Department of Physics and Astronomy
University of Heidelberg

Bachelor Thesis in Physics
submitted by

David Stöckel

born in Waiblingen (Germany)

January 2015

Boltzmann Sampling with Neuromorphic Hardware

This Bachelor Thesis has been carried out by David Stöckel at the
Kirchhoff Institute in Heidelberg
under the supervision of
Prof. Karlheinz Meier

Abstract

The Boltzmann machine is an artificial neural network of stochastic binary units. It is a very general model which allows the implementation of efficient machine learning algorithms. These algorithms make the Boltzmann machine a powerful discriminative and generative model. The sampling steps of a Boltzmann machine can be calculated highly in parallel and are typically limited by the Von-Neumann bottleneck of conventional computer architectures.

To overcome this limitation, this thesis aims to implement a Boltzmann machine of three sampling units on the Spikey neuromorphic computing platform. This device emulates leaky integrate-and-fire neurons highly accelerated compared to neurons in the human brain. Every sampling unit is modeled as a thoroughly calibrated network of 61 neurons.

There are two networks implemented which closely approximate the Boltzmann distribution: one with positive weights and another with negative weights. Measurements of the probability distributions of the different states are consistent with the fitted Boltzmann distribution. Further high precision measurements with relative uncertainties less than 0.5% show systematic deviations from the Boltzmann distribution.

Zusammenfassung

Die Boltzmann-Maschine ist ein künstliches neuronales Netzwerk welches aus binären Zufallsvariablen besteht. Für dieses sehr allgemein formulierte Netzwerk sind effiziente Lernalgorithmen bekannt. So erzielt die Boltzmann-Maschine gute Ergebnisse sowohl als diskriminatives als auch als generatives Modell. Die Berechnungen sind stark parallelisierbar und auf konventionellen Systemarchitekturen typischerweise durch den Von-Neumann-Flaschenhals limitiert.

In dieser Arbeit wird versucht eine Boltzmann-Maschine auf dem Spikey Chip umzusetzen, um diese Parallelisierbarkeit auszunutzen. Dieser integrierte Schaltkreis emuliert „Integrate-and-Fire-Neuronen“, welche stark beschleunigt sind, verglichen mit Neuronen im menschlichen Gehirn. Jede Zufallsvariable der Boltzmann-Maschine wird von einem sorfältig kalibrierten Netzwerk aus 61 Neuronen repräsentiert.

Während dieser Arbeit wurden zwei Netzwerke erstellt, deren Zustandsverteilungen der Boltzmann-Maschine sehr nahe kommen: ein Netzwerk mit positiven, ein weiteres mit negativen Gewichten auf den Verbindungen. Die zugehörigen gemessenen Wahrscheinlichkeiten der Zustände stimmen mit den angepassten Boltzmann Verteilungen im Rahmen der erwarteten Fehler überein. Systematische Abweichungen zur Boltzmann Verteilung zeigen sich bei weiteren Präzisionsmessungen mit relativen Fehlern der Wahrscheinlichkeiten von unter 0.5%.

Contents

1. Introduction

The analysis of noisy or incomplete data as performed in the brain is a difficult task for computer algorithms. The Boltzmann machine – an artificial neural network introduced by Ackley et al. [1985] – is capable of learning important aspects of such a data set [Fischer and Igel, 2012]. The trained Boltzmann machine can be used for classification of unknown samples and furthermore as a generative model. It was shown that it outperforms other common machine learning algorithms, such as Support Vector Machines and Linear Discriminative Analysis models, for particular tasks [Srivastava and Salakhutdinov, 2014].

The learning and sampling process of the Boltzmann machine has a high computational cost [Fischer and Igel, 2012], however it can be processed highly in parallel [Ackley et al., 1985]. This thesis aims to implement a Boltzmann machine on the neuromorphic Spikey chip [Pfeil et al., 2013]. The sampling process emulated by this integrated circuit is not only parallel by design for all neurons, but is also performed at a speed-up factor of $\approx 10^4$ compared to biological neurons.

Additionally, this work is as test of the usability of the Spikey device and the PyNN programming interface. It is one the first experiment carried out on Spikey v5 that involves a large number of neurons and synapses. Methods were developed to efficiently correct the hardware variations. This project may therefore serve as a reference for future network emulation experiments with Spikey.

2. Emulation Platform and Computational Model

Section 2.1 introduces the Spikey chip and the implemented leaky integrate-and-fire (LIF) neuron model with conductance-based synapses. The Boltzmann machine is described in section 2.2 and its emulation with LIF neurons in section 2.3.

2.1. The Spikey Neuromorphic Computing Platform

The network emulations for this thesis were run on a Spikey v5 chip [Pfeil et al., 2013]. This is the latest generation of a neuromorphic chip with 384 analog neurons. In sections 2.1.1 and 2.1.2 the implemented neuron and synapse model are discussed. A short overview of the layout and signal paths on the chip is given in the following. The chip design is described in detail in [Schemmel et al., 2006, 2007, Indiveri et al., 2011].

Figure 1 shows a photo of the microchip with the signal path highlighted for one neuron. The chip is separated into a left and a right block of synapses and neurons each consisting of 192 neurons and 256 synapse line drivers. Therefore, every neuron can receive input from up to 256 different sources. The way these neurons are addressed with the PyNN [Davison et al., 2009] interface is shown in a sketch of the Spikey layout in appendix A. The neuron acquires its input with an excitatory and an inhibitory vertical input line that crosses the horizontal input lines. These horizontal input lines are fed by the synapse line drivers which convert a binary spike input signal into a voltage curve.

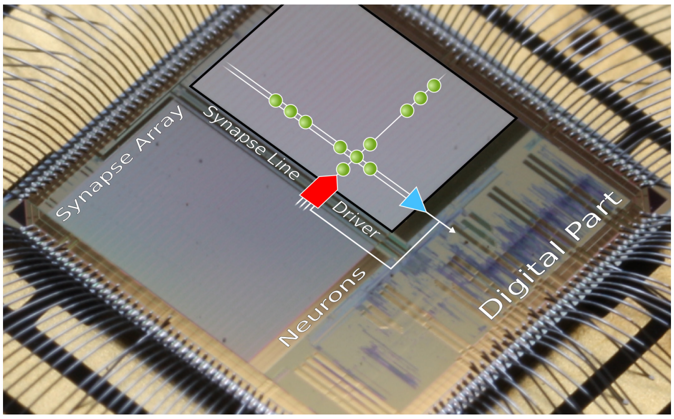

Figure 1: Photograph of the Spikey chip with a signal path for one neuron highlighted [Pfeil et al., 2013]. The neuron is indicated with a blue triangle. It receives its input from two input lines which cross the horizontal input lines. At the crossings which represent the synapses – colored in green – one can choose a 4 bit connection weight.

For each horizontal line the signal is either excitatory or inhibitory. The crossings of the horizontal lines with the neuron input lines represent synapses with a 4-bit connection weight each. The synapses convert the voltage curve into a current curve with the synaptic weight as a scale factor.

2.1.1. Implemention of the Leaky Integrate-and-Fire Neuron Model

The implemented neurons are designed to approximate conductance-based leaky integrate-and-fire (LIF) neurons. Such a neuron is modeled by a membrane capacitance C_{mem} that is charged or discharged by a current resulting from the leakage conductance g_{l} towards the rest potential E_{l}. The capacitor can also receive excitatory or inhibitory current input $I_{\mathrm{exc/inh}}$ through the synaptic conductances. If the membrane voltage U_{mem} reaches the threshold potential U_{thresh}, the neuron spikes and excites or inhibits neurons connected through the synaptic connections. At the same time the membrane voltage is clamped to the reset potential U_{reset} for a refractory time τ_{ref}. This temporal evolution of the membrane potential is defined by

$$-C_{\mathrm{mem}} \frac{dU_{\mathrm{mem}}}{dt} = g_{\mathrm{l}} \left(U_{\mathrm{mem}} - E_{\mathrm{l}} \right) + I_{\mathrm{exc}}(t) + I_{\mathrm{inh}}(t) \tag{1}$$

and the reset condition

$$U_{\mathrm{mem}}(t) = U_{\mathrm{reset}} \quad \text{for} \quad t \in \left(t_{\mathrm{spike}}, t_{\mathrm{spike}} + \tau_{\mathrm{ref}} \right) \quad \text{if} \quad U_{\mathrm{mem}}(t_{\mathrm{spike}}) = U_{\mathrm{thresh}} \,. \tag{2}$$

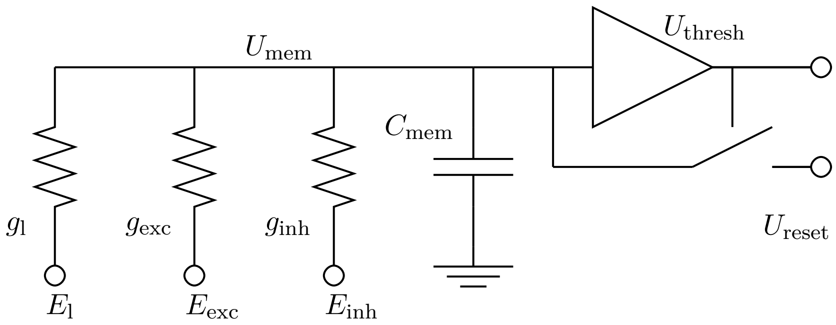

Figure 2: Circuit representation of the conductance-based LIF neuron model. If the membrane potential reaches the threshold potential it is clamped to the reset potential and a pulse is sent to the synaptic network.

The excitatory or inhibitory synaptic input (cf. section 2.1.2) from the i-th connected neuron is given by the conductance $g_i^{\text{exc/inh}}(t)$ and the reversal potentials $E_{\text{exc/inh}}$:

$$I_{\text{exc/inh}}(t) = \sum_i g_i^{\text{exc/inh}}(t) \left(U_{\text{mem}}(t) - E_{\text{exc/inh}} \right) . \tag{3}$$

Figure 2 shows a schematic of the circuit representation of the LIF neuron.

2.1.2. Synaptic Input and Spike Routing

The synaptic circuits on the device model conductance-based exponentially decaying synapses. Figure 3 shows a schematic of the synapse array and the spike routing. The excitatory and inhibitory conductances $g_{\text{exc/inh}}(t)$ are proportional to the control currents from the input lines. Since the input line sums up all input currents from the synapses, the resulting conductance is a linear sum over the feedback lines i that it crosses:

$$g_{\text{exc}}(t) = \text{const} \cdot \sum_i c_i I_i(t) \, . \tag{4}$$

The synaptic weight c_i is a programmable 4-bit value and the current curve $I_i(t)$ is generated by the i-th synapse from the voltage curve of the i-th synapse line driver. This equation holds for the inhibitory input as well. At each incoming spike a synapse line driver generates a voltage curve. Its shape can be configured for each driver in its rise time, fall time and amplitude. Additionally, all line drivers share one parameter that sets the voltage offset from which the voltage rises within the rise time. In [Petkov, 2012, p. 79] a sketch of the voltage curve and the resulting current curve generated by the synapses can be found.

The routing of the spikes to the synapse line drivers underlies some constraints of the hardware routing architecture. Each block has 256 synapse line drivers of which each can take input from one of four different spike sources. The first 192 synapse line drivers can be connected to:

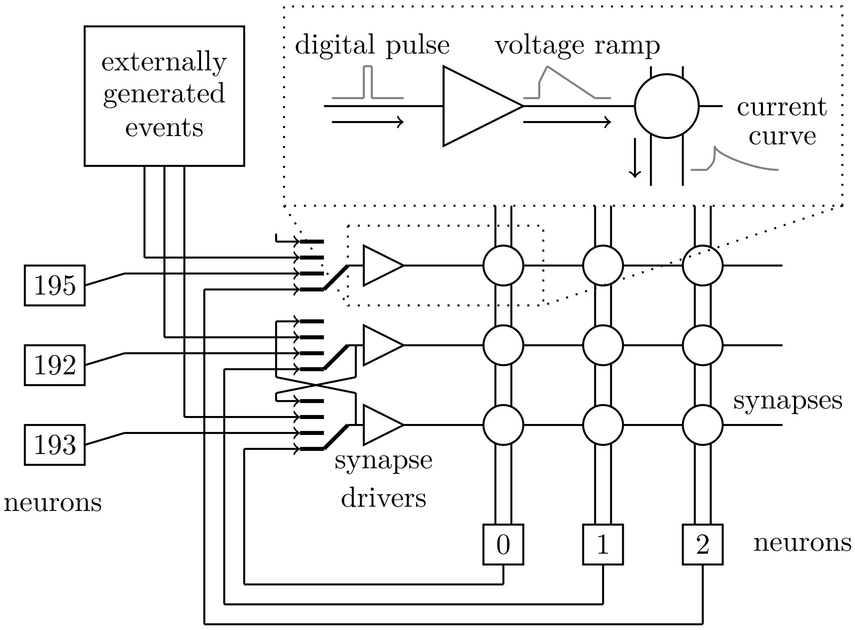

Figure 3: Schematic of the synapse array and the generation of the current curve to generate the excitatory and inhibitory conductance (cf. [Petkov, 2012, Figure 4.7]).

- the neuron on the same block with the same index.
- the neuron on the adjacent block but with even and odd indices permuted, particularly neuron 0 connects to driver 1, neuron 1 connects to driver 0, etc..
- the same input as the neighboring synapse line driver.
- externally generated events.

Those input switches are illustrated in Figure 3 as well. The synapse line drivers 192 to 255 cannot get input from neurons of the same or adjacent block. However, there is the additional option to receive an input signal from one of the pins as spike input as well. The routing options for the neuron feedback to the synapse line drivers are summarized in appendix B.

2.2. The Boltzmann Machine

A Boltzmann machine is an artificial neural network consisting of stochastic units which are symmetrically connected. The state $z_k \in \{0, 1\}$ of every unit k is a binary random variable, where $z_k = 1$ is called *on-state*. The probability to be in the on-state is determined by the neuron's bias b_k, the connection weight w_{kj} and the state of the connected neurons z_j:

$$p\left(z_k = 1 | z_j, j \neq k\right) = \frac{1}{1 + \exp\left[-\left(b_k + \sum_{j \neq k} w_{kj} z_j\right)\right]} \,. \tag{5}$$

For such a network the probability distribution of the states will converge towards the Boltzmann distribution as a stationary distribution [Hinton, 2007]. Then, the probability of a state $\vec{z} = (z_i)_{i=1}^{N}$ in a network of N sampling units is given by the Boltzmann distribution

$$p(\vec{z}\,) = \frac{1}{Z} \exp\left[-E\left(\vec{z}\,\right)\right] \tag{6}$$

with the energy function

$$E(\vec{z}\,) = -\frac{1}{2} \sum_{i \neq j} w_{ij} z_i z_j - \sum_i b_i z_i \,. \tag{7}$$

The partition function

$$Z = \sum_{\vec{z}} \exp\left[-E\left(\vec{z}\,\right)\right] \tag{8}$$

ensures the correct normalization.

This neural network can be modeled with spiking neurons [Buesing et al., 2011]. The state $z_k(t)$ is given by the spike times of neuron k. It is switched to one when the neuron fires and switched back to zero after the refractory period τ:

$$z_k(t) = \begin{cases} 1, & \text{if neuron } k \text{ fired in } (t - \tau, t] \\ 0, & \text{otherwise} \end{cases} \,. \tag{9}$$

Within the on-period the neuron cannot spike again. This is ensured by the instantaneous firing rate defined by

$$r_k(t) = \lim_{\Delta t \to 0} \frac{p(\text{spike in } [t, t + \Delta t))}{\Delta t} = \begin{cases} \frac{\exp(v_k(t))}{\tau}, & \text{if } z_k(t) = 0 \\ 0, & \text{if } z_k(t) = 1 \end{cases} \,. \tag{10}$$

The variable $v_k(t)$ is called the abstract membrane potential and is calculated to be

$$v_k(t) = b_k + \sum_{j \neq k} w_{kj} z_j(t) \,, \tag{11}$$

given the previous definitions (9), (10) and the target distribution (6). The stochastic firing behavior (10) of a neuron is therefore controlled by its membrane potential. It is the sum of a constant offset b_k and the rectangular postsynaptic potentials $w_{kj} z_j(t)$. The probability

$$p(z_k = 1 | v_k) = \frac{1}{1 + \exp\left[-v_k\right]} \tag{12}$$

to be in the on-state, given the membrane potential, is called the activation function. It is proportional to the mean firing rate ν_k of the neuron.

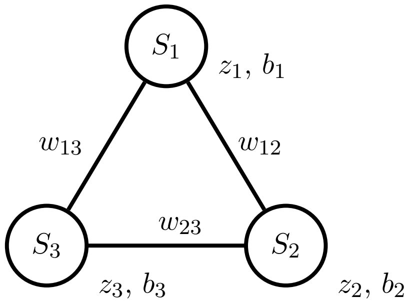

Figure 4: Sketch of the implemented Boltzmann machine with sampling units S, states z, biases b and weights w.

2.3. Boltzmann Sampling with Deterministic Spiking Neurons

Petrovici et al. [2013] showed that deterministic leaky integrate-and-fire neurons in a noisy spiking environment can sample from Boltzmann distributions. This holds for the high-conductance-state (HCS) approximation that assumes that a neuron's total conductance

$$g_{\text{tot}}(t) = g_{\text{l}} + g^{\text{inh}}(t) + g^{\text{exc}}(t) \tag{13}$$

is large compared to the leakage conductance g_{l}. In the high-conductance-state the membrane voltage follows the effective leak potential

$$u_{\text{eff}} = \frac{g_{\text{l}}E_{\text{l}} + g^{\text{inh}}(t)E_{\text{inh}} + g^{\text{exc}}(t)E_{\text{exc}}}{g_{\text{tot}}(t)} \tag{14}$$

nearly instantaneously since the effective time constant of the membrane dynamics,

$$\tau_{\text{eff}} = \frac{C_{\text{mem}}}{g_{\text{tot}}(t)} \tag{15}$$

goes to zero. With this approximation the activation function (12) can closely be reproduced [Petrovici et al., 2013] if the abstract membrane potential v_k is identified with

$$v_k = \frac{\bar{u}_k - \bar{u}_k^0}{\alpha}\,, \tag{16}$$

where $\bar{u}_k^0$ and α are constants and $\bar{u}_k$ denotes the average membrane potential. The constant $\bar{u}_k^0$ is equivalent to the zero-bias condition, i.e.: $p(z_k = 1|v_k = \bar{u}_k^0) = \frac{1}{2}$.

3. Methods

This thesis aims to implement a Boltzmann machine with three sampling units on the spikey chip. Figure 4 sketches this network with sampling units S_1 to S_3, states z_1 to z_3, biases b_1 to b_3 and weights w_{12}, w_{13} and w_{23}. The implementation of the sampling units is described in section 3.1, the implementation of the connections in section 3.2.

The calibration of those network components is explained in section 3.3. Section 3.4 covers the evaluation of the sampling statistics of the network and the comparison to the Boltzmann distribution.

All network emulations were implemented with PyNN. PyNN is a simulator-independent description language for building neural network models. In this thesis a back-end of PyNN was used that maps the neural network on the Spikey chip and records the spike trains. The *spikey_sampling* python module created during this thesis is thoroughly documented in [Stöckel, 2014].

3.1. Boltzmann Sampling Unit Implementation

The implementation of the abstract Boltzmann sampling unit with LIF neurons is described in two parts. Section 3.1.1 covers the implementation of the refractory mechanism while section 3.1.2 describes the implementation of the bias and the stochastic behavior.

3.1.1. Synfire Chain Refractory Mechanism

The configuration space of the Spikey chip does not provide precise refractory times that are much longer than the synaptic delay. Therefore, we cannot implement Boltzmann sampling units with single LIF neurons since the Boltzmann machine assumes an instantaneous interaction. To model the sampling units with LIF neurons this thesis therefore presents a mechanism to implement an effective refractory period with a synfire chain. At every spike of the sampling unit the synfire chain network is excited and the signal propagation time defines the refractory period. This prolongs the refractory period significantly in a controlled way. This network – called *sampling unit network* or simply *sampling unit* in the following – consists of one LIF neuron that defines the state of the abstract sampling unit and a synfire chain that implements the refractory mechanism. Figure 5 illustrates this network. The synfire chain is a feed-forward network of neurons. Every excitatory pool E_n excites the next excitatory pool E_{n+1} and the next inhibitory pool I_{n+1}. Therefore, an excitation of the first group starts a sequence of excitations of the populations along the chain. To enforce the same runtime of each propagating signal along the neuron populations, it is designed that every neuron in each population spikes exactly once for each signal. Otherwise, the spiking activity is likely to disperse along the synfire chain. This dispersion is inhibited by backward inhibition of the excitatory pools from population I_n to I_{n-1}. Feed-forward inhibition – inhibitory connections from I_n to E_n – as successfully demonstrated by Pfeil et al. [2013] proved to be less reliable and entailed a higher parameter tuning effort.

If the sampling neuron S spikes, it starts the synfire chain by exciting the two first populations of the chain which are E_1 and I_1. The spiking activity propagates along the synfire chain and therefore the sampling neuron is strongly inhibited due to the inhibitory connections towards it. The membrane potential is pulled towards the inhibitory reversal potential. While the synfire chain is active the sampling neuron is effectively deactivated since no other presynaptic connection is strong enough to compensate the strong inhibition. The duration of the signal propagation along the synfire

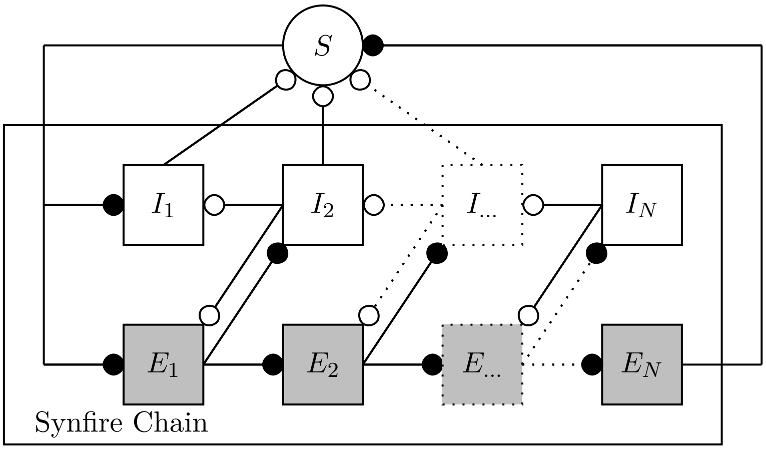

Figure 5: A network that models a Boltzmann sampling unit. The round circle represents a single LIF neuron, the rectangles populations of neurons. Grey shaded rectangles are excitatory neuron pools, the white rectangles are inhibitory neurons. The connection between the neurons are indicated by lines where the circle indicates the target. These connections between the neuron populations are all-to-all connections. A black circle represents an excitatory connection while a white circle represents an inhibitory connection.

chain is therefore interpreted as a refractory period. At the end of the synfire chain the last excitatory population excites the sampling neuron. This lifts the membrane potential towards the neuron's mean free potential quickly. Figure 6 shows the spike activity within a sampling unit network for one excitation of the sampling neuron. The membrane potential of the sampling neuron S is shown as well.

3.1.2. Implementation of the Stochasticity of the Sampling Neuron

The close reproduction of the desired activation function (12) shown by Petrovici et al. [2013] holds for a spiking noisy environment. The sampling neuron of the sampling unit network is therefore connected to an excitatory and inhibitory input of Poisson-distributed spikes. To fulfill the HCS assumption, the stimulus has to be at a high rate. As depicted in Figure 7, the sampling neuron receives also input from another external spike source called the bias stimulus. It fires regularly at a high rate. Its purpose is to generate a constant synaptic input current and therefore shift the mean free membrane potential $\bar{u}$ to a higher or lower value according to the connection weights. We can therefore vary the bias b_k of the k-th sampling unit according to

$$b_k = \frac{\bar{u}_k - u_k^0}{\alpha} \,. \tag{17}$$

Equation (17) results from the activation function in terms of the LIF domain (cf. equation (16)). The constants u_k^0 and α are fitted to the measured activation function. The

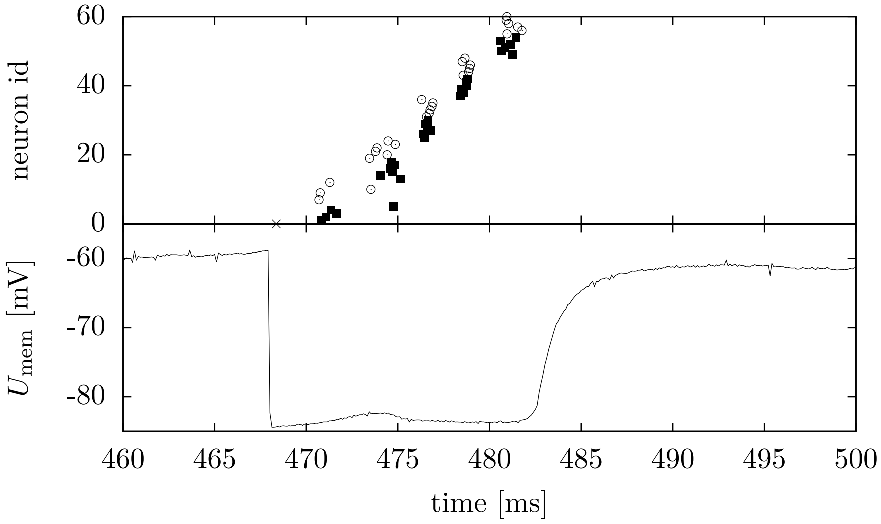

Figure 6: The upper plot shows the activity of the neurons of one sampling unit network after an excitation of the sampling neuron. The cross at neuron index 0 is the spike time of the sampling neuron S. The spike times of neurons of the chain are indicated by filled squares (excitatory) and circles (inhibitory). In the lower plot the membrane voltage of the sampling neuron is shown.

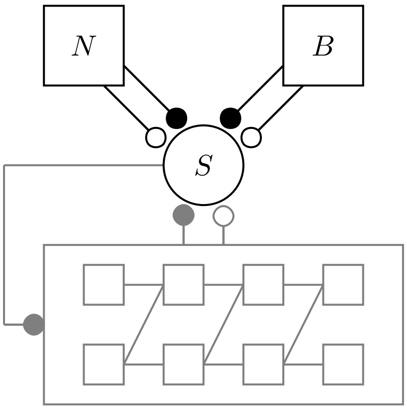

Figure 7: Schematic of the sampling unit network with the noise stimulus N and bias stimulus B from the external event generator. Each of the two populations consist of an excitatory and an inhibitory pool. The Poisson-distributed spikes of the noise population are sampled independently for the excitatory and the inhibitory pool. The bias stimuli are regularly distributed spikes at high rates.

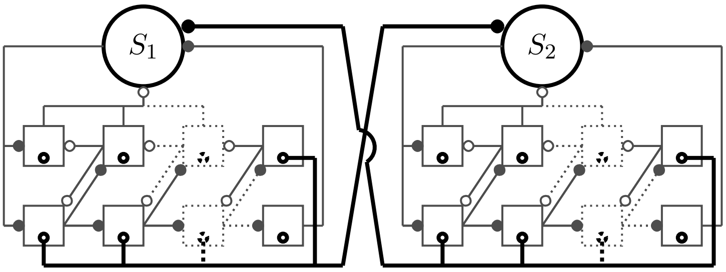

Figure 8: Schematic of the realization of a positive connection weight between two sampling unit networks. The synaptic connections that implement the Boltzmann connection are drawn with bold black lines. The small black circles within the excitatory and inhibitory groups indicate single neurons within a population.

scale parameter α corresponds to the dynamic range of the membrane potential and therefore the strength of the noise input. Parameter u_k^0 refers to the difference between the mean free membrane potential and the threshold potential.

3.2. The Boltzmann Connections

The Boltzmann connections between the random variables are implemented with many synaptic connections between the synfire chains and the sampling neurons. Figure 8 shows a schematic of the connection between two sampling unit networks with a positive weight. It is implemented with excitatory connections from one neuron of each synfire chain population to the target sampling neuron. Additionally the connection from the neuron of the last synfire chain population to the sampling neuron is inhibitory. Its purpose is to shift the membrane potential of the target neuron back to its mean free potential. This process approximates a rectangular postsynaptic potential. A connection with a negative Boltzmann weight is implemented by connecting the selected neurons from the inhibitory pools $I_1, \ldots, I_{N-1}$ to the other sampling neuron. The selected neuron from the last excitatory population E_N is connected excitatorily to the other sampling neuron.

There are several advantages in constructing the synaptic connections between the synfire chain and the sampling neuron and not simply between the two sampling neurons.

- The neuron-neuron connection cannot be inhibitory since the sampling neurons are all defined to be excitatory, as they have to excite their first synfire chain population.

- The interaction time is in the same order as the refractory time by construction: one synfire chain implements the refractory period as well as the interaction between the sampling units.

- The curve of the post synaptic potential (PSP) can be varied to approximate a rectangular PSP (cf. section 2.2).

- The dynamic range of the connection strengths between the sampling neurons can be changed over several orders of magnitude. This is explained in the following paragraph.

The dynamic range of a single neuron-neuron connection is limited to 16 different values. The proposed implementation of the connection involves as many synaptic connections as the length of the synfire chain. Each Boltzmann connection of the network presented in section 4 therefore consists of 5 synapses in each direction. Furthermore, the strength of the synapse line drivers can be varied (see section 2.1.2). This cannot be varied for the sampling neuron since the strength has to be set to the maximum, otherwise it would not be strong enough to start the synfire chain.

3.3. Calibration of the Network Components

The analog circuits of the Spikey chip are subject to variations due to the manufacturing process. All measurements were carried out with the uncalibrated Spikey with chip number 504. Especially handling the variations of the strength of the synaptic connections is crucial. The same connection strength may excite one neuron to spike twice but may not be sufficient to excite another neuron to spike. Not only the synapses do vary in their connection strength but also the synapse line drivers in their pulse shape and the neurons in the input gain factor and resting potential. These variations influence the effective connection strength as well.

The weight matrix, which is the result of a calibration, is strongly dependent on the neural circuits on the hardware the network is mapped to. Therefore all parts of the Boltzmann machine are calibrated within the complete network and not as an isolated part.

3.3.1. Calibration of the Synfire Chain

The synfire chain is calibrated to ensure that every neuron spikes ideally once on every activation of the chain. Figure 9 shows a flowchart of the calibration process. After an emulation of the network the spike trains of the neurons in the synfire chain are evaluated. At first the algorithm determines the times at which the chain should have started based on the spike times of the initial excitatory synfire chain pool. One cannot use the spike times of the sampling neuron S due to ghost spikes[1]. A correct estimate of count of synfire chain cycles is required for a reliable calibration. The spike times of the initially activated excitatory group have to fulfill two conditions for triggering that the chain should have started at time T:

- There are N_{thresh} different neurons that spiked in $[T, T + t_{\text{window}})$
- The first condition was never fulfilled within $[T - t_{\text{silent}}, T)$

[1] Spike events that are accidentally recorded although they never happened on the membrane [Brüderle, 2009, p. 144]. This only happens at very low spike rates all over a block of 64 neurons. This regime of an activity next to zero is reached for example when measuring the lower part of the activation function (cf. section 4.4).

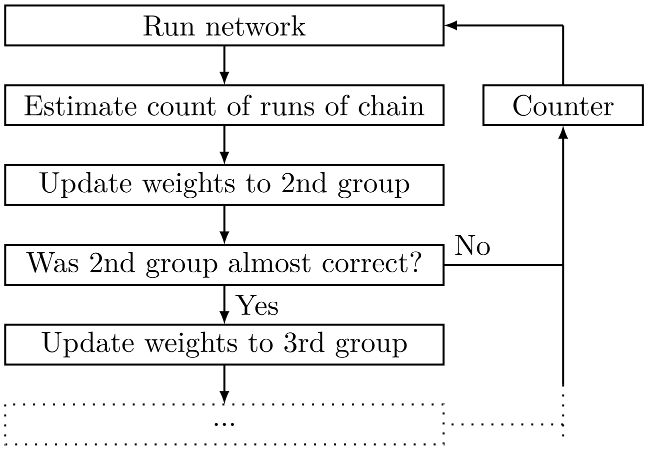
Run network
Estimate count of runs of chain
Counter
Update weights to 2nd group
Was 2nd group almost correct?
No
Yes
Update weights to 3rd group
...

Figure 9: Flowchart of the calibration process. The spike times of the different chains in the network are evaluated in parallel for each emulation.

The choice of the parameters N_{thresh}, t_{window} and t_{silent} is explained in section 4.2.

Once given the starting times of the chain, the spike times of the next excitatory and inhibitory group are evaluated. The spike count n of every neuron is compared to the desired count n_{target} which is the number of activations of the synfire chain. This difference determines an update factor f that was chosen to be

$$f\left(n, n_{\text{target}}\right) = \frac{2f_{\max}}{\pi} \arctan\left(\frac{\pi}{2r}\frac{n - n_{\text{target}}}{n_{\text{target}}}\right) . \tag{18}$$

The update factor is a smooth function with a positive derivative that saturates for large positive and large negative rate differences at $\mp f_{\max}$. A linear shape was tested as well. However, the limitation of the factor f increases the robustness. The parameter $f_{\max}$ determines the largest relative update of the weights. To prevent a change of the sign of the weight $f_{\max}$ should be less than 1. The factor r sets the slope of the curve and therefore the strength of the correction. A larger scale factor r results in a weaker correction. Its curve is sketched in Figure 10. The new weights of the connections to the neuron that should have fired n_{target} times but fired n times is set to

$$w_{\text{new}} = w_{\text{old}}\left[1 + f\left(n, n_{\text{target}}\right)\right] . \tag{19}$$

An update of the weights with $\Delta w = \text{const} \cdot f$ was tested as well but was found to be not as stable as the update equation (19). This calculated weight w_{new} has to be discretized to a 4-bit weight for each synapse[2]. The updated connection is an n to 1 connection. It is updated such that the first n' synapses get the weight $\lfloor w_{\text{new}} \rfloor$ and the remaining

[2] The PyNN implementation uses stochastic rounding to yield discrete weights for the connections when mapping it to the hardware. If the network configuration is written to the chip multiple times there are variations in synaptic weights due to stochastic rounding. An alternative solution would be to configure the seed of the random number generator.

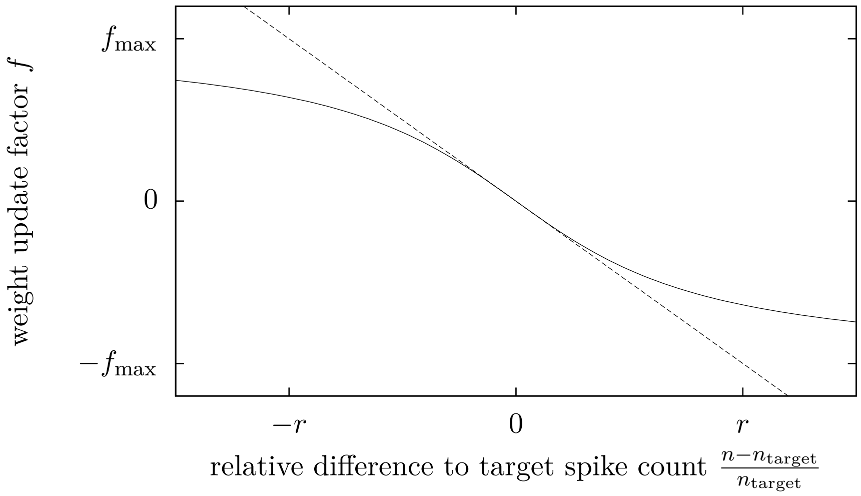

Figure 10: The curve of the update factor to correct for too much or too few spiking of a neuron within the chain. The horizontal axis shows the relative difference of the measured spiking rate of one neuron to the desired target spiking rate. The weight update factor for a given difference is shown on the vertical axis.

$n - n'$ are set to $\lceil w_{\text{new}} \rceil$. The functions $\lfloor x \rfloor$ and $\lceil x \rceil$ denote the floor and ceiling function, respectively. The integer value n' is chosen such that the difference of the mean of the weight array to the desired weight is minimized.

There is a stop condition in the algorithm to abort the weight update for the next populations. After the update of the weights towards a synfire chain group, the calibration routine checks if the activity of this group was close enough to the expected one. This means that the relative variation of the total count of spikes in the group compared to the desired count of spikes is less than a specific value, e.g. 0.05. If this variation is smaller than the threshold, the next group of neurons is updated in the same way.

The calibration algorithm is executed for a fixed number of network emulations. This is done for simplification and modularization of the program code. The synfire chain network module with its calibration routine is separated from the network emulation (cf. [Stöckel, 2014]).

3.3.2. Calibration of the Compensation Strength

The compensation group is the last excitatory group in the sampling network's synfire chain. This group excites the sampling neuron to lift its membrane potential up to its mean free potential. The rise time has to be short compared to the refractory time to fulfill the conditions for the proposed theory of Boltzmann sampling with LIF-neurons (cf. section 2.3). On the other hand, the compensation must not be too strong to prevent an overshoot of the membrane potential.

The compensation strength is calibrated manually by evaluating the autocorrelation function of the spike train of the sampling neuron. The spike train of a sampling neuron is defined by its spike times t_i:

$$\hat{s}_{\vec{t}}(t) = \sum_{i=1}^{N} \delta(t - t_i)\,. \tag{20}$$

Before the calculation of the autocorrelation this spike train is convolved with a Gaussian kernel with width σ_t:

$$s_{\vec{t}}(t) = \sum_{i=1}^{N} \exp\left[-\frac{1}{2}\left(\frac{t - t_i}{\sigma_t}\right)^2\right]. \tag{21}$$

In the following, the index $\vec{t}$ is omitted due to readability. The unnormalized autocorrelation of the spike train function is calculated as

$$\rho_{s,s}(\Delta t) = (s \star s)(\Delta t) = \int_{-\infty}^{\infty} s^{\star}(t')s(t' + \Delta t)\,\mathrm{d}t'\,. \tag{22}$$

It is a measure of the similarity of a signal to itself with a time lag Δt. Since the spike train signal is real-valued, the complex conjugation $s^{\star}$ is omitted in the following. To analyze finite-lasting spike trains with small computational effort, the autocorrelation was calculated in discrete time. The signal $s(t)$ with duration T is processed as an array s_n with

$$s_n = s\left(n \cdot \frac{T}{N_T}\right), \tag{23}$$

where the count of measured points is N_T. This array is cross-correlated with itself as commonly used in signal processing: the mean of the signal $\langle s \rangle$ is subtracted and the result is normalized to the variance of the signal σ_s^2:

$$\rho_{s,s}(\Delta n) = \frac{1}{N_T} \frac{\sum_{i=1}^{N_T} \left[(s_i - \langle s \rangle)(s_{i+\Delta n} - \langle s \rangle)\right]}{\sigma_s^2}\,. \tag{24}$$

At the boundaries where the sum accesses invalid indices the addend is set to zero. The shift Δn corresponds to a correlation time $\Delta t = T\Delta n / N_T$.

The autocorrelation function depends on the width σ_t of the Gaussian function and the width of the time bins. To prevent overlapping of the Gaussians of two subsequent spikes, the width has to be chosen shorter than the shortest expected interval between two spikes.

The autocorrelation is suitable for calibrating the compensation strength since it shows enhanced or suppressed spiking after a refractory period. An enhanced spiking probability after the refractory period τ compared to the average spiking probability is identified with an autocorrelation larger than zero at $|\Delta t| > \tau$, a suppression as an autocorrelation less than zero.

The calibration using the autocorrelation is cross-checked by looking at the spike-triggered average (STA) of the membrane potential. This is the average of the membrane potentials where the time dependent potential curves are aligned with the corresponding spike times. After the refractory period the membrane potential is expected to rise quickly towards the mean free potential without an overshoot.

3.3.3. Configuration of the Bias

Each sampling network has a bias that is related to the mean spike frequency in a network with all weights set to zero. For each sampling network a target spike frequency ν_{target} can be set. After the calibration of the synfire chain the bias is calibrated automatically as well. The network is run 16 times where the synaptic connection strength of the excitatory and inhibitory bias stimulus is swept over. It is scanned such that the total conductance of the synapses is constant. The synapse settings that achieved a firing rate of the sampling neuron that comes the closest to the target rate is chosen for the experiments.

3.3.4. Calibration of the Connections

The connections are tuned manually with the help of the cross-correlation of the spike trains of the two sampling neurons. The cross-correlation of two real valued signals $f(t)$ and $g(t)$ is given by

$$\rho_{f,g}(\Delta t) = (f \star g)(\Delta t) = \int_{-\infty}^{\infty} f(t')g(t' + \Delta t)\,\mathrm{d}t' \,. \tag{25}$$

The two spike trains $\hat{s}_i(t)$ and $\hat{s}_j(t)$, which are the sum of δ-functions as in equation (20), are convolved with a Gaussian kernel with width σ_t. The cross-correlation of those two signals is approximately proportional to the probability that the sampling node j is in the on-state at Δt given that neuron i fired within $-\sigma_t < t < \sigma_t$. As for the autocorrelation (cf. section 3.3.2) the signals are converted to a time discrete array $s_n^{(i)}$ with N_T sampling points covering the time frame $[0, T)$ linearily. The index n denotes the index within the array and i the index of the sampling neuron. The normalized time discrete cross-correlation with zero padding is given by

$$\rho_{i,j}(\Delta n) = \frac{1}{N_T} \frac{\sum_{n=1}^{N_T} \left[\left(s_n^i - \left\langle s^{(i)} \right\rangle \right) \left(s_{n+\Delta n}^j - \left\langle s^{(j)} \right\rangle \right) \right]}{\sigma_i \sigma_j} \,. \tag{26}$$

The sum is only evaluated for valid indices n that do not access invalid indices of the array in the second factor. The angular brackets indicate the mean value. The square roots of the variances of the arrays $s_n^{(i)}$ and $s_n^{(j)}$ are labeled σ_i and σ_j, respectively. As in section 3.3.2, the shift Δn corresponds to a time lag $\Delta t = T\Delta n / N_T$. Without subtracting the mean values of the arrays, the denominator is referred to as the unnormalized cross-correlation.

This cross-correlation of two spike trains is a measure for the spiking of neuron j induced by neuron i. Since it is symmetric under the exchange of the neuron indices

$$\rho_{i,j}(\Delta n) = \rho_{j,i}(-\Delta n) \,, \tag{27}$$

it is a measure for induced spiking in the backwards direction as well for negative time lags. Figure 11 shows a sketch of two neurons with a directed connection such that neuron 2 spikes after neuron 1 with a delay δt. The cross-correlation then shows a peak

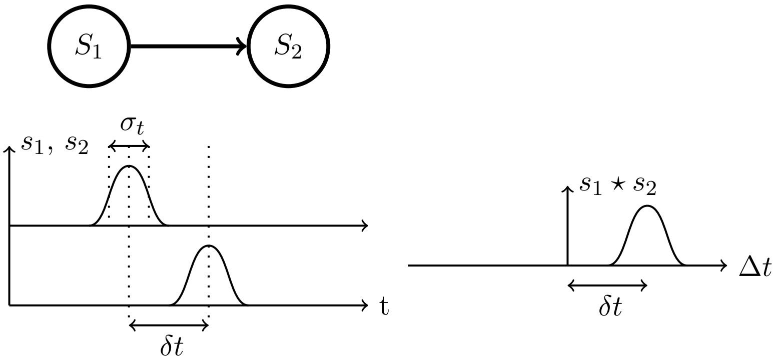

Figure 11: Neuron S_1 is connected to neuron S_2 such that the second neuron spikes after the first neuron with a delay δt. The Gaussian convolved spike trains are shown on the left, the cross-correlation of the spike signals is sketched on the right.

at $\Delta t = \delta t$.

The connection weights between the sampling unit networks are tuned with the correlation function of the two sampling neuron spike trains. For excitatory connections the weights are tuned such that the peaks at the positive correlation time and at the negative correlation time are of the same height and width. This corresponds to a symmetric connection in forward and backward direction, in particular with the same strength and with the same interaction time.

3.4. Evaluation of the Sampling Property

The network of 3 sampling units can be in one of $2^3 = 8$ different states at time t:

$$\vec{z}^{(t)} \in Z = \left\{ \begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}, \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}0\\1\\1\end{pmatrix}, \begin{pmatrix}1\\0\\1\end{pmatrix}, \begin{pmatrix}1\\1\\0\end{pmatrix}, \begin{pmatrix}1\\1\\1\end{pmatrix} \right\}. \tag{28}$$

The state vector $\vec{z}^{(t)}$ is determined by the spike times of the 3 sampling neurons with

$$z_i^{(t)} = 1 \quad \Leftrightarrow \quad \text{sampling neuron } i \text{ spiked in } (t-\tau, t]\,. \tag{29}$$

The refractory time τ is given by the duration of the synfire chain run. An emulation of the network with duration T yields an estimate

$$p_{\text{single}}(\vec{z}\,) = \frac{t_{\vec{z}}}{T} \tag{30}$$

for the probability distribution of the states. Time $t_{\vec{z}}$ is the total duration of state $\vec{z}$ being active during the network emulation. Therefore the probability estimates are properly normalized.

For N runs of the network the best estimate for the probability $p(\vec{z})$ is given by the mean value over $p_{\text{single}}(\vec{z})$ of each run:

$$p_{\text{est}}(\vec{z}) = \left\langle p^i_{\text{single}}(\vec{z}) \right\rangle = \frac{1}{N} \sum_{i=1}^{N} p^i_{\text{single}}(\vec{z}) \,. \tag{31}$$

It is easily shown that the mean values of the state's probabilities still form a correctly normalized probability distribution. The expectation value of the error is given by:

$$\Delta p_{\text{est}}(\vec{z}) = \frac{\sigma_{\vec{z}}}{\sqrt{N}} \,. \tag{32}$$

The enumerator $\sigma_{\vec{z}}$ denotes the standard deviation of the probability estimates for state $\vec{z}$ over the N single runs.

The Boltzmann distribution (6) is fitted to the probabilities by minimizing the sum of squared normalized residuals

$$\chi^2 = \sum_{\vec{z} \in Z} \left[\left(\frac{p_{\text{est}}(\vec{z}) - p(\vec{z})}{\Delta p_{\text{est}}(\vec{z})} \right)^2 \right] \tag{33}$$

using the Levenberg-Marquardt algorithm implemented in numpy [Walt et al., 2011]. This fit leaves only one degree of freedom. There are 8 different states and therefore 8 different equations for fitting the 6 free parameters of the Boltzmann distribution. Additionally, the condition of the normalization of the probability distribution removes a further degree of freedom. The free parameters of the Boltzmann distribution are the biases b_1 to b_3 and the symmetric weights w_{12}, w_{13}, w_{23}.

With only one degree of freedom left, the fit results have to be assessed with care. This is done by minimizing the uncertainty of the estimated probabilities with multiple runs and comparing the resulting χ^2 to its expected distribution. The smaller the expected error of the estimated probability becomes, the smaller the absolute residuals should become as well. The χ^2-distribution $f_n(\chi^2)$ for n degrees of freedom is the probability density function of the sum of the squared normalized residuals of n independent stochastically distributed variables. Figure 12 shows the probability density function $f_1(\chi^2)$ for an experiment with 1 degree of freedom. The mean χ^2 value of f_n is n and the variance is $2n$. If the fit yields a large sum of squared residuals where the probability density function is next to zero, it has to be considered unlikely that the fitted function models the measured data appropriately.

4. Experiments and Results

The basic configuration for all involved neurons is described in section 4.1. For a specific neuron configuration the synaptic weights of the synfire chain can be tuned. This is described in section 4.2. Section 4.3 covers the calibration of the compensation strength which completes the process of building a sampling unit network. The measurement of the activation functions of the sampling unit networks is presented in section 4.4.

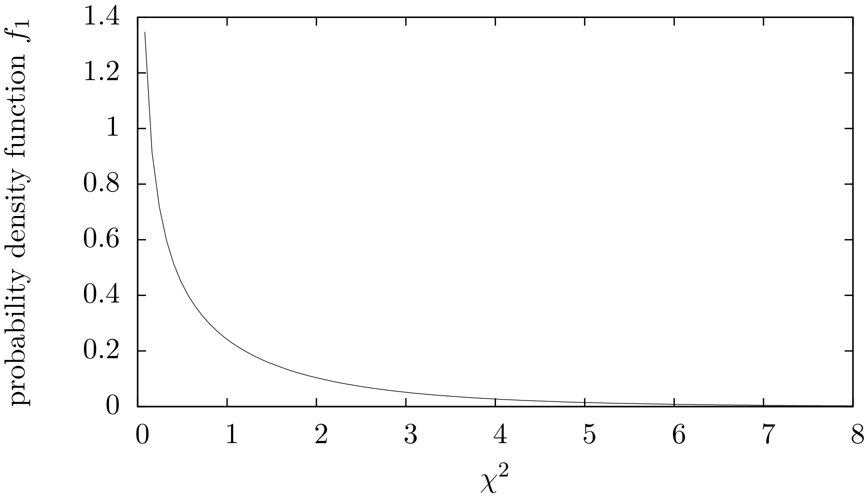

Figure 12: Distribution of χ^2 for one degree of freedom. The horizontal axis shows the sum of the squared residuals, the vertical axis denotes the probability density that the fit with one degree of freedom yields the corresponding χ^2-value. The probability density diverges to infinity for $\chi^2 \to 0$.

The tuning of the connections between the sampling units is described in section 4.5. Section 4.6 covers the measurement of the probabilities and the comparison to the Boltzmann distribution.

All experiments were emulated on the same Spikey system with chip number 504. The network as it is presented in the methods section 3 cannot be extended to both neuron blocks of the spikey chip due to the spike feedback routing constraints in Table 4. A neuron that is connected to neurons of both blocks disables another neuron effectively since this neuron cannot be connected to any neuron. The current network topology does not allow a separation such that the neurons give either feedback to the same or the adjacent block but not both blocks at once.

Since the emulation results vary due to various noise sources, all measurements were averaged over multiple executions of the experiment. Although the results of the single experiments vary in the recorded spike times and rates, the averaged measurements are reproducible. Over the course of one month no drifts were observed.

4.1. Configuration of the Spikey Neurons

The parameters of LIF neurons on the chip are configurable within a certain range. These configurable parameters are the resting potential U_{rest}, the threshold potential U_{thresh}, the reset potential U_{reset}, the inhibitory reversal potential E_{inh} and the leakage conductance g_{l}. The excitatory reversal potential is fixed to 0 V in the biological domain. There can be two types of neurons since all even and all odd neurons share one set of configuration variables, except for the leakage conductance that can be set individually for each neuron. To simplify the PyNN implementation of the network, one neuron configuration was applied to all neurons. This simplifies the mapping of the abstract PyNN neurons to the hardware neurons. On the other hand, this leads to almost contradicting requirements to the neuron parameters.

For the sampling neuron, the **resting potential** should be much lower than the excitatory reversal potential and much higher than the inhibitory reversal potential. In this case the excitations and inhibitions are expected to add up almost linearly. The excitatory and inhibitory neurons of the synfire chain can have an arbitrary resting potential.

The **threshold potential** is the most critical parameter due to hardware imperfections. For the first excitatory and inhibitory populations in the synfire chains it has to be close to the resting potential, otherwise one spike of the sampling neuron is not sufficient to excite these populations. However it should be far away from the resting potential for all other neurons whose excitabilities are not critical. A threshold potential that lies close to the reset potential causes many neurons to spike continuously due to variations in the manufacturing process of the analog circuits. These neurons have to be discarded.

Regarding the **inhibitory reversal potential**, all neurons require a value that is much lower than the resting potential. The sampling neuron needs to be inhibited very strongly, such that it cannot spike during the activity of the synfire chain. The synfire chain populations need to have a low inhibitory reversal potential as well. A low value prevents the neurons from spiking twice in one run of the chain.

The same argument as for the inhibitory reversal potential holds for the **reset potential**: A low value is suitable for the sampling neuron and for the neurons in the synfire chain to inhibit spiking twice.

The **leakage conductance** defines the time constant of the membrane potential. The synfire chain populations need to have a short time constant and therefore a large leakage conductance. This pulls the membrane potential of the synfire chain neurons quickly to the resting potential and allows multiple cycles of the synfire chain after another. The sampling neuron can have an arbitrary leakage conductance since it is expected to be in the high-conductance-state, such that the time constant is negligible for the membrane potential dynamics (cf. section 2.3). However, a high leakage conductance weakens the influence of the synaptic conductances. This worsens the problem of the choice of the threshold potential since the effective synaptic weights are lower.

The choice of neuron parameters is given in Table 1. These are the biological parameters as they are set with PyNN. They are translated to a configuration of the device that emulates this neuron model. The set was found by tuning the functional behavior of the sampling neuron and the synfire chain and balancing the different problems mentioned previously. With this configuration, six neurons had to be discarded due to continuous spiking.

Parameter	Symbol	Value
resting potential	U_{rest}	$-65.0\,\text{mV}$
threshold potential	U_{thresh}	$-62.5\,\text{mV}$
reset potential	U_{reset}	$-100.0\,\text{mV}$
inhibitory reversal potential	E_{inh}	$-100.0\,\text{mV}$
leakage conductance	g_{l}	$1.0\,\text{nS}$

Table 1: Configuration of the LIF neurons as set with PyNN.

Parameter	Description	Value
N_{thresh}	number of spikes that trigger a detection of a run	3
t_{window}	time frame for N_{thresh} spikes	4.0 ms
t_{silent}	time frame in which the chain must not run twice	10 ms
f_{max}	maximum of relative weight update factor	0.80
r	refers to how fast f_{max} is reached	2.2

Table 2: Parameters of the calibration algorithm of the synfire chain.

4.2. Synfire Chain Configuration and Calibration

The synfire chains which are connected to the sampling neuron consist of 5 excitatory and 5 inhibitory populations. Each population contains 6 neurons. Therefore, one sampling unit network consists of 61 neurons: one sampling neuron and 60 neurons in the synfire chain. The forward excitation is initialized with an arbitrary weight larger than zero since this one is set to an appropriate value by the calibration algorithm[3]. Each backward inhibition connection is given the synaptic weight 1 in units of the 4-bit weight between 0 and 15 on the device. The synapse line drivers are configured to yield the largest signal amplitude. The parameter *drvifall*, which is proportional to the inverse fall time of the synaptic current, is set to 1.5 times the default configuration. Therefore every neuron in the synfire chain excites or inhibits the connected neurons with a strong and short pulse.

The calibration algorithm of the synfire chain takes 5 different parameters. Table 2 lists the parameters and a short description of their purpose. The detailed description of the parameters is given in the methods section 3.3.1. The value N_{thresh} was chosen by comparison of different runs of the synfire chain. It also defines how many spikes of the first population have to be enough to excite the next population reliably, since the weights are changed according to the count of runs, that the trigger algorithm detects. The time window t_{window} is set to 4.0 ms, which is longer than the observed timescale of a PSP. Therefore, it covers all spikes that are caused by the same excitation of the sampling neuron. The time t_{silent} ensures that one excitation of the first group is never interpreted as two starts of the synfire chain. This time is expected to be longer than

[3] The synaptic weight of the synapses of the forward excitation is mostly 4 or 5 in dimensionless units of the hardware after calibration.

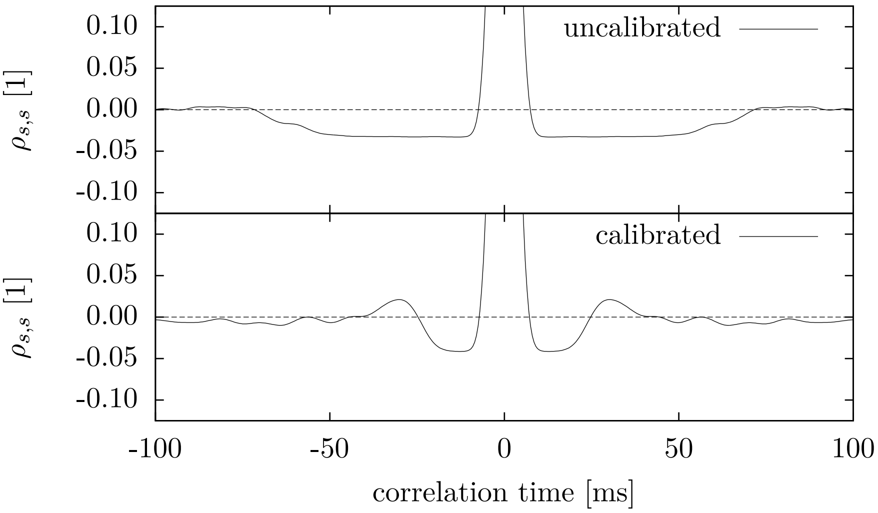

Figure 13: Autocorrelation for a sampling neuron. The upper plot shows the autocorrelation function without the tuned compensation strength, the lower plot is measured with this calibration. The autocorrelation is averaged over 50 runs with a duration of 5000 ms each.

the excitatory synaptic time constant and shorter than the refractory period.

The parameters $f_{\max}$ and r of the weight update factor are found by a manual search. The calibration of the synfire chains in the two Boltzmann networks is done with 50 iterations. Due to the break condition within the calibration process, the count of calibration iterations that are necessary rises quickly with the length of the synfire chain. While 10 iterations already yield a reliable synfire chain with a length of 5 populations, a chain with 15 populations needs approximate 200 iterations for the same quality.

4.3. Calibration of the Compensation Strength

The compensation strength is tuned manually by increasing it carefully until the correlation function indicates a slight overshoot after the refractory period. This was done for every neuron individually. It was tuned at the mean firing rate of the sampling unit. The Boltzmann connections to the other sampling units are set to zero. The mean firing rate is given by the strength of the bias connection. Figure 13 shows the measured autocorrelation for a calibrated sampling neuron in the lower plot. The overshoot was considered to be small enough to allow meaningful experiments. The upper plot shows the autocorrelation function of the spike train of a sampling unit without tuning the compensation strength. The suppressed spiking probability for correlation times between -50 ms and 50 ms shows a compensation strength that is too weak.

With the spike-triggered average of the membrane potential of the calibrated sampling neuron, the refractory period is determined. Figure 21 in the appendix shows the spike triggered average for one sampling neuron. The refractory period is determined to be

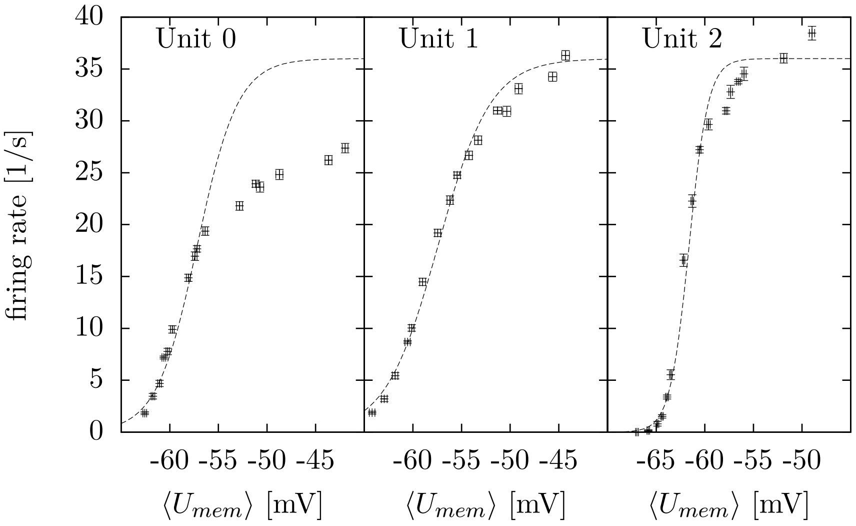

Figure 14: Measured activation function for the three sampling unit networks. The horizontal axis shows the mean free membrane potential. The fitted activation function is given in equation (34). The activation function of the first sampling unit is fitted only for voltages lower than $-55\,\mathrm{mV}$. The measured deviations at higher rates are considered to be systematic errors and are therefore discarded for the fit. An explanation is given in section 4.4. The second and third activation functions are fitted with all measured data points.

$(28 \pm 2)\,\mathrm{ms}$. This corresponds to a maximum firing rate of $(36 \pm 3)\,\mathrm{Hz}$.

4.4. Activation Function Measurements

The activation functions of the sampling unit networks are measured by increasing and decreasing the mean free membrane potential of the sampling neuron with the excitatory and inhibitory bias stimulus. The description of the network layout and the bias stimulus is given in section 3.1. Figure 14 shows the activation functions for the three calibrated sampling unit networks. The rate

$$\nu(\bar{u}) = \frac{\nu_{max}}{1 + \exp\left[-\frac{\bar{u} - \bar{u}^0}{\alpha}\right]}\,, \tag{34}$$

which is the activation function (12) with the mapping of the abstract membrane potential given in equation (16), is fitted to the emulation results. The mean free membrane potential $\langle U_{mem} \rangle$ of the sampling neuron is abbreviated with $\bar{u}$. The maximum firing rate ν_{max} is the inverse of the refractory period τ. There are two free parameters which are α and $\bar{u}^0$. Table 3 lists the parameters returned by the Levenberg-Marquard fit algorithm. The first sampling unit shows the largest deviations from the desired curve. The large deviations at high firing rates might be due to a too long regeneration time of the synfire chain. Figure 15 indicates that the inhibitory backward inhibition of the

Sampling Unit	**Slope α**	**Bias offset $\bar{u}^0$**
0	$(2.10 \pm 0.12)\,\mathrm{mV}$	$(-57.17 \pm 0.11)\,\mathrm{mV}$
1	$(2.71 \pm 0.11)\,\mathrm{mV}$	$(-57.40 \pm 0.14)\,\mathrm{mV}$
2	$(1.04 \pm 0.09)\,\mathrm{mV}$	$(-61.61 \pm 0.16)\,\mathrm{mV}$

Table 3: Fit parameters of the activation functions of the three sampling units. The fit for the first sampling neuron is restricted to membrane voltages lower than $-55\,\mathrm{mV}$ (cf. section 4.4).

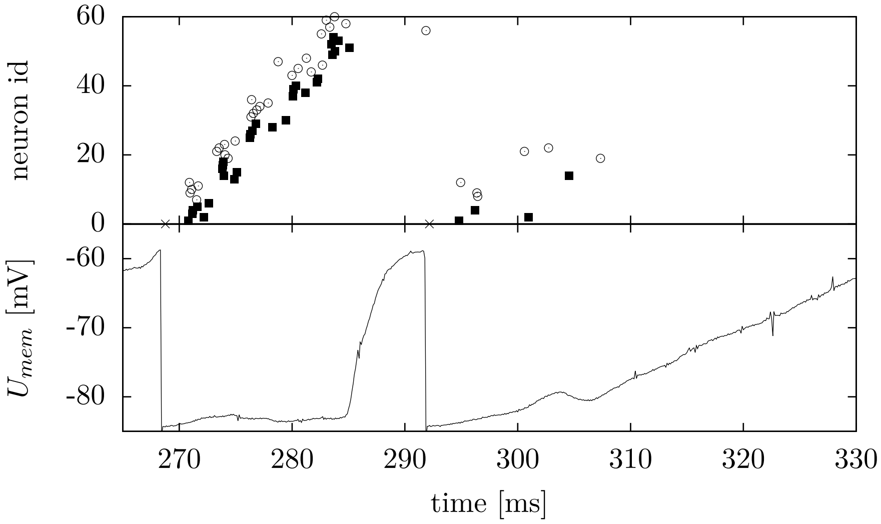

Figure 15: Early termination of the synfire chain of the first sampling unit network in its second run. The spikes of the sampling neuron and the synfire chain neurons are shown in the upper plot. The membrane potential of the sampling neuron is shown below.

previous run might be still too strong to allow another run. If the activity of the synfire chain stops before the excitation of the compensation group, the sampling neuron does not receive any compensation. After such an incomplete cycle, the sampling neuron is therefore inhibited much longer than the expected refractory period. This limits the maximum firing rate to a value lower than $1/\tau$. Although this occurs very often for the first sampling neuron in the network, this effect is barely seen for the other two sampling units. A better selection of neurons and an improved calibration of the synfire chain might solve this problem. To prove the feasibility of Boltzmann sampling with LIF neurons, only networks with firing rates lower than $\nu_{max}/2$ were tested in order to avoid further parameter tuning. At this rate, the synfire chain of the first sampling unit works reliably.

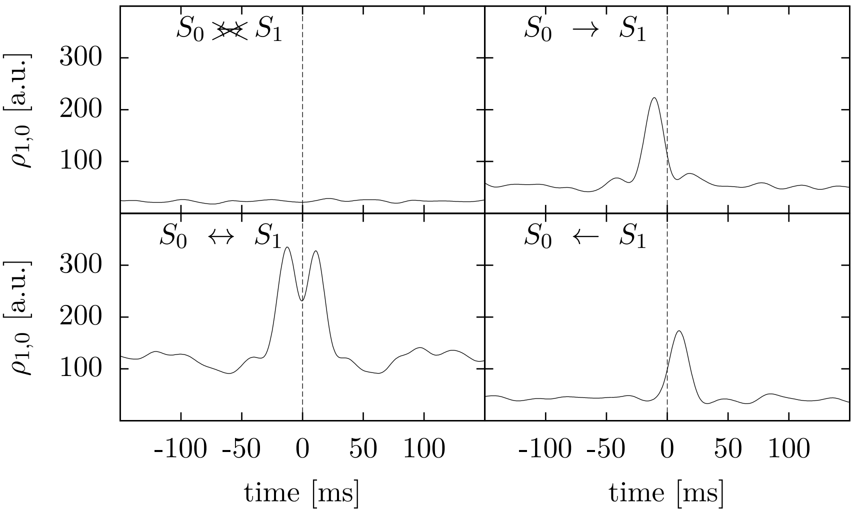

Figure 16: Cross-correlation between the spike trains of the first and the second sampling neuron. It is measured for four different configurations which are from the upper left in clockwise direction: a disabled connection, an enabled forward connection, an enabled backward connection and both directions enabled.

4.5. Calibration of the Connections

For testing the sampling properties of the entire network of 183 neurons, two types of networks were tested: a network with a high bias and inhibitory connections and a network with a low bias and excitatory connections.

The tuning of the symmetry of the connections is done with the help of the cross-correlation function as described in section 3.3.4. It was configured such that the connected sampling unit does not reach the saturation regime in its activation function after a spike of the presynaptic neuron. For a positive connection weight this refers to an excitatory connection that excites the connected neuron with a conditional probability less than 1. Figure 16 shows the unnormalized cross-correlation of a calibrated excitatory connection. The plot for the same connection before the calibration can be found in appendix D.

4.6. Measurements of the Sampling Properties

The sampling probabilities of the 8 different states the network can attain, are measured as described in section 3.4. The Boltzmann distribution is then fitted to these probabilities. The fit is done with one degree of freedom (cf. section 3.4). Therefore the sum of the squared residuals χ^2 is expected to be very small. Figure 12 shows the probability density function for χ^2 for one degree of freedom.

The probabilities are also averaged over many emulations of the network such that the expected error of the mean values is small according to Equation (32). Figure 17

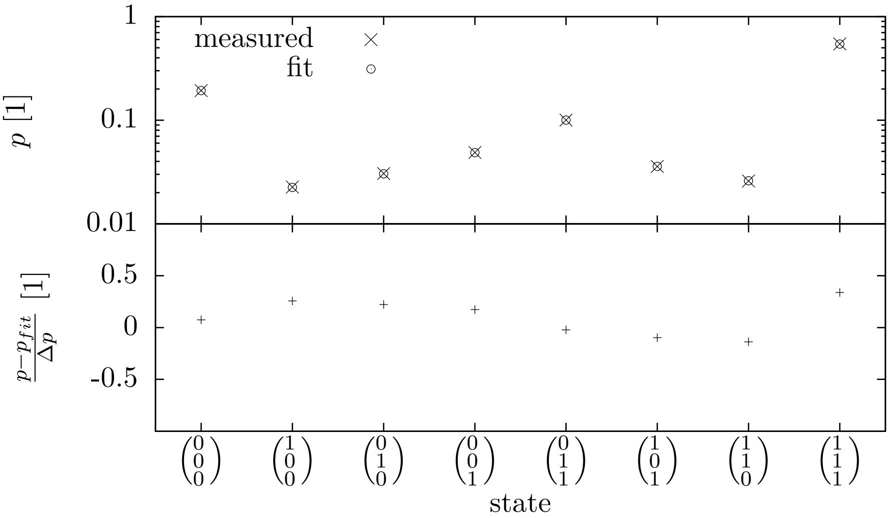

Figure 17: Measured probabilities and deviations from the fitted Boltzmann distribution. The probabilities are averaged over 100 emulations of the network. The deviations are normalized to the uncertainty Δp of the mean value. For this fit, the sum of squared residuals is $\chi^2 = 0.295$.

shows the measured probabilities averaged over 100 emulations in the upper plot and the normalized residuals below. The error bars of the probabilities are not drawn since they cannot be seen on this scale. The sum of squared residuals is $\chi^2 = 0.295$ for this fit. For one degree of freedom the probability that χ^2 is larger than 0.295 is 58.7%.

Figure 18 shows the probabilities averaged over 10000 emulations of the network. The relative uncertainties of the probabilities are less than 0.3%. Although the deviations to the fitted Boltzmann distribution are less than 1%, these differences cannot be explained with the statistical uncertainty of the measurement. The sum of squared residuals between fitted and measured probabilities in Figure 18 is $\chi^2 = 25.6$. Such a large χ^2 due to statistical fluctuations is likely to happen in one of 1.8×10^6 cases. These systematic deviations are expected to be due to the simplifying assumptions made in theory (cf. [Petrovici et al., 2013]). In section 5 these systematic errors are discussed in detail.

It should be noted that the execution and evaluation of the 10000 experiments took 13 min. This time covers retrieving the spike times from the device and writing new Poisson spike trains to the device for each run as well. The biological time of those 10000 experiments is 14 h.

The network was also emulated with a higher bias and negative Boltzmann weights. The Boltzmann distribution fits the measured probabilities for 10000 runs within the expected error if one assumes a refractory period of 20 ms. This is much shorter than the observed refractory period which is (28 ± 2) ms. Figure 19 shows the measured probabilities and the deviations to the fitted Boltzmann distribution. The sum of the squared residuals is 2.06. The probability to get $\chi^2 > 2.06$ with one degree of freedom

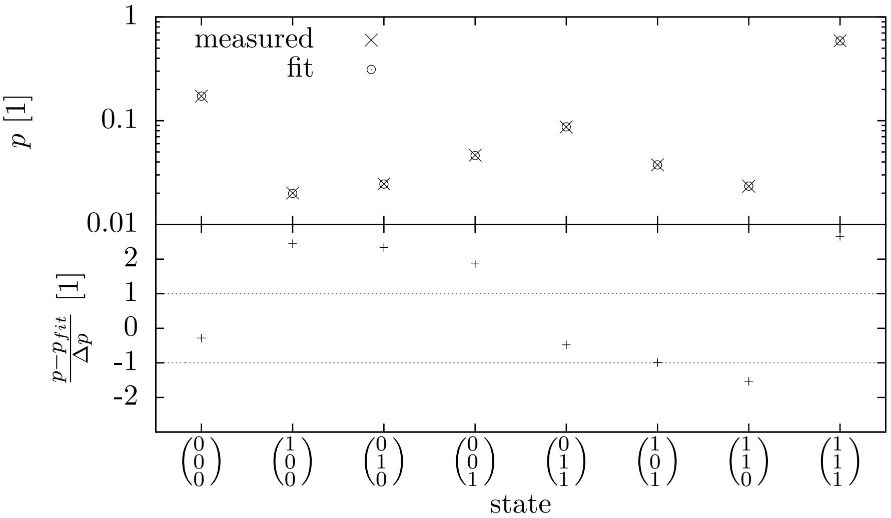

Figure 18: Measured probabilities and deviations from the fitted Boltzmann distribution. The probabilities are averaged over 10000 emulations of the network. The deviations are normalized to the uncertainty Δp of the mean value. The dotted lines indicate the $\pm 1\sigma$ interval. For this fit, the sum of squared residuals is $\chi^2 = 25.6$.

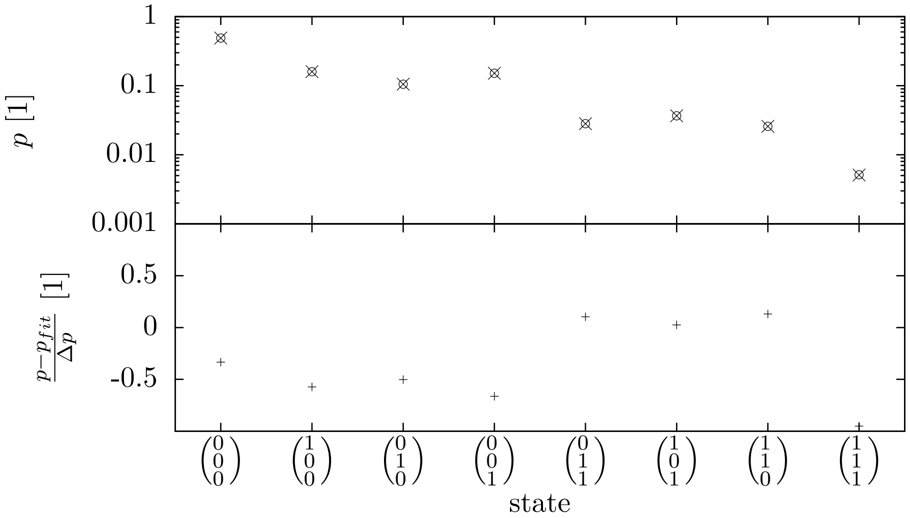

Figure 19: Probabilities of the 8 network states for 10000 runs of the network with negative connection weights. The errors which are less than 0.5% are not shown since they are not visible on this scale. The lower plot shows the deviations to the fitted Boltzmann distribution.

is 15%. With the correct refractory period of 28 ms used for the analysis of the states, the $(1, 1, 1)$ state is enhanced compared to the fitted Boltzmann distribution.

The fit parameters for the two calibrated networks are listed in Table 5 in the appendix.

5. Discussion and Outlook

The presented work indicates that Boltzmann sampling might be realizable on the Spikey microchip. In this thesis, several achievements towards Boltzmann sampling are presented.

The refractory mechanism proved to be successful and applicable. This includes the implementation of reliable synfire chains that were tuned with a robust calibration algorithm – especially without any need of adjusting the connection weight matrix manually. The calibration algorithm does not need any calibrated mapping of the LIF parameters to the hardware parameters. For tuning the compensation strength and connection strength, the autocorrelation and cross-correlation proved to be helpful tools. With these methods, the compensation and connection strengths can be calculated through the recorded spike times. Those can be read from the device quickly for all neurons at once. Using this flexible framework, there is no need to record the membrane potential. The membrane potential has the disadvantage of being accessible for only neuron at a time. With the calibrated synfire chain and the tuned compensation strength, the desired activation function can be reproduced for two of the three sampling unit networks. Although it does not fit for the third sampling unit an explanation for the deviations is given in section 4.4. They are expected to be minimized with further calibration effort.

The final measurements indicate that the two networks sample from the Boltzmann distribution. The first tested Boltzmann machine has a low bias and positive weights and the second a high bias with negative weights. The sampling units of both networks operate in the lower half of their activation function. Since the fit of the Boltzmann distribution to the measured probabilities is done with one degree of freedom, the errors were analyzed carefully. Although deviations between the measurements and the fit are less than 1% in all cases there are systematic deviations if the measured probabilities were averaged over 10000 emulations. This reduces the uncertainty of the mean value. At this precision, the deviations to the Boltzmann distribution are higher than the expected uncertainty. The measured deviations between measurement and fit for the network with excitatory connections are likely to happen in one of 1.8×10^6 cases due to statistical reasons. Therefore, these deviations are likely to be systematic errors. There are many possible reasons for those systematic deviations which are mentioned in the following paragraph.

The theory assumes the high-conductance-state approximation. It should be measured if this approximation holds for the chosen configuration of the network as well. The synaptic delay may also be a reason for the systematic deviations since an instantaneous interaction is assumed in theory. It takes $3 - 4$ ms for one synfire chain group to excite the next one (cf. Figure 6). This time is not only the signal traveling time but also the rise time of the membrane voltage towards the threshold potential. However, it may indicate the timescale of the interaction time. Furthermore, the refractory period is assumed to be constant. In the presented networks the refractory period may vary due to the background noise, the bias and the synaptic input of the other neurons. The finite rise time of the sampling neuron's membrane potential from the inhibitory reversal potential back to its mean free potential is changed by these current inputs. For the same reason the interaction time of the Boltzmann connection varies as well. Ideally, it is assumed be as long as the refractory period. Since it was tuned for symmetry and strength but not for its precise temporal evolution it can happen that the interaction has a different duration than the refractory period. This is not the case for the Boltzmann machine. As shown in section 4.4, the configuration of the synfire chains of the sampling unit networks highly influences the sampling units behavior. An incomplete run of the synfire chain increases the effective refractory period significantly and therefore strongly influences the sampling properties.

These proposed reasons for the systematic deviations between the Boltzmann distribution and the measured probabilities each leave room for further investigation. Especially further tuning of the synfire chain robustness, of the compensation strength and of the curve of the interaction strength are three promising approaches to minimize the systematic deviations. The results could also be cross-checked by simulations of the same setup to see if the same systematics occur with ideal LIF neurons as well.

If the network did not sample from the Boltzmann distribution, these deviations should get larger if the sampling network contains more than 3 sampling units. For 4 neurons there are 16 different states in the Boltzmann domain and therefore 15 independent probabilities that could be measured. The probabilities of the 16 states are given in the Boltzmann distribution by 10 independent variables leaving 5 degrees of freedom for the fit algorithm. This could be realized on one Spikey block by reducing the size of the inhibitory synfire chain populations. The current implementation uses $3 \times 61 = 183$ out of 191 neurons of one neuron block. Six neurons had to be discarded due to continuous spiking. The adjacent block of the chip could be used as well if the network topology was adjusted accordingly. The network as it is described in this thesis cannot be extended to both halves due to the constraint of the spike feedback routing to the synapse line drivers.

The Spikey device proved to be a sufficiently reliable computing platform for the given task. It allows an impressively fast emulation of the experiments – the runtime of 10000 experiments covering 14 h biological time was done within 13 min, including the configuration of the device for each run and the evaluation of the probabilities. The short runtime of the network emulations comes at the disadvantage of the high effort of configuration and tuning. The PyNN network description language is a useful tool for the network description. A helpful improvement of the Spikey back-end could be an improved warning system for ambiguous neuron configurations. The neuron mapping could be improved as well. In the current version, the neuron populations with the same parameter configuration have to be mapped manually to the Spikey chip such that the correct neurons share the same parameters.

This thesis provides not only a well-documented python module that aims towards Boltzmann sampling with neuromorphic hardware, it may also serve as an example code base for further Spikey projects.

Appendix

A. PyNN Indices of Neurons and Synapse Line Drivers on Spikey

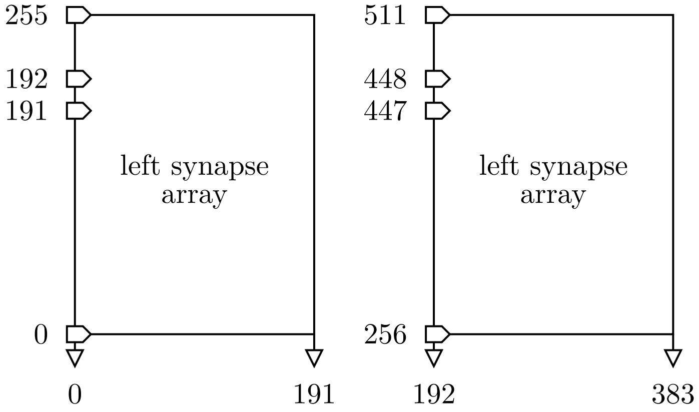

Figure 20: Indices of the hardware neurons and synapse line drivers as they are used by the PyNN interface. The synapse line drivers are on the left of each synapse array, the neurons at the bottom. The line drivers 0 to 191 and 256 to 447 can take the feedback from the neurons. Externally generated spikes can be fed to the network with all drivers. These external sources are mapped to the line drivers in decreasing order as they are build with PyNN from index 256 or 511 on.

B. Routing of Neuron Feedback

Each synapse line driver can take input from one of four spike sources as described in section 2.1.2. The following table lists the routing options for the spike feedback from the neurons.

Driver Index	**Feedback neuron index**	
	same block	**adjacent block**
0	0	193
1	1	192
2	2	195
3	3	194
...	...	...
191	191	342
192	None	None
...	...	...
255	None	None
256	192	1
257	193	0
258	194	3
259	195	2
...	...	...
447	383	190
448	None	None
...	...	...
511	None	None

Table 4: Connection options for feedback from neurons to the synapse line drivers. One synapse line driver cannot take feedback from more than one source. While the neurons of the same block connect to the synapse line drivers in the same order, the even and uneven indices are interchanged for feedback from the adjacent block.

C. Spike Triggered Average

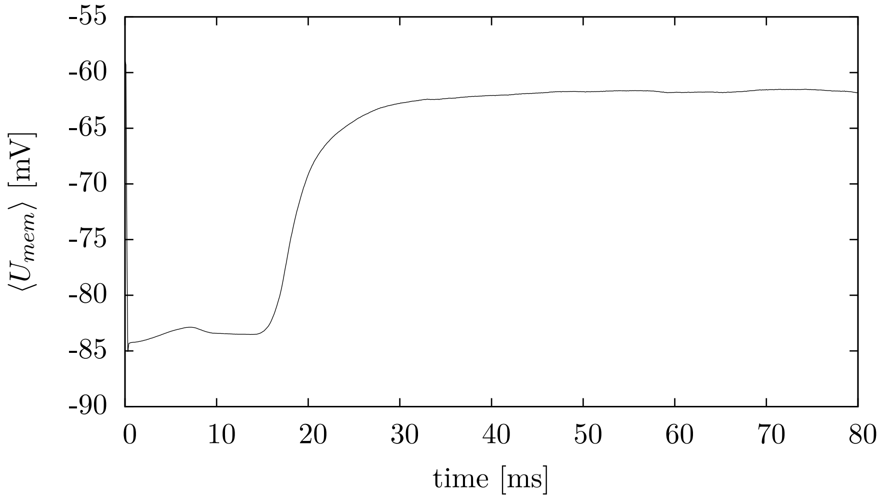

Figure 21: Spike triggered average of a sampling neuron with a tuned compensation strength. The membrane potential is averaged over 50 runs of the network with a target firing rate of 5 Hz. The error of the mean membrane potential is not drawn since it cannot be seen on the given scale.

D. Cross-correlation for an Uncalibrated Connection

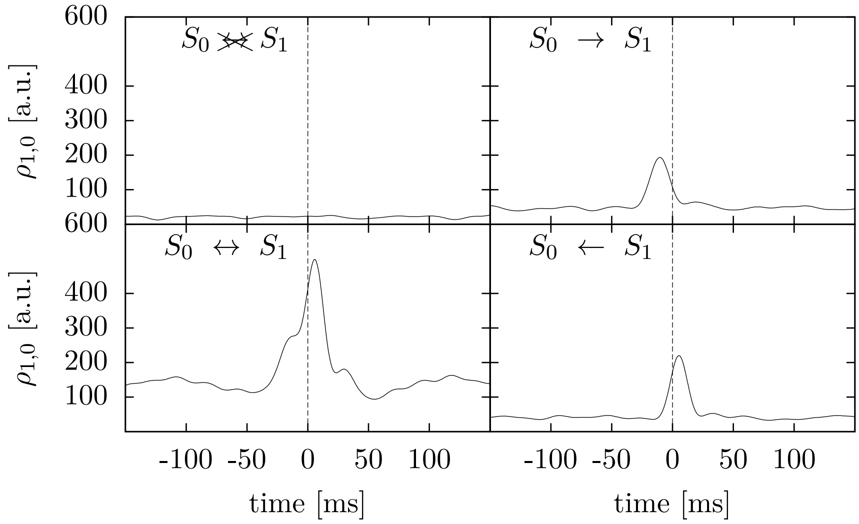

Figure 22: Cross-correlation between the spike trains of the first and the second sampling neuron with an excitatory connection. It is measured for four different configurations which are from the upper left in clockwise direction: a disabled connection, an enabled forward connection, an enabled backward connection and both directions enabled.

E. Boltzmann Parameters

The table shows the fit results for the two networks. One with a low bias and positive weights and one with a high bias and negative weights. The Boltzmann distribution is fitted to the probabilities which are averaged over 10000 emulations of the network.

Parameter	**Positive Weights**	**Negative Weights**
b_0	$2.139\,33 \pm 0.000\,22$	$1.117\,476\,0 \pm 0.000\,002\,3$
b_1	$1.910\,72 \pm 0.000\,22$	$1.538\,805\,7 \pm 0.000\,003\,4$
b_2	$1.291\,17 \pm 0.000\,18$	$1.174\,067\,9 \pm 0.000\,003\,0$
w_{23}	$2.406\,83 \pm 0.000\,24$	$-0.141\,420 \pm 0.000\,027$
w_{13}	$1.849\,05 \pm 0.000\,23$	$-0.298\,583 \pm 0.000\,023$
w_{12}	$2.125\,72 \pm 0.000\,17$	$-0.286\,901 \pm 0.000\,028$

Table 5: Fitted parameters of the Boltzmann distribution for the network with positive and with negative connection weights.

References

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, pages 147–169, 1985.

Daniel Brüderle. *Neuroscientific Modeling with a Mixed-Signal VLSI Hardware System.* PhD thesis, Universität Heidelberg, 2009.

Lars Buesing, Johannes Bill, Bernhard Nessler, and Wolfgang Maass. Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLoS computational biology*, 7(11):e1002211, 2011.

Andrew P Davison, Daniel Brüderle, Jochen M Eppler, Jens Kremkow, Eilif Muller, Dejan Pecevski, Laurent Perrinet, and Pierre Yger. Pynn: a common interface for neuronal network simulators. *Frontiers in Neuroinformatics*, 2(11), 2009. ISSN 1662-5196. doi: 10.3389/neuro.11.011.2008. URL `http://www.frontiersin.org/neuroinformatics/10.3389/neuro.11.011.2008/abstract`.

Asja Fischer and Christian Igel. An introduction to restricted boltzmann machines. In Luis Alvarez, Marta Mejail, Luis Gomez, and Julio Jacobo, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 7441 of *Lecture Notes in Computer Science*, pages 14–36. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33274-6. doi: 10.1007/978-3-642-33275-3_2. URL `http://dx.doi.org/10.1007/978-3-642-33275-3_2`.

Geoffrey E. Hinton. Boltzmann machine. *Scholarpedia*, 2(5):1668, 2007. revision #91075.

Giacomo Indiveri, Bernabe Linares-Barranco, Tara Julia Hamilton, André van Schaik, Ralph Etienne-Cummings, Tobi Delbruck, Shih-Chii Liu, Piotr Dudek, Philipp Häfliger, Sylvie Renaud, Johannes Schemmel, Gert Cauwenberghs, John Arthur, Kai Hynna, Fopefolu Folowosele, Sylvain SAÏGHI, Teresa Serrano-Gotarredona, Jayawan Wijekoon, Yingxue Wang, and Kwabena Boahen. Neuromorphic silicon neuron circuits. *Frontiers in Neuroscience*, 5(73), 2011.

Venelin Petkov. Toward Belief Propagation on Neuromorphic Hardware. Master's thesis, Universität Heidelberg, 2012.

Mihai A. Petrovici, Johannes Bill, Ilja Bytschok, Johannes Schemmel, and Karlheinz Meier. Stochastic inference with deterministic spiking neurons, 2013.

Thomas Pfeil, Andreas Grübl, Sebastian Jeltsch, Eric Müller, Paul Müller, Mihai A. Petrovici, Michael Schmuker, Daniel Brüderle, Johannes Schemmel, and Karlheinz Meier. Six networks on a universal neuromorphic computing substrate. *Frontiers in Neuroscience*, 7:11, 2013.

Johannes Schemmel, Andreas Grübl, Karlheinz Meier, and Eilif Mueller. Implementing synaptic plasticity in a vlsi spiking neural network model. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages 1–6, 2006.

Johannes Schemmel, Daniel Bruderle, Karlheinz Meier, and Boris Ostendorf. Modeling synaptic plasticity within networks of highly accelerated i f neurons. In *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, pages 3367–3370, May 2007.

Nitish Srivastava and Ruslan Salakhutdinov. Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research*, 15:2949–2980, 2014. URL `http://jmlr.org/papers/v15/srivastava14b.html`.

David Stöckel. *Spikey Sampling Module Documentation*. Universität Heideberg, 2014. Generated by Doxygen 1.7.6.1.

Stéfan van der Walt, S. Chris Colbert, and Gaël Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011. doi: http://dx.doi.org/10.1109/MCSE.2011.37. URL `http://scitation.aip.org/content/aip/journal/cise/13/2/10.1109/MCSE.2011.37`.

Danksagung

Ich möchte mich herzlich bei Prof. Dr. Karlheinz Meier bedanken, dass er mir ermöglicht hat zu diesem interessanten Thema meine Bachelorarbeit zu schreiben und dass er die Bachelorprüfung durchführt. Dieser Dank gilt ebenso Dr. Johannes Schemmel. Die vertrauensvolle Bereitstellung der Hardwareressourcen war eine unglaubliche Hilfe bei dieser Arbeit.

Ich bin meinen Betreuern Ilja Bytschok und Mihai Petrovici sehr dankbar für die Zeit, die sie sich genommen haben um zu diskutieren und bei Fragen zu helfen. Ganz besonders deren Interesse an den Ergebnissen hat meine Arbeit so schnell voran gebracht und motiviert. Danke, dass sie mir die Freiheit bei der Wahl des Themas gegeben haben. Ich habe wirklich gerne daran gearbeitet.

Danke an Thomas Pfeil, dass er für mich einen Spikey Chip eingerichtet hat, der für meine Emulationen reserviert war. Bei Fragen zu Spikey oder dem Spikey Backend von PyNN hat sich Thomas Pfeil, ebenso wie Eric Müller, die Zeit genommen mir weiter zu helfen. Für weitere Diskussionen und Hilfe bin ich Paul und Oliver dankbar.

Ein riesiges Dankeschön möchte ich auch besonders allen aussprechen, die für die gute Arbeitsatmosphäre gesorgt haben. Dazu gehören Agnes, Laura, Boris und Robin im „Bachelorbüro“, sowie die Mittagessen und Kaffeepausen mit Mitja, Sebastian, Paul, Christoph, Thomas, Roman, Arno, Tobias und Johann. Es ist schön mit euch zu arbeiten.

Erklärung

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