiotemporal temporal patterns, is the primary subject of my research. It has been hypothesized that spike-timing-dependent plasticity of excitatory-to-excitatory synapses (see section 1.4 “Spike-Timing Dependent Plasticity”) is a confidence detector (Abbott & Nelson, 2000), in other words it is capable of detecting correlations between multiple spike trains. It has been demonstrated in multiple studies (Masquelier et al., 2008, 2009; Song et al., 2000; Guyonneau et al., 2005) that STDP is capable of learning spatiotemporal patterns and detecting the beginning of such patterns even if the occurrence of such patterns is not periodic (Masquelier et al., 2008, 2009).

2. Phenomenological Models of Spiking Neuron and Synaptic Plasticity

2.1. Spiking Neuron Models

2.1.1. Hodgkin-Huxley Model

Hodgkin and Huxley (Hodgkin & Huxley, 1952) experimented with the giant axon of the squid. Giant squid axons reach up to 1 mm in diameter, a real gift of nature to researchers. Hodgkin and Huxley discovered three different types of ionic currents: sodium, potassium and the leak current which consisted mostly of chlorine (for further details see section 1.1 "The Spike"). Hodgkin and Huxley developed the model of the spiking neuron based on their findings. They received the 1963 Nobel Prize in Physiology and Medicine for this work.

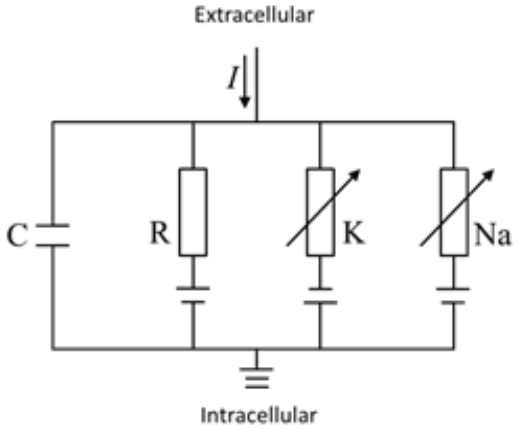

Figure 2.1. Components of the Hodgkin–Huxley model

Hodgkin-Huxley is a detailed neural model which allows for the modeling of different ion channels. Although this model is quite old and rather complex, it is still used today. It has been improved and extended in multiple ways (Forrest, 2014; Pakdaman 2010). The Hodgkin-Huxley model is also an important reference model for the derivation of simplified neuron models. In this section I briefly describe the original Hodgkin-Huxley model without going into too much details, since I don't use it directly in my work. An understanding of the Hodgkin-Huxley neuron is critical, however, for understanding simplified and

formal models of the spiking neuron. This explanation of the model is based on Gerstner and Kistler (Gerstner & Kistler, 2002).

The Hodgkin-Huxley model can be portrayed as the electronic circuit depicted in Figure 2.1. When input current I is injected into the cell, it either charges capacitor C, which represents a lipid bilayer, or the input current leaks through ion channels R, K or Na. Here K represents the potassium ion channel, Na the sodium ion channel and R a leakage channel of unspecified ions with resistance R. Thus at the moment of time t, the input current might be split into:

$$I(t) = I_C(t) + \sum_k I_k\,(t) \tag{2.1}$$

where I_C is the capacitive current which charges capacitor C and I_k are the components which pass through the ion channels. Capacitive current may be expressed as $I_C = C\,du/dt$. Hence:

$$C\frac{\mathrm{d}u}{\mathrm{d}t} = I(t) - \sum_k I_k\,(t) \tag{2.2}$$

All channels can be characterized by their resistance or by their conductance g. The conductivity of the leaky channel is voltage independent and is simply $g_L = 1/R$. Sodium and potassium ion channels are voltage-gated (for more details, see section 1.1 "The Spike"). If the sodium and potassium channels are open, they transmit currents with maximal conductance g_{Na} and g_{K}. The voltage-gated ion channels, however, are not always open. The probability of the channel being open is described by additional variables m, n and h. The sodium (Na^+) voltage-gated channel has three states: deactivated, activated and inactived. Therefore it requires two variables to describe probability: m and h. For the potassium (K^+) voltage-gated channel Hodgkin and Huxley used the single variable n, that is, the channel can be in one of two states: active or closed. In other words, the potassium channel lacks an inactive state. It can be noted that we now know there are potassium channels which do have an inactived state as well (Kevelaitis

et al., 2006). In the original Hodgkin-Huxley model this property is ignored, and thus the sum of ionic currents is expressed by:

$$\sum_k I_k = g_{Na} m^3 h(u - E_{Na}) + g_K n^4 (u - E_K) + g_L (u - E_L) \qquad (2.3)$$

where the parameters E_{Na}, E_K and E_L are reversal potentials. The values of conductance and reversal potentials are empirical parameters. Table 2.1 has the original values reported by Hodgkin and Huxley.

Table 2.1. The parameters for the Hodgkin-Huxley equation.

x	E_x (mV)	g_x (mS/cm^2)
Na	115	120
K	-12	36
L	10.6	0.3

The variables *m*, *n* and *h* are called gated variables. They are dimensionless quantities between 0 and 1 and they evolve according to the differential equations:

$$\frac{dm}{dt} = \alpha_m(u)(1 - m) - \beta_m(u)m$$

$$\frac{dn}{dt} = \alpha_n(u)(1 - n) - \beta_n(u)n \qquad (2.4)$$

$$\frac{dh}{dt} = \alpha_h(u)(1 - h) - \beta_h(u)h$$

where α and β are empirical functions which were originally fitted by Hodgkin and Huxley to the data from the giant axon of the squid:

$$\alpha_n(u) = \frac{0.1 - 0.01u}{e^{1-0.1u} - 1}; \quad \alpha_m(u) = \frac{2.5 - 0.1u}{e^{2.5-0.1u} - 1}; \quad \alpha_h(u) = 0.07\mathrm{e}^{-0.05u};$$

$$\beta_n(u) = 0.125\mathrm{e}^{-0.0125u}; \quad \beta_m(u) = 4\mathrm{e}^{-0.055555u}; \quad \beta_h(u) = \frac{1}{e^{3-0.1u} + 1} \tag{2.5}$$

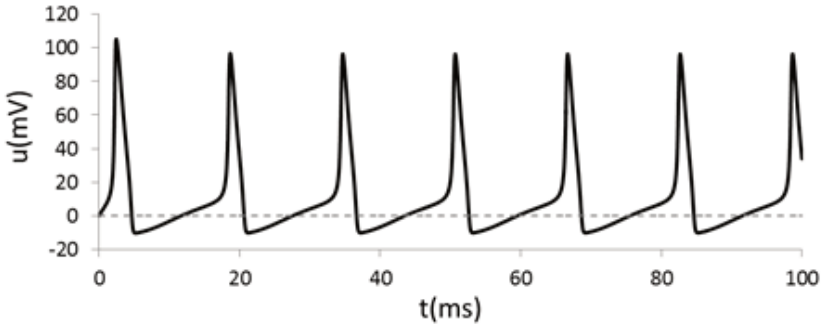

Figure 2.2. Periodic spikes of the Hodgkin-Huxley model in response to constant input current *I*.

Figure 2.2 shows the response of the Hodgkin-Huxley model to a constant input current. We see a spike train with each spike followed by the afterhyperpolarization phase. By changing the value of the input current, the frequency of spikes increases or decreases correspondingly. Within a certain range of low input current, the neuron only fires once. Constant input is just the simplest example of the Hodgkin-Huxley model's behavior. Multiple biologically plausible effects can be achieved by manipulating the input current.

Detailed neuron models such as the Hodgkin-Huxley model are capable of reproducing electrophysiological measurements to a high degree of accuracy. Such models are complex, however, and that makes them more difficult to analyze. Simple phenomenological spiking neuron models are more popular in the study of network dynamics, neural coding and memory (Gerstner & Kistler, 2002). Neural coding is the primary subject of my work, and not surprisingly my choice of research subject was a formal neural model. In subsequent sections I

will briefly introduce two formal models of the spiking neuron: the leaky integrate-and-fire and the spike response model. In my work I have used the spike response model, which is a generalization of the leaky integrate-and-fire neuron.

2.1.2. The Leaky Integrate-and-Fire Model

The leaky integrate-and-fire neuron model is probably the best known model of the formal spiking neuron (Gerstner & Kistler, 2002). Integrate-and-fire neurons can be stimulated by external current or by the modeling of synaptic inputs from presynaptic neurons.

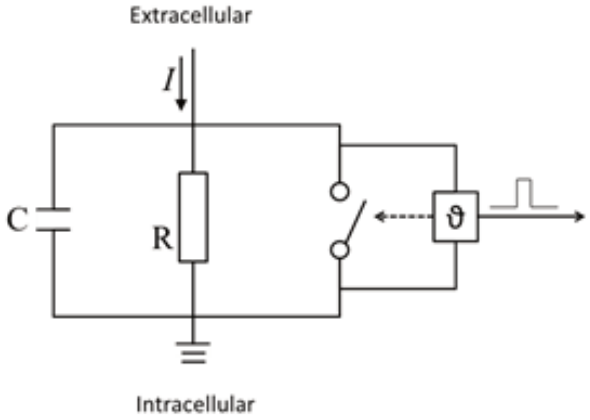

Figure 2.3. Schematic diagram of the leaky integrate-and-fire neuron model.

The basic circuit of leaky integrate-and-fire is composed of capacitor C and resistor R (Figure 2.3). The driving current is a sum of resistive current I_R which passes through linear resistor R and current I_C which charges capacitor C. Thus

$$I(t) = I_R(t) + I_C(t) \tag{2.6}$$

I_R can be calculated using Ohm's law $I_R=u/R$; from the definition of capacity $C=q/u$, where q is the charge and u is the voltage, so that capacitive current is $I_C=C\ du/dt$. Thus:

$$I(t) = \frac{u(t)}{R} + C\frac{du}{dt} \tag{2.7}$$

The standard form of the leaky integrate-and-fire neuron is obtained from equation (2.7) by introducing the time constant $\tau_m=RC$ and multiplying the equation by R:

$$\tau_m \frac{du}{dt} = RI(t) - u(t) \tag{2.8}$$

The spike in the leaky integrate-and-fire neuron is a formal event, it is not described explicitly. The spike is characterized by firing time $t^{(f)}$ which occurs when voltage reaches the threshold ϑ. So the threshold criterion is $u(t^{(f)})=\vartheta$ (see Figure 2.3). Immediately after the spike, potential is reset to a new value, $u_r < \vartheta$.

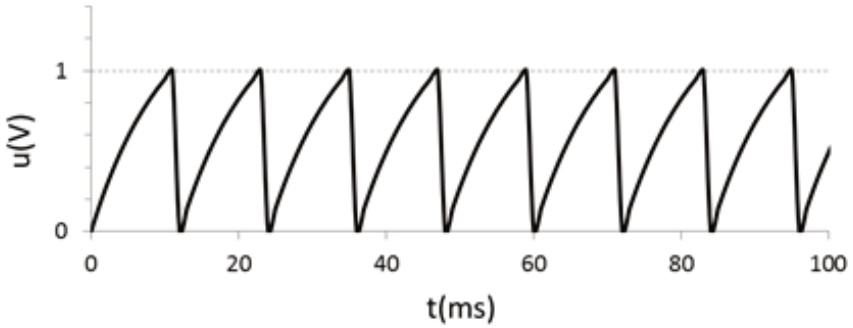

Figure 2.4. Periodic spikes of leaky integrate-and-fire model in response to constant input current I=1.5. Resistance R was set to 1, τ_m=10, threshold ϑ=1, and reset potential u_r=0.

In the case of a constant input current I_0, assuming that the spike occurred at $t= t^{(1)}$ with initial conditions $u(t^{(1)})= u_r = 0$, the solution for equation (2.8) is:

$$u(t) = RI_0\left(1 - e^{-\frac{t-t^{(1)}}{\tau_m}}\right) \tag{2.9}$$

It is easy to understand the behavior of equation (2.9). Membrane potential approaches the value *RI* asymptotically. If *RI* is less than the threshold value ϑ, the neuron will never spike. If $RI>\vartheta$, then the neuron fires periodically (Figure 2.4).

In the case of time-dependent stimulus *I(t)*, the solution for equation (2.8) is:

$$u(t) = u_r e^{-\frac{t-t_0}{\tau_m}} + \frac{1}{C}\int_0^{t-t_0} e^{-\frac{s}{\tau_m}} \cdot I(t-s)ds \qquad (2.10)$$

where t_0 is the time of the occurrence of the spike. This expression describes the membrane potential for $t>t_0$ and is valid until the occurrence of the next spike. If membrane potential reaches the threshold $u(t)=\vartheta$, membrane potential is reset to u_r and integration restarts (see Figure 2.5).

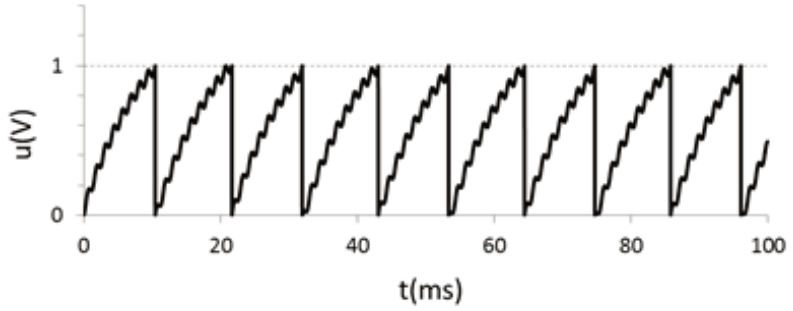

Figure 2.5. Periodic spikes of leaky integrate-and-fire model in response to a time dependent input current $I=1.5 + 2sin(5t)$. Resistance R was set to 1, $\tau_m=10$, threshold $\vartheta=1$ and reset potential $u_r=0$.

As mentioned at the beginning of this section, the leaky integrate-and-fire neuron can be stimulated by modeling synaptic inputs from presynaptic neurons. In other words, we can organize leaky integrate-and-fire neurons into a neural network. In this case input current *I(t)* is generated by the activity of presynaptic neurons. In the network each presynaptic spike generates a postsynaptic current pulse, and *I(t)* is the sum of all current pulses. Thus:

$$I_i(t) = \sum_j w_{ij} \sum_f \alpha\left(t - t_j^{(f)}\right) \qquad (2.11)$$

where w_{ij} is the strength of the synapse from presynaptic neuron *j* to postsynaptic neuron *i*. The usual choice (Gerstner & Kistler, 2002) for α would be a Dirac delta function:

$$\alpha(t) = q\delta(t) \tag{2.12}$$

where *q* is the total charge injected into the postsynaptic neuron when synaptic strength is $w_{ij} = w_{max}$. A more realistic approach is where postsynaptic current α has a duration with exponential decay:

$$\alpha(t) = \frac{q}{\tau_s} \cdot e^{-\frac{t}{\tau_s}} \cdot \theta(t) \tag{2.13}$$

where Θ is the Heaviside step function with $\Theta(t)=1$ when $t>0$ and otherwise $\Theta(t)=0$; time constant τ_s defines the slope of exponential decay. Even more realistic functions of α include an exponential rise time τ_r and transmission delay Δ^{ax}:

$$\alpha(t) = \frac{q}{\tau_s - \tau_r}\left(e^{-\frac{t-\Delta^{ax}}{\tau_s}} - e^{-\frac{t-\Delta^{ax}}{\tau_r}}\right)\theta(t). \tag{2.14}$$

In the literature exponential functions such as (2.13) and (2.14) are often called α-functions (Gerstner & Kistler, 2002).

It is important to note that I used function (2.14) with $\Delta^{ax} = 0$ and $q = \tau_s - \tau_r$ as a standard function in my work.

In this section I have described just the basic model of the leaky integrate-and-fire neuron; there are multiple extensions (Abbott & Vreeswijk, 1993; Ermentrout, 1996; Feng, 2001; Hansel & Mato, 2001; Latham et al., 2000).

2.1.3. Spike Response Model

The Spike Response Model (SRM) is a generalization of the leaky integrate-and-fire neuron. As with integrate-and-fire neurons, SRM action potentials are fired when the membrane potential surpasses a threshold value. In contrast to the

leaky integrate-and-fire model, the SRM model includes a phase of relative refraction. Another difference is that integrate-and-fire models are formulated using differential equations while the Spike Response Model is formulated using filters.

In the SRM model the state of the neuron i is described by the single variable u_i. The neuron can be stimulated by input spikes or an external driving current I. In the absence of input stimulation, the neuron is at its rest potential u_i=0. Each incoming spike or pulse of injected current I will perturb u_i and it will take time until u_i returns to rest potential 0. The course of u_i is given by:

$$\begin{aligned} u_i(t) = \eta(t - t_0) \\ + \sum_j w_{ij} \sum_f \epsilon_{ij}\left(t - t_0, t - t_j^{(f)}\right) \\ + \int_0^\infty \kappa(t - t_0, s) I(t - s) ds \end{aligned} \tag{2.15}$$

- η – response kernel function which describes the action potential and afterhyperpolarization;
- w_{ij} - synaptic strength;
- t_0 – time of the last spike of postsynaptic neuron i;
- $t_j^{(f)}$ – times of spikes of presynaptic neurons j;
- ϵ – response kernel function which describes the response over time to an incoming spike;
- κ - response kernel function which describes voltage response to injected current, also called the linear filter.

If the value u_i reaches threshold ϑ a postsynaptic potential is triggered and t_0 is reset to the current time. In contrast to the leaky integrate-and-fire neuron discussed in the previous section (see section 2.1.2 “The Leaky Integrate-and-Fire Model”), the threshold value ϑ is not necessarily fixed and may also depend on t-t_0.

The external driving current I and the κ kernel function are typically used to fit the SRM model to the empirical data when a living neuron is being injected with a known current. The Spike Response Model can reproduce the behavior of the biological neuron with remarkable accuracy (Jolivet et al., 2006). When implementing a neural network by stimulating a neuron with input spikes from presynaptic neurons, however, the external driving current can be ignored. For the ϵ kernel we can use the alpha functions discussed in the previous section (see section 2.1.2 "The Leaky Integrate-and-Fire Model"). The kernel η describes the action potential and afterhyperpolarization. Alternatively, afterhyperpolarization can be mimicked using the κ kernel (Gerstner & Kistler, 2002).

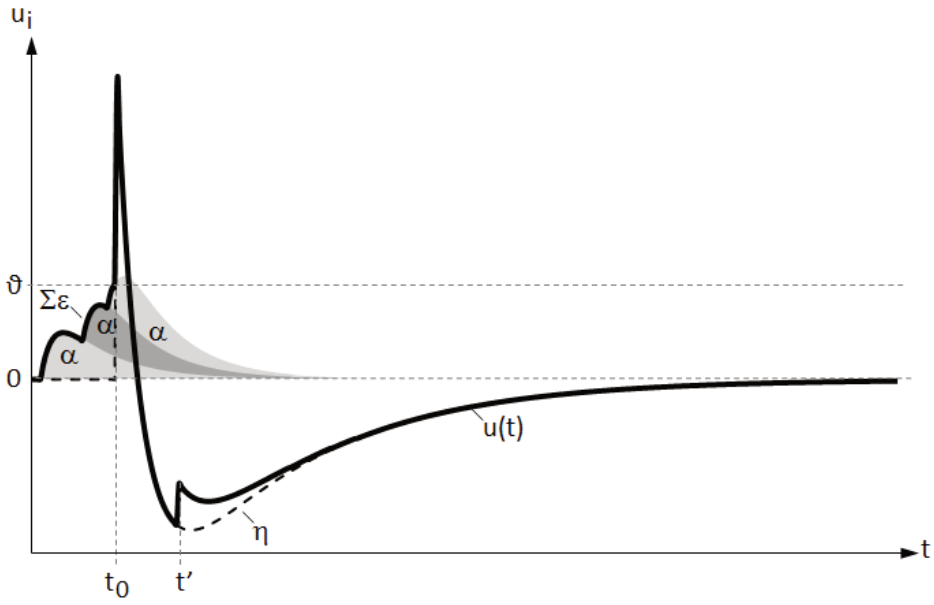

Figure 2.6. Example of the Spike Response Model.

Figure 2.6 is a schematic example of the Spike Response Model. In this diagram the external drive current is omitted; the ϵ kernel is the alpha function $\epsilon=exp(-t/\tau_s)-exp(-t/\tau_r)$ and the η kernel is $\eta=2exp(-t/\tau_\eta)-exp(-t/\tau_h)$ where $\tau_s>\tau_r$ and $\tau_h>\tau_\eta$. When the sum of ϵ reaches the threshold ϑ at moment in time t_0, the η function is injected into $u(t)$. From spike time t_0 to t' there is a period of absolute refraction. In this example the integration of ϵ was stopped right at t_0

and resumed at t', when absolute refraction ended. In general this is not necessary: instead of suspending the integration of ϵ, threshold ϑ can be temporarily increased to an unachievable value, thus implementing absolute refraction. In this case the additional value added to ϑ can be a modeled by a decaying function. The injection of the η function moves the value of membrane potential $u(t)$ below resting potential 0, and it takes time for the membrane potential to return to zero, so that injection of η mimics afterhyperpolarization.

It must be emphasized that the leaky integrate-and-fire model is a special case of the Spike Response Model where the kernel $\eta(t-t_0, t-t_j^{(f)}) = u_r exp(-(t-t_0)/t_m)$ and the kernel $\kappa(t-t_0, s) = exp(-s/t_m)$ (see Eq. (2.10), in the section 2.1.2 "The Leaky Integrate-and-Fire Model").

2.1.4. Compartmental Models

Neural models described in previous sections do not take into account the structure of a biological neuron. By structure I mean the spatial and morphological properties of soma, dendrites, axon and axon terminal. Models that take biological structure into account are called multi-compartment models. I did not use multi-compartment models in my work directly, but I did use some aspects of the spatial properties of dendrites. Thus I will not describe multi-compartment models in detail, but will instead just provide a simple explanation of the concept.

The multi-compartment model allows the numerical solving of the membrane potential of a complex dendritic tree. The dendritic tree is modeled by a network of small cylindrical compartments (Figure 2.7). Each compartment modeled has membrane capacity C^{μ} and transversal resistance R^{μ}. Connections between compartments are characterized by longitudinal resistance R_L allowing for the modeling of geometrical properties of the compartments. In Figure 2.7 I^{μ} in the middle compartment denotes external current. In the leftmost compartment in Figure 2.7 there is a variable resistance component which is used to model non-

linear ion channels. The modeling of non-linear ion channels may be added to all compartments.

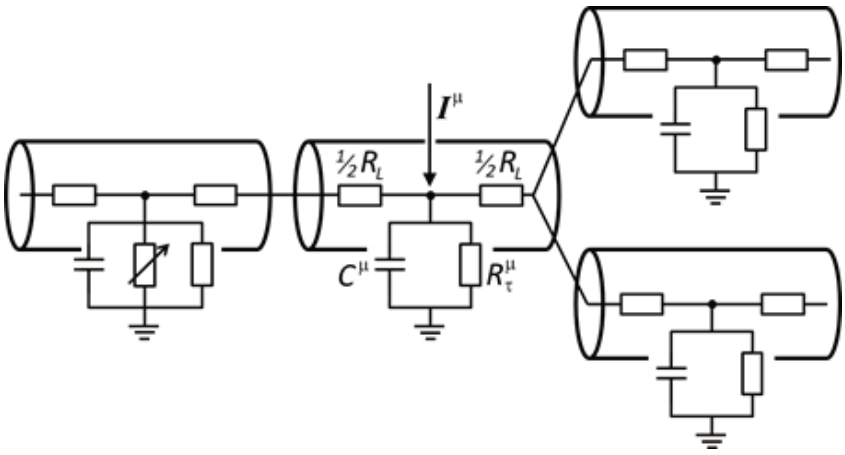

Figure 2.7. Multi-compartment neuron model. Adapted from Gerstner and Kistler (2002).

It is important to note that formal neural models such as SRM can also be extended into multi-compartment models (Gerstner & Kistler, 2002).

2.1.5. Other Models of the Spiking Neuron

So far I have introduced models of spiking neurons which are relevant to my work. There are a number of other models which have been developed but I will not discuss them in detail. Noteworthy are FitzHugh-Nagumo (FHN), which is a simplified version of the Hodgkin-Huxley model (FitzHugh, 1961; Nagumo et al., 1962); Morris–Lecar which is a combination of Hodgkin-Huxley and FitzHugh-Nagumo (Morris & Lecar, 1981) and Hindmarsh-Rose, which is an extension of FitzHugh-Nagumo (Hindmarsh & Rose, 1984).

2.2. Phenomenological Models of STDP

Spike-timing dependent plasticity was introduced earlier in the discussion of the biological mechanics of the neuron (see section 1.4 “Spike-Timing Dependent Plasticity”). In this section I will discuss phenomenological models of STDP and some important features of STDP associated with the different models.

It must be said that besides phenomenological models of STDP, there are biophysical models which attempt to model more precise mechanisms of STDP,

such as internal states of the NMDA receptor (Senn et al., 1997, 2001), intracellular calcium concentration (Shouval et al., 2002), and etc. Since biophysical models of STDP are not the subject of my research, I will not describe them in detail.

2.2.1. Basic STDP Implementation

Change of strength of the synapse Δw_{ji} from the presynaptic neuron j to the postsynaptic neuron i depends on the difference in time between presynaptic and postsynaptic spikes. Let's define spike arrivals from the presynaptic neuron as t_j^f, where f=1,2,3,… labels the firing times of presynaptic spikes, and let's define the spike times of the postsynaptic neuron as t_i^n, where n=1,2,3,… labels the firing times of the postsynaptic spikes. Then the total change of synaptic strength Δw_{ji} induced by pairs of presynaptic and postsynaptic spikes is (Gerstner et al., 1996, Kempter et al., 1999):

$$\Delta w_{ji} = \sum_{f=1}^{N} \sum_{n=1}^{N} W(t_i^n - t_j^f) \tag{2.16}$$

where $W(t)$ is the STDP window. In the case of an STDP window of excitatory-excitatory synapses, typically $W(t)$ is set to (Zhang et al., 1998; Song et al., 2000):

$$W(t) = \begin{cases} A_+ \cdot e^{-\frac{t}{\tau_+}} & if\ t > 0 \\ -A_- \cdot e^{\frac{t}{\tau_-}} & if\ t < 0 \end{cases} \tag{2.17}$$

where variables A_+ or A_- may or may not depend on the current value of synaptic strength (see section 2.2.3 "Multiplicative Update vs. Additive Update").

2.2.2. Online Implementation of STDP

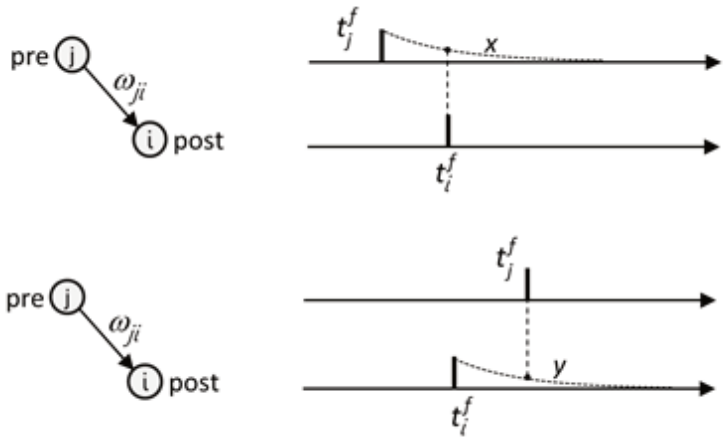

Figure 2.8. STDP implementation with local variables.

An online version of STDP update can be implemented easily by introducing two local variables (see Figure 2.8): one for a low-pass filtered presynaptic spike train, the other for a low-pass filtered postsynaptic spike train. Let's consider the synapse between neuron *j* and neuron *i*. Suppose that each spike from presynaptic neuron *j* leaves trace *x*, and each spike from postsynaptic neuron *i* leaves trace *y* (Morrison et al., 2008). Then:

$$\frac{dx_j}{dt} = -\frac{x_j}{\tau_x} + \sum_{t_j^f} \delta(t - t_j^f) \tag{2.18}$$

$$\frac{dy_i}{dt} = -\frac{y_i}{\tau_y} + \sum_{t_i^f} \delta(t - t_i^f) \tag{2.19}$$

where t_j^f denotes the spike times of the presynaptic neuron, t_i^f denotes the spike times of the postsynaptic neuron and δ is the Dirac delta function. Variables *x* and *y* increase at the moment of presynaptic or postsynaptic spike correspondingly, then decrease exponentially. The amount of change in synaptic strength can be expressed:

$$\Delta w_{ji}^{+}(t_i^f) = A_{+}(w_{ji}) x_j(t_i^f) \tag{2.20}$$

$$\Delta w_{ji}^{-}(t_j^f) = -A_{-}(w_{ji})y_i(t_j^f) \tag{2.21}$$

or:

$$\frac{dw_{ij}}{dt} = A_{+}(w_{ji})x_j(t)\delta\left(t - t_i^f\right) - A_{-}(w_{ji})y_i(t)\delta(t - t_j^f) \tag{2.22}$$

The biophysical meaning of variables *x* and *y* is not well established yet. Hypothetically the variables *x* and *y* could be related to interactions between back-propagating action potential (BAP) and NMDA receptors (Caporale & Dan, 2008).

2.2.3. Multiplicative Update vs. Additive Update

Synaptic strength is usually restricted within the range $w_{min}<w_j<w_{max}$. This can be achieved by multiplicative update, where update is proportional to the current value of synaptic strength (Sjostrom & Gerstner, 2010). This method is also often referred to as soft boundaries:

$$\begin{aligned} A_{+}(w_j) &= (w_{max} - w_j)\eta_{+} \\ A_{-}(w_j) &= (w_j - w_{min})\eta_{-} \end{aligned} \tag{2.23}$$

where η_{+} and η_{-} are positive constants and are referred to as multiplicative weight dependence, or soft boundaries. The implementation of additive update, also referred to as hard boundaries, would be:

$$\begin{aligned} A_{+}(w_j) &= \theta(w_{max} - w_j)\eta_{+} \\ A_{-}(w_j) &= \theta(w_j - w_{min})\eta_{-} \end{aligned} \tag{2.24}$$

where Θ is the Heaviside step function. In this case synaptic strength update would occur with the fixed value η_{+} or η_{-} until the boundaries are reached (Gerstner et al., 1996, Kempter et al., 1999, Roberts et al., 1999, Song et al., 2000).

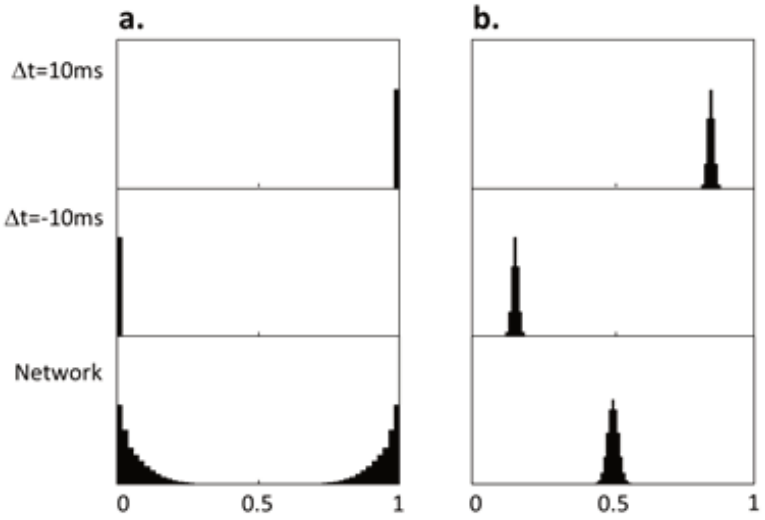

Figure 2.9. Equilibrium distributions of synaptic strengths. **a**: Additive STDP update. **b:** Multiplicative update. The two upper rows are the behaviors of a single synapse; the bottom row is the distribution of synaptic strength when neuron input was generated from 1,000 presynaptic synapses generating Poisson-distributed input spikes (the input rate was 10Hz). Schematically redrawn from Rubin et al. (2001).

While these methods may look similar, they result in very different behavior. In the case of input spikes generated from a Poisson distribution, for example, additive update tends to push synaptic weights towards the extremes and towards bimodal distribution (Figure 2.9a), while multiplicative update tends towards a unimodal distribution of synaptic strengths (Figure 2.9b) (Rubin et al., 2001). Gutig and colleagues (Gutig et al., 2003) merged the multiplicative and additive update rules:

$$A_+(w_j) = \left(w_{max} - w_j\right)^{\mu} \eta_+$$
$$A_-(w_j) = \left(w_j - w_{min}\right)^{\mu} \eta_- \qquad (2.25)$$

where μ=0 results in additive update; μ=1 results in multiplicative update and intermediate values for μ result in intermediate behavior. Gutig and colleagues demonstrated that unimodal distribution is the rule rather than the exception and bimodality is achieved only when very small values are put for μ (Figure 2.9c). Moreover, as the number of input synapses increase, the critical value of μ at which bimodality begins to emerge further decreases.

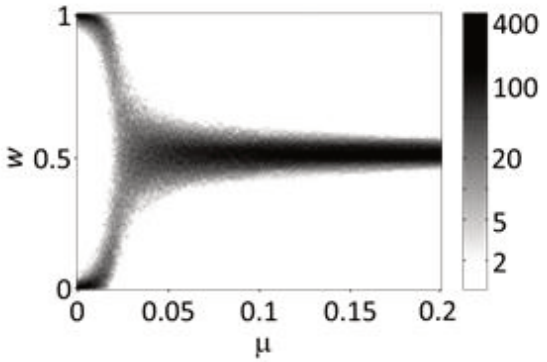

Figure 2.10. Equilibrium distributions of synaptic strengths of weight-dependence exponent μ. Synaptic strengths depicted in logarithmic scale. Results obtained by Gutig and colleagues (Gutig et al., 2003), the single integrate-and-fire neuron had inputs from 1,000 uncorrelated Poisson processes at 10 Hz. Adapted from Gutig et al. (2003).

2.2.4. All-to-All vs. Nearest-Neighbor Spike Interaction

When change of synaptic strength is caused by interaction between all presynaptic spikes and all postsynaptic spikes, such an implementation of spike-timing-dependent plasticity is called all-to-all interaction (Sjostrom & Gerstner, 2010). Alternately, interaction may be restricted to nearest neighbors. In other words, each presynaptic and postsynaptic spike resets the traces of variables x and y (see Eq. (2.22)). Such an implementation is referred to as a nearest-neighbor interaction (Figure 2.11).

All-to-all and nearest-neighbor interactions result in very different behavior in respect to training stability (Izhikevich & Desai, 2003). By making the assumption that presynaptic and postsynaptic spike trains are weakly correlated Poisson processes, Izhikevich and Desai derived a number of equations for permutations of multiplicative and additive updates and all-to-all and nearest-neighbor interactions. Izhikevich and Desai used more than one neighborhood implementation (see their original paper for details). Plots of the expected changes in synaptic strengths for additive all-to-all and additive nearest-neighbor are depicted in Figure 2.12.

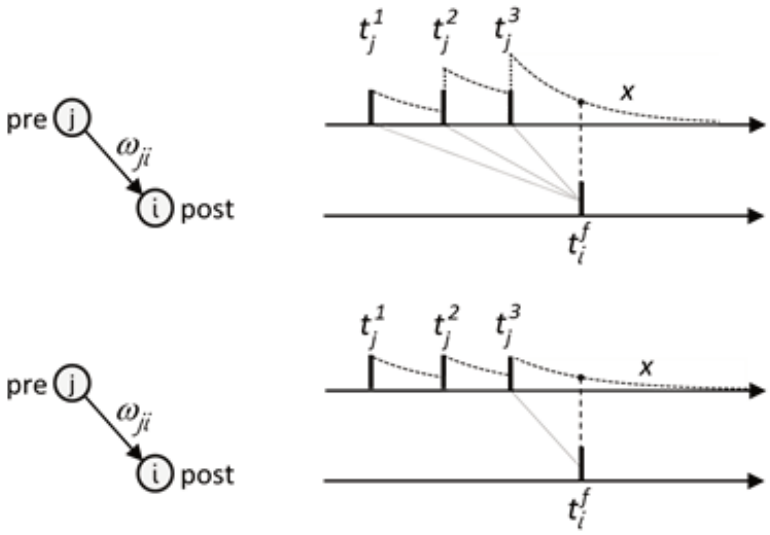

Figure 2.11. All-to-all vs. nearest-neighbor spike interaction. Only variable x is depicted in this diagram. At the top in the all-to-all interaction the trace of x is cumulative. At bottom is the nearest-neighbor interaction where only the most recent postsynaptic spike interacts with a postsynaptic spike. In this case, the trace of variable x is restarted after each presynaptic spike.

Izhikevich and Desai's solution for all-to-all interaction with additive STDP update yielded a linear dependency of the amount of synaptic change upon the postsynaptic firing rate; there was a point of stable equilibrium (Figure 2.12a, black marker). In the case of additive nearest-neighbor implementation, however, dependency was non-linear and there were two points of equilibria, stable (Figure 2.12b, black marker) and unstable (Figure 2.12b, white marker).

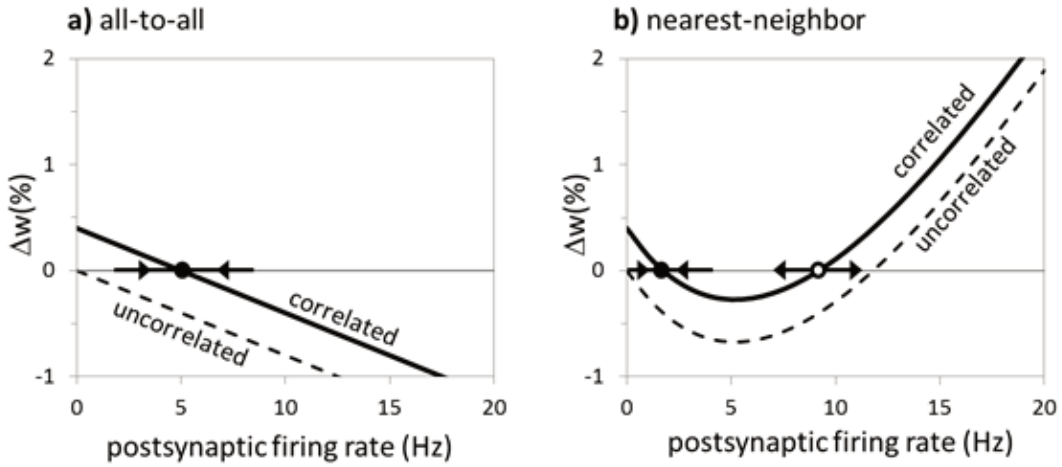

Figure 2.12. Stability diagram of all-to-all and nearest-neighbor spike interactions. Adapted from Izhikevich and Desai (2003).

It must be said that Izhikevich and Desai's solution for an additive nearest-neighbor implementation relates spike-timing-dependent plasticity to the Bienenstock-Cooper-Munro (BCM) theory (Bienenstock et al., 1982). BCM theory will be discussed in a separate section.

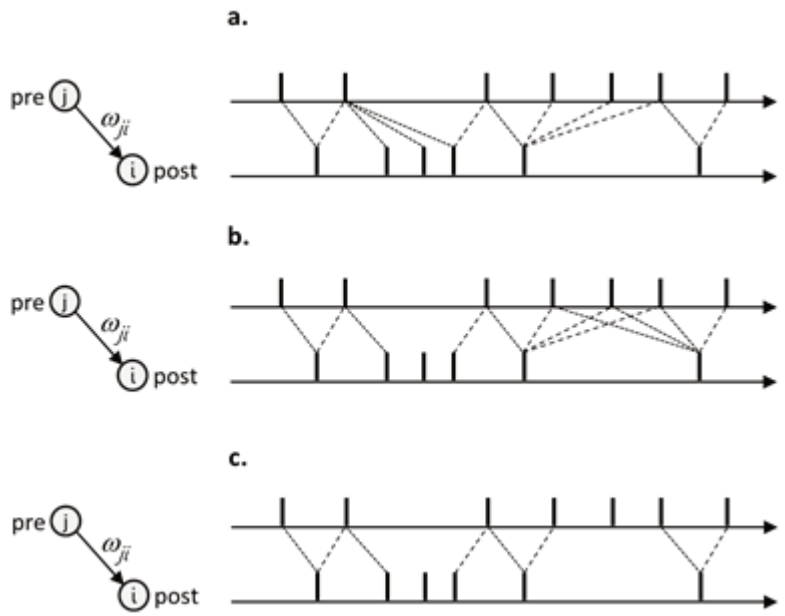

Figure 2.13. Different models of spike neighborhood. Adapted from Morrison et al. (2008) a: Symmetric interaction: each presynaptic spike interacts with the last postsynaptic spike and *vice versa* (Morrison et al. 2007). b: Presynaptic-centered interaction: each presynaptic spike interacts with the last postsynaptic spike and the next postsynaptic spike (Izhikevich & Desai, 2003; Burkitt et al., 2004). c: Reduced symmetric interpretation, only immediate spike pairings interact (Burkitt et al., 2004).

There is more than one model of nearest-neighbor interaction (see Figure 2.13).

2.2.5. The Triplet Rule

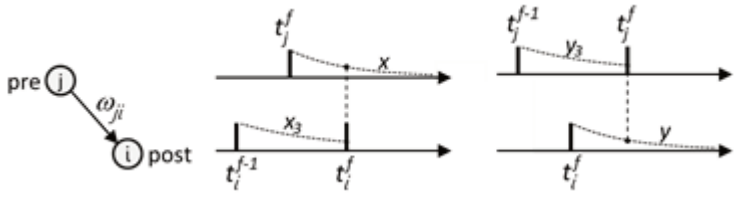

Figure 2.14. Triplet STDP update.

Triplet STDP update (Figure 2.14) was originally suggested by Froemke and Dan (Froemke & Dan, 2002), based on *in vivo* experiments with pyramidal

neurons in the visual cortex of the rat. Later Pfister and Gerstner (Pfister & Gerstner, 2006) successfully reproduced STDP behavior found in biological neurons, in the visual cortex (Sjostrom et al., 2001) and hippocampal culture (Wang et al., 2005). Such behavior could not be reproduced with two-spike interactions. The triplet rule introduces two additional trace variables: x_3 and y_3. Pfister and Gerstner suggested the function of the triplet STDP window, which in the case of simple exponentials would be:

$$\Delta w(t) = \begin{cases} x(t) \cdot \left(A_+ + A_{3+} \cdot x_3(t)\right) & if\ t > 0 \\ -y(t) \cdot (A_- + A_{3-} \cdot y_3(t)) & if\ t < 0 \end{cases} \tag{2.26}$$

where A_{3+} and A_{3-} denote the amplitude of the triplet term of potentiation or depression. These variables are similar to the variables A_+ and A_- discussed previously in this section. The variables x, x_3, y and y_3 can be expressed in the form of simple exponential functions, in the same way as in Eq. (2.17).

3. Results of Research

3.1. Neural Processing of Long-Lasting Sequences of Temporal Codes

I focused my research on STDP learning of spatial and spatiotemporal patterns. The simplest example of a spatiotemporal pattern is a binary on/off map of spikes in a short temporal window, where the probability of a spike at the "on" synapse is significantly larger than at the "off" synapses, and "on" spikes are largely correlated in time, while "off" spikes are not and produce only Poisson noise. In its simplest form, it is a spatial pattern. In the case of STDP learning, under a specific range of parameters the strengths of synapses associated with the pattern grow, while the strengths of other synapses which receive only noise decay. In other words, the individual neuron acts as a coincidence detector (Abbott & Nelson, 2000). In the simplest possible case this sort of training could be reduced to supervised learning as a simple assignment operation: if the input is already in the pattern, then set strength to 1, otherwise set to 0.

I began my work by reproducing the experiments of Masquelier and colleagues (Masquelier et al., 2008, 2009). In their original work (Masquelier et al., 2008) the authors used a single SRM neuron for the learning of spatiotemporal patterns injected into Poisson noise, even when the sample pattern was injected without a strict time reference. Masquelier and colleagues found that the neuron was capable of detecting the very beginning of the pattern. Later Masquelier and colleagues extended their experiment with multiple competing neurons (Masquelier et al., 2009), and demonstrated that it is possible to train multiple neurons for multiple patterns. Again, in such a network the neurons were capable of detecting the beginning of the spatiotemporal patterns (see Figure 3.1).

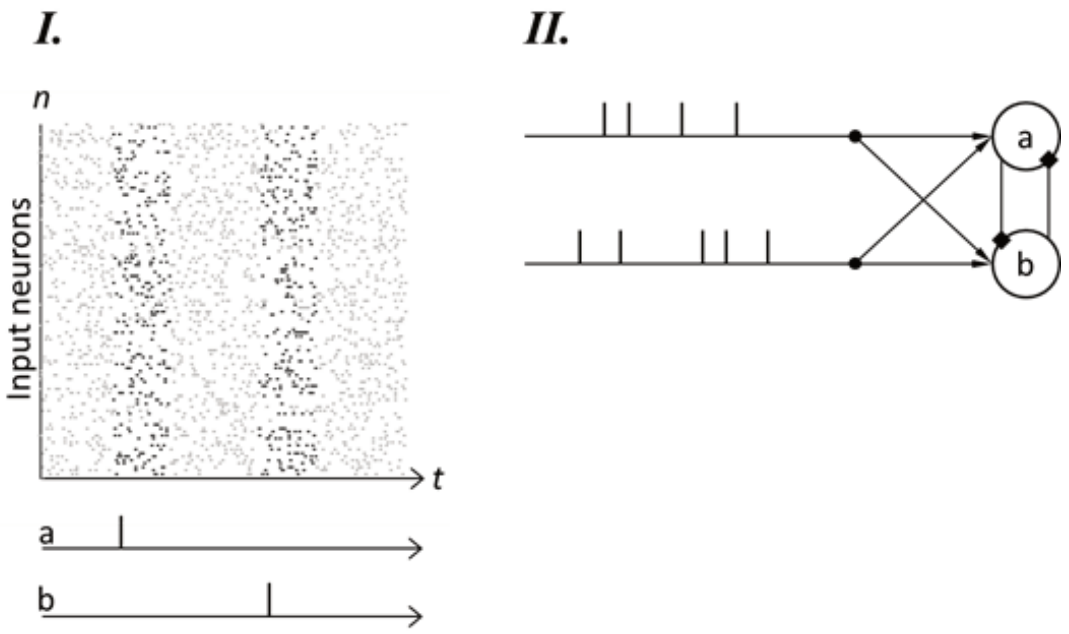

Figure 3.1. Experiment made by Masquelier and colleagues. Schematically redrawn from Masquelier et al., 2009. **I.** Sample pattern. Black dots denote constant pattern, grey dots are Poisson-distributed pattern. Masquelier used patterns generated from the same Poisson distribution. Occurrence of patterns was stochastic. At bottom are the responses from two trained neurons which were selective to two different patterns. **II.** Schematic representation of the Masquelier network. Arrows denote excitatory synapses, diamonds denote inhibitory synapses.

After examining the training process for the spatiotemporal pattern more closely, it is evident that the neuron is learning only for the spatial component, essentially the first in time. To illustrate this behavior I made a simple experiment. I used a pattern composed of three spatial components A, B and C, which were always aligned in time in the same order (Figure 3.2 I.). Most of the synapses did not produce a consistent pattern at all and are denoted by the letter D. The C component is larger than A and B, so that it would be the one preferred by the neuron at the beginning of the training. Initial synaptic weights were distributed uniformly and at the beginning of the training the neuron was not selective to any pattern. Initially, the weights of synapses associated to the largest component C grew most rapidly, while synapses associated to D decayed (Figure 3.2 II.). When the neuron became selective to C, STDP subsequently increased the strengths of synapses associated to the preceding components, B and A. The closer in time the component was, the more LTP the synapse gained. When the

weights of synapses associated to the B pattern were large enough to move the timing of the postsynaptic spike prior to the occurrence in component C, synapses associated to C experienced long-time-depression, and weights decayed. The same thing happened with component A and would have continued had there been more components. In the end, only synapses associated to the very first spatial components were large enough to produce a postsynaptic spike, and all other memory was lost.

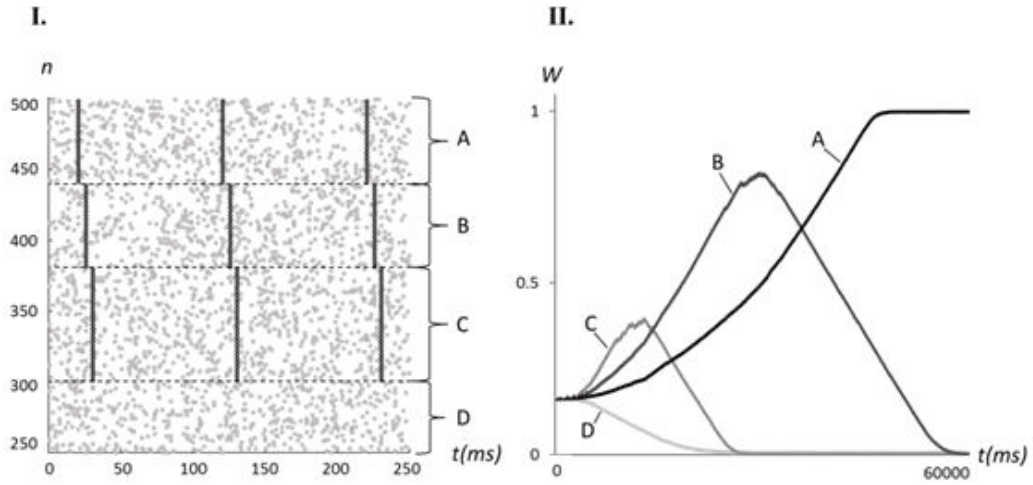

Figure 3.2. Illustration of detection of the onset of a pattern. Here a single neuron was trained for a pattern produced by 500 afferents. **I.** Sample pattern. Here in the vertical axis are individual afferents. Grey dots denote spikes produces by the Poisson process and black dots denote the injected pattern. There were four groups of afferents; three of them, A, B and C, produced different fractions of the spatiotemporal pattern, while group D produced only Poisson noise. **II.** Dynamics of synaptic weights. In the vertical axis are the mean synaptic weights of synapses associated to the specific group of afferents.

From the training results depicted in Figure 3.2 it is evident that after the training, the entire pattern, except for the very first spatial component, can be removed, shuffled or replaced with another pattern without any effect on the behavior of the network. The same applies to the experiments conducted by Masquelier and colleagues. Therefore, such a training indicates that in actuality one cannot encode data in a spatiotemporal pattern. Hence I formulated the problem for myself: would it be possible to build a network which could learn actual spatiotemporal patterns? If the spatial component is defined as a letter, could we build a network which could learn words composed of individual letters?

I designed the network based on STDP learning exclusively, capable of learning spatiotemporal patterns, but also capable of reproducing a sequence of observed spatial components (letters). I took an engineering approach here, and did not intend to design the network to replicate any known neural circuitry in living neural systems, but rather to determine merely if it were possible, and what the minimal set of components required to build such network might be.

This work was presented at the 2011 NCTA conference and was published in the conference proceedings and in the SCITEPRESS digital library (Krunglevicius, 2011).

3.1.1. The Network Model

Figure 3.3 is a conceptual representation of the proposed network model presented in this section. In the network there are two winner(s)-takes-all (WTA) layers. The first one, WTA1, is responsible for recognizing spatiotemporal patterns. Responses from WTA1 are memorized later in a temporal memory by assigning the individual spike from WTA1 to the appropriate time slot. Latter this memory is read, thus generating a new spatial pattern in which the individual spike corresponds to the spike from the WTA1 neuron and to the time slot. The secondary pattern is an input for the WTA2 layer. In this diagram, serial events A, then B and then C were converted to a single pattern which encodes events in a parallel manner, that is, in a spatial code.

The network model diagram is shown in Figure 3.4. It consists of six main layers: L1 and L5 are competitive WTA layers (in this particular network implementation I did not prohibit several neurons from learning the same pattern, and therefore we can speak of winners). L1 and L5 have corresponding inputs from L0 and L4. L3 is a layer of temporal memory; it is modulated by layer L2. In this specific network model I did not attempt to emulate any layers in the cortex or hippocampus, and the network structure and layer names are purely arbitrary.

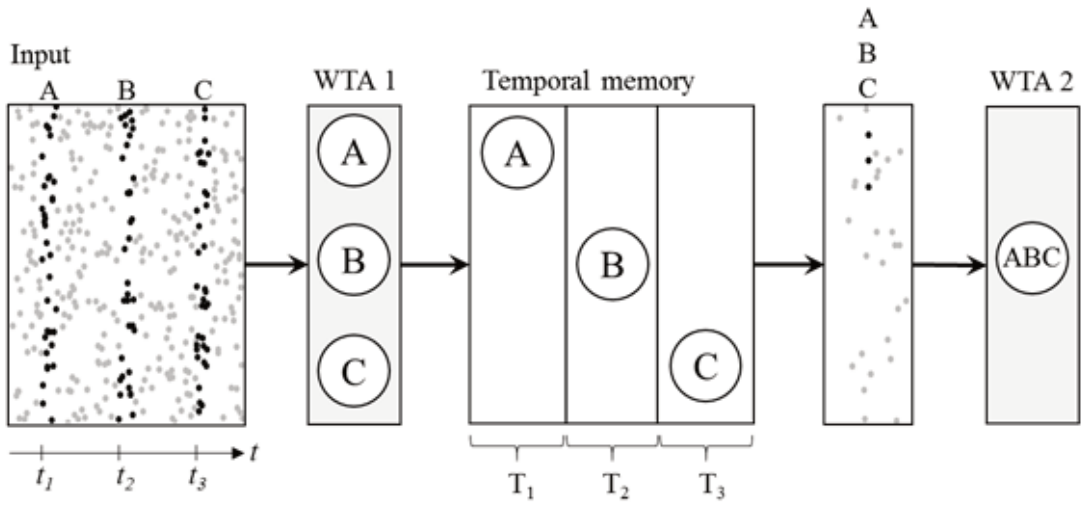

Figure 3.3. Conceptual diagram of the network model. Here black dots denote periodic patterns which are learned by WTA neurons and grey dots denote background noise. Noise is necessary for reducing synaptic weights in synapses which don't belong to the learned pattern.

Neurons in layer L0 periodically fire a sample pattern. L0 neurons also fire spontaneously with the probability P_{L0} in each iteration of the experiment (every millisecond). Spontaneous firing produces Poisson noise. Noise increases the probability of LTD in synapses L0 to L1 and is responsible for strength-decay of synapses which do not participate in the sample pattern. Although I used noisy input in my experiments, it is not mandatory, for neurons can be trained successfully without it, although noiseless patterns wouldn't be biologically realistic. In that case, synapses which do not carry spikes from the sample would not be affected by STDP.

Neurons in layer L1 receive input from L0 and are interconnected with inhibitory synapses. The strengths of inhibitory L1 to L1 synapses are constant.

The L1 layer produces input for L1.1 interneurons via strong synapses with fixed weights. The strengths of L1 to L1.1 synapses are large enough to give rise to a postsynaptic spike from resting potential with a single presynaptic spike. Layer L1.1 was introduced so that later memory readings would not affect the L0 to L1 synapses.

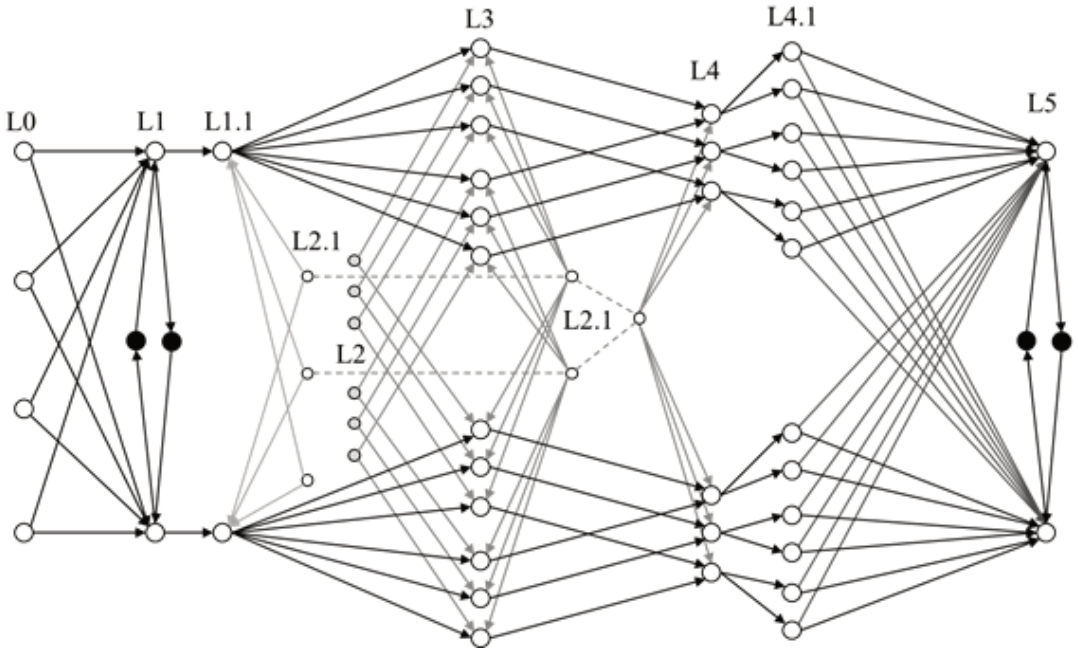

Figure 3.4. Diagram of a network model with temporal memory. The color black denotes inhibitory interneurons. In the actual simulation inhibitory neurons were replaced by direct inhibitory synapses. Grey lines denote synapses from the L2 and L2.1 subnetwork of temporal modulation. Dotted lines denote the same neuron, split in the diagram for ease of visualization. Layer L4.1 was added only for programming convenience and in one particular experiment it served as the input multiplier for the L5 WTA network.

Layer L2, including the neurons in L2.1, is used for temporal modulation. Excitation of L2 and L2.1 neurons imitates wave propagation in excitable media in a single direction, and only one neuron fires at once; it is looped. While L2 neurons produce a chain of spikes during the excitation period, L2.1 produces only a single spike. Weights of synapses outgoing from L2 and L2.1 do not change. See Figure 3.5 for details.

Each synapse from the L1.1 neurons to the L3 neurons represents a binary memory unit. It memorizes the spike event from L1.1 relative to the timing of the corresponding L2 neuron. L3 neurons are grouped by synapse from L1.1. Each L3 neuron in the group receives a strong excitatory input from a different L2 neuron. This input is not strong enough, however, to produce a spike. Initially L1.1 to L3 synapses are weak and are prohibited from growing strong enough to fire a spike without additional excitation from L2 or L2.1. If an L3 neuron is

excited by spikes from L2 and during that period L1.1 fires, it, too, would fire, and synapse strength would grow by strong LTP.

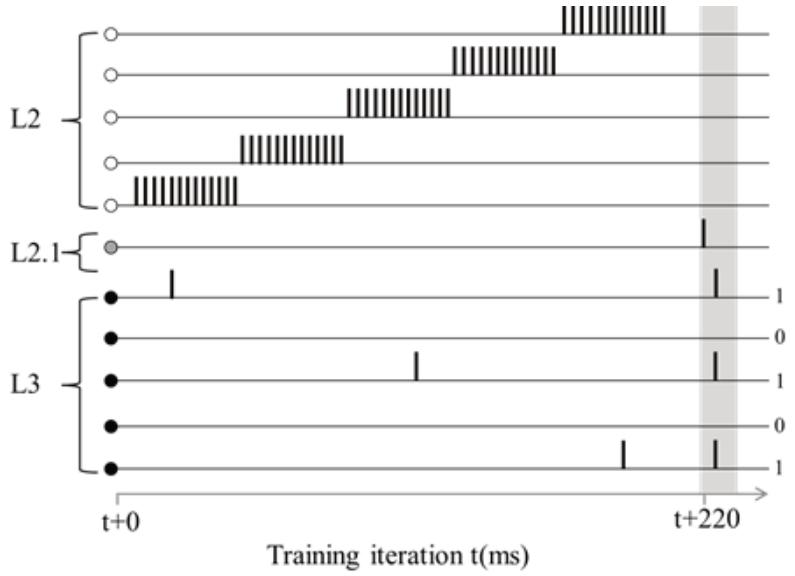

Figure 3.5. Temporal modulation of five neurons in layer L3 which receive input from a single neuron from layer L1.1. Layer L2 excites each neuron in L3 for approximately 40 ms. If within that window the L3 neuron receives EPSP from L1.1, it produces a spike and the corresponding synapse is updated by strong LTP. After 220 ms, the L2.1 neuron causes an additional spike in L1.1 and adds weak EPSP to the neuron groups in L3 and all L4 neurons. L3 retains the memory of the previous spike and passes on a compressed pattern to L4. See the network diagram in Figure 3.4. In one particular case the L1.1 neuron fired three times; as a result L3 produced the spatial pattern 10101.

During the experiment, the strengths of synapses L1.1 to L3 did decay over time, so that the memory slot could be reused during the next L2/ L2.1 loop iteration. It is known that LTP in living synapses lasts from a few hours to months or longer (Abraham, 2003), so synaptic strength decay in this particular model is consistent with the biological features of synapses.

L2.1 neurons activate memory readings. Each L2.1 has strong synapses to all L.1.1 neurons, weak synapses to all L4 neurons and weak synapses to subgroups in L3. L3 neurons grouped by L2.1 represent the memory window. A spike from L2.1 causes a spontaneous spike in L1.1. Excitation from L2.1 to L3 is much weaker than from L2 and is produced by a single spike, and therefore only a strong synapse from L1.1 to L3 can cause a spike in L3.

The L4 layer serves as an input to WTA layer L5. L4 has moderate fixed strength synapses from L3; therefore a spike from L3 can cause a spike in L4 only when an L4 neuron is excited by L2.1.

I added layer L4.1 only for convenience of programming. I found that multiplying inputs to the WTA layer would make the training process more robust over a wider range of parameters. It also increases the chance of a beneficial permutation of initial synaptic strengths. Since I experimented with a relatively small network, I duplicated inputs to L5 to obtain a more stable training process. With a larger network this would not be necessary. Alternately the L4.1 layer can be replaced by multiplying the synapses from L4 to L5 instead of adding an entire layer of interneurons. Analogous to layer L0, L4.1 produces Poisson noise.

Layer L5 is analogical to L1, but I tuned it with different STDP parameters. Additionally, I introduced a stochastic threshold in the L5 neurons, see section 3.1.5 “Learning Conditions in Layer L5”.

Layer L1 was trained throughout the entire simulation, while the training of layer L5 started only after the first 100,000 iterations of the simulation. I simply prohibited neurons in layer L4 from firing during the first stage of the experiment.

3.1.2. Materials and Methods

In this section I provide the model of the neuron and plasticity used in this work. The model is a version of SRM (see section “2.1.3 Spike Response Model”). I chose SRM because of its simplicity and high degree of accuracy (Jolivet et al., 2006). In this work this was a discrete version of SRM similar to the one used by Masquelier (Masquelier et al., 2008; 2009), although I have made some modifications to reduce computational costs. The η–kernel function, which defines the action potential and afterhyperpolarization, was:

$$\eta(t) = W_{ap}\left(K_{dpl}\, e^{\frac{-\Delta t_h}{\mathrm{T}_m}} - K_{hpl}\left(e^{\frac{-\Delta t_h}{\mathrm{T}_m}} - e^{\frac{-\Delta t_h}{\mathrm{T}_{ap}}}\right)\right) \tag{3.1}$$

where $\Delta t_h = t - t_{spike}$, t_{spike} is the time of the last postsynaptic spike; constants K_{dpl} = 3, K_{hpl} = 5 and W_{ap}=40 define the amplitude of the function of action potential; T_m=10 ms is the membrane time constant that defines the slope of the hyperpolarization phase, and T_{ap}=0.5 ms is the constant that defines the slope of the spike.

I executed my experiments at a precision of one millisecond relative to the function of action potential; therefore I refer to a single iteration of the simulation as one millisecond.

The value of postsynaptic potential $\epsilon(t)$ arriving from individual synapse j is given by:

$$\epsilon_j(t) = \phi_j w_j\left(e^{\frac{-\Delta t}{T_m}}\left(1 + x_{m_j}(t)\right) - e^{\frac{-\Delta t}{T_s}}\left(1 + x_{s_j}(t)\right)\right) \tag{3.2}$$

where $\Delta t = t - t_{pre}$, t_{pre} is the time of the last presynaptic spike; w_j is the strength of the synapse; ϕ = 1 for excitatory synapses and ϕ = -1 for inhibitory. T_m = 10 ms and T_s =2.5 ms are the time constants. Variables x_m and x_s simplify the integration of exponentials during the simulation and are given by:

$$x_{m_j}(t) = \begin{cases} \frac{w_{j(t-1)}}{w_{j(t)}} e^{\frac{-\Delta\tau}{T_m}}\left(1 + x_{m_j}(t-1)\right) & \text{if } t = t_{pre} \\ x_{m_j}(t-1) & \text{if } t \neq t_{pre} \end{cases} \tag{3.3}$$

$$x_{s_j}(t) = \begin{cases} \frac{w_{j(t-1)}}{w_{j(t)}} e^{\frac{-\Delta\tau}{T_s}}\left(1 + x_{s_j}(t-1)\right) & \text{if } t = t_{pre} \\ x_{s_j}(t-1) & \text{if } t \neq t_{pre} \end{cases} \tag{3.4}$$

where $\Delta\tau$ is the time difference between the previous and last presynaptic spike. Initial values for x_m and x_s are zero. Equations (3.3) and (3.4) were derived in the

following way: the summed values of individual PSPs of a single synapse at the moment *t* can be expressed as the finite series:

$$\epsilon_j(t) = w_0 e^{\frac{-(t-t_0)}{T_m}} - w_0 e^{\frac{-(t-t_0)}{T_s}} + \cdots + w_n e^{\frac{-(t-t_n)}{T_m}} - w_n e^{\frac{-(t-t_n)}{T_s}} \qquad (3.5)$$

where w_n is the set of strengths at the moment of each spike and t_n is the set of times of spikes. The expression is valid assuming that all $t_n < t$. Treating the positive and negative parts of the series separately, the first two members of the series could be expressed as the equation:

$$(1 + x_0) w_0 e^{\frac{-(t-t_0)}{T_m}} + w_1 e^{\frac{-(t-t_1)}{T_m}} = (1 + x_1) w_1 e^{\frac{-(t-t_1)}{T_m}} \qquad (3.6)$$

where $x_0 = 0$ at the beginning of the simulation. Algebraically solving equation (3.6) gives equations (3.3) and (3.4). Since discrete-time simulation exponentials can be pre-calculated, and *x* can be computed at the time of the spike, computational costs can be decreased.

The neuron membrane potential at any given time is:

$$u(t) = \begin{cases} \eta(t) & if\ t = t_{spike} \\ \eta(t) + \sum \epsilon(t) & otherwice \end{cases} \qquad (3.7)$$

The spike occurs when membrane potential reaches threshold value ϑ.

The STDP function used in my particular neural circuit is expressed in equation (3.8). The synaptic strength change for excitatory synapses where $\Delta t = t_{post} - t_{pre}$ is:

$$\Delta w_j = \begin{cases} A_{LTP} \cdot e^{\frac{\Delta t}{T_{LTP}}} & if\ \Delta t < 0 \\ -A_{LTD} \cdot e^{\frac{-\Delta t}{T_{LTD}}} & if\ \Delta t > 0 \\ 0 & if\ \Delta t = 0 \end{cases} \qquad (3.8)$$

Synaptic strength values are limited to between w_{min} and w_{max}, which in this particular circuit vary depending on synapse type. To simplify the calculations

of postsynaptic potentials, I prohibited synapses from decaying below $1*10^{-6}$. See equations (3.3) and (3.4). See section "3.1.3 Parameters of the Simulation" for the A_{LTP}, A_{LTD}, T_{LTP} and T_{LTD} constants.

In this neural network I used the closest-neighbor rule, where only the two closest spikes participate in the modification of synaptic strength. Alternatively, the all-to-all rule could be used.

3.1.3. Parameters of the Simulation

I used a basic genetic algorithm to tune L1 and L5 WTA sub-networks, while the rest of the parameters were arbitrary. Initial conditions for genetic optimization were based on experimental observations (Bi & Poo, 2001) and were identical to the ones used in the work of Masquelier (Masquelier et al., 2008). In such an initial setup, the L1 layer could learn the pattern already, and the genetic algorithm only improved the learning success rate for L1, whereas in training of L5, where input conditions were very different, training was never successful using STDP parameters based on the Bi and Poo research. Eventually genetic optimization found the parameters suitable for such conditions, but the values obtained were very different from the initial setup.

The general parameters of the model are listed in Table 3.1, Table 3.2 and Table 3.3. For the parameters of the training sample data and for the special case of layer L5 threshold, see sections "3.1.4 Training Samples" and "3.1.5 Learning Conditions in Layer L5".

For evolutionary tuning I used a multi-agent system with a population of 100 agents. Each agent represented in itself a functional WTA network. The genome of each agent contained initial and maximal synaptic strengths w_0 and w_{max} and parameters for the STDP function A_{LTP}, A_{LTD}, T_{LTP} and T_{LTD}. Initial genome values for each agent were normally distributed around arbitrary mean values. In each generation, each agent was trained 20 times with different training sample sets. Synaptic strengths were reset at the beginning of each training. Errors made by each agent were counted for all 20 trainings. When the training

was completed, the worst performing agents (60% of the population) were replaced by the new mutants made from the best performing agents. Each gene mutated with a probability of 0.3; the new value was random in the range of +/- 5% from the inherited value. I did not use any crossover. Multiple experiments with different initial values were executed for a several hundred generations each. Genome values of the best performing agent from the final generation were used as parameters for the model.

Table 3.1. STDP Parameters for synapse types

Synapse type		**Parameter**				
From	**To**	w_{max}	A_{LTP}	A_{LTD}	T_{LTP}	T_{LTD}
L0	L1	0.56	0.064	0.037	9.01	55.71
L1.1	L3	21	30	0.03	24	34
L4.1	L5	0.75	0.32	0.076	8.37	459

Table 3.2. Initial synaptic strengths

Synapse type		w_0	**Synapse type**		w_0
From	**To**		**From**	**To**	
L0	L1	0.44*	L2.1	L3	8
L1	L1	1.411	L2.1	L4	18
L1	L1.1	30	L3	L4	10.9
L1.1	L3	15	L4.1	L4	28
L2	L3	2.5	L4.1	L5	0.68*
L2.1	L1.1	40	L5	L5	5

** Initial synaptic strengths randomly distributed around mean value W_0 in the range of +/- 2.5% of W_0.*

Table 3.3. Sizes of layers and ϑ threshold values

Layer	**Number of neurons**	**Neuron threshold ϑ**
L0	250	-
L1	20	6.89
L1.1	20	11.71
L2	125	-
L2.1	25	-
L3	250	11.71
L4	100	11.71
L4.1	200	11.71
L5	20	10.976

3.1.4. Training Samples

Sample spike patterns produced from layer L0 represented themselves as five 4x250 matrices with values of 0 and 1. One indicated spike time relative to the sample start time and column position. Spikes were distributed uniformly across all sample matrices with the probability of occurrence p=0.04. For convenience, I called L0 samples “letters" and denominated them by the lower-case letters a, b, c, d and e. "Letters" were displayed at 40 ms intervals. During the gaps between letters and during letter display, L0 produced random spikes with the same probability of p=0.04. During the first 100,000 iterations letters were displayed in random order.

After 100,000 iterations, letters were combined into consistent "words," denominated by the capital letters A, B, C, D and E. Each “word” was made up of five non-repeating letters, that is, from random permutations of a, b, c and d. Words were displayed in random order and aligned to start right after the L2.1 scan time. During scan time L0 produced a random letter. Intervals between letters remained the same at 40 ms.

Additionally I injected Poisson noise into L0 and L4.1 outputs. Poisson noise was created by firing a random spike with a probability of p_{L0}=0.04 for the L0 layer and with a probability of p_{L41}=0.01 for the L4.1 layer. In my experiments spike density during the display of samples was greater than in the intervals between samples; it has been demonstrated earlier, however, that neurons can learn successfully when densities are equal (Masquelier et al. 2008, 2009).

3.1.5. Learning Conditions in Layer L5

Patterns of "words" produced by the L4 layer were quite different from strictly fixed samples of "letters." The pattern represented itself as only a single "column" of incoming spikes, although these were asynchronous. Spikes fluctuated in the 2-3 millisecond range (See Figure 3.6).

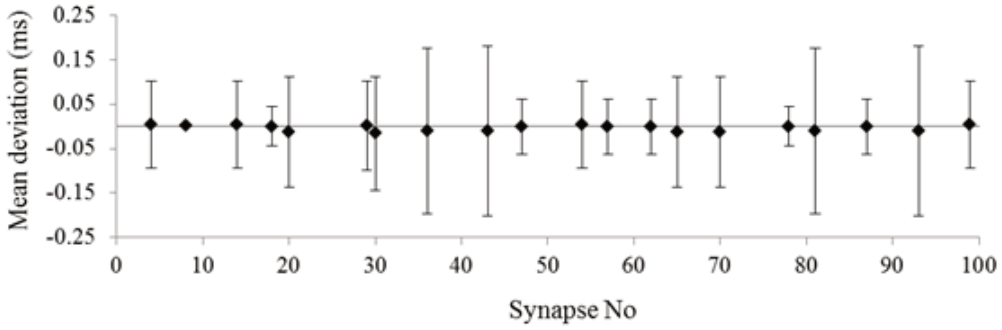

Figure 3.6. Mean deviation from pattern center in a single "word" in the L4 output (L4 to L4.1 synapses). Error bars denominate standard deviations. Data retrieved from a single experiment, the pattern was repeated 521 times, and only consistent spikes which repeated more than 80% of the time were included.

Fluctuations in patterns of "words" were caused by variations of synaptic strength in L1.1 to L3 synapses and also depended on the pre-existing value of postsynaptic potential in L4 and L3 neurons. I did not perform an analysis to determine which factor was dominant. Another important detail was that, due to the presence of errors in L1, not all spikes were equally consistent. Figure 3.5 shows L4 to L4.1 synapses which produced consistent spikes in the range of 83% to 100% for all occurrences of the "word." In the other synapses spikes occurred less than 3% of the time.

Initially I failed to achieve training of the L5 layer with an acceptable rate of error under these conditions. Usually all neurons learned a single pattern or a few at once. I solved this problem by introducing a stochastic threshold in L5 neurons: when the neuron reached its firing threshold, it didn't fire immediately but with a probability of 0.8. This accelerated inhibition by "lucky" competing neurons. It should be noted that attempts to apply a stochastic threshold in layer L1 only increased the error rate.

3.1.6. Results

I conducted a series of simulations of the entire model in a continuous mode. Also, because of the high computational cost of simulating the entire network, I conducted experiments with each of the WTA sub-networks separately in order to estimate performance. Each simulation had 700,000 iterations; the first

100,000 iterations were dedicated exclusively to training the L1 layer with random "letters." Typical output from the L5 layer at the beginning and end of training is shown in Figure 3.7.

Overall Performance of the Model

I counted responses of individual neurons relative to sample occurrence times during the last 5,000 iterations at the end of the experiment to estimate the rate of error. For the L1 layer I used a bias of 8 iterations of latency for neuron response, and a bias of 16 iterations for the L5 layer. The sample to which the neuron was the most selective was assigned to the neuron as a learned one. If the neuron response count was less than half of the average sample, that neuron was treated as non-selective to any sample. Every missed sample or neuron response outside of the biased sample window was treated as an error. I did not analyze cases where the neuron learned more than one sample, but instead treated responses to other samples as errors.

I conduced 100 experiments to estimate the mean error rate for the L1 and L5 layers separately. Initial synaptic strengths were reset for each experiment. Errors were counted across a sliding 3,000-iteration window for L1 and across 18,000 iterations for L5. Window sizes were proportional to the rate of samples 1 to 6: each word consisted of 5 letters plus 1 letter for scan time. The window was moved by a step of 1,000 iterations (See Figure 3.8). For the L1 layer I generated sample "letters" for each experiment, and for layer L5 I used recorded input from three different simulations of the entire model. Therefore my estimation of the L5 error rate has a larger bias.

The L5 layer produced a significantly greater rate of error, and in the last 10,000 iterations of the experiments it reached a mean value of 4.514, while L1 came to only 1.207.

There is an interesting observation to be made in the layer L1 error rate: at the moment when random "letters" are replaced by consistent sequences, we see a modest but steep drop in the error rate (Figure 3.8 (a)). Most likely this was

caused by a reduced rate of sequences of the same "letter," which makes it harder for a trained neuron to subsequently fire because of the previous hyperpolarization.

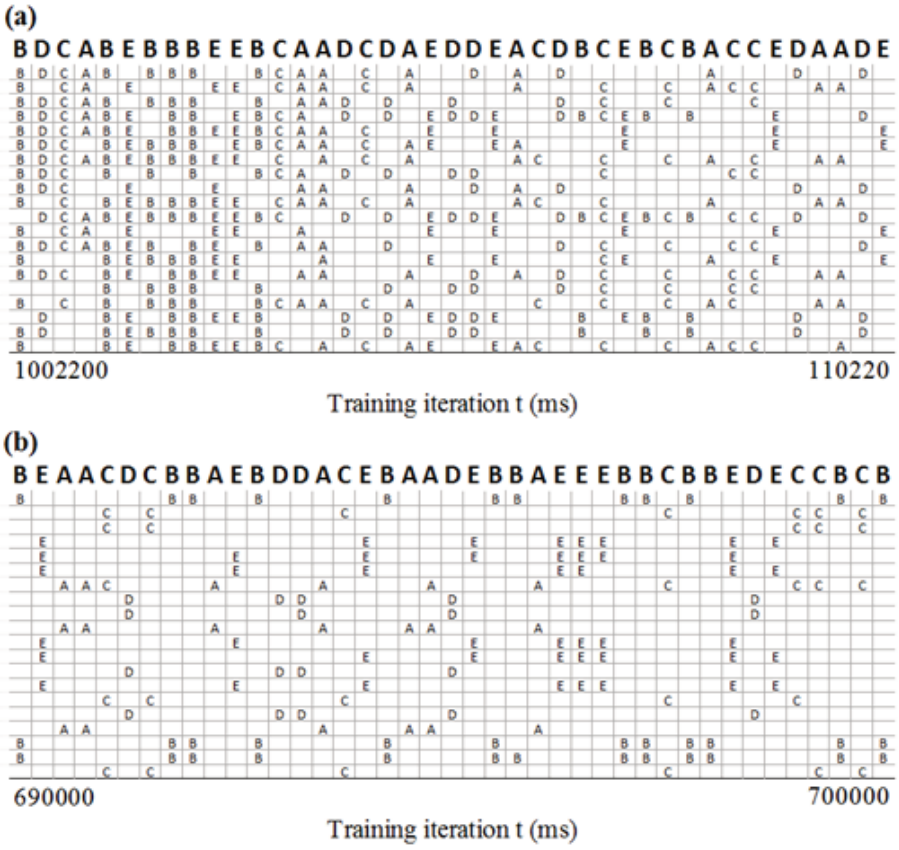

Figure 3.7. Spike output from the L5 layer at the beginning and end of training. Outputs from each of the 20 L5 neurons aligned along the vertical axis. Letters above the pattern constitute one of the five sample "words" displayed at the time. **(a)** Output at the beginning of the training. Even though the WTA network was exposed to very few appearances of each sample of "word," a consistent pattern started to emerge at the very beginning of the training. **(b)** Output at the end of the training.

There were noticeable differences between layer L1 and L5 in the distribution of neurons selective to a single pattern (See Figure 3.8 (b) and (d)). Mean values of the number of neurons per single sample were quite close: 3.956 for L1 and 3.99 for L5, but with significantly different standard deviations: 1.24 for L1 and 2.59 for L5. There were no non-learned samples in L1, while in L5, non-learned samples occurred at the rate of 0.046. The rate of neurons which did not learn any pattern was significantly larger for the L1 layer, 0.22, while in L5 it was just 0.05. I could not determine what factor had the greatest influence over this

difference: a different set of parameters, the stochastic threshold in L5, a difference in input patterns, or perhaps it was due simply to biased measurements of the L5 layer. This requires a detailed theoretical study of the limiting and optimal parameters of STDP rules.

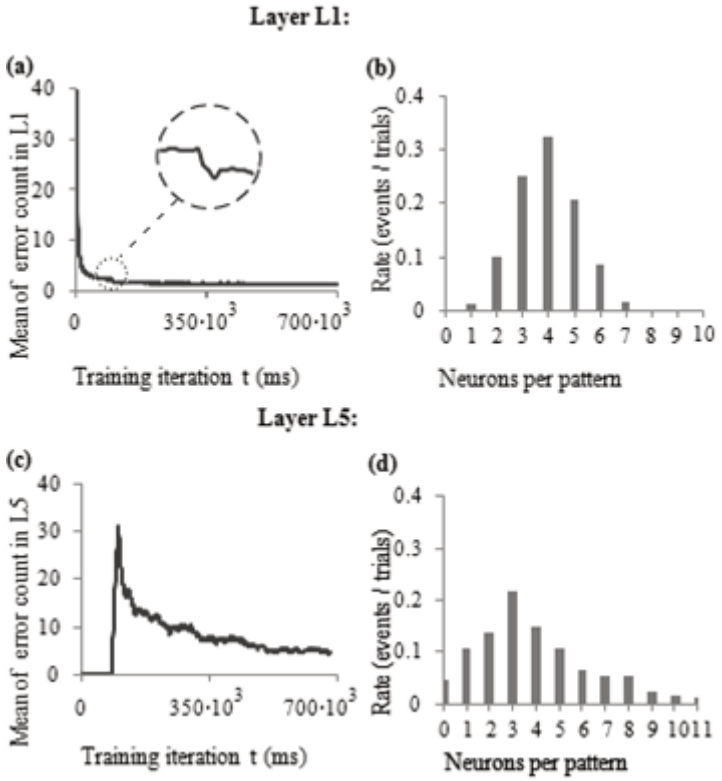

Figure 3.8. Mean error rate and selectivity distribution in WTA layers L1 and L5. Data obtained from 100 experiments, each experiment conducted for 700,000 iterations. **(a)** Mean error rate in layer L1. Sample patterns were regenerated for each experiment. The magnified selection of the series indicates a drop in error rate after the stochastic occurrence of one of the five samples (random letters) was replaced by consistent sequences (random words). **(b)** Distribution of the number of neurons selective to one sample in layer L1. **(c)** Mean error rate in layer L5. Three pre-recorded sample patterns of "words" were used in 100 experiments, each in 1/3 of experiments. **(d)** Distribution of the number of neurons selective to one sample in layer L5.

3.1.7. Discussion

I have demonstrated a model for an unsupervised neural network capable of learning prolonged combinations of spatiotemporal patterns of spikes in continuous mode. In this way I have shown that STDP learning rules alone can be applied to train a neural network to learn long sequences comprised of short samples. Moreover, this model is capable of memorizing and reproducing

sequences in which network input samples were displayed. The reproduction of sequences was achieved by subsequently activating L2.1 neurons.

The fact that memory of events in time can be reproduced implies that such memory could be copied, transferred, compared, etc. Also, it should be relatively easy to extend this particular model to enable it to learn combinations of "words," although that would require additional, more complex modulation at different time scales.

Biological Plausibility of the Model

The model itself and the range of parameters for simulation are arbitrary and cannot be used as a reference to simulation of a true biological process. The model is based on known biological processes, however, and the presence of temporal coding is supported by experimental evidence.

Since I designed this particular network to be as simple as possible, there likely exist many other ways to implement a neural network with the same or similar features which would be more realistic in the biological sense, or show better performance.

For instance, for temporal modulation it would be more realistic to use inhibitory neurons instead of excitatory ones. There is experimental evidence that gamma wave oscillations are generated by inhibitory interneurons (Cardin et al., 2009).

Limitations of the Model

The model requires explicit timing for the occurrence of training samples. In order to use this particular neural network model for real-world data, timing of sensory input must be aligned to the activation periods of the L2 layer. Additional chains of modulation which synchronize sensory input with L2 layer activation periods and/or *vice versa* would solve that problem.

Another obvious limitation of the model is a "blind spot" at each memory read, but this problem could be overcome by multiplying the L1.1 and L4 layers, thus creating an overlapping or sliding memory window.

The simplicity of the structure of the WTA networks used in this model is disputable as well. With increases of different sample counts, the intervals between the same repeating sample increase as well, making learning more and more difficult. Training individual or groups of neurons one by one with a limited number of samples would solve that problem and boost performance. How we could implement that approach for a short temporal code in a rapidly changing environment, however, is a question I cannot yet answer. The well-known adaptive resonance theory (ART) (Carpenter & Grossberg, 2009) solves a similar problem by introducing a self-organizing network and a resonant state between input and previously learned data. Achieving the resonance necessary for ART requires a prolonged state of neural activity (rate code), however, which is not the case with this model. Even so, there are various modifications possible for this particular model.

3.2. STDP Learning under Variable Noise Levels

It has been demonstrated that one of the STDP learning rules is suited for learning spatiotemporal patterns in a very noisy environment. Parameters of the neuron are only optimal, however, for a certain range of quantity of injected noise. This means the level of noise must be known beforehand so that the parameters can be set accordingly. That could present real problems when noise levels vary over time. This is, to some extent, evident from the results of the previous section, 3.1, "Neural Processing of Long-Lasting Sequences of Temporal Codes," where I had to choose different parameters for two different WTA circuits when the input conditions were different.

I found that the model of a leaky-integrate-and-fire inhibitory neuron with an inverted STDP learning rule is capable of adjusting its response rate to a particular level of noise. I propose a method which uses an inverted SDTP learning rule to modulate the spiking rate of the trained neuron. This method is adaptive to noise levels; subsequently the spiking neuron can be trained to learn the same spatiotemporal pattern with a wide range of background noise injected during the learning process.

This work was presented at the 2014 NCTA conference and was published in the conference proceedings and in the SCITEPRESS digital library (Krunglevicius 2014).

3.2.1. Some Properties of the Inverted STDP Rule

Training for Poisson Noise

I exposed neurons with different threshold values to Poisson noise. Each trained neuron received input from 4,096 input neurons which produced Poisson noise by producing an input spike with a probability of 0.02 at each discrete step in the simulation. STDP rules A and B were compared (see Figure 3.9). Results are presented in Figure 3.10.

When exposed to Poisson noise only, STDP rule A, as expected, leads to two possible outcomes: either synaptic strengths decay until the neuron is not capable of firing, or all synaptic strengths grow and the neuron is activated by any random spike from the input.

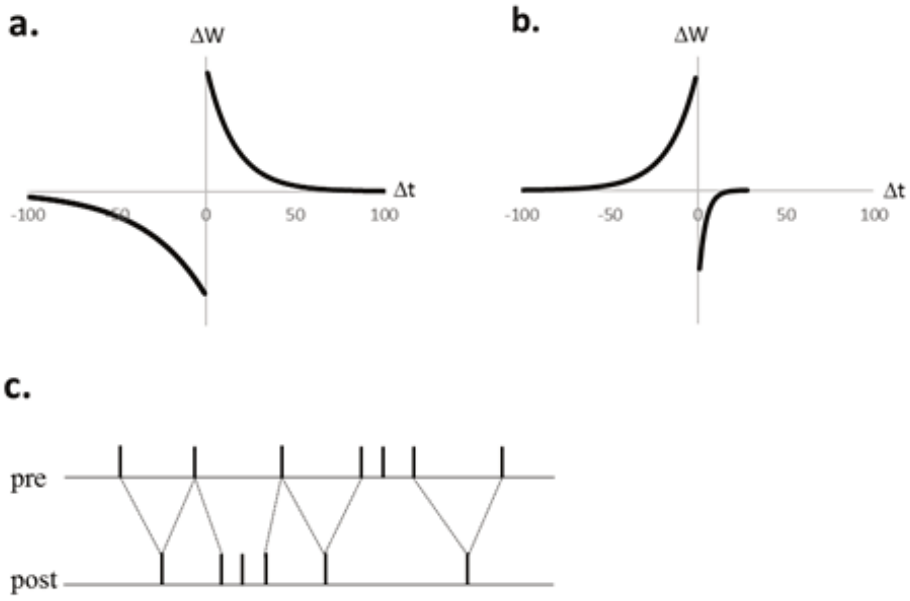

Figure 3.9. STDP training rules addressed in this section. Δw is the amount of change in synaptic strength; Δt is time difference between postsynaptic and presynaptic spikes. **a)** Rule A, STDP rule of excitatory-to-excitatory synapses. **b)** Rule B, STDP rule of excitatory-to-inhibitory synapses. **c)** Update is guarded by the nearest–neighbor rule with immediate pairings only (Burkitt et al. 2004: Model IV).

The behavior of inverted rule B is far more interesting: the neuron tends to stabilize its firing rate at a certain point. The point of stable firing rate depends on more than just threshold variables and the level of noise: the training step and initial values of synaptic strengths are very important as well. It seems that in case of rule B the capping of synaptic strengths at some maximal value is not required.

In this case, if noise is mixed with a recurring spatiotemporal pattern of sufficient size, STDP rule B also leads to remembering the pattern in synaptic strengths, but in an inverted manner: synapses which are associated to the pattern are weaker than those not associated. Compared with rule A, variance of synaptic strengths after training is significantly higher.

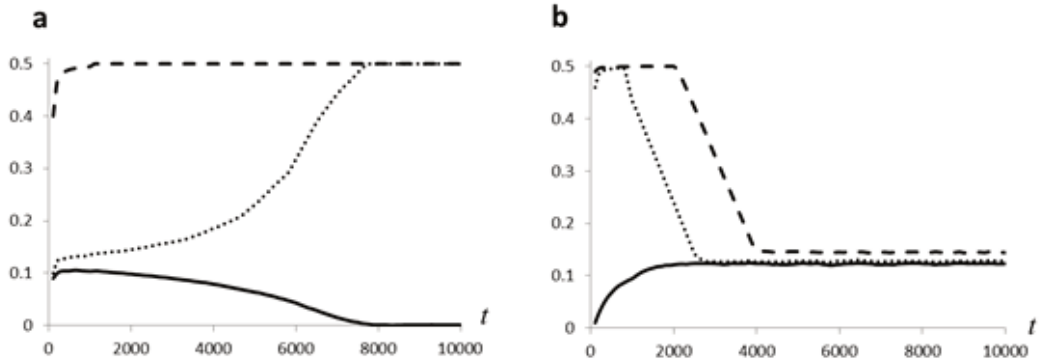

Figure 3.10. Comparison of STDP rules A and B, response rates to the same level of Poisson noise and different neuron thresholds. Vertical axis represents the response rate; horizontal axis represents simulation time. a) STDP rule A, dashed line denotes a threshold value $\vartheta=100$, dotted line $\vartheta=340$, solid line $\vartheta=900$. b) STDP rule B, dashed line at threshold value $\vartheta=100$, dotted line $\vartheta=160$, solid line $\vartheta=170$.

Results depicted in Figure 3.10 are compatible to some extent with the prediction made by the Izhikevich equation for nearest-neighbor STDP interaction for uncorrelated Poisson spike trains (Izhikevich & Desai, 2003). For the standard STDP rule, the Izhikevich equation predicts a point of unstable equilibria of synaptic strengths when the firing rate of the postsynaptic neuron is relatively high, and thus synaptic strengths would likely grow and the postsynaptic firing rate would eventually increase even more. For the inverted STDP rule the same equation predicts a stable point of equilibria, thus the firing rate of the postsynaptic neuron should tend to stabilize. For details please see Figure 2.12b in section 2.2.4 "All-to-All vs. Nearest-Neighbor Spike Interaction." The prediction made by the Izhikevich equation is only valid, however, when both presynaptic and postsynaptic spike trains are Poisson-distributed, which is not the case of the SRM neuron (see section 3.4 "Some Properties of the Postsynaptic Process of the SRM Neuron"), and therefore I measured the behavior of the neuron experimentally.

Stability of Response Rate at Different Noise Levels

To illustrate the dependency of stable firing rate points on the noise level of STDP rule B, I repeated the experiment described in the previous section over a range of Poisson noise. The results are presented in Figure 3.11.

While noise levels increase, depending on the neuron's threshold value the firing rate slowly approaches the maximum value, which is 0.5, since the neuron has a period of absolute refraction equal to one step of the simulation in the model.

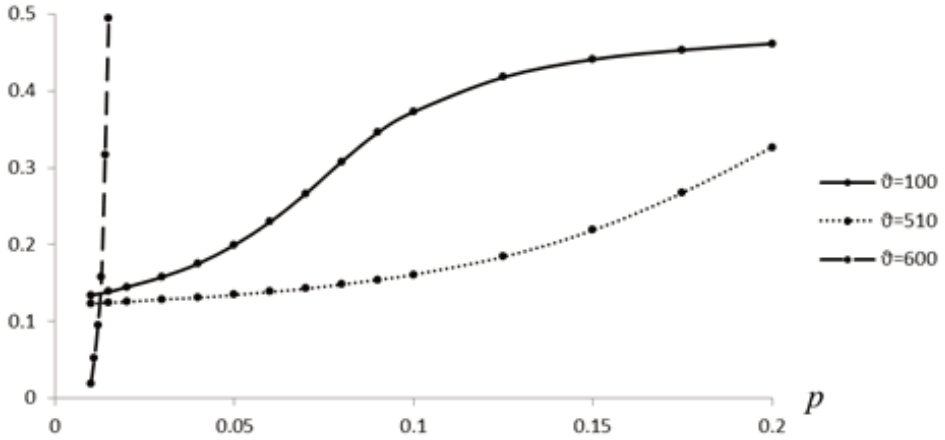

Figure 3.11. Points of stability in STDP rule B. Vertical axis represents the spiking rate, horizontal axis represents the probability of a spike of an individual input neuron at each discrete step of the simulation. Solid line denotes a threshold value of $\vartheta=100$; dotted line means $\vartheta=510$; dashed line denotes response rates when synaptic strengths are static, at $\vartheta=600$.

A neuron with static synapses approaches maximal response rate very rapidly over a narrow range of stimulation (Figure 3.11, dashed line). My goal was to get a neuron to provide inhibition in proportion to the amount of background noise. Therefore I preferred STDP rule B over static synapses.

3.2.2. Materials and Methods

In this work I used a version of the SRM model (see section "2.1.3 Spike Response Model") somewhat modified from the one in my previous experiment, "3.1 Neural Processing of Long-Lasting Sequences of Temporal Codes." I found that the function of action potential is difficult to analyze, so I replaced the η-kernel with the following equation:

$$\eta(t) = -W_r e^{\frac{-\Delta t_h}{T_r}} \tag{3.9}$$

where $\Delta t_h = t - t_{spike}$ and W_r and T_r are the parameters which define the amplitude and duration of relative refraction.

The neuron membrane potential at any given time was:

$$u(t) = \begin{cases} 5\vartheta & if\ t = t_{spike} \\ \eta(t) + \sum \epsilon(t) & otherwice \end{cases} \quad (3.10)$$

where the ϵ-kernel is the same as in section 3.1 given by equation (3.2) and ϑ is the threshold value. Notice at the moment of the spike that membrane potential is set to the constant value. Since at the time of the spike the neuron is in the phase of absolute refraction, the value of membrane potential plays no role in training; it is set to a constant merely to aid in visualization and for the sake of convenience.

Constants during the simulations were the set values: T_m=10; T_r=10; T_s=0.5 and W_r=2ϑ. The threshold value of inhibitory neurons was fixed such that ϑ_{inh}=1,835.

Synaptic plasticity was the same as in section 3.1 and expressed in equation (3.8). Simulation constants for excitatory-to-excitatory synapses were:

A_{LTP}=0.75; A_{LTD}=0.63; T_{LTP}=16; T_{LTD}=35; w_{min}=0.5 and w_{max}=30. Initial synaptic strengths were uniformly distributed between 4.5 and 5.5.

Simulation constants for excitatory-to-inhibitory synapses were:

A_{LTD}=6.048; A_{LTP}=7.2; T_{LTD}=4; T_{LTP}=16; w_{min}=10^{-6} and w_{max}=1.0. Initial synaptic strengths were uniformly distributed between 0.9 and 1.0. Here LTD and LTP switched their places.

The synaptic strengths of static inhibitory synapses was w=7.3 in the case of STDP rule B and w=2.0 otherwise.

3.2.3. Results

I measured the performance and success of the training of a neuron for a spatiotemporal pattern. The sample pattern was generated from 122 neurons firing at the same time. The sample pattern was demonstrated to the network periodically, at intervals of 40 iterations. I executed the experiment on the 1 ms scale so that one iteration corresponded to one millisecond. Overall there were

4,096 neurons in the input layer. All neurons in the input layer produced noise except for the neurons which associated to the pattern at the moment of exposure to the pattern (see Figure 3.14a).

The success of training was evaluated by measuring differences between means of the synaptic strengths of synapses associated to the pattern and of those which were not: $\Delta\mu_w = \mu_{w\_in} - \mu_{w\_out}$. Mean values were scaled to range at the interval [0, 1] respectively to the minimal and maximal values of synaptic strengths. The criterion for successful training was $\Delta\mu_w > 0.85$ at the end of the simulation. Neurons which were unresponsive at the end of the simulation were counted as unsuccessful, despite possible large values for $\Delta\mu_w$. Performance of the training was evaluated by measuring the velocity of $\Delta\mu_w$.

I compared the performance of a simple neural network with that of a network with vertical inhibition (Figure 3.12).

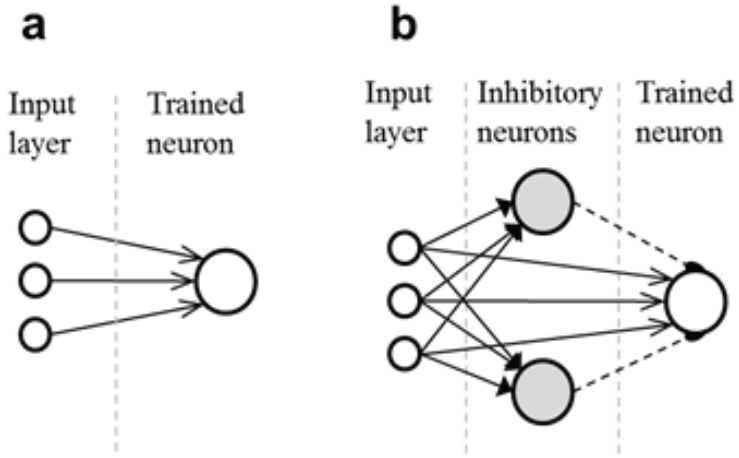

Figure 3.12. Neural network model. **a)** A simple network is composed of the trained neuron and a number of afferents in the input layer. Here only the STDP rule A is applied. **b)** Network with vertical inhibition. This network is extended with multiple inhibitory neurons stimulated from an input layer and subsequently inhibit the trained neuron. Inhibitory synapses (dashes lines) have static weights. Synapses from the input layer to the inhibitory neurons are updated by STDP rule B, and synapses from the input layer to the trained neuron are updated by STDP rule A.

The neural network with vertical inhibition consisted of an input layer, multiple inhibitory neurons and the training neuron. The training neuron received input from all neurons in the input layer, while each inhibitory neuron received input from a random fraction of the input layer (~10%). The training neuron had

synapses with STDP rule A, while inhibitory neurons had synapses with STDP rule B. In addition the training neuron received inhibition from inhibitory neurons via static synapses (Figure 3.12b).

In order to reduce variance in inhibitory postsynaptic potentials (IPSP), I added multiple inhibitory neurons instead of just one. Variance of IPSPs reduces correlation between the presynaptic spike of the sample pattern and the postsynaptic spike; therefore it has a negative influence on the training process. By selecting only a fraction of input neurons I ensured inhibitory neurons would not fire synchronously. The network contained 50 inhibitory neurons.

Training at Different Levels of Constant Noise

I conducted a number of experiments at different levels of Poisson noise mixed with a recurring spatiotemporal pattern. Poisson noise was generated by setting a fixed probability for an input spike at each iteration of the simulation. Success of the training was measured over a range of neuron threshold values ϑ. The amplitude of relative refraction was set to $W_r=2\vartheta$.

In the case of a simple network (Figure 3.13a) I observed, as expected, that below a fixed threshold value, training is only possible within a narrow range of noise levels.

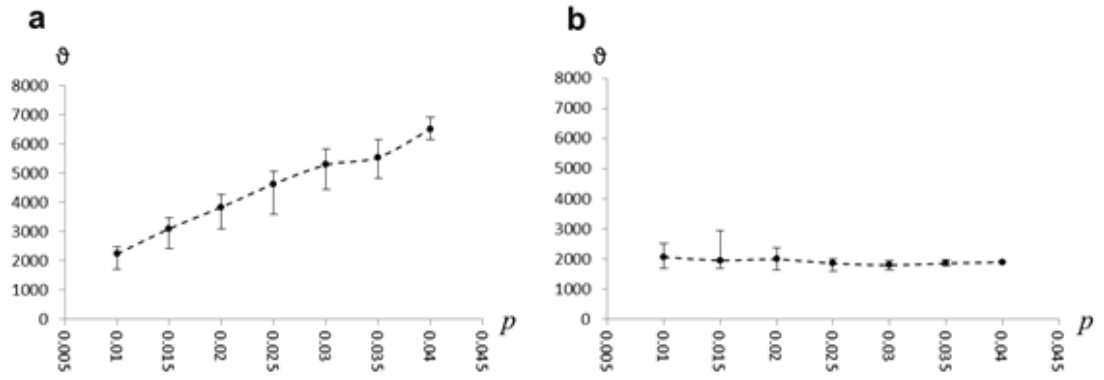

Figure 3.13. Dependency of training success on neuron threshold value and the level of injected Poisson noise. Vertical axis represents the neuron threshold value ϑ, horizontal axis represents the level of noise. **a)** Results from a simple network. Markers represent the point where training was most rapid; error bars represent the range of ϑ when training was successful. **b)** Results from a network with vertical inhibition and STDP rule B.

In the case of the network with adaptive vertical inhibition (Figure 3.13b), the optimal value for the threshold was much less dependent on the level of noise, and remained more or less stable. The same neuron with a fixed threshold could be trained over the broad range of noise levels I used in my experiment (0.01 to 0.04). The range of possible threshold values narrows, however, as noise increases. This was due, most likely, to increased variance of postsynaptic potentials, which reduced the correlation between the spike from the input neuron (presynaptic spike) and the spike of the trained neuron (postsynaptic spike).

Training with Varying Noise Levels

In my next experiment I trained neurons with variable levels of injected noise. I used a sine function for setting the probability for the input neuron to fire: $p=0.01+0.015*((sin(t/\lambda)+1))$. See Figure 3.14a. I evaluated training performance values of λ of 50, 100 and 150.

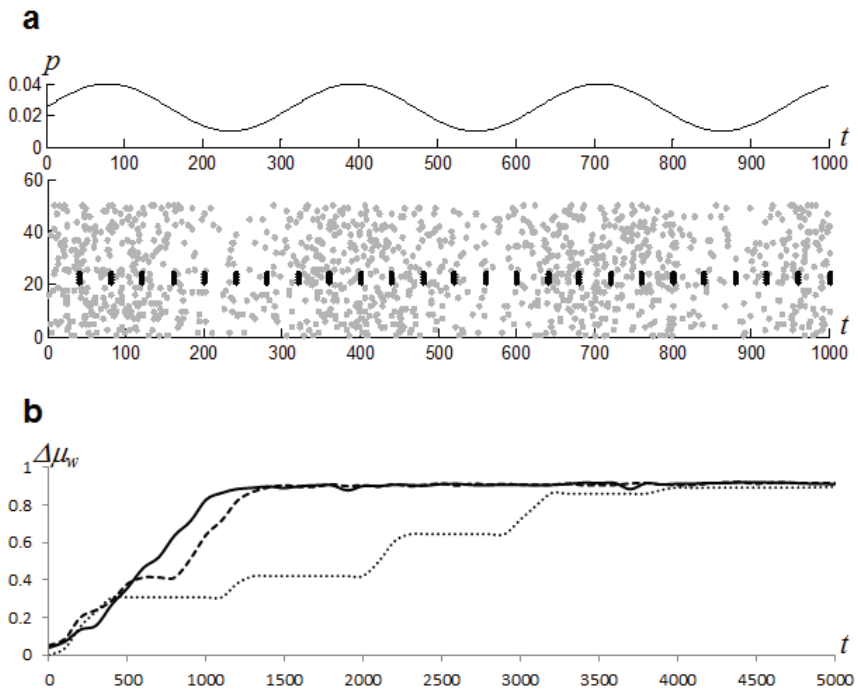

Figure 3.14. Training with varying noise level. a) Example of input spikes. Black dots represent a fraction of the sample pattern, grey dots represent injected noise. b) Values of $\Delta\mu_w$ during the first 5,000 training iterations. Results are from an experiment where λ=150. Solid line denotes a network with STDP rule B, dashed line denotes a network with static synapses of inhibitory neurons; dotted line denotes a simple network.

I compared the performance of a simple network, a network with vertical inhibition and STDP rule B, and a network with static synapses of vertical inhibition. The results are presented in Figure 3.14b.

The training was executed over a range of a neuron threshold values and only the best results were taken into account.

In all cases of λ, the network with STDP rule B performed best. The network with static inhibitory neurons performed only slightly worse, which was a somewhat surprising result. The simple network was the worst performer because the trained neuron was capable of firing only at peaks of stimulation from the input layer.

3.2.4. Discussion

I proposed a method which uses an inverted SDTP learning rule to modulate the spiking rate of the trained neuron. I have shown that this method can be applied to extend the range of noise levels under which a neuron is able to learn a spatiotemporal pattern. There are upper limits, however, for the level of noise under which a neuron can be successfully trained. By tuning the threshold value, the neuron can be trained under conditions of much more intense noise than I achieved in my experiments. This is likely caused by the increased variance introduced by vertical inhibition. This problem requires additional research.

In my experiments I used a sample pattern of a fixed size encoded as parallel singular spikes. This is not a necessary condition: the sample pattern can be encoded as parallel spike bursts or as parallel fixed temporal patterns (Gilson et al., 2011; Masquelier et al., 2008) and the sample patterns can vary in size. Plainly these factors influence the amount of stimulation received by the trained and inhibitory neurons, so that the effect of vertical inhibition might be very different. This is the subject of my continuing research.

The main motivation for this research was to explore prospects for building a practical machine based on STDP. I did not intend to simulate any particular biological neural system. It is difficult to determine to what extent the training

model is possible biologically, and my model ignores the many non-linearities of STDP known from biological research (Caporale & Dan, 2008; Pfister & Gerstner, 2006; van Elburg & van Ooyen, 2010), and likewise it does not take into account short-term plasticity, meta-plasticity and etc.

3.3. Competitive STDP Learning of Overlapping Spatial Patterns

This work was published in the scientific journal "Neural Computation," MIT Press (Krunglevicius, 2015).

This work is a continuous of the effort to improve the WTA circuits described in section 3.1 "Neural Processing of Long-Lasting Sequences of Temporal Codes."

It has been demonstrated that one STDP learning rule is suited for learning spatiotemporal patterns. When multiple neurons are organized in a simple competitive spiking neural network, this network is capable of learning multiple distinct patterns. If patterns overlap significantly, i.e., patterns are mutually inclusive, however, competition would not preclude trained neurons responding to a new pattern and adjusting synaptic weights accordingly.

Here I present a simple neural network which combines vertical inhibition and a Euclidean distance-dependent synaptic strength factor. This approach helps to solve the problem of pattern-size-dependent parameter optimality and significantly reduces the probability of a neuron forgetting an already learned pattern. For demonstration purposes, the network was trained for the first ten letters of the Braille alphabet.

I address two problems associated with the learning of spatial patterns using the STDP learning method. Consider a neuron trained for a two-dimensional pattern in the shape of the letter F. Later the neuron is presented with a pattern in the shape of the letter E, which includes the letter F. In a simple competitive network there is no clear mechanism preventing this neuron from being selective for the pattern E.

The second problem is closely related to the first. For training to be successful, the values of the threshold and initial synaptic weights must be tuned correspondingly to the amount of anticipated stimulation. The amount of

stimulation received, besides the properties of the neuron and synaptic weights, depends on the spatial and temporal properties of the sample pattern. Consider two simple spatial patterns in two-dimensional space: one occupies 2% of incoming synapses, the other 50%. Now suppose the patterns do not overlap. Suppose there are two neurons trained to each of these patterns. In that case, the synapses of the neuron capable of firing at the 2% size pattern must be 25 times stronger than those of the neuron firing at the 50% size pattern (conversely, the threshold value would be 25 times smaller). Otherwise the second neuron would respond to random noise, and the response rate would be very high, potentially leading to unstable learning, especially in the case of the nearest-neighbor STDP update rule (Izhikevich & Desai, 2003). Even if one assumes that in the case of the larger pattern training stops much earlier, there is no way to set initial weights to fit both patterns.

It isn't known whether these problems hold true in biological neural systems, but they definitely pose an obstacle to applying competitive spiking neural networks to pattern recognition.

To solve this problem one could build a highly heterogeneous network, but that approach could be too complicated, and inefficient for practical application. Another method would be to introduce additional mechanisms such as external modulation or homeostatic meta-plasticity to maintain the neuron at a manageable response rate.

I took the simple approach of setting up a two-layer network in such a way that each neuron is activated only within the constraints of a certain range of input stimuli. In other words, the underlying idea is to place individual neurons in conditions where the neuron capable of learning the F pattern is less likely to fire when exposed to E, due to bottom-top inhibition, and the neuron capable of learning E does not reach firing threshold when exposed to F, due to insufficient synaptic strengths. This was achieved by assigning each trained neuron a spatial coordinate, introducing an inhibitory counterpart into the training layer and

setting synaptic strength factors in proportion to the Euclidean distance to the input neuron.

3.3.1. STDP Learning Success Dependency on Quantity of Stimulation and Synaptic Strength Factor

To demonstrate the influence of synaptic strength factor on the successful learning of spatiotemporal patterns of different sizes, I conducted an experiment with a simple neural network simulation. The network consisted of a single training neuron with 4,096 input neurons. The neuron was exposed to the pattern at 30 ms intervals. During pattern exposure time the synapses which were assigned to the pattern received 10 ms spike trains with a 1 ms refractory period after each spike. The other synapses and all synapses during the interval between patterns produced Poisson noise exclusively by firing with a probability of 0.02. Maximal synaptic strength was 1 and initial strengths were set to 0.333.

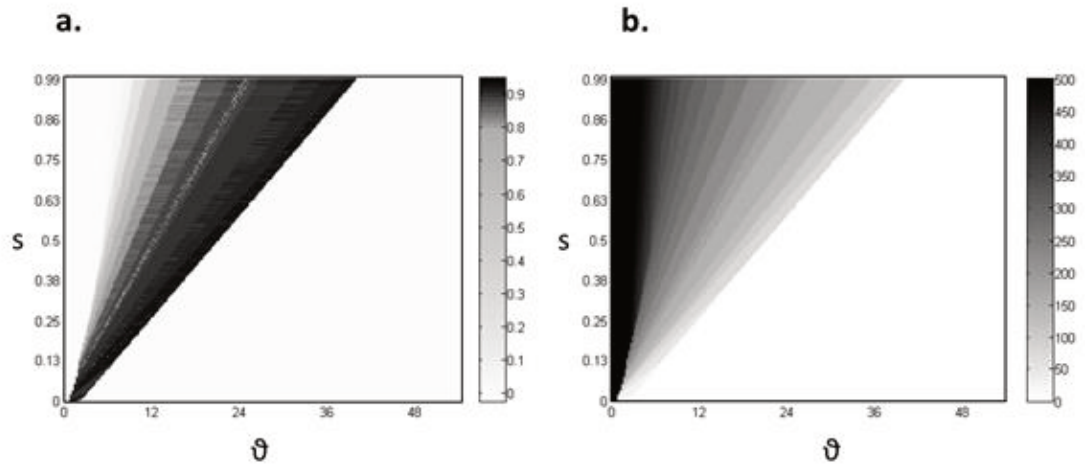

Figure 3.15. STDP learning success dependency on pattern size and synaptic strength factor. The plots represent values after 5,000 training iterations. Here *S* is the ratio of incoming synapses which transmit the pattern to the total count of incoming synapses. **a:** Difference in average strengths of synapses which received the pattern and those which did not. **b:** Neuron response rate.

Initial pattern size *S* was set to 50 and increased up to 4,050 by increments of 5. This was done by simply taking the first 50 synapses on the list, 55 the next time, and so on. The success of training for each of patterns was tested across a range of threshold values ϑ. The neuron was reset after each test for pattern and threshold value.

Trainings were executed at 1 ms discrete time steps, each training taking 5,000 ms. Synaptic strengths and neuron firing rate were measured after each 1,000 ms.

3.3.2. The Network Circuit

I have proposed a network circuit capable of being trained for a wide range of patterns of different sizes (Figure 3.16a).

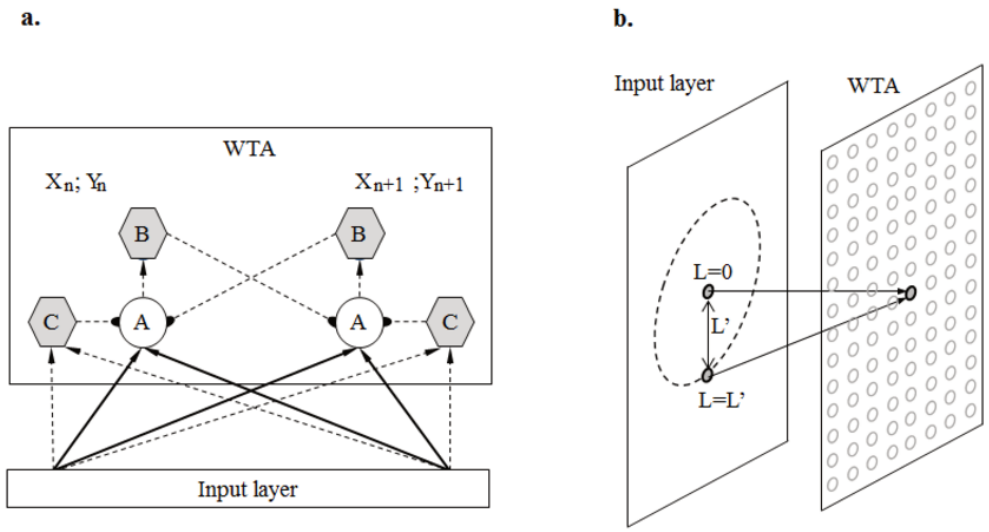

Figure 3.16. The network model. **a:** Circuit of neurons in the competitive neural network. Solid lines indicate synapses with STDP learning, dotted lines with arrows are static synapses. Arrows indicate excitatory synapses, rounded endings indicate inhibitory synapses. **b:** Schematic representation of synaptic strength scaling factors.

The network is organized as two layers: the input layer and the winner(s)-takes-all (WTA) layer. In both layers the neurons are arranged as a rectangular grid. There are no lateral connections in the input layer. In the WTA layer there are three neurons in each cell of the grid: A neurons which have plastic synapses from the input layer and are subject to training, B neurons which are inhibitory interneurons, and inhibitory C neurons which are the antagonistic counterparts to the A neurons. Each B neuron receives input from a single A counterpart and inhibits neighboring A neurons. C neurons are stimulated from the input layer and each inhibits a single A neuron.

Synapses from the input layer to both A and C neurons have distance-dependent synaptic-strength scaling factors. Distance is the Euclidean distance between the

neuron in the WTA layer and the individual neuron in the input layer, assuming that the distance between layers is zero (Figure 3.16b). A single input neuron occupies a single unit of space. Weight scaling factors are assigned by the Gaussian function $\phi_j = exp(-L^2/2\sigma^2)$. Neurons A and C have different σ^2 parameters; the network is tuned in such a way that the C neuron requires slightly more stimulation to fire than the naïve A neuron. Inhibitory synapses from neurons C to A have strong static weights, so that even relatively small activity by neuron C prohibits neuron A from firing. Therefore neuron C is an activity gate; there is no subtractive normalization involved. The network eliminates potentially over-active neurons from competition in the training layer. Otherwise, the overly rapid firing-rate of a trained neuron combined with the nearest-neighbor implementation of STDP would result in an unstable training where the weights of all synapses which have any input at all would grow (Izhikevich & Desai, 2003).

Inhibitory synapses from B to A neurons also have distance-dependent strength factors. Taking this approach, only close neighbors in the WTA layer actually compete.

This particular circuit assigns an individual receptive field for each A and B neuron in the WTA layer. In the event of a spatially concentrated pattern, stimulation of the WTA layer is transformed into a crater-shaped or peak-shaped function, which is discrete in space and where lateral competition between A neurons maximizes this function locally. In other words, if the pattern is relatively large then the winning neurons are those surrounding the pattern, and if pattern is relatively small they form a geometrical approximation of the pattern.

In the case of a simple competitive WTA circuit, competing winning neurons occur because of the random advantage obtained by assigning random initial synaptic strengths. In the case of this particular circuit, advantage is predetermined by the properties of the input pattern and the geometrical position

of a given neuron. Moreover, this circuit allows keeping firing rates of trained neurons within a manageable range, to some extent independent of the spatial size of the input pattern, thus reducing the necessity of carefully tuning the values of the threshold and initial weights.

Summarizing the above: let's assume we have three neurons which are not interconnected, and these three have large, moderate and small threshold values. Let's assume the same feed to all three, but one optimal for the moderate one. The "large" neuron would barely fire; training performance would be poor or fail. The "small" neuron would fire rapidly from the beginning; the performance of training would be poor or unstable. In this particular network there is no difference in thresholds; instead, the initial response rate is predefined by the geometrical properties of the input pattern and the position of the neuron itself. Instead of letting the large and small ones compete "as equals" with the moderate one, both of the former are inhibited by vertical and horizontal inhibition, respectively. Such neurons remain silent and naïve, so they could be used to learn other patterns when the input feed is changed to a lower or higher level of stimulation.

The neuron model and the model of synaptic plasticity were identical to the one described in section "3.2 STDP Learning under Variable Noise Levels". During the simulations the constants were set at arbitrary values: threshold value ϑ=100, T_m=10 ms; T_r=10 ms; T_s=0.5 ms; W_r=200; A_{LTP}=0.3; A_{LTD}=0.252; T_{LTP}=12 ms and T_{LTD}=48 ms. The only difference was that synaptic strength factor ϕ in the ϵ-kernel function (3.2) was not restricted to +1 and -1, but was dependant on the Euclidian distance between the afferent and postsynaptic neurons.

3.3.3. Results

Learning a Stochastic Pattern

To illustrate some of the properties of this neural network model, I trained it for a single stochastic pattern where the firing rates of the input layer were defined by a probability map (Figure 3.17).

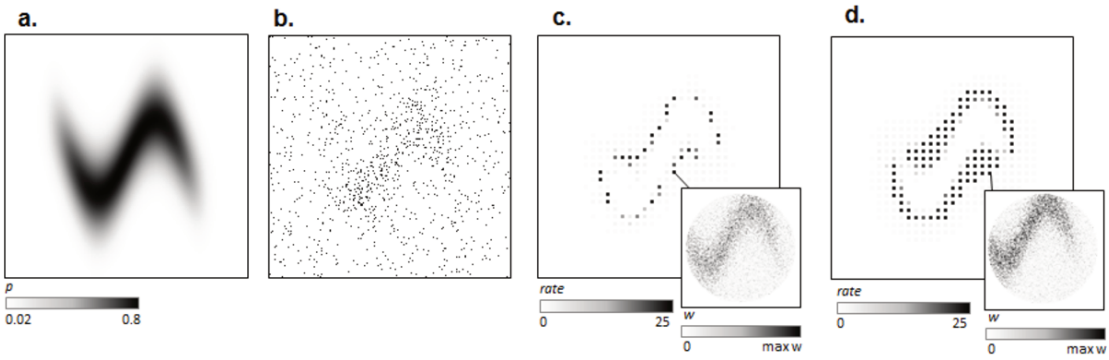

Figure 3.17. Network training for a stochastic pattern. **a:** Probability map of a sample pattern. **b:** Single frame of an input layer. **c** and **d:** Activity of the neurons and synaptic strengths of an arbitrarily chosen neuron at the end of the training. Large rectangles represent the activity of neurons in the training layer and small rectangles represent synaptic weights of a particular neuron at the end of the training. **c:** lateral inhibition present. **d:** lateral inhibition absent.

The simulation was done in 1 ms time steps. The pattern was shown to the network in a periodic manner, for 10 ms at 30 ms intervals. Input neurons fired independently with a fixed probability assigned to produce a spike at each discrete step of the simulation. During the period of exposure to the pattern, the probability of a spike from an individual input neuron was assigned a value from the probability map (Figure 3.17a); values were in the range of from 0.02 to 0.8. During the period between exposure to the patterns, the probability of an input spike was set to 0.02. The size of the input layer was 196 x 196 and the training layer was 29 x 29. Training was conducted over 5,000 iterations.

The parameters of the simulation were set manually: σ_A^2=256; σ_C^2=25; σ_{BA}^2=25; w_{BA}=400; w_{AB}=200; w_{CA}=10000; w_{IC}=0.15; initial synaptic strength w_{ini}=0.26; and maximal synaptic strength w_{max}=1.28.

The winning neurons are those surrounding the pattern in Figure 3.17c and in Figure 3.17d. Neurons which are proximate to the most stimulated area or are too distant from the pattern remain naïve. Neurons which are proximate to the most stimulated area are inhibited from firing because of the activity of neuron C, and neurons which are too distant cannot fire because of insufficient excitation. Consequently, active neurons are organized around the edges of the pattern. In the case of horizontal competition (Figure 3.17c), only the most

stimulated neurons which are closest to the edge win. Figure 3.17d represents neural activity in a network without lateral inhibition, i.e., training neurons do not compete. Lateral inhibition reduces the number of active neurons, but it also reduces the speed of training: in Figure 3.17d synaptic weights associated to the pattern grew stronger than in Figure 3.17c, although in both ceases training was conducted for the same number of iterations.

Learning of Overlapping Braille Symbols

The network (Figure 3.16) was trained for the first ten letters of the Braille alphabet, a, b, c, d, e, f, g, h, i and j. Each Braille dot was represented by a cluster of 61 neighboring synapses in a 64 x 64 grid. All four possible dots had the same fixed locations. The training layer was arranged as a 15 x 15 grid centered in respect to the input layer. The distance between neurons in the training layer was 3 units, while it was a single unit for the input layer.

The network was exposed to the pattern at 30 ms intervals. The simulation was performed in 1 ms time steps. During the period of exposure to the pattern, input neurons assigned to the pattern produced 10 ms spike trains with a 1 ms refractory period. The other input neurons, and all input neurons during the period between pattern exposures, produced only Poisson noise, firing with a probability of 0.02. The same pattern was used for 1,000 iterations, followed by switching over to a different pattern in random order. All patterns were used only once during a training epoch, with 60 training epochs in total. There was a total of 600,000 training iterations.

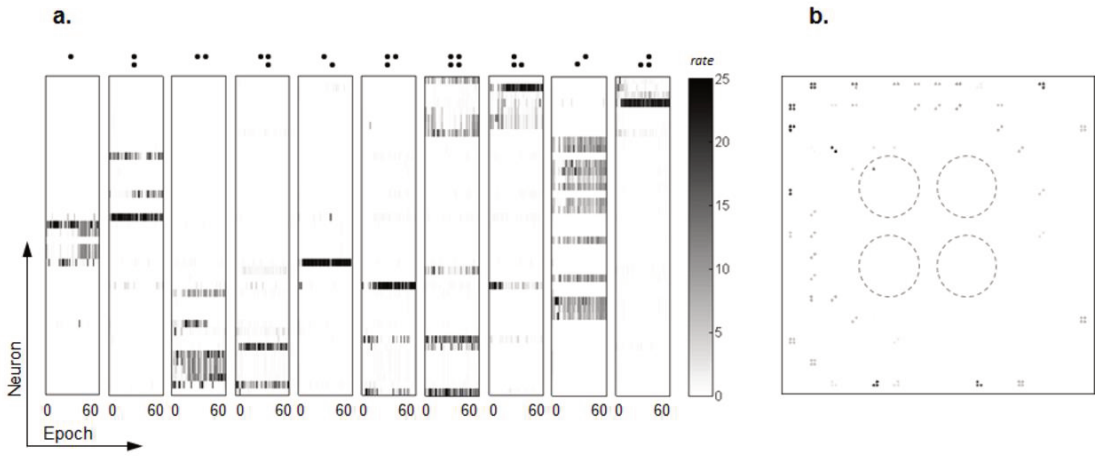

Figure 3.18. Network training of Braille characters. **a:** Columns represent A neuron firing rates per individual Braille character. Characters represented at the top of each column. Each line in a column represents natural activity during all 60 training epochs. During the actual training all characters were used in a single epoch in random order, but in this diagram data are filtered for an individual pattern in each column. **b:** distribution of A neurons selective for an individual pattern in the WTA layer. Grey levels are proportional to synaptic weights. Dashed circles represent the location of Braille dots.

The parameters of the simulation were set manually: σ_A^2=256; σ_C^2=400; σ_{BA}^2=4096; w_{BA}=140; w_{AB}=200; ω_{CA}=1000; w_{IC}=0.12; initial synaptic strength w_{ini}=0.234; and maximal synaptic strength w_{max}=0.594. Notice that σ_{BA}^2 was set to a very high value, essentially enabling all-to-all inhibition. This was done to minimize the number of active neurons. The value of σ_C^2 was also significantly larger than in the previous experiment. While in the previous experiment σ_C^2 was set so that neurons would organize themselves close to the edges of the input pattern but never inside of the pattern, this time it was tuned so that neurons responsive to the four-dot pattern would be the ones close to the boundaries of the training layer, and neurons responsive to a single-dot pattern would be the ones close to or inside of the single-dot pattern.

Figure 3.18a shows the firing rates of the 42 most active A neurons. The activity of the rest of the A neurons in the network was insignificant, with less than 0.25 spikes per pattern on average. Each of the columns represents an individual pattern and the response rates over 60 epochs.

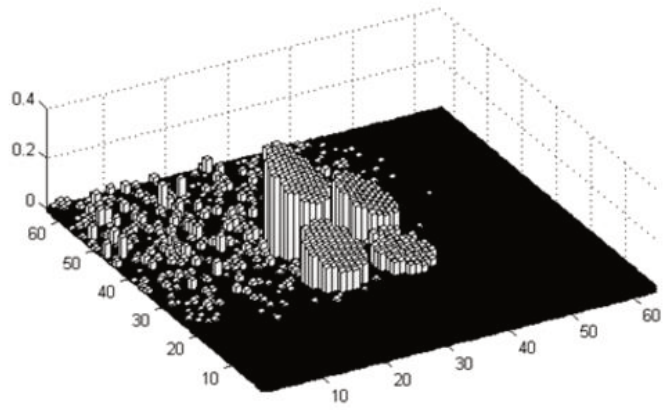

Figure 3.19. Synaptic strengths in proportion to the scaling factor of an individual neuron. This particular neuron was partially selective for a four-dot and three-dot pattern. Strengths associated with one of the dots were scaled down significantly, and thus exerted less influence.

The results show the majority of neurons were mainly selective for a single pattern, and while there were a few neurons which showed some selectivity for more than one pattern, in most cases it was a selectivity switch from a three-dot pattern to a four-dot pattern and back again (see Figure 3.18). There was significant competition between neurons selective for the same pattern, however, and especially in the case of pattern 'i', resulting in decreased activity and consequently slower learning.

In comparison, a simple network with no distance-dependent synaptic-strength scaling and no C neuron failed to discriminate these patterns. Most frequently one or several neurons dominated the entire network and were selective for all 2- to 4-dot Braille patterns, ignoring the single dot pattern at the conclusion of the training. Synaptic strengths varied constantly in attempting to fit the presented sample pattern.

I used only the first ten Braille characters for convenience of visualization. Since most of the neurons in the training layer remained naïve, the network could be trained for more than ten patterns; the number of possible patterns in general is limited by the size of the training layer and the parameters of horizontal

inhibition w_{BA}, w_{AB} and σ_{BA}^2 which define how many of the neurons can be successful in competing for the same pattern.

3.3.4. Discussion

The proposed neural network was successfully trained for ten Braille characters. Although some neurons were selective for more than one character, the pattern could still be identified unambiguously from the combinations of responsive neurons. Attempts to train a simple competitive network for the same characters failed; the network was unable to discriminate most of the patterns. The results show this particular circuit is more efficient than the simple competitive network for the learning of concentrated spatial patterns.

One of the weakest parts of the circuit is competition via lateral inhibition. It negatively affected the speed of learning, especially at the beginning of the training. Also, the model of lateral inhibition is not robust enough to prohibit neighboring neurons from firing. This is associated with the latency of inhibition due to inhibition taking place through an inhibitory interneuron. In current simulations the time step was 1 ms, consequently resulting in a 2 ms latency for the inhibition to take effect. To work around this problem, simulation at the sub-millisecond scale, or a different model of competition, for example, an external modulation circuit, might be considered.

The main motivation for this research was to explore possibilities for building a practical machine based on known STDP properties. I did not intend to simulate the workings of the visual or primary somatosensory cortex where much more complicated processes occur. Cells in the visual cortex are selective for length, width, color and orientation (Hegde & Van Essen, 2000). The sense of touch is also far from simple, with at least four distinct types of mechanical receptors in the human skin, all producing different output patterns (Gardner, 2010).

Simplistic bottom-top training, where associative memory is stored in the synaptic strengths of receiving neurons, appears well suited for pattern recognition tasks, especially because this kind of network is capable of learning

in a very noisy environment. It lacks the ability, however, to reproduce memory, whereas living neural network systems obviously can reproduce memory. Furthermore, non-reproducible memory cannot be compared to the input. Dealing with reproducible and comparable memory requires a different network architecture, such as the one proposed by Adaptive Resonance Theory (Carpenter & Grossberg, 2009).

While the parameters of the STDP function itself were chosen within the range of findings discovered in biological neurons (Caporale & Dan, 2008), other parameters were tuned manually, in what can be considered a stochastic hill-climbing optimization where the overall speed of the training was the main goal. I cannot claim to have achieved global or even local minima here. For that reason, it is still too early at this stage of the research to provide a reliable evaluation of the quality of the training, other than to say that robust training occurs within the range of input stimuli used in these particular experiments.

3.4. Some Properties of the Postsynaptic Process of the SRM Neuron

By making assumptions about presynaptic and postsynaptic processes, such as assuming that presynaptic and postsynaptic spike trains are Poisson-distributed, it is possible to gain theoretical insight into the STDP training process. Izhikevich and Desai (Izhikevich & Desai, 2003), for example, derived equations for the expected value of change in synaptic weight for multiple models of spike neighborhood interactions for uncorrelated and weakly correlated Poisson-distributed spike trains. Other authors used the Fokker-Plank equation to predict the evolution of synaptic weights (Rubin et al. 2001; Gütig et al. 2003; Cateau & Fukai 2003). These studies are based on assumptions of a Poisson process, or on modeling membrane potential through the Ornstein-Uhlenbeck process (Cateau & Fukai 2003), again assuming membrane potential is a Gaussian process.

In this section I discuss some theoretical properties of the SRM model and demonstrate that the assumption of a Poisson process for postsynaptic spikes, at least in the case of the SRM model, is incorrect. The assumption that the process of postsynaptic potential is Gaussian is incorrect as well, unless afterhyperpolarization is absent in the model.

3.4.1. The Normality of the Process of Postsynaptic Membrane Potential

Let's assume that individual postsynaptic potential is expressed by the alpha-function:

$$f_\alpha(t) = w(e^{-\frac{t}{T_m}} - e^{-\frac{t}{T_s}}) \tag{3.11}$$

where $T_s < T_s$ and w is the synaptic strength of an individual synapse. In fact, postsynaptic potential could be expressed by any other continuous decaying function; I'm using a simple one for purposes of illustration.

Let's assume we have a single SRM neuron which receives input from N afferent neurons. Let's assume that afferent spikes are Poisson-distributed, thus firing

with fixed probability *p* at each discrete time step of the simulation. If *N* and p are sufficiently large, integrated postsynaptic potentials are indeed distributed normally, but only if we ignore the component of afterhyperpolarization in the SRM model. We can easily find the mean and variance of this normal distribution:

$$\mu = Np\overline{w}\int_0^\infty f_\alpha(t)dt \tag{3.12}$$

$$\sigma^2 = Np(1-p)(\overline{w}^2 + \sigma_w^2)\sigma_\alpha^2 \tag{3.13}$$

where $\overline{w}$ is the mean synaptic strength the inputs, σ_w^2 is the variance of input strengths and σ_α^2 is given by:

$$\sigma_\alpha^2 = \int_0^\infty f_\alpha(t)^2 dt \tag{3.14}$$

Equation (3.13) follows from the variance of binomial distribution multiplied by the variances of the postsynaptic potentials. For the discrete model, equations (3.12) and (3.14) can be rewritten as:

$$\mu = Np\overline{w}\sum_{t=0}^{\infty} f_\alpha(t) \tag{3.15}$$

$$\sigma_\alpha^2 = \sum_{t=0}^{\infty} f_\alpha(t)^2 \tag{3.16}$$

Figure 3.20 provides the experimental results from the process of the postsynaptic membrane potential of a single SRM neuron with 1,000 afferents. In the experiment all synaptic strengths were set to 1; T_m=10 and T_s=0.5; afferents fired with a probability p=0.04; the simulation time step was 1 ms. Figure 3.20A shows the distribution of integrated PSPs, ignoring postsynaptic spikes and the afterhyperpolarization phase. The process is obviously Gaussian; equations (3.15) and (3.16) give a perfect approximation of the process. When afterhyperpolarization is taken into account (Figure 3.20A), however, the actual

process takes on a very different, multimodal shape. Here threshold $\vartheta=\mu+\sigma$ and the amplitude of afterhyperpolarization $W_r=-\mu$; $T_h=T_m$ (for more details please see Materials and Methods). Therefore postsynaptic potentials cannot be modeled with the Ornstein-Uhlenbeck process, at least in the case of the SRM model. Since the SRM model is able to mimic actual membrane potential with a very high degree of accuracy, there is little doubt that the Ornstein-Uhlenbeck process is not applicable to the biological neuron either.

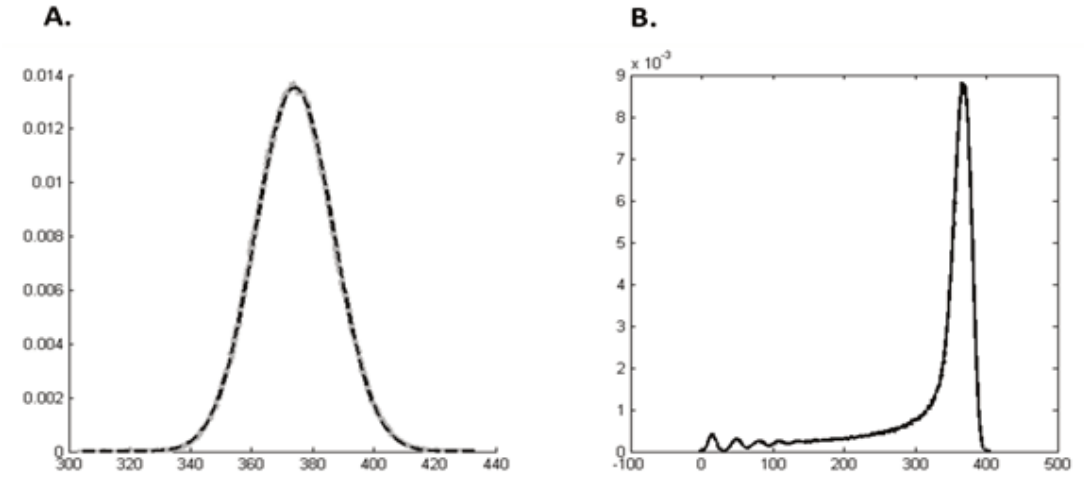

Figure 3.20. Process of integrated postsynaptic potentials of the SRM neuron when exposed to Poisson-distributed input spikes. **A:** process of postsynaptic potentials, excluding afterhyperpolarization. Grey line is experimental results and dashed line is the theoretical prediction, normal distribution with the mean and variance obtained from equations (3.15) and (3.16). **B:** the same process including afterhyperpolarization.

3.4.2. Distribution of the Latencies of Postsynaptic Spikes in the Case of Absence of Afterhyperpolarization

When postsynaptic spikes are modeled by the Poisson process, expected latencies between the postsynaptic spikes are distributed exponentially. Izhikevich and Desai (Izhikevich & Desai, 2003) derived their equations based on exponential distribution. Here I will show an experimentally measured distribution of postsynaptic latencies. It is obvious that these do not follow an exponential distribution even when afterhyperpolarization is absent. See Figure 3.21.

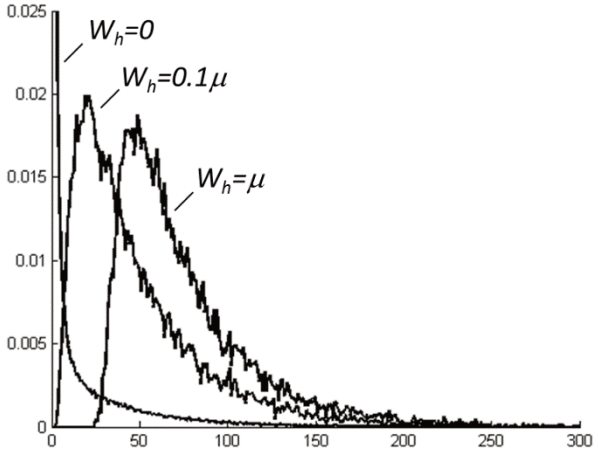

Figure 3.21. Latency distribution of postsynaptic spikes. Experimental data. Here the line is truncated when W_h=0, that is, where afterhyperpolarization is absent (the probability at point 0, in other words the probability of immediate firing without delay, is 0.77).

Unfortunately, the distribution depicted in Figure 3.21 cannot be derived. I tried to develop an algorithm to compute this distribution numerically but I have achieved only limited success. Below I will explain this algorithm and the problems associated with it.

For the sake of simplicity, let's consider a case without afterhyperpolarization. Let's define that the phase of absolute refraction is equal to the single step in discrete time, so that we can ignore it. Then it is easy to compute the probability of the spike. The spike is fired when membrane potential is larger than threshold value, thus the expected firing rate is:

$$E = 1 - \Phi(\mu, \sigma, \vartheta) \tag{3.17}$$

where Φ is the cumulative normal distribution; μ and σ can be obtained from equations (3.13) and (3.15); and ϑ is the threshold value.

Now let's compute the conditional probability of the spike at the moment $t=t_{spike}+1$. It is important to understand that membrane potential is an auto-correlated stochastic process, and thus it has momentum. The degree of

autocorrelation obviously depends on the time constants of the alpha-function. Let's define the auto-correlation function:

$$\rho_{\alpha\alpha}(\tau) = \sigma_{\alpha}^{-2} \int_{0}^{\infty} f_{\alpha}(t) * f_{\alpha}(t+\tau) dt \tag{3.18}$$

the discrete case:

$$\rho_{\alpha\alpha}(\tau) = \sigma_{\alpha}^{-2} \sum_{t=0}^{\infty} f_{\alpha}(t) * f_{\alpha}(t+\tau) \tag{3.19}$$

where τ is the time step and σ_{α}^{2} is computed from the equations (3.14) or (3.16) correspondingly.

For computing the conditional probability of the spike at the moment $t=t_{spike}+1$, we can use a bivariate normal distribution, where $\rho=\rho_{\alpha\alpha}(1)$ and marginal distributions are identical with μ and σ^2 obtained from equations (3.13) and (3.15). Truncating this distribution with threshold ϑ, we can obtain the conditional probability of the consequent spike at the moment $t=t_{spike}+1$. See Figure 3.22, top left distribution, the upper-right quadrant is the conditional probability of firing and the bottom-right quadrant is the conditional probability of not firing.

Truncated bivariate normal distribution (top or bottom part of the distributions in the first column in Figure 3.22) yields an extended skew-normal distribution (ESN) (Azzalini, 1996; 2005):

$$ESN(x) = \frac{1}{\sigma}\varphi\left(\frac{x-\mu}{\sigma}\right)\frac{\Phi\left(\tau\sqrt{1+\lambda^2} + \lambda\left(\frac{x-\mu}{\sigma}\right)\right)}{\Phi(\tau)} \tag{3.20}$$

where

$$\lambda = \frac{\rho}{\sqrt{1-\rho^2}} \tag{3.21}$$

$$\tau = \frac{\mu - \vartheta}{\sigma} \tag{3.22}$$

where φ is a normal probability distribution and Φ is a normal cumulative distribution; ρ is a correlation coefficient and ϑ is the threshold. Notice that normal probability distribution is a special case of ESN when $\lambda = 0$ and ESN approaches normal PDF when τ is large.

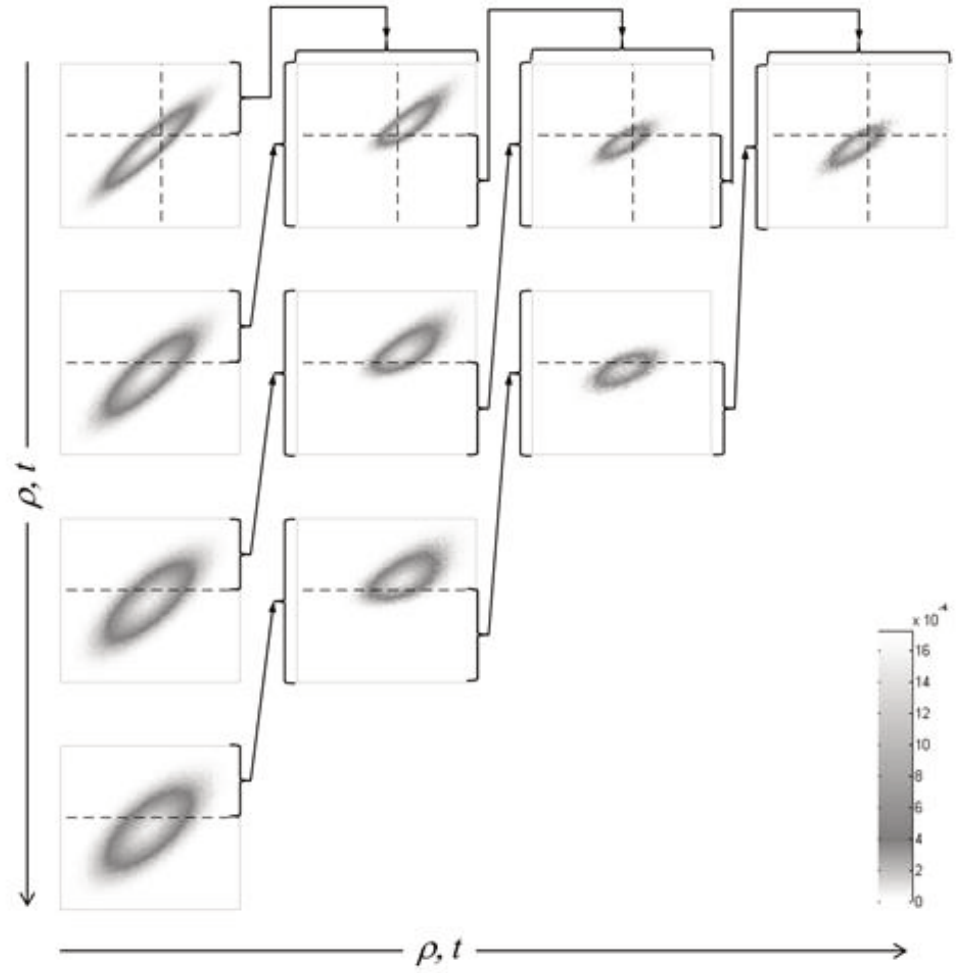

Figure 3.22. Distributions of the postsynaptic membrane potential process when afterhyperpolarization is absent. Dashed lines represent threshold value ϑ. Arrows show the origin of marginal distributions. Drawing is based on experimental data. In the first column there is bivariate normal distribution with correlations for τ=1, 2, 3 and 4. In the second column there are conditional distributions at the time moment t=1 immediately after the spike. In the third column is the conditional distribution when there was no spike at the moment t=1, and in the fourth column there were no spikes at the moment t=1 and t=2.

When we have two extended skew-normal distributions obtained from truncating bivariate normal distributions with correlations $\rho_{\alpha\alpha}(1)$ and $\rho=\rho_{\alpha\alpha}(2)$, we may use these two as marginal distributions to obtain the entire conditional distribution at the time moment t=2 (Figure 3.22, second column). Here we have a bivariate extended skew normal distribution (ESN_2) (Azzalini, 2005):

$$ESN_2 = \varphi_2(x-\mu_1, y-\mu_2, \Sigma)\frac{\Phi(\alpha)}{\Phi(\tau)} \tag{3.23}$$

where:

$$\alpha = \alpha_0 + \frac{\alpha_1(x-\mu_1)}{\sigma_1} + \frac{\alpha_2(y-\mu_2)}{\sigma_2} \tag{3.24}$$

$$\alpha_0 = \tau\sqrt{1+\alpha_1^2+\alpha_2^2+2\alpha_1\alpha_2\omega} \tag{3.25}$$

$$\alpha_1 = \frac{\delta_1-\delta_2\omega}{\sqrt{(1-\omega^2)(1-\omega^2-\delta_1^2-\delta_2^2+2\delta_1\delta_2\omega)}} \tag{3.26}$$

$$\alpha_2 = \frac{\delta_2-\delta_1\omega}{\sqrt{(1-\omega^2)(1-\omega^2-\delta_1^2-\delta_2^2+2\delta_1\delta_2\omega)}} \tag{3.27}$$

where φ_2 is a bivariate normal PDF; $\delta_1=\rho_{\alpha\alpha}(1)$; $\delta_2=\rho_{\alpha\alpha}(2)$; $\mu_1=\mu_2=\mu$ and $\sigma_1=\sigma_2=\sigma$ from equations (3.15) and (3.16); $\omega=\delta_1$; δ_1, δ_2 are the correlation coefficients of a bivariate normal PDF which produced ESN marginal distributions and ω is the correlation coefficient used in the covariance matrix of φ_2 in equation (3.23). These correlation coefficients are from equation (3.19), when the time moment t=1.

The extended skew-normal distribution (3.23) perfectly fits the experimental data, see Figure 3.23.

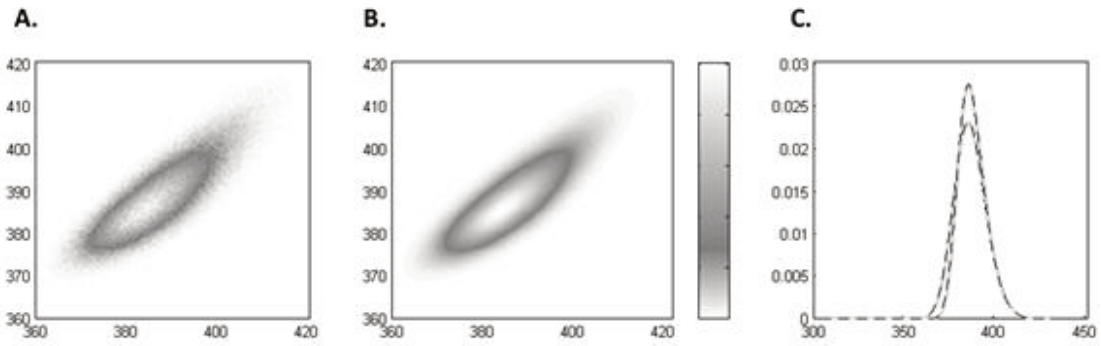

Figure 3.23. Bivariate skew normal distribution is a perfect prediction of membrane potential at the moment t=1. **A:** experimental data. **B:** theoretical prediction. **C:** marginal distributions; grey line is experimental data, dashed line the prediction.

At this point I have solved the first two columns of the triangle depicted in Figure 3.22. Thus it is possible to estimate the conditional probabilities of the spike at the moments t=1 and t=2 with high precision. The third and subsequent columns are problematic. They cannot be derived, but it is highly likely that it is possible to estimate these distributions numerically. The marginal distributions of the distribution in the third column could be expressed as:

$$ESN(x) = \frac{1}{\sigma}\varphi\left(\frac{x-\mu}{\sigma}\right)\frac{\int_{-\infty}^{\vartheta}\frac{1}{\sigma_2}\varphi(\xi)\Phi(\beta_0(x)+\beta_1(\xi))dy}{\int_{-\infty}^{\vartheta}\frac{1}{\sigma}\varphi\left(\frac{z-\mu}{\sigma}\right)\frac{\Phi\left(\tau\sqrt{1+\lambda^2}+\lambda\left(\frac{z-\mu}{\sigma}\right)\right)}{\Phi(\tau)}dz} \tag{3.28}$$

where:

$$\xi = \frac{y-\mu-\omega(x-\mu)}{\sigma_2} \tag{3.29}$$

$$\beta_0(x) = \alpha_0 + \alpha_2\left(\frac{x-\mu}{\sigma}\right) + \alpha_1\omega\left(\frac{x-\mu}{\sigma}\right) \tag{3.30}$$

$$\beta_0(x) = \alpha_1\sqrt{1-\omega^2} \tag{3.31}$$

Equation (3.28) cannot be solved because of the unknown Gaussian integrals. I attempted to use skew-normal distribution again, that is, to fit distribution (3.28) to (3.20) by estimating the moments of truncated ESN_2 and applying the method of the moments to make it fit. There are, however, multiple roots in the solutions, and I was not able to find a reliable way to select the best solution to fit the experimental data. On the other hand, using normal distribution for column 3 yields quite good results. Absence of afterhyperpolarization is a very special case, however, and I saw little value in pursuing a solution to this problem to the end. The whole point of this section is to demonstrate that even in the absence of afterhyperpolarization one cannot derive a distribution of the latencies of postsynaptic spikes.

Introducing afterhyperpolarization, the method I described above is only applicable in cases where the threshold $\vartheta > \mu$. One should adjust the values of ϑ when solving the triangle depicted in Figure 3.22. In the case of $\vartheta < m$, one may encounter multimodal distributions (see Figure 3.23), and thus this method cannot be the generic solution.

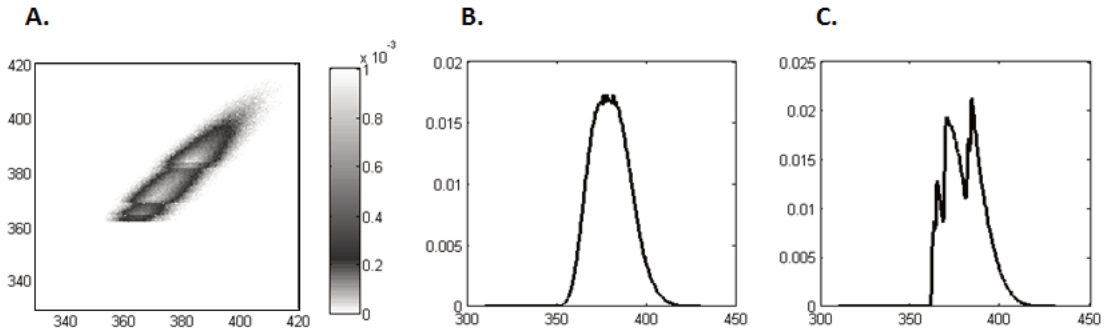

Figure 3.24. A multimodal case of the process of membrane potential at the moment t=1. From experimental data recorded when threshold $\vartheta = \mu - \sigma << \mu$, afterhyperpolarization is a very steep $T_h=1$, and there is a large $W_h=-4\sigma$. **A:** bivariate distribution. **B:** marginal distribution along the *x* axis. **C:** marginal distribution along the *y* axis.

3.4.3. Discussion

I have demonstrated that the SRM neuron cannot produce Poisson-distributed postsynaptic spikes when exposed to a Poisson-distributed input. This applies to the SRM model but not to the biological neuron *per se*. But since the SRM model is known to be a quite an accurate model of the biological neuron, this might be considered strong evidence this is true of the biological neuron as well.

Since afferent neurons are supposed to mimic their biological counterparts, it is questionable whether Poisson-distributed spike trains are a viable model for encoding data.

This puts under scrutiny the work of Izhikevich and Desai (Izhikevich & Desai, 2003), where they relate the STDP learning rule to the BCM theory (Bienenstock et al. 1982). The findings of Izhikevich and Desai will probably stand when applied to spike probability distributions similar to the one depicted in Figure 3.21, but this has not yet been proven.

I am certainly not the first researcher to question the biological realism and viability of Poisson-distributed spike trains, see also Linder, 2006; Cateau & Reyes, 2006.

3.5. Modified STDP Triplet Rule Significantly Increases Neuron Training Stability in the Learning of Spatial Patterns

This work was published in the scientific journal "Advances in Artificial Neural Systems" (Krunglevicius, 2016).

Besides a variety of different STDP rules known from biological studies, there is also a number of phenomenological models of STDP. It is important to understand the limitations and advantages of the different STDP models. In this section three different additive STDP models of spike interactions were compared in respect to training performance when the neuron is exposed to a recurrent spatial pattern injected into Poisson noise. The main motivation for this work was to explore which model of STDP would be the best option for the neural circuits introduced in sections 3.1, 3.2 and 3.3. I compared three different additive STDP implementations: all-to-all interaction, nearest-neighbor interaction with immediate pairings (Burkitt et al., 2004), and the same nearest-neighbor interaction with triplet update (Pfister & Gerstner, 2006). The parameters of the neuron model and STDP training rules were optimized for a range of spatial patterns of different sizes by means of a heuristic algorithm. The size of the pattern, i.e., the number of synapses containing the pattern, was gradually decreased from what amounted to a relatively easy task with a large number down to a single synapse. Optimization was performed for each size of the pattern. The parameters were allowed to evolve freely, without adherence to the constraints of biological realism. The triplet rule performed better by far in most cases than the other two rules, while the evolutionary algorithm immediately switched the polarity of the triplet update. The all-to-all rule achieved moderate results.

In order to build a useful pattern recognition machine based on STDP learning, it is important to understand the limitations of the model; for example, ask the simple question: how small can the spatial pattern be in respect to the number of overall incoming synapses? By making assumptions about presynaptic and

postsynaptic processes, such as assuming that presynaptic and postsynaptic spike trains are Poisson-distributed, it is possible to gain theoretical insight into the STDP training process. Izhikevich and Desai (Izhikevich & Desai, 2003), for example, derived equations for the expected value of change of synaptic weight for multiple models of spike neighborhood interactions with uncorrelated and weakly correlated Poisson-distributed spike trains. Other authors used the Fokker-Plank equation to predict the evolution of synaptic weights (Rubin et al., 2001; Gutig et al., 2003; Cateau & Fukai, 2003). These studies are based on assumptions of a Poisson process, or on modeling membrane potential through the Ornstein-Uhlenbeck process (Cateau & Fukai, 2003), again assuming that membrane potential is a Gaussian process. While such methods might be valuable, they are limited to simple distributions, which is not the case in a mixture of Poisson noise and spatial pattern. Moreover, the spike-response-model I use in my research cannot produce Poisson-distributed postsynaptic spikes, because membrane potential is a highly auto-correlating process, except when postsynaptic potentials are modeled by unit impulses, which is biologically implausible (see section "3.4 Some Properties of the Postsynaptic Process of the SRM Neuron"). Neither is membrane potential a Gaussian process, because of the skewness induced by relative hyperpolarization. Relative hyperpolarization also deforms the distribution of postsynaptic latencies, in some cases resulting in a bimodal distribution.

To answer the question of how small the spatial pattern can be, I used a very simple experimental approach: a basic genetic algorithm to optimize neuron and STDP parameters in training for a spatial pattern in a simulation. The size of the pattern, i.e., the number of synapses containing the pattern, was gradually decreased from what was a relatively easy task down to the point training failed. The parameters were optimized for each size of pattern. Parameters were allowed to evolve freely, without restrictions. Such optimization was made for two different setups: in the first setup afferents participating in the pattern fired at a rate of 64 Hz and others fired at 39 Hz; in the second setup all afferents fired

at 64Hz. Also, I conducted limited experiments with firing rates at 39 Hz/39 Hz and 25 Hz/39 Hz.

The results were quite unexpected. In the case of the triplet update, the pattern was successfully scaled down to a single synapse without significant degradation in performance over the entire range of patterns when the setup was 64 Hz/39 Hz. In the case of the single synapse transmitting a periodic spike, obviously no spatial pattern remained, and the neuron was tuned to detect either differing rates in the incoming synapse or, it seems, the periodic occurrence of the spike. The genetic algorithm immediately changed the polarity of the long-term depression of the third spike coefficient. Also, in the case of the 64 Hz/64 Hz setup, the polarities of both third spike coefficients were changed (see Materials and Methods). In the case of all-to-all interaction I did achieve a single synapse as well, but the training success rate was significantly reduced. The simple nearest-neighbor interaction rule reached a definite limit on pattern size and could not be optimized further.

3.5.1. Materials and Methods

The neuron was trained for spatial patterns of different sizes. Neuron and STDP parameters were optimized for each size of pattern. Simulations were executed in discrete time steps at 1 ms precision. The pattern was created by a number of selected input neurons firing at the same time after each 40 ms of simulation (Figure 3.25). Also, all neurons fired spontaneously with a fixed probability at each simulation step, thus producing Poisson noise. During pattern exposure all other neurons were inhibited from firing. Three different spike neighborhood rules were compared: all-to-all, nearest-neighbor with immediate pairings (Burkitt et al., 2004) and nearest-neighbor with triplet update (Pfister & Gerstner, 2006).

This experiment is similar to the one conducted by Masquelier and colleagues (Masquelier et al., 2008). The key difference is that I used a spatial rather than a

spatiotemporal pattern, and the patterns were inserted into the noise signal at a regular frame rate.

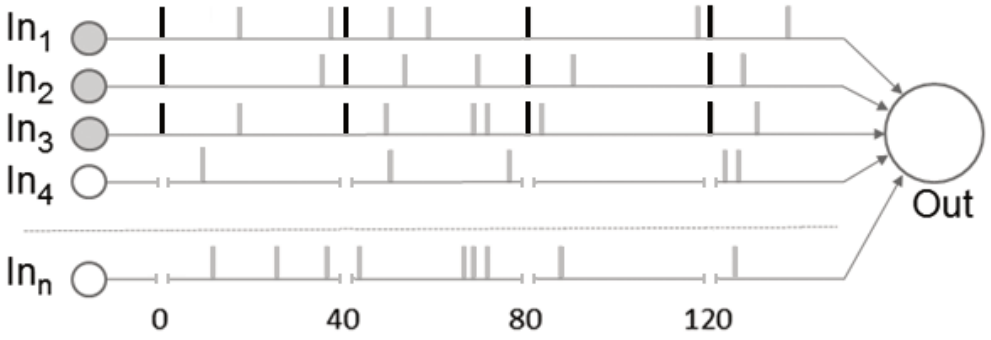

Figure 3.25. Sample pattern. Spikes generated by Poisson process are grey, injected spatial pattern is black. Afferents which do not belong to the pattern did not fire during pattern exposure.

I measured training performance in three different setups of Poisson noise. Initially I set the probability of a noisy spike to p=0.04 in all afferents. In this setup, the neurons which participated in the pattern fired more frequently than those which did not, at 64 Hz and 39 Hz respectively. In the case of a spatiotemporal pattern of sufficient duration it is possible to eliminate this difference in firing rate, if the count of spikes in the pattern from individual synapses is equal to the expected value of Poisson noise. This is not the case, however, for spatial patterns: in order to maintain equal firing rates, synapses which belong to the pattern must fire less frequently during Poisson periods, thus reducing noise and, presumably, resulting in easier training. On the other hand, the difference in firing rates also has a great influence on the training: the heuristic optimization I used could tune the neuron with the triplet rule to detect increased firing rate instead of spatial pattern. This was the case when the pattern was sufficiently small, but not the case when it was large enough to cause the postsynaptic spike (see Results). Such a mix of spatial and rate coding is compatible to some extent with observations of the auditory cortex of primates (Kayser et al., 2009), where a mix of different coding systems seems to convey more information than spatial or rate coding alone.

Later I reran the optimization with a 64 Hz/64 Hz setup, where the firing rate of afferents not participating in the pattern recognition were increased to 64 Hz by setting the probability of a noisy spike to p=0.065641026. Here the triplet rule also performed better than the other two, but the rate of success was lower, and the behavior of the neuron when the pattern became small was very different (see Results).

I also conducted limited experiments with reduced noise in the afferents which did participate in the pattern at 39Hz/39Hz and 25Hz/39Hz. Here the probabilities of a noisy spike were set to 0.014358974 and 0 respectively. In this case the triplet rule lost its advantage over all-to-all, but still performed significantly better than the nearest-neighbor rule.

Heuristic Optimization

The basic idea was to discover the lower limits of SDTP training with respect to spatial pattern, or, in other words, to answer the question of how small the spatial pattern could be. Since additive STDP tends to produce a bimodal distribution of synaptic strengths, the idea was to maximize the difference between the strengths of synapses which transmit the pattern and those which do not. In addition, the neuron must remain responsive at the end of the training, and ideally only selective to the pattern. Instead of minimizing the firing latencies of the trained neuron, for the sake of simplicity I made the assumption that the neuron firing rate should be approximately the same as the rate of pattern injection. For this purpose I introduced a Gaussian component into the objective function:

$$f = \begin{cases} \sum_t e^{-(\lambda-\xi)^2/\delta}\Delta\mu_w & \text{if } \Delta\mu_w > 0 \\ \sum_t \Delta\mu_w & \text{if } \Delta\mu_w \leq 0, \end{cases} \tag{3.32}$$

where $\Delta\mu_w$ is the observed difference between the means of strengths of synapses which were associated to the pattern and those which were not; λ is the observed

firing rate (times per second); $\xi = 25$ is the target firing rate and $\delta = 20$ defines the tolerated deviation from the target rate. At the beginning of the training all synaptic strengths were set to the same value ω_0, so that at the very beginning of the training the value of $\Delta\mu_w$ was zero. The value of the objective function was the sum of observations at each time step in the simulation. In this way, the performance of the training was taken into account from the very beginning of the simulation, and in this manner, the speed of the training was also increased by maximizing the objective function.

The heuristic search to maximize the objective function (3.32) was executed in 7-dimensional space for nearest-neighbor and all-to-all rules, and in 11-dimensional space for the triplet rule. The optimized parameters were: ϑ, w_0, w_{min}, α, A_{pre}, T_{pre}, and T_{post}. For the triplet rule there were four additional parameters: A_{pre3}, A_{post3}, T_{pre3} and T_{post3} (see equations below). For the heuristic search I used a very basic genetic algorithm. There were 100 agents, and after each trial 60 agents were replaced by the offspring of the top 20 performers. Offspring were generated from the parent agent by applying normally distributed mutations. The mean of the normal distribution was the parent value; the standard deviation for mutations was 1 for time dimensions (T_{pre}, T_{pre3}, T_{post} and T_{post3}) and 0.01 for all other dimensions. Each agent in each generation executed 10 independent trials and the values of the objective function were added together from all 10 trials. The heuristic search was executed for 1,500 generations. There were 300 afferents in each agent.

In my first 64 Hz/39 Hz experiment initial pattern size n was set to 24, equal to twice the expected spike count generated by Poisson noise. Pattern size was then decreased to 12, 8, 4, 2 and 1.

In the 64 Hz setup, pattern sizes were 19, 15, 12, 8 and 4. Pattern size n=19 approximates the expected spike count generated by Poisson noise, which was 19.2. Here for the initial conditions I reused the parameters obtained for n=24 in the 64 Hz/39 Hz setup, with an exception for the triplet rule (see Results).

For the 39 Hz/39 Hz and 25 Hz/39 Hz experiments, initial parameters were taken from the 64 Hz/39 Hz setup n=8 results (an exception was again made for the triplet rule, see Results for details) and the pattern size remained the same, n=8.

After optimization of parameters, the success rate of training was evaluated by training the neuron for the same pattern 1,000 times. The success criteria was $\Delta\mu_w >= 0.3$ and $12 < \lambda < 50$ at the end of the training (see Eq. 1).

I reran the genetic optimization several times and the results were all similar.

The Neuron Model

Neurons were modeled on a simplified version of the Spike Response Model (SRM) (Gerstner & Kistler, 2002) identical to the one described in section 3.2 "STDP Learning under Variable Noise Levels".

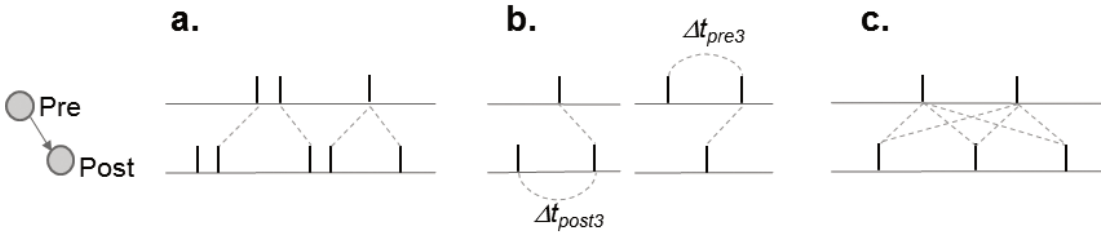

Figure 3.26. STDP neighborhood rules. **a:** Nearest-neighbor with immediate pairings. **b:** Triplet interaction. **c:** All-to-all.

During all experiments time constants were set to T_r=10, T_m=10 and T_s=0.5. W_r was dependent on a threshold value and was set to 2ϑ.

In the neuron model presented in this section only one STDP rule of excitatory-to-excitatory synapses is applied. Neighborhood functions used for comparison are represented in Figure 3.26. Triplet update Figure 3.26b was used in combination with the nearest-neighbor interaction in Figure 3.26a. STDP updates were modeled by equations (3.33), (3.34) and (3.35).

Nearest-neighbor:

$$\Delta w_j = \begin{cases} \alpha e^{\frac{-\Delta t}{T_{post}}} & \text{if } \Delta t > 0 \\ -\alpha A_{pre} e^{\frac{-\Delta t}{T_{pre}}} & \text{if } \Delta t < 0 \\ 0 & \text{if } \Delta t = 0 \end{cases} \tag{3.33}$$

Nearest-neighbor with triplet update:

$$\Delta w_j = \begin{cases} \alpha \left(1 + A_{post3} e^{\frac{-\Delta t_{post3}}{T_{post3}}}\right) e^{\frac{-\Delta t}{T_{post}}} & \text{if } \Delta t > 0 \\ -\alpha \left(A_{pre} + A_{pre3} e^{\frac{-\Delta t_{pre3}}{T_{pre3}}}\right) e^{\frac{-\Delta t}{T_{pre}}} & \text{if } \Delta t < 0 \\ 0 & \text{if } \Delta t = 0 \end{cases} \tag{3.34}$$

All-to-all:

$$\Delta w_j = \begin{cases} \alpha e^{\frac{-\Delta t}{T_{post}}} (1 + y_{post}) & \text{if } \Delta t > 0 \\ -\alpha A_{pre} e^{\frac{-\Delta t}{T_{pre}}} (1 + y_{pre}) & \text{if } \Delta t < 0 \\ 0 & \text{if } \Delta t = 0 \end{cases} \tag{3.35}$$

where Δw_j is the amount of change of strength of the individual synapse; α is the training step; Δt is the time difference between postsynaptic and presynaptic spikes; and A_{pre}, A_{pre3}, A_{post3}, T_{pre}, T_{pre3}, T_{post} and T_{post3} are the parameters which control the amplitudes and slopes of STDP functions. Variables y_{post} and y_{pre} were computed the same way as the x variables in the neuron model in equations (3.3) and (3.4), the only difference being that in this case weights were not present.

3.5.2. Results

The results from the 64 Hz/39 Hz and 64Hz experiments are presented in Figure 3.27. The triplet rule in both setups performed much better than its competitors, although in the 64 Hz triplet experiment there was a significant degradation in performance for n=8. The simple nearest-neighbor rule performed the worst, and heuristic search failed to find suitable parameters for the n=4 spike pattern in the 64 Hz/39 Hz setup and for the n=12 in the 64 Hz setup.

It has to be said that in the 64 Hz setup (Figure 3.27b) the genetic algorithm initially failed to find the point where the triplet rule would perform better than all-to-all, and when the pattern size was 15, it performed worse than nearest-neighbor. This could not be the global optimum, because nearest-neighbor is a special case of the triplet, where A_{pre3} and A_{post3} are zero, therefore at the optimal point the triplet rule should perform at least equally to nearest-neighbor. So here optimization stuck in a local optimum. To validate this, I used the nearest-neighbor parameters obtained for n=15 as initial parameters for the triplet, except A_{pre3} and A_{post3} were set to zero, $T_{pre3}= 2T_{pre}$ and $T_{post3}= 2T_{post}$. The results were significantly better: the triplet performed better than the other two. In order to eliminate possible unfair competition, I reran genetic optimization for nearest-neighbor and all-to-all for 3,000 generations, with no success in improving the parameters. Although these results cannot be conclusive, it strongly suggests that the triplet rule can perform better.

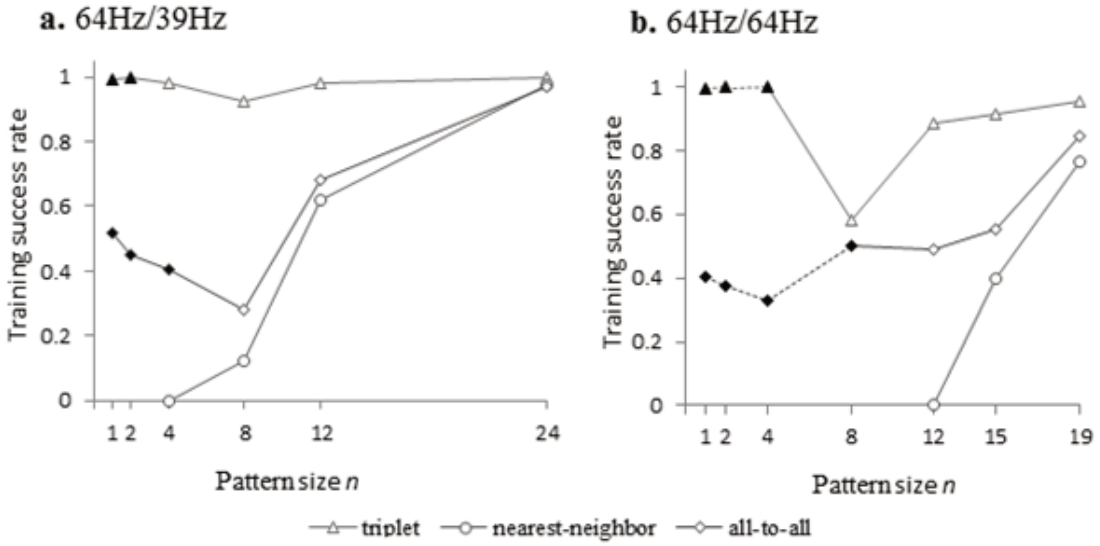

Figure 3.27. Training success rate vs. pattern size. Black markers denote the training when synaptic strengths were bimodal but the neuron was not selective to the pattern. **a:** Results from the 64 Hz/39 Hz setup. **b:** Results from the 64 Hz setup. Dashed lines indicate that there was no heuristic optimization done and previous optimized parameters were simply reused.

When pattern size was relatively large, results from both the 64 Hz/39 Hz and 64 Hz experiments were quite similar: the trained neuron was selective to the pattern and fired mostly after the pattern time with 2-millisecond latency. The

latency was caused by the PSP kernel function chosen, see Eq. (3.2). In the 64 Hz/39 Hz setup the triplet rule retained selectivity down to the n=4 pattern and in the case of all-to-all this was n=8 (see Figure 3.28); in the 64 Hz setup selectivity was lost sooner: the triplet rule retained selectivity down to n=8, and n=12 for the all-to-all rule.

When the pattern became too small, the genetic algorithm found conditions where STDP training would result in certain equilibria of synaptic strengths, and consequently the neuron firing rate was more or less constant, but even so, synaptic strengths associated to the pattern tended to grow close to the maximal value, which was 1, while the remaining strengths were distributed above the minimal value. The neuron was not selective because the combined strength of the synapses associated to the pattern was not sufficient to cause a postsynaptic spike, especially when n=1, at which a spatial pattern does not even exist. In the case of the simple nearest-neighbor rule, this kind of behavior was not observed.

It must be said that in the 64 Hz setup genetic optimization could not improve the training success rate for the triplet and all-to-all, as the pattern became too small and neuron was not selective to it. In Figure 3.27b the dotted line indicates that optimization there was discontinued, and for measuring success rate, parameters were taken from previous optimization results, which was n=4 for the triplet, and n=8 for the all-to-all.

When the pattern is relatively small (see Figure 3.27, black markers), training with the parameters obtained from the 64 Hz/39 Hz and 64 Hz experiments for the triplet rule results in very different behaviors. Parameters from the 64 Hz/39 Hz experiment were tuned to detect an increased rate: I replaced the spatial pattern with a pure Poisson process with firing rates of 64 Hz and 39 Hz respectively, and repeated the triplet experiment with the same parameters. Training failed when n>4, and thus a coincidence of spikes was required to train the neuron under the given parameters. Training was successful, however, when $n\leq4$, and therefore when the pattern was small, synapses grew stronger because

of increased input firing rate, not because of coincidences of input spikes. This was not the case, however, with the 64 Hz experiment. I made a few tests with the triplet rule and n=1 pattern size. In my experiment the spatial pattern consists of spikes and gaps. When spikes and gaps are replaced with pure 64 Hz Poisson noise, training obviously fails; the success rate is simply equal to the measured probability for a random synapse to grow stronger than the mean value plus 0.3 (see "Heuristic Optimization" for training success criteria). When only gaps were replaced with Poisson noise, the training success was reduced from 0.99 to 0.7. When only spikes in the pattern were replaced with noise, but gaps were persistent, the training success rate was reduced to 0.14, which seems to be slightly above random chance (the measured probability of a random chance of success was ~0.12). This suggests that parameters were tuned to detect deformations of a Poisson process, and these deformations could be induced by either a periodic spike or periodic gap. I cannot claim with certainty, however, that STDP can detect periodic gaps in a Poisson process.

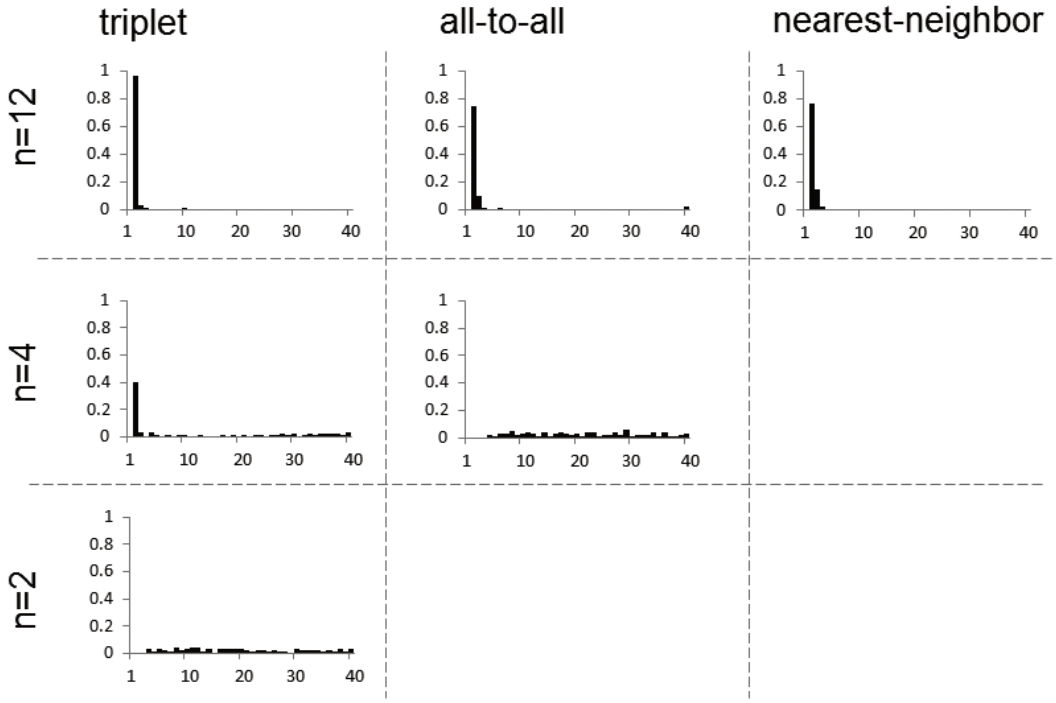

Figure 3.28. Distribution of trained neuron response latencies. Zero point is the sample pattern occurrence time. Results are from the 64 Hz/39 Hz setup.

I conducted an additional test with the triplet rule in the 64 Hz setup: I replaced the spatial pattern with a spatiotemporal one, by distributing the spikes of a pattern in time with 10 ms latency; gaps from noisy afferents were removed. In this case, training was successful for n=4 with a success rate of ~0.5 and for n=8 of ~0.2. Training failed for n=12. This indicates that parameters obtained for relatively small patterns were appropriate for detecting certain deformations of a Poisson process of spikes from an individual afferent, but not coincidences of spikes.

In the all-to-all rule and the small pattern in the 64 Hz/39 Hz setup where the STDP window was inverted (see Figure 3.27a, black markers; Table 3.4), training failed when the pattern was replaced with pure noise. I tried to preserve only the gaps and only the periodic spike, with no success in training. In the 64 Hz setup, however, removing gaps and replacing the spatial with a spatiotemporal pattern with 10 ms latency between spikes boosted the performance of the all-to-all rule to a success rate above 0.8 for n<=12. It should be noted that when n=8 and n=12 the spatiotemporal pattern overlaps, so there were accordingly coincidences of two or three spikes. Training failed when n=15.

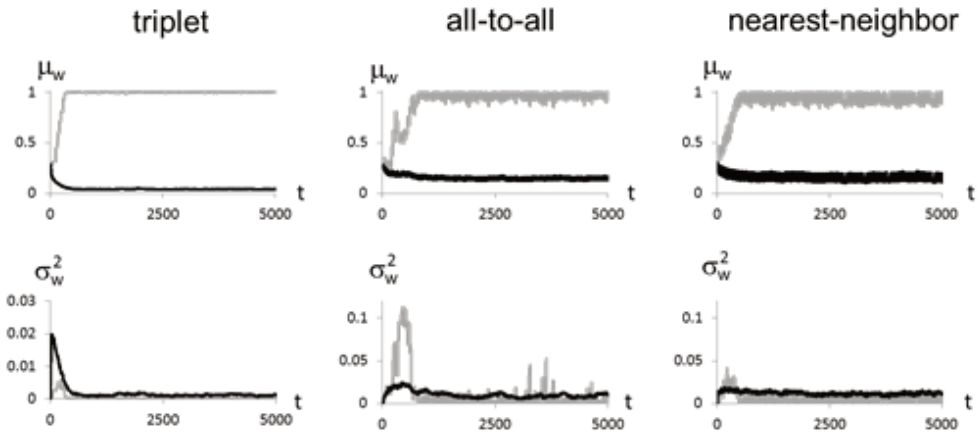

Figure 3.29. Evolution of synaptic strengths. Results were taken from individual successful trainings when pattern size was n=12 for the 64 Hz/39 Hz setup. The top row is mean synaptic strengths; the bottom row shows variances. Grey denotes synapses associated to the pattern, black denotes synapses not associated to the pattern.

The nearest-neighbor rule in all cases could only be trained to the spatial pattern, and all attempts to replace spatial pattern with Poisson noise or a spatiotemporal pattern resulted in training failure.

It is worth noting that the number of input neurons easily could be scaled up by any factor, by scaling the threshold value ϑ and the size of the spatial pattern by the same factor and keeping STDP parameters unchanged except for training step α, which required additional tuning in the case of the 64 Hz/39 Hz setup and n=1 (in this particular case, I had to change the value of α from 0.662 to 0.9, otherwise training was unsuccessful). This nonlinear dependency of the training step needs further research. In the case of a small pattern in non-selective mode, the pattern size may remain unchanged after scaling. For the 64 Hz/39 Hz setup I successfully scaled the triplet model up by a factor of 20, which is 6,000 inputs with pattern sizes of n=1 and n=8. The neuron was trained thus without any noticeable degradation in the success rate of the training. In fact, I observed a slight improvement. That is, the neuron was able to find a single synapse with increased firing rate among 5,999 others, and to learn a spatial pattern made by 160 input neurons among 5,840 others. I also successfully repeated the same scaling by a factor of 20 with the 64 Hz setup, for n=1 and n=19.

Figure 3.29 is a comparison of the synaptic strength evolutions of successful trainings where pattern size was n=12 for the 64 Hz/39 Hz setup. Results were gathered from single runs of 5,000 ms duration. The three columns represent the three rules, and the upper row shows means of synaptic strengths, the bottom variances. In the case of the triplet rule, synaptic strengths were much more stable. When training with the triplet rule, variance of strengths was almost one-tenth that of nearest-neighbor or all-to-all. Comparing the all-to-all rule to the simple nearest-neighbor rule, synaptic strengths were a bit more stable in the case of all-to-all. Strength evolutions in the 64 Hz setup were very similar.

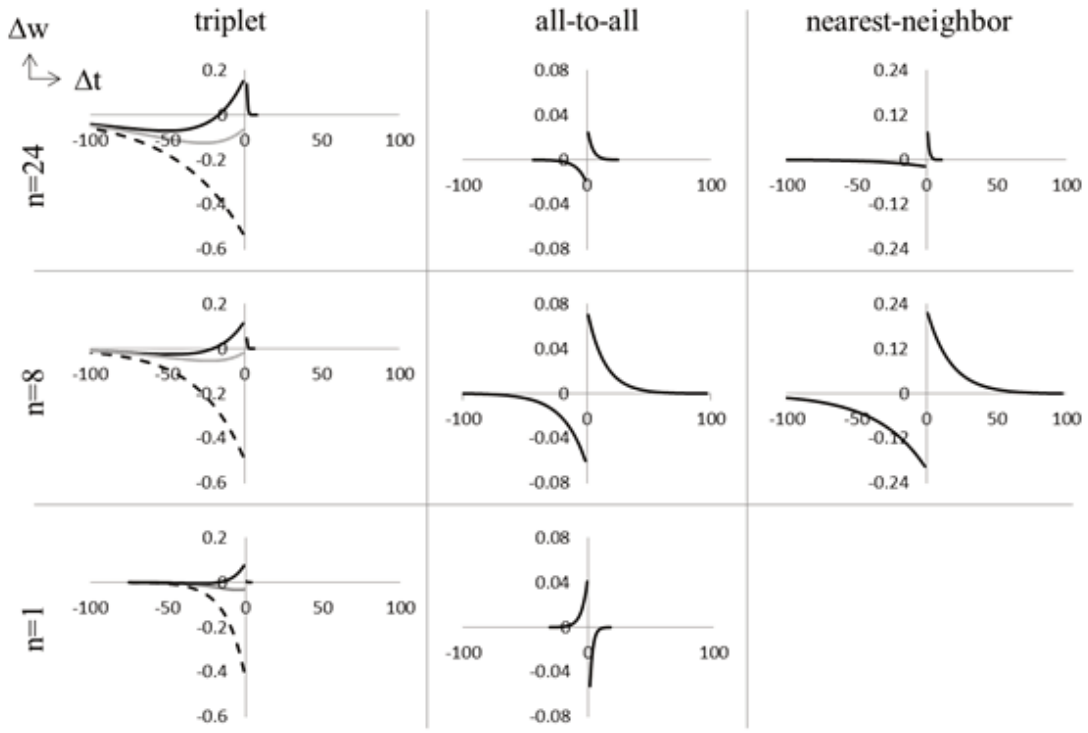

Figure 3.30. Optimized STDP windows for the 64 Hz/39 Hz setup. In the case of the triplet rule, black lines denote $\Delta t_{pre3}=\Delta t+1$, grey lines $\Delta t_{pre3}=\Delta t+25$ and dashed lines $\Delta t_{pre3}=\infty$.

From the parameters obtained through heuristic optimization (Figure 3.30; Table 3.4), we can make a few interesting observations. In the case of the triplet rule, LTP occurred at the left side of the STDP window (Figure 3.31), that is, where $t_{pre}>t_{post}$, and presynaptic spikes were closely correlated to postsynaptic ones. The right side of the STDP window shows a very steep slope, and its amplitude diminishes as the pattern size becomes smaller; in the case of pattern size n=1, the right side of the STDP window shows little or no influence.

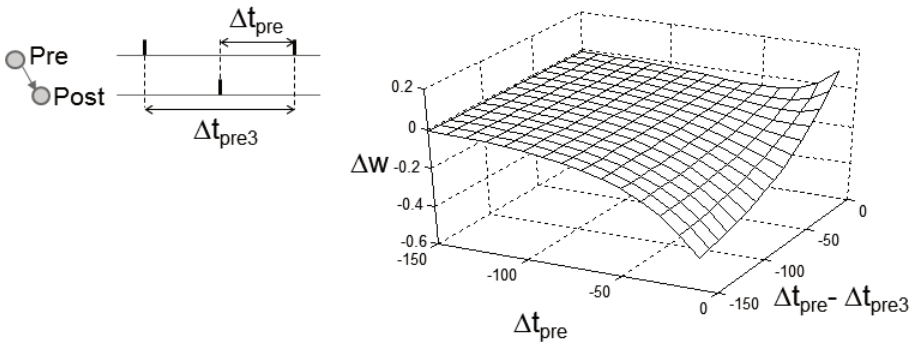

Figure 3.31. The left side of the STDP triplet update window. Values obtained by genetic optimization when pattern size was n=12, 64 Hz/39 Hz setup.

In the 64 Hz setup LTP also occurred at the left side of STDP window, but the value of this LTP was significantly lower (see Figure 3.32). Also, compared to the 64 Hz/39 Hz experiment, the right side of the STDP window was not diminished, and LTD occurred on the right side when presynaptic and postsynaptic spikes were close in time.

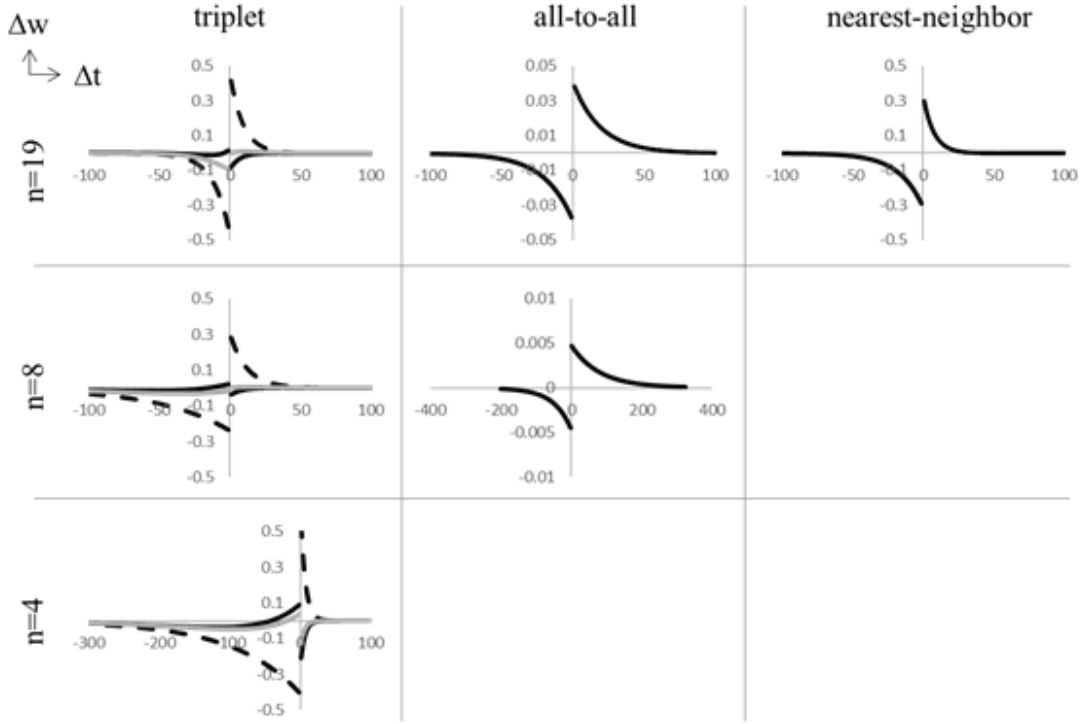

Figure 3.32. Optimized STDP windows for the 64 Hz setup. In the case of the triplet rule, black lines denote $\Delta t_{pre3}=\Delta t+1$, grey lines $\Delta t_{pre3}=\Delta t+25$ and dashed lines $\Delta t_{pre3}=\infty$.

Also, in the case of all-to-all rule and 64 Hz/39 Hz setup, LTP and LTD switched places when the pattern became small at n=4, and, at the same time, the neuron lost its selectivity to the pattern (see Figure 3.28). It is interesting to note that switches of LTP and LTD in the synapses of the same neuron have been observed in biology: synapses distant from the soma have different STDP window polarity than synapses proximate to the soma. This has been observed in the visual cortex (Sjostrom at al., 2006) and in the barrel cortex (Letzkus et al., 2006). Another interesting observation about the all-to-all rule is that, in the case of a small pattern, the neuron fires at a persistent rate, despite the w_{min} value approaching close to zero. This indicates that such an inverted STDP window is capable of attaining equilibria in synaptic strengths when exposed to Poisson

noise, whereas in the case of a non-inverted STDP window, non-correlated input spikes tend to minimize synaptic strengths. It was also interesting to note that the behavior of the inverted STDP window for all-to-all interaction contradicted the equilibrium properties predicted by the Izhikevich and Desai equation (Izikevich & Desay, 2003): equilibria for parameters found by heuristic optimization should not be stable. The Izikevich and Desay equations, however, are based on the assumption of a Poisson-distributed postsynaptic spike train, which was not the case for an SRM neuron with relative refraction. At this time, I have no good explanation why the neuron retained a stable firing rate. This requires additional research.

In the 64 Hz setup, the all-to-all rule did not switch the polarity of LTP and LTD (see Figure 3.32). The behavior, however, was somewhat similar: the all-to-all rule attained equilibria in synaptic strengths and the postsynaptic neuron fired at a persistent rate.

I conducted a limited experiment with the 39 Hz/39 Hz and 25 Hz/39 Hz setups, where noise in afferents participating in the pattern produced reduced Poisson noise or no noise at all (see Figure 3.33 and Table 3.6).

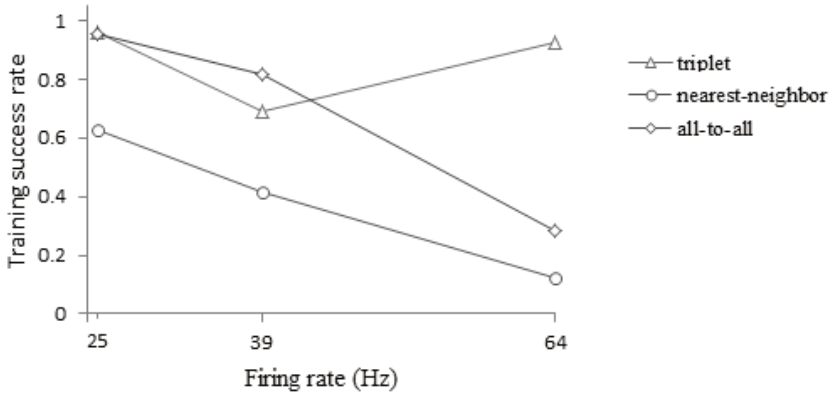

Figure 3.33. Training success rate vs. firing rate. The firing rate of all eight afferents participating in the pattern was reduced from 64 Hz to 39 Hz and 25 Hz. Other afferents fired at 39 Hz. Success rate values at 64 Hz are taken from Figure 3.27a.

In this experiment, the all-to-all and nearest-neighbor rules resulted in an increased success rate as the noise of the afferents participating in the pattern was reduced, but the success rate of the triplet rule decreased at the point of 39

Hz; the triplet rule performed worse than the all-to-all rule, but better than nearest-neighbor.

Initially in this experiment I took the optimized parameters from the 64 Hz/39 Hz and n=8 results as the initial conditions for all three rules and evolved parameters with altered firing rates, but maintaining n=8. Later, in the same way as in the 64 Hz/64 Hz setup, I reused optimized parameters from the nearest-neighbor for initial conditions for the triplet. This helped improve the performance of the triplet, but not to the point where it could perform better than all-to-all at 39 Hz. At 25 Hz triplet and all-to-all had a very similar success rate.

Table 3.4. Optimized parameters for the 64 Hz/ 39 Hz setup.

	n	T_{post}	T_{pre}	α	A_{pre}	ϑ	w_{min}	W_0	T_{post3}	T_{pre3}	A_{post3}	A_{pre3}
Triplet	1	0.21	12.25	0.662	0.600	21.69	0.104	0.273	59.91	92.49	0.134	-0.794
	2	0.28	11.20	0.715	0.602	24.64	0.117	0.319	58.07	90.55	0.051	-0.793
	4	0.31	17.55	0.718	0.652	14.23	0.087	0.377	62.42	89.44	0.122	-0.825
	8	0.36	29.61	0.748	0.666	8.88	0.003	0.282	71.90	94.94	0.082	-0.840
	12	0.59	34.97	0.610	0.671	13.76	0.023	0.275	67.78	70.72	-0.017	-0.887
	24	0.59	44.27	0.740	0.731	20.39	0.001	0.428	72.34	62.48	0.042	-0.973
All-to-all	1	2.34	4.58	-0.081	0.493	63.37	0.000	0.580				
	2	1.60	3.37	-0.090	0.451	65.29	0.015	0.594				
	4	1.72	3.32	-0.106	0.497	65.16	0.004	0.599				
	8	14.22	18.63	0.075	0.842	18.57	0.057	0.261				
	12	9.27	14.28	0.121	0.941	21.12	0.081	0.255				
	24	4.01	7.70	0.031	0.677	23.76	0.031	0.206				
Nearest	8	16.98	35.29	0.230	0.876	15.57	0.010	0.204				
	12	17.39	42.78	0.244	0.852	18.66	0.041	0.240				
	24	1.21	30.43	0.166	0.112	21.83	0.007	0.195				

Table 3.5. Optimized parameters for the 64 Hz/64 Hz setup.

	n	T_{post}	T_{pre}	α	A_{pre}	ϑ	w_{min}	W_0	T_{post3}	T_{pre3}	A_{post3}	A_{pre3}
Triplet	4	6.66	94.65	0.636	0.635	54.31	0.006	0.732	213.04	198.39	-1.392	-0.788
	8	11.99	49.93	0.308	0.773	21.25	0.007	0.794	197.47	157.43	-1.144	-0.851
	12	20.27	51.06	0.286	0.978	13.25	0.002	0.318	153.34	183.43	-1.081	-0.942
	15	17.83	49.78	0.305	0.693	18.02	0.002	0.344	105.88	96.65	-1.214	-0.703
	19	9.22	11.45	0.465	0.992	19.63	0.005	0.416	134.27	88.52	-1.219	-1.048
All-to-all	8	80.05	50.04	0.005	0.969	86.00	0.027	0.729				
	12	40.92	33.33	0.011	0.836	43.63	0.024	0.368				
	15	16.52	18.08	0.049	0.947	26.94	0.047	0.266				
	19	19.36	20.19	0.040	0.964	33.70	0.080	0.319				
Nearest	15	12.53	22.10	0.246	1.000	26.96	0.046	0.214				
	19	7.51	17.02	0.341	0.897	31.37	0.054	0.239				

Table 3.6. Optimized parameters for the 39 Hz/39 Hz and 25 Hz/39 Hz setups.

	f (Hz)	T_{post}	T_{pre}	α	A_{pre}	ϑ	w_{min}	W_0	T_{post3}	T_{pre3}	A_{post3}	A_{pre3}
Triplet	25	15.89	11.2	0.272	0.981	8.18	0.010	0.164	45.71	72.113	-1.183	0.393
	39	14.27	15.45	0.353	0.733	10.36	0.013	0.197	28.76	88.54	-1.428	0.389
All-to-all	25	29.73	28.08	0.018	0.935	12.07	0.051	0.159				
	39	48.60	37.59	0.006	0.999	12.97	0.056	0.169				
Nearest	25	13.83	27.55	0.154	0.989	8.75	0.016	0.134				
	39	8.19	19.84	0.232	0.912	11.90	0.030	0.178				

3.5.3. Discussion

The main purpose of this work is to demonstrate that changing the sign (±) of an additional trace variable of the triplet STDP implementation potentially can result in a far better coincidence detector than STDP implementations based on two trace variables. The triplet rule (Figure 3.26b) was originally suggested by Froemke and Dan (Froemke & Dan, 2002), based on *in vivo* experiments with pyramidal neurons in the visual cortex of the rat. Later Pfister and Gerstner

(Pfister & Gerstner, 2006) successfully reproduced STDP behavior found in biological neurons, both in the visual cortex (Sjostrom et al., 2001) and hippocampal culture (Wang et al., 2005). Pfister and Gerstner used positive values for A_{pre3} and A_{post3} (see section "3.5.1 Materials and Methods", equation (3.34)). In the case of training for spatial patterns, however, genetic optimization immediately changed the polarity of A_{pre3}; and in the 64 Hz setup, A_{post3} as well (see Table 3.4 and Table 3.5). Taking a closer look at the LTD side of the original triplet rule (Figure 3.26b), it is evident that positive A_{pre3} increases LTD in cases where the previous presynaptic spike was strongly correlated to the postsynaptic one, and therefore reduces the existing correlation. This feature, while it might be biologically plausible, has a negative impact on training for spatial patterns. If A_{pre3} is set to a negative value, the result is the opposite, and LTD is either lessened, or replaced by LTP. Moreover, this setup of the triplet rule favors spike triplets in a window of a specific duration, and therefore is suitable for selecting synapses with a higher spiking rate, because the higher the rate is, the higher the probability of the occurrence of a triplet in a smaller temporal window. Thus it was not surprising in the least that the heuristic search changed the polarity of A_{pre3}. What was surprising, however, was the magnitude of positive impact on training overall. At the same, I have no good explanation for why the genetic algorithm changed the polarity of A_{post3} in the case of the 64 Hz setup, and why only in this setup. Such negative A_{post3} would cause LTD when two postsynaptic spikes are close in time and the presynaptic spike is closely correlated to the last postsynaptic one (see Figure 3.32, triplet). At this time, I can only speculate that this LTD is induced mostly when the postsynaptic neuron fires frequently, thus helping prevent too high a firing rate.

Another interesting observation which follows from my experiment is that the triplet implementation of STDP can achieve stable equilibria of synaptic strengths when exposed to a Poisson process of input spikes. At the same time, STDP can detect an increased spiking rate or a certain deformation of a Poisson process even in a single synapse, when the influence of that individual synapse

on the overall postsynaptic membrane potential is negligible. In such case the neuron is incapable of encoding the data, which makes it difficult to apply this feature to competitive learning, for example, in a single winner-take-all circuit, such as the one used by Masquelier and colleagues (Masquelier et al., 2009). There is no reason why, however, synaptic weights cannot be modified after the training by increasing the contrast of synaptic weights, thus making the neuron selective to the input of even a single synapse.

When the pattern was relatively small (Figure 3.27, black markers), the neuron was incapable of detecting a spatial pattern, and so failed at coincidence detection, even in the case of the inverted all-to-all rule in the 64 Hz/39 Hz setup, since training was successful at n=1. Nevertheless, this demonstrates a variety of interesting properties of STDP learning which require additional research. It is important to understand that STDP training may increase synaptic strength for multiple reasons. Without a good understanding of why and when synaptic strengths grow or decay, interpretation of STDP training results could be problematic.

Admittedly this work only covered a fraction of the many phenomenological models of STDP. Spike pairings in nearest-neighbor can be implemented differently by using symmetric or postsynaptic-centered interpretations (Morrison et al., 2008). I did not explore reduced multiplicative update (Gütig et al., 2003), nor did I include the all-to-all version of the triplet (Pfister & Gerstner, 2006).

In this work, optimized parameters were different under different training conditions. This would suggest that the parameters of the neuron must be tuned according to the properties of the input spike trains, and/or *vice versa*. From a practical point of view this makes little sense: if good prior knowledge about the data is required before applying STDP training, then one may use other traditional tools which are much more efficient than STDP. the results of this research suggest, however, that the triplet rule is a good candidate for use in

more sophisticated neural circuits, as presented in my dissertation in sections 3.2 “STDP Learning under Variable Noise Levels” and 3.3 “Competitive STDP Learning of Overlapping Spatial Patterns”.

This research should be treated with caution because the results of heuristic optimization are only approximate and there is no proof that heuristic optimization approached global optima, rather than getting stuck in local optima. While it shows that using the triplet rule makes very good results possible, it does not prove it is impossible to achieve better results with the all-to-all or nearest-neighbor rule. At this stage of research the amount of data is insufficient to draw solid conclusions other than that the triplet rule can perform extremely well under certain conditions. The results of this experiment should therefore be accepted as evidence, but not proof.

The biological plausibility of the triplet parameters discovered is questionable, but this experiment was not intended to validate biological hypotheses. The heuristic search discovered parameters appropriate for a mixture of the Poisson process and periodic spatial patterns. Such conditions do not necessarily exist in the biological realm. In the case of the Poisson process, for example, intervals between input spikes are distributed exponentially, while this is questionable in the case of the actual postsynaptic potential process (Linder, 2006). The Spike Response Model with relative refraction can’t even produce a Poisson spike train. The results of this experiment should nonetheless prove interesting from the perspective of machine learning.

Conclusions

Despite tremendous advances in neuroscience, we are far from a clear understanding of how neurons function and how learning and memory work. The major problem here is not a lack of data, but the notorious diversity and complexity of biological neural systems. Besides the gaps in our knowledge of the physiology of the neuron, for the moment there are simply too many data, too many hypotheses to fit into a comprehensive, unified and systematic theory of neurons, never mind a general theory of the brain. Put simply, the puzzle is too large, and too many of the pieces are still missing. Hopefully large ongoing projects such as the Blue Brain Project, the Human Brain Project (Markram, 2012) and the BRAIN Initiative (Markoff, 2013) aimed at systematizing our knowledge of biological neural systems will lead to greater general understanding.

My own research was limited to STDP learning, also not yet completely understood, with multiple non-linearities discovered and numerous phenomenological models proposed (Caporale & Dan, 2008; Morrison et. al, 2008). I investigated a rather hypothetical form of Hebbian learning with respect to the learning of spatial and spatiotemporal patterns. I identified a few of the problems associated with such learning, and found solutions to them.

I. It is possible to build a neural circuit for the learning of spatiotemporal patterns based on STDP learning only.

I have designed, implemented and tested a novel STDP-based neural circuit capable of learning long-lasting sequences of spatiotemporal patterns of spikes. This circuit is capable of reproducing memory to some extent.

II. It is possible to build a neural circuit for learning overlapping spatial patterns.

I have designed, implemented and tested a novel STDP-based neural circuit with distance-dependent synaptic strength factors, capable of learning and discriminating mutually inclusive spatial patterns. Moreover, this circuit is

capable of solving the problem of optimization of the neuron threshold value when exposed to patterns of different spatial size.

III. It is possible to use the inverted SDTP rule for compensating for variable background noise when using STDP learning in an environment with variable Poisson noise.

I have designed, implemented and tested an STDP-based neural circuit which includes adaptive vertical inhibition also based on STDP but with an inverted learning window. The experimental results indicate such an approach works well within a certain range of parameters, but requires more precise tuning as the intensity of background noise increases.

IV. The STDP triplet rule with inverted additional trace variables can result in far better training performance than traditional nearest-neighbor and all-to-all STDP implementations.

I have benchmarked three different phenomenological models of STDP. I have used the genetic algorithm to tune the parameters of the model in training for spatial patterns of different size. STDP with triplet interaction demonstrated far better results than I expected when compared to other rules for spike interaction.

References

Abbott LF, Nelson SB. 2000. Synaptic plasticity: taming the beast. Nat. Neurosci. 3:1178-1183

Abbott, L. F. and van Vreeswijk, C. (1993).Asynchronous states in a network of pulse-coupled oscillators. Phys. Rev. E, 48:1483-1490.

Abraham, W.C., and Bear, M.F. (1996). Metaplasticity: the plasticity of synaptic plasticity. Trends Neurosci 19, 126-130

Abraham WC. 2003. How long will long-term potentiation last? Philos Trans R Soc Lond B Biol Sci 358: 735–744.

Adrian ED & Zotterman Y. (1926). "The impulses produced by sensory nerve endings: Part II: The response of a single end organ.". J Physiol (Lond.) 61: 151–171.[1]

Arnold, B. C. & Beaver, R. J. (2000), Hidden truncation models. Sankhya, series A 62, 22–35.

Azzalini A., Dalla Valle A. (1996), The multivariate skew-normal distribution, Biometrika 83 715-726.

Azzalini, A. (2005). The skew-normal distribution and related multivariate families. Scand. J. Statist., 32, 159–200.

Bailey, C.H., Giustetto, M., Huang, Y.Y., Hawkins, R.D., Kandel, E.R. (2000) Is heterosynaptic modulation essential for stabilizing hebbian plasiticity and memory. Nature Reviews Neuroscience, 1:1, 11-20

Bienenstock, E. L., Cooper, L. N., &Munro, P.W. (1982).Theory for the development of neuron selectivity: Orientation speci□city and binocular interaction in visual cortex. J. Neurosci., 2, 32–48.

Bell CC, Han VZ, Sugawara Y, Grant K. 1997. Synaptic plasticity in a cerebellum-like structure depends on temporal order. *Nature* 387:278–81

Benke, T. A., Lüthi, A., Isaac, J. T., & Collingridge, G. L. (1998). Modulation of AMPA receptor unitary conductance by synaptic activity. Nature, 393(6687), 793-797.

Bi GQ, Poo MM. 1998. Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *J. Neurosci.* 18:10464–72

Bi GQ, Poo MM. 2001. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annu.Rev. Neurosci.* 24:139–66

Boettiger CA, Doupe AJ. 2001. Developmentally restricted synaptic plasticity in a songbird nucleus required for song learning. *Neuron* 31:809–18

Burkitt AN, Meffin H, Grayden DB (2004) Spike-timing-dependent plasticity: the relationship to rate-based learning for models with weight dynamics determined by a stable fixed point. Neural Comput 16:885–940

Butts DA, Weng C, Jin J et al. (September 2007). "Temporal precision in the neural code and the timescales of natural vision". Nature 449 (7158): 92–5. Bibcode:2007Natur.449...92B. doi:10.1038/nature06105. PMID 17805296.

Caporale, N., & Dan, Y. (2008). Spike timing-dependent plasticity: a Hebbian learning rule. Annu. Rev. Neurosci., 31, 25 – 46.

Câteau, H., & Fukai, T. (2003). A stochastic method to predict the consequence of arbitrary forms of spike-timing-dependent plasticity. Neural Computation, 15(3), 597-620.

Cateau, H., & Reyes, A. D. (2006). Relation between single neuron and population spiking statistics and effects on network activity. Physical review letters, 96(5), 058101.

Cardin, J. A. et al. (2009) Driving fast-spiking cells induces gamma rhythm and controls sensory responses. Nature 459, 663–667.

Carpenter, GA., & Grossberg, S. (2009). Adaptive Resonance Theory. CAS/CNS Technical Report 2009–008. Boston, USA: Boston University Press.

Cassenaer S, Laurent G. 2007. Hebbian STDP in mushroom bodies facilitates the synchronous flow of olfactory information in locusts. *Nature* 448:709–13

Davis, RE., & Stretton, AOW. (1989). Signalling properties of Ascaris motor neurons: graded synaptic transmission and tonic transmitter release. J. Neurosci., 9, 415 – 425.

Dayan, Peter; Abbott, L. F. (2001). Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. Massachusetts Institute of Technology Press. ISBN 978-0-262-04199-7.

Debanne D, Gahwiler BH, Thompson SM. 1998. Long-term synaptic plasticity between pairs of individual CA3 pyramidal cells in rat hippocampal slice cultures. *J. Physiol.*507(Pt. 1):237–47

Egger V, Feldmeyer D, Sakmann B. 1999. Coincidence detection and changes of synaptic efficacy in spiny stellate neurons in rat barrel cortex. *Nat. Neurosci.* 2:1098–105

Elman, J. L. (1998). Rethinking innateness: A connectionist perspective on development (Vol. 10). MIT press.

Ermentrout, G. B. (1996). Type i membranes, phase resetting curves, and synchrony. Neural Comput., 8:979-1001.

Feng, J. (2001). Is the integrate-and-fire model good enough - a review. Neural Comput., 13:xx.

Forrest MD (May 2014). "Can the Thermodynamic Hodgkin–Huxley Model of Voltage-Dependent Conductance Extrapolate for Temperature?". Computation 2 (2): 47–60.

Froemke RC, Dan Y. 2002. Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature* 416:433–38

Froemke RC, Poo MM, Dan Y. 2005. Spike-timing-dependent synaptic plasticity depends on dendritic location. *Nature* 434:221–25

Gardner, EP. (2010). Touch. In: Encyclopedia of Life Sciences (ELS). John Wiley & Sons, Ltd: Chichester.DOI: 10.1002/9780470015902.a0000219.pub2

Gautrais, J., & Thorpe, S. (1998). Rate coding versus temporal order coding: a theoretical approach. Biosystems, 48(1), 57-65.

George Jr, A. L. (2005). Inherited disorders of voltage-gated sodium channels.*Journal of Clinical Investigation*, *115*(8), 1990.

Gerstner, W., Kempter R., van Hemmen J.L., and Wagner H. (1996). A neuronal learning rule for sub-millisecond temporal coding. Nature, 386:76-78.

Gerstner, W., & Kistler, WM. (2002). Spiking neuron models. Cambridge: Cambridge UP

Gilson, M., Masquelier, T., & Hugues, E. (2011). STDP allows fast rate-modulated coding with Poisson-like spike trains. PLoS Comput Biol, 7(10), e1002231.

Grossberg, S. (2013). Adaptive resonance theory. Scholarpedia, 8(5), 1569.

Gupta A, Long LN (2007) Character recognition using spiking neural networks. IJCNN, pages 53–58

Gütig R, Aharonov R, Rotter S, Sompolinsky H (2003) Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity. J Neurosci 23(9):3697–3714

Guyonneau R, VanRullen R, Thorpe SJ (2004) Temporal codes and sparse representations: a key to understanding rapid processing in the visual system. Journal of Physiology-Paris, 98(4), 487-497.

Guyonneau R, VanRullen R, Thorpe SJ (2005) Neurons tune to the earliest spikes through STDP. Neural Comput 17: 859–879.

Havenith MN, Yu S, Biederlack J, Chen NH, Singer W, Nikolić D (June 2011). "Synchrony makes neurons fire in sequence, and stimulus properties determine who is ahead". J. Neurosci. 31 (23): 8570–84. doi:10.1523/JNEUROSCI.2817-10.2011. PMID 21653861.

Hindmarsh J. L., and Rose R. M. (1984) A model of neuronal bursting using three coupled first order differential equations. Proc. R. Soc. London, Ser. B 221:87–102.

Izhikevich E, Desai N (2003) Relating stdp to bcm. Neural Comput., 15, 1511–1523.

Jolivet R., Rauch A., Luscher H.R., and Gerstner, W. (2006), Predicting spike timing of neocortical pyramidal neurons by simple threshold models. Journal of Computational Neuroscience 21:35-49

Haas JS, Nowotny T, Abarbanel HD. 2006. Spike-timing-dependent plasticity of inhibitory synapses in the entorhinal cortex. J. Neurophysiol. 96:3305–13

Hansel, D. and Mato, G. (2001). Existence and stability of persistent states in large neuronal networks. Phys. Rev. Lett., 86:4175-4178.

Hebb, D. O. (2005). *The organization of behavior: A neuropsychological theory*. Psychology Press.

Hegde, J., & Van Essen, DC. (2000). Selectivity for complex shapes in primate visual area V2. J Neurosci, 20, RC61(1 – 6).

Hodgkin, AL., Huxley, AF. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. Journal Physiology, 117, 500 – 544.

Holmgren CD, Zilberter Y. 2001. Coincident spiking activity induces long-term changes in inhibition of neocortical pyramidal cells. J. Neurosci. 21:8270–77

Fellous, J. M., Tiesinga, P. H., Thomas, P. J., & Sejnowski, T. J. (2004). Discovering spike patterns in neuronal responses. The Journal of Neuroscience, 24(12), 2989-3001.

FitzHugh, R. (1961). Impulses and physiological states in models of nerve membrane. Biophys. J., 1:445-466.

Fries P, Nikolić D, Singer W (July 2007). "The gamma cycle". Trends Neurosci. 30 (7): 309–16. doi:10.1016/j.tins.2007.05.005. PMID 17555828.

Kayser, C., Montemurro, MA., Logothetis, NK., & Panzeri, S. (2009). Spike-phase coding boosts and stabilizes information carried by spatial and temporal spike patterns. Neuron, 61, 597 – 608.

Kandel, E. R., Schwartz, J. H., & Jessell, T. M. (Eds.). (2000). *Principles of neural science* (Vol. 4, pp. 1227-1246). New York: McGraw-Hill.

Kasabov, N. (2012). Evolving spiking neural networks and neurogenetic systems for spatio-and spectro-temporal data modelling and pattern recognition. In Advances in Computational Intelligence (pp. 234-260). Springer Berlin Heidelberg.

Kasabov N, Dhoble K, Nuntalid N, Indiveri G (2013) Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. Neural Netw. 41: 188-201.

Kempter, R., Gerstner, W., and van Hemmen, J. L. (1999). Hebbian learning and spiking neurons. Phys. Rev. E, 59:4498-4514.

Kėvelaitis, E., Illert, M., & Hultborn, H. (2006). Žmogaus fiziologija. *Kauno medicinos universiteto leidykla, Kaunas*.

Krunglevicius, D. (2011). Neural Processing of Long Lasting Sequences of Temporal Codes-Model of Artificial Neural Network based on a Spike Timing-dependant Learning Rule. IJCCI (NCTA), 196-204. SciTePress, (2011).

Krunglevicius, D. (2014). STDP Learning Under Variable Noise Levels. IJCCI (NCTA), 165-171. SciTePress, (2014).

Krunglevicius, D. (2016). Modified STDP Triplet Rule Significantly Increases Neuron Training Stability in the Learning of Spatial Patterns. Advances in Artificial Neural Systems, 2016, 1.

Krunglevicius, D. (2015). Competitive STDP Learning of Overlapping Spatial Patterns. Neural Comput, 27(8):1673-85.

Lampe, P. D., & Lau, A. F. (2004). The effects of connexin phosphorylation on gap junctional communication. *The international journal of biochemistry & cell biology*, *36*(7), 1171-1186.

Latham, P. E., Richmond, B. J., Nelson, P. G., and Nirenberg, S. (2000). Intrinsic dynamics in neuronal networks. I. Theory. J. Neurophysiol., 83:808-827.

Lehmann-Horn, F., & Jurkat-Rott, K. (1999). Voltage-gated ion channels and hereditary disease. *Physiological reviews*, *79*(4), 1317-1372.

Letzkus JJ, Kampa BM, Stuart GJ. 2006. Learning rules for spike timing-dependent plasticity depend on dendritic synapse location. J. Neurosci. 26:10420–29

Lindner, B. (2006). Superposition of many independent spike trains is generally not a Poisson process. Physical Review E, 73(2), 022901.

Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. Neural networks, 10(9), 1659-1671.

Magee JC, Johnston D. 1997. A synaptically controlled, associative signal for Hebbian plasticity in hippocampal neurons. *Science* 275:209–13

Malinow, R., & Malenka, R. C. (2002). AMPA receptor trafficking and synaptic plasticity. Annual review of neuroscience, 25(1), 103-126.

Markoff J. 2013. Obama seeking to boost study of human brain. New York Times 17.

Markram H. 2012. The human brain project. Scientific American 306.6: 50-55.

Markram H, Lubke J, Frotscher M, Sakmann B. 1997. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* 275:213–15.

Markram, H. and Sakmann, B. (1995). Action potentials propagating back into dendrites triggers changes in efficacy. Soc. Neurosci. Abs. 21.

Masquelier, T., Guyonneau, R., Thorpe, S. J. (2008). Spike timing dependent plasticity finds the start of repeating patterns in continuous spike trains. PLoSONE, 3(1), e1377.

Masquelier T., Guyonneau R., Thorpe S.J. (2009) Competitive STDP-based spike pattern learning. Neural Comput 21:1259–1276.

Montemurro MA, Rasch MJ, Murayama Y, Logothetis NK, Panzeri S (March 2008). "Phase-of-firing coding of natural visual stimuli in primary visual cortex". Curr. Biol. 18 (5): 375–80. doi:10.1016/j.cub.2008.02.023. PMID 18328702.

Morris, Catherine; Lecar, Harold (July 1981), "Voltage Oscillations in the barnacle giant muscle fiber", Biophys J. 35 (1): 193–213

Morrison, A., Diesmann, M., & Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike timing. Biological cybernetics, 98(6), 459-478.

Nagumo, J. S., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. Proc. IRE, 50:2061-2070.

Nessler B, Pfeiffer M, Maass M (2009). STDP enables spiking neurons to detect hidden causes of their inputs. Proceedings of NIPS Advances in Neural Information Processing Systems (Vancouver: MIT Press).

Nishiyama M, Hong K, Mikoshiba K, PooMM,Kato K. 2000. Calcium stores regulate the polarity and input specificity of synaptic modification. *Nature* 408:584–88

Pakdaman, K (2010). "Fluid limit theorems for stochastic hybrid systems with applications to neuron models" . Adv.Appl.Proba 43.

Pasley, B. N., David, S. V., Mesgarani, N., Flinker, A., Shamma, S. A., Crone, N. E., ... & Chang, E. F. (2012). Reconstructing speech from human auditory cortex. PLoS biology, 10(1), e1001251.

Pfister, JP., & Gerstner, W. (2006). Triplets of spikes in a model of spike timing-dependent plasticity. J Neurosci., 26, 9673 – 9682.

Prut, Y., Vaadia, E., Bergman, H., Haalman, I., Slovin, H., & Abeles, M. (1998). Spatiotemporal structure of cortical activity: properties and behavioral relevance. Journal of neurophysiology, 79(6), 2857-2874.

Purves, D., Augustine, G.J., Fitzpatrick, D., Hall, W.C., LaMantia, A.S., White, L.E. (2012). Synaptic Plasticity. In Neuroscience (5th ed.) (pp. 163-182). Sunderland, Massachusetts: Sinauer Associates.

Rolls, E. T., Aggelopoulos, N. C., Franco, L., & Treves, A. (2004). Information encoding in the inferior temporal visual cortex: contributions of the firing rates and the correlations between the firing of neurons. Biological Cybernetics, 90(1), 19-32.

Rubin J, Lee D, Sompolinsky H (2001) Equilibrium properties of temporally asymmetric Hebbian plasticity. Phys Rev Lett 86:364–367

Schmidt-Nielsen, K (1997). Animal Physiology: Adaptation and Environment (5th ed.). Cambridge: Cambridge University Press. ISBN 978-0-521-57098-5. LCCN 96039295. OCLC 35744403.

Senn, W., Tsodyks, M., and Markram, H. (1997). An algorithm for synaptic modification based on exact timing of pre- and post-synaptic action potentials. In Gerstner, W., Germond, A., Hasler, M., and Nicoud, J.-D., editors, Artificial Neural Networks - ICANN '97, pages 121-126. Springer.

Senn, W., Markram, H., & Tsodyks, M. (2001). An algorithm for modifying neurotransmitter release probability based on pre-and postsynaptic spike timing. Neural Computation, 13(1), 35-67.

Shouval, H. Z., Bear, M. F., and Cooper, L. N. (2002). A unified theory of NMDA receptor-dependent bidirectional synaptic plasticity. Proc. Natl. Acad. Sci. USA, 99:10831-6.

Sjostrom, J., & Gerstner, W. (2010). Spike-timing dependent plasticity. Scholarpedia, 5(2):1362.

Sjostrom PJ, Hausser M. 2006. A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. Neuron 51:227–38

Sjostrom PJ, Turrigiano GG, Nelson SB. 2001. Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron* 32:1149–64

Song, S., Miller, K.D., and Abbott, L.F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. Nat Neurosci 3, 919-926.

Stein RB, Gossen ER, Jones KE (2005). "Neuronal variability: noise or part of the signal?". Nat. Rev. Neurosci. 6 (5): 389–97. doi:10.1038/nrn1668. PMID 15861181.

Stevens, C. F.; Wesseling, J. F. (1999). "Augmentation is a Potentiation of the Exocytotic Process". Neuron 22 (1): 139–146

Surmeier, D. J.; Foehring, R. (2004). "A mechanism for homeostatic plasticity". *Nature Neuroscience* **7** (7): 691

Sutton, R. S., & Barto, A. G. (1998). Reinforcement learning: An introduction(Vol. 1, No. 1). Cambridge: MIT press.

Szatmáry, B., & Izhikevich, E. M. (2010). Spike-timing theory of working memory. PLoS Comput Biol, 6(8), e1000879.

Thorpe, S. J. (1990). Spike arrival times: A highly efficient coding scheme for neural networks. Parallel processing in neural systems, 91-94.

Turrigiano, G. G.; Nelson, S. B. (2004). "Homeostatic plasticity in the developing nervous system". Nature Reviews Neuroscience 5 (2): 97–107

Tzounopoulos T, Kim Y, Oertel D, Trussell LO. 2004. Cell-specific, spike timing-dependent plasticities in the dorsal cochlear nucleus. *Nat. Neurosci.* 7:719–25

Tzounopoulos T, Rubio ME, Keen JE, Trussell LO. 2007. Coactivation of pre- and postsynaptic signaling mechanisms determines cell-specific spike-timing-dependent plasticity. *Neuron* 54:291–301

VanRullen, R., & Thorpe, S. J. (2001). Rate coding versus temporal order coding: what the retinal ganglion cells tell the visual cortex. Neural computation, 13(6), 1255-1283.

VanRullen, R., Guyonneau, R., & Thorpe, S. J. (2005). Spike times make sense. Trends in neurosciences, 28(1), 1-4.

Zhang LI, Tao HW, Holt CE, Harris WA, Poo M. 1998. A critical window for cooperation and competition among developing retinotectal synapses. *Nature* 395:37–44

Zucker, Robert S.; Regehr, WG (Mar 2002). "Short-term Synaptic Plasticity". Annual Review of Physiology 64: 355–405.

Wang HX, Gerkin RC, Nauen DW, Bi GQ (2005) Coactivation and timing dependent integration of synaptic potentiation and depression. Nat. Neurosci 8:187–193.

Weisskopf, M. G., & Nicoll, R. A. (1995). Presynaptic changes during mossy fibre LTP revealed by NMDA receptor-mediated synaptic responses. Nature, 376(6537), 256-259.

Wise RA (1996). "Addictive drugs and brain stimulation reward".*Annu. Rev. Neurosci.* **19**: 319–40.

Wittenberg GM, Wang SS. 2006. Malleability of spike-timing-dependent plasticity at the CA3-CA1 synapse. *J. Neurosci.* 26:6610–17

Woodin MA, Ganguly K, Poo MM. 2003. Coincident pre- and postsynaptic activity modifies GABAergic synapses by postsynaptic changes in Cl- transporter activity. Neuron 39:807–20

Dalius Krunglevičius

STDP LEARNING OF SPATIAL AND SPATIOTEMPORAL PATTERNS

Doctoral Dissertation

Physical Sciences (P000),

Informatics (09 P)

Editor Geoffrey Noah Vasil