Lab Report FP09: Neuromorphic computing

Students: Robin Dorstijn, Moritz Epping
Tutor: Jakob Kaiser

February 2021

Contents

Abstract

In this paper we examine the SPIKEY Chip for neuromorphic computing and come to the conclusion that though it is useful for prototyping neural networks in a hardware fashion, it is likely not ready for real life applications as the electronics are too sensitive to environmental factors. Methods for calibrating individual neurons as well as networks are suggested.

1 Theoretical Foundations

1.1 General background

1.1.1 Motivation

Brains and computers often are compared because of their shared purpose, decision making, though they have irreconcilable differences in architecture. Modern computers function on the basis of the van Neumann architecture[**?**], which separates the control unit from the memory and the computation unit. Though it simplifies the structure of the computer and makes it modular, it does create a clear bottleneck at the communication layer, commonly known as the von Neumann bottleneck. This creates an energy inefficiency that is absolutely unacceptable for biological systems like that of humans, who spend around 20% of its total energy uptake on the brain[**?**]. This most likely has provided significant evolutionary pressure for hominids to optimize metabolic resource usage[**?**]. For this reason a new architecture for computers has been suggested: neuromorphic[1] computing.

1.1.2 Signal processing in the brain

Neurons are the basic computational components of the brain, transmitting and morphing signals. Each neuron receives signals from others at the dendrites, which shift the electric potential at the membrane of the cell at the site of the synapse[2], creating an ion wave crossing the entire membrane of the neuron, reaching it's axon, allowing it to pass the signal on to other neurons.

The potential at the neuron membrane as a function of time is plotted in figure 1, here it is demonstrated how a potential shift caused by pre-synaptic[3] neurons initiates a spike in the membrane potential, which forces a release of ion containing vesicles at the terminal bulb[4], facilitating the transfer of the signal by changing the relative potential at the dendrites.

1.1.3 The LIF model

In order to emulate the behaviour electronically, a leaky integrate and fire (LIF) model is used. A single neuron is represented by an electronic circuit like the

[1] "Neuro" as in brain, "-morphic" as in having the shape of, forming "having the shape of a brain".

[2] The connection point of two dendrites where the axon of one meets the dendrite of the other.

[3] That is the neuron that passes on a signal to the post-synaptic one.

[4] **Terminal bulb**: the collections of appendages at the end of the axon that it uses to pass the signal on to connected neurons.

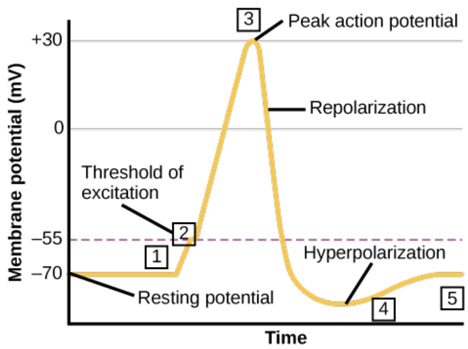


Figure 1: Formation of an action potential. Source: `https://courses.lumenlearning.com/boundless-biology/chapter/how-neurons-communicate/`

one in figure 2. When the chip is not excited, nor inhibited there is a constant "leak" voltage present on the membrane, representing the biological rest potential of the membrane potential. When the threshold voltage V_{thres}, the amplifier functioning as a comparator in this setting, sends out a V_{CC} signal to a digital unit that processes that the neuron has fired and is responsible for closing the switch that keeps the "membrane potential" V_m at some V_{reset} for the refractory period. This time is meant to mimic the period the voltage at the membrane increases non-linearly and therefore loses its sensitivity to input. The digital processing unit is also responsible for sending the signal (regardless of if it is inhibitory or excitatory) to the neurons that the are connected to one that just fired.

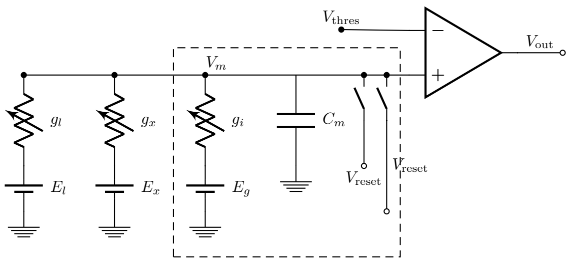


Figure 2: Circuit of a single neuron in a LIF chip.

1.2 Investigating A Single Neuron

1.3 Calibrating Neuron Parameters

In the case of a single unconnected neuron the membrane potential can be understood with application of the Kirchoff formula to the circuit in figure 3.

$$C_m \frac{dV_m}{dt} = g_l(E_l - V_m)$$

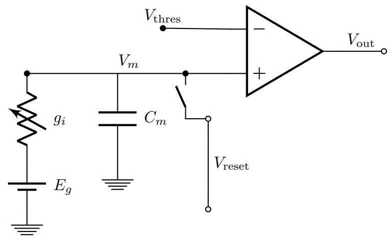


Figure 3: Circuit of a single neuron in a LIF chip.

Clearly, this is a ordinary first order differential equation and has the solution

$$V_m(t) = A \exp(\frac{-g_l}{C_m} t) - E_l \tag{1}$$

where A is given by the initial condition:

$$A = V_m(0) - E_l.$$

In order to set a neuron to fire at a regular frequency τ_m, we have to take into account that V_m will be reset to V_{reset} when it hits V_{thres}. Equation (1) can be rephrased in terms of t.

$$\tau_m = -\ln(\frac{V_{\text{thres}} - E_l}{V_{\text{reset}} - E_l}) \frac{C_m}{g_l}$$

The characteristic time constant τ_c of the system can be found by setting V_{thres} as a function of E_l and V_{reset}.

$$V_{\text{thres}} = E_l - (E_l - V_{\text{reset}}) \exp(-1)$$

The resulting τ_c is then only dependent on C_m and g_l:

$$\begin{aligned}
\tau_m &= -\ln(\frac{E_l - (E_l - (E_l - V_{\text{reset}}) \exp(-1))}{V_{\text{reset}} - E_l}) \frac{C_m}{g_l} \\
&= -\ln(\frac{(E_l - V_{\text{reset}}) \exp(-1))}{V_{\text{reset}} - E_l}) \frac{C_m}{g_l} \\
&= \frac{C_m}{g_l} = \tau_c
\end{aligned}$$

This time constant does not take into account the refractory period which should adds a fixed contribution. This was mentioned in equation (**??**), but becomes relevant again when considering the total time.

This describes a good theory of the neuron, however in reality there are some serious production artifacts that require calibration.

1.4 A Single Neuron with Synaptic Input

1.5 Short Term Plasticity

1.6 Feed-Forward Networks

One of the simplest neural networks one can consider is the feed-forward network in which every neuron is arranged so that it passes on the signal it receives to the next. In nature, a single neuron firing has relatively little meaning until it is put in a larger context. Most of these contexts can be simplified by viewing them as 'superpositions' feed forward networks. It is however important to consider that in nature robustness and redundancy is highly favored in critical systems. The function of a feed-forward network can be shutdown entirely if one neuron fails for whatever reason. Hence the concept of populations is introduced, these are groups of neurons that perform the same logical operations[5] This is visualised in figure 4.

In this experiment, connection weights also become relevant for the first time. Any synaptic connection is assigned a weight, this weight represents the membrane potential increase that a signal transfer induces. The importance of this variable is crucial for networks as it nearly completely describes the behaviour of a network.

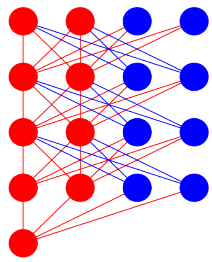


Figure 4: Feed forward network with length 4 and population size 2. Two input neurons are drawn additionally. Exhibitory neurons and connections are drawn in red, inhibitory ones in blue.

1.7 Recurrent Networks

1.8 A Simple Computation - XOR

A very common binary operation that is particularly non-linear (and therefore more complex) is the XOR operation. It can be described with the truth table 1. A trivial solution for this problem was found and dismissed thanks to its probable fragility. Instead the institute provided a different network architecture that has significant benefits over the suggested one: it is much more robust to what is called referred to with 'width' in section **??**. The inhibitory neurons that are operating in a recurrent fashion prevent leaking voltage from causing the any excitatory neuron in the network to fire out of line. This is especially important in an XOR gate, as the chance that noise would 'cancel out' is negligible.

2 Experimental Procedure

2.1 Investigating A Single Neuron

[5] That is to say, they are all either inhibitory or excitatory neurons, they have the same neurons as inputs and the share the same output neurons.

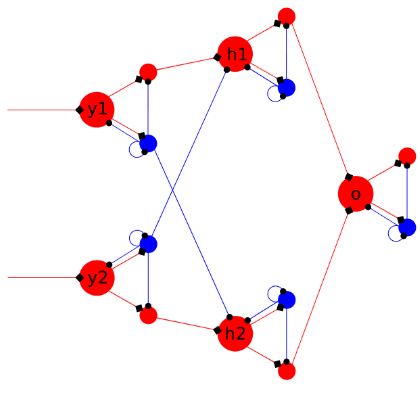


Figure 5: Network that was used to test the ability of neurons to perform logical operations. Functions as an XOR gate.

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Table 1: Truth table of the XOR gate. For more information see: [?]

To assess the reliability of the spikey chip, it is first necessary to investigate the most simple configuration. This consists of a single neuron without any input. As described in section 1.3, the circuit then simplifies significantly. The chip can be configured in this setup by setting the conductances corresponding to other inputs to zero.

By setting the leakage potential above the threshold potential, we can bring the neuron into a constant firing regime despite no other synaptic input. Values of $E_l = -50\text{mV}$ and $V_{thres} = -55\text{mV}$. The firing rate was measured to equal $t_{fir} = (16.10 \pm 0.11)\text{ms}$. According to the circuit diagram, the operation of the spikey chip in this setup corresponds to charging a capacitor with a current given by $I(t) = g_l(t)E_l$. This characteristic voltage behaviour of charging a capacitor can be seen in 6. When increasing the leakage conductance g_l we could observe shorter spike times. This agrees well with our theoretical understanding of the model since we essentially charge the current with a stronger current, thus reaching the firing-inducing threshold voltage sooner.

2.2 Calibrating Neuron Parameters

On the SPIKEY chip 4 neurons were isolated programmatically and given the same parameters:

- V_{reset}: $-80.0\,\text{mV}$
- V_{thresh}: $-55.0\,\text{mV}$
- E_{leak}: $-50.0\,\text{mV}$
- g_{leak}: $20.0\,\text{nS}$

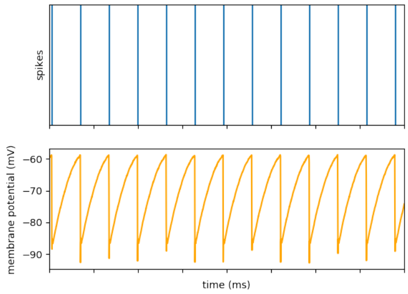


Figure 6: Voltage over a single neuron with leakage potential over threshold potential

This however resulted in wildly differing frequencies. In order to compensate for this g_l was adjusted individually for all of the membranes so that they were all correct within their respective standard deviation: 20.1 mV, 55.0 mV, 60.0 mV, 20.5 mV.

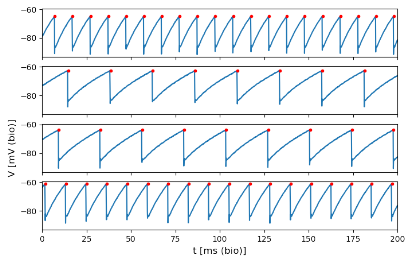


Figure 7: Biological membrane potential of four membranes with the same setting. The red dots indicate when the digital processing units recognizes that voltage reaches the threshold. The rates for the signals are $1.02(2) \times 10^1$ ms, $2.41(3) \times 10^1$ ms, $2.36(1) \times 10^1$ ms, $1.15(1) \times 10^1$ ms, in the same order that they presented in the image.

The scale of the problem becomes even more clear when considering the that one half of the SPIKEY chip has a rate distribution like presented in figure 8, the previous method of trial and error becomes rather infeasible.

Instead an algorithm is suggested that should help find a proper calibration for all neurons that are able to converge on the desired rate.

1. Set the g_l to the default value and measure the rate of the neuron.
2. Set the g_l to double the default value and measure the rate of the neuron.
3. Describe the response of the neuron linearly using the last two points and estimate where the desired rate would lie.
4. Set the g_l to this value and measure the rate, if it is within the standard error: success! If not return to step 3.

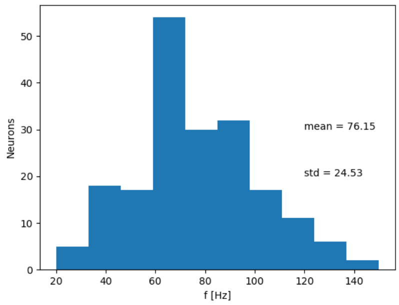


Figure 8: Distribution of firing rate for the same across one side of the chip.

Sadly we were unable to implement the algorithm, but we encourage the reader to.

2.3 A Single Neuron with Synaptic Input

Having confirmed that the most basic behaviour of the chip agrees with the expectations of the model, a more complicated setup could be investigated. The computational power of neurons stems from their interconnectedness. Thus it is crucial to evaluate how a neuron responds to input from other neurons (synaptic input). The quantity of interest here is the membrane potential of the neuron under scrutiny. This voltage can be easily monitored as a function of time using the pynn.record() function. As discussed in the theory section, we have two types of synapses: Excitatory ones inducing positive spikes and inhibitory ones inducing negative spikes. It is actually crucial to have both types of synapses to perform calculations. This can be understood analogously to positive and negative weights in artificial neural network that describe the influence of some features on the neuron. Besides the requirement to have two different types of synapses that have different behaviour it is also crucial to examine in what ways the synapse can form the standard input of a spike into an EPSP. That is, how big the transmitted spike will be and how fast it decays.

2.3.1 •

To start with the latter, the effect of the parameters **drvifall** and **drviout** was investigated. The following results were found:

- **Drviout** controls the magnitude of the voltage spike. By increasing its value the spike height was reduced.
- **Drvifall** controls the width of the peak, that is, how fast it decays. Increasing this parameter makes the peak more sharp.

It could also be observed that those two parameters did not only affect the height or width of the voltage spike exclusively. Instead, decreasing drvifall was also found to lead to spike of smaller magnitude besides making the voltage ramp fall more slowly. This is important to consider since it affects the degree of precision to which we can fine-tune the EPSP of a neuron.

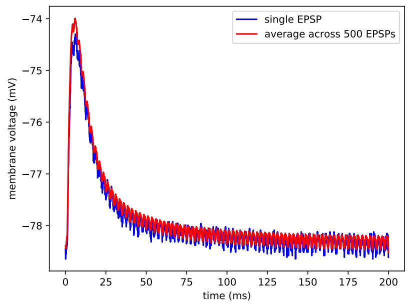


Figure 9: Post synaptic potential for default settings of parameters

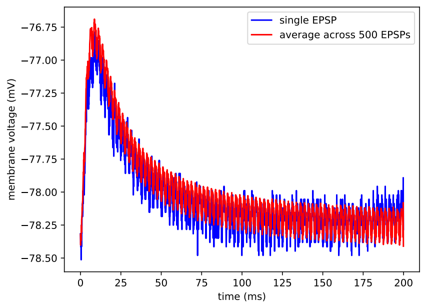


Figure 10: EPSP for low drviout.

2.3.2 •

To verify that apart from the shape the general form of the signal could also be modified, the parameters were set in such a way that inhibitory synapses were achieved. For the same parameters (**drvifall** = 0.3, **drviout** = 0.5) the EPSP was observed for both inhibitory and excitatory neurons. It can clearly be seen how they lead to different shapes.

2.3.3 •

Besides investigating whether the simple circuits behave as expected, it is also crucial to develop an idea of what role fluctuations play in this experiment. Only if we know the source and extent of fluctuations can we try to work on a model that corrects them when we want to use the neuromorphic chip for actual computations. Before, we have already looked at temporal noise (e.g. for the regular spiking regime in ex.1). Now we want to investigate the synapse-to-synapse noise. For this we vary the row of the stimulating synapse for identical parameters and plot a histogram of the EPSP heights. In 2.3.3 this distribution can be seen. It is quite obvious that despite identical parameters the individual stimulating synapse matters a long with regards to the resulting EPSP height. The mean is 3.06mV, but since the distribution is clearly asymmetric, the median is a more meaningful quantity. This was determined to be 2.64. The

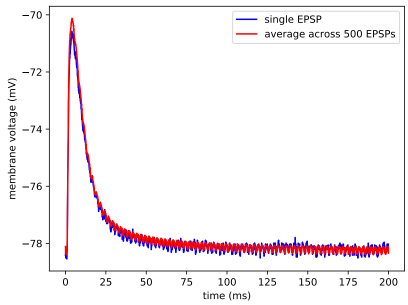


Figure 11: EPSP for high drviout.

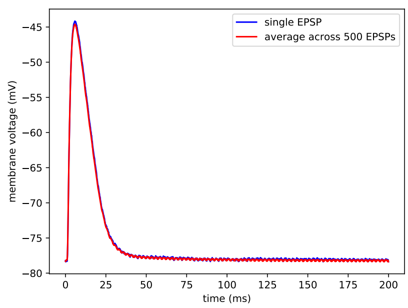


Figure 12: EPSP for low drvifall.

standard deviation is with 1.17mV very large. We can conclude that production errors of different synapses need to be taken into account to get correct results.

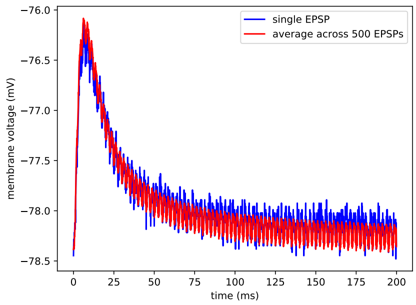


Figure 13: EPSP for high drvifall.

2.4 Short Term Plasticity

2.5 Feed-Forward Networks

2.6 Recurrent Networks

2.7 A Simple Computation - XOR

3 Results

3.1 Investigating A Single Neuron

3.2 Calibrating Neuron Parameters

3.3 A Single Neuron with Synaptic Input

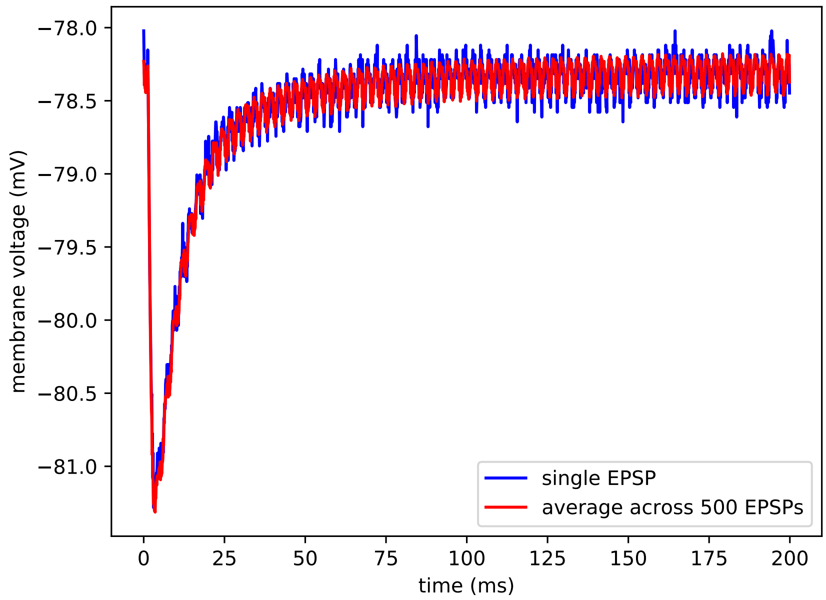
3.4 Short Term Plasticity

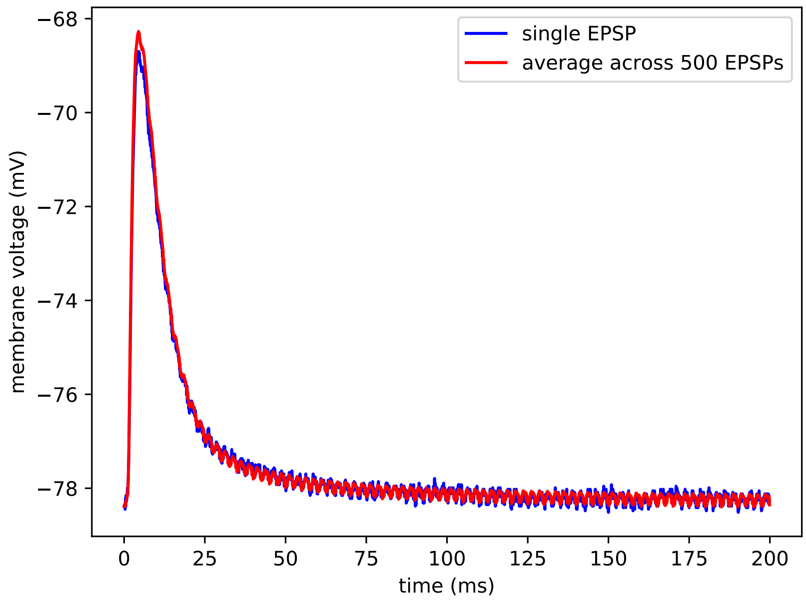
3.5 Feed-Forward Networks

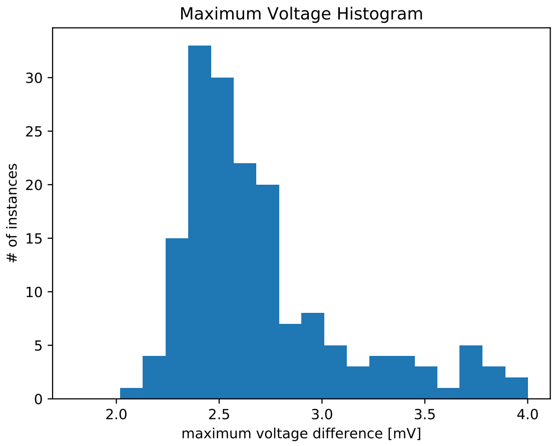
A feed-forward network is described by the population number, the chain length (see figure 4) and the several weights of the following type connections:

- From the initial excitatory neuron to the first excitatory member of the chain. (ex0 → ex)
- From the initial excitatory neuron to the first inhibitory member of the chain. (ex0 → inh)
- From an excitatory neuron in the chain to the next excitatory neuron. (ex → ex)
- From an excitatory neuron in the chain to the next inhibitory neuron. (ex → inh)
- From an inhibitory neuron in the chain to the next excitatory neuron. (inh → ex)

This coincidentally is also the order of sensitivity (top being most sensitive), particularly when discussing the ‘width’ of the signal. With this is meant the


−78.0
−78.5
−79.0
−79.5
−80.0
−80.5
−81.0
membrane voltage (mV)
single EPSP
average across 500 EPSPs
0
25
50
75
100
125
150
175
200
time (ms)


−68
−70
−72
−74
−76
−78
membrane voltage (mV)
single EPSP
average across 500 EPSPs
0
25
50
75
100
125
150
175
200
time (ms)



signals loop.png

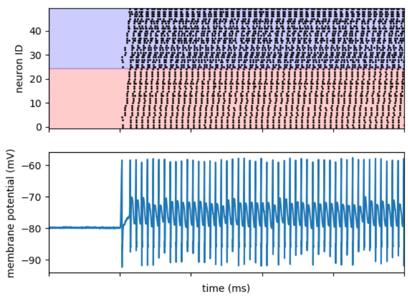


Figure 14: Feed-forward network configured in a loop. The population is 5, as is the chain length. Note that at the end the width starts to increase. The top figure shows all signals send by the neurons. The blue area represents inhibitory neurons, the red excitatory. The bottom figure shows the membrane potential of the first excitatory neuron.

the amount of consequent signals that follow a single input signal. When having a large population, a high (inh $\rightarrow$ ex) wheight was capable of stopping the signal before it would reach the last neuron.

An issue that occurred as the chain length was increased the width of the signal did as well. A length of 60 was reached creating a width of 10. This worked for low populations, however with higher populations the membrane voltage was overall raised to such a degree that the neurons started firing randomly and the signal was lost. The theoretical maximum amount of neurons that could be used in a chain is 192, so that for a population of size n the max chain length $c_{\max}$ would be $c_{\max} = 192/n$.

An interesting phenomena occurs when the network is configured to loop. That is to say the last neuron excites the first. Sadly it was not possible to set the signal delay, which would have allowed for more careful investigation, but still there was a clear effect visible in figure 14.

3.6 Recurrent Networks

3.7 A Simple Computation - XOR

4 Final Conclusions

Our results clearly show that the SPIKEY chip can be used for experimenting with neuromorphic computing in the physical world. Most importantly, it could verified that the elementary building block (LIF-circuit) behaved in a very favourable way. Different characteristic features in different realizations of that circuit in the hardware (due to production errors) can be observed but it was shown to be possible to calibrate each "neuron" in a way that the effect of production errors is mostly cancelled. The circuit allows short term plasticity, a key trait of neurons that allows them to 'learn'. Finally, leaving the few-neuron regime, it was verified that the hardware can be used to model circuits with ≈ 200 neurons, even binary logic was achievable with the hardware.

However thanks to the temporal instability of the chip it is clearly not production ready. It is hard to think of a critical application that would have environmental conditions suited for this chip, that would be simultaneously cost effective. Calibration would likely be sufficiently time consuming that it would ruin any cost effectiveness. That is not to say that it is without application in the world of research, where it can be a useful tool to investigate behaviour of neural structures by making it behave like physical neurons.

5 Discussion

Comparing our quite complicated realization of the XOR-gate to the standard electrical implementation of the binary gate in digital computing leads us to one final important insight: When switching computer hardware to neuromorphic hardware, it does not suffice to merely copy the brain's architecture and continue to use boolean logic like is done in modern computers. It is far too fragile, requiring an enormous amount of neurons and an inefficient use of compute power.

Instead, if neuromorphic computing is to become an integral part of computational methods, it is key the field adapts to the biological blueprint. Naturally that would inspire radically new ways of problem solving in signal processing.

Before that is reached however, a stark increase in the stability of the hardware is required, as even a jelly fish has over 5000 neurons [**?**]. This paper has provided a qualitative review of the chip, however quantitative measurements are still lacking. We hypothesize that temporal differences in the internal temperature of the chip is likely the main culprit of this instability, therefore describing both the sensitivity to a change in temperature and heat generation by the chip during average usage could be useful indicators of the chips ability to perform consistently.