Optimization of Spiking Neural Network

Klara M. Gutekunst

University of Kassel, 34125 Kassel, Germany
klara.gutekunst@student.uni-kassel.com

Abstract. The human brain is particularly good at pattern recognition requiring little power. Due to this fact, scientists are trying to improve their knowledge of the brain and to build systems to model it. Since conventional *artificial neural network* (ANN) often use backpropagation and are thus, biologically implausible or require a lot of power to do computations researchers have started focusing on implementing alternatives such as *spiking neural network* (SNN).
A SNN's neuron fires if its membrane potential exceeds a certain (adaptable) threshold. The membrane's potential is calculated by the sum of incoming spikes with respect their timing. To make this process more biologically plausible, the membrane potential decays exponentially. Synapses are modelled by the connections between neurons. The influence of a synapse on the postsynaptic neuron is represented by the weight of the connection. This weight is altered during training to disconnect neurons from those that have little influence on their spiking activity. The usage of homoeostasis (i.e. adaptive membrane threshold and limiting the firing rate of neurons) prevents single neurons from firing all the time and enables the neurons to learn to different input patterns.
The approach of this paper is tested on the MNIST dataset. TODO

Keywords: unsupervised learning · spike-timing-dependent plasticity · biologically plausible · lateral inhibition · adaptive spiking threshold · homoeostasis

1 Introduction

In order to find methods of computation requiring little power consumption, researchers have started creating structures modelled on the neurons of the brain. The authors of [4] found that SNNs were suitable means to tackle this task. SNNs differ from normal ANNs in terms of their architecture and learning method. Their inputs are 1-bit spike trains as opposed to 32- or 64-bit messages sent by ANNs. Moreover, instead of backpropagation used for ANNs, SNNs have different learning rules to optimise their weight, such as *spike-timing-dependent plasticity* (STDP) with exponential time dependence. The authors of [4] used *leaky-integrate-and-fire* (LIF) neurons and lateral inhibition. Hence, the approach models the leak of current of real neurons, as well as competition among the neurons.

Applications for this approach include pattern recognition presented in [4] and object shape recognition from [5]. The accuracy is surprisingly high for an unsupervised method.

This paper is structured as followed: In Section 2 the tackled problem regarding SNNs, context-specific terms, the network architecture, methods applied in the approach, as well as the approach itself are described.

Section 3
Section 4
Section 5
Section 4

2 Main part

This section outlines problems, topic-specific terms, learning methods and architecture of the approach SNN.

2.1 Problem

Since SNNs are not only influenced by the input values but also by the temporal dependencies of these inputs, algorithms aiming to optimise their parameters are rare. Moreover, researchers working with SNNs focus on biologically plausible methods and therefore refuse to use conventional techniques like gradient descent.

2.2 Terms

In this section, technical terms specific to the domain are defined and briefly discussed.

Spike trains are defined as multiple spikes. They are encoded as a 1-bit array, where each entry indicates whether a spike (1) or no spike (0) occurred at a certain point in time.

Rate-based model According to [3] there are two central beliefs with regard to the question of how the neurons communicate with each other. The firing rate of a neuron is an abstract measurement of the average number of spikes per unit (e.g. duration, neuron, trial). The Rate-based model believes that the firing rate captures most of the information, hence the timing of spikes is meaningless.

Spike-based model According to the Spike-based model, the firing rate is not sufficient to describe the neural activity. The spike timing defines spike trains and individual spikes. As stated in [3], the Rate-based model assumption is stronger than the one of the Spike-based model.

Rate-based learning uses backpropagation during training, but converts the ANN to a SNN (i.e. transfers trained weights) afterwards. According to [4] and [1] the usage of backpropagation is biologically unrealistic.

2.3 Network architecture

According to [4] the ANN has an input layer and a preprocessing layer. The input layer consists of $28x28$ neurons, i.e. one for each input pixel. All the neurons of the input layer are connected to all the neurons of the preprocessing layer (all-to-all).

The preprocessing layer has excitatory neurons and inhibitory neurons. While every excitatory neuron is connected to exactly one inhibitory neuron (one-to-one), every inhibitory neuron is connected to all excitatory neurons except the one it is already connected to. This structure creates lateral inhibition and creates competition among the excitatory neurons.

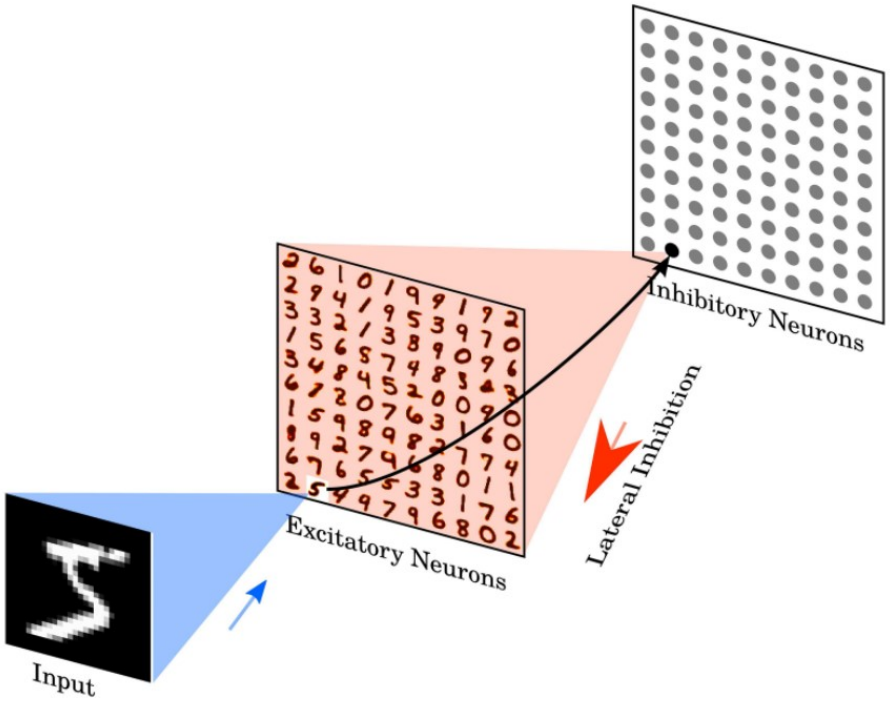

Fig. 1: architecture of the SNN from [4]

2.4 Methods

Concepts used include *leaky-integrate-and-fire* (LIF) neurons, lateral inhibition, intrinsic plasticity and conductance-based synapses.

Timing of spikes According to [2], the timing of spikes determines the type of synaptic changes: If multiple postsynaptic spikes occur close after presynaptic spikes a *long-term potentiation* (LTP) is induced, whereas repetive postsynaptic spike before the presynaptic ones lead to *long-term depression* (LTD).

Learning using STDP The synapses of the SNN are trained using unsupervised *spike-timing-dependent plasticity* (STDP). The weight of a connection between two neurons models a synapse. According to [1], STDP is a synaptic learning rule, which adapts weights of synapses according to their degree of causality, i.e. how likely the input causes postsynaptic neuron excitation. STDP increases synaptic weight if the postsynaptic neuron reacts immediately after the presynaptic neuron fires, as stated by [5]. According to [5], the goal of STDP is to strengthen synapses of pre- and postsynaptic neuron pairs, whose postsynaptic neuron reacts immediately after the presynaptic neuron fires. If the postsynaptic neuron fires before the presynaptic one the spike has another origin and thus, the synapse is weakened to disconnect the neurons.

$$\Delta w = \eta(x_{pre} - x_{tar})(w_{max} - w)^{\mu} \tag{1}$$

Equation 1 from [4] calculates the weight change after a postsynaptic spike arrives (i.e. the synapses' importance is changed according to its influence on the postsynaptic neuron). η is the learning rate, presynaptic trace x_{pre} tracks the number of recent presynaptic spikes (decaying if no spike arrives, increased by one otherwise), μ is the dependence of the update on the previous weight and x_{tar} is the target value of the presynaptic trace at the moment of the postsynaptic spike. If x_{tar} is high, i.e. many spikes arrived at the postsynaptic neuron, the weight will possibly not be increased (especially if fewer spikes arrived at the presynaptic neuron indicated by x_{pre}).

The authors of [4] also outline other STDP learning rules. Some STDP models rely on a teaching signal, which provides the right response. The authors of [1] describe training with the usage of teaching signals: Teaching signals are generated by Poison spike generators. A teaching signal is sent to the correct pool of neurons representing the class (i.e. digit) after a stimulus was presented in the training phase. The goal of this approach is to make each neuron pool selective to one class of input.

Neuron model In reality, a neuron fires if the membrane's potential crosses the membrane's threshold ν_{thresh}. On the report of [4], after firing the neuron's membrane potential is reset and within the next few milliseconds cannot spike

again.

$$\tau \frac{dV}{dt} = (E_{rest} - V) + g_e(E_{exc} - V) + g_i(E_{inh} - V) \tag{2}$$

Equation 2 describes the membrane voltage (i.e. the value compared to the threshold V_{thres} which determines whether the neuron fires) change over time constant τ. According to [4] E_{rest} is the resting membrane potential, E_{exc}, E_{inh} are equilibrium potentials of excitatory and inhibitory synapses, and g_e, g_i the conductances of excitatory and inhibitory synapses (i.e. the influence of respective synapses on membrane voltage of neuron). The voltage decays (first parenthesis) and is influenced by the excitatory synapses adding to the voltage and the inhibitory synapses subtracting from the voltage.

Synapse model The synapses' conductance g_e/g_i (e excitatory, i inhibitory) model the influence of a presynaptic neuron on another neuron. If a presynaptic spike arrives at the synapse the weight $w_{i,j}$ between neuron i and neuron j is added to g_e/g_i. Otherwise, g_e/g_i is decaying.

$$\tau_{g_e} \frac{dg_e}{dt} = -g_e \tag{3}$$

The decay is computed using Equation 3 from [4]. The change over time constant τ is an exponential decay.

LIF neurons As depicted in Equation 2 from Section 2.4, the membrane potential/ voltage V's current leaks out of the neuron (i.e. decays without incoming spikes over time). Due to the exponential decay, these neurons are called LIF neurons.

lateral inhibition Since every inhibitory neuron is connected to all excitatory neurons except the one it is already connected to, whenever a spike is triggered in an excitatory neuron all inhibitory neurons receive a spike as well. Hence, neurons that did not fire are inhibited and thus, lateral inhibition is created.

intrinsic plasticity

conductance-based synapses

Homoeostasis In order to ensure the neurons have a similar firing rate, the excitatory neuron's membrane threshold is calculated by $\nu_{thresh} + \theta$, where θ is increased every time the neuron fires and exponentially decaying otherwise. Since the membrane potential is limited to E_{exc} a neuron stops firing when its membrane threshold is higher than the membrane potential.

This technique countersteers the effect of inhomogeneity of the input and the lateral inhibition.

Input encoding Wozu?
Wie?
$60,000$ training examples, $10,000$ test examples of $28x28$ pixel images of the digits 0 to 9 compose the MNIST dataset used in [4]. Inputs are Poison spike trains, which are presented for 350ms. The intensity of a pixel (0 to 255) is proportional to the firing rates (0 to $65.75Hz$) of the neurons. If the network does not react to the input, the maximum input firing rate is augmented until it fires in the desired fashion.

Training In order to allow all variables to decay there is 150ms phase without any input between images.

Testing First the learning rate η is set to zero to appoint the neuron's threshold. After that the training set is presented once more. The highest response among the ten digit classes is used to assign a class to each neuron, i.e. labels are used. The predicted value for an input is the class whose neurons have the highest average firing rate. The classification accuracy is determined on the MNIST test set.

3 Results

Benchmark
zugrundeliegende Daten

4 Comparison

to other implementations
to biology

5 Evaluation

Inhibition

Spike-based Learning in ML

Competitive Learning

Robustness

6 Conclusion and Outlook

wdhl wie gut es ist
was kommt noch?
wie zu verbessern?

References

1. Beyel, M., Dutt, N.D., Krichmar, J.L.: Categorization and decision-making in a neurobiologically plausible spiking network using a stdp-like learning rule. Elsevier **48**, 109–124 (2013)
2. Bi, G., Poo, M.: Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. The journal of Neuroscience pp. 10464–10472 (1998)
3. Brette, R.: Philosophy of the spike: Rate-based vs. spike-based theories of the brain. Frontiers in Systems Neuroscience **9**, 1–14 (2015)
4. Cook, M., Diehl, P.U.: Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in Computational Neuroscience **9**, 1–8 (2015)
5. Kim, J., Kim, S.P., Hwang, H., Park, D., Jeong, U.: Object shape recognition using tactile sensor arrays by a spiking neural network with unsupervised learning. IEEE pp. 178–183 (2020)