DEGREE PROJECT, IN ENGINEERING PHYSICS , FIRST LEVEL
STOCKHOLM, SWEDEN 2014

Implementation and Robustness of Hopfield Networks with Spiking Neurons

EMIL WÄRNBERG

KTH ROYAL INSTITUTE OF TECHNOLOGY

SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION

Implementation and Robustness of Hopfield Networks with Spiking Neurons
Emil Wärnberg
Supervised by Prof. Örjan Ekeberg

Spring 2014

Degree Project in Engineering Physics, First Cycle
SA104X, 15 credits
KTH Royal Institute of Technology

The author can be contacted by email: emilwa@kth.se

Typeset using LaTeX

This is a non-revised version of the report

Abstract

Computational models of neural activity and neural networks have been an active area of research as long as there have been computers, and have led several important discoveries in the field of machine learning. One kind of artificial network proposed by John J. Hopfield in 1982 has been among the more successful ones, and is still in active use today. It has been suggested that in addition to its merits in machine learning, it could also serve as a foundation of the explanation of human ability of recollection and association. However, Hopfield's original design used a very simplified model of neurons. By using so called integrate-and-fire models, higher realism can be achieved.

This report begins with a discussion of mechanistic and quantitative description of neurons, in particular the induction of action potentials, and furthermore why an integrate-and-fire model is a reasonable choice for a model of intermediate complexity. By explicitly describing individual spikes, a fundamental but often neglected characteristic of communication between neurons is captured. Integrate-and-fire models are included in the Neural Simulation Tool (NEST), and in this report such a neural model is applied to Hopfield networks. Both spike-rate coding and temporal coding are studied, as well as a simple model of synaptic Spike-Timing Dependent Plasticity (STDP) for online learning. The networks' robustness is evaluated with respect to changes in (a) global scaling of the synaptic weights, (b) delays in the synaptic connections, (c) level of noise and (d) strength of input stimuli. They are found to be somewhat sensitive, with (a) giving the most definite results, suggesting that the used description of Hopfield networks might not be an immediately plausible biological model. In particular, networks using temporal coding are found to be especially difficult to calibrate. This could reveal a potential weakness in relatively recent and apparently successful models.

Sammanfattning

Beräkningsmässig modellering av hjärnaktivitet och neuronnät har varit ett aktivt forskningsområde lika länge som det har funnits datorer och har lett till ett flertal viktiga upptäckter inom maskininlärning. En typ av artificiella neuronnät som föreslogs av John J. Hopfield 1982 är ett exempel på en sådan upptäckt, och används frekvent än idag. Det har föreslagits att denna modell, utöver sina förtjänster inom maskinlärning, också skulle kunna utgöra en grund för en förklaringsmodell av den mänskliga förmågan att minnas och göra associationer. Dock använde Hopfield en väldigt förenklad modell av neuroner i sin ursprungliga design, och genom att använda så kallade *Integrate-and-Fire*-modeller kan en högre grad av realism uppnås.

Denna rapport inleds med en redogörelse om mekanistiska och kvantitativa beskrivningar av neuroner, med speciellt fokus på hur aktionspotentialer induceras, och vidare varför *Integrate-and-Fire*-neuroner är ett rimligt val vid behov av en medelutförlig modell. Genom att beskriva varje aktionspotential separat återges en fundamental men ofta försummad aspekt av neural kommunikation. *Integrate-and-Fire*-neuroner finns implementerade i simuleringsverktyget NEST (*Neural Simulation Tool*) och appliceras i detta arbete på Hopfields modell. Både spikfrekvenskodning och tidsberoende kodning undersöks, samt en enkel modell av tidsberoende synaptisk plasticitet (STDP). Neuronnätens robusthet mot förändringar av (a) global skalning av synapsvikter (b) fördröjningar i synapser (c) brusnivå och (d) styrka på stimulansen undersöks. De befinns vara någorlunda känsliga, i synnerhet (a) ovan har tydlig inverkan på prestandan. Detta indikerar att den använda beskrivningen av Hopfieldnät inte är omedelbart rimlig som en trovärdig bilologisk modell. Nät som använder tidsberoende kodning visade sig vara speciellt svåra att kalibrera, vilket tyder på en eventuell svaghet i relativt nya och till synes framgångsrika modeller.

Contents

1 Introduction

Already in the first half of the 19th century, following the recent invention of the telegraph, analogies between the nervous system and electrical circuits were made by several scientists and journalists.[8, p. 126] In 1948, Turing proposed a concept called the *unorganised machine*[27], which he argued could be used as a starting point to model the human brain on a computer. In the years since, a vast number of computational models of the human brain and nervous system have been suggested, describing the brain in its entirety or, perhaps more commonly, with focus on brain regions or features of special interest ([20] poses an early example, and a more recent, popular scientific example is [11], but there are many more.). As always, however, a great difficulty in modeling is discerning the aspects of reality required to describe the phenomena of interest. Selecting these aspects for a system as complex as (a part of) the human nervous system performing cognitive tasks, is far from trivial. [4, p. 195]

In this report, the impact of one such aspect on a well known model called a *Hopfield network* is investigated, specifically whether explicit description of individual action potentials, or *spikes*, affects its functionality. Explicit modeling of spikes is in contrast to the most widely used models today, as well as Hopfield's orignial formulation[14], in which the spike trains[1] are reduced to a single scalar function denoting the time varying spike *rate* or *frequency*. In fact, Hopfield approximates this further by modeling neurons as either active/firing or inactive/not firing, and thus restricting this scalar function to taking the values zero or one.

Hopfield networks have been suggested to describe how the associative memory in the neocortex operates (and that one result of this is the requirement of REM-sleep). [3] If Hopfield networks are to prevail as a plausible biological model, it is crucial that they can be implemented using spiking neurons, as real neurons undoubtedly use spikes for communication beyond their utmost proximity.[4]

Furthermore, any computational model describing the brain must be robust and not overly sensitive to small changes in parameters. This is due to multiple causes, the foremost being the intrinsically dynamic environment in organic systems. Neurons sometimes fail to emit an action potential although properly stimulated[4, chap. 5] and could also die completely. This must not severely diminish the global computational ability of the system.

Another reason for the required robustness is that the system must be self organizing to a certain degree. An entire human genome contains a few solitary gigabytes of information[8, p 296] making it impossible to explicitly code, for instance, synaptic weights and synaptic delays for larger groups of individual synapses. Moreover, there is extensive empirical evidence of substantial plasticity and malleability in the brain. [22] If a brain region can gradually adapt to a new task, the intermediate states must continuously be operational, and thus cannot be allowed to fail due to a small, gradual changes in parameters. Both

[1] The set of spikes and the times they are generated.

Turing [27] and Hopfield [14] recognize that the brain cannot rely on the same kind of fine tuning present in electronics and computers. Hopfield [14] writes,

> Our understanding of such simple circuits in electronics allows us to plan larger and more complex circuits which are essential to large computers. Because evolution has no such plan, it becomes relevant to ask whether the ability of large collections of neurons to perform "computational" tasks may in part be a spontaneous collective consequence of having a large number of interacting simple neurons.
> /.../
> I will seek collective properties that are robust against change in the model details.

The purpose of this report is therefore to investigate *qualitatively* whether Hopfield's model can be implemented using simple spiking neural models, and how sensitive to parameters (noise, exact values of synaptic weights and delays etc) such an implementation is. This is done by implementing a spike rate coded Hopfield network in a simulation engine for spiking neurons. However, in recent years, there are emergent indicia that (some) neurons code information not only in their firing rate, but also in the *relative times* that action potentials arrive. This is referred to as *temporal coding* and is supported both theoretically [19, 1, 17] and empirically[9]. Hence, it is possible that the classic model using only the spike rate is neither biologically correct nor shows the full potential of Hopfield's idea. Therefore, such a spike-time-dependent model, similar to the one presented in [19], is also investigated, and its robustness against changes in parameters is evaluated.

2 Background

2.1 Neuron models

2.1.1 The Hodgkin-Huxley model

By performing careful experiments on giant nerves from a squid, Hodgkin and Huxley [13] successfully devised a mathematical description of the electric characteristics of the neuron. In 1963, they were awarded with The Nobel Prize in Physiology or Medicine for this achievement.[21] Since, the biochemical explanations [4, chap. 5] have been improved and refined, but the general equations and ideas remain and are well established. Here, Hodgkin and Huxley's model will be briefly covered as a justification of the more simplified model actually used. The purpose of this section is to illustrate the complexity of the model, and point out some key concepts that will be used later on. It is not intended to provide an exhaustive explanation, and it is not crucial to comprehend all details in order to follow the subsequent line of thought. For a more thorough explanation, see for example [4].

The general idea is to consider the membrane potential V that arises from different ionic concentrations inside and outside the cell. This potential causes

the lipid membrane to act as a capacitor with specific capacitance c. Special channels in the membrane allows ions to pass in and out of the cell, resulting in a net (possibly negative) outflow current per cell area, i. Thus, the membrane potential obeys

$$c\frac{\mathrm{d}V}{\mathrm{d}t} = -i \tag{1}$$

Three different contributions to i are considered — the flow of potassium ions (K^+), the flow of sodium ions (Na^+) and the total flow of other ions. Two different mechanisms drive these flows. The first is electric forces driving the charged ions to cancel the potential difference, and the second is diffusion striving to eliminate concentration gradients. The potential difference at which these two mechanisms cancel is called the equilibrium potential and depends on the ion. The equilibrium potentials are denoted E_K, E_Na and E_L for potassium, sodium and the rest (the "leakage" equilibrium potential), respectively. When the potential is not at equilibrium for potassium, there will be a net flow per area $g_\mathrm{K}(V - E_\mathrm{K})$ where g_K is the conductance per area due to the ion channels, and similarly for sodium and the other ions. However, the channels for potassium and sodium are gated with gates that are only open with probabilities n and m respectively. In order for the potassium channels to be open, four consecutive, independent gates have to be open, resulting in a probability n^4 for a channel to be open. In the sodium channels there are also four gates that need to be open, but the last of them have opposite voltage dependance and is open with probability h. The probability of the sodium channel being open is therefore m^3h. As the number of channels is large, these probabilities are approximately equal to the fraction of open channels. In this model, the voltage dependance of the other gates are ignored, and the fraction of open channels for the other ions is assumed constant and incorporated in g_L. Hence, the net current per area is given by

$$i = g_\mathrm{L}(V - E_\mathrm{L}) + g_\mathrm{K}n^4(V - E_\mathrm{K}) + g_\mathrm{Na}m^3h(V - E_\mathrm{K}) \tag{2}$$

The fractions n,m and h of open gates also vary with the potential. The rates at which closed gates open are denoted α_n, α_k and α_h respectively, and the rates at which open gates close are denoted β_n, β_k and β_h. Hence

$$\frac{\mathrm{d}n}{\mathrm{d}t} = \alpha_n(V)(1 - n) - \beta_n(V)n \tag{3}$$

And similarly for m and h. All functions α andβ depend non-linearly on V, and are modeled used exponential and sigmoid functions whose exact forms are not relevant for this discussion.

What is relevant on the other hand, is that these differential equations explain the *spiking* behavior of neurons. An *action potential* occurs when the potential reaches a certain threshold where m quickly jumps from almost 0 to almost 1. This leads to a large inflow of sodium (E_Na is much larger than this threshold), which creates a positive feedback loop — the larger the potential, the larger m, and the larger m, the larger potential. The resulting rapid increase

in potential (or *depolarization* as the potential is normally negative) continues until the potential is high enough to set $h \approx 0$ which terminates the inflow. However, the new, high potential makes $n \approx 1$, resulting in a large outflow of potassium (E_{K} is low) which quickly presses the potential back to the resting potential. This process result in a short, but very definite spike in the potential. This spike can propagate through axon of the neuron to the synapses, causing them to release neurotransmitters (see section 2.2).

As can be seen, the Hodgkin-Huxley model is a set of coupled, non-linear, ordinary differential equations. An even more precise model take the morphology of the neuron into account, and recognizes that there is considerable internal resistance resulting in non-homogenous potential and ionic concentrations. An approximation of this can be achieved by discretizing the neuron into multiple *compartments*. However, for this study, even the single compartment description is too complex to be effective and further approximations are required.

2.1.2 The model used

In a simplified neuron model, the current's explicit dependance on the concentration of Na^+ and K^+ is neglected, and their impacts are instead incorporated into the "leakage"-term. This is a reasonable approximation below the threshold at which an action potential occurs, but it disables the spiking behavior. Spikes are then reintroduced by simply including an artificial threshold — when the membrane potential V exceeds a specific value V_{th} the neuron sends out a spike and the potential is reset to V_{reset}. To account for the time required to reset the ionic concentrations and ready the neuron for another spike, a *refractory period* is introduced. During the refractory period, the potential is clamped to V_{reset} and any synaptic currents are ignored.

This model belongs to a class of models called *integrate-and-fire* models. Basically, all incoming currents (synaptic or external) are summed/integrated until the threshold is reached, whereupon the neuron "fires" an action potential. The term is sometimes extended to *leaky-integrate-and-fire* to emphasize the leak current which drives the potential towards the resting potential.

A comparison between the Hodgkin-Huxley model and the integrate-and-fire model is shown in figure 1. The shapes do differ, but they qualitatively share many traits, indicating that the integrate-and-fire indeed is a reasonable approximation. Several interesting features can also be observed. The sub-threshold membrane potential exponentially approaches an asymptotic equilibrium value, the exact shape of this can be obtained by solving equation 1 using a simplified version of equation 2, as shown in equation 4,

$$\begin{cases} c\dfrac{\mathrm{d}V}{\mathrm{d}t} = -i \\ i = g_{\mathrm{L}}(V - E_{\mathrm{L}}) - i_{\mathrm{external}} \end{cases} \tag{4}$$

Observe that the explicit dependence on the currents of sodium and potassium now have been incorporated into the leakage term. The exponential behavior is however rather promptly interrupted, when the threshold potential is reached

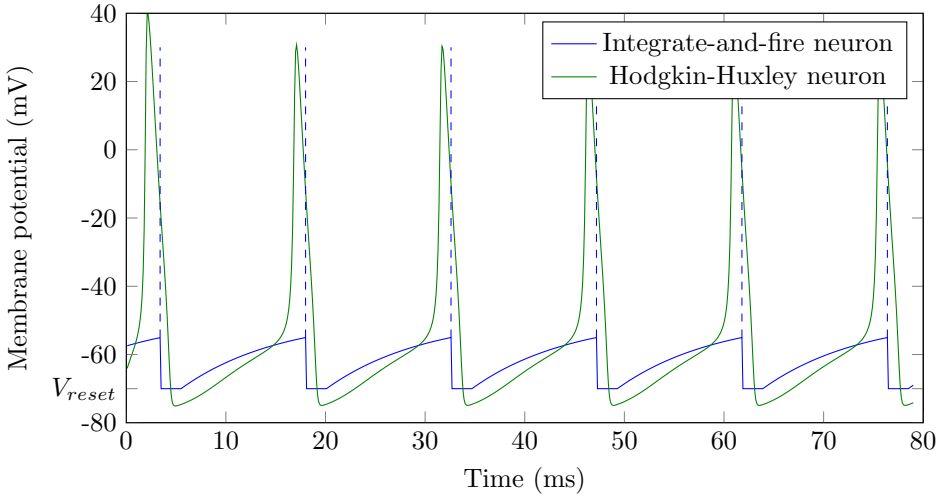

Figure 1: The membrane potential using the integrate-and-fire *model compared to the Hodgkin-Huxley model. The neurons are stimulated by a constant external current. The dashed lines indicate spikes from the integrate-and-fire model and are added post hoc.*

and an action potential is fired. As can be seen in figure 1, the membrane potential is then reset and the refractory period causes it to be clamped at the reset potential for a few milliseconds.

The integrate-and-fire model of figure 1 is the same as was used in these experiments. Here, the parameters (membrane capacitance, reset and threshold potentials, conductivities etc) are set to the defaults of the `iaf_psc_alpha` model in NEST (see section 3.2), which provide fairly realistic values. The two neurons are stimulated by constant currents set so that the spiking frequency would be approximately the same. Furthermore, the starting potential of the integrate-and-fire neuron was set so that the phase would also be approximately equal, making the shapes easier to compare. The dashed lines illustrate that a rise causing an action potential is present but not explicitly modeled.

2.2 Synapses

Heretofore, only single neurons have been considered. However, the focus of this study is not mainly on the rich dynamics of single neurons, but on their computational ability when connected. From a computational point of view, the interesting part of the connection between two neurons is the *synapse*. Synapses are junctions (usually) formed between the axon of one neuron, called the *presynaptic* neuron, and a dendrite of another neuron, called the *postsynaptic* neuron.

When the presynaptic neuron fires an action potential, it will propagate

through the axon and cause a group of chemicals called *neurotransmitters* to be released at the synapses. These neurotransmitters bind to receptors on the postsynaptic neuron, causing gated ion channels to open and enable a current. If the current flow inwards and the membrane potential is thus increased, the synapse is called *excitatory*. If the membrane potential on the other hand is reduced, the synapse is called *inhibitory*.

The morphology and biochemistry of synapses are well studied, and the above description is of course very simplified. However, further details on those aspects are beyond the scope of this report. Instead, the rest of this section will be focused on two features — the shape of the synaptic current and synaptic plasticity.

2.2.1 Synaptic current

As mentioned above, the release of neurotransmitters enables a current through the cell membrane. What proves to be an interesting problem is how to describe the time dependance of this current. A common way to do this is to model the current as a Dirac-pulse yielding a discontinuous jump in the potential of the postsynaptic neuron, and this is for most applications sufficient. However, a slightly more descriptive model could potentially yield interesting results, and is also crucial for the temporal coded networks explored by Maass and Natschläger [19].

A common model for the synaptic current is the *alpha*-function [2], i.e.

$$i \propto te^{-t/\tau} \tag{5}$$

However, it is the potential and not the current that is the primary state variable of the neuron and, as can be seen in equation 4, the synaptic current contributes to the derivative of the potential. Rotter and Diesmann [24] has suggested a way of integrating the current, and thus the effect on the potential, efficiently. It is called *Exact integration* and yields more accurate results. Diesmann's implementation of Exact integration of alpha-shaped synaptic currents is included in the simulation software, and was used throughout the simulations.

Three examples of post-synaptic potential excitation are shown in figure 2. At times t_1, t_2 and t_3 spikes arrive at *different* synapses. The synapses have relative weights of 1, 5 and 12. The spikes arriving at the first two synapses are not on their own able to induce a new action potential, but the third spike is. Note the refractory period and that when it is over the synaptic current has still not worn off but keeps increasing the potential for another few milliseconds. The synaptic weights in figure 2 are unrealistically large for illustrative purposes. In the subsequent simulations the synaptic weights are set so that several consecutive incoming spikes are typically required to induce an action potential in the post-synaptic neuron.

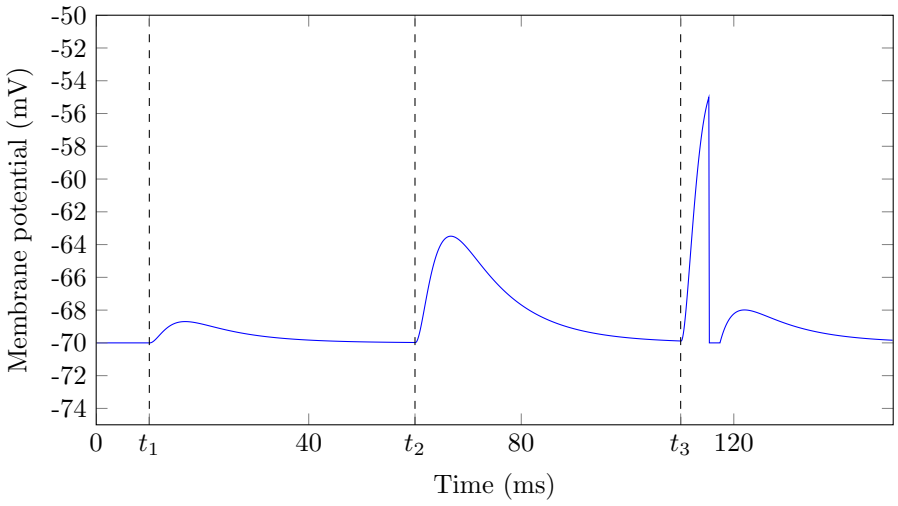

Figure 2: Illustration of the synaptic current's effect on membrane potential of an integrate-and-fire neuron, calculated using Diesmann's implementation of alpha-shaped synaptic current.

2.2.2 Plasticity

The efficiency of synapses vary from synapse to synapse. When studying networks, the efficiency is often referred to as the *strength* or the *weight* of one connection between two neurons. The weight of all synapses is believed to encode the "software" of the mind, and the set of weights is sometimes called the *connectome* of the individual[26].

In order to be able to form new memories and acquire new skills, the synaptic weights must be able to change — they must be plastic. The predominate theory for this is called *hebbian* learning, after Hebb's original formulation [12]. The idea is that synapses participating in the induction of a post-synaptic action potential get their weights increased. Spikes that arrive just before a post-synaptic action potential thus causes the increment of synaptic weights. Correspondingly, spikes that arrives just after a post-synaptic action potential cause the synaptic strength to be reduced. This has also been empirically observed[4, p. 282], and because of the very explicit time dependance, the phenomenon is called *Spike-Time-Dependent-Plasticity* (STDP).

2.3 Hopfield networks

A number of different connection schemes giving neural networks computational abilities have been suggested. However, this study focuses solely on *Hopfield networks*. It is a simple and elegant model which could describe how our associative memories operates [3]. Moreover, it is *Turing complete* [16], and other computational uses could also be present in the brain.

The synaptic connections between the neurons of a Hopfield network constitute a symmetrically weighted complete graph, i.e. every neuron is connected to every other neuron. Nevertheless, synaptic weights of zero are allowed, which in the non-plastic case can be considered equivalent to the absence of a connection. The symmetry of the weights is required in order to guarantee convergence.

In a Hopfield network of binary (ON/OFF) neurons, the states of the neurons are updated sequentially in random order. The update rule is very simple — if the total input to the neuron exceeds a threshold (usually zero) the neuron is set to on, otherwise it is set to off. The input is defined as the sum of the weights of all incoming synapses that originate from neurons currently in the ON state, plus a bias term that depends on the input or stimuli. Note that weights very well can be negative or equivalently, that the synapses can be inhibitory. Hopfield have showed that if one defines an "energy" function as the negative sum of the weights of edges/synapses between all pairs of neurons both in the ON state, this update rule will never increase the energy. This energy thus constitutes a Lyapunov function, guaranteeing that the network converges to a stable, albeit not necessarily optimal, state. If one considers the state space of the network, and the update rule as a dynamic rule of trajectory, these stable states are *fix points*.

The Hopfield networks investigated in this report do not consist of binary neurons, however, but of spiking neurons which can fire at a continuum of spiking rates. Time-dependent coding also permits a continuous range of delays. Nevertheless, it is still possible to find a similar Lyapunov function, and if the input-to-firing-rate function is non-linear enough, the network will converge to the same fix points [15, 16].

2.3.1 Storing memories

Hopfield proposes that the fix points of the network are what constitutes memories and concepts in the human brain. The convergence towards these fix points is thus why partial or noisy stimuli is sufficient for successful recollection of an entire memory. For this to be useful, it is essential that new memories, i.e. fix points, easily can be written to the network. Hopfield describes how to set the weights such that a set of memories $\mathcal{M}$ will be fix points. The weight w_{ij} between neuron i and neuron j is set using the formula

$$w_{ij} = \frac{1}{|\mathcal{M}|} \sum_{\mathbf{m} \in \mathcal{M}} 4 \left(m_i - \frac{1}{2} \right) \left(m_j - \frac{1}{2} \right) \tag{6}$$

This is an equivalent formulation of equation 2 in [14]. $|\mathcal{M}|$ denotes the total number of memories to be written into the network. Each memory $\mathbf{m} \in \mathcal{M}$ is described by a binary vector of the same dimension as the number of neurons, which will be denoted N. Every memory vector thus represents a configuration of ON/OFF states — a one in the i'th position in the memory vector implies that neuron i is active in the corresponding memory. Equation 6 will increase the weights between two neurons if the neurons are either both active or both

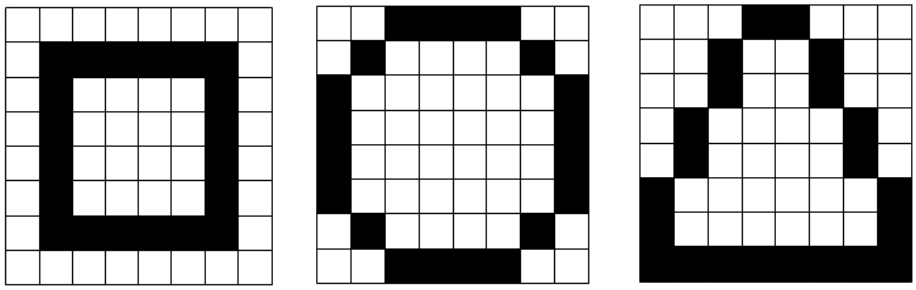

Figure 3: The memory vectors used. They are very coarse and low resolution caricatures of a square, a circle and triangle, respectively.

inactive in a given memory, and decrease it otherwise. A more concise mathematical formulation is that the weight matrix is proportional to the sum of the correlation matrices of the respective memories.

This rule will make the memories in $\mathcal{M}$ fix points of the network. However, there is a limit to how many memories can be stored. Hopfield empirically estimates this limit to about $0.15N$. Using improved learning rules, this limit can be increased [6]. However, the purpose of this report is not to achieve optimal storage capacity and the rule of equation 6 will be used due to its simplicity and slightly higher biological plausibility.

3 Method

3.1 Software

The simulation engine Neural Simulation Tool (NEST) [7] was used to to perform the experiments. NEST offers a selection of different models of neurons and synapses, making it suitable for the intended investigations and comparisons between different computational models of neural networks. Simulations in NEST were originally initialized and controlled by a language called *Simulation Language Interpreter* (SLI), but the slightly more recent interface by Eppler et. al. called *PyNEST*[5] was more convenient for the experiments presented in this report. Being implemented as a Python library, it enables more direct possibilities of data post processing, and seamless integration with Python scripts for automated generation of input and initial conditions.

3.2 Memory vectors

Networks consisting of 64 neurons were used and none of them were *hidden*[2]. Hence, binary vectors of length 64 were needed as predefined fix points (memories) of the network. These binary vector could be arbitrarily, or even randomly,

[2]Meaning each neuron directly corresponds to one element in the memory vector. One could imagine having additional "helper"-neurons which are not included in the memory.

Defined in	Description	Symbol	Value
Equation 4	Leakage equilibrium potential or resting potential	E_L	-70.0 mV
Equation 4	Membrane capacitance	c	250 pF
Equation 4	Membrane time constant	c/g_L	10.0 ms
Equation 4	External current	i_{external}	0
Section 2.1.2	Refractory time	t_{ref}	2.0 ms
Section 2.1.2	Spike threshold	V_{th}	-55.0 mV
Section 2.1.2	Reset potential	V_{reset}	-70.0 mV
Equation 5	Spike rise time	τ	2.0 ms
(Not used)	Absolute lower value for the membrane potential	V_{min}	$-\infty$

Table 1: The parameters for the `iaf_psc_alpha`-model. These values are the default in NEST and was used throughout the simulations.

chosen, but in order to achieve slightly more realistic conditions they were instead set to binary bitmaps describing the contours of simple geometrical shapes. Empirical research suggests that the mammal visual system extracts contours from images early in the visual process [23]. Hence, it is reasonable to model the input of an associative memory of shapes as the contours of such shapes. This naturally also results in invariance to color and brightness when identifying shapes. Therefore, contours of simple shapes will constitute a pertinent example of data where there are significant covariance between certain bits across memories. This results in slightly different statistical properties compared to random or arbitrary data. Consequently, the shapes shown in figure 3 were used throughout all the simulations.

3.3 Overview

Three different premises were investigated. In the first premise, the synaptic weights were fixed before the simulation, using the method presented in section 2.3.1. In the second premise, the method for setting the weights was implemented in the simulation itself, by using a STDP rule (see section 2.2.2). The weights generated using this rule is then compared to the weights that was used in the first premise. In both these premises, spike-rate coding was utilized. In the third premise, on the other hand, temporal coding as described by Maass and Natschläger [19], was used. These three premises are described more exhaustively below.

3.4 First premise: spike rate coding with fixed weights

In this premise, the relative synaptic weights are fixed using equation 6. The neurons are described by the `iaf_psc_alpha`-model (see section 2.1.2), with the default values as defined in NEST (presented in table 1).

The values presented in table 1 are plausible, but the low number of interconnected neurons is not. The resulting low amount of synapses is compensated by scaling the weight of each synapse, so that the neurons spike with reasonable rates. Some kind of scaling is inevitable since equation 6 only describes the *relative* weights on a scale ranging from 0 to 1 and provides neither an external scale nor a unit for these weights. Hence, these relative weights need to be transformed to absolute weights. By setting the maximum synaptic weight to as high as 225 times NEST's default value, a decent performance is achieved. No further justification for this exact value will be provided. Instead, this "global scaling" of weights is the first parameter to be investigated.

In the simulations of the premise, the input from earlier stages in the visual process is thought to consist of spikes arriving at random, Poisson-distributed times[3]. The Poisson process is set so that each neuron receives on average 1000 such "input"-spikes per second. Given a binary input/bias vector, these spikes are then modeled to arrive at an excitatory synapse of the neuron if the corresponding in the vector value is one, and an inhibitory synapse of it is zero. All of these "input"-synapses have a weight of 60 relative to NEST's default, i.e. approximately 27% of the value of the maximum synaptic weight between neurons. Note that the presence of inhibitory synapses from the input means each value in the input/bias vector is actively known — a zero indicates that the neuron is biased towards being off/silent, not that it is unbiased or "unknown". The input/bias vectors are chosen randomly from the set of vectors described in section 3.2.

The pattern-completion ability of the network is evaluated while four global parameters are varied consecutively. The first parameter is the scale of the synaptic weights of neuron-to-neuron synapses, as described above. The second is the synaptic delay and the third is the synaptic strength of the input/bias connections. The forth is the presence and strength of Gaussian noise. For each set of parameters, 1000 tests are run.

In each test, a target memory vector is randomly selected from the set of shapes. Then, four random bits are flipped, and this is set to the input. This is a rather small deviation — Hopfield networks normally should be able to correct larger errors than this — but it gives measurable performance even when the choice of parameters is rather bad. 100 milliseconds of neural activity is subsequently simulated with a resolution of 100 microseconds. This is enough time for the network to settle in a stable state. At the end of this simulation, a retranslation of neural activity to a binary state vector is required for performance evaluation. This is achieved by considering a neuron that spiked in the last 10 milliseconds of the simulation to be ON, or OFF otherwise. This is obviously a very coarse classification of the spike rate. However, due to the large number of simulations, it can be treated statistically. Consider a neuron that spikes at a very low rate — it will only be counted as ON in a few of the tests. Thus, it will, correctly, affect the statistics slightly differently than a neuron that is completely silent. Anyhow, using this rule, a result memory vector is created

[3]This is a common approach, and is often used for example in [4]

and compared to the target memory vector.

3.5 Second premise: synaptic plasticity

A relevant question is whether the synaptic weights described by equation 6 and used in the first premise, can emerge from a simple STDP learning rule[4]. In this premise, a standard STDP rule is introduced to the network. Initially, all synaptic weights are set to zero. Then, a very strong input bias consecutively force the network into states corresponding to the memories. This way, each memory is active for one full second of simulation time. The synaptic weights were restricted to weights below 225, to ease comparisons with the previous premise.

In order to restrict the weights, and to achieve a compromise between a multiplicative and additive increments, Gütig et al. [10] have suggested that the weights are updated according to equation 7,

$$\Delta w = \begin{cases} -\lambda \alpha w^{\mu} e^{\frac{\Delta t}{\tau}} & \text{if } \Delta t \leq 0 \\ \lambda (1-w)^{\mu} e^{-\frac{\Delta t}{\tau}} & \text{if } \Delta t > 0 \end{cases} \tag{7}$$

Here, w is the current weight (the scale of Gütig et al. ranges from 0 to 1) and Δw is the change induced by an incoming spike. Δt is the time difference between the arrival of the incoming spike and the post-synaptic action potential.

This rule is implemented in the NEST model `stdp_synapse` which was used for the simulations in this premise. Probationary investigations indicated that an additive variant of the rule ($\mu = 0$) yields the most reasonable distribution of weights. However, to maintain the restrictive property, μ was set to the small non-zero value 0.01.

The remaining parameters were set to the defaults of NEST. Hence, τ was 20 ms and λ was 0.01. No asymmetry between potentiation (increment of the weight) and depression (decrement) was thought to be computationally justified, so α was also kept at the default of 1.

The weights generated during these simulations were compared to the theoretical weights described by Hopfield's model. This was done by calculating the Pearson correlation coefficient between the learned weights and the theoretical weights. The distribution of learned weights with respect to theoretical weights was also investigated.

3.6 Third premise: temporal coding

Finally, temporal coding, as described by Brody and Hopfield [1] and by Maass and Natschläger [19], was explored, and several attempts were made at combining the slightly different approaches in the respective articles.

One of the differences was the choice of driving source. Brody and Hopfield use an external oscillation in the external current (i.e. letting $i_{external} =$

[4] See section 2.3.1.

Defined in	Description	Symbol	Value
Equation 4	Leakage equilibrium potential or resting potential	E_L	10.0 mV
Equation 4	Membrane capacitance	c	200 pF
Equation 4	Membrane time constant	c/g_L	20.0 ms
Section 2.1.2	Refractory time	t_{ref}	2.0 ms
Section 2.1.2	Spike threshold	V_{th}	20.0 mV
Section 2.1.2	Reset potential	V_{reset}	0.0 mV
Equation 5	Spike rise time	τ	2.0 ms
(Not used)	Absolute lower value for the membrane potential	V_{min}	$-\infty$
This section	Oscillation current frequency	f	35 Hz
This section	Oscillation current amplitude	A	50 pA

Table 2: The parameters of [1] was used in the third premise.

$A\sin(2\pi ft)$) to create synchrony while Maass and Natschläger introduce a special "driving"-neuron. Both approaches were attempted, but the main focus was on using external oscillation. The two articles use similar frequencies for the oscillations, 35 Hz and 40 Hz respectively. In this report, the value 35 Hz was used. Furthermore, Brody and Hopfield also introduce an additional bias current that differ from neuron to neuron. Here, such a bias current is used as input to the network.

Two global parameters were given special interest — the synaptic delay and the global scaling of the synaptic weights, similar to the first two parameters investigated in the first premise. However, since no particularly good set of parameters was found on which to base the comparisons, no similar quantitative description could me made. Instead, the effect of changes in the parameters are described qualitatively in section 4.3.

The primary set of neuron parameters used was that of Brody and Hopfield, as Maass and Natschläger used a multi-compartment Hodgkin-Huxley neuron model which does not enable an unambiguous translation to integrate-and-fire parameters. The parameter set differ slightly from NESTs default and is presented in table 2. Note especially that E_L does no longer equal V_{reset} and that another convention for the zero potential is used.

A benefit of using exactly the same set of parameters is that the choice of drive and bias current can be based on Brody and Hopfield's study as well. From figure 1 in [1], one finds that the best synchronization given these parameters appears to arise when the bias current is between 160 pA and 180 pA. Thus, the bias current for neurons that were OFF in the input memory vector was set to 165 pA and neurons that were ON in the input memory vector were correspondingly set to having a bias current of 175 pA. The implementation of the bias current was loosely inspired by Sven Schrader's example code which is shipped with NEST.

The number of neurons simulated was 64, as in the previous premises. Ad-

ditionally, the same theoretical weights (i.e. memory vectors) were used, and the number of erroneous bits in the starting vector was likewise 4.

In order to determine whether the neuron spike early, a reference neuron was used. This reference was not connected to the rest of the population, and was stimulated by a bias current of 170 pA. Neurons spiking before this reference were considered "early" or ON. Conversely, neurons spiking after the reference neuron were considered "late" or OFF.

4 Results

4.1 First premise: spike rate coding with fixed weights

Simulations of 100 milliseconds of neural activity was carried out 1000 times for each set of parameters. Each test case was evaluated as follows. First, the spiking activity was translated into binary vectors. A neuron was considered ON if it spiked in the last 10 milliseconds of simulation, and OFF otherwise. This vector was then compared to the target vector, and the number of errors was counted. This is called the *Hamming distance*. The Hamming distance from the target vector to the input vector is always fixed at 4. Hence, if the distance between the result and the target vector is larger than 4, it is more wrong than the input vector. This is interpreted as convergence towards an incorrect minimum of the Lyapunov function, and the exact Hamming distance is no longer of relevance. Thus, in figures 4 to 7, the results are divided into two categories based on whether the Hamming distance was smaller or larger than 4. The first is convergence towards the correct minimum, and for this the measure of interest is the average Hamming distance, as this indicates how well the pattern was completed. The average here is only taken from the first category, i.e. convergence towards incorrect minima does not worsen the average. The second category is convergence towards an incorrect minimum, and the indicated value is the *fraction* of such erroneous completions.

As can be seen in figure 4, there is a dichotomy between a more successful completion with less errors, and the risk of erroneous completion. This is expected, as a stronger synaptic weight reduces the relative influence of the input, implying that erroneous bits in the input are easier to suppress. Correspondingly, the reduced influence of the input decreases the Lyapunov "energy"-value of the incorrect states, making them more likely. In either way, any larger deviation from the "good" synaptic strength reduces performance significantly, by either lessen the pattern-completion ability or by increasing the risk of erroneous completion. However, the network does still have decent performance when the deviation is small to moderate, indicating that although a good value of the global synaptic scaling is required, extreme fine tuning is not.

Figure 5 does not reveal any clear trend when the synaptic delays are varied slightly. This would imply that as long as the delay is just a few milliseconds, the exact value does not matter.

Noise, on the other hand, does seem to have some interesting effects. When

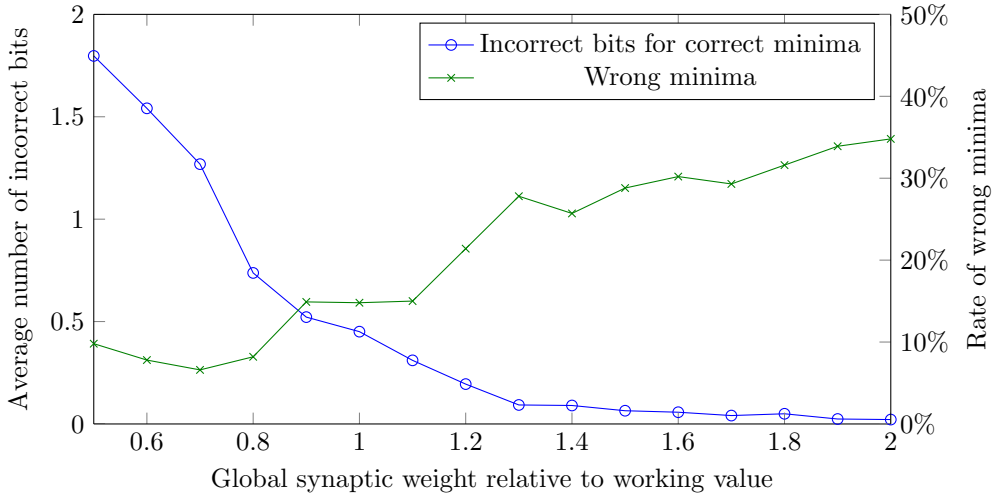

Figure 4: Sensitivity to changes in the global scale of the synaptic weights. The scale factor is measured relatively to a value (225 times the NEST default) that was found to generate decent performance.

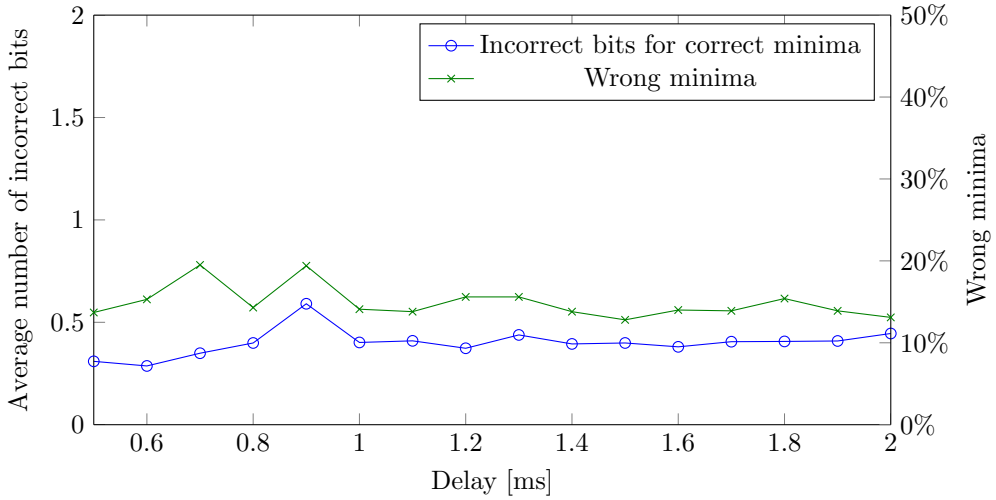

Figure 5: Sensitivity to synaptic delays. A delay of 1 ms was default in the other simulations, thus the scale can be read as either relative to that value, or as absolute value in ms.

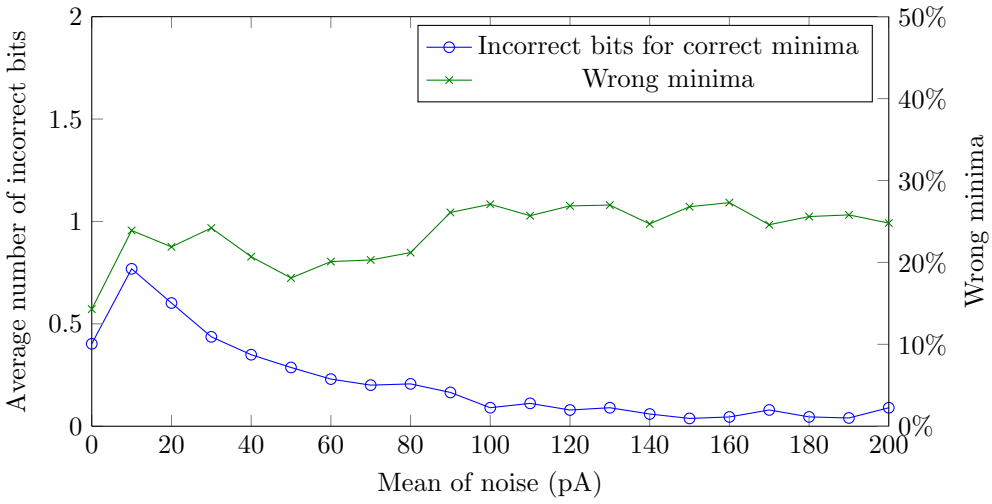

Figure 6: Sensitivity to noise. When noise is first introduced, the performance drops slightly. However, the average number of incorrect bits almost completely vanishes when noise is increased.

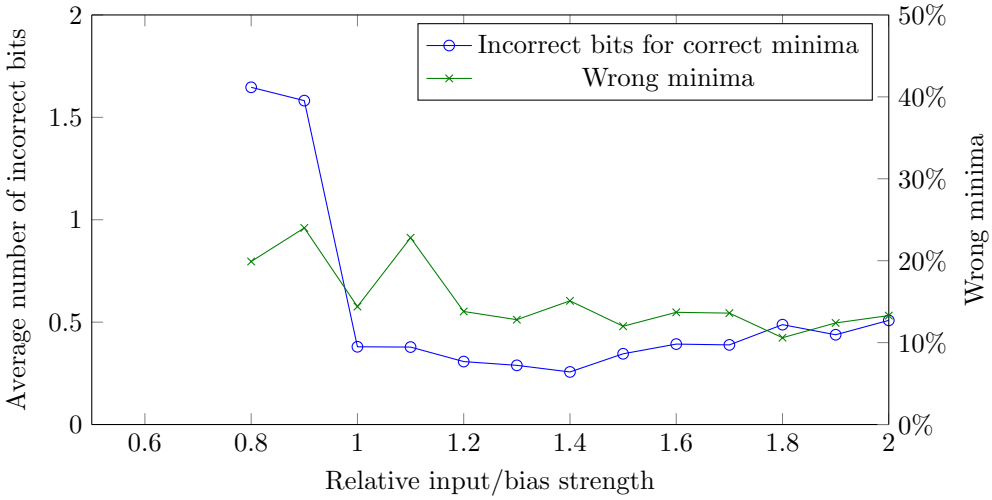

Figure 7: Input strength. When the relative input strength is lower than 0.8, no neuron reaches the spike threshold potential, and no spiking occurs. Once the input strength is strong enough, even stronger input does not alter performance significantly.

a small noise is first introduced, both kind of errors increase. However, as larger noise is introduced, the noise seem to help the network settling to a better energy minimum. Nevertheless, even the presence of a small noise seem to inevitable increase the risk of converging to the wrong minimum, and this risk appears rather constant even as the noise is increased. One possible explanation for this is that noise let the network escape poor minima and find better ones. However, when the current minima is the deepest in the reachable proximity and the noise "excites" the network to a state with higher energy, it will eventually fall back. Thus, if the target state is not a very deep minimum, there will be a risk that the network converges towards another minimum. Anyhow, the following statement by Hopfield [15] could in general be said to be verified,

> Because the system operates by moving downhill on an energy surface, the injection of a small amount of quantal noise will not greatly change the minimum-seeking behavior.

Surprisingly, the weight of the input connections does not seem to be of big importance. Once the weight is high enough to start exciting the network, the exact value does not seem to have any significant effect. An exception is very low values barely managing to excite the network, which result rather poor performance for pattern-completion. Anyhow, this could potentially dispute the explanation given above regarding the prominent effect of neuron-to-neuron synaptic scaling, according to which stronger input weights ought to increase the average number of incorrect bits, which apparently is not the case in figure 7.

4.2 Second premise: synaptic plasticity

Unfortunately, the current model of synaptic plasticity did not generate synaptic weights as described by the model. The mismatch between the *theoretical* and the *learned* weights is illustrated in figure 8. Since only three memories were used, there were only four possible theoretical weights ($-1, -1/3, 1/3$ and 1) that could come from equation 6. In figure 8, the distribution of learned weights is shown per class of synapses with same theoretical weights.

Even though it could be unsound to expect a linear relationship between the theoretical weights and the learned ones (in premise one, they were fixed to 225 times the theoretical value and thus linear), one would expect to find that the learned synaptic weights give rise to four distinct peaks, corresponding to the four theoretical weights. However, this is clearly not the case. In fact, the distribution of learned weights for synapses with theoretical weights of $1/3$ is of almost identical shape as the learned weights for synapses with theoretical weights of $-1/3$. The difference in amplitude is due to the fact there were 523 non-zero synapses with theoretical weight $1/3$ and only 156 with theoretical weight $-1/3$.

On the other hand, almost all synapses with theoretical weight -1 did get the learned weight of zero, which is the lowest possible. Furthermore, almost all learned weights that is in the two bins corresponding to the largest weights belonged to synapses with theoretical weight 1. However, even synapses with

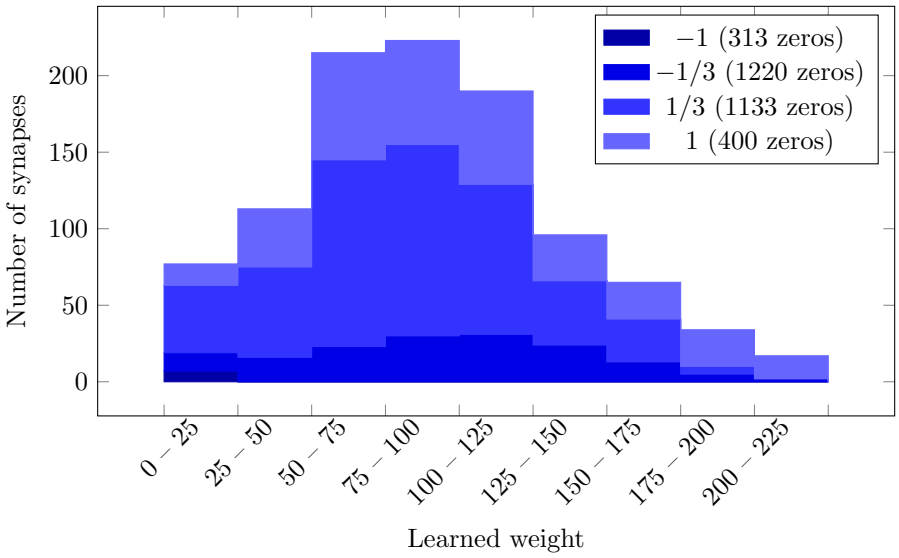

Figure 8: Histogram describing the distribution of learned weights. There are only four different expected weights (unscaled). Zeros are treated separately and are not included in the plot.

theoretical weight 1 have learned weights that are fairly evenly distributed across all bins.

Note that in figure 8, the learned weights of synaptic weight zero are not included. They posed such a big part of the learned weights that they would reduce the clarity of the plot. Instead, the numbers of synapses with learned weight zero are indicated in the legend.

The Pearson correlation coefficient between the array of expected weights and the array of learned weights was 31%. Given that many weights were zero, and definitely neither uniformly nor Gaussian distributed a priori, that must be considered a really low value.

Some other choices of μ (see equation 7) were also examined, but did not result in improved correlations (neither by inspection nor by calculation of Pearson correlation coefficient). Similarly, longer simulation time did increase the average learned synaptic weight, but did not lead to better correlation.

4.3 Third premise: temporal coding

Finding a good set of parameters resulting in acceptable performance proved to be quite difficult. Especially, retaining synchronization when each neuron receives different input seems to require some extraordinary fine tuning. Maass and Natschläger [19] indicate that when using integrate-and-fire neurons, the exact synaptic delay need to be exact at millisecond precision for the prerequisite of their theoretical result to be fulfilled (this is however not necessarily required

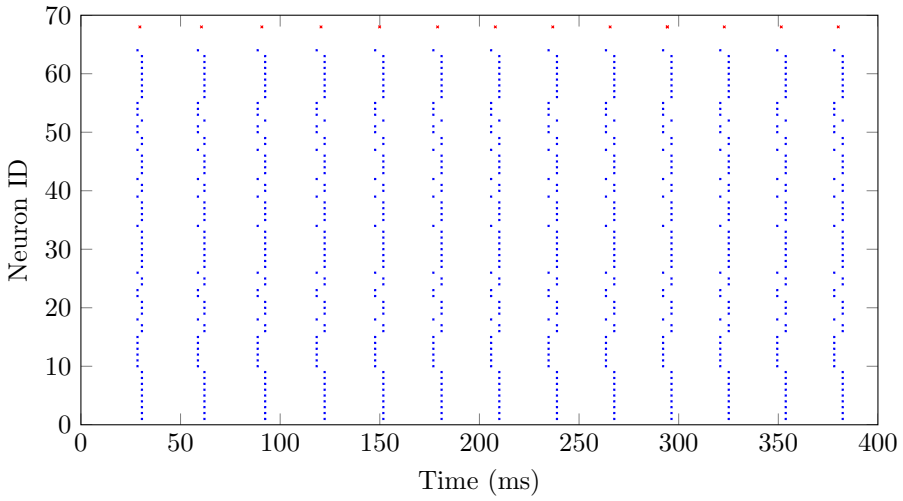

Figure 9: Temporal coding without any weights. The bias currents are set to 160 pA and 180 pA respectively. Each mark indicate that the neuron with the number on the y-axis emitted a spike at the time indicated on the x-axis. Note the stable and recurring pattern — it is the same neurons that fire early in each wave. The spikes of the reference neuron with bias current 170 pA are marked red.

for a more advanced model). Nevertheless, no set of parameters yielding stable performance was found in this study even though the global delay was varied with millisecond, and in some cases even sub-millisecond, resolution.

Indeed, even achieving a completely stable synchronization using only external stimulation, omitting connections between neurons and any noise, is non-trivial. In figure 9, such a synchronization is shown. The different "waves" of activity are clearly separated. The pattern of each wave represents a memory vector where four random elements have been inverted. As no connections between neurons were present, i.e. the global scaling of synapses was zero, no convergence occurs. Conversely, the absence of such connections combined with the absence of noise keep the pattern static.

Figures 10 to 12 illustrate what happens when neuron-to-neuron connections are introduced. At two times the default synaptic weight of NEST (a very low value compared to previous premises), the difference from figure 9 is clearly visible. However, the effect is not strong enough to correct any of the four erroneous bits, if a correction is considered to be made when the neuron fires on the correct side of the reference neuron. Nonetheless, if the synaptic weight is increased by 25%, 23 out of 64 neurons fire with incorrect timing with respect to the reference neuron. Furthermore, some neurons are so far ahead that one might start to question whether they are still synchronized with the rest.

If the weight is increased by another 100% as shown in figure 12, the syn-

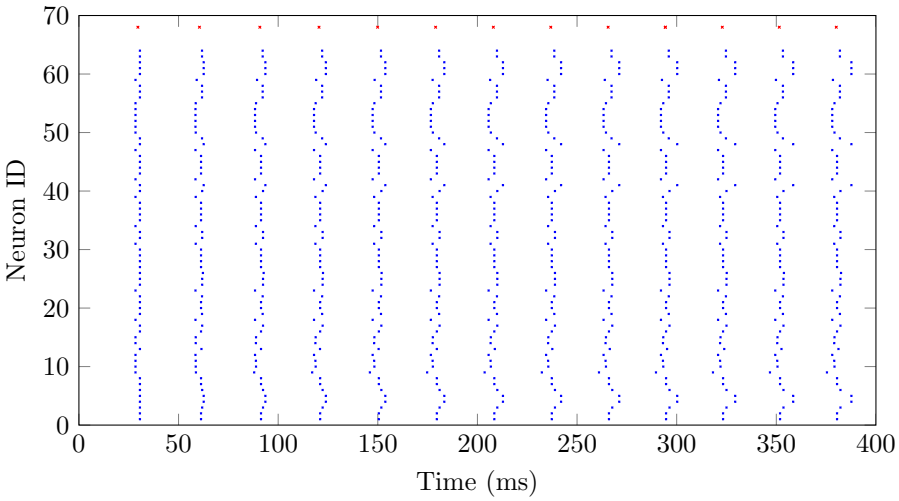

Figure 10: Synaptic weight 2 times NEST default (i.e. less than 1% of weights used in the previous premises). The pattern is still stable, but the four incorrect bits remain uncorrected.

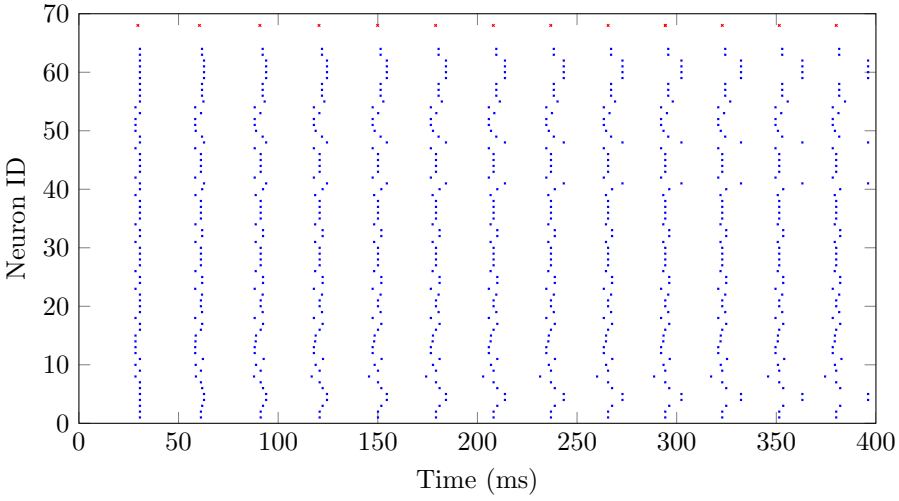

Figure 11: Synaptic weight 2.5 times NEST default (i.e. 25% higher than in figure 10). Now the synchronization is on the verge of collapse, as some neurons spike so far ahead that they are in between waves.

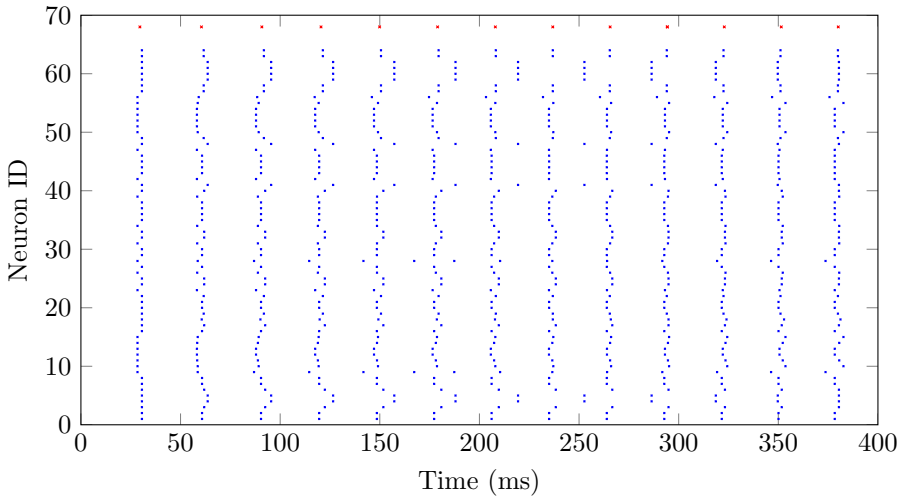

Figure 12: Synaptic weight 5 times NEST default (i.e. 150% higher than in figure 10). The synchronization is broken. All neurons no longer spike the same number of times.

chronization collapses — all neurons no longer fire the same number of times during the course of the simulation, and for some spikes it is ambiguous which wave they belong to. The number of erroneous bits are still much worse than the starting value of 4.

Under the assumption that the rest of the parameter choices are correct, in particular, that the synaptic delay is correct, one would thus expect to find the working value between 2 and 2.5, if anywhere. However, in spite of careful testing, no such value could be found.

The remaining contingency is thus foremost the synaptic delay time. In the above investigations, the delay time was set to exactly match the wave frequency. However, one could argue that a slightly shorter delay time could lead to better performance by reducing the drift of the very actively spiking neurons. Furthermore, the neurons ought to receive their stimuli *before* spiking, not spike as a direct consequence of the stimuli. Hence, it would be reasonable to let the spikes arrive a little early compared to the peak if the drive current. Nevertheless, no working pair of synaptic strength and delay was found even though the examined range of the delay was as large as $\pm 20\%$ of the wave period.

A few other configurations, including other ways of classify the time difference and other ways of creating the drive current, were also tested without success. However, due to the limited scope of this report, no quantitative analysis will be provided.

5 Discussion

As two out of three premises gave negative results, no definitive conclusion can be drawn from them alone. On one hand, a basic but operational Hopfield network appears to be reasonably easy to achieve with spiking neurons, but on the other, the performance is rather poor compared to non-spiking implementations. Furthermore, STDP-learning does not seem suitable for this kind of spike-rate coded Hopfield network.

The obvious question on the occasion of negative results is whether positive results could have been achieved using a different set of strategies, parameters or features. Naturally, this is a very difficult question to answer definitively. Nevertheless, the mere fact that there is some difficulty in implementing an operational Hopfield network using spiking neurons, in spite of having access to model parameters from empirical experiments as well as previous functioning simulations of similar algorithms, suggests that they possess an intrinsic transience.

Nonetheless, one way of expanding the searched volume of the parameter space would have been to use a more thoroughly planned automation scheme. The current approach is from an optimization point of view rather naive. In particular, the use of a genetic algorithm could have been justified, partially from a computational perspective, but also from a biological perspective. Evolution has found a set parameters resulting in working networks with extraordinary computational abilities. Thus, if those networks resemble Hopfield networks, a working set of parameters might very well be obtainable using similar methods.

One possible objection to this study is the very low number of neurons used. The human primary visual cortex (V1 area) contains approximately between one and two hundred million neurons [18], and 64 is undeniably a very small fraction of this. The small number of neurons was chosen for increased visual clarity and general manageability. A greatly increased amount of neurons would furthermore have led to longer simulation times, obstructing the search for a working set of parameters. Additionally, the theory does not predict any explicit dependence on the neural population size, merely on its relation to the number of memories stored. As only three memories were used, the unrealistic low number of neurons can thus be considered compensated by an equally unrealistic low number of memories. Although potentially interesting, it is beyond the scope of this report to theoretically examine whether a greater number of neurons would be crucial for the algorithm's performance when using spiking neurons.

Another objection is that although the integrate-and-fire model is significantly more realistic than the scalar or two-state model, there is still a significant gap to reality. Maass and Natschläger [19] found that their multi-compartment Hodgkin-Huxley model resulted in better performance than using corresponding integrate-and-fire neurons. Similarly, one could speculate that essential features could have been neglected in the model presented here. In particular, the fact that all outgoing synapses of a neuron are either all excitatory or all inhibitory [4] could significantly alter the learning examined in the second premise. The used learning rule promoted connections between simultaneously active neurons, but

Hopfield's rule also promotes connections between simultaneously *inactive* neurons, as well as increases the *inhibitory* connections between neurons that are not simultaneously in the same state. Taking this into consideration, the poor correlation between the theoretical weights and the learned weights is not very surprising.

A possible combined solution to these two problems could perhaps be as follows. If two excitatory neurons were connected to a single inhibitory neuron, and the inhibitory neuron was in turn connected to the two excitatory ones, the unit of three neurons would have three stable states — one excitatory neuron spiking, the other excitatory neuron spiking, or none of the neurons spiking. The inhibitory neuron would cause the others to fire mutually exclusive of each other. Thus, they could represent the two state system in Hopfield's original formulation. Then, by letting one of the two excitatory neuron represent 1 and the other represent 0, an equivalent network using only positive synaptic weights between units could be formed. Using a similar learning rule as in the second premise, all three kinds of relations between neurons (ON–ON, OFF–OFF, ON–OFF) would be correctly captured. A small pilot study of this model was carried out in connection to this report, but was unfortunately not successful. The pilot study used spike-rate coding, but there is a possibility that temporal coding could be more efficient — neurons fire in synchrony, but are connected in such a way that each neuron fires mutually exclusively to a particular other neuron. Another advantage of this implementation is that neurons also can be in an "indeterminate" state, where neither of the neurons in a pair fire.

Regardless of whether the above model has any merit, the presence of STDP in multiple brain regions is rather well established [25]. Thus, although another learning rule might give acceptable correlations between learned and theoretical weights, rendering the above model unnecessary, the fact that STDP learning does not yield satisfactory results is somewhat discrediting to the explicit[5], spike-rate coded version of Hopfield networks. In general, the pronounced time dependance of the learning rule would appear to better match some variant of temporal coding.

One might furthermore consider the contribution of using memory vectors based on simple geometry, compared to random data. Although the used memory vectors would imply that a larger part of the synaptic weights were at the two extremes of the range, their small number most likely prevented any significant statistical effect. Additionally, the real visual pathways of humans (called the dorsal and ventral streams) include many more processing steps than simply extracting the contours[4]. The use of contours as opposed to any other method of choosing memory vectors should therefore not be given too much concern.

However, a more fundamental question is whether visual stimuli are spatially coded at all in the recognition phase of human visual processing. For example, the current coding does not provide any a priori translational or rotational invariance. Since timing appears to be such an intrinsic and important aspect of

[5]I.e., a model where a neuron in the Hopfield network directly corresponds to a biological neuron

neural circuits [9], one could speculate that recoding spatial data into a temporal sequence could be reasonable. For example, if one sweeps along a contour and defines an input signal to the network as the derivative with respect to the tangential[6] direction, a contour could be treated as a temporal signal, being invariant to both translation and rotation of the shape. Nonetheless, for this discussion, having recognized completely different coding principles could exist is considered enough, and further details are not included here.

In conclusion, although the biological possibility of Hopfield networks cannot be ruled out — one operational implementation was demonstrated — it has been somewhat diminished. Care should be taken when analyzing theoretical models, as there clearly is some discrepancy in the ease of implementation between very simplified models and models using spiking neurons. However, with that in mind, there is still much need for further work — both empirically and theoretically.

Acknowledgments

First and foremost, I would like to thank my supervisor Örjan Ekeberg for supporting and advising me, while also giving me the freedom to explore my own ideas. Thanks to him, I have had this wonderful opportunity to spend a large part of the semester learning about a big interest of mine.

Furthermore, two MOOCs[7] on Coursera have been most helpful by excellently describing the background information I needed. They are *Computational Neuroscience* by Rajesh P. N. Rao and Adrienne Fairhall, and *Neural Networks for Machine Learning* by Geoffrey Hinton.

Throughout this project, I have been using the Neural Simulation Tool (NEST)[8], which is generously freely available. Similarly, the availability of free, high-quality software such as Python (including the libraries NumPy, SciPy and Matplotlib), Cygwin (with all the included packages) and LaTeX (especially including the TikZ and Pgfplots libraries) has been most valuable.

Finally, any errors, mistakes or shortcomings of this report are entirely my own fault and I claim full responsibility for them.

References

[1] Carlos D. Brody and John J. Hopfield. Simple networks for spike-timing-based computation, with application to olfactory processing. *Neuron*, 37:843–852, 2003.

[2] C.Bernard, Y. C. Ge, E. Stockley, J.B. Willis, and H.V. Wheal. Synaptic integration of nmda and non-nmda receptors in large neuronal network

[6] Or, correspondingly, the normal
[7] Massive Open Online Course
[8] See References

models solved by means of differential equations. *Biological Cybernetics*, 70(3):267–273, 1994.

[3] Francis Crick and Graeme Mitchison. The function of dream sleep. *Nature*, 304:111–114, 1983.

[4] Peter Dayan and L.F. Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT Press, 2001.

[5] Jochen Martin Eppler, Moritz Helias, Eilif Muller, Markus Diesmann, and Marc-Oliver Gewaltig. PyNEST: A convenient interface to the nest simulator. *Frontiers in Neuroinformatics*, 2(12), 2009.

[6] Elizabeth Gardner. Maximum storage capacity in neural networks. *EPL (Europhysics Letters)*, 4(4):481, 1987.

[7] Marc-Oliver Gewaltig and Markus Diesmann. Nest (neural simulation tool). *Scholarpedia*, 2(4):1430, 2007.

[8] James Gleick. *The information: a history, a theory, a flood*. Pantheon Books, 2011.

[9] Anubhuti Goel and Dean V. Buonomano. Timing as an intrinsic property of neural networks: evidence from in vivo and in vitro experiments. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 369(1637), 2014.

[10] R. Gütig, R. Aharonov, S. Rotter, and Haim Sompolinsky. Learning input correlations through nonlinear temporally asymmetric hebbian plasticity. *The Journal of Neuroscience*, 23(9):3697–3741, 2003.

[11] Jeff Hawkins and Sandra Blakeslee. *On Intelligence*. Times Books, 2004.

[12] Donald O. Hebb. *The organization of Behavior*. Wiley, 1949.

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

[14] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[15] John J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81:3088–3092, 1984.

[16] John J. Hopfield. Hopfield network. *Scholarpedia*, 2(5):1977, 2007.

[17] Nikola Kasabova, Kshitij Dhoblea, Nuttapod Nuntalid, and Giacomo Indiveri. Dynamic evolving spiking neural networks for on-line spatio- and spectro-temporal pattern recognition. *Neural networks*, 41:188–201, 2013.

[18] G. Leuba and R. Kraftsik. Changes in volume, surface estimate, three-dimensional shape and total number of neurons of the human primary visual cortex from midgestation until old age. *Anatomy and Embryology*, 190:351–366, 1994.

[19] Wolfgang Maass and Thomas Natschläger. Emulation of hopfield networks with spiking neurons in temporal coding. In James M. Bower, editor, *Computational Neuroscience*, pages 221–226. Springer US, 1998.

[20] David Marr. A theory of cerebellar cortex. *The Journal of Physiology*, 202(2):437–470, 1969.

[21] nobelprize.org. The nobel prize in physiology or medicine 1963. Visited on 14/04/14.

[22] Alvaro Pascual-Leone, Amir Amedi, Felipe Fregni, and Lotfi B Merabet. The plastic human brain cortex. *Annual review of neuroscience*, 28:377–401, 2005.

[23] R.W. Rodieck. Quantitative analysis of cat retinal ganglion cell response to visual stimuli. *Vision Research*, 5(12):583–601, 1965.

[24] S Rotter and M Diesmann. Exact digital simulation of time-invariant linear systems with applications to neuronal modeling. *Biological Cybernetics*, 81:381–402, 1999.

[25] Jesper Sjöström and Wulfram Gerstner. Spike-timing dependent plasticity. *Scholarpedia*, 5:1362, 2010.

[26] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: A structural description of the human brain. *PLoS Computational Biology*, 1(4):e42, 09 2005.

[27] Alan Turing. Intelligent machinery, 1948. First publicly published in 1968 in the book Cybernetics: Key Papers.