

University of Tartu
Faculty of Mathematics and Computer Science
Institute of Computer Science
Computer Science curriculum

Taivo Pungas

**Implementing and investigating
a biologically realistic neuron
model**

Bachelor's thesis (9 ECTS)

Supervisors: Dr. Raul Vicente
Dr. Jaan Aru

Tartu 2015

Implementing and investigating a biologically realistic neuron model

Abstract:

Calcium-based single neuron models have been shown to elicit different modes of synaptic plasticity. In the present study one such model was implemented and its learning behaviour studied.

Behaviour of the implemented neuron agreed qualitatively with prior work in all regards except selectivity to correlation in input. The neuron was found to implement a linear filter responding linearly to partial presentations of learned patterns. Simulating probabilistic neurotransmitter release had an expected effect of de-correlating input and was found to improve the efficiency of information transfer. In the regimes explored, the neuron was found to be incapable of performing principal component analysis. The insensitivity of results to changes in parameters was mostly untested.

The neuron did not exhibit more advanced information processing capabilities in the tests conducted. However, the implemented neuron model is capable of meaningful information processing and forms a good basis for further research.

Keywords: single neuron model, numerical simulation, synaptic plasticity, unsupervised learning, probabilistic neurotransmitter release

Bioloogiliselt realistliku neuroni mudeli ehitamine ja uurimine

Lühikokkuvõte:

On näidatud, et kaltsiumiioonide töö põhinevad üksikneuroni mudelid võimaldavad erinevate sünaptilise plastilisuse vormide teket. Käesolevas töös koostati üks selline mudel ja uuriti mudeli õppimisvõimet.

Koostatud mudeli käitumine kordas kvalitatiivselt varasemate uurimuste tulemusi, välja arvatud korreleeritud sisendmustrite korral. Leiti, et neuron töötab lineaarse filtrina, kuna õpitud sisendmustrit vaid osaliselt nähes sõltub väljund lineaarselt õpitud mustri osakaalust. Tõenäosuslik virgatsaine vabanemine vähendas oodatult sisendmustrite korrelatsiooni ja parandas infoülekanne tõhusust. Peakomponentanalüüsiks neuron töös formuleeritud viisil võimeline ei olnud. Tulemuste sõltuvust parameetrite täpsetest väärtustest ei kontrollitud.

Neuroni võime keerulisemat infotöötlust teha ei leidnud tõestust. Hoolimata sellest on koostatud neuroni mudel võimeline kasulikuks infotöötluseks ja seega hea alus edasisteks uuringuteks.

Võtmesõnad: üksikneuroni mudel, numbriline simulatsioon, sünaptiline plastilisus, juhendamiseta õppimine, tõenäosuslik virgatsaine vabanemine

Acknowledgements

First and foremost, I would like to thank my supervisors Raul Vicente and Jaan Aru for providing feedback and ideas throughout the process, and for introducing me to the worlds of academic research and neuroscience. I am grateful to Kristjan Korjus, without whom I might not have discovered the exciting field of computational neuroscience. I would also like to acknowledge the indispensable contributions of Elena Sügis, Lauri Koobas, and Anti Ingel for reading and providing feedback on various drafts of this thesis.

Finally, I want to thank the Information Technology Foundation for Education for their financial support granted through the IT Academy scholarship program.

Contents

Acknowledgements	3
Introduction	6
1 Background and related work	8
1.1 Overview of relevant neuroscience	8
1.1.1 Neurobiology of the brain	8
1.1.2 Neurons as biological computation units	8
1.2 Prior work	10
1.2.1 Models of neural information integration	10
1.2.2 Models of synaptic plasticity	11
1.2.3 Probabilistic neurotransmitter release	11
1.2.4 Learning in single neuron models	11
2 Methods	13
2.1 Neuron model used in simulations	13
2.1.1 Input generation	14
2.1.2 Integrate-and-fire model	14
2.1.3 Model of synapses	15
2.1.4 Plasticity	16
2.2 Implementation details	18
3 Results	19

3.1	Validation of model behaviour	19
3.1.1	Input-output relationship	19
3.1.2	Spike timing-dependent plasticity	20
3.1.3	Metaplasticity	21
3.1.4	Selectivity to temporal correlation	21
3.2	Learning patterns of input rate	22
3.2.1	Response to learned inputs	23
3.2.2	Response to partial presentation of learned inputs	24
3.2.3	Selectivity to patterns with different channel widths	25
3.3	Probabilistic neurotransmitter release	26
3.4	Principal component analysis task	27
4	Discussion	30
4.1	Interpretation of results	30
4.2	Limitations	31
4.3	Future work	31
	Conclusion	33
	Bibliography	33
	Appendix A: MATLAB code	36
	Appendix B: Glossary	37
	Licence	38

Introduction

Your brain is a complex organ capable of very sophisticated thought. Even though the chess-playing supercomputer Deep Blue won the reigning world champion Garry Kasparov, Kasparov was also able to do anything human beings do every day, whereas the computer could only play chess. The human brain is remarkable not for its ability to perform very well at one particular task—be it playing chess, reading an article in *Nature*, preparing a six-course meal, or slam dunking in a basketball game—it is remarkable for being able to do all of those things, and much more.

On the other hand, gaining an understanding of the brain’s information processing mechanisms remains one of the major scientific challenges today. A key aspect of the brain is neural plasticity, its ability to learn over time as new knowledge and experience are accumulated and processed. Thus, neuronal plasticity, and specifically synaptic plasticity (change in the strength of neuronal connections) is thought to underlie our ability for learning and memory. To understand the dynamics of these neuronal changes, computational models are built to assess how well they explain the brain’s behaviour as observed in experiments in neuroscience and, more generally, how the models perform in a variety of tasks.

Artificial systems with significant capacity for learning have recently been created using deep convolutional neural networks, performing well in activities ranging from playing video games [MKS⁺15] to describing scenes in natural language [KFF14]. At the same time the neuron models used in these systems are very simple, often composing of a scalar product of the inputs and weights, and a thresholding function. In contrast, evidence from brains has shown that each neuron is a much more complex and powerful information processing unit. There is a clear gap between the learning rules used in networks of neurons, and biophysical mechanisms of learning at the cellular level. The primary goal of this thesis is to help narrow this gap by studying the behaviour of a more detailed, biologically realistic neuron model. To do so, a detailed implementation of such a model was built and some of its learning capabilities tested.

In particular, the approach taken in this thesis is using numerical simulations to describe the behaviour of the implemented neuron resulting from basic principles of neuronal biophysics. This work builds on top of a plasticity model (described

as a set of differential equations) previously published in [YSBC04] and which has been shown to produce a range of neural plasticity phenomena.

The model allows inspection of three aspects which are of particular interest. Firstly, the changing of neural plasticity–metaplasticity–is important for producing well-known learning phenomena. Secondly, probabilistic neurotransmitter release which causes signal transmission failure might have an effect of increasing information transfer efficiency. Thirdly, the neuron’s ability to perform principal component analysis (PCA), a well-known unsupervised learning algorithm, can be tested. This is motivated by prior work showing that very simple neuron models are capable of PCA [Oja08].

After a brief overview of the background and related work in Chapter 1 and describing the methods used in Chapter 2, Chapter 3 starts with validation of the behaviour of the implemented model against previously published work. Then, the neuron input-output relation is characterised for various kinds of input. As an addition to the model, the effect of probabilistic neurotransmitter release is studied. Finally, a simple task is simulated to study the neuron’s ability to implement PCA.

1 Background and related work

There are a few prerequisites to understanding the context of this thesis. The first section of this chapter will give a very short overview of aspects of neuroscience related to the thesis. In the second section prior work regarding the most important aspects of the model will be summarised.

1.1 Overview of relevant neuroscience

1.1.1 Neurobiology of the brain

The brain is the central information processing organ in animals. The computational properties of the brain arise from the heavily interconnected networks and sub-networks of specialised cells called *neurons*. It has been approximated that the human brain consists of 10^{11} neurons with 10^{15} inter-neuronal connections called *synapses* [HH09]. This work focuses on modelling one of many different types of neurons—pyramidal neurons.

1.1.2 Neurons as biological computation units

Every model of a brain that aspires for biophysical reality must include a way to model the behaviour of neurons and synapses. Each neuron can be seen as a small unit performing computation on some inputs and producing corresponding output. There are many computational models of neurons, with the level of detail ranging from complex multi-compartment models to very simple models such as those used in many artificial neural networks (ANNs).

Information flow through a neuron follows a path from the inputs to the output, with integration of information in between, as shown in Figure 1.1. A rough mapping of the corresponding neuron anatomy is shown in the same figure.

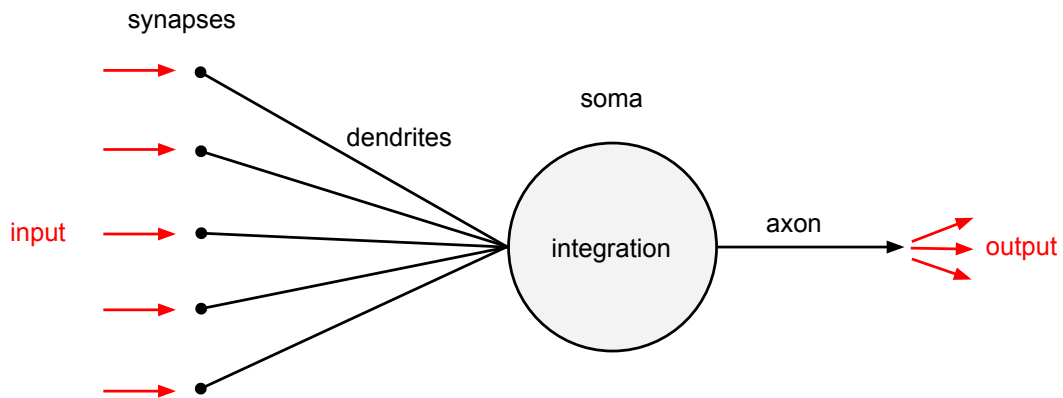


Figure 1.1: Schematic of a pyramidal neuron.

Input

At the synapse, information flows from the axon of the *presynaptic* neuron to a dendrite of the *postsynaptic* neuron. When the presynaptic neuron produces significant output, neurotransmitter molecules are released from the presynaptic neuron into the *synaptic cleft*, a small gap between the presynaptic and postsynaptic neurons at the synapse. Neurotransmitters then bind to receptors on the dendrite of the postsynaptic neuron causing ion channels to open. This causes ion flow across the membrane which causes a voltage change at the dendrite of the postsynaptic neuron. However, the neurotransmitter release is unreliable: upon some presynaptic spikes no neurotransmitter is released. This effect is called *probabilistic neurotransmitter release* (PR) or *synaptic failure*.

Synapses can be divided into two classes: input arriving at *excitatory* synapses tends to increase the activity of the postsynaptic neuron, input to *inhibitory* synapses tends to decrease postsynaptic activity.

Computation

A single neuron can be viewed as an integrator of information from its inputs over time, over different inputs, or both. At each synapse, a neuron has intrinsic biological components that determine how much this synapse should affect the output of the postsynaptic neuron. *Synaptic weights* are key parameters used in computational modelling of such biological components, and the modification of synaptic weights is the basis of learning in neurons and thus, in the brain.

The phenomenon of synaptic weights increasing due to learning is named long-term potentiation (LTP), and similar decreasing is named long-term depression (LTD). LTD and LTP are the methods the neuron uses to assign different synaptic

weights based on the long-term input to that synapse.

After the contribution of input from each synapse is known, this information must be integrated to produce the output of the neuron. An overview of some of the most common ways to model information integration in neurons will be given in Section 1.2.1.

Output

Output of a neuron takes the form of *action potentials* (also known as *spikes*), rapid increases in membrane voltage followed by a return to the equilibrium voltage. Information about action potentials is conducted from a neuron to the dendrites of other neurons using the axon. Axons are usually not part of single neuron models as the output of the modelled neuron is studied in and of itself.

1.2 Prior work

There are many computational models of neurons, with the level of detail ranging from complex multi-compartment models to very simple models such as those used in ANNs. Using these models, it is possible to simulate neurons in a computer and investigate the models' behaviour in a very controlled and detailed way. As opposed to real experiments that are hindered by many difficulties, using a simulation approach allows detailed investigation and control of inputs and all parameters.

Two aspects of neurons can be modelled and used in different combinations: the mechanism for translating input signals into output (information integration) and the mechanism for updating synaptic weights (synaptic plasticity); a summary of prior work in these will be given in the following two sections. Prior research into probabilistic neurotransmitter release will then be given followed by a short overview of the literature on learning in single neuron models.

1.2.1 Models of neural information integration

Some of the best known models, from least to most detailed, are the McCulloch-Pitts (MP) [MP43], integrate-and-fire (IF) [Abb99], and Hodgkin-Huxley (HH) [HH52] models. In the MP model which is widely utilised in ANNs, a weighted sum of inputs is calculated, to which a sigmoid function is applied to produce the output of the neuron. In the IF and HH models the neuron is modelled as an electric circuit, and then corresponding differential equations are used to calculate the output of the neuron.

1.2.2 Models of synaptic plasticity

Synaptic plasticity underpins learning in a neuron. It is possible to implement learning directly by calculating the error in the neuron’s output: the difference between expected output (the correct label) and the neuron’s prediction. Examples of these simple phenomenological rules include Hebbian rules (summarised as ”Cells that fire together, wire together”) [Heb52], Oja’s rule (shown to produce PCA computation regardless of its simplicity) [Oja08] and others, and are typically used in conjunction with MP neuron models. Learning rules can also be derived to minimise information-theoretic measures such as Fisher information [EG14].

Alternatively, learning can arise from biophysical properties of more detailed biological models of synapses. A summary of biophysical mechanisms of synaptic plasticity is given in [Sho07]; in Section 1.2.4, calcium-based models will be looked at in more detail.

1.2.3 Probabilistic neurotransmitter release

Recordings from hippocampal pyramidal neurons have shown synaptic transmission to be unreliable due to failure of release of neurotransmitter into the synaptic cleft upon a single presynaptic spike [SW95]. Furthermore, the probability of successful release p has been shown to have a wide distribution with predominant low values ($p < 0.4$) in the hippocampus [MSS97].

PR with $p < 1$ has been shown to have a beneficial effect of increasing the information-carrying efficiency of a single synapse, caused primarily by a decrease in noise [Gol04]. However, [GL12] conclude that for a population of neurons, encoding performance is increased in the presence of weak noise and decreased in the presence of strong noise, adding that ”several important synaptic parameters . . . have significant effects on the performance of the population rate coding”.

1.2.4 Learning in single neuron models

Plasticity in calcium-based models

An overview of calcium-based biophysical models of synaptic plasticity is given in [Sho07]. The model used in this work is a calcium-based model notable for exhibiting several synaptic plasticity phenomena within one model, providing a unified model of synaptic plasticity [SBC02]. Other unifying models have been proposed with different underlying mechanisms and differing results.

Building on top of the [SBC02] model, it was shown in [YSBC04] that adding a stabilisation mechanism (metaplasticity) synapses become stable, avoiding posi-

tive feedback loops that would result in very high synaptic weights. In addition, synaptic competition is achieved, allowing synaptic weights to "reflect the statistical properties of the inputs" [YSBC04]. This allows the neuron to learn complex input patterns while maintaining a relatively constant range of output.

The model used in this thesis will be discussed in more detail in Section 2.1.

Unsupervised learning

A special class of learning tasks—unsupervised learning—refers to learning hidden structure in unlabeled data.

Several single neuron models have been shown to be capable of unsupervised learning: simple artificial neurons can do PCA when trained according to the Oja rule [Oja08], the tempotron model [GS06] is able to learn spike timing-based patterns, the SKAN model [AGT⁺14] is capable of unsupervised classification of patterns. However, none of the aforementioned models are based on biophysical principles, and no other biophysical models have been shown to do unsupervised learning on the single neuron level.

Having now taken a look at the state of the art, the main goals of this work can be formulated as: (a) implementing the chosen neuron model and validating its behaviour, (b) exploring the learning behaviour of this model, (c) finding effects of probabilistic neurotransmitter release on behaviour, and (d) testing whether the neuron can perform PCA.

2 Methods

This chapter describes in detail the methods used for simulating the chosen neuron model in a computer. The first section summarises the most significant aspects of the neuron model used; the most important implementation details are given in the second section. More detailed descriptions of the performed simulations are given in Chapter 3.

2.1 Neuron model used in simulations

In this section, an overview of the most important aspects of the model used in simulations will be given. The neuron model follows the model of a single neuron published in [YSBC04]: an integrate-and-fire model of information integration is used with a calcium-dependent model of synaptic plasticity [SBC02]. In this model, synaptic weight changes are determined by the simulated concentration of Ca^{2+} ions, $[\text{Ca}]$.

The programmatic approach is to use update rules to change the simulated values of parameters (such as $[\text{Ca}]$), looping over time. Such rules are applied to inputs to calculate the values of intermediate parameters; applying relevant update rules to these intermediate parameters one obtains the output parameters such as postsynaptic voltage and timings of postsynaptic spikes. Due to the way information is represented in computers, time is discretised into time-steps of length dt regardless of the continuous nature of the biophysical processes.

Key aspects of the model are A) a rule to update the simulated value of $[\text{Ca}]$ as a function of presynaptic and postsynaptic activity, and B) a rule to calculate the change in synaptic weights as a function of $[\text{Ca}]$. Specifics of the $[\text{Ca}]$ update rule (A) are discussed in [SBC02]; an analysis of the synaptic weight update rule (B) is given in Section 2.1.4.

2.1.1 Input generation

Regardless of methods of integration and learning, it is necessary to create input for the neuron to learn. Rather than simulating presynaptic neurons, inputs to the model are simulated as N spike trains, one for each excitatory synapse, with a given mean firing rate r specified independently for each synapse. Each spike train contains one binary number per time-step: 1 if a spike occurred and 0 otherwise.

The spike trains are produced in a homogeneous Poissonian process, i.e. the occurrence of a spike is independent of the time since the last spike, and the average rate of spiking remains constant over time. Correlation between inputs is achieved by specifying the correlation coefficient $0 \leq c \leq 1$, and using code from [MBE⁺09] to generate correlated homogeneous Poissonian spike trains. For $c = 1$, all synapses receive identical input; for $c = 0$, all synapses receive independently generated spike trains.

In all simulations, inputs to inhibitory synapses are generated as uncorrelated Poissonian spike trains with a mean firing rate of 10Hz.

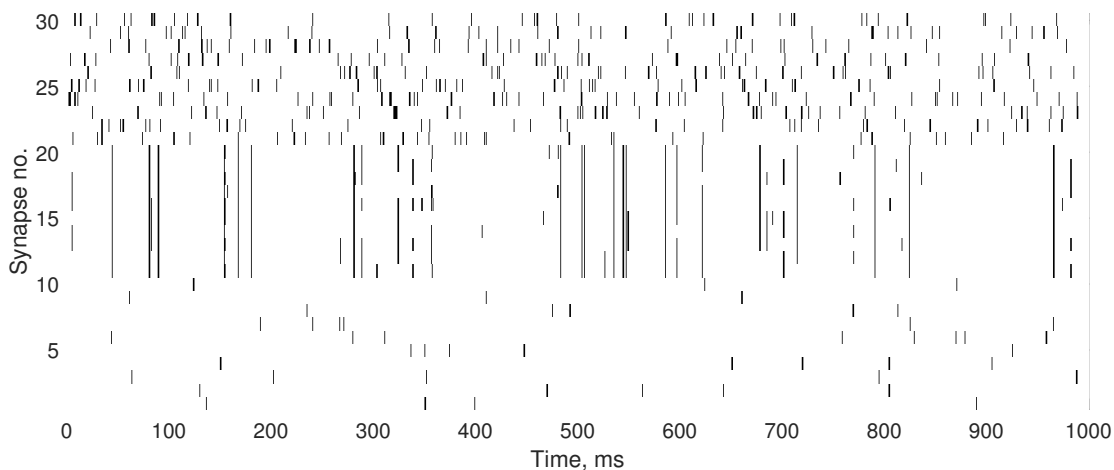


Figure 2.1: Sample input to 30 synapses, generated for a 1000ms period with a 0.1ms time-step. In this figure, black bars indicate timesteps at which spike trains (running horizontally) contain spikes. Synapses 1-10 receive 5Hz uncorrelated input, synapses 11-20 receive 40Hz correlated input with $c = 0.8$, synapses 21-30 receive 40Hz uncorrelated input.

2.1.2 Integrate-and-fire model

In this section, information integration methods of the model will be examined.

For modelling the postsynaptic neuron, equations and parameters from [EK98] are used. One parameter differs significantly in the model used in this thesis: the basal current I_0 , which causes spontaneous voltage increase, is set to 0. [EK98] utilises

a Traub-Miles cell, which is a Hodgkin-Huxley-like model for estimating postsynaptic voltage. Compared to Hodgkin-Huxley, the Traub-Miles model includes a more detailed simulation of Na^+ and K^+ currents by modelling the dependence of the gating of abovementioned channels on postsynaptic voltage.

The core of the integrate-and-fire model used in this thesis are five currents affecting postsynaptic voltage: a voltage-increasing current caused by presynaptic spikes at excitatory synapses, a voltage-decreasing current caused by presynaptic spikes at inhibitory synapses, Na^+ and K^+ currents not directly affected by presynaptic spiking, and a leak current, which tends to return voltage to the resting potential $V_{rest} = -65\text{mV}$ (membrane voltage of the neuron in absence of input). When postsynaptic voltage V_{post} reaches the threshold $V_{thres} = -55\text{mV}$, a postsynaptic spike is recorded, and V_{post} is set to the spiking voltage $V_{spike} = 40\text{mV}$. In the next time-step, V_{post} is returned to the reset potential $V_{reset} = -65\text{mV}$.

As an addition to the Traub-Miles model, spike-frequency adaptation is implemented following [YSBC04]: the resting voltage is decreased by 2mV upon each postsynaptic spike, and decays back to the baseline with a time constant of 100ms. This temporarily increases the gap between resting and threshold voltages, requiring more presynaptic input to reach the threshold and thus decreasing firing rate.

2.1.3 Model of synapses

This section contains a more detailed look into the way input spike trains are translated into postsynaptic voltage. The total number of synapses is 120: the neuron has 100 excitatory synapses and 20 inhibitory synapses.

Excitatory and inhibitory synapses are modelled by ion gates that are controlled by the respective neurotransmitter molecules: glutamate and gamma-aminobutyric acid (GABA). Upon each spike at an excitatory synapse, the opening of glutamate-controlled ion channels at dendrites is simulated, with the degree of opening simulated as a sum of two decaying exponentials peaking at the time of spike. The resulting ion flow through the channels tends to drive V_{post} towards the excitatory reversal potential $V_{rev,exc} = 0\text{mV}$. This makes the postsynaptic neuron more likely to produce a spike.

Similarly, upon each spike at an inhibitory synapse, the opening of GABA-controlled ion channels is simulated, and the resulting ion flow tends to drive V_{post} towards the inhibitory reversal potential $V_{rev,inh} = -65\text{mV}$. This makes the postsynaptic neuron less likely to produce a spike. Compared to inhibitory synapses, spikes at excitatory synapses have a larger relative influence on V_{post} due to the higher simulated conductivity (maximum throughput of ions) of glutamate-controlled ion channels.

The effect of presynaptic spikes on the gating of glutamate-controlled and GABA-controlled channels is simulated according to the model from [BEK08].

2.1.4 Plasticity

To facilitate learning, some aspect of the model needs to change: the model needs to be plastic. The plastic part of the model will be described in this section.

In addition to components mentioned in the previous section, the extent to which presynaptic spikes influence postsynaptic voltage is modelled by the synaptic weights $\mathbf{W} = (W_i)$, with i denoting the number of the synapse. The modification of these weights is the basis of learning in this model: increasing W_i will increase the effect spikes at the i -th synapse have on postsynaptic voltage, and vice versa. In this model, only excitatory synapses are plastic, i.e. the weights of inhibitory synapses remain unchanged.

The mechanism for modifying synaptic weights follows the Ca^{2+} -dependent model from [SBC02], where the aim was to construct a model based on a minimal number of assumptions. The model detects coincidences of presynaptic and postsynaptic spikes: the weight of synapse i will be increased if the presynaptic spike occurs right before the postsynaptic spike (i.e. there is reason to believe the postsynaptic spike was *caused* by the input), and decreased otherwise. The process of spike-timing-dependent plasticity (STDP) captures this idea that learning should depend on the exact timing on postsynaptic and presynaptic spikes. It was shown in [SBC02] that the plasticity model used in this thesis implements STDP in a way similar to what has been observed in experiments in neuroscience.

The value of $[\text{Ca}]$ is numerically estimated independently for each synapse allowing simultaneous potentiation at some synapses and depression at others. No dendritic distance is simulated, i.e. the possibility that some synapses are situated further from the soma is not taken into account.

The coincidence detection mechanism relies on N-methyl-D-aspartic acid (NMDA) receptors. NMDA receptors control gates that can allow calcium ions to flow into the neuron, which in turn causes an increase of the synaptic weight. The extent to which the calcium channels open depend on two factors: concentration of glutamate (the neurotransmitter that also causes voltage increases upon spikes at excitatory synapses) and dendritic voltage caused by a back-propagating action potential (BPAP). The former relays information about presynaptic spikes: if a spike has occurred recently, the level of glutamate at the synapse is at a high level. The latter signifies a postsynaptic spike: when the postsynaptic neuron fires, an increase in voltage is propagated to the dendrites. The influx of calcium is dependent on the exact timing of spikes because both glutamate concentration and BPAP voltage decay in time.

The synaptic weights' update depends on the level of intracellular calcium as shown in Equation 2.1, with shapes of Ω and η shown in Figure 2.2. λ is a regularisation parameter: it prevents weights from increasing indefinitely.

$$\frac{dW_i}{dt} = \eta([Ca]_i)(\Omega([Ca]_i) - \lambda W_i) \quad (2.1)$$

This mechanism allows the synaptic weights to increase when there has been a suitably timed pair of presynaptic and postsynaptic spikes. However, [SBC02] show that the same mechanism also accounts for decreasing and stable synaptic weights through the effect of Ω : at medium levels of Ca^{2+} , weights are decreased ($\Omega < 0.5$), and at low levels of Ca^{2+} , synaptic weights remain unchanged (Ω is constant at 0.5). The learning rate η also depends on calcium: higher levels of Ca^{2+} elicit larger changes in synaptic weights. λ , together with the value of Ω in the range of low calcium, determine the stable point of synaptic weights.

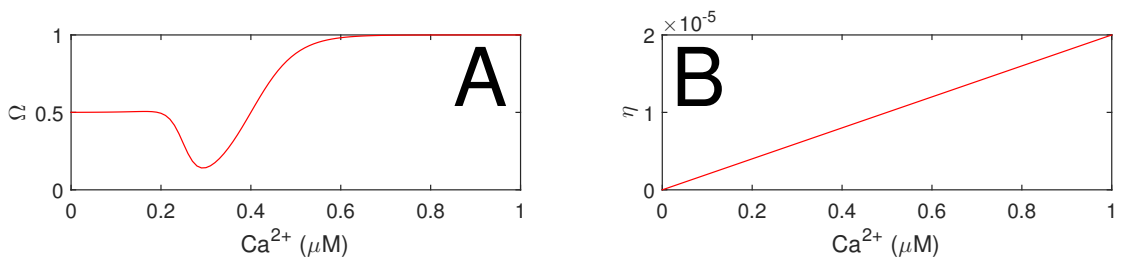


Figure 2.2: A. The Ω function. B. The learning rate function η .

Metaplasticity

Metaplasticity refers to plasticity of synaptic plasticity—the dependence of plasticity on the history of the previous activity of the neuron—which may play a role in LTP, LTD, and other learning mechanisms. In the scope of this thesis it refers to the changing of NMDA receptor (NMDAR) conductance as a function of output voltage, which leads to changes in the calcium influx and therefore changes in the sign and magnitude of changes in synaptic weights.

The metaplasticity mechanism follows the model of [YSBC04], where the NMDAR conductance g is the same for all synapses and is updated according to Equation 2.2:

$$\frac{dg}{dt} = -(k_-(V_{post} - V_{rest})^2 + k_+)g + k_+g_t \quad (2.2)$$

Here, g_t shows the total supply of NMDARs available in the internal pool of the neuron. k_+ and k_- are kinetic constants describing the speed of insertion and removal of NMDARs into synapses.

Metaplasticity can be disabled in any simulation, resulting in a constant value of g .

2.2 Implementation details

The neuron model of this thesis was implemented and data analysis conducted in MATLAB; source code is attached as an online supplementary to the thesis and described in Appendix A. The simulations were carried out in part in the High Performance Computing Center of University of Tartu.

For updating biological variables according to differential equations, the Euler method is used with a 0.1ms time-step length, chosen as the highest value allowing numerical stability. At the start of each simulation, some parameters are initialised to previously found equilibrium values to avoid unstable behaviour.

Input is generated in 1-second blocks for ease of implementation and the release of neurotransmitters caused by each presynaptic spike is assumed to last 1 ms. The computational complexity is linear with respect to the total simulation time; the computation time required for x seconds of simulated neuron time varies between $2x$ and $15x$ seconds on the machines used for simulation and depends on the amount of input to the neuron.

Unless mentioned otherwise, learning speed was increased by a factor of 100 by multiplying the values of learning rate η and NMDA receptor kinetic constants k_+ and k_- by 100. According to [YSBC04], the fixed points of the system do not change upon speeding up the simulations by this factor.

3 Results

In this chapter, the main results of this thesis will be presented. In accordance with the four goals of this thesis, the chapter is divided into four sections. The neuron's behaviour will be validated against prior work in the first section, learning behaviour will be characterised in the second, and in the last two chapters PR and PCA will be studied.

3.1 Validation of model behaviour

As a prerequisite to analysing more complex behaviour of the model, it is necessary to make sure the model is implemented correctly. The simplest way to do this is to observe the behaviour of the model under some conditions and compare the result with previously published results on the same model in the same conditions. As [SBC02] and [YSBC04] form the basis of the model, the behaviour of the model implemented in this thesis will be compared to these two sources. Attention will also be paid to keeping parameters and outputs in biophysically realistic ranges.

3.1.1 Input-output relationship

The input-output curve of a neuron is one of the most important characteristics, showing the output of the neuron to a specified amount of input. The exact magnitude of the curve is highly dependent on the specific parameters used. In particular, EPSP amplitude has a major effect on this relationship. Figure 3.1 shows the curve for six different neurons, each with a different EPSP amplitude. It is evident that the relationship is linear for a range of EPSP amplitudes, which is consistent with the findings of [YSBC04] for simulations with metaplasticity enabled. For low values of EPSP amplitude and input rate there is no output, indicating subthreshold activity: postsynaptic voltage never reaches the threshold required for producing a spike.

Increasing EPSP amplitude by a factor, the curve remains linear, but output rates are scaled up by a similar factor. This means the plausible EPSP voltage

range is not much constrained by considering the shape of the input-output curve, and EPSP voltage can be tuned to scale the output of the neuron according to other considerations. The EPSP amplitude producing the closest curve to that of [YSBC04] lies between 1.0mV and 2.0mV. However, another crucial consideration here is biophysical reality: for completely unstructured input (noise) at 30Hz, even a firing rate of 40Hz is excessive. For this reason, the problem of high firing rates was not given much weight and in further simulations, EPSP voltage was set to 3.0mV, a value found to produce higher selectivity to rate-based patterns.

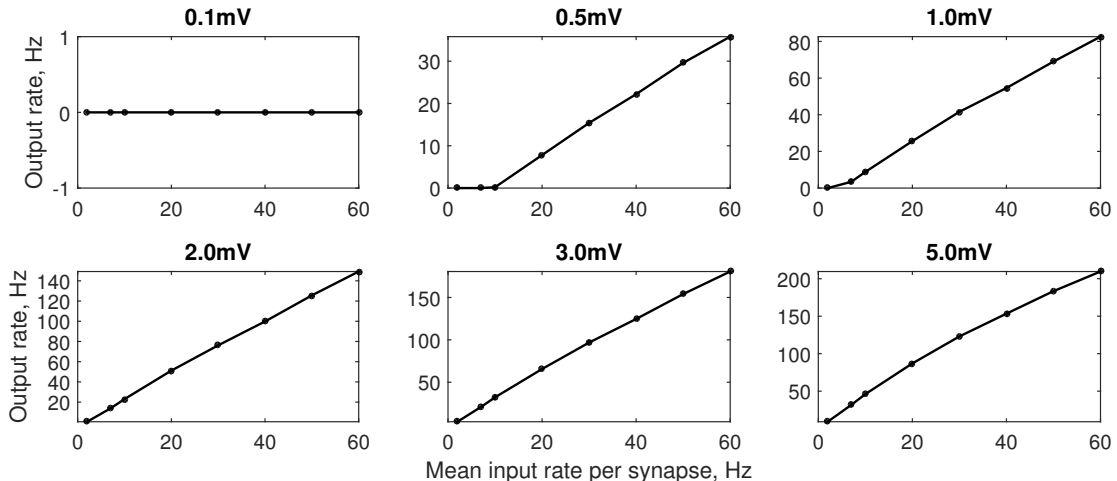


Figure 3.1: Input-output relationship of the neuron for different values of EPSP amplitude parameter (shown above each plot). Each data point in a plot is one experiment, in which input to the neuron consists of uncorrelated spike trains with mean input rate per synapse shown on the x-axis. Due to plasticity and metaplasticity, output rate varies until a fixed point is reached; for this reason, output rate is measured over the last 500 seconds in a 10000-second experiment.

3.1.2 Spike timing-dependent plasticity

The spike-timing-dependent plasticity curve is an important aspect of neuron models: it shows how the neuron determines the sign and magnitude of synaptic weight updates in response to the relative timing of pre- and postsynaptic spikes. In the model used here, two parameters significantly affect STDP curve shape: BPAP amplitude and the amount of glutamate released on the presynaptic side upon each spike. The shape of STDP curve using parameters from [YSBC04] is given in Figure 3.2A. For comparison, Figure 3.2B shows the same figure for a higher BPAP amplitude 100mV, which is the value used in [SBC02].

Both curves are qualitatively similar to [SBC02] with two regions of LTD induction—one at negative values of Δt and another in the region of $\Delta t > 100\text{ms}$ —and a single region of potentiation in between. However, for BPAP amplitude 100mV, the curve is smoother and weight differences larger, indicating that BPAP amplitude can be

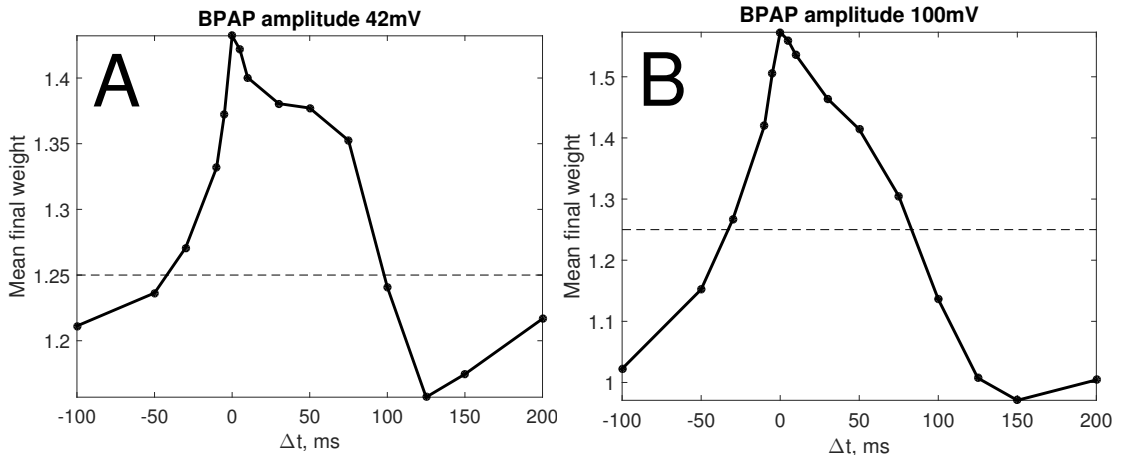


Figure 3.2: **A.** STDP curve for BPAP amplitude at 42mV. The relative timing of pre- and postsynaptic spikes $\Delta t = t_{post} - t_{pre}$ is plotted on the x-axis, and the weight resulting from a 100-second 1Hz stimulation of pre-post spike-pairs with the respective Δt is on the y-axis. The dashed line shows the value to which synaptic weights converge in the absence of postsynaptic spikes. **B.** STDP curve for BPAP amplitude at 100mV. Note the change in scale.

used for tuning STDP curve as necessary. In all following simulations, BPAP amplitude is fixed to 42mV to adhere to parameters published in [YSBC04].

3.1.3 Metaplasticity

The simplest effect of metaplasticity studied by [YSBC04] was a slow scaling down of weights after a fast potentiation, or scaling up of weights after a fast depression. In both cases, the fast change is induced by regular synaptic plasticity, and the subsequent scaling by metaplasticity. In Figure 3.3, these dynamics are visible: after an initial quick potentiation, weights are slowly scaled down to a value where they remain stable.

The model produces two more effects of metaplasticity observed in [YSBC04]. Firstly, it increases the variance in weights so rather than weights being nearly equal for all synapses, the distribution is wider. Secondly, after the observed initial quick increase in weights, they are scaled down to a stable value.

3.1.4 Selectivity to temporal correlation

Temporal correlation between spike trains, i.e. increased probability of coincidence of spikes, indicates underlying structure in the input. Thus, if the neuron is to learn this structure, in the presence of two groups of inputs with only one group correlated, the synapses of correlated inputs should be selectively potentiated. A

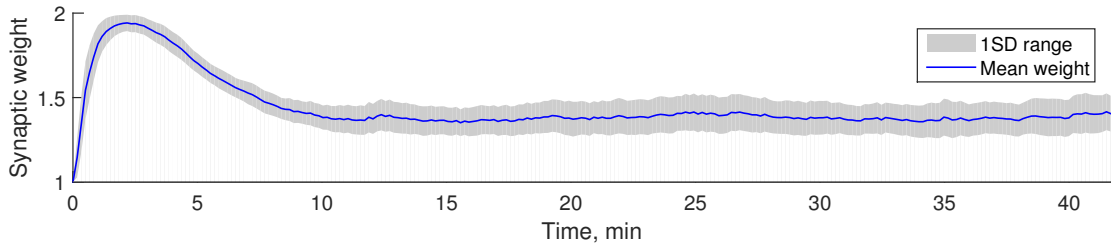


Figure 3.3: Evolution of synaptic weights in a metaplasticity experiment. For weights of excitatory synapses, both the mean (blue line) and ± 1 standard deviation range (grey area) are shown. Input to all synapses was uncorrelated with a mean rate of 20Hz.

strong effect of this sort is observed in [YSBC04], with weights of the uncorrelated channel going to zero. However, it can be seen in Figure 3.4 that in the model used in this thesis, selectivity to correlation is weak: the difference in weights of the two channels is existent but small with group mean weights differing by approximately 20%. A parameter sweep was conducted, but selectivity to correlation did not increase.

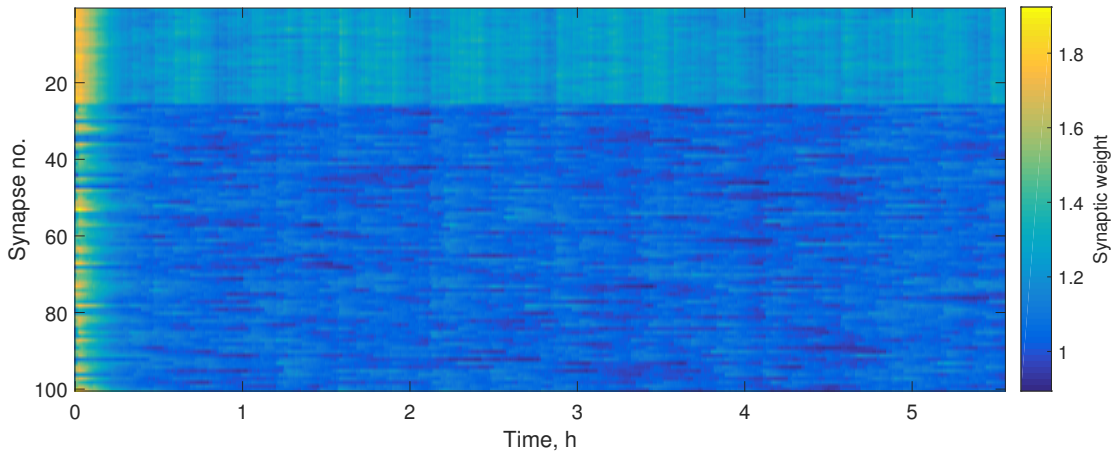


Figure 3.4: Evolution of synaptic weights over time in a selectivity experiment. Synapses 1-25 receive correlated input ($c = 0.8$) at 30Hz, synapses 26-100 receive uncorrelated input at 30Hz. The values of synaptic weights were sampled every 100ms.

3.2 Learning patterns of input rate

As shown in Section 3.1.1, the neuron's output rate is dependent on its input rate. In this section, the focus will be on the neuron's ability to learn and respond to patterns encoded in the mean firing rates of inputs. First, it will be shown that the neuron distinguishes between learned inputs and noise. Secondly, the neuron will be shown to be linear upon presenting learned input patterns partially. Finally, the effect of pattern width on learning will be studied.

A black-box approach will be taken: parameter values will be fixed and only input will be varied. The output will then be studied to make inferences about the neuron’s information processing capabilities.

3.2.1 Response to learned inputs

Before investigating the neuron’s ability to learn complex input patterns, it is necessary to ensure the neuron is capable of learning anything at all. In particular, it should be capable of something very simple: distinguishing between a learned pattern and random noise. To study this, a neuron was trained on a pattern (shown in Figure 3.5A) consisting of high-rate (40Hz) input to 25 synapses, and low-rate (10Hz) background input to all other synapses, both the pattern and the background uncorrelated.

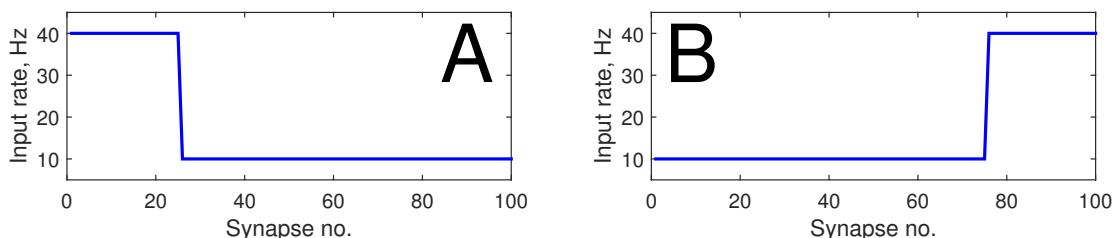


Figure 3.5: Two input rate patterns. A. Synapses 1-25 receive a higher rate. B. Synapses 76-100 receive a higher rate.

After training, plasticity and metaplasticity were disabled by fixing the weights, and two tests were run. In Test 1 the neuron received exactly the same input as in training. As control, in Test 2 the high-rate inputs were shown to a different, non-overlapping set of synapses, with all other synapses receiving background input at 10Hz (as shown in Figure 3.5B). The total amount of spikes received by the neuron in both tests remained constant. The resulting output rate in a 5-second simulation was 64Hz for Test 1 and 44Hz for Test 2, suggesting that the neuron fired more when it was presented with the learned input (compared to non-learned input).

It is also important to understand the neuron’s response when input varies between trained and untrained input. To this end, the input was alternated so that pattern 3.5A was shown in time periods $t \in [1s, 2s) \cup [3s, 4s) \cup \dots$. In between presentations of the trained pattern, all synapses received input at a rate of 17.5Hz (‘flat’ input), keeping the total input received by the neuron constant at all times. The resulting instantaneous output rate, shown in Figure 3.6, clearly oscillates between a high rate and a low rate. In addition, the mean firing rate in periods of learned input is significantly higher. Thus, this experiment demonstrates the neuron’s ability to switch between two modes of firing depending on whether it is presented with the learned input.

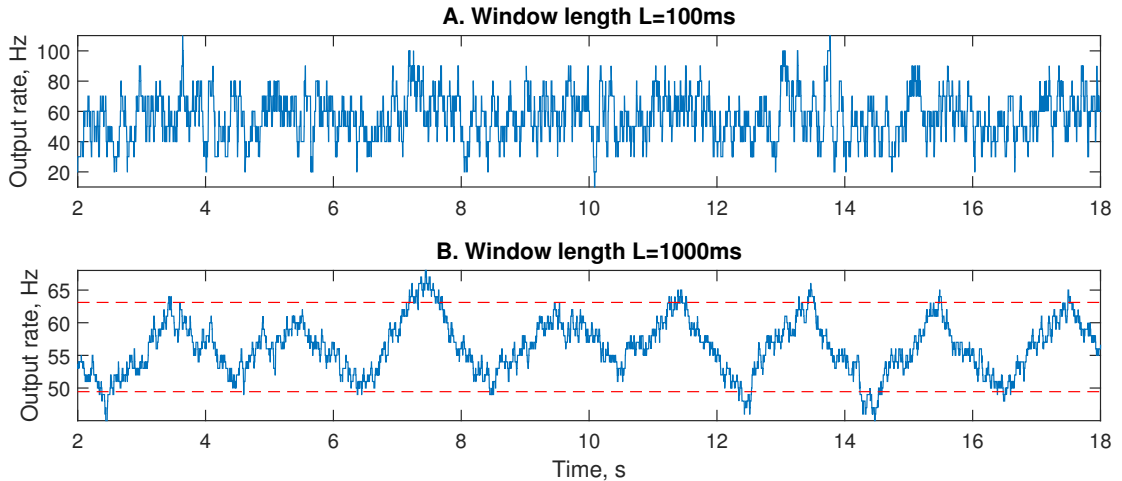


Figure 3.6: The evolution of firing rate over time. To find instantaneous firing rates, a sliding time window of length L was chosen, and for each timestep t , instantaneous firing rate was calculated as the firing rate in time window $[t - \frac{L}{2}, t + \frac{L}{2}]$. A. For $L = 100\text{ms}$, little to no oscillation is visible. B. For $L = 1000\text{ms}$, a clear oscillation is seen. The triangular shape of oscillation is caused by the method of calculating instantaneous firing rate. The top dashed line shows mean output rate over all time periods when the neuron was shown trained input; the bottom dashed line shows the mean over all time periods when the neuron was shown flat input.

3.2.2 Response to partial presentation of learned inputs

An important aspect of the behaviour of a neuron is its ability to recognise partial or incomplete inputs. To study this, a neuron was trained on the pattern shown on Figure 3.5A and plasticity subsequently disabled. In the testing phase, intensity of the learned pattern was varied and output rate recorded. The results, given in Figure 3.7, indicate that the neuron responds linearly to changing pattern intensity. The response is linear even when the pattern has negative intensity (the background rate is higher than pattern rate), i.e. the neuron linearly determines the absence of input. A control neuron trained on flat input of 17.5Hz was also tested; the control neuron did not exhibit a strong systematic response.

In addition to varying intensity of the learned pattern, one can show partial input by replacing some of the signal-carrying synapses with noise. For this, a neuron was trained on input consisting of a 40-synapse high-rate (40Hz) channel, with total input normalised to 1750Hz by choosing an appropriate input rate for the other synapses. As shown in Figure 3.8, this neuron was tested by reducing the number of signal-carrying synapses. Again, the neuron exhibited linear behaviour: output rate increased linearly with the proportion of trained pattern present. For control, a neuron trained on flat input of 17.5Hz was tested; the control neuron did not exhibit a strong systematic response.

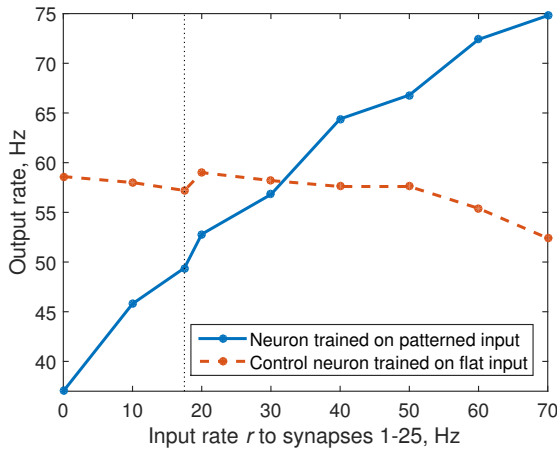


Figure 3.7: Response to varying intensity of learned input. Each datapoint shows output rate in a 5-second simulation. Intensity was varied by giving synapses 1-25 (those trained on a high rate) uncorrelated input r varying from 0Hz to 70Hz, and giving all other synapses such a rate r_b that the total input to the neuron remained constant at 1750Hz. The vertical dotted line shows the value of $r = r_b$ at 17.5Hz.

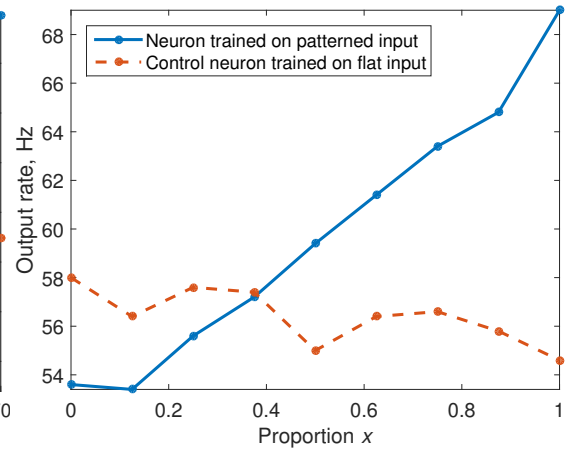


Figure 3.8: Response to varying proportions of trained pattern. Each datapoint shows output rate in a 5-second simulation. x is defined as the proportion of the original 40 high-rate synapses receiving high-rate input in the test. For all other synapses a suitable input rate was chosen to normalise total input to 1750Hz.

3.2.3 Selectivity to patterns with different channel widths

Running any simulation of a neuron requires specifying several parameters governing the structure of the input: number of synapses, rate of firing, inter-synapse correlation, number and size of distinct channels among others. Here the effect of pattern width on output rate is studied.

A neuron is trained by distributing a total of 1750Hz of input among two channels: 1000Hz evenly among synapses $1 \dots w$ (high-rate channel), and 750Hz evenly among synapses $w \dots 100$. For each value of w , a new neuron is trained, plasticity is disabled and the neuron is tested on the training pattern. A control neuron trained on flat input of 17.5Hz ('untrained' neuron) is tested for each value of w with the test pattern corresponding to w . The results for both the trained and the untrained neurons are shown in Figure 3.9.

Evidently, the neuron has a maximal response rate at $w \in [20, 25]$. However, relative difference between the response of a trained neuron and that of an untrained neuron is a better measure for selectivity, as it shows the effect of learning on output rate. This difference is largest in the range $w \in [10, 20]$, suggesting that a pattern of input should cover a small proportion of all available synapses to elicit maximal selectivity. It should be noted, though, that the effect of pattern width

on output rate is low (less than 10Hz) compared to other parameters varied before.

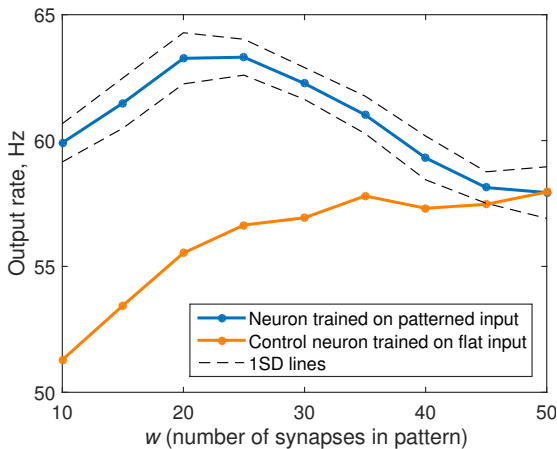


Figure 3.9: Learning patterns represented in varying numbers of synapses. Each datapoint shows mean output rate over 9 test runs lasting for 5 seconds; dashed lines show standard deviation of output rates in these runs.

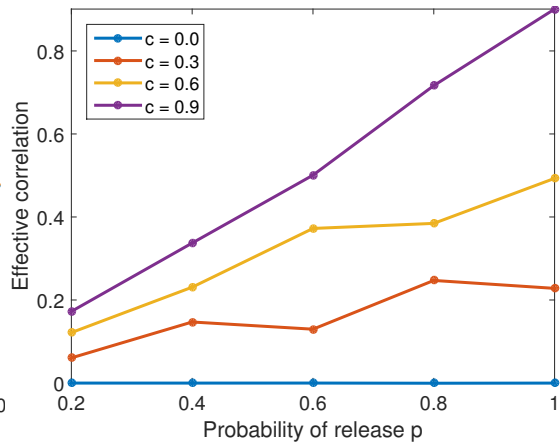


Figure 3.10: The destructive effect of PR on effective input correlation. Each of the four coloured lines show effective correlation c_{eff} for the respective values of probability of release p and correlation c . c_{eff} is calculated by generating 100 correlated spike trains, randomly removing each spike with probability p and calculating mean of the correlations of all pairs of spike trains.

3.3 Probabilistic neurotransmitter release

The effect of probabilistic neurotransmitter release (PR) was studied by varying the probability p of neurotransmitter release upon a presynaptic spike. For all other simulations in this thesis, neurotransmitter is released from the presynaptic side upon every presynaptic spike (i.e. $p = 1$). In contrast, neurotransmitter is released with probability $p \in [0, 1]$ in all simulations of this section. Choosing a value of $p < 1$ introduces synaptic failure into the model.

For uncorrelated Poissonian input of rate r , the only effect of PR is to decrease input rate, resulting in an effective input rate $r_{eff} = p \cdot r$. For correlated spike trains PR has an effect of destroying correlation in the input, as shown in Figure 3.10. Thus, to study effects of PR unrelated to input rate correlated inputs must be studied.

To study the effect of PR on the neuron's input-output relation, an input-output curve similar to those in Section 3.1.1 was produced for different values of p with results shown in Figure 3.11. The only effect of PR for $p \geq 0.4$ is, predictably, to decrease the rate. However, for $p = 0.2$, the shape of the curve is slightly

sigmoidal, which suggests that PR causes higher input rates to have a higher (relative) impact on the output rate, and suppresses lower input rates.

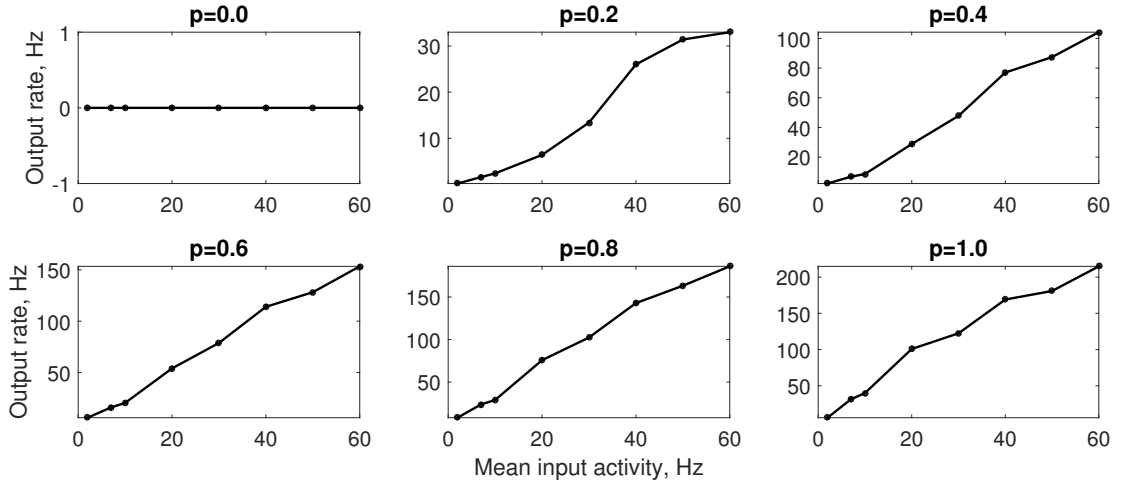


Figure 3.11: Input-output curves for various values of p . Each data point shows output rate in the last 5 seconds of a 100-second simulation. In all simulations, input spike trains were correlated with $c = 0.4$. To avoid errors in numerical integration caused by highly correlated input are included in this thesis, EPSP amplitude was set to 0.5 and spike-frequency adaptation was disabled in these simulations.

Taking a wider perspective, the existence of PR in biological neurons implies that there exists a value of $p \neq 1$, for which some measure of information transfer is at its optimal value. Two such candidate measures are easy to test: output rate and variance in output rate. Figure 3.12 shows the dependence of these two measures on p . At each input rate tested both the output rate and variance (shown as standard deviation) increase as p approaches 1, suggesting no optimal value of p for these two measures in the interval $p \in [0, 1)$. However, at $p = 0.8$, mean output rate is roughly equal to or slightly larger than output rate at $p = 1$. This indicates that, for the values of parameters used, a value of p lower than 1 is useful because a lower p causes lower energy expenditure on spiking and thus a more efficient information transfer.

3.4 Principal component analysis task

The neuron's ability to perform PCA, can be tested in various ways. The approach taken in this thesis is to define two channels of input and to specify the input to each channel independently. Channel 1 consists of half of the 100 synapses; channel 2 contains the other half. Input to two independent channels is generated by choosing input rates (r_1, r_2) from a multivariate Gaussian distribution with

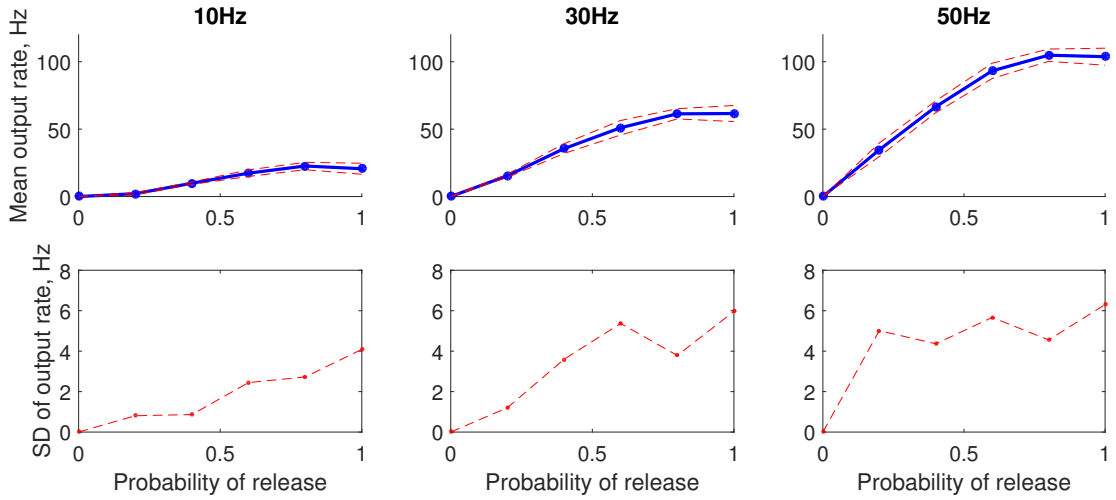


Figure 3.12: The dependence of output rate on PR parameter p . For each combination of rate and p , 10 5-second experiments were run, resulting output rates measured and mean and standard deviation over the 10 results calculated. In all simulations plasticity was disabled with 5 synaptic weights fixed at 1.0 and all others at 0.05. Spike-frequency adaptation was also disabled.

means $(\mu_1, \mu_2) = (10, 10)$ and covariance matrix

$$C = \begin{pmatrix} 10 & 1 \\ 1 & 0.5 \end{pmatrix}$$

with all rates in Hz, and using intra-channel correlation $c = 0.5$. In this setup, each input pattern can be represented in two-dimensional space, with (r_1, r_2) specifying the independent coordinates. Sampling the Gaussian distribution described above, an ellipse forms in this input space, centered at (μ_1, μ_2) .

Training a neuron on these sampled inputs is a formulation of a PCA task: if the weight vector were to align with one of the axes of the ellipse, the neuron could be said to recognise one of the principal components, i.e. perform PCA. However, as shown in Figure 3.13, the weights do not align as required for success in this setup. In informal parameter searches varying μ_1, μ_2 and C , no principal components were recognised either.

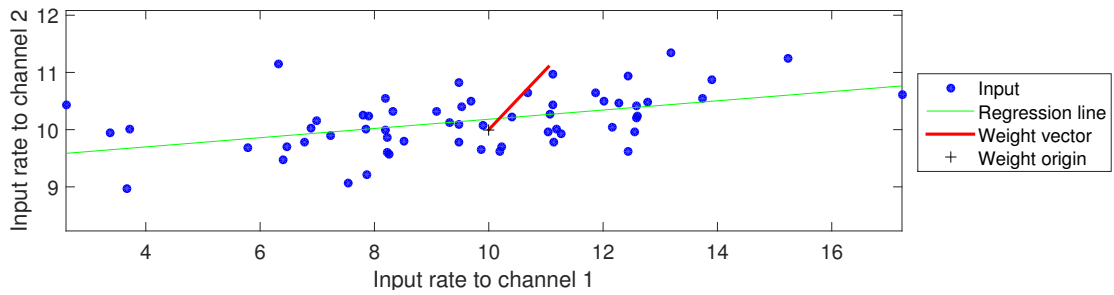


Figure 3.13: Input space and result of PCA task. The 60 blue dots represent inputs shown to the neuron, each for 1s. The green line is the best-fit linear trend line for inputs. The red line is the weight vector: x - and y -components are calculated as mean of channel 1 and 2 weights, respectively. Mean output rate was measured as 29Hz. Spike-frequency adaptation was disabled and EPSP amplitude set to 0.5mV in this experiment.

4 Discussion

This chapter is organised into three sections: first a commentary of the results will be provided, followed by a discussion of the limitations of this thesis in the second section and suggestions for future work in the third section.

4.1 Interpretation of results

Validation of the implemented neuron in Section 3.1 was largely successful with results mostly agreeing with prior work. The input-output curves, STDP curve and metaplasticity evolution look qualitatively similar to those of [SBC02] and [YSBC04]. A significant divergence in this thesis is a much weaker selectivity to correlation. However, by changing the characteristics of the STDP curve (by changing parameters affecting NMDA currents) could produce a stronger correlation selection effect—a hypothesis that can be easily tested in further research. The output rates are high compared to biological neurons, but this is affected a lot by the specific value of EPSP amplitude used.

The neuron clearly learns and exhibits mostly linear behaviour as shown in Section 3.2. Furthermore, the learning can be verified from the output rate of the neuron, which suggests meaningful information processing takes place. In the context of networks of neurons, ANNs often use sigmoid or rectified linear filters; due to the neuron’s linear response to partial input, it can be said to implement a linear filter. The results presented in 3.2.3 also suggest an optimal pattern width for maximal learning capability.

In Section 3.3, the effect of probabilistic neurotransmitter release on the neuron’s behaviour was found to be small but significant. For all but one tested values of p , the results were unsurprising given the correlation-breaking and rate-reducing effect of PR. However, it was found that PR 0.8 is, in some cases, equivalent to PR 1. For $p = 0.8$, PR is indeed useful, eliminating unnecessary spikes but producing the same output rate. Given the large amount of energy used by the brain, PR can be significantly useful even at high values of p . This is in agreement with the conclusion of [Gol04] about PR increasing information transfer efficiency.

The neuron did not successfully perform a PCA task in the regimes explored in Section 3.4. However, a PCA task can be formulated in many ways, and experiments with purely rate-based patterns (those of Section 3.4) do not disprove the neuron’s ability to do PCA. Rather, adjusting the STDP curve and increasing selectivity to correlation might have a positive effect.

4.2 Limitations

Several issues must be taken into account when interpreting the results.

One of the most severe limitations of the methods of this thesis is lack of a systematic basis for choosing parameters. Even though informal parameter searches were conducted, it is possible to do a more systematic search in parameter space and vary more parameters. Doing so would increase confidence that the results hold generally, not only for the specific parameter sets used.

A related issue concerns the speed of simulations. While the simulation code could certainly be optimised in various ways, a significant increase in speed would result from increasing the simulation time-step (0.1ms in this thesis). This would require numerically estimating differential equations with a method other than Euler’s. A longer time-step would allow more simulations and thus more parameter sweeps to be conducted with the same amount of computation.

The lack of strong selectivity to correlation shown in Section 3.1.4 is another significant limitation: without it, some learning modes are not exhibited and thus cannot be studied. To achieve selectivity to correlation, one should do a systematic search of parameters, especially with the goal of getting a narrower STDP curve.

Finally, throughout the work, firing rates are exceptionally high for both inputs and output in comparison with biological neurons. The high input rates are explained by the small number of synapses used: a typical pyramidal neuron might receive input from tens of thousands of synapses, but the neuron simulated in this thesis only has 100 inputs. The output rate is greatly affected by EPSP voltage (as shown in Figure 3.11) and the stable point for synaptic weights. To alleviate the problem of high output rates, these parameters should be varied.

4.3 Future work

Studying probabilistic neurotransmitter release, an approach more grounded in information theory might be in order: using information measures to characterise input and output. More specifically, for some particular measure there might exist an optimal value of probability of release $p < 1$, giving insight into the utility of the PR phenomenon.

For unsupervised learning, a more fundamental approach should be taken to find conditions in which the neuron could implement PCA. In particular, the effect of selectivity to correlation might influence the neuron's ability to perform PCA. Similarities to Oja's rule could also be studied to inform further unsupervised learning tasks. In addition, there are several ways to put the neuron to a PCA task; more of these can be tested.

The plasticity of inhibitory neurons, a topic not in the scope of this thesis, could be studied as well. Building atop of results characterising STDP in inhibitory synapses of the mouse auditory cortex [DF15], one could implement inhibitory plasticity in a biologically realistic way.

Finally, the neuron's potential to implement Bayesian inference could be studied. This could be achieved using a theoretical framework for analyzing and modelling local plasticity mechanisms in the context of probabilistic inference [KHLM15].

Conclusion

In this thesis, a biophysically realistic single neuron model was implemented and investigated. In the first section of Chapter 3, the behaviour of the neuron was found to agree with prior work with the exception of selectivity to correlated input, which was not exhibited in our simulations. In the second section, the neuron was found to implement a linear filter. In the last two sections, probabilistic neurotransmitter release and a PCA task were studied, resulting in some evidence towards the usefulness of PR and no evidence towards the neuron's potential capability to implement PCA in the tasks explored.

The author of this thesis was responsible for implementing and validating the model, running simulations and analysing the resulting data. Parts of the research questions as well as components of the neuron model used were suggested by the supervisors.

Bibliography

- [Abb99] Larry F Abbott. Lapicque’s introduction of the integrate-and-fire model neuron (1907). *Brain research bulletin*, 50(5):303–304, 1999.
- [AGT⁺14] S. Afshar, L. George, J. Tapson, A. van Schaik, and T. J. Hamilton. Racing to learn: Statistical inference and learning in a single spiking neuron with adaptive kernels. *ArXiv e-prints*, August 2014.
- [BEK08] Christoph Börgers, Steven Epstein, and Nancy J Kopell. Gamma oscillations mediate stimulus competition and attentional selection in a cortical network model. *Proceedings of the National Academy of Sciences*, 105(46):18023–18028, 2008.
- [DF15] James A D’amour and Robert C Froemke. Inhibitory and excitatory spike-timing-dependent plasticity in the auditory cortex. *Neuron*, 86(2):514–528, 2015.
- [EG14] Rodrigo Echeveste and Claudius Gros. Generating functionals for computational intelligence: the Fisher information as an objective function for self-limiting hebbian learning rules. *arXiv preprint arXiv:1410.0507*, 2014.
- [EK98] G Bard Ermentrout and Nancy Kopell. Fine structure of neural spiking and synchronization in the presence of conduction delays. *Proceedings of the National Academy of Sciences*, 95(3):1259–1264, 1998.
- [GL12] Daqing Guo and Chunguang Li. Population rate coding in recurrent neuronal networks with unreliable synapses. *Cognitive neurodynamics*, 6(1):75–87, 2012.
- [Gol04] Mark S Goldman. Enhancement of information transmission efficiency by synaptic failures. *Neural computation*, 16(6):1137–1162, 2004.
- [GS06] Robert Gütig and Haim Sompolinsky. The tempotron: a neuron that learns spike timing-based decisions. *Nature neuroscience*, 9(3):420–428, 2006.

- [Heb52] Donald Olding Hebb. *The organisation of behaviour: a neuropsychological theory*. Wiley, 1952.
- [HH52] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.
- [HH09] Suzana Herculano-Houzel. The human brain in numbers: a linearly scaled-up primate brain. *Frontiers in human neuroscience*, 3, 2009.
- [KFF14] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014.
- [KHLM15] David Kappel, Stefan Habenschuss, Robert Legenstein, and Wolfgang Maass. Network plasticity as Bayesian inference. *arXiv preprint arXiv:1504.05143*, 2015.
- [MBE⁺09] J. Macke, P. Berens, A. Ecker, A. Tolias, and M. Bethge. Generating spike trains with specified correlation-coefficients. *Neural Computation*, 2009.
- [MKS⁺15] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [MP43] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.
- [MSS97] Venkatesh N Murthy, Terrence J Sejnowski, and Charles F Stevens. Heterogeneous release properties of visualized individual hippocampal synapses. *Neuron*, 18(4):599–612, 1997.
- [Oja08] Erkki Oja. Oja learning rule. *Scholarpedia*, 3(3):3612, 2008.
- [SBC02] Harel Z Shouval, Mark F Bear, and Leon N Cooper. A unified model of NMDA receptor-dependent bidirectional synaptic plasticity. *Proceedings of the National Academy of Sciences*, 99(16):10831–10836, 2002.
- [Sho07] Harel Z Shouval. Models of synaptic plasticity. *Scholarpedia*, 2(7):1605, 2007.
- [SW95] Charles F Stevens and Yanyan Wang. Facilitation and depression at single central synapses. *Neuron*, 14(4):795–802, 1995.
- [YSBC04] Luk Chong Yeung, Harel Z Shouval, Brian S Blais, and Leon N Cooper. Synaptic homeostasis and input selectivity follow from a calcium-dependent plasticity model. *Proceedings of the National academy of Sciences of the United States of America*, 101(41):14943–14948, 2004.

Appendix A: MATLAB code

The MATLAB scripts for simulations and data analysis performed in this thesis are included as an online supplementary material. The material will be withheld from publication until 26.06.2016. The online supplementary contains the following files and folders:

- `data_out/` – contains no files; all simulation results will be saved into this folder.
- `helper_functions/` – contains the η and Ω functions and ion channel gating functions.
 - `eta2004.m` – the η function.
 - `learning_curve2004.m` – the Ω function.
 - `makePyrGatingFuns.m` – creates gating functions for the pyramidal neuron.
 - `Mg_block.m` – the magnesium gating function.
- `input_generation/` – contains functions required for generating input spike trains.
 - `mackeetal/` – contains code from [MBE⁺09]. `mackeetal/lib/bivnor.c` needs to be MEX-compiled before use.
 - `GenerateInputSpikesMacke.m` – function for generating correlated input using code from [MBE⁺09].
 - `GenerateInputSpikesUncorrelated.m` – function for generating uncorrelated input.
 - `HomoPoisSpkGenTaivo.m` – function for generating homogenous Poisson spike trains.
 - `ProbabilityOfRelease.m` – function for applying probabilistic neurotransmitter release with specified p to spike trains.
- `parameters/SingleNeuron_IF_Taivo_Parameters_2004` – a script that sets the values of most simulation parameters.
- `SingleNeuron_IF_Taivo.m` – the main script for simulating the neuron.

Appendix B: Glossary

ANN artificial neural network

BPAP back-propagating action potential

EPSP excitatory postsynaptic potential

GABA gamma-aminobutyric acid

IPSP inhibitory postsynaptic potential

LTD long-term depression

LTP long-term potentiation

NMDA N-Methyl-D-aspartic acid

NMDAR NMDA receptor

PCA principal component analysis

PR probabilistic [neurotransmitter] release, probability of release

STDP spike-timing-dependent plasticity

Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, Taivo Pungas (21.02.1994),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
 - 1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
 - 1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, as of 26.06.2016 until expiry of the term of validity of the copyright,

“Implementing and investigating a biologically realistic neuron model”, supervised by Raul Vicente and Jaan Aru,
2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 14.05.2015