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 and compared to other spike-timing-dependent plasticity (STDP) approaches.

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. SNNs are allegedly suitable means to tackle this task [6]. They 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. SNNs' inputs are streams of events contrary to ANNs' inputs presentation at one time [7]. 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 model uses *leaky-integrate-and-fire* (LIF) neurons and lateral inhibition [6]. 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 [6] and object shape recognition [11], [10]. 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 regarding performance, optimal parameter choice and other metrics of similar approaches are presented. Methods to train SNNs, which differ more than the ones outlined in Section 3 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. An '1' indicates a spike, whereas '0' indicates inactivity [14]. 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 There are two central beliefs with regard to the question of how the neurons communicate with each other [4]. 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. The Rate-based model assumption is stronger than the one of the Spike-based model [4].

The authors of [6] 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 [14]. Then a SNN with integrate-and-fire (IF) neurons and the weights of the ANN is initialized. The usage of backpropagation is ostensibly biologically unrealistic [6], [2].

2.3 Network architecture

The ANN has an input layer and a preprocessing layer [6]. 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. An 1:1 excitatory to inhibitory neuron ratio is chosen, rather than the biologically plausible 4:1 ratio, to reduce computional complexity [6]. 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. The architecture is depicted in Figure 1. This structure creates lateral inhibition and creates competition among the excitatory neurons.

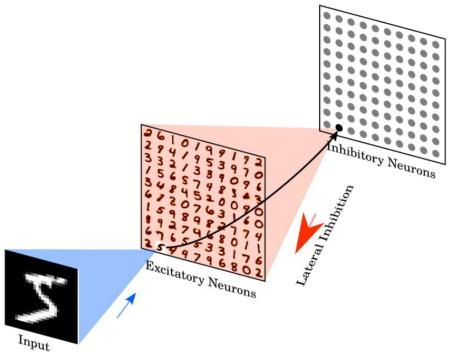

Fig. 1: architecture of the SNN from [6]

2.4 Methods

Concepts used in [6] include *leaky-integrate-and-fire* (LIF) neurons, lateral inhibition, synaptic plasticity and homoeostasis, which are outlined in this section.

Timing of spikes The timing of spikes determines the type of synaptic changes [3]: 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).

Synaptic plasticity Synapses play a crucial role in SNN learning tasks such as recognition and computation [1]. Synaptic plasticity rules is the term used for the different mathematical formulaes realised in activity-dependent modification of synaptic weights. There is a great variety of synaptic plasticity rules, raging from abstract models to detailed models [1].

Instances of abstract models include those based on the timing of spikes, such as pair-based STDP models. Equation 1 from [1] shows the weight update of a pair-based STDP model. $\Delta t = t_{post} - tpre$ is the time difference between the presynaptic spike t_{pre} and the postsynaptic spike t_{post}. If a postsynaptic spike arrives with time window τ_+ before a presynaptic one, potentiation occurs, i.e. the synaptic weight is increased. Analogously to induce depression the postsynaptic spike needs to precede the presynaptic one within time window τ_-. The amount of weight change depends on Δt and the amplitude parameters A^+ and A^-.

$$\Delta w = \begin{cases} \Delta w^+ = A^+ e^{\frac{-\Delta t}{\tau_+}} & \Delta t > 0 \\ \Delta w^- = -A^- e^{\frac{\Delta t}{\tau_-}} & \Delta t \leq 0 \end{cases} \tag{1}$$

Other models, for instance the Triplet-based STDP (TSTDP), not only consider a single pair of presynaptic and postsynaptic spikes, but triplet combinations of spikes (current pre- and post-, current post- and previous post-, current pre- and previous presynaptic one) [1].

Detailed models on the other hand, may take into account state variables, such as membrane potential, accounting for more biophysically realistic models. Changes in the synaptic weight of an instance of the spike driven synaptic plasticity (SDSP) model depend on the postsynaptic membrane potential V_{mem} and the calcium concentration $C(t)$ [1]. The calcium concentration $C(t)$ modification with regard to the influence of decay and incrementation is outlined in [15]. Whenever a presynaptic spike arrives either potentiation of amount a occurs, if the membrane potential V_{mem} is higher than a certain threshold V_{mth} and the calcium concentration $C(t)$ is within certain bounds, or depression of amount b analogously displayed in Equation 2 from [1].

$$W = \begin{cases} W + a, \text{if } V_{mem} > V_{mth} \text{ and } \Theta^l_{up} < C(t) < \theta^h_{up} \\ W - b, \text{if } V_{mem} \leq V_{mth} \text{ and } \Theta^l_{dn} < C(t) < \theta^h_{dn} \end{cases} \tag{2}$$

If the conditions from Equation 2 are no saisfied or no spike arrives, the synaptic weight drifts towards either high or low synaptic weight asymtotes dependent on the weights at a specific time t with respect to a threshold θ_W [1]. The SDSP approach models synaptic weight W decay as displayed in Equation 3 from [1].

$$\frac{dW(t)}{dt} = \begin{cases} \alpha, \text{if } W(t) > \theta_W \\ -\beta, \text{if } W(t) \leq \theta_W \end{cases} \tag{3}$$

Other detailed models, such as BCM-like local correlation plasticity (LCP) are outlined in [1]. The Bienenstock, Cooper and Munro (BCM) rule is based on the Hebbian principles, i.e. strengthening the synaptic connection when two connected neurons have correlated firing activity, and it is a rate-based synaptic plasticity rule introducing synaptic competetion [15]. The weight change over time $\frac{dW(t)}{dt}$ from Equation 4 from [15] consists of two terms: The first one computes the potentiation ($r_{post} \geq \theta$) or depression ($r_{post} < \theta$) of the synaptic weight w, r_{pre}, r_{post} being the firing rates of the pre- and postsynaptic neurons and θ being the modification threshold. δ is the learning rate and ϵ a weight decay parameter. The second term models the decay of the synaptic weight w.

$$\frac{dW(t)}{dt} = [r_{post}(r_{post} - \theta)r_{pre}]\delta - \epsilon \cdot w \tag{4}$$

Learning using STDP The synapses of the SNN are trained using unsupervised *spike-timing-dependent plasticity* (STDP), which has been observed in a range of species from insects to humans [5]. The weight of a connection between two neurons models a synapse. STDP is a synaptic learning rule, which adapts weights of synapses according to their degree of causality [2], [10], 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 [11]. The goal of STDP is to strengthen synapses of pre- and postsynaptic neuron pairs, whose postsynaptic neuron reacts immediately after the presynaptic neuron fires [11]. If the postsynaptic neuron fires before the presynaptic one the spike has another origin and thus, the synapse is weakened to disconnect the neurons.

$$\Delta w = \eta(x_{pre} - x_{tar})(w_{max} - w)^{\mu} \tag{5}$$

Equation 5 from [6] 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 probably not be increased (especially if fewer spikes arrived at the presynaptic neuron indicated by x_{pre}).

If multiple presynaptic and postsynaptic spikes arrive closely in time, there are different methods to compute the total weight change [15]: One could consider only nearest neighbour pairs of presynaptic and postsynaptic spikes, or sum up the weight changes for all pairs.

There are also other STDP learning rules [6]. Some STDP models rely on a teaching signal, which provides the right response [2]. 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.

STDP focusses on target object features, generelly not learning background, since they are usually too different to converge on them [10]. Hence, STDP allegedly extracts informative and diagnostic features.

Neuron model In reality, a neuron fires if the membrane's potential crosses the membrane's threshold ν_{thresh}. After firing the neuron's membrane potential is reset and within the next few milliseconds, i.e. the refactory period t_{ref} [13], the neuron cannot spike again [6].

$$\tau \frac{dV}{dt} = (E_{rest} - V) + g_e(E_{exc} - V) + g_i(E_{inh} - V) \tag{6}$$

Equation 6 from [6] 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. 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) [6]. 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.

$$\tau_{g_e} \frac{dg_e}{dt} = -g_e \tag{7}$$

The decay of g_e (g_i analogously) is computed using Equation 7 from [6]. The change over time constant τ is an exponential decay.

LIF neurons The LIF neuron is a simple, yet biological plausible model of a spiking neuron [13]. As depicted in Equation 6 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 [6].

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 maximum membrane potential.

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

Competitive Learning The goal of the SNN-model is to train its neurons to represent prototypical inputs or an average of similar inputs [6]. To achieve 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 too similar in the course of adapting to the input. This results in the receptive fields of the neurons exploring the input space. To ensure that an approximately constant number of neurons' receptive fields is similiar to an input, homoeostasis guarantees similiar firing rates among the neurons. The learning procedure is similar to k-means-like learning algorithms [6]. Hence, increasing the number of neurons may result in at most 95-97 % accuracy.

Input encoding SNNs transmit information through spikes and thus, analog values have to be encoded into spikes [14]. There are different types of encodings based on certain beliefs, for instance those outlined in Section 2.2.

$60,000$ training examples and $10,000$ test examples of $28x28$ pixel images of the digits 0 to 9 compose the MNIST dataset used in [6], [7] and [13]. There is a method to encode analog values into spikes: 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 originally discussed SNN model has been tested using multiple different STDP rules, as well as different training set sizes. Hence, there is evidence for

the model's robustness and good performance in a variaty of different situations, due to the competition among the neurons resulting in dissimilar receptive fields [6]. Moreover, the authors of [6] argue that their model is more biologically plausible.

However, there is a shortcoming in terms of biological fidelity of the SNN model presented in [6]: 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.

The [2] model was evaluated on the MNIST dataset. The dataset is a suitable Benchmark for the SNN model, since it provides different difficulty levels of categorization. However, the MNIST dataset less complex than biological vision [2], since projections on the retina greatly vary due to object position, size, pose, illumination condition and background [10].

The addition to accuracy with regard to the classification of the MNIST dataset impulses [2] 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 threshold). 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 Kolmogorov-Smirnov test [12] is a non-parametric test for evaluating whether two samples came from the same distribution function.

The performance of the SNN model is compared for different training set sizes n_{train} [2]. The results suggest that the network is able to generalize well if the training set size is sufficiently large. The optimum training set size $n_{train} = 1000$ did not produce remarkably higher misclassifaction rates than bigger training set sizes. The most frequent misclassification was the digit nine as a zero.

In Figure 2 the authors compare the performance of the SNN model consisting of different numbers of excitatory neurons and different learning rules. The standard deviation of the test accuracy is indicated by the error bars. The black line denotes power-law weight dependence STDP, the red one denotes exponential weight dependence STDP, the green one indicates pre-and-post STDP, wheras the blue line's learning rule is TSTDP. The differences between these learning rules are explained in [6]. The visualization suggests that highest number of excitatory neurons (i.e. 6400) produces the best results for all learning rules. The paper offers possible reasons for the distribution of misclassified digits proposed in Figure 3. The most common misclassification was the digit four as a nine.

Table 1 is a summary of the table visualizing performances of different SNNs in [6]. Rate-based learning methods achieve the best results. Supervised methods imply the usage of a teaching signal. The last learning rule from Table 1 denotes the shape of the STDP time window. The calcium variable from the second learning rule is occasionally used to model the influence of the Ca^{2+} level on activation neurons observed in biology [5]: Presynaptic action potentials release

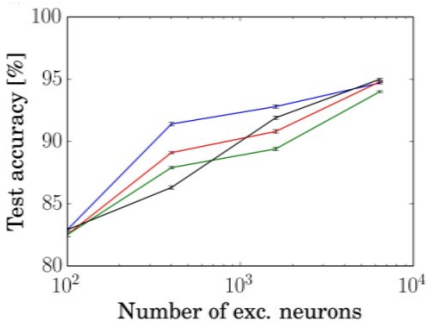

Fig. 2: Test accuracy for different learning rules (colored lines) and different number of excitatory neurons from [6].

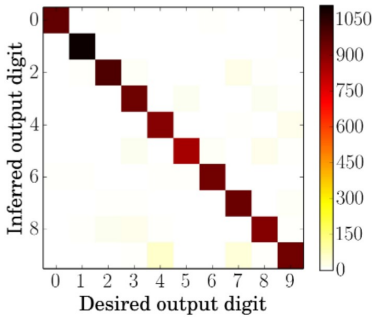

Fig. 3: Confusion matrix presenting the average results over ten presentations of the 1000 MNIST test set digits from [6].

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

Table 1: Summary of methods compared in [6].

neurotransmitters that bind to certain receptores and when the postsynaptic activities provide sufficient constant membrane depolarization the Ca^{2+} level rises [1]. A large Ca^{2+} rise are associated with LTP, whereas modest Ca^{2+} rise may result in LTD [5].

4 Comparison

The authors of [6] present an unsupervised approach to train a SNN model. However, there are other approaches to train SNNs, as well as opinions with regard to the biological plausibility of the methods used. In the following, the SNN model from [6] is compared to other SNN models.

The SNN model from [2] 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 [2] model but also the [6] model use inhibition.

The authors of [2] 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.

The direct input encoding, leakage and threshold optimization (DIET)-SNN model from [14] is an entirely different approach to those discussed before. This alternative approach uses supervised learning. As depicted in Figure 4, an ANN

is trained and thereafter converted into a SNN. The ANN-SNN-conversion is outlined in [14] and [7]. The initialized parameters speed up the subsequent training with spike-based backpropagation. The converted SNN is trained using error-backpropagation.

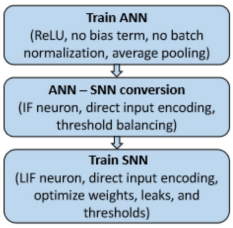

Fig. 4: DIET-SNN training pipeline from [14]

Instead of using a Poisson generator (i.e. 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.

$$u_i^t = \lambda_i u_i^{t-1} + \sum_j w_{ij} o_j^t - v_i o_i^{t-1} \tag{8}$$

The membrane potential u_i^t of a postsynaptic neuron i at timestep t is calculted using Equation 8 from [14]. $\lambda_i \in [0, 1]$ 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.

The model from [10] uses a different architecture than [6]. It is designed to execute object recognition tasks in a biologically plausible manner.

The first difference is the usage of a five layer hierarchical network structure, consisting of alternating so-called simple cells S_1, S_2 and complex cells C_1, C_2. The simple cells employ intensity-to-latency conversion, i.e. the stronger a cell is stimulated the earlier it will fire. In a group of four simple cells, the cell with the earliest firing time inhibits the other cells and thus, creating winner-take-all inhibition. The spike omitted by the winner cell is then propagated asynchronously. It serves as input for the complex cells. Since there are fewer complex cells than simple ones, the complex cells propagate only the maximum, i.e. the earliest spike, of the input spikes.

The second difference is the use of a classifier in the fifth layer to carry out the classification task instead of deciding for the class whose neurons have the highest average firing rate, as suggested by the authors of [6].

Moreover, the model has a multi-scale form. In other words: The original image is scaled to different processing scales (100%, 71%, 50%, 30% and 25%) and parallelly processed until all processing scales are combined in the forth layer.

However, there are also similarities to the [6] model: The C_1 to C_2 synaptic connections are trained using STDP. Yet, the time difference of presynaptic and postsynaptic spikes is only used to determine the sign of the modification of the synaptic weight, but not relevant for the amount of weight change. As mentioned before, both the [10] and the [6] model use inhibition. The [10] model is trained and evaluated on the ETH-80 and 3D-Object datasets [10], which contain objects with large deviations. The data is converted to grayscale values. A representational dissimilarity matrix (RDM) is used to determine whether the quality of the model is good enough to provide representation of objects which have a high inter-category dissimilarity and a low intra-category dissimilarity. The result of the RDM in Figure 5 suggests that the model is able to distinguish between feature values of different categories. The authors include a RDM of a non-SNN model performing considerably worse. The multi-scale SNN is well

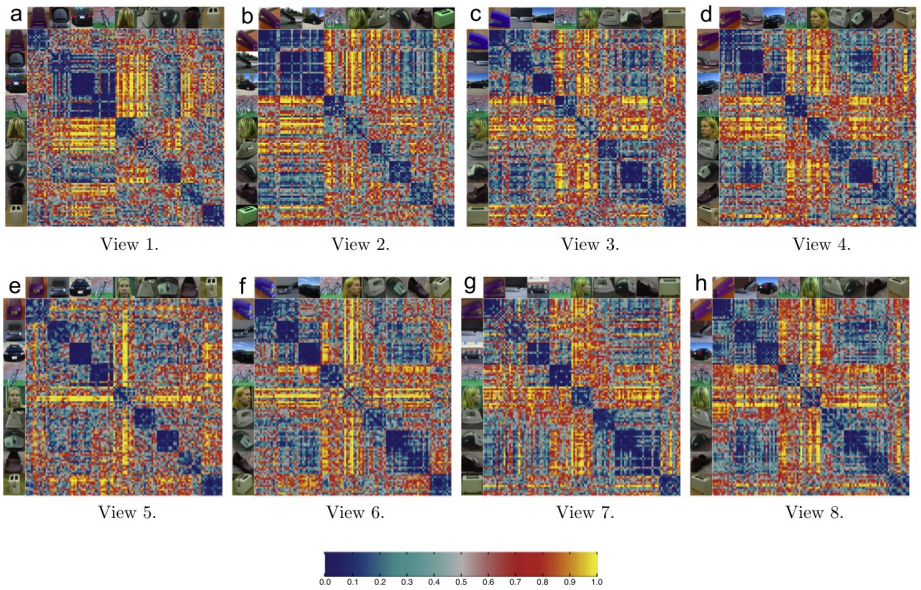

Fig. 5: RDMs of object representations of different image stimuli for different views from [10]. A sample image for each category is placed next to the rows and columns. Intra-class dissimilarity is low, inter-class dissimilarity is high.

suited for object recognition tasks consisting of few classes and many view points [10]. The authors admit that processing time for many classes would greatly increase.

5 Conclusion and Outlook

The study of SNNs, especially with regard to efficient learning procedures poses an interesting insight in the possibilities of modeling systems on the human brain. According to [7], in theory SNNs are as computationally powerful as ANNs, even though it has proven to be difficult to find equivalent solutions in reality. 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 [6] proposes an unsupervised approach with formidable performace in a variety of situations. The authors of [6] and [1] also emphasise the energy efficience of SNNs on neuromorphic hardware. Another reason for the usage of SNNs is their configurability, i.e. high spiking thresholds leading to high accuracy and low thresholds resulting in short latencies, [7].

Besides the unsupervised approach proposed by [6] there is a variety of supervised methods to train a SNN, for instance ANN-SNN conversion from [14]. Even though existing approaches perform well, the authors emphasise shortcomings and potential of further research. For instance, it remains unclear whether a single model can explain STDP at different synapses [5]. Furthermore, biological interplay of factors such as calcium signals in certain cell types remains to analysed to determine its role and thus, being able to model is properly [5]. Some models, incuding the one presented in [11], have yet to determine the model's robustness to noisy input sensor signals. The authors of [11] also point out that SNNs may learn to represent input patterns more compact.

Neural networks (NNs) are simulated using certain software [8]. The python interface Brian omits the requirement of learning a new programming language to simulate SNNs. Vectorisation and the presence of several C routines improve run-time efficiency [8]. However, the authors admit it is not designed for very large scale simulations or large biophysical models.

Since simulation of synaptic plasticity dominates the computing costs of SNNs due to the fact that von Neumann architectures are not built for processing large number of small messages, i.e. spikes, other strategies to improve run-time efficiency rely on among others neuromorphic hardware [15]. This paper neglects the topic neuromorphic hardware. Other work, such as [1], cover several different implementations and offer a discussion about challenges (e.g. memory elements) and adchievements (e.g. successful synaptic rule implementations in hardware) in this area. Yet another problem hindering the application of SNNs in real-world problems is accessibility of certain elements, such as individual synapses and neurons, to read and (re)configure the network topology [9]. The hardware seems to be in need of more improvements to be applicable to real-world problems.

References

1. Azghadi, M.R., Iannella, N., Al-Sarawi, S.F., Indiveri, G., Abbott, D.: Spike-based synaptic plasticity in silicon: Design, implementation, application, and challenges. IEEE **102**, 717–737 (2014)

2. 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)
3. 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)
4. Brette, R.: Philosophy of the spike: Rate-based vs. spike-based theories of the brain. Frontiers in Systems Neuroscience **9**, 1–14 (2015)
5. Caporale, N., Dan, Y.: Spike timing-dependent plasticity: A hebbian learning rule. Annual Review of Neuroscience **31**, 25–46 (2008)
6. Cook, M., Diehl, P.U.: Unsupervised learning of digit recognition using spike-timing-dependent plasticity. Frontiers in Computational Neuroscience **9**, 1–8 (2015)
7. Diehl, P.U., Neil, D., Binas, J., Cook, M., Liu, S.C., Pfeiffer, M.: Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. International joint conference on neural networks (IJCNN). IEEE pp. 1–8 (2015)
8. Goodman, D., Brette, R.: Brian: a simulator for spiking neural networks in python. Frontiers in Neuroscience **2**, 1–10 (2008)
9. Indiveri, G., Chicca, E., Douglas, R.: A vlsi array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. IEEE Transactions on Neural Networks **17**, 211–221 (2006)
10. Kheradpisheh, S.R., Ganjtabesh, M., Masquelier, T.: Bio-inspired unsupervised learning of visual features leads to robust invariant object recognition. Neurocomputing **205**, 383–392 (2016)
11. 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)
12. Lopes, P.H.C., Reid, I., Hobson, P.R.: The two-dimensional kolmogorov-smirnov test. XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research pp. 1–12 (2007)
13. O'Connor, P., Neil, D., Liu, S., Delbruck, T., Pfeiffer, M.: Real-time classification and sensor fusion with a spiking deep belief network. frontiers in neuroscience **7**, 1–13 (2013)
14. 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)
15. amd X. Lagorce, F.G., Stromatias, R., Pfeiffer, M., Plana, L.A., Furber, S.B., Benosman, R.B.: A framework for plasticity implementation on the spinnaker neural architecture. Frontiers in Neuroscience **8**, 1–20 (2015)