Department of Physics and Astronomy
University of Heidelberg

Bachelor Thesis in Physics
submitted by

Jannik Fehre

born in Münster (Germany)

October 2017

The Effect of Asymmetric Weight Variability on Sampling Processes based on Boltzmann Machines in Neuromorphic Hardware Applications

This Bachelor Thesis has been carried out by Jannik Fehre at the
Kirchhoff Institute for Physics in Heidelberg
under the supervision of
Prof. Karlheinz Meier

Abstract

The mathematically traceable Boltzmann machine approach as a description for neuronal networks requires symmetric synaptic connections. On neuromorphic hardware, which emulates leaky integrate-and-fire neurons, connections are unidirectional and the weight symmetry gets broken by individual weight variations. In order to simultaneously benefit from the framework of Boltzmann machines, which is successfully used in machine learning, and the inherent parallelism and speed-up compared to biology, a hardware emulation can sample approximately from a given Boltzmann distribution.

In this thesis, the effect of relative and asymmetric weight variations in sampling processes based on Boltzmann machines is investigated. For sampling with leaky integrate-and-fire neurons, those effects only prevail for variations above 20% against other limitations. Also the weight variations of three hardware devices designed by the Electronic Vision(s) Group are characterized. On the Spikey chip they exceed 50%, on the BrainScaleS System they lie about 16.4% and on the HICANN-DLSv2 chip they are 4.91%. This implies that sampling on the BrainScaleS System and the HICANN-DLS chip is not primarily restricted by weight variations.

Zusammenfassung

Der mathematisch ergründbare Ansatz einer auf Boltzmann-Maschinen beruhenden Beschreibung neuronaler Netzwerke erfordert symmetrische synaptische Verknüpfungen. Auf neuromorpher Hardware, welche leaky integrate-and-fire Neuronen emuliert, sind Verknüpfungen unidirektional und die Gewichtssymmetrie wird von individuellen Gewichtsschwankungen gebrochen. Um gleichzeitig von dem Framework der Boltzmann-Maschinen, welches erfolgreich im maschinellen Lernen Verwendung findet, und dem inhärenten Parallelismus und der hohen Geschwindigkeit verglichen zur Biologie zu profitieren, kann eine Hardwareemulation annäherungsweise eine gegebene Boltzmann-Verteilung abtasten.

In dieser Arbeit wird der Effekt von relativen und asymmetrischen Gewichtsschwankungen in Samplingprozessen, welche auf Boltzmann-Maschinen basieren, untersucht. Bei Sampling mit leaky integrate-and-fire Neuronen überwiegt er nur für Schwankungen oberhalb von 20% gegenüber anderen Einschränkungen. Zudem werden die Gewichtsschwankungen auf drei von der Electronic Vision(s) Group entwickelten Hardwaregeräten charakterisiert. Auf dem Spikey-Chip überschreiten sie 50%, auf dem BrainScaleS System liegen sie bei 16.4% und auf dem HICANN-DLSv2-Chip sind sie 4.91%. Dies suggeriert, dass Sampling auf dem BrainScaleS System und dem HICANN-DLS-Chip nicht primär von Gewichtsschwankungen eingeschränkt wird.

Contents

1 Introduction

1.1 Motivation

The functionality of the human brain [1] differs fundamentally from conventional computers and offers a lot of advantages over it, as for example a more efficient power consumption, an intrinsically parallel processing of information and a high level of robustness towards incomplete and noisy external input and damages in the brain itself.

While individual neurons show a deterministic behaviour in laboratory experiments, major networks of neurons exhibit stochastic dynamics. Although it is unknown if this stochasticity is "a bug or a feature" and if it is an emerging phenomenon or based on "true" randomness the brain somehow has access to, a descriptive model of the brain should include it as a key property.

In the first and necessary abstraction step, a manageable set of states the brain can take is defined. Then the time evolution of the network is interpreted as a sampling process in a probability landscape.

In machine learning this idea has been used very successfully in an abstract and simplified way with the concept of a Boltzmann machine [2] and Markov chain Monte Carlo sampling methods [3]. These have the advantage of being mathematically accessible for many applications, especially in the development of learning algorithms.

Petrovici et al. (2015) [4] succeeded in demonstrating that neuronal networks based on the leaky integrate-and-fire (LIF) model [5] can be configured in a way that they sample approximately from a target Boltzmann distribution and therefore benefit from the well known and powerful framework of machine learning.

As a member of the Human Brain Project, the Electronic Vision(s) Group at the Kirchhoff institute in Heidelberg designs neuromorphic hardware that emulates LIF neurons, and explores its applications. The hardware shares the same advantages of biological neuronal networks mentioned above, but runs significantly faster because of its purely electric nature.

Due to imperfect production processes the hardware components differ from their target specifications, thus each neuron and synapse is individual and not all variations can be compensated with calibrations. In particular, the synaptic connection strength is subject to those variations. Weights between two units are symmetric in a Boltzmann machine, but a connection between two neurons is unidirectional and the symmetry has to be implemented explicitly by using two in principal independent connections with the same strength, whose individual variations break the symmetry.

This thesis aims to contribute to the investigation of the consequences of asymmetric weight variations in sampling networks based on Boltzmann machines, both Gibbs sampling

and LIF sampling networks, and to characterize the weight variations on three different hardware devices, the Spikey chip, the BrainScaleS System and the HICANN-DLSv2 chip.

1.2 Outline

In the first part I give a brief introduction to the backgrounds. In particular these are Boltzmann machines, Markov chain Monte Carlo sampling, Gibbs sampling, the Kullback-Leibler divergence, biological neurons and synapses, the leaky integrate-and-fire model and the neuromorphic hardware used.

As a theoretical analysis of weight variations, I then take a look at the Markov chain which originates from the Gibbs sampling network designed to sample from a Boltzmann distribution but uses asymmetric weights.

In the next part, I present the effects on calculated and simulated stationary distributions of asymmetrically pertubated networks based on Boltzmann machines compared to the unpertubated Boltzmann distribution.

Then, I characterize the weight variations on the neuromorphic hardware via measurements of post-synaptic potentials for different combinations of neurons and synapse drivers and compare it to the results gained with the simulations.

Finally, I discuss the results and give a short outlook on further investigations.

1.3 Boltzmann machines and sampling methods

1.3.1 Boltzmann machines

A Boltzmann machine [2] is a set of n binary random variables together with a joint probability distribution. The two states of each random variable are defined as 0 and 1. Each random variable z_i is associated with a bias b_i and each pair of random variables $z_i, z_j, i \neq j$ with a weight $w_{ij} = w_{ji}$. The probability distribution $p(\boldsymbol{z})$ is defined as:

$$p(\boldsymbol{z}) = \frac{1}{Z} \exp(-E(\boldsymbol{z})) \quad \text{with} \quad Z = \sum_{\boldsymbol{z} \in \{0,1\}^n} \exp(-E(\boldsymbol{z})), \tag{1}$$

$$\text{where} \quad E(\boldsymbol{z}) = -\sum_{i=1}^{n} b_i z_i - \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} \frac{1}{2} w_{ij} z_i z_j. \tag{2}$$

The expression $E(\boldsymbol{z})$ can be interpreted physically as an energy of the whole system being in the state $\boldsymbol{z}$. It is also closely related to the Ising spin model [6].

The probability distribution describes a landscape in the state space with regions of

high and low probability and correlations between different units and groups of units in terms of its conditional distributions. Since there are 2^n possible states, the complexity of the Boltzmann machine grows rapidly with an increased number of units. That enables a Boltzmann machine to represent a broad spectrum of data sets in an abstract and convenient way, for example a set of images with a common motive [7]. The simple analytic expression for the probability distribution (1) makes learning algorithms mathematically traceable.

1.3.2 Markov chain Monte Carlo sampling

The downside of the complexity of Boltzmann machines are huge computational costs. In applications we are interested in conditional and marginal distributions to extract desired informations, but in larger networks it is not possible to compute them directly. One approach to this problem is sampling. The marginal distribution for a subset of the units is obtained by simply ignoring the other units, and the conditional distribution can be obtained by clamping the corresponding units to fixed values, which is called stochastic inference.

A Markov chain [3] is a random time series of states $z^{(0)} \to z^{(1)} \to z^{(2)} \to \dots$ drawn from a fixed countable set Ω, where transition probabilities $P(z^{(t+1)} = z|z^{(t)})$ are independent of the history and can only depend on the current state $z^{(t)}$. We restrict ourselves to discrete time and a finite state space. The time evolution of states is fully given by all transition probabilities from one state into another. While the chain evolves, it produces occurrences of states which define a probability distribution over all possible states. This distribution will be called sampled distribution.

Instead of considering one single state at each time step, we can introduce a probability distribution and use the transition probabilities to define a (deterministic) probability flow per time step. A probability distribution that does not change that way is called a stationary distribution.

Since we are interested in finite state spaces, we can enumerate the states as $z_0, z_1, z_2, \dots$, a probability distribution can be represented by a vector $\boldsymbol{p} \in [0,1]^{\#\Omega}$ with a 1-norm ($= \sum_i |p_i|$) equal to 1 and the transition probabilities $t_{ij} := P(z^{(t+1)} = z_i|z^{(t)} = z_j)$ can be combined to a quadratic matrix T such that

$$\boldsymbol{p}^{(t+1)} = T\boldsymbol{p}^{(t)}. \tag{3}$$

A distribution $\boldsymbol{p}$ is stationary if and only if

$$T\boldsymbol{p} = \boldsymbol{p} \tag{4}$$

holds. This is the condition for an eigenvector with the eigenvalue 1. In the case that T contains only strictly positive entries, which will be the case for all of our purposes, the Perron-Frobenius theorem [8] ensures that an unique real positive eigenvalue exists, whose eigenvector consists of non-negative entries. The matrix T preserves the 1-norm of $\boldsymbol{p}$ (if all entries are non-negative, as given here), therefore that eigenvalue must be 1 and the corresponding normalized eigenvector the unique stationary distribution.

Further, the non-zero transition probabilities assure the Markov chain to be aperiodic (all states can turn into themselves within one evolution step), which in turn tells us that its sampled distribution converges towards its stationary distribution.

This property is the reason why we can use Markov chains to approximate a target distribution. If the target distribution is the stationary distribution of the Markov chain, we can take a state, let it evolve and gather samples. This sampling method is called Monte Carlo sampling. It allows the usage of a complex probability distribution if we are able to construct a Markov chain with the desired stationary distribution.

While analytically the computation cost of Boltzmann distributions and their conditionals and marginals grows exponentially with the amount of units in the network, the needed time for the sampling process grows linearly (in the limit of many units).

1.3.3 Gibbs sampling

Gibbs sampling is a special case of Markov chain Monte Carlo sampling for joint distributions of two or more random variables z_i that offers a generic way to find a proper Markov chain making use of the conditional probabilities of one random variable given the state of all others.

A time step, in this context called global update, is composed of sequential local updates of the individual random variables according to their conditional probabilities:

$$P(z_i' = x) = p(z_i = x | \boldsymbol{z}_{\setminus i}). \tag{5}$$

Here $P(\dots)$ denotes the probability for a given event, z_i' the state after the local update of the i-th random variable, $p(\boldsymbol{z})$ the target stationary distribution and $\boldsymbol{z}_{\setminus i} = (z_1, z_2, \dots, z_{i-1}, z_{i+1}, \dots)^T$. The property $p(z_i = x|\boldsymbol{z}) = p(z_i = x|\boldsymbol{z}_{\setminus i})$ is a crucial one: The probability $P(z_i' = x)$ is not allowed to depend on z_i.

A local update of z_i satisfies the detailed balancing condition (here for $\boldsymbol{z} \in \{0,1\}^n$ where $\boldsymbol{z} = (z_0, \dots, z_{i-1}, 0, z_{i+1}, \dots)^T$ and $\boldsymbol{z}' = (z_0, \dots, z_{i-1}, 1, z_{i+1}, \dots)^T$)

$$P(\boldsymbol{z}'|\boldsymbol{z})p(\boldsymbol{z}) = \frac{p(\boldsymbol{z}')}{P(\boldsymbol{z}_{\setminus i})}p(\boldsymbol{z}) = \frac{p(\boldsymbol{z})}{P(\boldsymbol{z}_{\setminus i})}p(\boldsymbol{z}') = P(\boldsymbol{z}|\boldsymbol{z}')p(\boldsymbol{z}'). \tag{6}$$

This means that the probability flux from state $\boldsymbol{z}$ to state $\boldsymbol{z}'$ and the other way around is the same, therefore conserving their probabilities. That property extends to a global update, therefore (5) leads to the desired stationary distribution $p(\boldsymbol{z})$.

In the case of a Boltzmann distribution (1) we can calculate (5):

$$P(z_i' = 1) = p(z_i = 1|\boldsymbol{z}_{\setminus i}) = \frac{p(z_i = 1|\boldsymbol{z}_{\setminus i})}{p(z_i = 1|\boldsymbol{z}_{\setminus i}) + p(z_i = 0|\boldsymbol{z}_{\setminus i})} \tag{7}$$

$$= \frac{1}{1 + \frac{p(z_i=0|\boldsymbol{z}_{\setminus i})}{p(z_i=1|\boldsymbol{z}_{\setminus i})}} = \frac{1}{1 + \exp\left\{-b_i - \sum\limits_{\substack{j=1 \\ j\neq i}}^{n} w_{ji} z_j\right\}} \tag{8}$$

$$\Rightarrow \quad P(z_i' = 1) = \sigma(u_i(\boldsymbol{z})) \tag{9}$$

$$\text{with} \quad \sigma(x) = \frac{1}{1 + \exp(-x)} \tag{10}$$

$$\text{and} \quad u_i(\boldsymbol{z}) = b_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{ji} z_j. \tag{11}$$

The quantity (11) is called the abstract potential of the random variable z_i, since it corresponds directly to the membrane potential of a biological neuron (see section 1.4.1).

1.3.4 Kullback-Leibler divergence

In this thesis, stationary distributions of asymmetrically pertubated Gibbs samplers and their LIF counterparts (see section 1.4.3) will be compared to the Boltzmann distribution they originate from. In machine learning, an usual measure for the difference of two probability distributions p and q over a common state space Ω, the Kullback-Leibler divergence [9], is defined as:

$$D_{\mathrm{KL}}(p||q) = \sum_{x\in\Omega} p(x) \cdot \ln \frac{p(x)}{q(x)}. \tag{12}$$

The base of the logarithm is arbitrary in principal, since it is only a scale for the measure. Here we choose the natural logarithm because of better comparison to other results of the Electronic Vision(s) Group. [10]

The D_{KL} is asymmetric in its two arguments. It describes the degree of discrepancy when using the distribution q instead of p. In our case, the stationary distribution of a pertubated network is regarded as the true distribution, which is approximated by the ideal Boltzmann distribution for matching experiment and theory. A second reason for that assignment is the

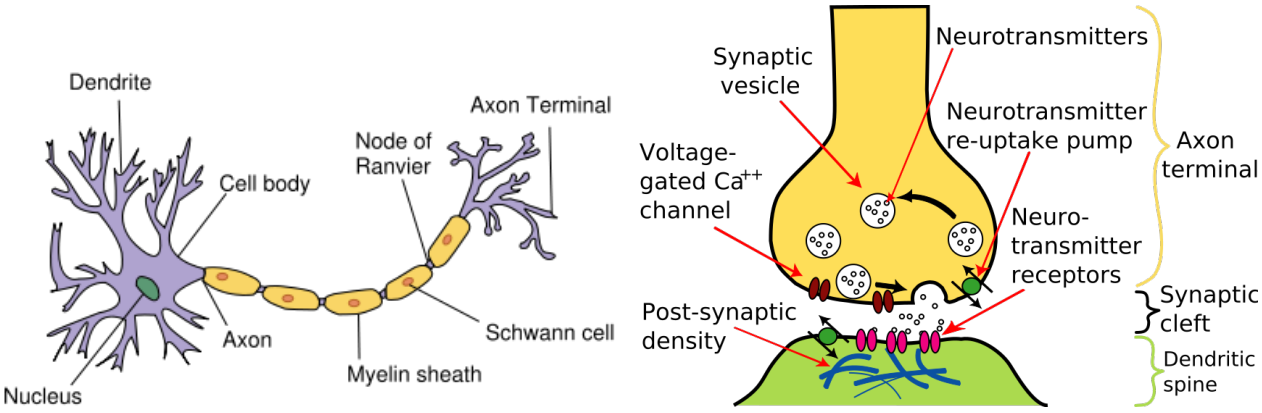

Figure 1: **a)** On the left side, the plot shows a sketch of a neuron. The nucleus can be found in the centre of the main cell body. The cell membrane ramifies and forms its dendrites. To the right the axon extends from the cell. It is wrapped in myelin sheaths to isolate the axon electrically from the outside. **b)** On the right side we can see a synaptic gap between a pre- and a post-synaptic neuron. The pre-synaptic neuron ends in an axon terminal. It is filled with vesicles that contain the neurotransmitter, in this case acetylcholine, and Calcium ion channels are located on its membrane. The post-synaptic neuron's dendrite's membrane has potassium and sodium ion channels with acetylcholine receptors. Images taken from [12] and [13].

limit

$$\lim_{p(x)\searrow 0} p(x) \cdot \ln \frac{p(x)}{q(x)} = 0, \tag{13}$$

that leads to a natural treatment of states which were not sampled at all. So all values for a Kullback-Leibler divergence will refer to p as the distribution of the pertubated network, while q is an analytic Boltzmann distribution.

1.4 Biological background

1.4.1 Neurons and synapses

Nerve cells [11], called neurons, are the central components of the human brain, it contains about a 100 billion of them. They are connected via synapses, through which they exchange information. In figure 1 we see the most important components of a neuron and a synapse.

A neuron has dynamic properties that modulate the voltage between its inside and the outside medium across its membrane, which is called membrane potential. The so called Na^+-K^+-pump transports 3 Na^+-ions to the cell outside and 2 Ka^+-ions to the cell inside per pump cycle, therefore the resting potential is negative. Permanently opened ion channels stabilize the process. This process maintains a chemical potential in order to be able to quickly decrease and increase the membrane potential when it opens a corresponding ion

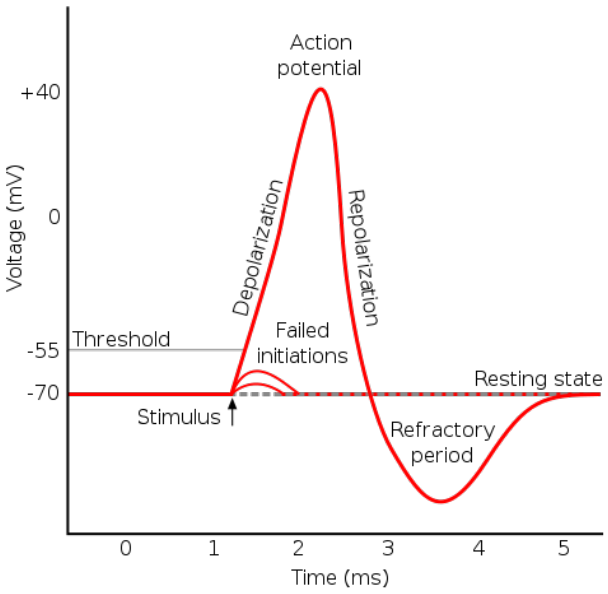

Figure 2: A spike is a sharp electric peak, the neuron first increases its membrane potential strongly and then causes a quick drop-off even below the resting potential. There it takes a while to recover the chemical potential and to return to the resting state. In between, it hardly reacts to further input, hence the neuron is refractory. Image taken from [14].

channel.

When the membrane potential reaches a threshold, the neuron triggers a spike (this description is an approximation; the mechanism is more complex and under special circumstances a spike can be triggered without reaching the typical threshold). The stereotypical shape of a spike, also called action potential, can be seen in figure 2.

The spike travels down the neuron's axon and effects all post-synaptic neurons via synapses.

In a chemical synapse (see figure 1), an arriving spike opens Ca^{++}-ion channels in the membrane of the axon which cause vesicles filled with neurotransmitters to merge with the membrane and release the neurotransmitter into the synaptic gap. There it diffuses towards the dendrite of the post-synaptic neuron and is detected by specific receptors which in turn open attached ion channels. Depending on the ion type, the effect and the synapse are called either excitatory, if the membrane potential increases, or inhibitory, if it decreases. The resulting membrane potential and its time course is called post-synaptic potential (PSP).

Each synapse holds a sensitivity that regulates the synaptic process individually. This synaptic weight is not fixed but is subject to changes the brain performs in learning processes to improve its ability to solve certain tasks. This phenomenon is called synaptic plasticity.

1.4.2 The leaky integrate-and-fire neuron model

The behaviour of a biological neuron is complex due to the amount of components contributing to it. A precise description is given by the Hodgkin-Huxley neuron model [15], but it

uses four differential equations, and therefore it is hard to do exact computations. Another model, the Leaky Integrate-and-Fire neuron model (LIF) [5], offers the simplest description which still preserves crucial biological properties. Due to only one single linear differential equation, it is a mathematically convenient choice.

A neuron's dynamics are encoded by one scalar, the membrane potential u, and its electrical properties are abstracted to one quantity, the capacity C_m. Therefore it obeys the capacitor equation

$$C_\mathrm{m}\frac{\mathrm{d}u}{\mathrm{d}t} = I_\mathrm{tot}, \tag{14}$$

where I_tot is the total current applied to the neuron. The equilibrium mechanism described in the previous section that pushes the membrane potential to a resting potential is represented by a leak potential E_l which is coupled to the neuron trough a leakage conductance g_l:

$$I_\mathrm{l} = g_\mathrm{l}(E_\mathrm{l} - u). \tag{15}$$

There are two possible ways of modelling synaptic input. The first is called current based (CUBA). An incoming spike leads to a current on the membrane

$$I^\mathrm{syn} = I_0 \sum_k w_k \sum_s \epsilon(t - t_s), \tag{16}$$

where I_0 is some unit contributing and normalizing constant. The first sum goes over all pre-synaptic partners k and w_k is the corresponding synaptic strength. The second sum goes over all spikes s of that neuron and ϵ is the synaptic kernel. Its shape depends on the decision of the modellers, it can be a delta function, a decaying exponential function or a difference of exponential functions.

In the second synapse model, the conductance based (COBA) approach, a synaptic input couples the membrane to a reversal potential E^rev through a time dependent conductance. Because a conductance can only be positive, we have to distinguish between excitatory and inhibitory reversal potentials E^rev:

$$I^\mathrm{syn} = g_0 \sum_{x\in\{\mathrm{exc,\ inh}\}} (E_x^\mathrm{rev} - u) \sum_k w_k \sum_s \epsilon(t - t_s), \tag{17}$$

where again g_0 is a constant, $E_\mathrm{exc}^\mathrm{rev} > E_\mathrm{l}, E_\mathrm{inh}^\mathrm{rev} < E_\mathrm{l}$ and the rest as in the CUBA case, except that the sum over k only includes the synapses of type x.

Additionally, an external current I^ext can be included to I_tot which is a way to model

sensoric input to the neuron. In total, the capacitor equation (14) then reads:

$$C_{\mathrm{m}}\frac{\mathrm{d}u}{\mathrm{d}t} = I_{\mathrm{l}} + I^{\mathrm{syn}} + I^{\mathrm{ext}}. \tag{18}$$

Finally a firing mechanism is added with the help of a threshold potential ϑ, a refractory time constant τ_{ref} and a rest potential ϱ, together with

$$\text{neuron spikes at } t_s \Leftrightarrow u(t_s = \vartheta) \tag{19}$$

$$u(t_s < t \leq t_s + \tau_{\mathrm{ref}}) = \varrho \text{ for all spikes } s. \tag{20}$$

Later on, the shape of a post-synaptic potential will be important. It is dominated by two time constants, the synaptic time constant τ_{syn} that appears in the synaptic kernel and the membrane time constant

$$\tau_{\mathrm{m}} = \frac{C_{\mathrm{m}}}{g_{\mathrm{l}}}. \tag{21}$$

Then the time course of a post-synaptic potential for a single excitatory spike at t_{s} with weight w and a exponential decaying synaptic kernel is (see [4] for a derivation):

$$u(t) = \Theta(t - t_{\mathrm{s}})\frac{w}{C_{\mathrm{m}}}\frac{\tau_{\mathrm{m}}\tau_{\mathrm{syn}}}{\tau_{\mathrm{syn}} - \tau_{\mathrm{m}}}\left[\exp\left\{-\frac{t - t_{\mathrm{s}}}{\tau_{\mathrm{syn}}}\right\} - \exp\left\{-\frac{t - t_{\mathrm{s}}}{\tau_{\mathrm{m}}}\right\}\right]. \tag{22}$$

In the COBA case, τ_{m} has to be replaced with τ_{eff} where g_{l} is replaced with the total conductance

$$g_{\mathrm{tot}} = g_{\mathrm{l}} + g_{\mathrm{syn}}. \tag{23}$$

For COBA neurons, formula (22) is an approximation in the limit $|E^{\mathrm{rev}}_{\mathrm{exc}} - E_{\mathrm{l}}| \gg \sup_t |u(t) - E_{\mathrm{l}}|$.

1.4.3 Sampling with leaky integrate-and-fire neurons

As discussed in the previous section, single LIF neurons behave deterministically. In order to be able to use LIF neurons to sample from a target distribution (see [4]), we have to make them stochastic. This is done by feeding them with high frequent and Poisson distributed external input, excitatory and inhibitory separately. A typical time course of the resulting membrane potential is shown in figure 3 on the left side.

Also, the mapping to a binary interpretation can be seen. We define the neuron's state to be 1 when it is refractory and 0 otherwise. By varying the resting potential with all other parameters fixed, the neuron spends a varying fraction of the time in its refractory period, which corresponds to its probability to be in the state 1. This dependency leads to the so called activation function. A logistic function (see (10)) can be fitted to it. It is a good

approximation, although in detail the activation function has a deviating shape. In figure 3 on the right side a typical activation function is shown.

In the fit we identify the abstract membrane potential u_k (see (11)) with the neuron's leak potential E_l via

$$u_k = \frac{E_\mathrm{l} - E_\mathrm{l}^0}{\alpha}. \tag{24}$$

The fit parameters E_l^0 and α enable us to translate the abstract quantities bias and weight to the LIF domain. Without further input, the abstract membrane potential u_k is equal to the bias b_k:

$$u_k = b_k = \frac{E_{\mathrm{l},k} - E_\mathrm{l}^0}{\alpha} \tag{25}$$

$$\Rightarrow E_{\mathrm{l},k} = \alpha b_k + E_\mathrm{l}^0 \tag{26}$$

If we require the mean influence on the membrane potential to be the same as in the abstract domain during the refractory period τ_ref,

$$w_{ij}^\mathrm{bio} \tau_\mathrm{ref} \alpha \overset{!}{=} \int_0^{\tau_\mathrm{ref}} PSP(t)\mathrm{d}t, \tag{27}$$

and set $\tau_\mathrm{syn} = \tau_\mathrm{ref}$ it follows the translation for the weights (for COBA LIF neurons):

$$w_{ij}^\mathrm{bio} = w_{ij}^\mathrm{theo} \frac{\langle g_\mathrm{tot} \rangle}{g_\mathrm{l} \alpha C_\mathrm{m}} \frac{E_{ij}^\mathrm{rev} - \langle u \rangle}{1 - \frac{\tau_\mathrm{syn}}{\tau_\mathrm{eff}}} \left[\tau_\mathrm{syn} \left(\frac{1}{e} - 1 \right) - \tau_\mathrm{eff} \left(\exp\left\{ -\frac{\tau_\mathrm{syn}}{\tau_\mathrm{ref}} \right\} - 1 \right) \right]. \tag{28}$$

The LIF network then samples approximately from a given Boltzmann distribution, where samples typically are taken in time steps of one refractory period. Therefore we can keep the descriptive power of Boltzmann machines, while at the same time dealing with a system that can be transferred to parallel neuromorphic hardware.

The strength of the noise input with given frequency determines α, stronger excitations and inhibitions lead to a smaller value for α because small changes in E_l mean big changes in the activation probability. This dependency is complex and in general not linear. [4]

1.5 Neuromorphic hardware

The neuromorphic hardware designed by the Eletronic Vision(s) Group emulates LIF neurons (or more general models that include LIF neurons) and therefore follows an alternative computation paradigm compared to established computers. Each neuron's membrane potential is represented physically by a capacitor and its dynamics, as for example the coupling

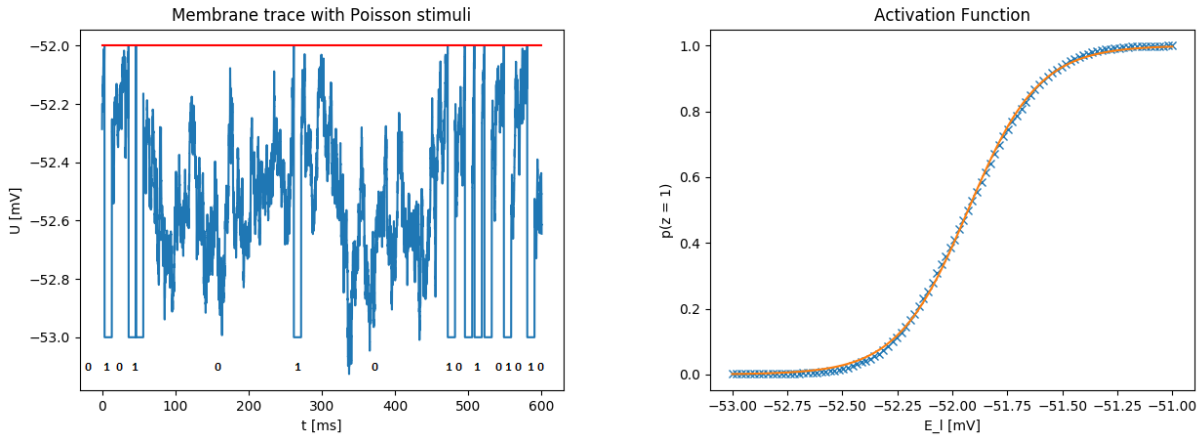

Figure 3: **a)** On the left side we see the time course of the membrane potential of a neuron with excitatory and inhibitory Poisson stimuli with $\nu = 5000$ Hz. The red line marks the spiking threshold ϑ. On the bottom the assigned binary state is added. **b)** On the right side the activation function of the same neuron is plotted (blue crosses). By varying the leak potential E_l the mean time spent in the 1-state changes too. A logistic function (orange curve) is fitted to the data points. Both simulations where done with NEST v2.2.2 [16] and the PyNN interface [17]. Parameters are listed in the appendix A.1.

to a reversal potential, is performed locally. So the system inherently works in parallel and execution times do not increase with the network size. Spikes are transferred via synaptic drivers as digital informations and their effect is produced locally. Synapses are either excitatory or inhibitory and carry an individual digital weight. As in contrast to biological neurons and synapses, electric currents and information transports do not rely on ion streams and the diffusion of neurotransmitters, the hardware has a huge speed-up compared to biology of about 10^3 to 10^5.

Spikey [18] is the first neuromorphic chip based on spiking neurons the Electronic Vision(s) Group has designed. It emulates COBA LIF neurons. A chip consists of 384 neurons and 256 synapse drivers which can be connected almost freely. The synaptic weight resolution is 4 bit.

The long term goal is the large scale integration of more than one chip for building large networks. In the production process, multiple identical chips are placed on one wafer and later separated. For the BrainScaleS System [19] the wafer is left intact and the chips get connected to form one single network. On one wafer there are 48 reticles, per reticle 8 HICANN chips, the successor of Spikey, and per HICANN chip 512 neurons, thus in total 196608 neurons. It is able to emulate COBA LIF neurons and the synaptic weight resolution is 4 bit.

The idea behind the development of the HICANN-DLS chip [20] is to implement learn-

ing algorithms on chip, since the data transfer from the chip to the host computer, their evaluation and the transfer back to the chip in learning procedures consume the most time in preparing a network to perform a certain task. Pre- and post-synaptic events are traced, stored and interpreted locally to apply weight updates. As a prototype chip it is subject to current development. HICANN-DLS v2, which was used for this thesis, emulates CUBA LIF neurons and the synaptic weight resolution is 6 bit.

2 Gibbs like sampling with asymmetric weights

For a Boltzmann machine, the symmetry in the weights is a direct consequence of its physical relationship. In physics, interactions are symmetric in the sense that we can define an energy, and the concept of a Boltzmann machine is based on the notion of an energy for each state. If we would define the weights connecting two units asymmetrically, their difference would average out in the energy expression (2), yielding the same probability distribution as the averaged network.

In theory, the construction of the Boltzmann machine comes first and sampling methods as a computational instrument come second. On the other side, in a biological neuronal network, as well as in the neuromorphic hardware, a connection from neuron A to neuron B is inherently independent from the connection the other way around. The consequences of variations in the weights are clear in terms of the behaviour of a single neuron. The mapping between neurons and abstract binary random variables still works. The description of a synaptic connection as a contribution to the (abstract) membrane potential that determines the activation or spiking probability of the neuron is not affected by asymmetry. The only thing that changes is the probability distribution the network now samples from. Therefore the description of the network as a Markov chain with binary random variables and known update probabilities shall be the starting point.

We aim to find the stationary distribution. We index the states $\boldsymbol{z}$ as

$$\begin{aligned} \boldsymbol{z}_0^I &= (0, \dots, 0, 0, 0)^T \\ \boldsymbol{z}_1^I &= (0, \dots, 0, 0, 1)^T \\ \boldsymbol{z}_2^I &= (0, \dots, 0, 1, 0)^T \\ \boldsymbol{z}_3^I &= (0, \dots, 0, 1, 1)^T \\ &\vdots \\ \boldsymbol{z}_{2^n-1}^I &= (1, \dots, 1, 1, 1)^T \end{aligned} \tag{29}$$

with corresponding state probabilities a_k, $k = 0, 1, \dots 2^n - 1$. Next we shorten (9) for a given time t with p_i^k and introduce $q_i^k = 1 - p_i^k$, both depending on the current state $\boldsymbol{z}_k$ and all before updated units $z_j, j < i$. The latter dependency will be encoded implicitly in the preceding factors in a product (for example in $\{p_1^k \textbf{ or } q_1^k\} \cdots p_i^k \cdots$ the underlying state $\boldsymbol{z}$ for p_i^k will be composed by $z_1 = \{1 \textbf{ or } 0\}, \dots$ and $z_j = z_{k,j}^I$ for $j > i$).

The transition probability for a global update is the product of transition probabilities

for local updates, therefore the stationary condition (4) now reads:

$$\begin{pmatrix} q_1^0 \cdots q_{n-1}^0 q_n^0 & q_1^1 \cdots q_{n-1}^1 q_n^1 & \dots & q_1^{2^n-1} \cdots q_{n-1}^{2^n-1} q_n^{2^n-1} \\ q_1^0 \cdots q_{n-1}^0 p_n^0 & q_1^1 \cdots q_{n-1}^1 p_n^1 & \dots & q_1^{2^n-1} \cdots q_{n-1}^{2^n-1} p_n^{2^n-1} \\ \vdots & \vdots & & \vdots \\ p_1^0 \cdots p_{n-1}^0 p_n^0 & p_1^1 \cdots p_{n-1}^1 p_n^1 & \dots & p_1^{2^n-1} \cdots p_{n-1}^{2^n-1} p_n^{2^n-1} \end{pmatrix} \cdot \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{2^n-1} \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{2^n-1} \end{pmatrix}. \quad (30)$$

Since $q_1^k \cdots q_{n-1}^k q_n^k + q_1^k \cdots q_{n-1}^k p_n^k + \cdots + p_1^k \cdots p_{n-1}^k p_n^k = (q_1^k \cdots q_{n-1}^k + q_1^k \cdots q_{n-1}^k + \cdots + p_1^k \cdots p_{n-1}^k)(p_n^k + q_n^k) = \cdots = \prod_{i=1}^{n}(p_i^k + q_i^k) = \prod_{i=1}^{n} 1 = 1 \quad \forall k$, the sum of all rows in (30) yields a trivial expression. This degree of freedom corresponds to the normalization of the eigenvector $\boldsymbol{a}$. We therefore can set $a_0 = 1$, eliminate the first row and find a formula for the (not yet normalized) solution $\boldsymbol{a}$:

$$\begin{pmatrix} a_1 \\ \vdots \\ a_{N-1} \end{pmatrix} = - \left[\begin{pmatrix} q_1^1 \cdots q_{n-1}^1 p_n^1 & \dots & q_1^{2^n-1} \cdots q_{n-1}^{2^n-1} p_n^{2^n-1} \\ \vdots & & \vdots \\ p_1^1 \cdots p_{n-1}^1 p_n^1 & \dots & p_1^{2^n-1} \cdots p_{n-1}^{2^n-1} p_n^{2^n-1} \end{pmatrix} - \mathbb{1} \right]^{-1} \cdot \begin{pmatrix} q_1^0 \cdots q_{n-1}^0 p_n^0 \\ \vdots \\ p_1^0 \cdots p_{n-1}^0 p_n^0 \end{pmatrix} \quad (31)$$

We see that the system is still stable. Boltzmann distributions are embedded in the above considerations and the solution for the eigenvector problem is continuous in the matrix entries which are continuous in all network parameters. Small variations in the weights lead to small variations in the sampled distribution, even if weights vary asymmetrically.

3 Effects of weight variations in networks based on Boltzmann machines

In section 4 we will motivate the investigation of relative weight variations instead of absolute ones. For different Boltzmann machines we pertubate the weights, then do Gibbs sampling and compare the resulting stationary distribution to the original Boltzmann distribution as discussed in section 1.3.4. This will yield one D_{KL} value each time. The weights w_{ij} and w_{ji} will be treated independently, so the pertubations will be asymmetrically.

This investigation focusses on asymmetric weights due to variations only in themselves and not in biases. Those have an important role in the network's stationary distribution, so each network will be studied additionally while setting all biases to 0.

On hardware, there are correlations between different effective synaptic weights, because groups of neurons share the same synaptic driver. Here we simplify the situation and assume independent and relatively equal weight variations which are normal distributed. Within this simplification, the mean variability of the network is represented by one scalar. We choose the standard deviation of one individual variation to quantify it. Therefore, for one computation or simulation, all weights are pertubated independently with a multiplicative factor drawn from a normal distribution with $\mu = 1$ and the standard deviation σ.

The abstract parameters of typical networks lie between -2 and 2 or are even smaller [4]. At this point we are not able to use weights with an asymmetry between positive and negative values or to impose a special architecture like a restricted Boltzmann machine [21], because that would open a further degree of freedom and exceed the scope of this work. So we consider fully connected Boltzmann machines with randomly drawn and symmetrically distributed parameters. The beta distribution [22] with parameters $\alpha = \beta = 0.5$ is a sensible choice for the underlying distribution because it favours large values. In more detail, parameters are drawn from $2d \cdot (\mathcal{B}(0.5, 0.5) - 0.5)$ with one open coefficient d. In all calculations and simulations, there are 10 values for the parameter width d, ranging from 0.2 to 2.0 in steps of 0.2. The network size preferably should be as large as possible, but in order to avoid long simulation times we use 16 neurons. Because of computational restrictions the analytic calculations are done for $n = 10$.

For each combination of d and σ we do calculations and simulations for different network parameters. In the LIF case, the time did not allow to repeat the simulation for one network to obtain different weight variations, but in the other two cases there are repetitions. The amount of data is not sufficient to separate the influence of network parameters and specific weight variations, therefore all results for the same values of d and σ are treated the same way and get averaged. As a measure for the error we use the usual standard deviation of

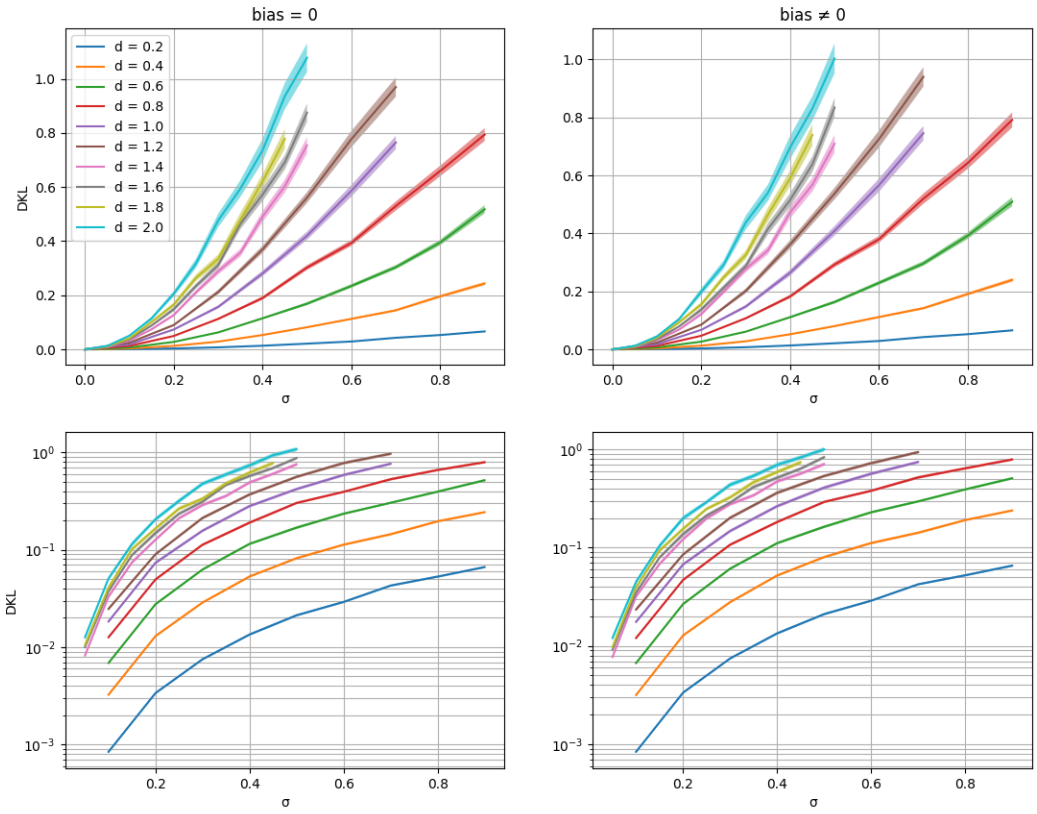

Figure 4: The D_{KL} between the computed distribution of the pertubated Boltzmann machine and the distribution of the original network in dependence of the standard deviation σ of the pertubation for different widths of network parameters (colours), without (left) and with (right) biases, linear (top) and logarithmic (bottom) y-axis.

the sample mean.

3.1 Analytic computations of stationary distributions

We consider 10 neurons and compute the exact stationary distribution of the pertubated network with (31) directly. Per value for d and σ we use 40 networks and do 5 repetitions.

The results can be seen in figure 4. There is no notable difference between the case with and the case without biases. The curves have a convex shape and therefore grow faster than linear, but slower than exponential as is obvious looking at the logarithmic scale. Networks with larger parameters are influenced stronger due to the variations being relative.

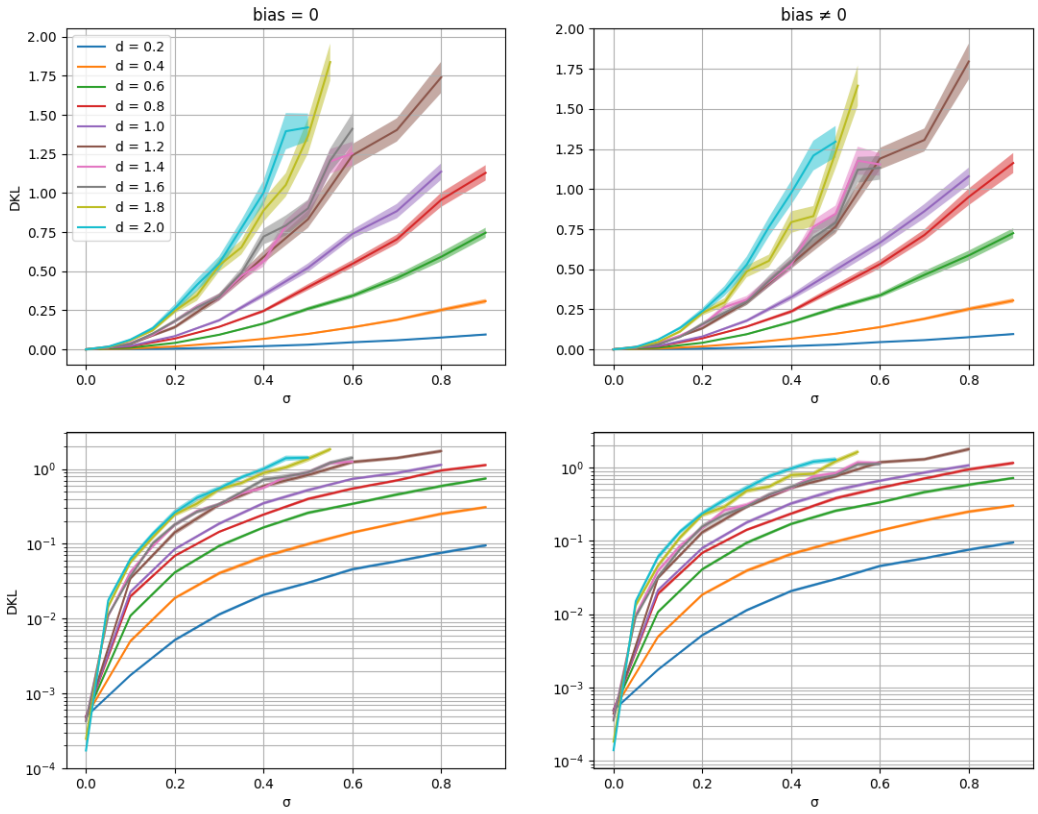

Figure 5: The D_{KL} between the sampled distribution of the pertubated Boltzmann machine and the distribution of the original network in dependence of the standard deviation σ of the pertubation for different widths of network parameters (colours), without (left) and with (right) biases, linear (top) and logarithmic (bottom) y-axis.

3.2 Gibbs like sampling

For 16 neurons we use Gibbs sampling with 2^{26} samples in order to get the stationary distribution of the pertubated Boltzmann machine. For each value for d there are 10 different networks. For each network, we perform 6 simulations with different weight variations.

The result can be seen in figure 5. In comparison to figure 4, the values for D_{KL} are larger because our knowledge of the stationary distribution is imperfect due to sampling. Also we got finite values for $\sigma \to 0$, especially for small d. This is probably caused by the high amount of states with no negligible probability. Aside from those differences, the curves look qualitatively the same. Here the bias also makes no difference.

3.3 Sampling with leaky integrate-and-fire neurons

LIF neurons are simulated with the software NEST v2.10.0 [23]. The configuration for using LIF neurons to sample from a Boltzmann distribution, or in our case from the distribution (31), is done by the software tool SBS (spike based sampling) v1.5.2 [24]. Parameters are listed in the appendix A.2.

We use the same 100 networks with 16 neurons as for the Gibbs sampling, but for each network we only do one simulation. The simulation duration is 1 hour biological time.

In figure 6 we see the results. The region with small σ, where we observe the most differences to figure 5, especially for large network parameters, is dominated by deviations due to the approximations in sampling with LIF neurons. With increasing σ, the pertubation dominates the deviation and the D_{KL} approach the same order of magnitude as in the earlier results and the curve shapes get similar. Roughly, the transition happens around $\sigma \approx 0.2 - 0.3$.

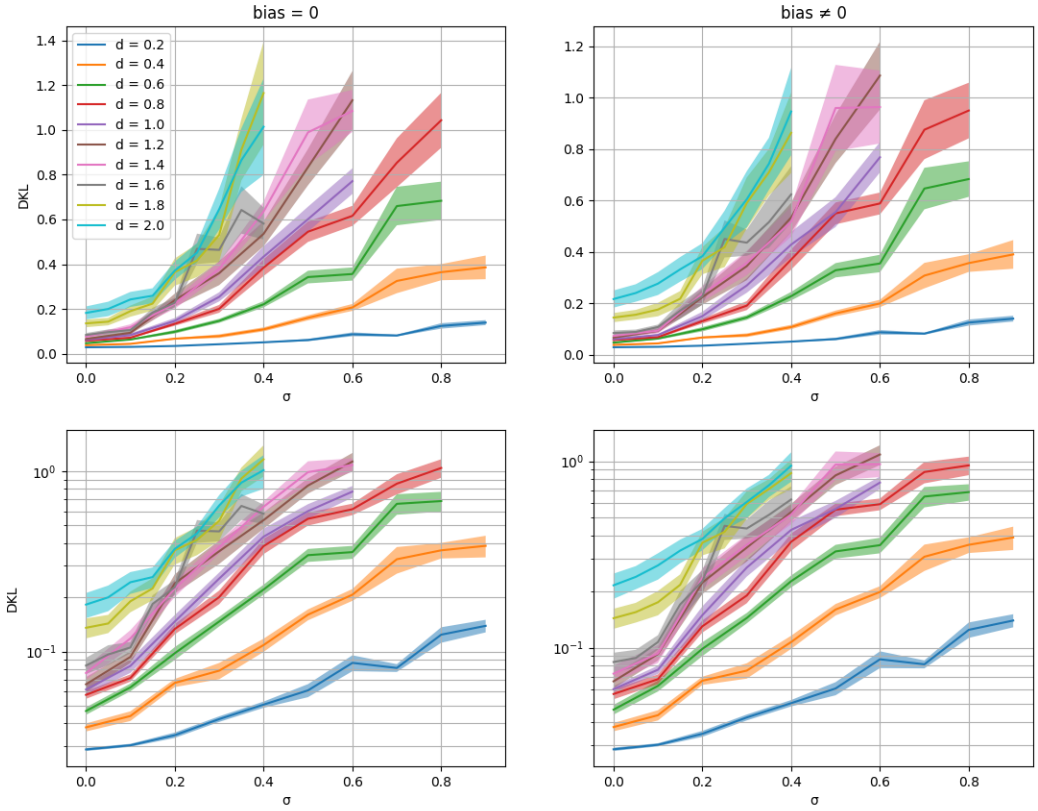

Figure 6: The D_{KL} between the sampled distribution of the pertubated LIF network and the Boltzmann distribution of the original network in dependence of the standard deviation σ of the pertubation for different widths of network parameters (colours), without (left) and with (right) biases, linear (top) and logarithmic (bottom) y-axis.

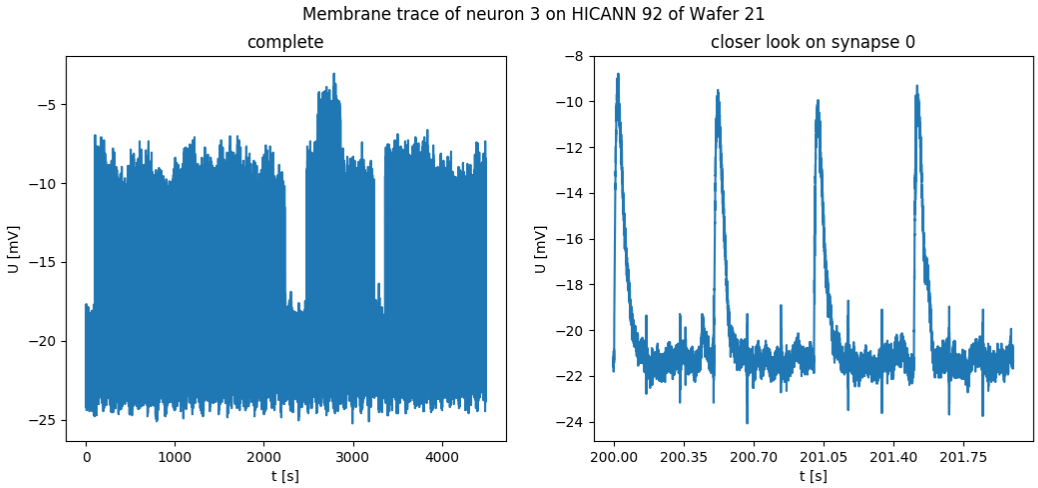

Figure 7: The membrane trace of one neuron on the BrainScaleS System. The units refer to the biological domain. **a)** On the left side all data is shown. It starts with a period where no spikes are sent to the neuron in order to get the resting potential. **b)** On the right side we see four single post-synaptic potentials.

4 Weight variability on three different hardware devices

As the variation of network parameters on the hardware depends on multiple factors and affects different quantities, the question arises how to characterize it. Since in the end we want to compare the synaptic connection strength to the corresponding abstract weight in the theoretical domain of Boltzmann machines, it would be necessary to first build up the stochasticity with the help of Poisson input, measure the activation function and then compute the conversion factor between biological and theoretical parameters, if we wanted to consider absolute values for the weights and their variations.

Without noise or other input the shape of one single post-synaptic potential is linear in the synaptic weight (see (22)), which is proportional to its abstract counter part (see (28)). That throughout linearity allows us to investigate relative weight variations, so we not need to worry about units and constant factors. We measure single post-synaptic potentials with a fixed digital hardware weight and determine their synaptic strength. For no particular reason, we choose them to be excitatory.

Parameters are listed in the appendices B.1, B.2 and B.3, respectively.

Since measurements are not perfect, for example due to readout and thermal influences, we measure 250 post-synaptic potentials. An example trace can be seen in figure 7. The 250 post-synaptic potentials then get averaged.

The shape of a post-synaptic potential depends on multiple quantities and their variations have different effects on the network's dynamic. In this thesis, we investigate variations in

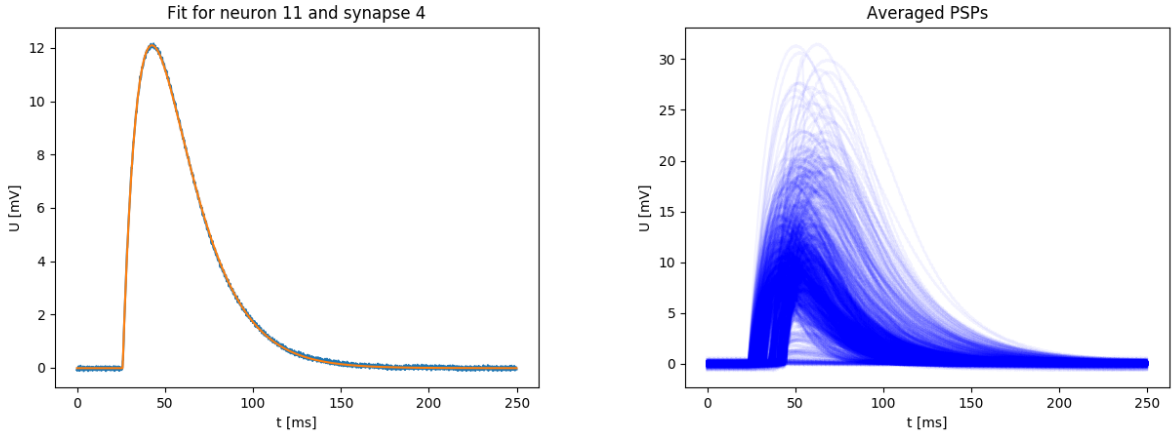

Figure 8: Averaged PSPs on the BrainScaleS System. The units refer to the biological domain. The resting potential is set to 0 V. **a)** On the left side the averaged PSP for neuron 11 and synapse 4 (blue) and the fit (orange) are shown. **b)** On the right side are the averaged PSPs for all measured combinations of neurons and synapses.

the multiplicative strength of synaptic connections, that is the scalar factor w in (22). It can be extracted from the whole post-synaptic potential via fitting. The term $\frac{1}{C_\mathrm{m}}$ cannot be separated from w, so we combine them into one parameter

$$A = \frac{w}{C_\mathrm{m}}. \tag{32}$$

C_m is assumed to be constant and therefore will cancel out in relative quantities. The remaining parameters t_s, τ_m, τ_syn and an offset b will be ignored in further evaluations. A typical fit is shown in figure 8 on the left side; it matches the data well.

This measurement is repeated for different pairs of neurons n and synapses s, therefore yielding values A_{ns}. The square root of the corresponding diagonal element of the covariance matrix gives an error ΔA_{ns}.

As the effective weight, concerning the influence on the activation probability, is determined relatively to the Poisson input, a global factor on all synaptic weights specific for each single neuron cancels out to some extend. There is no linear dependency (as mentioned in section 1.4.3), but it is monotonous. An individual calibration (measuring the activation function for each neuron in terms of the digital weight) would even compensate such a factor completely.

Therefore, we first compute the relative standard deviation for each neuron separately,

$$\sigma_n = \frac{\mathrm{std}_s(A_{ns})}{\mu_n}, \tag{33}$$

$$\text{where} \quad \mu_n = \mathrm{mean}_s(A_{ns}), \tag{34}$$

and then take the mean:

$$\sigma = \mathrm{mean}_n(\sigma_n). \tag{35}$$

The errors $\Delta\sigma_n$ are derived via Gaussian error propagation to

$$\Delta\sigma_n = \frac{1}{S_n \mu_n} \sqrt{\sum_{i=1}^{S_n} \left[\Delta A_{ni} \left(\frac{1}{\mu_n \sigma_n} \sum_{j=1}^{S_n} (A_{nj} - \mu_n) \left(\delta_{ij} - \frac{1}{S_n} \right) - \sigma_n \right) \right]^2}, \tag{36}$$

$$\text{where} \quad S_n = \#_s A_{ns} \tag{37}$$

$$\Rightarrow \Delta\sigma = \frac{1}{N} \sqrt{\sum_{i=1}^{N} (\Delta\sigma_n)^2}, \tag{38}$$

$$\text{where} \quad N = \#\sigma_n. \tag{39}$$

Since all errors $\frac{\Delta\sigma}{\sigma}$ are smaller than 1%, they will be neglected.

The three hardware devices will be presented in order of increasing difficulty of dealing with outlier values. Clearly, some components are broken, as can be seen in figure 8 on the right side, and should be excluded from calculations. Others are not completely broken, but give extreme results. The indication of a final value for σ will be a trade-off between the amount of considered units, which therefore are declared as "usable", and the weight variation in comparison to the D_{KL}'s from simulations.

First we start with HICANN-DLS, where the measurement was performed for 32 neurons and 32 synapses by Yannik Stradmann. In figure 9, a histogram of all values A_{ns} is plotted. They are normalized to their mean value.

There are two clearly separated peaks. So making a cut-off in between (in the plot arbitrarily set at 0.4) and then discarding the small peak at 0.1 is a natural choice. 32 of the 33 discarded values stem from neuron 19.

We are left with:

$$\sigma_{\mathrm{DLS}} = 4.91\%. \tag{40}$$

Next we take a look at the BrainScaleS System. 35 neurons and 35 synapses were measured. The distribution of A_{ns} is shown in figure 10 on the left side.

Here we also get two peaks, but this time they intersect each other. Since the left peak

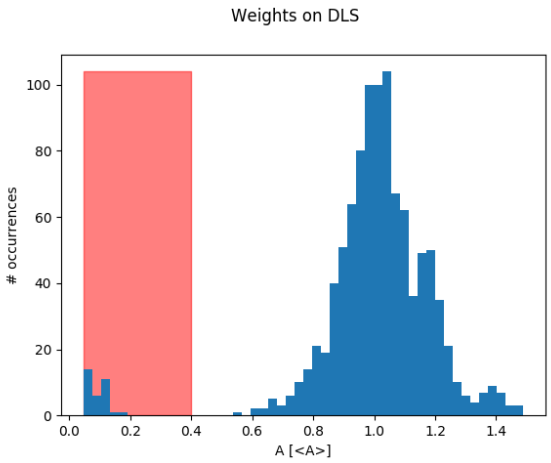

Figure 9: Weight distribution on HICANN-DLS. Beside an apparently normal distributed peak around 1, there is a second, small peak at 0.1. With one exception, they all stem from neuron 19. They will be discarded (red area) since the peaks are clearly separated without intersection.

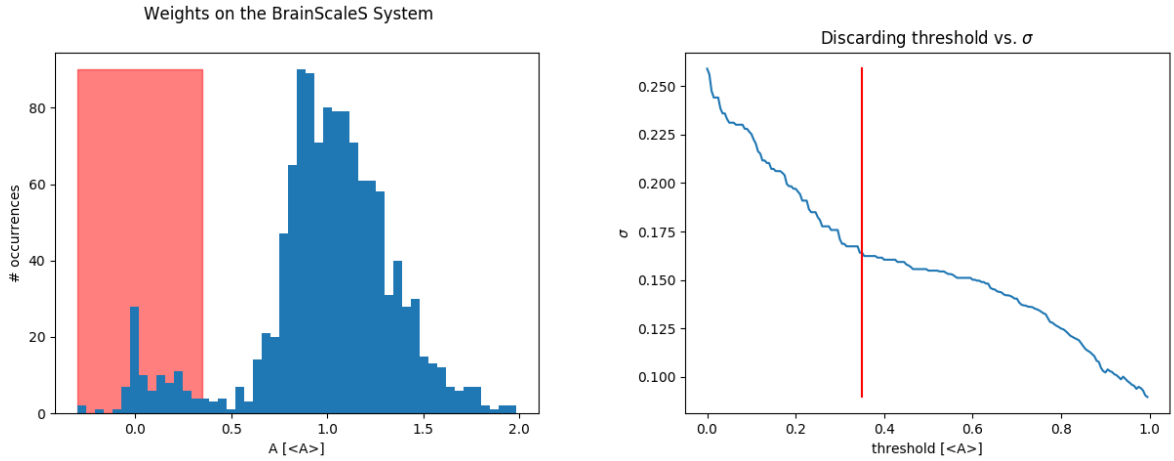

Figure 10: **a)** On the left side the weight distribution on the BrainScaleS System is shown. The red area marks the discarded values. **b)** On the right side is the resulting relative standard deviation σ in dependence of the discarding threshold. At 0.35 there is a notable bend.

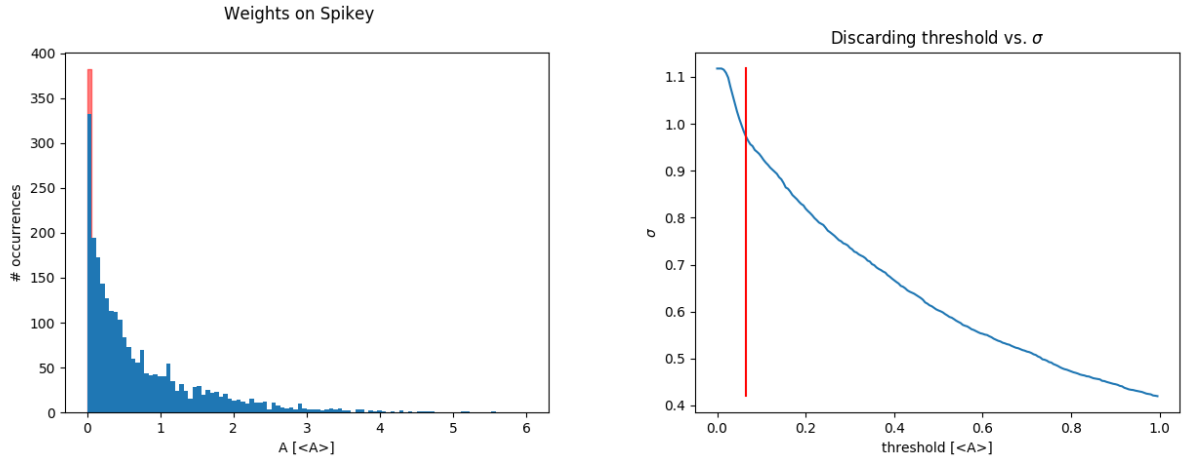

Figure 11: **a)** On the left side the weight distribution on Spikey without synapse 27 is shown. The red area shows the discarded values. **b)** On the right side is the resulting relative standard deviation σ in dependence of the discarding threshold, after discarding all values above 6. An arguable bend can be seen around 0.065.

lies around 0 and even includes negative values, we again choose a threshold and discard all values below. In figure 10 on the right side we see the resulting value for σ while varying the threshold. Around 0.35 the curve bends and starts falling more gently. It marks the transition from real outliers to normal boundary values, so we do the cut-off there and get:

$$\sigma_{\text{BrainScaleS}} = 16.4\%. \tag{41}$$

96 of 1225 pairs of neurons and synapses were discarded this way. Mainly, they include neuron 8 or synapse 17, 18 or 25.

On Spikey 50 neurons and 50 synapses were measured. The histogram and the threshold curve are shown in figure 11.

Instead of one or two peaks, we see an exponential decay with an accumulation near 0. There is a more than 3 units large gap around 6, followed by all 50 values of synapse 27 which have a mean value of 9.7. They will be discarded before the threshold curve is evaluated.

Besides an arguable bend around 0.065, the threshold curve is very regular. Doing a cut-off at that value, we get:

$$\sigma_{\text{Spikey}} = 97.3\%. \tag{42}$$

415 values were discarded this way, mainly values with synapse 7, 17, 25, 28, 43 or 49.

Although the cut-off seems to filter completely broken units reliably, there is no peak whose boundary corresponds with the bend in the threshold curve. Hence, the cut-off is based on more arbitrariness than in the previous cases. Nonetheless, we can say that the

weight variation is greater than 50%.

5 Discussion and outlook

In the theoretical part, formula (31) summarizes all results. Gibbs like sampling with asymmetric weights extends the special case with complete symmetry and leads to a stationary distribution which is continuous in all parameters. The system stays stable and converges towards its target behaviour for small weight variations.

As a next step, a nicer expression for (31), similar to the Boltzmann distribution (1), would be very useful. Probably, it would make analytic calculations for $D_{\mathrm{KL}}(p_{\mathrm{pertubated}}||p_{\mathrm{target}})$ possible and enable results as shown in figure 4 with more units and higher precision in far less time.

All computations and simulations coincide in the investigated phenomenon apart from their individual additional effects. Small deviations in the Gibbs sampling and big deviations in the LIF case from exact computations were expected and are explainable. The different network size apparently caused no significant difference, although for larger networks that relation might change.

The plots show that biases in the same order of magnitude as the weights make no important difference. In the LIF case they may do slightly, but it is hard to tell due to the large uncertainty.

Below a transition phase starting at a weight variation of $\sigma \approx 20\%$, the deviations from the target distribution caused by the the approximations in LIF sampling dominate the effects of pertubated weights.

Further investigations should start to separate all assumptions made in this thesis, as for example introduce correlations in the weight variations or impose a specific network structure.

The weight variability has improved significantly over the generations of neuromorphic hardware. On Spikey we have $\sigma = 97.3\%$ and even with an unreasonable amount of discarded units it stays above 50%. On the BrainScaleS System we have $\sigma = 16.4\%$ and on HICANN-DLS $\sigma = 4.91\%$.

Compared to the transition phase mentioned above, the weight variability on the BrainScaleS System and its successor, the HICANN-DLS chip, is not the most limiting factor, if outlier neurons and synapses are not used. On Spikey, sampling is not feasible. Note that these statements refer to networks where each random variable is represented by one neuron.

This thesis has focussed on the variations in the synaptic strength. There are more

important quantities that vary on the hardware and have complexer consequences. For example, variations in the time constants cannot be translated directly to variations in the abstract Boltzmann parameters. Instead, they have to be varied in the LIF simulations. For support from abstract results, one could use neuronal sampling [25].

References

[1] P. Dayan, L. Abbott, *et al.*, "Theoretical neuroscience: computational and mathematical modeling of neural systems," *Journal of Cognitive Neuroscience*, vol. 15, no. 1, pp. 154–155, 2003.

[2] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.

[3] L. Tierney, "Markov chains for exploring posterior distributions," *the Annals of Statistics*, pp. 1701–1728, 1994.

[4] M. A. Petrovici, *Form vs. Function: Theory and Models for Neuronal Substrates*. PhD thesis, Universität Heidelberg, 2015.

[5] P. Dayan, L. Abbott, and L. Abbott, "Theoretical neuroscience: computational and mathematical modeling of neural systems," *Philosophical Psychology*, pp. 563–577, 2001.

[6] E. Ising, "Beitrag zur theorie des ferromagnetismus," *Zeitschrift für Physik A Hadrons and Nuclei*, vol. 31, no. 1, pp. 253–258, 1925.

[7] R. Salakhutdinov and G. Hinton, "Deep boltzmann machines," in *Artificial Intelligence and Statistics*, pp. 448–455, 2009.

[8] F. G. Frobenius, "Über matrizen aus nicht negativen elementen," 1912.

[9] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[10] A. Kungl, "Sampling with leaky integrate-and-fire neurons on the hicannv4 neuromorphic chip," masterarbeit, Universität Heidelberg, 2016.

[11] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson, "Molecular biology of the cell. 3rd," *New York: Garland Pub*, vol. 43, no. 1294, p. 67, 1994.

[12] Neuron. (2006). [image] Available at: `https://commons.wikimedia.org/w/index.php?curid=1474927` [Accessed 8 Sep. 2017].

[13] Synapse Illustration2 tweaked. (2006). [image] Available at: `https://commons.wikimedia.org/w/index.php?curid=4001388` [Accessed 8 Sep. 2017].

[14] Action potential. (2007). [image] Available at: `https://commons.wikimedia.org/w/index.php?curid=2241513` [Accessed 23 Sep. 2017].

[15] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[16] M.-O. Gewaltig and M. Diesmann, "Nest (neural simulation tool)," *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.

[17] A. P. Davison, D. Brüderle, J. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "Pynn: a common interface for neuronal network simulators," *Frontiers in neuroinformatics*, vol. 2, 2008.

[18] T. Pfeil, A. Grübl, S. Jeltsch, E. Müller, P. Müller, M. A. Petrovici, M. Schmuker, D. Brüderle, J. Schemmel, and K. Meier, "Six networks on a universal neuromorphic computing substrate," *Frontiers in neuroscience*, vol. 7, 2013.

[19] J. Schemmel, J. Fieres, and K. Meier, "Wafer-scale integration of analog neural networks," in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pp. 431–438, IEEE, 2008.

[20] S. Friedmann, J. Schemmel, A. Grübl, A. Hartel, M. Hock, and K. Meier, "Demonstrating hybrid learning in a flexible neuromorphic hardware system," *IEEE transactions on biomedical circuits and systems*, vol. 11, no. 1, pp. 128–142, 2017.

[21] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," tech. rep., COLORADO UNIV AT BOULDER DEPT OF COMPUTER SCIENCE, 1986.

[22] P. B. Brockhoff, "The statistical power of replications in difference tests," *Food Quality and Preference*, vol. 14, no. 5, pp. 405–417, 2003.

[23] H. Bos, A. Morrison, A. Peyser, J. Hahne, M. Helias, S. Kunkel, T. Ippen, J. M. Eppler, M. Schmidt, A. Seeholzer, M. Djurfeldt, S. Diaz, J. Morén, R. Deepu, T. Stocco, M. Deger, F. Michler, and H. E. Plesser, "Nest 2.10.0," Dec. 2015.

[24] O. Breitwieser, "Towards a neuromorphic implementation of spike-based expectation maximization," Master's thesis, Universität Heidelberg, 2015.

[25] L. Buesing, J. Bill, B. Nessler, and W. Maass, "Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons," *PLoS computational biology*, vol. 7, no. 11, p. e1002211, 2011.

Appendices

A Simulations with NEST

A.1 Single neuron

Program parameter	Value	Description
i_offset	0.0	external current
tau_refrac	10.0	refractory time
tau_syn_E	10.0	excitatory synaptic time constant
tau_syn_I	10.0	inhibitory synaptic time constant
cm	0.1	membrane capacity
tau_m	0.02	membrane time constant
v_rest	-52.2	leak potential (for figure 3 a))
v_reset	-53.0	rest potential
e_rev_E	0.0	excitatory reversal potential
e_rev_I	-90.0	inhibitory reversal potential
v_thresh	-52.0	spiking threshold
rate	5000	Poisson input frequency
w_E	0.0035	excitatory synaptic weight
w_I	0.0055	inhibitory synaptic weight

A.2 Sampling network

Program parameter	Value	Description
cm	0.2	membrane capacity
tau_m	1.0	membrane time constant
e_rev_E	0.0	excitatory reversal potential
e_rev_I	-100.0	inhibitory reversal potential
v_thresh	-52.0	spiking threshold
v_thresh	-50.0	spiking threshold
tau_syn_E	10.0	excitatory synaptic time constant
v_rest	-50.0	leak potential
tau_syn_I	10.0	inhibitory synaptic time constant
v_reset	-50.001	rest potential
tau_refrac	10.0	refractory time
i_offset	0.0	external current
rate	3000	Poisson input frequency
w_E	0.001	excitatory synaptic weight
w_I	0.001	inhibitory synaptic weight

B Hardware measurements

B.1 Spikey

The Spikey station 503 was used.

Program parameter	Value	Description
v_reset	-80.0	rest potential
e_rev_I	-80.0	inhibitory reversal potential
v_rest	-75.0	leak potential
v_thresh	0.0	spiking threshold
g_leak	20.0	leakage conductance
weight	10	digital weight

B.2 BrainScaleS System

The HICANN 92 on the Wafer 21 of the BrainScaleS System was used.

Program parameter	Value	Description
cm	0.2	membrane capacity
v_reset	-70.0	rest potential
v_rest	-20.0	leak potential
v_thresh	50.0	spiking threshold
e_rev_I	-100.0	inhibitory reversal potential
e_rev_E	60.0	excitatory reversal potential
tau_m	20.	membrane time constant
tau_refrac	0.1	refractory time
tau_syn_E	5.0	excitatory synaptic time constant
tau_syn_I	5.0	inhibitory synaptic time constant
weight	10	digital weight

B.3 HICANN-DLS

The HICANN-DLSv2-chip 22 was used.

Program parameter	Value	Description
inh_synin_timeconstant	10E-6	inhibitory synaptic time constant
exc_synin_timeconstant	10E-6	excitatory synaptic time constant
membrane_timeconstant	0.5E-6	membrane time constant
leak_potential	0.3	leak potential
spike_threshold	0.8	spiking threshold
refractory_time	10E-6	refractory time
global_reset_potential	0.3	rest potential
weight	24	digital weight

Danksagung

Ohne Professor Meier und die Arbeitsgruppe hätte sich diese Chance, in ein interessantes Thema hinein zu schnuppern und über den Tellerrand der Physik zu schauen, nicht ergeben können.

Desweiteren möchte ich meinem Betreuer, Ákos Kungl, der mir mit Rat und Tat zur Seite stand und so die nötige Orientierung für dieses Projekt bot, danken.

Mit Fragen konnte ich mich ansonsten jederzeit an jeden wenden, insbesondere erwähnen möchte ich Andreas Baumbach, Oliver Breitwieser, Dominik Dold, Eric Müller und Mihai Petrovici.

Yannik Stradmann hat die Messungen auf HICANN-DLSv2 freundlicherweise durchgeführt und mir so den Vergleich mit neueren Chip-Generationen ermöglicht.

Torben Skrzypek steuerte eine andere Perspektive bei und half so bei der Darstellung.

Vielen Dank an alle für die Unterstützung und das angenehme Arbeitsklima.

Erklärung

Ich versichere, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Heidelberg, den 03.10.2017,