Charles University in Prague
First Faculty of Medicine

Dissertation Thesis

Mgr. Július Štroffek

Activity and Memory in Biologically Motivated Neural Network

Institute of Pathological Physiology

Thesis Mentor: prof. MUDr. RNDr. Petr Maršálek, Ph.D.
Institute of Pathological Physiology
First Faculty of Medicine
Charles University in Prague

Prohlášení

Prohlašuji, že jsem závěrečnou práci vypracoval samostatně a že jsem řádně uvedl a citoval všechny použité prameny a literaturu. Současně prohlašuji, že práce nebyla využita k získání jiného nebo stejného titulu.

Souhlasím s trvalým uložením elektronické verze mé práce v databázi systému meziuniverzitního projektu Theses.cz za účelem soustavné kontroly podobnosti kvalifikačních prací.

V Praze, 12.6.2018

Identification Record

ŠTROFFEK, Július. *Activity and memory in Biologically Motivated Neural Network.* Prague, 2018, 113 pages, 1 appendix. Dissertation thesis. Charles University in Prague, First Faculty of Medicine, Institute of Pathological Physiology. Thesis Mentor: Maršálek, Petr

Acknowledments

I would like to thank to my mentor Peter Marsalek for his help and guidance of my work, numerous comments and suggestions to the formulation of the text I was writing. I would like to give thank you to all my family and especially to my wife Pavla for the support of all my efforts.

Abstract

This work presents biologically motivated neural network model which works as an auto-associative memory. Architecture of the presented model is similar to the architecture of the Hopfield network which might be similar to some parts of the hippocampal network area CA3 (Cornu Amonis). Patterns learned and retrieved are not static but they are periodically repeating sequences of sparse synchronous activities. Patterns were stored to the network using the modified Hebb rule adjusted to store cyclic sequences. Capacity of the model is analyzed together with the numerical simulations. The model is further extended with short term potentiation (STP), which is forming the essential part of the successful pattern recall process. The memory capacity of the extended version of the model is highly increased. The joint version of the model combining both approaches is discussed. The model might be able to retrieve the pattern in short time interval without STP (fast patterns) or in a longer time period utilizing STP (slow patterns). We know from our everyday life that some patterns could be recalled promptly and some may need much longer time to reveal.

Keywords

auto-associative neural network, Hebbian learning, neural coding, memory, pattern recognition, short-term potentiation

Abstrakt v češtině

Tato práce prezentuje biologicky motivovaný model neuronové sítě, který funguje jako autoasociativní paměť. Architektura prezentovaného modelu odpovídá architektuře Hopfieldovy sítě, jež může odpovídat některým částem, které byly identifikovány v hipokampální oblasti CA3 (Cornu Amonis). Vzory v modelu nejsou statické stavy neuronů, ale cyklicky se opakující synchronní aktivity s nízkým relativním počtem současně aktivních neuronů. Vzory jsou do sítě uloženy pomocí Hebbova pravidla upraveného na ukládání sekvencí. Navrhnutý model je analyzován z pohledu kapacity spolu s numerickými simulacemi. Model je dále rozšířen o krátkodobé posilování synapsí (STP), které je v modelu nutnou součásí správného vybavování vzorů. Důsledkem tohoto rozšíření je další výrazné zvýšení kapacity modelu. V práci je diskutována možnost kombinace obou přístupů. Síť může zpracovat vzory v krátkém časovém intervalu bez STP (rychlé vzory) nebo pomocí STP v delším časovém intervalu (pomalé vzory). Z vlastní zkušenosti víme, že některé vzory se mohou vybavit rychle a některé k vybavení potřebují daleko delší čas.

Klíčová slova

auto-asociativní neuronová síť, Hebovské učení, kódování v nervovém systému, paměť, rozpoznávání vzorů, krátkodobé posilování synapsí

Contents

Glossary

Activation Function
the function that is transforming the "membrane potential" of the neuron in a form of weighted sum of inputs to the neuron output. Membrane potential is represented by just a single number in analogy to the experimentally measured membrane potential.

AMPA
is chemical compound alpha-Amino-3-hydroxy-5-Methyl-4-isoxazole-Propionic Acid that activates certain type of postsynaptic receptors. These receptors are therefore named AMPA receptors. AMPA has no effect on NMDA receptors.

Associative Memory
the memory where content is addressed (associated) with another content.

Attractor
the stable state of the dynamic system. In this theses the dynamic system is an auto-associative memory.

Auto-Associative Memory
the memory where the stored content is addressed by the content itself.

CA3
means Cornu Amonis Layer 3 which is the anatomically identified group of neurons (nucleus) in hippocampus.

GABA
is gamma-Aminobutric acid which acts as the main inhibitory neurotransmitter. There are two types of GABA based receptors $\mathrm{GABA_A}$ and $\mathrm{GABA_B}$.

Hard-Limiter Function
is non-continuous activation function that is activating the neuron if the weighted sum of inputs

is above the specified threshold and keeping the neuron quiet otherwise.

Hebb Rule
The learning rule inspired by the hypothesis of Donald Hebb. He proposed that the synaptic efficiency increases as a result of repeatable excitation between neuron cells.

NMDA
is chemical compound N-Methyl-D-Aspartate that activates certain type of postsynaptic receptors. These receptors are therefore named NMDA receptors. The original neurotransmitter for NMDA receptors is glutamic acid. NMDA has no effect on AMPA receptors.

Sigmoidal Function
is the most commonly used continuous activation function. As the weighted sum of inputs is getting higher than the specified threshold, the neuron output is getting closer to 1. As the weighted sum of inputs is getting lower than the specified threshold, the neuron output is getting closer to 0.

STP
means short-term potentiation which is the process that increases the synaptic excitation between pre-synaptic and post-synaptic neuron in the short time window.

List of Figures

List of Equations

Chapter 1

Introduction

Memory is an ability of the organism or system to store, retain and recall information. In biology memory is mostly associative and stored pattern is recalled as a response to some other input activity previously perceived or being very close to the activity previously perceived. The stimulus that leads to the response is called input and the response is called output. There are multiple types of associative memories and possibilities how they could be implemented.

Usually, the same memory can store lot of information and during a process of storing and recalling information an addressing of the proper piece has to take place. There are two technical approaches how to address information, which leads to two different types of memories.

Location addressable memory uses the exact location of the information to store and recall the appropriate information. This is the way computer memory works or a sheet of a paper with notes is used. This type of memory is mostly used by computers or other machines. When using a computer memory the location is specified by the memory address.

In *content addressable memory* we store and recall information based on its content. This type of memory is often called *associative memory.* It is natural to most of the organisms and the human brain. The stream of thoughts of our internal monologue (James, 1892) uses the content of thoughts to address the stored information. The thoughts are of course influenced by our perception. The recalled information may have a form of some movement, thought or some chemical reactions. If we feel hunger and smell a good food we would have a clear picture of the food just from its smell even without seeing it. This may lead even to spontaneous salivation without seeing a food as studied by I. P. Pavlov on dogs under different conditions in Pavlov (1927). Pavlov described a conditional reflexes and processes that led to strengthening or weakening of those reflexes. From the memory point of view, these reflexes might be just a store and retrieval processes to/from some associative memory.

In computer science, there exist a lot of data structures with algorithms which try to change the location addressable memory to content addressable memory. They allow us to use efficiently the location addressable memory as “associative dictionary”.

The view on types of memory is different in physiology and psychology. Memory is classified based on the experience it brings to individual. The memory could be declarative or procedural. The declarative memory is used consciously to store or recall information. It could be further divided into semantic memory containing the abstract information and episodic memory storing the contextual information related to time, place, emotions etc.

Other memory classification divides the memory into sensory memory, short term memory and long term memory. The short term memory is transferred into long-term memory by memory consolidation process. This classification was proposed in Atkinson and Shiffrin (1968). However, all these types of memory might have the same or similar underlying neural mechanisms that form their behavior.

In our thesis we simply understand the information as the real vector or the array of real numbers that correspond to some type of neural activity and we do not take care of any further interpretation. We will focus on associative models of neural networks.

One of the most studied memory models in neural networks are auto-associative memories. In this case, the network does not need any separate input and output neurons. The input is presented to the network as external excitation or simply by setting up the neuron outputs. The network afterwards evolves based on its dynamics and it converges to the stable state. This stable state is considered to be the output of the network. This is achieved by the recurrent connections in the network where output of each neuron can be potentially connected as input to all other neurons. The first published model of an auto-associative network was in Hopfield (1982). More detailed overview about the auto-associative networks can be found in Amit (1989).

Artificial neural networks usually distinguish between learning and recall processes. The learning process is used to set up synaptic efficiencies in a way that will allow the model to retrieve the required patterns. This is often based on the Hebb’s hypothesis (Hebb, 1949; Kuriscak et al., 2015). During recall process these synaptic efficiencies do not change and the network dynamics is used to recall the corresponding pattern. It was shown in Tsodyks and Markram (1997); Tsodyks et al. (1998) that the changes in synaptic efficiencies could occur in a time scale small enough to also affect the pattern recall processes. We provide a brief overview of the models relevant to our work with the reference to the literature.

1.1 Neuron Models

Neuron consists of many dendrites, the cell body and the only one axon that could be split into multiple branches. The dendrites “capture” the signals from other neuronal cells and convey them to the cell body. The cell body integrates the signals over period of time and across all the dendrites. If the excitation in the cell body reaches certain threshold the neural spike will be generated in the axon. Therefore the dendrites are incoming connections and axon is the outgoing connection of the neuron.

1.1.1 Formal Neuron

To study more complex behavior of the neural system it is required to build a network of multiple neurons with connections between them. The output of neuron is usually modeled as a number that changes in time based on the neuron inputs. The synaptic strengths are modeled as real numbers as well and are referred as synaptic weights. The weighted sum of inputs ξ is calculated as

$$\xi = \sum_{i=1}^{n} w_i x_i + \vartheta, \tag{1.1}$$

where x_i denotes the input, w_i denotes the weight of the corresponding synaptic connection and ϑ denotes the bias which mimics the excitation threshold and external input. The output of the neuron is calculated from the above equation as

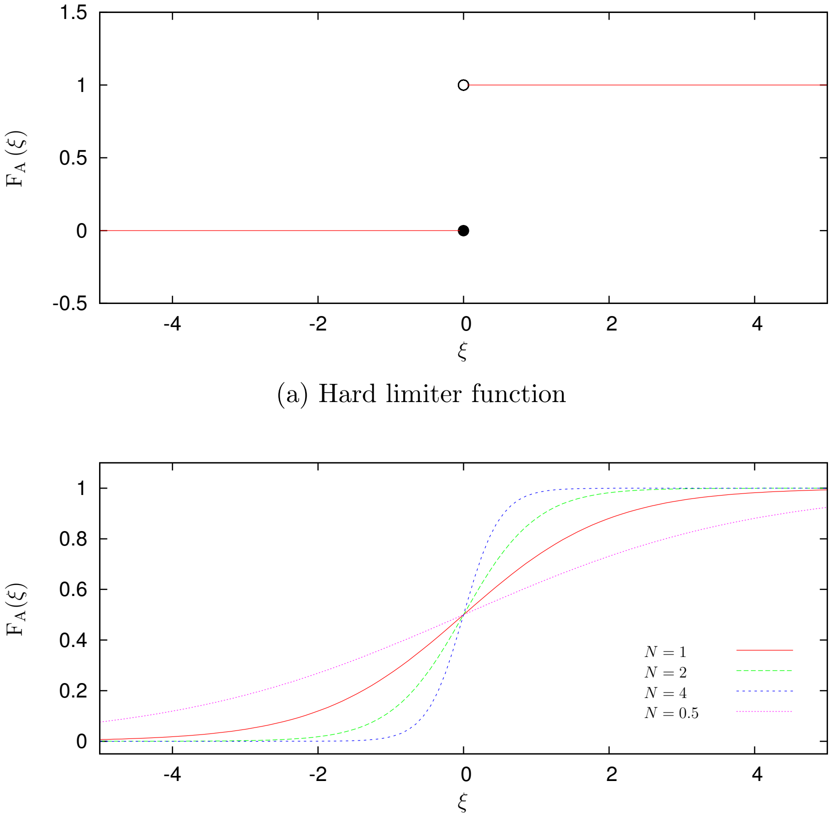
$$x = \mathrm{F}_{\mathrm{A}}(\xi) = \mathrm{F}_{\mathrm{A}} \left(\sum_{i=1}^{n} w_i x_i + \vartheta \right) \tag{1.2}$$

where F_A is the activation function that translates the weighted sum of inputs into neuron output. The most commonly used activation functions are shown in Figure 1.1.

There are additional neuron models used for modeling artificial neural networks.

1.1.2 Hodgkin-Huxley Model

Hodgkin and Huxley have published the detailed neuron model based on the measurements of voltage on giant axon of loligo (Hodgkin and Huxley, 1952). They have anticipated the existence of ion channels and have created the model that describes the voltage on a fixed point of a neuron’s membrane (usually



(a) Hard limiter function

(b) Sigmoidal function with multiple steepness values N.

Figure 1.1: Most commonly used activation functions in artificial neurons. Hard limiter activation function is used in discrete models where the neuron output gains only binary values. The sigmoidal function with the steepness parameter is used in the models with continuous neuron outputs.

in so-called initial segment of axon). They have proposed the system of differential equations and adjusted its parameters to fit their measurements. The main idea of the model is based on dynamics of ion channels.

Ion Channels

A neural membrane consists of phospholipid bilayer which has bultin protein molecules having various functions. Some of them are ion channels which are “builtin holes” which allows a gradient based flow of ions across the membrane. These channels could be specific – allowing flow of specific ions. The permeability of these channels could be “controlled” – channel could be either open or closed, i.e. *gated.* Based on this, we can categorize the relevant channels into

- *Receptor-Gated Channels* – they are opened when the specific molecule binds to the receptors. They are a key part in synaptic transmission.
- *Voltage-Gated Channels* – they are opened when the voltage on cell membrane reaches certain value. They are a key part in signal transmission along the neuron membrane.
- *Mechanically Gated Channels* – they are opened under the mechanical stimulation and they are based in receptors of sense organs like inner ear or touch.
- *Leakage Channels* – they are open to all ions under normal circumstances.

A neural membrane contains the voltage-gated channels for sodium and potassium ions. Sodium channels are opened at lower voltage threshold than the potassium channels. In quiet period, the intracellular concentration of potassium, as well as extracellular concentration of sodium, are higher. When the voltage increases on the membrane, the sodium channels are opened (while the potassium channels are still closed).

Sodium ions are flowing into the cell which further increases the membrane potential. This leads to less sensitive potassium channels to be opened. The potassium ions start to flow into the extracellular area. This balances the previous voltage changes. Furthermore, the sodium and potassium ions are switched between intracellular and extracellular areas by the sodium-potassium pump which then further leads to the same ion concentrations. This process consumes a lot of energy. We will not describe the sodium-potassium pump in more detail as the model considering the ion channels provides a good enough approximation.

We can use the Ohm's Law to calculate the ion current as $i = g(u - E)$, where u is membrane voltage, g is conductivity of ion channels (can dynamically change in time) and E is the equilibrium potential.

Model Definition

Cellular membrane has properties of capacitor and the relation of voltage and current on capacitor is determined by the equation

$$C \frac{\mathrm{d}\,u}{\mathrm{d}\,t} = -i, \tag{1.3}$$

x	G_x [mS/cm^2]	E_x [mV]
Na	120	115
K	36	-12
L	0.3	10.6

Table 1.1: Model parameters. The capacity of a neural membrane is $1\,\mu\text{F/cm}^2$. Values of equilibrium potentials are adjusted in a way that will lead to the resting potential of 0 mV. Taken from Gerstner and Kistler (2002).

where C is capacity of neural membrane. Current i on the right side could be described as the sum of currents on individual channel types as

$$i = \sum_{i=1}^{N} g_i(u - E_i) + i', \tag{1.4}$$

where i' is additional or external current. We will be interested in three types of channels – leakage channels, voltage-gated sodium channels and voltage-gated potassium channels. Other channel types can affect the signal transmission as well, but the main dynamics of action potential could be described well enough with this limitation. In our case

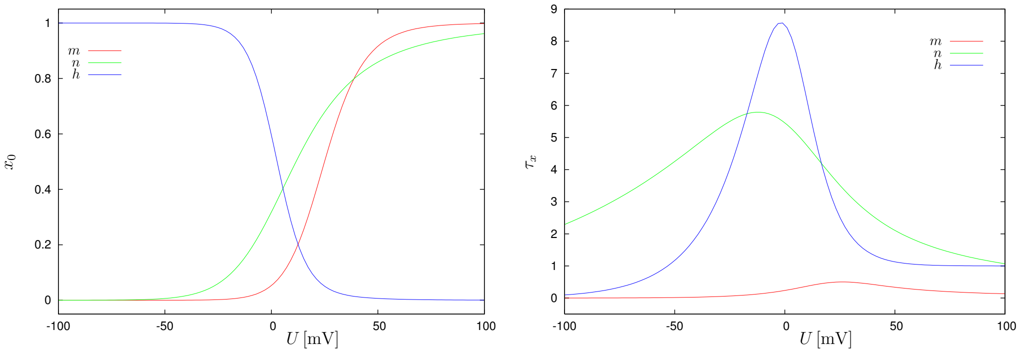
$$i = g_{Na}(u - E_{Na}) + g_K(u - E_K) + g_L(u - E_L) + i'. \tag{1.5}$$

Leakage channels are not gated nor otherwise controlled and their conductivity does not change in time. The conductivity of sodium and potassium channels changes in time considering the membrane voltage. Hodgkin and Huxley have formulated the equations which describe dynamics of these channels. They have rewritten the equation 1.5 into the form

$$i = G_{Na}m^3h(u - E_{Na}) + G_K n^4(u - E_K) + G_L(u - E_L) + i', \tag{1.6}$$

where G_{Na}, G_K a G_L correspond to maximal conductivity when all the channels for the specific ion are open. Gating functions m, n a h describe the activation dynamics of the ion channels. They are given by the equations

$$\begin{aligned}
\frac{\mathrm{d}m}{\mathrm{d}t} &= \alpha_m(u)(1 - m) - \beta_m(u)m \\
\frac{\mathrm{d}n}{\mathrm{d}t} &= \alpha_n(u)(1 - n) - \beta_n(u)n \\
\frac{\mathrm{d}h}{\mathrm{d}t} &= \alpha_h(u)(1 - h) - \beta_h(u)h.
\end{aligned} \tag{1.7}$$



(a) Equilibrium functions x_0

(b) Time constants τ_x

Figure 1.2: Model functions τ_x a x_0 from the equations 1.8.

Every equation from 1.7 could be rewritten and further adapted as

$$\begin{aligned}\frac{\mathrm{d}\,x}{\mathrm{d}\,t} &= \alpha_x(1-x) - \beta_x x = \alpha_x - x\alpha_x - \beta_x x = \alpha_x - x(\alpha_x+\beta_x) \\ &= -\left(\alpha_x+\beta_x\right)\left(x - \frac{\alpha_x}{\alpha_x+\beta_x}\right) = -\frac{1}{\tau_x(u)}\left(x - x_0(u)\right), \qquad (1.8)\end{aligned}$$

where x could be any of the gating functions m, n or h and furthermore

$$\tau_x(u) = \frac{1}{\alpha_x(u)+\beta_x(u)} \qquad (1.9)$$

$$x_0(u) = \frac{\alpha_x(u)}{\alpha_x(u)+\beta_x(u)}\,. \qquad (1.10)$$

Functions α_x and β_x were chosen to match the experimentally measured data. See the table 1.2. If the membrane voltage would be constant the solution to the equation 1.8 would be

$$x(t) = -\tau K e^{-\frac{t}{\tau}} + x_0\,, \qquad (1.11)$$

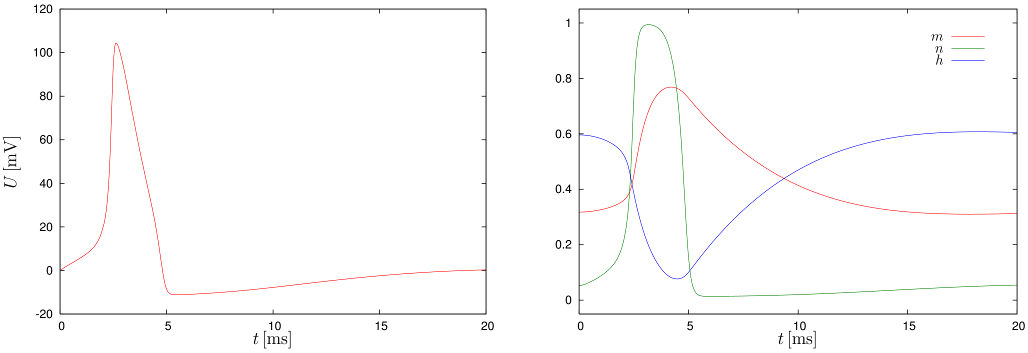
where K is the constant determined according the initial constraint and therefore $\lim_{t\to\infty} x = x_0$. For the fixed u the value of x_0 in equation 1.8 corresponds to the convergent value of the conductivity. The parameter τ_x influences the convergence speed to this value. However, the underling properties of the ion channels are influenced by the membrane voltage and thus τ_x a x_0 are not constants but functions of voltage. See the Figure 1.2.

External Stimulation

Model of Hodgkin and Huxley could be used to describe the behavior of the neural action potential. There is a graph of action potential triggered by the

x	α_x	β_x
m	$\dfrac{(2.5-0.1u)}{e^{2.5-0.1u}-1}$	$4e^{-\frac{u}{18}}$
n	$\dfrac{0.1-0.01u}{e^{1-0.1u}-1}$	$0.125e^{-\frac{u}{80}}$
h	$0.07e^{-\frac{u}{20}}$	$\dfrac{1}{e^{3-0.1u}+1}$

Table 1.2: Empirically determined functions of the model.



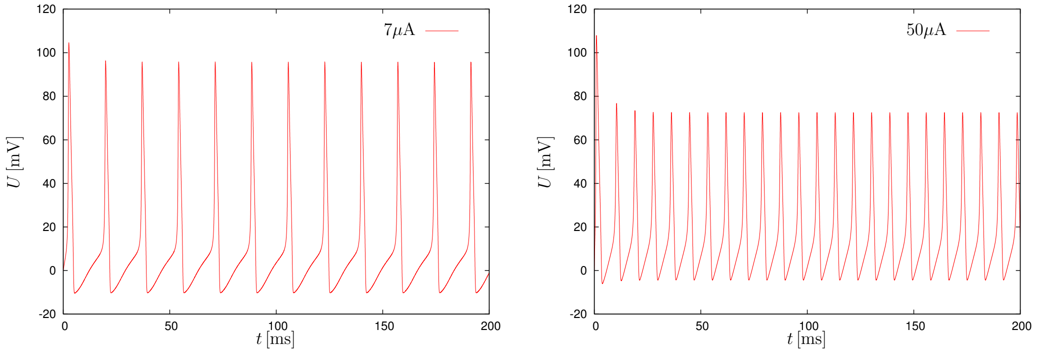
(a) Action potential (b) Gating functions

Figure 1.3: Action potential and gating functions for Hodgkin-Huxley model triggered by the stimulation of constant current of 7 μA for 2 ms. The plots were produced by the numerical calculations using Runge–Kutta–Fehlberg method(Fehlberg, 1969, 1970).

minimal possible constant current on the left of the Figure 1.3. The graphs of gating functions for the same time are shown on the right side of the figure.

If the stimulation with the constant current will last longer, the action potentials will be triggered periodically. The frequency of action potentials will depend on the intensity of the current. When the intensity of the current is increased the frequency will be increased as well but the amplitude of the action potentials will be decreased. There are repeating action potentials shown in Figure 1.4 triggered by the currents of different intensities.

The interesting behavior of the model is *hysteresis* which could be seen in Figure 1.5. It is the phenomenon observed if the neuron membrane is stimulated by the linearly increasing current which will be further linearly



(a) The current with intensity of 7 μA (b) The current with intensity of 50 μA

Figure 1.4: Action potentials triggered by the currents of different intensities.

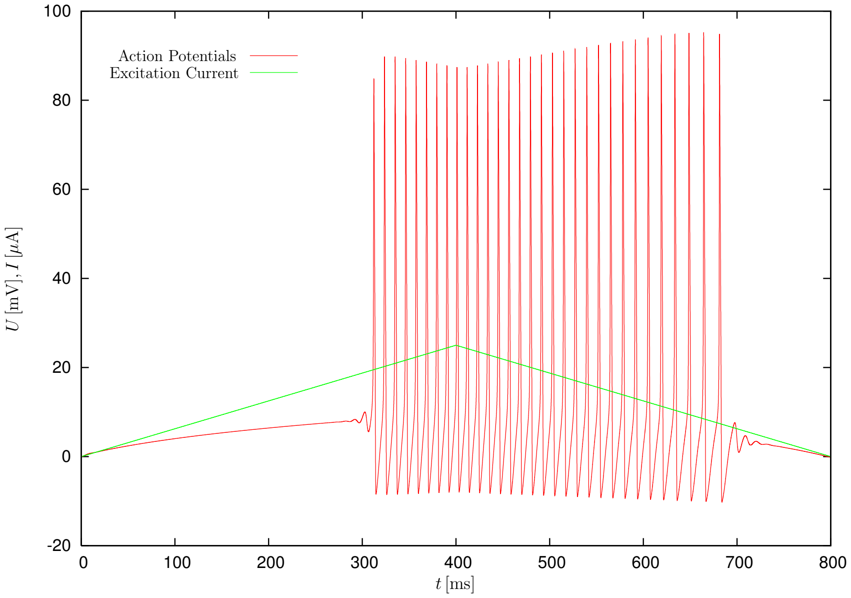


Figure 1.5: Hysteresis in external stimulation

decreasing at the same steepness after reaching the maximum. Neuron starts to trigger action potentials very close the the local maximum. Although, it will still be triggering the action potentials on descent further even for the currents with much lower intensity. It was first modeled an later measured on giant axon of loligo (Guttman et al., 1980).

1.1.3 Other Neuron Models

FitzHugh-Nagumo model (FitzHugh, 1955) is approximation of Hodgkin and Huxley's model with simplified equations. The timing of spikes is important in spiking neurons (Gerstner and Kistler, 2002). The first neuron model capturing the concept of spikes was integrate-and-fire neuron published in Lapicque (1907) and it was integrating the excitation from multiple inputs over time, see (Abbott, 1999). Leaky integrate-and-fire neuron has in addition a decrease of the integrated neuron potential with time (Brette and Gerstner, 2005).

Another concept in artificial neural models was presented in Kohonen (1982) as self-organizing maps. Kohonen used local rules which are used to assign synaptic weights to the neurons without a need for the required neuron output. He selected the *winner* neuron having the highest neuron output. He then changed the synaptic weights that the output of the winner neuron will further increase for the same input. The surrounding neurons were changed using the variable learning strength based on the distance from the winner neuron. The neurons in the weight space mimic the density and shape of the presented input data.

1.1.4 Inhibition

The transmission of a neural spike on a synapse occurs with the support of mediator that is released into the synaptic cleft. Based on the type of the mediator, the neural spike will induce excitation or inhibition on the target post-synaptic neuron. The type of the response is determined globally for the whole neuron based on the mediator that the pre-synaptic cell produces. This means that the neuron either reacts excitatory or inhibitory for all the neurons the outgoing connections are connected to.

In artificial neural networks the inhibition is modeled by the synaptic weights gaining also negative values. This allows the output of the neuron to react inhibitory for target neurons when the synaptic weights are negative, see equation 1.1. However, the same neuron can be both inhibitory and excitatory for different target neurons based on the value of synaptic weights.

There is also a possibility that inhibitory neurons can be chained. Assume that we have three neurons chained where the first neuron is inhibitory and it connects to the second neuron. The second neuron is again the inhibitory neuron and connects to the third excitatory neuron. The excitation of the first inhibitory neuron inhibits the second interneuron which has finally the excitatory effect on third neuron, see drawing A in Figure 1.6. More complex architectures and chains are also possible and are shown in drawings B and C. There are similar neural circuits found in hippocampus which are shown in

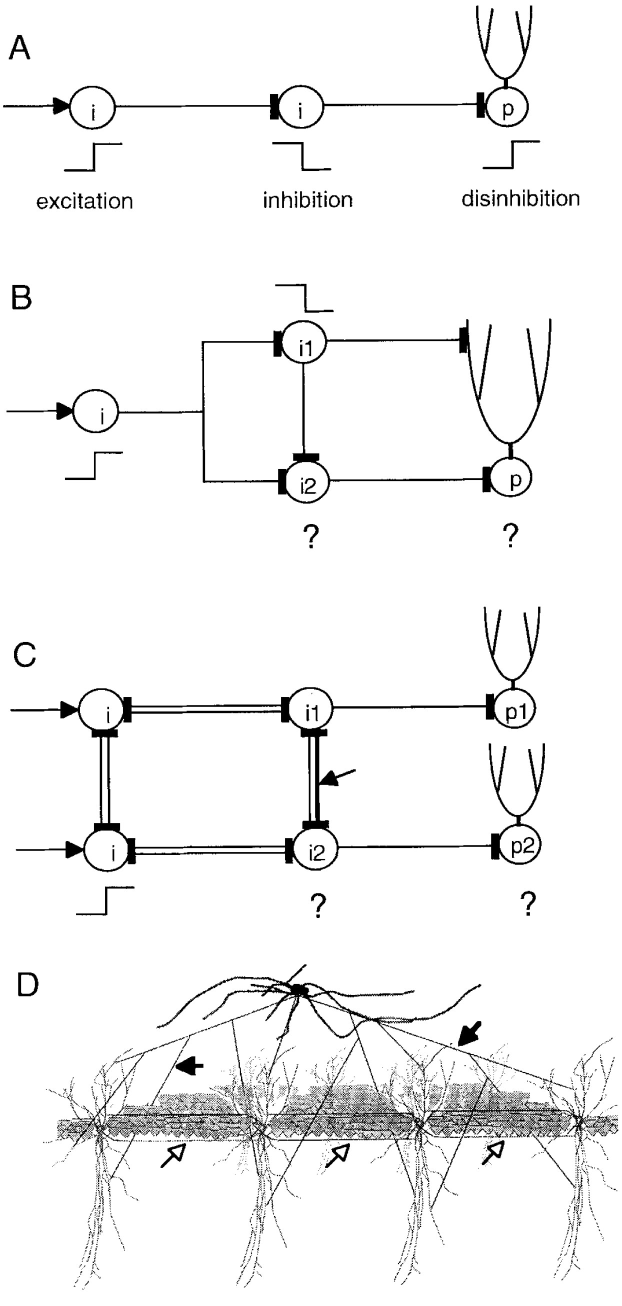


Figure 1.6: The schema of inhibition and dis-inhibition taken from Freund and Buszaki (1996). The figures A, B and C show the various schematic connections of chained inhibitory neurons. The figure D shows the schematics of the corresponding neural circuits found in hippocampus.

drawing D.

1.2 Neural Coding

There are multiple ways how the information can be encoded in neural activities. The proper encoding of information in the human brain is not yet known. It is very likely that the encoding differs for different areas of the neural system (Rybar et al., 2002).

The neural coding can be sparse which means that there is a certain meaning of individual spikes or small groups of spikes. These spikes can have direct effect like muscle fiber contraction. The overall neural activity for sparse encoding is very low. The opposite of sparse encoding is distributed encoding where individual spikes do not have any effect and the large group of neurons spiking together and being distributed across the whole network provides the relevant information. The relative activity in distributed encoding is much higher than in sparse encoding.

There are multiple options how events in neural spikes could encode information. It is very likely that there is no unique encoding of information within neural system. Various areas of the nervous system can use different encoding mechanisms. We will provide a brief overview of the most commonly discussed neural encoding mechanisms.

1.2.1 Mean Firing Rate

First mechanism of the information encoding in neural spikes is the mean firing rate. This is usually the case when the artificial neuron output mentioned in the equation 1.2 is continuous. It is assumed that the artificial neuron output corresponds to the mean firing frequency. The maximal possible artificial neuron output corresponds to the maximal firing frequency. The minimal possible artificial output corresponds to the minimal firing frequency. Very often the artificial neuron outputs are from the interval $(0, 1)$.

Individual spikes are expected to provide no information. This encoding would lead to slower response times as the longer time period is required to identify the mean firing rate (Rybar et al., 2002). This is the encoding that is used in attractor neural network models such as Hopfield network (Hopfield, 1982, 1984) mentioned later in this thesis. First version of Hopfield network updates neuron outputs sequentially one after the other, which mimics the mean firing rate encoding and results in longer response times of the model as well.

1.2.2 Time to First Spike

It is proposed by many studies that in some areas of the brain the first spike in the spike train is more significant than following spikes in the spike train (Tuckwell and Wan, 2005; Gerstner and Kistler, 2002). This encoding is proposed mainly for cortical neurons (Tuckwell and Wan, 2005). The first spike on certain neural pathway usually corresponds to the new event or object observed by the sensory system. Likely, all the animals have to promptly react on certain events that signal a potential danger for them. A very fast reaction is required in some of these cases. It can be achieved by the time to first spike neural encoding.

1.2.3 Timing and Coincidence of Spikes

It is assumed that coincidence of spikes is important for sound localization (Jeffress, 1948). Jeffress proposed the existence of the delay line with coincidence detectors. This delay line is assumed to be present on pathway from each ear. Neurons that act as coincidence detectors are connected to both delay lines at places with different spike delay. The coincidence detectors will fire an action potential only if the inputs arrive synchronously at the same time or within a very short time window. The sound can be then localized based on the neuron firing. This delay line was found in birds, see Young and Rubel (1983) but it was not found in mammals. An alternative mechanism was proposed for humans in Marsalek and Lansky (2005). It has been shown that Hodgkin-Huxley neurons could be used as coincidence detectors with appropriate precision (Marsalek, 2000).

Coincidence of spikes might look similar to time to first spike encoding. The importance of the first spikes is much higher for "time to first spike" encoding than the rest of the spike train. However, for "timing and coincidence of spikes" all the spikes within the spike train are equally important.

1.2.4 Polychronization

The transfer of spikes between neurons has different delays based on the local properties. Therefore the spikes originating on one neuron can excite different neurons in different times. When there is a group of neurons with synaptic connections with different delays, the overall neural activity produces complex spatio-temporal behavior which is called polychronization. The network of Hodgkin-Huxley neurons with different synaptic delays having various[1] synaptic dynamics and synaptic plasticity was studied in (Izhikevich, 2006). It has

[1] AMPA, NMDA, $\mathrm{GABA_A}$ and $\mathrm{GABA_B}$

been demonstrated that if the same pattern was repeatedly presented to the network, the network synaptic parameters were adjusted and these complex spatio-temporal patterns were seen. These patterns are formed by the selected neurons together with their precise spike timings. These patterns were called polychronous groups.

1.3 Learning in Neural Networks

Learning is the process that tries to change the behavior of the model to give the desired results. Neural network models have parameters that can be adjusted and the network will give different response. These parameters can be synaptic weights, activation function steepness or other parameters of the activation function. During the *learning* process, the values of these parameters are adjusted according the learning algorithm.

One of the first hypothesis how the synaptic weights should be adjusted was formulated in Hebb (1949). He postulated: *When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.* Most of the approaches to learning in neural networks emerged from this hypothesis (Kuriscak et al., 2015). Numerous biological processes have been described that confirm the Hebb's hypothesis (Kuriscak et al., 2015; Rolls and Treves, 1998; Tsodyks and Markram, 1997; Tsodyks et al., 1998). There is no need to know the desired network outputs for changing the weights using this learning rule. This type of approach to learning is referred as unsupervised learning.

Another approach to learning of neural networks is purely mathematical. Neural network model is seen as a function of it's input. There is a set of training patterns where each training pattern consists of the network input and the corresponding desired network output. The parameters of the network model are adjusted by numerical methods. The aim is to minimize the error function defined as the sum of squares of differences between actual network output and the desired network output (Rumelhart et al., 1986). There are numerous other approaches when the desired network output is already known and are called supervised learning as opposite to the previous approach. It is used mainly in models called feed forward neural networks. These models are commonly used in various technical applications.

There are numerous use cases of neural networks that are originally motivated by the biology, but in the specific application the focus is primarily on the performance of the model in the desired case, not that much on its biological interpretation. The artificial neural networks are used in applications that recognize certain properties of objects. They are also used for the translations of texts between languages, to identify the mostly suitable advert to show, or even to control the self-driving vehicles.

1.4 Models of Associative Memories

In computer science, the read only memory is a function from the address space to the stored value space.

$$\mathcal{M} : \mathbb{A} \to \mathbb{V} \tag{1.12}$$

where $\mathbb{A}$ denotes the address space and $\mathbb{V}$ denotes the stored value space. Information storage into the memory requires changes of the memory function $\mathcal{M}$.

In case of *location addressable memory* the address space simply contains all the addresses of memory locations available for information retrieval. The size of the address space is always the same as the capacity of the memory.

For the *content addressable memory* the address space contains all the possible values of the content. The number of possible values that can be stored into the memory is very large. Therefore the size of the address space might be larger than the capacity of the memory. This is especially the case for brains of higher mammals. The content addressable memory can not give the reasonable response to all the possible inputs.

1.4.1 Willshaw Model

One of the first models of artificial neural networks was published in Willshaw et al. (1969). Willshaw network consists of two layers of neurons where the output of every neuron in the first layer is connected to all other neurons in the second layer. There are no connections between the neurons in the same layer nor the backward connections going out from the neurons in the second layer. The schematic diagram of the network topology is shown in Figure 1.7.

The outputs of neurons and synaptic weights have binary values 0 or 1. We denote the number of neurons in the fist layer as N, the number of neurons in the second layer as M and the outputs of neurons in the first and second layers as $x_1 \dots x_N$ and $y_1 \dots y_M$ respectively. The outputs of neurons $x_1 \dots x_N$ are determined by the network input. Afterwards the output of every neuron y_j in the second layer is calculated using the formula 1.2 and thus

$$y_j = \mathrm{H}\left(\sum_{i=1}^{N} x_i w_{ij} + \vartheta_j\right), \tag{1.13}$$

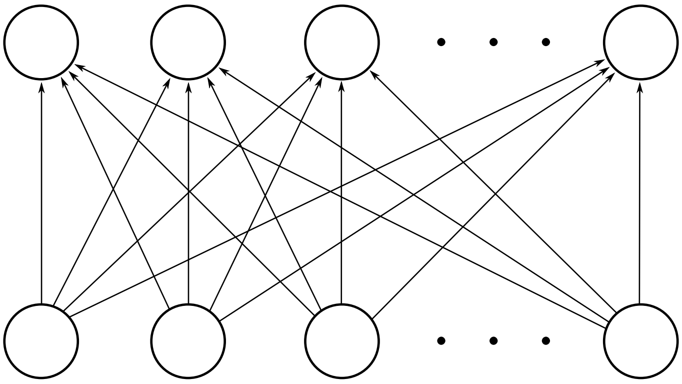


Figure 1.7: Schematic diagram of Willshaw's network

where w_{ij} denotes the synaptic weight between the i-th neuron in the first layer and the j-th neuron in the second layer, ϑ_j denotes the neuron bias and H denotes the hard-limiter function shown in Figure 1.1a

$$\mathrm{H}(x) = \begin{cases} 1 & for \quad x \geq 0 \\ 0 & for \quad x < 0. \end{cases} \tag{1.14}$$

We have a pattern p which we would like to store to the network. The pattern input to the network is p^x, the states of individual neurons are $p_1^x, p_2^x \dots p_N^x$ and the required output is p^y with states of individual neurons $p_1^y, p_2^y \dots p_M^y$. We will denote the number of neurons with output 1 in the pattern's input as N' and the number of neurons in the pattern's output having output 1 as M'. Now we will use the following formulas to determine the synaptic weights and neuron biases

$$\begin{aligned} w_{ij} &= p_i^x p_j^y \\ \vartheta_i &= -N', \end{aligned} \tag{1.15}$$

which will activate the corresponding synapses. We can store more patterns to the network than just one but all of them have to have the same number of active neurons in the input due to the neuron bias. This would lead to the following equations

$$\begin{aligned} w_{ij} &= \mathrm{H}\left(\sum_{t=1}^{T} {}^t p_i^x {}^t p_j^y\right) \\ \vartheta_i &= -N', \end{aligned} \tag{1.16}$$

where T denotes the number of patterns, ${}^t p^x$ and ${}^t p^y$ denotes the t-th network inputs and desired network outputs respectively.

To retrieve the output patterns we have to present exactly the same neurons active as we have active in the input pattern. However, some more active neurons might be present but this may lead to more patterns to be retrieved simultaneously.

1.4.2 Attractor Neural Networks

There are neural network models where the network evolution always leads to a stable state in time. These stable states are called attractors and models of neural networks where the activity is leading to attractors are called attractor neural networks. The research in attractor neural networks was started by Hopfield with a paper presenting a neural network as an analogy to the physical theory of a spin glass in Hopfield (1982). Later on, Hopfield published the continuous time version of the model (Hopfield, 1984) where he studied also the possibility of constructing the hardware version of the network. Other researchers and scientists worked in this field and studied many different variants of these models and their properties, see Amit et al. (1987); Amit (1989); Golomb et al. (1990); Rolls and Treves (1998); Wilson (1999); Stroffek et al. (2007); Gorchetchnikov and Grossberg (2007).

In most cases there exist an abstract energy function of a network state for the model which approaches a local minimum during the network evolution. If the energy function reaches the local minimum, the network state does not change and it remains in an attractor. These kind of abstract energy functions are called Lyapunov functions, more details can be found in Wilson (1999).

Attractor neural networks are often used in optimization tasks due to the abstract energy function minimization. The function to be minimized in the optimization task has to be rewritten in a form similar to the model's energy function. The parameters of the model are then adjusted accordingly. Afterwards, the network will minimize its energy function during its spontaneous evolution from the random state which leads to minimization of the original optimization task.

1.4.3 Hopfield Model

Hopfield network consists of n neurons connected with each other in a way that the output of every neuron is the input to all other neurons in the network. We will denote the neurons with numbers $1 \ldots n$, the outputs of the neurons as $x_1 \ldots x_n$ and the synaptic weight of the connection between i-th and j-th neuron as w_{ij} (neuron i is a pre-synaptic neuron and neuron j is a post-synaptic neuron). The outputs of neurons have only binary values 0 or 1.

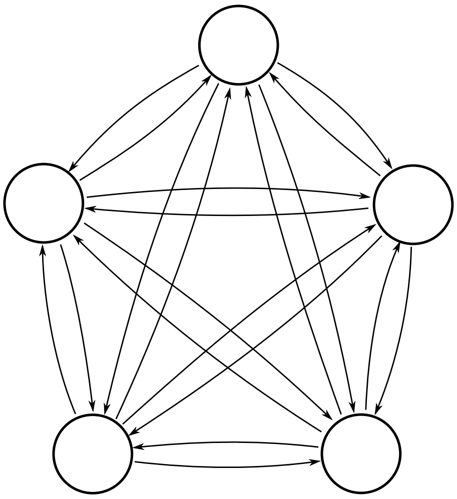


Figure 1.8: Schematic diagram of Hopfield network with 5 neurons

In discrete time $t = 0$, the outputs of neurons are determined by the network input $p_1 \dots p_n$. Then, in every following time step $t = 1, 2 \dots$, the output of a randomly chosen neuron x_j is updated using the formula

$$x_j = \mathrm{H}\left(\sum_{i=1}^{n} x_i w_{ij} + \vartheta_j\right), \tag{1.17}$$

where H is the hard-limiter function (see 1.14) and ϑ_j is the bias which mimics threshold and external excitation of j-th neuron. It is possible to show, see Hopfield (1982), that if the synaptic weights are symmetric and diagonal weights are non-negative – $\forall i, j : w_{ij} = w_{ji}$ and $\forall i : w_{ii} \geq 0$, this dynamics will lead the network to a stable state so that after some time there will be no changes in neuron outputs.

Now, we will use the Hebb's rule to assign weights so that the stable states of the network would be the desired states. There might be more variants of learning rules. We will describe the original rule used in Hopfield (1982) motivated by Hebb (1949).

Let us have T training patterns $t^1, t^2 \dots t^T$ which we would like to be the stable states of the network. The individual outputs of neurons in the training pattern t^k are denoted as $t_1^k t_2^k \dots t_n^k$. Then we will setup every weight using the formula

$$w_{ij} = \begin{cases} \dfrac{1}{n} \displaystyle\sum_{k=1}^{T} \left(2t_i^k - 1\right)\left(2t_j^k - 1\right) & \text{for } i \neq j \\ 0 & \text{for } i = j. \end{cases} \tag{1.18}$$

It is possible to show that the patterns learned by the equation 1.18 are stable states of the network if they are orthogonal or pseudo-orthogonal (Hopfield, 1982; Kvasnicka et al., 1997). However, there might be more stable states than those learned. It is easy to see that the learning rule does not make any difference between pattern and its inverse, thus the inverse patterns are also stable states. There are also different linear combinations of odd number of patterns that might be stable states. These "unwanted" stable states are called phantom (or spurious) patterns. See Figure 1.9 for an example of network's patterns and phantom patterns.

```
....xx....  xxxxxxxx..  xxx....xxx  xx......xx
...xxxx...  xxxxxxxx..  xxx....xxx  xxx....xxx
..xxxxxx..  xx........  xxx....xxx  .xxx..xxx.
.xxx..xxx.  xx........  xxx....xxx  ..xxxxxx..
xxx....xxx  xxxxxxxx..  xxx....xxx  ...xxxx...
xxx....xxx  xxxxxxxx..  xxx....xxx  ...xxxx...
xxxxxxxxxx  xx........  .xxx..xxx.  ..xxxxxx..
xxxxxxxxxx  xx........  ..xxxxxx..  .xxx..xxx.
xxx....xxx  xxxxxxxx..  ...xxxx...  xxx....xxx
xxx....xxx  xxxxxxxx..  ....xx....  xx......xx
```

(a) Network patterns

```
xxxx..xxxx  ....xx....  ........xx  xxxxxxxx..
xxx....xxx  ...xxxx...  ........xx  xxxxxxxx..
xx......xx  ..xxxxxx..  ..xxxxxxxx  xx........
x...xx...x  ..xxxxxxx.  .xxx..xxxx  x...xx....
...xxxx...  ........xx  xxx....xxx  ...xxxx...
...xxxx...  ........xx  xxx....xxx  ...xxxx...
..........  x.xxxxxxxx  .xxxxxxxxx  x.........
..........  xxxxxxxxxx  ..xxxxxxxx  xx........
...xxxx...  xxx....xxx  ........xx  xxxxxxxx..
...xxxx...  xxx....xxx  ........xx  xxxxxxxx..
```

(b) Phantom patterns

Figure 1.9: Example of network patterns and phantom patterns. The four patterns displayed in (a) were stored to the network of 100 neurons. Some of the phantom patterns that were also stable states of the model are shown in (b).

There are several variants of the model. The dynamics of update presented here is called *asynchronous* – at every time just one neuron is picked up and its output is updated. In *synchronous* dynamics the states of all neurons are updated at the same time. However, this change in dynamics may cause that the network's evolution does not have to end in a stable state and may be left in a cycle. It is also possible to show that if the weight matrix is symmetric the network will end in a stable state or in a cycle with just two different states (Sima and Neruda, 1996).

The capacity of the Hopfield model was studied in (Hopfield, 1982; Wilson, 1999; Kvasnicka et al., 1997). The maximal capacity of the model was estimated between $0.13N$ and $0.15N$, where N is the number of neurons in the network. It is also expected that the number of active neurons in every pattern is close to 50%. This result is also considered to be the capacity of auto-associative memory in general (Wilson, 1999).

Chapter 2

Goals of the Thesis

Initial motivation for our work were the papers Tsodyks and Markram (1997); Tsodyks et al. (1998) where the dynamics of detailed neural synapse was presented. It was shown also in biological experiments that the time window for the change of the synaptic efficiency could be as low as few milliseconds having a recovery in range of seconds (Tsodyks et al., 1998). Also, the hysteresis shown in Figure 1.5 provides good enough reason to expect that the excitability of the neural cell can change in such a short time. This offers the possibility to include the changes of synaptic weights into recall phase of neural network models.

Our goal was to find some improvement of the existing models that will be extended by this dynamics and will improve the model performance. First, we have build our model on Hodgkin and Huxley equations (Hodgkin and Huxley, 1952) with the synapse implemented according Tsodyks et al. (1998). We have observed improvements in capacity and further we wanted to simplify our model as much as possible for further analysis and simulations while keeping its improved properties. We ended up with a combination of Willshaw and Hopfield models which we further studied in more detail. Our main results were published in papers which are attached to this thesis. Similar topics were also studied in Torres et al. (2002, 2007) with a focus on synaptic depression.

Chapter 3

Neural Network with Cyclic Activation

We have defined a new mathematical model of the neural network motivated by some of the already existing artificial neural network models. The presented neural network model works as auto-associative memory and it has unique pattern encoding properties.

We have presented the model encoding the patterns as cycles in Stroffek et al. (2007) which we call the *regular model* in this thesis. We have extended this model with the short-term potentiation dynamics and have investigated how this influences the memory capacity in Stroffek and Marsalek (2012) which we refer to as *extended model* later in this thesis.

The presented model was implemented as computer program in C++ and several computer simulations have been tested on a regular desktop computer with the focus on network capacity. There were various computer configurations used. Example of one of these configurations is mentioned in section 3.3. We have generated the desired number of patterns that we have stored to the network. In subsequent steps, we have checked the subset of stored patterns for retrieval. Additionally, the newly generated patterns not stored to the network have been tested for retrieval. It was desired to get no response on patterns not stored to the network. The execution of the tests was running in the range of weeks per one complete test.

3.1 Model Definition

The model consists of N neurons. Each neuron is connected with all the others and also self connections are present. The output of the neuron is calculated according the equation 1.2 and therefore

$$x_j = \mathrm{H}\left(\sum_{i=1}^{N} w_{ij} x_i + \vartheta\right), \tag{3.1}$$

where H is hard-limiter function, x_j denotes the output of the j-th neuron, w_{ij} denotes the weight of the synaptic connection from the i-th neuron to the j-th neuron, ϑ denotes the bias which is the same for all neurons. The synaptic weights and also the outputs of neurons attain only binary values 0 or 1. If the neuron output is 0, the corresponding neuron is called inactive. If the neuron output is 1, the corresponding neuron is called active.

The evolution of the model is described in discrete time steps $t = 0, 1, 2, \ldots$. The outputs of neurons at the time $t = 0$ are given by the presented pattern. In every time step iteration t, the values of all the neurons are computed using the equation (3.1) from the outputs of neurons from a previous time iteration $t - 1$.

The extended model has additionally 3-state synapses simulating a short-term synaptic potentiation. The synapses for the extended version of the model attain values $0, 1$ and P. The learning rule produces only synapses with values 0 and 1. See the sections 3.6 and 3.7 for reference on how the short term potentiation is performed during pattern recall.

3.2 Learning Rule

The patterns stored into the network are not static, but they are cycles composed of low activities in the subsequent discrete time-frames. The patterns are stored as cycles of the activity of a very small number of neurons, which can be as low as 0.1 % of all neurons if the number of neurons is sufficiently large. One *pattern* is represented by one cycle. The synchronous activity at the exact time frame in the cycle is called a *sub-pattern*.

We randomly generate the patterns as described in Stroffek et al. (2007). Let us denote N the number of neurons in the network. In a simplified view, in the formalism above, the sub-pattern is the vector of N elements, $(x_1, x_2, \ldots, x_N)$, where $x_i \in \{0, 1\}$. The pattern is a sequence of sub-patterns.

We define a *relative activity* as the ratio of the number of active neurons in the whole pattern cycle to the total number of neurons. We denote the sub-pattern relative activity as a. Briefly, $a = A/N$, where A is number of active neurons in the sub-pattern. Obviously, $a \in (0, 1)$. We assume that a is a constant number near 0 and we assume that is is the same for all the sub-patterns stored into the network.

Let us denote the length of i-th pattern as $l(i)$, the successive sub-patterns

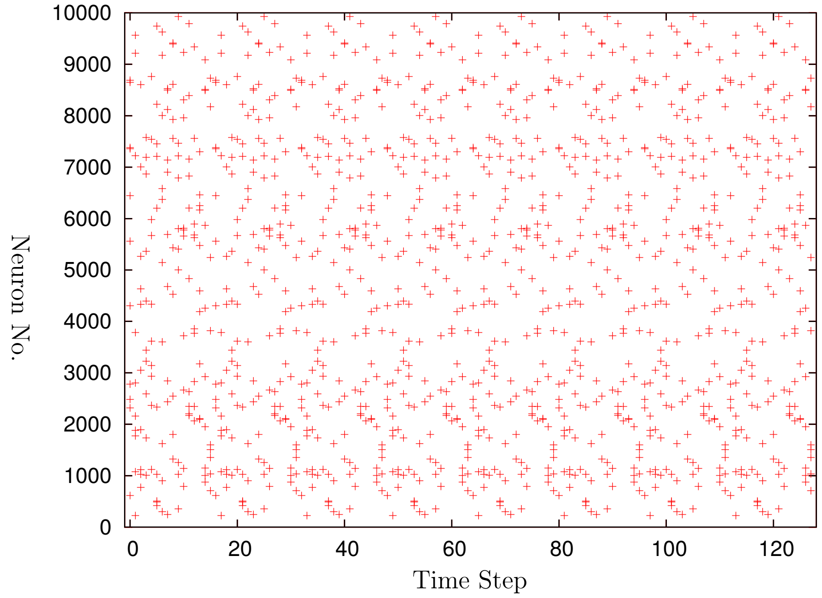


Figure 3.1: Example of the network's pattern. The pattern activity is 1.5% and sub-pattern activity is 0.1%. The cycle is repeated eight times. A cross denotes the activity of the corresponding neuron at the given time step.

of the i-th pattern as ${}^{i}t^{1}, {}^{i}t^{2} \dots {}^{i}t^{l(i)}$ and the activity of n neurons in j-th sub-pattern as ${}^{i}t^{j}_{1}, {}^{i}t^{j}_{2} \dots {}^{i}t^{j}_{n}$. In a nutshell, the lower right index denotes the neuron number, upper left and upper right indices denote pattern and sub-pattern numbers, respectively.

Now, let us have p patterns which we would like to store to the network. In each k-th iteration of the learning process the network learns the k-th pattern (a cycle of sub-patterns) using the modification of the Hebb's rule. For every pattern, we would like to activate the connections from the neurons active in one subpattern leading to those active in the following sub-pattern.

Let us denote weights after the k-th iteration as w^{k}_{ij}, we start with $w^{0}_{ij} = 0$ for all i and j. In each iteration we modify the weights according the equation

$$w^{k}_{ij} = \max\left(w^{k-1}_{ij}, \mathrm{H}\left({}^{k}t^{l(k)}_{i} \; {}^{k}t^{1}_{j} + \sum_{q=1}^{l(k)-1} {}^{k}t^{q}_{i} \; {}^{k}t^{q+1}_{j} \right)\right). \tag{3.2}$$

This gives a value of 0 or 1 to every synaptic weight. If the value of the weight is 1, the corresponding synapse is called active or activated. Otherwise, it is

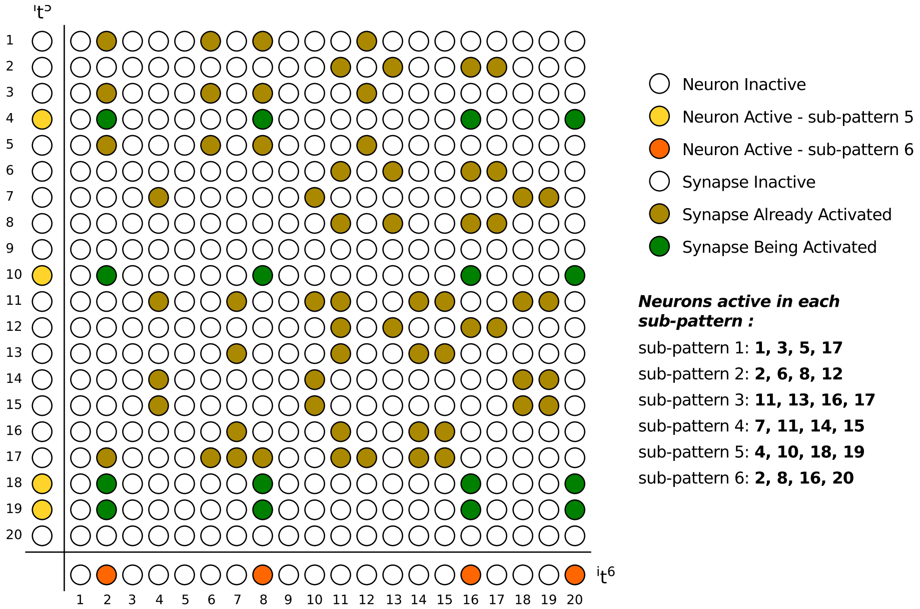

t5
t6
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
Neuron Inactive
Neuron Active - sub-pattern 5
Neuron Active - sub-pattern 6
Synapse Inactive
Synapse Already Activated
Synapse Being Activated
Neurons active in each sub-pattern :
sub-pattern 1: 1, 3, 5, 17
sub-pattern 2: 2, 6, 8, 12
sub-pattern 3: 11, 13, 16, 17
sub-pattern 4: 7, 11, 14, 15
sub-pattern 5: 4, 10, 18, 19
sub-pattern 6: 2, 8, 16, 20

Figure 3.2: **Schematics of the learning rule.** The pattern learned here consists of 6 sub-patterns. This cartoon shows how the synapses are activated based on the neuron output values in example sub-patterns 5 and 6. The left column corresponds to neurons in sub-pattern 5, the bottom row corresponds to neurons in sub-pattern 6.

called inactive or not activated. The visualization of the learning rule is shown in Figure 3.2.

We have to set up the bias in a way which facilitates the correct recognition of similar patterns. We have defined a relative activity as a ratio of active neurons in the sub-pattern to the total number of neurons. Later on, we assume that all the patterns have the constant sub-pattern relative activity.

If we require exact matching of patterns, which means that the pattern is recognized only when all the neurons active in the pattern are also active in the network's input, the bias has to be $\vartheta = -\lfloor an \rfloor$, where a is the sub-pattern relative activity and $\lfloor\rfloor$ denotes the nearest lower integer. We define the similarity of sub-pattern A to sub-pattern B as the number of neurons active in both sub-patterns A and B at the same time divided by the number of active neurons in sub-pattern B. The similarity function is not symmetric from its definition and attains only values from the interval $[0, 1]$. Now the sub-pattern presented to the network has to be similar to some stored sub-pattern. Only the neurons active in both sub-patterns (the presented sub-pattern and the corresponding stored sub-pattern) will contribute to excitation of neurons in the next time step. The neurons active in the stored sub-pattern will contribute because the corresponding weights are activated. However, only neurons active in the presented sub-pattern are firing. Thus, some similarity between these sub-patterns is desired. Let us denote α the lowest similarity for which the presented sub-pattern is still recognized as a stored sub-pattern. From this we get $\vartheta = -\lfloor \alpha a n \rfloor$.

The extended model has short-term synaptic potentiation and the bias have to be set that only potentiated synapses are able to recall the subsequent sub-pattern in a cycle and thus $\vartheta = -\lfloor \alpha a n P \rfloor$.

3.3 Learning Rule Optimization

It can be seen from the equation 3.2 that the the learning algorithm's time complexity is the square of the number of neurons multiplied by the number of sub-patterns of all the learned patterns,

$$O(n^2 \sum_{i=1}^{r} l(i)), \tag{3.3}$$

where O denotes the asymptotic upper bound, r is the number of patterns and $l(i)$ is the number of sub-patterns of i-th pattern. However, if the value of the neuron in the learned pattern is 0 it can not modify any weights. Thus, it is sufficient to perform a learning only using non zero values of neurons.

For an optimal performance the sub-patterns may be represented as an array of active neurons and during a learning process the weights corresponding to neurons active in the first sub-pattern and the neurons active in the following sub-pattern should be modified.

Because the sums in the equation 3.2 are counted only for neurons active in the sub-pattern, the time complexity of optimized version of learning algorithm is

$$O(a^2 n^2 \sum_{i=1}^{r} l(i)), \tag{3.4}$$

where a is relative sub-pattern activity. The asymptotic complexity have not changed. However we have gained a fixed constant speed improvement and thus the learning algorithm runs $1/a^2$-times faster.

During the computer simulation the generation of 30 patterns with the learning process took 18 minutes. Using an optimized version of the learning process, the learning time was much shorter – it took only 0.3 seconds. The simulations were performed on AMD64 3400+ CPU with 512MB of RAM running Linux, the code was compiled using the gcc compiler. The time of the execution duration was measured using the 'time' command. The generated patterns had a pattern relative activity 2 % a sub-pattern relative activity 0.1 % and the network has 10000 neurons.

3.4 Pattern Generation

The generation of patterns might be done in many different ways. We did it as follows. First we generated the set of neurons active in a whole pattern. Afterwards we decomposed that set to the sub-patterns. The pattern is supposed to provide cyclic activity where the last pattern is supposed to be followed again by the first pattern. Therefore, we often refer to the number of sub-patterns as cycle length. The algorithm of the pattern generation process can be described by the following pseudo-code:

1. Randomly generate a set of neurons to be active in the whole pattern.
2. Calculate the desired number of sub-patterns as the number of active neurons divided by the number of active neurons in a sub-pattern $l = k/[an]$.
3. Calculate the number of active neurons for each sub-pattern. First, let the number of active neurons for each sub-pattern be $[an]$ and afterwards spread the rest of neurons uniformly using an analogy of the Bresenham's algorithm (Bresenham, 1965).
4. Do the pattern decomposition – generate a random distribution of neurons to the sub-patterns depending on the number of active neurons for each sub-pattern.
5. Create the resulting data structure representing the generated pattern.

3.5 Pattern Decomposition

We have a list of k active neurons in the pattern to be decomposed to the sub-patterns and we have the array $A[1..c]$ of integers which stores the number of neurons still missing in each corresponding sub-pattern. The process of distribution of active neurons to the sub-patterns is described by the following steps:

1. Take the next active neuron in the pattern and generate an integer random number
$$r \in \left\{ 0, 1 \ldots \sum_{i=1}^{c} A[i] \right\}.$$

2. Find the sub-pattern for which q is minimal argument of the function

$$q = \arg\min_p \left(\sum_{i=1}^{p} A[i] > r \right).$$

3. Assign the current active neuron into the q-th sub-pattern and decrease $A[q]$ by one.

4. If there is any neuron not assigned to a sub-pattern, go back to the step 1. Otherwise, the decomposition is finished.

5. Finally, we have to create a desired pattern representation. All the steps above are repeated for every pattern.

Finding a number q in the 2nd step takes maximally $\max\{l(i), i \in \{1, 2 \ldots r\}\}$ steps. Thus the time complexity of the pattern decomposition algorithm is

$$O(A \max\{l(i), i \in \{1, 2 \ldots r\}\}), \tag{3.5}$$

where A is the total number of active neurons in the training set.

3.6 Pattern Recall

The pattern recall starts in the discrete time 0 with a clean network where all the neurons are quiet and their outputs are 0. The input to the network is presented as external excitation in several subsequent sub-patterns. There exist one-to-one mapping between the input neuron and the neurons in the network. If the input neuron is active, the corresponding neuron in the network is activated in the given time step. The outputs of all the neurons are calculated synchronously after the external excitation processing is finished. Finally, the potentiation of the synapses is calculated. The pattern recall steps could be written as:

1. Clean all the neuron outputs and start in the discrete time step 0.

2. If the synaptic potentiation is used then mark all the potentiated synapses as active - i.e. the synaptic weights will have the values 0 and 1 produced by the learning rule.

3. If there is an input sub-pattern to be presented go over all the inputs and if the input is active, mark the corresponding neuron in the network active - i.e. set the neuron output to 1.

4. Calculate the outputs of all the neurons synchronously according the equation 3.1.

5. If the synaptic potentiation is used then determine which synapses are going to be potentiated. A synapse will be potentiated if the synapse was activated by the learning rule and the pre-synaptic neuron was active in the previous discrete time step and post-synaptic neuron is active in the current discrete time step.

6. Increase the time step and go back to step 2 and continue.

The pattern is considered to be properly recalled if the desired neural activity remains after the external excitation vanishes. The desired neural activity was tested after activity of several time steps without the external excitation. The number of these time steps was usually the same as the length of the presented pattern.

3.7 Short-Term Potentiation

The model includes the short-term potentiation that has the synaptic weights in three states

Non-activated (N) – No pattern has activated the synapse in the learning rule equation 3.2 and thus the synaptic weight value is 0.

Activated (A) – At least one pattern have activated the synapse and the synaptic weight value is 1.

Strengthened (S) – At least one pattern have activated the synapse and during the pattern recall process, the post-synaptic neuron have fired after the pre-synaptic neuron and thus the synapse was potentiated. The synaptic weight value is $P > 1$.

Pattern recall process started with only activated and non-activated synapses produced by the learning rule equation 3.2. We then used external input and let the network go over the first iteration of the cycle. If two neurons fired in subsequent time steps that the post-synaptic neuron fires after the pre-synaptic neuron and the corresponding synapse is activated, we have strengthened the synapse. This has been done for all the synaptic weights when applicable. The strengthened synaptic weights had value $P > 1$. We have studied the cases for $P \in (1, 2)$.

Chapter 4

Capacity and Computer Simulations

We were interested in the number of patterns that could be stored into the network and compare that with other relevant models. We have done simple theoretical analysis followed by numerous computer simulations.

4.1 Capacity Estimates

Assume first, that the relative sub-pattern activity in the network is a constant small number near 0 and the number of neurons is reaching infinity. We suppose that the network would be operating well until the critical filling of a weight matrix would be reached by the presented learning rule. Let us denote the relative critical filling of a weight matrix as K. Later on, we refer to this simply as to the critical filling. The relative filling of a weight matrix, which was produced by the learning rule, could be calculated as

$$F = 1 - (1 - a^2)^s, \tag{4.1}$$

where s is the total number of sub-patterns learned. The randomly chosen element of a weight matrix is 0 if and only if the two corresponding neurons in every two following sub-patterns do not have both values equal to 1. Now, by comparing the actual and critical fillings we obtain $(1 - a^2)^s > 1 - K$ and after taking a natural logarithm of both sides we get:

$$s < \frac{\ln(1 - K)}{\ln(1 - a^2)}, \tag{4.2}$$

which is the upper bound for sub-pattern storage for the number of neurons approaching infinity. The question of the exact value of the critical filling constant K is still open.

There was done a research of information storage in similar sparsely coded memory nets in Nadal and Toulouse (1990). The authors argue that the maximal information storage is provided by the network with relative filling of 0.5. This means that also critical filling constant K have to be less then 0.5.

Even though our explanation requires that the number of neurons is reaching infinity, it also works for smaller number of neurons. In this case the critical filling constant has to be changed – in general, it will be a function of activity and a number of neurons.

4.2 Pattern Recall Testing

We have performed pattern retrieval tests for both of the models. We have stored certain number of patterns into the model and then we have tested the subset of these patterns for successful retrieval. If the number of stored patterns in the extended model increases and the synaptic efficiency parameter P used for potentiated synapses is increasing, the network is capable of responding positively to the patterns including those that were not learned. Therefore we performed two types of memory tests as follows:

Positive memory test – We had stored the corresponding number of patterns into the network. We had randomly chosen the subset of patterns. We then tested the network's response to these patterns.

Negative memory test – We had stored the corresponding number of patterns into the network. We had randomly generated another set of patterns which we used as input to the network. We expected the network to give no response to these newly generated patterns.

The relative error in the positive memory test is calculated as the number of *incorrectly set* neurons (active or inactive, opposite than it should be), divided by the number of active neurons in the pattern. The relative error in the negative memory test is calculated as the number of *active neurons* in the network response, divided by the number of active neurons in the pattern. The relative error in both cases is rounded off at the value of 2.

4.3 Computer Simulations

We have done one set of computer simulations on the regular desktop computer for each version of the model. The goal of the simulations on the regular version was to investigate the capacity dependence on the pattern and sub-pattern activities and the number of neurons. The simulation results for the regular model are shown in Figures 4.1 and 4.2.

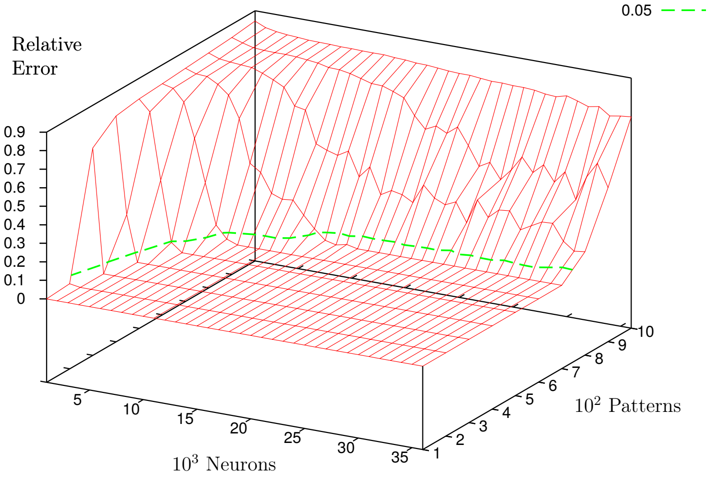


Figure 4.1: The results of the computer simulations of the regular model for pattern activity 10%, sub-pattern activity 1%, 40 patterns were tested for successful retrieval.

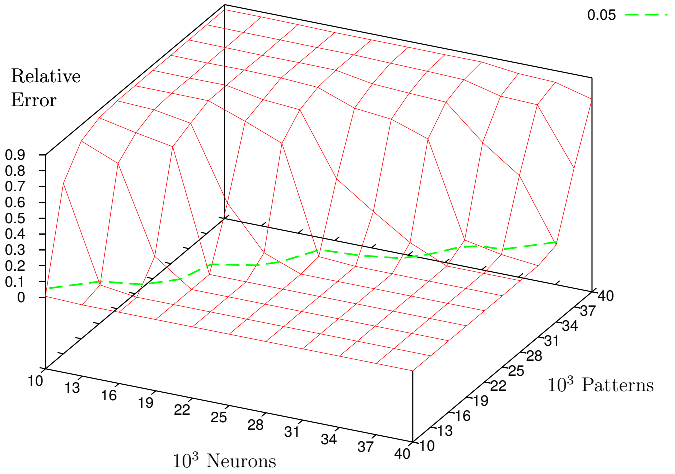


Figure 4.2: The results of the computer simulations of the regular model for pattern activity 1.5%, sub-pattern activity 0.1%, 200 patterns were tested for successful retrieval.

We have chosen the two pattern and sub-pattern activities we have performed the simulations on. On both of these activities we have done a series of the positive memory tests as described in chapter 4 and section 3.6. The activities chosen for simulations were 10% pattern activity with 1% sub-pattern activity and 1.5% pattern activity with 0.1% sub-pattern activity.

For the pattern activity of 10% we have done simulations on the networks from 10000 neurons up to 40000 neurons with the step of 3000 neurons. For each of these networks we have done a series of tests where the different number of patterns was stored to the network. The range of patterns stored was 100 up to 1000 patterns. We have tested 40 patterns for retrieval for every number of patterns stored. The results of these simulation are shown in Figure 4.1.

Similarly, for the pattern activity of 1.5% we performed simulations on the networks from 10000 neurons up to 40000 neurons with the step of 3000 neurons. For each of these networks we have stored from 10000 patterns up to 40000 patterns. We have tested 200 patterns for retrieval. The results are shown in Figure 4.2.

It can be seen from the regular model simulation shown in Figure 4.2 that the network with 40 thousand neurons can hold up to 34 thousand patterns of a pattern relative activity of 1.5% and a sub-pattern relative activity of 0.1%. This corresponds to the weight matrix filling ratio of 45.119%.

Extended version of the model brings one more additional parameter. We wanted to investigate it's affect on memory capacity. Therefore we have chosen the fixed number of neurons and activities used in previous testing. We used the network of 10000 neurons with pattern activity 1.5% and sub-pattern activity 0.1%. We have tested the storage of 10000 patterns up to 200000 patterns with the step of 5000 patterns. The parameter P was tested in the range between 1 and 1.8 with the step of 0.1. After the learning process 20 patterns were tested for successful retrieval.

Additionally we used another set of randomly generated patterns that were not stored. These patterns were tested for retrieval. It is desirable that the network gives no response. There is a thick contour line drawn for the relative error of 0.01 (1 %). The local minimum is visible for $P = 1.5$ in the surface and the contour. Therefore, the highest number of patterns could be stored into the network for $P = 1.5$.

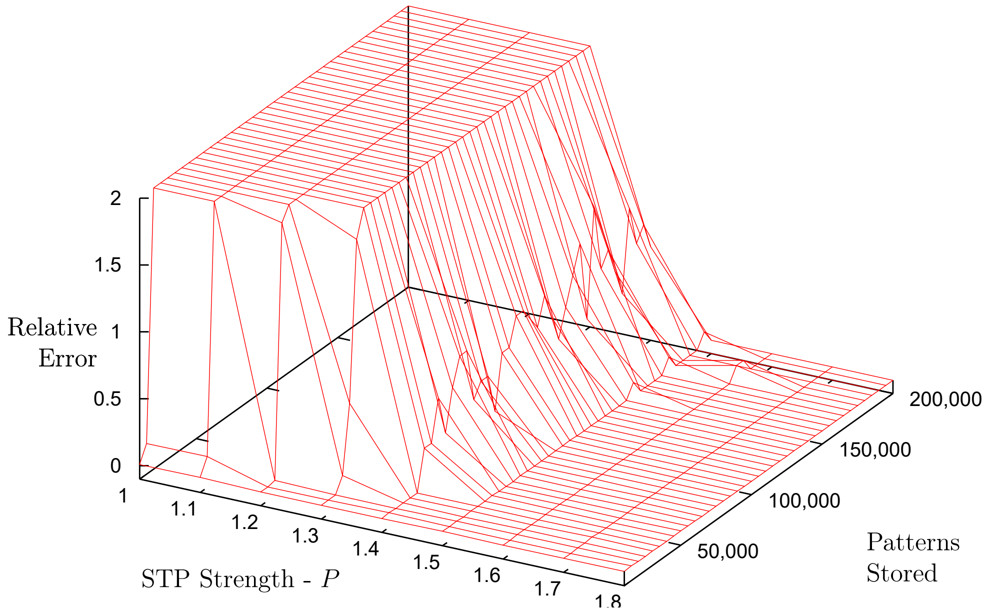


Figure 4.3: The results of the computer simulations of the positive memory test of the extended model.

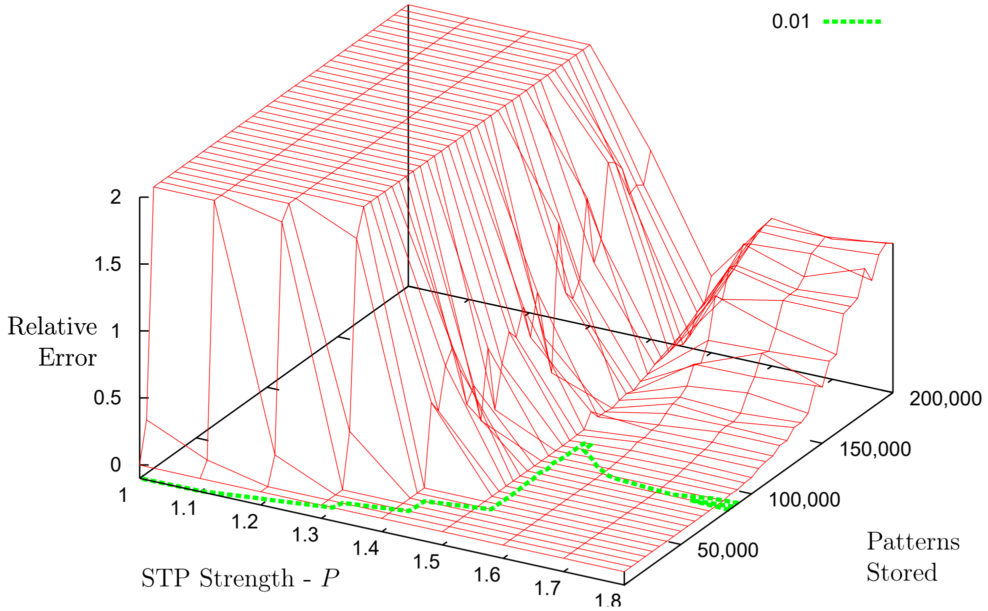


Figure 4.4: The results of the computer simulations of the negative memory test of the extended model.

Chapter 5

Discussion

5.1 Understanding the Results

We have presented very simple neural network as a combination of Willshaw and Hopfield models having binary neuron outputs as well as binary synaptic weights. We have changed the way how patterns are encoded and we additionally have extended the model with simple synaptic potentiation. We tried to keep the neural activity in the patterns at physiologically observed levels of $1 - 2\%$ (Wilson, 1999).

The above changes allowed us to store around 10-times more patterns into the network than its number of neurons and the model still worked well on pattern recall process. The Hopfield model is considered in general as the appropriate model of auto-associative memories in general. Its capacity was briefly discussed in section 1.4.3 and roughly it is $0.15N$ where N is the number of neurons in the network (Wilson, 1999; Kvasnicka et al., 1997). Thus, we have been able to store and retrieve more than 66-times more patterns. It is therefore very likely, that the memory capacity in the brain structures could be much higher than it was previously expected (Wilson, 1999; Treves and Rolls, 1991; Rolls and Treves, 1998). The high capacity auto-associative memories were studied in Davey et al. (2004). The studied models were based on Hopfield network with various learning rules. The highest capacity observed was $2N$ by the perceptron based learning rule (Davey et al., 2004; Krauth and Mezard, 1987). There are very limited resources that tries to measure possibility of pattern storage in the brain (Voss, 2009) and it is also not clear what exact brain areas are trained to give a proper response.

We refer to the level of activity in our model as sparse. There were more detailed studies of sparsely coded neural networks (Amari, 1989) which showed that the capacity of sparsely coded associative memory is $n^2/(2\log n)$. However, this expects that the level of activity $\lim_{n\to\infty} a = 0$. This does not correspond to physiologically observed level of neural activity (Wilson, 1999).

The idea of pattern decomposition into sub-patterns was also tried in Karbasi et al. (2014). They have generated the pattern which they later decomposed into sub-patterns of activities. In their case, the decomposition was intentionally done in a way that the same neurons were active in multiple sub-patterns. They also claim that the capacity of the model is exponential in the number of neurons. However, this is not discussed in more detail and it would be hardly achievable without additional assumptions on the pattern properties.

The analysis of network capacity in Stroffek et al. (2007); Stroffek and Marsalek (2012); Golomb et al. (1990) could help to give estimates on the number of patterns stored in biological neural networks. It is possible to have estimate of the connectivity between neurons in the specific brain area as well as to observer the level of neural activity. This will help to understand how many patterns could be stored in particular area.

Experiments with the capacity showed that there is a local maximum for the potentiation parameter P at the value of 1.5, see Figure 4.4. This was caused by a balance between positive and negative memory tests. In general, it is good to understand that for short-term synaptic potentiation there might exist an optimal value how much the synaptic weight should be increased.

5.2 Model Enhancements

5.2.1 Fast and Slow Patterns

We have presented two versions of the model where the pattern encoding is similar but the pattern recall processes are a bit different. In regular model, it is possible to re-construct the whole cycle just by presenting any of the sub-patterns. The recall of this pattern is fast. The extended version of the model needs the whole pattern to be presented so the corresponding synapses will get potentiated. The recall of this type of pattern takes much longer time and the whole pattern cycle have to be presented for successful recall. However, we could gain much higher capacity in comparison with regular model.

It is possible to make one model which could have these two types of recall mixed. We would call the patterns to be recalled as in regular model as fast patterns and patterns to be recalled as in extended version of the model as slow patterns. We would have the 3-state synapses already produced by the learning rule.

We would keep the original learning rule 3.2 to store the slow patterns and we would modify the learning rule to store the fast patterns. We would keep the short-term potentiation dynamics during pattern recall as in extended model. The synaptic weights that already have the value of P will not get potentiated. The learning rule equation 3.2 would be changed to store the fast patterns as

$$w_{ij}^{k} = \max\left(w_{ij}^{k-1}, \mathrm{H}\left({}^{k}t_i^{l(k)}\;{}^{k}t_j^{1} + \sum_{q=1}^{l(k)-1} {}^{k}t_i^{q}\;{}^{k}t_j^{q+1}\right)P\right). \qquad (5.1)$$

5.2.2 Polychronization

In our case, the model behaves as polychronization network having the constant delays in signal reaching the post-synaptic neuron. Therefore, the encoding of the presented model could be understood as simplified polychronization where the model was having equal synaptic delays and synchronous firing across all the neurons. However, the model could be further altered and different synaptic delays could be introduced. This could bring the realistic polychronization behavior and the overlap of active neurons in different patterns could be higher.

5.2.3 Inhibition

The most common case when the pattern will fail to be recalled correctly is the network activity divergence. The number of active neurons started to increase in the subsequent time steps and finally all the neurons in the network were firing at the same time.

There is always certain ratio of inhibitory neurons in real biological neural structures (Freund and Buszaki, 1996; Hasselmo et al., 1995) that are not present in the model. We tried to enhance the model with inhibitory dynamics that would become a vital part of the pattern recall process. We have done few variants but we have not succeeded to fulfill our original intention.

It would be possible to add the certain number of inhibitory neurons and set up their parameters that the inhibitory dynamics would be triggered by the activity significantly exceeding the desired sub-pattern activity. The inhibitory neurons would then fire and would make all the excitation neurons not to fire. This type of inhibitory dynamics would be beneficial in biological network or biological like usage of the network. However, this enhancement would most likely not improve anyhow the memory properties of the model.

5.3 Artificial Neural Network

We have implemented the feed forward neural network application for vehicle mass category recognition (Stroffek et al., 2010). The data were provided by the laser scanner as 3 hour long two lane highway recording. The goal was to give an online estimate of the mass category for the passing vehicle, e.g. between $3,5t$ and $7,5t$. The vehicles were passing the gantry with the laser scanner under the full speed. There were totally $2,367$ vehicles identified in the recording. We used $1,241$ vehicles directly for learning and the remaining $1,126$ vehicles were used for additional testing purposes. The best classification achieved by the neural network model was 95.86%.

5.4 Memory and Information

Everybody intuitively understands what is the information, but how it can be formally defined? The proper formal definition of general information is not known. There exist some non-exhaustive definitions from the certain point of view. Claude Shannon introduced his information theory in Shannon (1948) which was based on the probability theory. His quantification of information and information entropy could be used to effectively transmit a message across the noisy channel.

The concept of information usually seems to us intuitively well understandable in a certain context. Information might be written on a sheet of paper, might be spoken by someone, might be sensible to our perception and might be encoded in a certain sequence. We will call the entity holding the information a *message*. To receive information we always have to know the correct interpretation of the message.

When we talk about some information, we usually know already the interpretation of the corresponding message or we have an idea about the interpretation. We receive large number of messages through our perception every second. Based on our known interpretations of the messages, we always choose the messages which we think have some valuable information for us. This gives us an ability to talk and understand the information intuitively in a certain context without a need for a formal definition.

In most cases today, the information message is a stream of bytes. The interpretation of the message might differ and the information is useless if we do not know the correct interpretation or semantics. However, we can still compress the data in the message (decrease the number of bits required to encode the message) without knowing the interpretation due to Shannon's understanding of information.

Biological organisms are more complex than computers. What is the information in a context of a biological organism? If I would pick up a pen and hold it one meter above the ground and then I would release it. Is the information contained in a fact that I had released a pen? Would the pen interpret that information and would start falling? Intuitively no because the pen's fall would just happen because of physical laws and would not depend on the pen's will. When acetylcholine neuromediator reaches a post-synaptic membrane on a muscle fiber it is a signal for a fiber contraction. Is the information contained in transmitting an acetylcholine? Intuitively we would say yes. However, the process of neuromediator transmission and muscle contraction is also a well observed sequence of steps that just come because of physical laws. Why there is a difference in our understanding of information in these two example cases? It seems that the subject receiving an information makes a difference for our understanding of information. We know that a pen is not capable of understanding any information. Probably, post-synaptic membrane does just its job according physical laws without any understanding. However, intuitively we assume that there is someone who will understand that information and will interpret it correctly.

Chapter 6

Conclusions

Our goal was to find some improvement of the existing models utilizing the dynamics of short term synaptic potentiation. We have described the artificial neural network model that uses cyclic activities as patterns that are stored and retrieved from the network. We think that these cyclic patterns are in many cases more physiological then static patterns.

We additionally extended the model with short-term potentiation which was used as vital part of the pattern recall process. We have shown that the memory capacity significantly increases in case when the ratio of active neurons is around $1 - 2\%$ which corresponds to physiologically observed values. We have done simple theoretical analysis followed by the computer simulations.

It has been shown that the short-term potentiation occurs in time range that could directly influence the pattern recall processes in biological neural networks (Tsodyks and Markram, 1997; Tsodyks et al., 1998). We have shown that it is possible to find the case where these properties can rapidly improve the behavior of the artificial neural network model.

We think that there might be more cases where the synaptic dynamics of pattern recall process could improve the model performance. It might be possible that evolution managed to find more use cases for this phenomenon that still might be revealed.

Appendix A

Published Papers

A.1 Pattern Storage in a Sparsely Coded Neural Network with Cyclic Activation

Štroffek, J., Maršálek, P. and Kuriščák, E. (2007). Pattern storage in a sparsely coded neural network with cyclic activation, *BioSystems*, 89(1-3), pages 257–263. **IF(2008)=1.477.**

The attached paper was removed from the thesis as we do not have the permissions to freely provide the journal papers. Readers interested in the paper can find it online at

`https://www.sciencedirect.com/science/article/pii/S0303264706002656`

A.2 Short-Term Potentiation Effect on Pattern Recall in Sparsely Coded Neural Network

Štroffek, J. and Maršálek, P. (2012). Short-term potentiation effect on pattern recall in sparsely coded neural network, *Neurocomputing*, 77(1), pages 108–113. **IF(2009)=2.126.**

The attached paper was removed from the thesis as we do not have the permissions to freely provide the journal papers. Readers interested in the paper can find it online at

`https://www.sciencedirect.com/science/article/pii/S0925231211005182`

A.3 Biological Context of Hebbian Learning in Artificial Neural Networks, a Review

Kuriscak, E., Marsalek, P., Stroffek, J. and Toth P. G. (2015). Biological Context of Hebbian Learning in Artificial Neural Networks, a Review, *Neurocomputing*, 152, pages 27–35. **IF(2009)=2.126**

The attached paper was removed from the thesis as we do not have the permissions to freely provide the journal papers. Readers interested in the paper can find it online at

`https://www.sciencedirect.com/science/article/pii/S0925231214015239`

A.4 Highway Toll Enforcment: Real-Time Classification of Motor Vehicles

Štroffek, J., Kuriščák, E. and Maršálek, P. (2010). Highway toll enforcement: Real-Time Classification of Motor Vehicles, *IEEE Vehicular Technology Magazine*, 5(4), pages 56–65.

The attached paper was removed from the thesis as we do not have the permissions to freely provide the journal papers. Readers interested in the paper can find it online at

`https://ieeexplore.ieee.org/document/5641642/`

Author's Publications

Publications with IF

1. Štroffek, J., Maršálek, P. and Kuriščák, E. (2007). Pattern storage in a sparsely coded neural network with cyclic activation. *BioSystems*, 89(1-3), pp. 257–263. **IF(2007)=1.646.**

2. Štroffek, J., Kuriščák, E. and Maršálek, P. (2010). Highway toll enforcement: Real-Time classification of motor vehicles. *IEEE Vehicular Technology Magazine*, 5(4), pp. 56–65. **IF(2010)=1.184.**

3. Štroffek, J. and Maršálek, P. (2012). Short-term potentiation effect on pattern recall in sparsely coded neural network. *Neurocomputing*, 77(1), pp. 108–113. **IF(2012)=1.634.**

4. Kuriščák, E., Maršálek, P., Štroffek, J. and Wünsch, Z. (2012). The effect of neural noise on spike time precision in a detailed CA3 neuron model. *Computational and Mathemathical Methods in Medicine*, pp. 1–12. **IF(2012)=0.791.**

5. Kuriscak, E. and Marsalek, P. and Stroffek, J. and Toth, P. G. (2015). Biological Context of Hebbian Learning in Artificial Neural Networks, a Review. *Neurocomputing*, 152, pp. 27–35. **IF(2015)=2.083.**

Other Publications

1. Štroffek, J. and Kuriščák, E. (2005). Associative neural network with cyclic patterns, *Proceedings of the 5th workshop: Cognition and artificial life*, Editors: Kelemen, J., Kvasnička, V., Pospíchal, J., pp. 541–548.

2. Maršálek, P. and Štroffek, J. (2006). Sound localization neural pathway encoding in humans *Proceedings of the 6th workshop: Cognition and artificial life*, Editors: Kelemen, J., Kvasnička, V., pp. 283–288.

3. Štroffek J, Maršálek P and Kuriščák E (2008). Real time classification of motor vehicles in toll enforcement *Proceedings of the 8th workshop: Cognition and artificial life*, Editors: Kelemen, J., Kvasnička, V., Pstružina, K., pp. 371–323.

Bibliography

1. Abbott, L. F. Lapicque's introduction of the integrate-and-fire model neuron (1907). *Brain Research Bulletin*, 50:303–304, 1999.

2. Amari, S. Characteristics of sparsely encoded associative memory. *Neural Networks*, 2(6):451 – 457, 1989. ISSN 0893-6080. doi:https://doi.org/10.1016/0893-6080(89)90043-9.

3. Amit, D. J. *Modeling Brain Function. The World of Attractor Neural Networks*. Cambridge University Press, Cambridge, 1989.

4. Amit, D. J.; Gutfreund, H.; and Sompolinski, H. Information storage in neural networks with low level of activity. *Physical Review A*, 35:2293–2303, 1987.

5. Atkinson, R. C. and Shiffrin, R. M. Human memory: A proposed system and its control processes. volume 2 of *Psychology of Learning and Motivation*, pages 89 – 195. Academic Press, 1968.

6. Bresenham, J. E. Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30, 1965.

7. Brette, R. and Gerstner, W. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J Neurophysiol*, 94:3637–3642, 2005.

8. Brown, T. H. and Zador, A. M. Hippocampus. In Shepherd, G. M., editor, *The Synaptic Organization of the Brain*, pages 346–388. Oxford University Press, New York, 1990.

9. Davey, N.; Hunt, S.; and Adams, R. High capacity recurrent associative memories. *Neurocomputing*, 62:459 – 491, 2004. ISSN 0925-2312. doi:https://doi.org/10.1016/j.neucom.2004.02.007.

10. Fehlberg, E. Klassische runge-kutta-formeln fünfter und siebenter ordnung mit schrittweiten-kontrolle (German) [Classical fifth- and seventh-order Runge–Kutta formulas with stepsize control]. *Computing*, 4(2):93–106, 1969.

11. Fehlberg, E. Klassische runge–kutta-formeln vierter und niedrigerer ordnung mit schrittweiten-kontrolle und ihre anwendung auf wärmeleitungsprobleme. (German) [Classical Runge–Kutta formulas of fourth and lower order with stepsize control and their use for heat problems]. *Computing*, 6(1–2):61–71, 1970.

12. FitzHugh, R. Mathematical models of threshold phenomena in the nerve membrane. *Bulletin of Mathematical Biology*, 17:257–278, 1955. ISSN 0092-8240.

13. Freund, T. F. and Buszaki, G. Interneurons of the hippocampus. *Hippocampus*, 6:347–470, 1996.

14. Gerstner, W. and Kistler, W. M. *Spiking Neuron Models.* Cambridge University Press, Cambridge, 2002.

15. Golomb, D.; Rubin, N.; and Sompolinsky, H. Willshaw model: Associative memory with sparse coding and low firing rates. *Phys Rev A.*, 41(4):1843–1854, 1990.

16. Gorchetchnikov, A. and Grossberg, S. Space, time, and learning in the hippocampus: How fine spatial and temporal scales are expanded into population codes for behavioral control. *Neural Networks*, 20(2):182–193, 2007.

17. Guttman, R.; Lewis, S.; and Rinzel, J. Control of repetitive firing in squid axon membrane as a model for a neuroneoscillator. *Journal of Physiology*, 305:377–395, 1980.

18. Hasselmo, M. E.; Schnell, E.; and Barkai, E. Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in rat hippocampal region ca3. *J. Neurosci*, 15:5249–5262, 1995.

19. Hebb, D. O. *The organization of behavior – a neuropsychological theory.* John Wiley & Sons, 1949.

20. Hodgkin, A. L. and Huxley, A. F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, 117:500–544, 1952.

21. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *P. Natl. Acad. Sci. USA*, 79:2554–2558, 1982.

22. Hopfield, J. J. Neurons with graded response have collective computational properties like those of 2-state neurons. *P. Natl. Acad. Sci. USA*, 81:3088–3092, 1984.

23. Izhikevich, E. M. Polychronization: Computation with spikes. *Neural Computation*, 18:245–282, 2006.

24. James, W. *Psychology - The Stream of Consciousness*, chapter IX. Cleveland & New York, 1892.

25. Jeffress, L. A. A place theory of sound localization. *J Comp Physiol Psychol.*, 41:35–39, 1948.

26. Karbasi, A.; Salavati, A.; and Shokrollahi, A. Convolutional neural associative memories: Massive capacity with noise tolerance. *CoRR*, abs/1407.6513, 2014.

27. Koch, C. *Biophysics of computation – Information Processing in Single Neuron.* Oxford Univeristy Press, New York, 1999.

28. Kohonen, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

29. Krauth, W. and Mezard, M. Learning algorithms with optimal stability in neural networks. *Journal of Physics A: Mathematical and General*, 20(11):L745, 1987.

30. Kuriscak, E.; Marsalek, P.; Stroffek, J.; and Toth, P. G. Biological context of hebbian learning in artificial neural networks, a review. *Neurocomputing*, 152:27–35, 2015.

31. Kuriscak, E.; Marsalek, P.; Stroffek, J.; and Wünsch, Z. The effect of neural noise on spike time precision in a detailed ca3 neuron model. *Computational and Mathematical Methods in Medicine*, 2012.

32. Kvasnicka, V.; Benuskova, L.; Pospichal, J.; Farkas, I.; Tino, P.; and Kral, A. *Introduction into The Theory of Neural Networks.* IRIS, 1997. Written in Slovak.

33. Lapicque, L. Recherches quantitatives sur l'excitation electrique des nerfs traitee comme une polarization. *J. Physiol. Pathol. Gen.*, 9:620–635, 1907.

34. Marsalek, P. Coincidence detection in the Hodgkin–Huxley equations. *Biosystems*, 58:83–91, 2000.

35. Marsalek, P. and Lansky, P. Proposed mechanisms for coincidence detection in the auditory brainstem. *Biological Cybernetics*, 92:445–451, 2005.

36. Nadal, J. P. and Toulouse, G. Information storage in sparsely coded memory nets. *Network*, 1:61–74, 1990.

37. Pavlov, I. P. *Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex.* Oxford University Press, London, 1927.

38. Rall, W. Distinguishing theoretical synaptic potentials computed for different soma-dendritic distributions of synaptic input. *Journal of Neurophysiology*, 30:1138–1168, September 1967.

39. Rall, W.; Burke, R. E.; Smith, T. G.; Nelson, P. G.; and Frank, K. Dendritic location of synapses and possible mechanisms for the monosynaptic epsp in motoneurons. *Journal of Neurophysiology*, 30:1169–1193, September 1967.

40. Rolls, E. T. and Treves, A. *Neural Networks and Brain Function.* Oxford University Press, New York, 1998.

41. Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. Learning representations by back-propagating errors. *Nature*, pages 533–536, 1986.

42. Rybar, J.; Benuskova, L.; and Kvasnicka, V., editors. *Cognitive Sciences.* Kalligram, Bratislava, 2002. Written in Slovak.

43. Shannon, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423 and 623–656, 1948.

44. Sima, J. and Neruda, R. *Theoretical Issues of Neural Networks.* Matfyz-Press, Prague, 1996.

45. Stroffek, J.; Kuriscak, E.; and Marsalek, P. Highway toll enforcement: Real-time classification of motor vehicles. *IEEE Vehicular Technology Magazine*, 5:56–65, 2010.

46. Stroffek, J. and Marsalek, P. Short-term potentiation effect on pattern recall in sparsely coded neural network. *Neurocomputing*, 77:108–113, 2012.

47. Stroffek, J.; Marsalek, P.; and Kuriscak, E. Pattern storage in a sparsely coded neural network with cyclic activation. *BioSystems*, 89:257–263, 2007.

48. Torres, J.; Cortes, J.; J., M.; and Kappen, H. Competition between synaptic depression and facilitation in attractor neural networks. *Neural Computation*, 19(10):2739–2755, 2007. doi:10.1162/neco.2007.19.10.2739.

49. Torres, J. J.; Pantic, L.; and Kappen, H. J. Storage capacity of attractor neural networks with depressing synapses. *Phys. Rev. E*, 66:061910, 2002. doi:10.1103/PhysRevE.66.061910.

50. Treves, A. and Rolls, E. What determines the capacity of autoassociative memories in the brain? *Network: Computation in Neural Systems*, 2(4):371–397, 1991. doi:10.1088/0954-898X_2_4_004.

51. Tsodyks, M. and Markram, H. The neural code between neocortical pyramidal neurons depends on neurotransmitter release probability. *Proc.Natl. Acad. Sci. USA*, 94(2):719–723, 1997.

52. Tsodyks, M.; Pawelzik, K.; and Markram, H. Neural networks with dynamic synapses. *Neural Comput.*, 10:821–835, 1998.

53. Tuckwell, H. C. and Wan, F. Y. M. Time to first spike in stochastic Hodgkin–Huxley systems. *Physica A*, 351:427 – 438, 2005.

54. Voss, J. Long-term associative memory capacity in man. *Psychonomic Bulletin & Review*, 16(6):1076–1081, 2009. ISSN 1531-5320. doi:10.3758/PBR.16.6.1076.

55. Willshaw, D. J.; Buneman, O. P.; and Longuet-Higgins, H. C. Non-holographic associative memory. *Nature*, 222:960–962, 1969.

56. Wilson, H. R. *Spikes, Decisions, and Actions: The Dynamical Foundations of Neuroscience.* Oxford University Press, New York, 1999.

57. Young, S. R. and Rubel, E. W. Frequency-specific projections of individual neurons in chick brainstem auditory nuclei. *The Journal of Neuroscience*, 3(7):1373–1378, 1983.