



Christoph Rieger, Bsc.

Hierarchical architectures for spiking Winner-Take-All networks

Master's Thesis

to achieve the university degree of

Master of Science

Master's degree programme: Biomedical Engineering

submitted to

Graz University of Technology

Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Legenstein

Institute of Theoretical Computer Science

Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Robert Legenstein

Graz, April 2023

Contents

1	Protocol of work so far	1
1.1	Rotated lines experiment	1
1.1.1	Introduction	1
1.1.2	Methods	1
1.1.3	Results	5
	Bibliography	7

1 Protocol of work so far

1.1 Rotated lines experiment

1.1.1 Introduction

The goal of this task was to recreate example 2 from Nessler et al. (2013). In this example they fed a winner-take-all spiking neural network images with lines in different orientations on them. The network then clustered the images into ten groups depending on their orientation.

1.1.2 Methods

Input data The images used in this task were generated with a size of 29 x 29 pixel. Black lines going through the center of the image were drawn onto a white background. To simulate noise each pixel had a ten percent chance to have its color flipped. To ensure that all lines in the images have the same length regardless of their orientation a circular mask with a radius of 15 pixel was applied to the images. This recolored all pixel outside of the mask to white. During the training of the network these images were generated in uniformly distributed orientations for each training step. The random chosen orientation could lie between 0 and 359 degrees. Two examples of such images can be seen in figure 1.1.

Neuron model As in Nessler et al. (2013) the input neurons X are firing according to a poisson process with an average firing rate of 20 Hz when

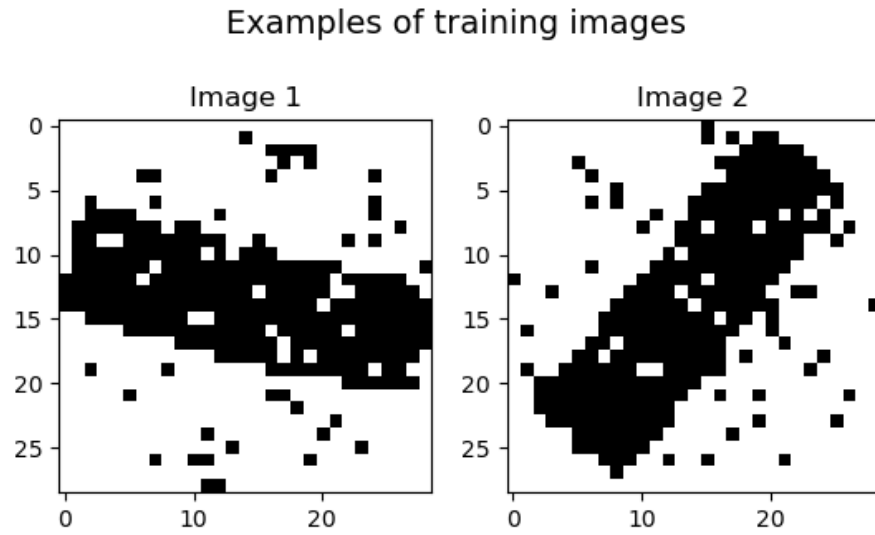


Figure 1.1: Two examples of the generated training data in this experiment.

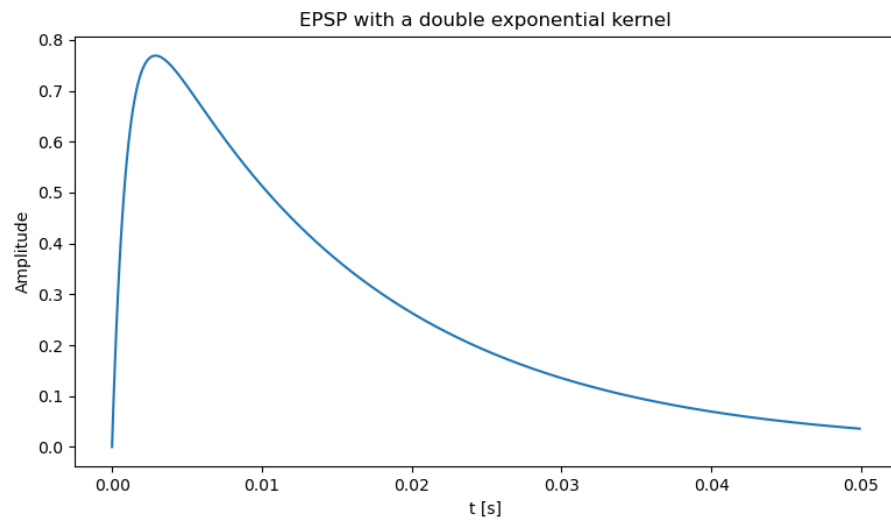


Figure 1.2: Form of an excitatory post synaptic potential generated by a X neuron over time. A double exponential kernel was used to generate this signal. These signals are fed to the next layer of the network.

active and with 0 Hz when in an inactive state. The excitatory post synaptic potentials (EPSPs) $x_i(t)$ these neurons produce can be seen in figure 1.2.

The firing rate of an output neuron Y_k depends exponentially on its membrane potential U_k and the inhibitory signal I_{inh} it receives. The membrane potential of an Y neuron depends on the EPSPs of all X neurons. This can be seen in equation 1.1. The chance of an individual Y neuron to fire within a time step δt is given in equation 1.2.

$$r_k(t) = e^{u_k(t) - I(t)} \quad (1.1)$$

$$r_k(t) \cdot \delta t \quad (1.2)$$

Network architecture Each pixel of an input image was connected to two neurons. The first of these neurons is in an active state when the pixel is black and in an inactive state otherwise. The second neuron expresses the opposite behaviour. As a consequence the network needs 1682 ($29 \cdot 29 \cdot 2$) excitatory input neurons x_1, \dots, x_n . These X neurons are then fully connected to ten excitatory output neurons y_1, \dots, y_k . This means that that every input neuron x_i is connected to each output neuron y_k . The membrane potential U_k of each Y neuron is calculated by multiplying the EPSP of each X neuron times the weight of the connection between the X and Y neuron.

$$U_k(t) = \sum_{i=1}^n w_{ki} \cdot x_i(t) \quad (1.3)$$

In Nessler et al. (2013) each Y neuron also had an intrinsic excitability w_{k0} which was learned for each neuron. For this experiment however it was omitted, as each orientation of input images was equally likely, thus the intrinsic excitability of each Y neuron would end up being the same.

The Y neurons are modelled in a winner-takes-all (WTA) circuit. This means that whenever one Y neuron spikes, a lateral inhibitory signal is fed to all Y neurons, thus preventing the further activation of all output neurons for 5 ms. After the training of the network each Y neuron should be active for lines in an 18° area.

Inhibition The inhibition signal was chosen to depend on the current membrane potential of the Y neurons. According to Nessler et al. (2013) the overall firing rate of the Y neurons is:

$$R(t) = \sum_{k=1}^K e^{U_k(t)-I(t)} \quad (1.4)$$

Solving this equation for I(t):

$$R(t) = \frac{\sum_{k=1}^K e^{U_k(t)}}{e^{I(t)}} \quad (1.5)$$

$$e^{I(t)} = \frac{\sum_{k=1}^K e^{U_k(t)}}{R(t)} \quad (1.6)$$

$$I(t) = \ln \frac{\sum_{k=1}^K e^{U_k(t)}}{R(t)} \quad (1.7)$$

$$I(t) = -\ln R(t) + \ln \sum_{k=1}^K e^{U_k(t)} \quad (1.8)$$

When implementing the inhibition the $-\ln R(t)$ term of equation 1.8 was overlooked, that means it was assumed to be zero. Because of that R(t) equals 1 when the inhibition is active. This error was not detected at first, as the chance that a Y neuron fires within a time step of 1 ms with active inhibition is 1/1000 due to that oversight.

Whenever a Y neuron produces a spike the inhibition signal I(t) is subtracted from the membrane potential $U_k(t)$ of every Y neuron. This happens for the duration of $\sigma_{inh} = 5ms$. Thus follows:

$$I(t) = \begin{cases} \ln(\sum_{i=1}^k e^{U_k}) & \text{if any } y_k \text{ fired in } [t^f, t^f + \sigma_{inh}] \\ 0 & \text{if any } y_k \text{ did not fire in } [t^f, t^f + \sigma_{inh}] \end{cases} \quad (1.9)$$

Spike time dependent plasticity The weights w_{ki} between neurons x_i and y_k are updated whenever a Y neuron fires. The time window σ was set as 10 ms according to Nessler et al. (2013). The other parameters c and learning rate λ were determined via grid search (see results).

$$\Delta w_{ki} = \begin{cases} \lambda \cdot (ce^{-w_{ki}} - 1) & \text{if } x_i \text{ fired in } [t^f - \sigma, t^f] \\ \lambda \cdot (-1) & \text{if } x_i \text{ did not fire in } [t^f - \sigma, t^f] \end{cases} \quad (1.10)$$

where:

λ ... learning rate

c ... shifts weights values

t^f ... time of y spike

σ ... time window in which X spikes are considered as "before" Y spike

1.1.3 Results

Bibliography

Nessler, Bernhard et al. (Apr. 2013). "Bayesian Computation Emerges in Generic Cortical Microcircuits through Spike-Timing-Dependent Plasticity." In: *PLOS Computational Biology* 9.4, pp. 1–30. DOI: [10.1371/journal.pcbi.1003037](https://doi.org/10.1371/journal.pcbi.1003037). URL: <https://doi.org/10.1371/journal.pcbi.1003037> (cit. on pp. 1, 3–5).