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 improve this process' biological fidelity, 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 of ANNs. Moreover, instead of backpropagation used for ANNs, many SNN models 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]. In both cases 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, as well as the approach itself are described.

In Section 3 the results of tests approach is presented. Different methods to train SNNs are described and compared in Section 4. The paper concludes with a outlook in Section 5.

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 often 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 binary signals distributed over time. According to [6] an '1' indicates a spike, whereas '0' indicates inactivity. The authors note that the number of timesteps determines the discretization error. A timestep is dependent on the hardware and its associated number of computations performable.

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

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.

The authors of [4] argue that spike-based is preferable learning to rate-based learning in terms of power consumption during the learning procedure and for dynamically adaptable systems.

Rate-based learning is a training method for SNN which uses backpropagation during training. First an ANN is trained with backpropagation with some restrictions as stated in [6]. Then a SNN with integrate-and-fire (IF) neurons and the weights of the ANN is initialized. 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.

2.4 Methods

Concepts used in [4] 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.

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

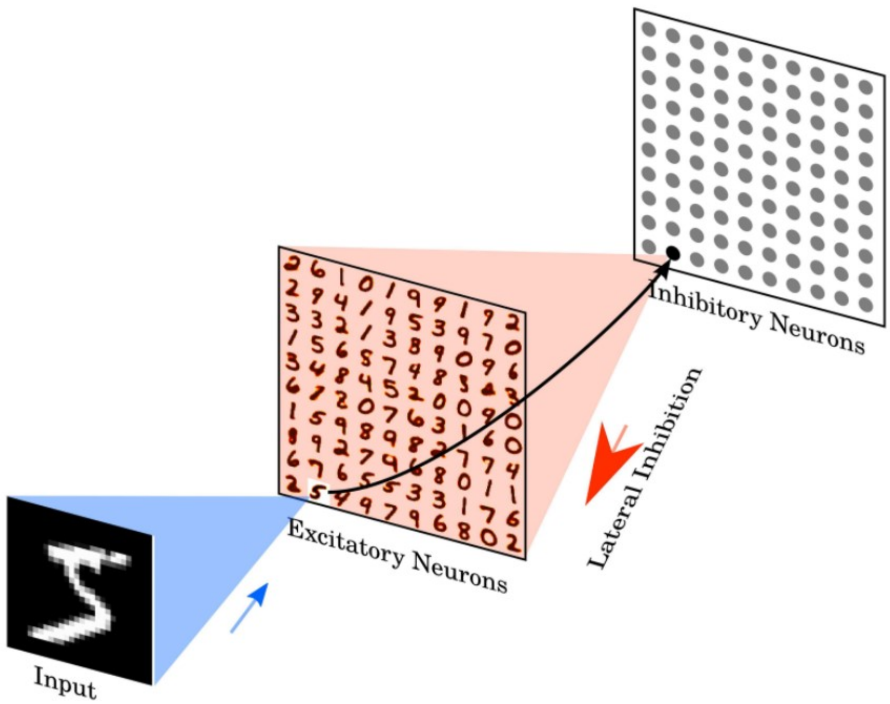

Fig. 1: architecture of the SNN from [4]

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.

Competitive Learning position schlecht: lateral inhibition und neuron nicht erklärt

The goal of the SNN-model presented in [4] is to train its neurons to represent prototypical inputs or an average of similar inputs. To adchieve this goal, the weights of spiking neurons are adapted to become more similar to the input. Lateral inhibition prevents too many neurons from spiking and thus, prevents them from becoming to similar in the course of adapting to the input. This results in the receptive fields of the neurons explorating the input space. To ensure that a approximately constant number of neurons' receptive fields is similiar to an input, homoeostasis guarantees similiar firing rates among the neurons. According to [4], the learning procedure is similar to k-means-like learning algorithms. Hence, increasing the number of neurons may result in at most 95-97 % accuracy.

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 and a soft winner-take-all mechanism are created. The authors of [4] note, that they chose a 1:1 excitatory to inhibitory neuron ratio, rather than the biologically plausible 4:1 ratio, to reduce computional complexity.

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 counteststeers the effect of inhomogeneity of the input and the lateral inhibition.

Input encoding According to [6] SNNs transmit information through spikes and thus, analog values have to be encoded into spikes. As stated in [6], there are different types of encodings based on certain beliefs, for instance those outlined in Section 2.2. The authors of [4] propose the following method to encode analog values into spikes: $60,000$ 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

The model was evaluated on the MNIST dataset. According to the authors of [1], the dataset is a suitable Benchmark for the SNN model, since it provides different difficulty levels of categorization. The authors of [1] emphasise that the MNIST dataset less complex than biological vision.

The addition to accuracy with regard to the classification of the MNIST dataset impulses [1] presents reaction time (RT) distributions. RT is defined as the time between the presentation of a stimulus and the response (i.e. the first pool to reach the descision pool). Usually, the RT of misclassified digits is higher than the RT of correctly classified digits. However, the network also makes fast errors. The results were verified with the Kolmogorov-Smirnov test, indicating that the correctly and misclassified RT distributions are significantly different, whereas as the distributions for stimuli from the training and test set were not.

The authors of [1] compare the performance of the SNN model for different training set sizes n_{train}. They find that the network is able to generalize well if the training set size is sufficiently large. The optimum training set size $n_{train} = 1000$ produced not remarkably higher misclassifaction rates than bigger training set sizes. The most frequent misclassification was the digit nine as a zero.

In [4] the authors compare the performance of the SNN model consisting of different numbers of excitatory neurons and different learning rules. The visualizations suggest that highest number of excitatory neurons (i.e. 6400) produces the best results for all learning rules. The authors also study propose possible reasons for the distribution of misclassified digits. The most common misclassification was the digit four as a nine.

Training type	**(Un-) Supervised**	**Learning rule**	**Performance range**
Rate-based	Supervised (Teaching signal)	different	90-99 %
Spike-based	Supervised	different including calcium variable	91-96 %
Spike-based	Unsupervised	rectangular/ exponential STDP	93-95 %

Table 1: Summary of methods compared in [4]

The authors of [4] include a table visualizing performances of different SNNs. A summary of the types compared is given in Table 1. Rate-based learning methods adchieve the best results.

calcium variable

The authors criticise, since neuron which are not in their refactory period can integrate incoming excitatory potentials and thus, increase their chance of firing possibly not all neurons have the same chance of firing after the refactory period.

Since the authors of [4] have tested multiple different STDP rules, as well as different training set sizes, they point out their model's robustness and good performance in a variaty of different situations, due to the competetion among the neurons resulting in dissimilar receptive fields. Moreover, the authors of [4] argue that their model is more biologically plausible.

4 Comparison

to other implementations, to biology

The authors of [4] present an unsupervised approach to train a SNN model. However, there are other approaches to train SNNs, as well as opions with regard to the biological plausibility of the methods used.

The SNN model from [1] determines its output by choosing the class of the first neuron pool to reach the descision threshold with its accumulated sensory evidence. The authors claim that the original method of using a majority vote of single threshold-based neuron acticity is less biologically plausible.

Not only the [1] model but also the [4] model use inhibition.

The authors of [1] point out that the classifier neuron will not recognize atypical digits (i.e. their stimulus). Therefore, the ability to generalize on new datasets depends on the inputs' similiarity to the learned patterns.

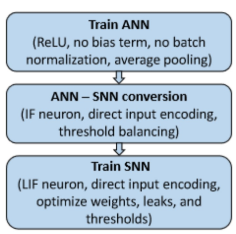

Fig. 2: DIET-SNN training pipeline from [6]

The direct input encoding, leakage and threshold optimization (DIET)-SNN model from [6] is an entriely different approach to those discussed before. This alternative approach uses supervised learning. As depicted in Figure 2, an ANN is trained and thereafter converted into a SNN. The authors of [6] outline the ANN-SNN-conversion in more detail. The initialized parameters speed up the subsequent training with spike-based backpropagation. The converted SNN is trained using error-backpropagation.

Instead of using a Poisson generator (rate-coding: converting analog values to spike-train), the image pixels are directly applied as input. The DIET-SNN model's first layer is trained to encode the input as spikes. The pixel intensities of an image are directly applied to the first convolutional layer's LIF neurons. The neurons accumulate the weighted pixel and generate ouput spikes. This encoding is learned beforehand using gradient-descent.

$$u_i^t = \lambda_i u_i^{t-1} + \sum_j w_{ij} o_j^t - v_i o_i^{t-1} \tag{4}$$

The membrane potential u_i^t of a postsynaptic neuron i at timestep t is calculted using Equation 4 from [6]. $\lambda_i \in [0, 1]$ is the leak factor of the membrane potential, w_{ij} the weight of the synapse between neuron i and j, o_j^t the output spike of neuron j at timestep t and v_i the threshold of neuron i. The first term calculates the leakage of the neuron, the second term the accumulated input of the presynaptic neurons and the third term enables a soft reset (reducing by threshold v_i instead of reset to zero) of the membrane potential after a spike was generated.

The neurons of the output layer accumulate their inputs without any leakage. The output is not a spike, but a softmax function of the membrane potential.

5 Conclusion and Outlook

The study of SNNs, especially with regard to efficient learning procedures poses a interesting insight in the possibilities of modeling systems on the human brain. Concerning the difficulties of training a model which works with input values and their temporal dependencies researchers have already made impressive efforts. The work of [4] proposes an unsupervised approach with formidable performace in a variety of situations. The authors also emphasise the energy efficience of SNNs on neuromorphic hardware.

Besides this example of an unsupervised approach there is a variety of supervised methods to train a SNN, for instance ANN-SNN conversion from [6]. Even though existing approaches performance well, the authors emphasise shortcomings and potential of further research. Bsp. offene Fragen

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)
6. Rathi, N., Roy, K.: Diet-snn: A low-latency spiking neural network with direct input encoding and threshold optimization. IEEE Transactions on Neural Networks and Learning Systems pp. 1–9 (2021)