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 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 to 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 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 [6]. The input of SNNs are streams of events which differs from the singular presentation of ANNs inputs [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 presented in the following 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 [14,13]. 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 is outlined. Section 3 covers context-specific terms, as well as the approach itself. The network architecture is described in Section 4. In Section 5 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 5 are described and compared in Section 6. The paper concludes with an outlook in Section 7.

2   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.

3   Methods

Firstly, technical terms specific to the domain are defined and briefly discussed. Afterwards, the used methods are described in detail. Concepts used in the approach by Cook and Diehl [6] include *leaky-integrate-and-fire* (LIF) neurons, lateral inhibition, synaptic plasticity and homoeostasis, which are outlined in this section.

3.1   Terms

Spike trains are defined as multiple spikes. They are binary signals distributed over time. A '1' indicates a spike, whereas '0' indicates inactivity [20]. The number of timesteps determines the discretization error. A timestep is dependent on the hardware and its associated number of computations performable. In simulations, a timestep is the time necessary to perform a forward pass, i.e. the flow of information from the input layer to the output layer [20].

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 the spike-based model is preferable 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 [20] and [7]. Afterwards, 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].

3.2   SNN's methods

Input encoding SNNs transmit information through spikes and thus, analogue values have to be encoded into spikes [20]. There are different types of encodings based on certain beliefs, for instance, those outlined below.

$60,000$ training examples and $10,000$ test examples of 28×28 pixel images of the digits 0 to 9 compose the MNIST dataset used in [6,7,18]. There is a method to encode analogue values into spikes: Inputs are Poisson 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 the neurons fire in the desired fashion.

Poisson model of spike generation The timing of successive action potentials in biology can be modelled by stochastic processes, due to the observed irregularity of interval lengths between spikes [11]. The hypothesis that the generation of an individual spike is independent of the generation of others is necessary to use Poisson processes to model spike trains. A Poisson spike generator generates spikes according to a Poisson process. Different methods to generate spikes, which are either assigned to continuous time values or to discrete time bins, are outlined in [11].

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 repetitive postsynaptic spikes 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]. The term *synaptic plasticity rules* is used for the different mathematical formulae realised in activity-dependent modification of synaptic weights. There is a great variety of synaptic plasticity rules, ranging 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. $\Delta t = t_{post} - t_{pre}$ is the time difference between the presynaptic spike t_{pre} and the postsynaptic spike t_{post}. If a postsynaptic spike arrives within the 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^-.

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

Figure 1 visualizes the concept of the weight modification dependence on the temporal relationship of the presynaptic and the postsynaptic spike.
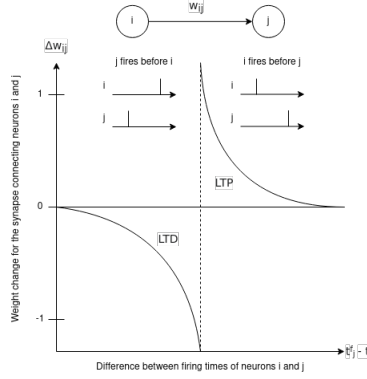

Fig. 1: Weight modification depending on the time difference $\Delta t = t_j^f - t_i^f$ between postsynaptic and presynaptic spike from [21]. The weight change Δw is positive if the presynaptic spike precedes the postsynaptic one, i.e. $\Delta t > 0$, analogously for $\Delta t < 0$. The size of Δw depends on the size of Δt.

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 [1,19].

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 variable is used to model the influence of the Ca^{2+} level on activation neurons observed in biology [5]: Presynaptic action potentials release neurotransmitters that bind to certain receptors and when the postsynaptic activities provide sufficient constant membrane depolarization, the Ca^{2+} level rises [1]. A large Ca^{2+} rise is associated with LTP,

whereas modest Ca^{2+} rise may result in LTD [5]. The calcium concentration $C(t)$ modification with regard to the temporal interplay of decay and incrementation is outlined in [22]. Whenever a presynaptic spike arrives, either potentiation of amount a takes place, 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^1 analogously displayed in Equation 2 from [1] occurs.

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

If the conditions from Equation 2 are not satisfied or no spike arrives, the synaptic weight drifts towards either high or low synaptic weight asymptotes 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 [1,22].

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 [2,13], 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 [14]. The goal of STDP is to strengthen synapses of pre- and postsynaptic neuron pairs, whose postsynaptic neuron reacts immediately after the presynaptic neuron fires [14]. If the postsynaptic neuron fires before the presynaptic one, the spike has another origin and thus, the synapse is weakened to disconnect the neurons. STDP can be considered the transposition of the Hebbian learning rule in a temporal context [16].

$$\Delta w = \eta(x_{pre} - x_{tar})(w_{max} - w)^\mu \tag{4}$$

Equation 4 from [6] calculates the weight change after a postsynaptic spike arrives (i.e. the synapse's 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,

---

[1] The origin of a and b is not discussed in literature [1].

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 [22]: 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 [6,17]. Some STDP models rely on a teaching signal, which provides the right response [2]. Teaching signals are generated by Poisson 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 focuses on the target object features, generally not learning the background, since they are usually too different to converge on them [13,16]. Hence, STDP allegedly extracts informative and diagnostic features.

Neuron model In reality, a neuron fires if the membrane's potential V 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 refractory period t_{ref} [18], the neuron cannot spike again [6]. The model at hand displays similar behaviour as displayed in Figure 2.

$$\tau \frac{dV}{dt} = (E_{rest} - V) + g_e(E_{exc} - V) + g_i(E_{inh} - V) \qquad (5)$$

Equation 5 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 synapse's conductance g_e/g_i (e excitatory, i inhibitory) models 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.

$$\tau_{g_e} \frac{dg_e}{dt} = -g_e \qquad (6)$$

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

LIF neurons The LIF neuron is a simple, yet biologically plausible model of a spiking neuron [18]. As depicted in Equation 5, 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.
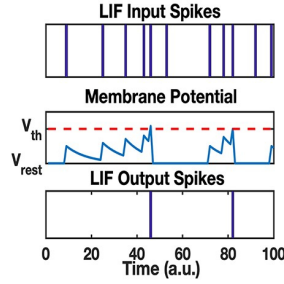
Fig. 2: The LIF neuron firing mechanism from [10]. LIF neurons integrate incoming spikes and decay otherwise. If the membrane potential V crosses the threshold V_{thres}, the neuron fires and thus, enters a refractory period at the resting membrane potential.

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 decays exponentially 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 similar to an input, homoeostasis guarantees similar 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.

Training In order to allow all variables to decay there is a 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.

4   Network architecture

The ANN has an input layer and a preprocessing layer [6]. The input layer consists of 28×28 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. A 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 3. This structure creates lateral inhibition and creates competition among the excitatory neurons.
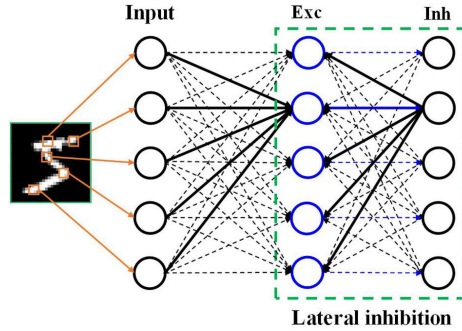

Fig. 3: Architecture of a SNN from [9].

5   Performance

The 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 variety 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 than other SNN models.

However, there is a shortcoming in terms of biological fidelity of the SNN model presented in [6]: Since neurons that are not in their refractory 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 refractory period.

The approach from [2] was also 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 is less complex than biological vision [2], since projections on the retina greatly vary due to object position, size, pose, illumination condition and background [13].

zitierung verwirrt im satzbau, es war wohl the stdb_like model und dann erst cite gemeint

In addition to the accuracy with regard to the classification of the MNIST dataset impulses, reaction time (RT) distributions are included [2]. RT is defined as the time between the presentation of a stimulus and the response (i.e. the first pool to reach the decision 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 [2] with the Kolmogorov-Smirnov test, indicating that the correctly classified and misclassified RT distributions are significantly different, whereas the distributions for stimuli from the training and test set were not. The Kolmogorov-Smirnov test is a non-parametric test for evaluating whether two samples came from the same distribution function [15].

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 misclassification rates than bigger training set sizes [2].

In Figure 4 Cook and Diehl [6] 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, whereas the blue line's learning rule is TSTDP. The differences between these learning rules are explained in the work of Cook and Diehl [6]. The visualization suggests that the highest number of excitatory neurons (i.e. 6400) produces the best results for all learning rules. The confusion matrix of Figure 5 serves as an error analysis tool [6]. The authors propose possible reasons for the distribution of misclassified digits. The most common misclassification was the digit four as a nine [6].

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

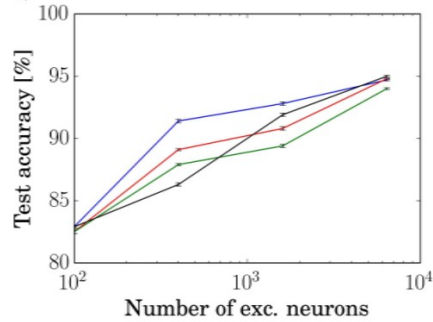| Training type	(Un-) Supervised	Learning rule	Performance range
Rate-based	Supervised	different	90-99 %
Spike-based	Supervised	different incl. calcium variable	91-96 %
Spike-based	Unsupervised	rectangular/ exponential STDP	93-95 %

Fig. 4: Test accuracy for different learn-
ing rules (colored lines) and different
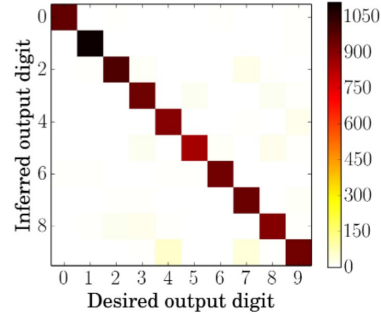numbers of excitatory neurons from [6].

Fig. 5: Confusion matrix pre-
senting the average results
over ten presentations of 1000
MNIST test set digits from [6].

Table 1 is a summary of the table visualizing performances of different SNNs.
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.

6   Comparison

Cook and Diehl present an unsupervised approach to train a SNN model [6].
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 Cook and Diehl [6] is compared to other SNN models.

The presented SNN model determines its output by choosing the class of the
first neuron pool to reach the decision threshold with its accumulated sensory
evidence [2]. The authors claim that the original method of using a majority
vote of single threshold-based neuron activity is less biologically plausible.

Not only the model of Beyel et al. [2] but also the model of Cook and Diehl
[6] use inhibition.

The Beyel et al. point out that the classifier neurons will not recognize atyp-
ical digits (i.e. their stimulus) [2]. Therefore, the ability to generalize on new
datasets depends on the inputs' similarity to the learned patterns.

TODO: compare with normalen NN

The model presented by both Kheradpisheh et al. [13] and Masquelier and
Thorpe [16] is different from the one from Cook and Diehl [6]. It is designed to
execute object recognition tasks in a biologically plausible manner.

The first difference is the architecture [13,16].

The second difference is the use of a classifier to carry out the classification
task instead of deciding on the class whose neurons have the highest average
firing rate, as suggested by Cook and Diehl [6].

Moreover, the model has a multi-scale form [13]. 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 one of the layers.

However, there are also similarities to the model of Cook and Diehl [6]: The synaptic connections are trained using STDP [13,16]. Yet, the time difference between the presynaptic and postsynaptic spike is only used to determine the sign of the modification of the synaptic weight but is not relevant for the amount of weight change [16]. As mentioned before, both the model from Kheradpisheh et al. [13] and the model from Cook and Diehl [6] use inhibition and thus, create competition and consequently ensure all neurons learn a distinct pattern to cover the whole variability of inputs [16]. The model from Kheradpisheh et al. [13] is trained and evaluated on the ETH-80 and 3D-Object datasets [13], 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 a representation of objects which have a high inter-category dissimilarity and a low intra-category dissimilarity. The result of the RDM in Figure 6 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 model becomes selective to
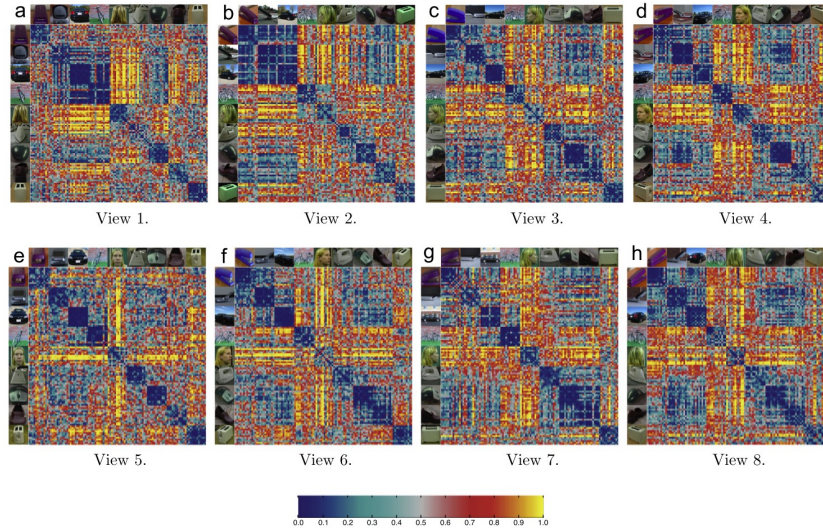

Fig. 6: RDMs of object representations of different image stimuli for different views from [13]. A sample image for each category is placed next to the rows and columns. Intra-class dissimilarity is low, and inter-class dissimilarity is high.

the prominently present patterns in the input and generates fast responses [16]. The multi-scale SNN is well suited for object recognition tasks consisting of few

classes and many viewpoints [13]. The authors admit that processing time for many classes would greatly increase.

Pfister and Gerstner [19] claim that classical pair based STDP models are not able to explain synaptic changes caused by triplets or quadruplets of spikes.

7   Conclusion and Outlook

The study of SNNs, especially with regard to efficient learning procedures poses an interesting insight into the possibilities of modelling systems on the human brain. According to Diehl et al. [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 Cook and Diehl [6] proposes an unsupervised approach with formidable performance in a variety of situations. Cook and Diehl [6] and Azghadi et al. [1] also emphasize the energy efficiency 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 Cook and Diehl [6], there is a variety of supervised methods to train a SNN, for instance ANN-SNN conversion from Rathi and Roy [20]. Even though existing approaches perform well, the authors emphasize shortcomings and the potential for further research. For instance, it remains unclear whether a single model can explain STDP at different synapses [5]. Furthermore, the biological interplay of factors such as calcium signals in certain cell types remains an ongoing field of research in order to improve existing models [5]. Some models, including the one presented by Kim et al. [14], have yet to determine the model's robustness to noisy input sensor signals. Kim et al. [14] also point out that SNNs may learn to represent input patterns more compactly.

ANNs are simulated using certain software. The Python interface Brian omits the requirement of learning a new programming language to simulate SNNs [8]. Vectorization 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 the simulation of synaptic plasticity dominates the computing costs of SNNs due to the fact that von Neumann architectures are not built for processing large numbers of small messages, i.e. spikes, other strategies to improve run-time efficiency rely on (among others) neuromorphic hardware [22]. This paper neglects the topic of neuromorphic hardware. Other work, such as the work by Azghadi et al. [1], covers several different implementations and offers a discussion about challenges (e.g. memory elements) and achievements (e.g. successful implementation of synaptic rules in hardware) in this area. Yet another problem hindering the application of SNNs in real-world problems is the accessibility of certain elements, such as individual synapses and neurons, to read and

(re)configure the network topology [12]. The hardware seems to require 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. Guo, W., Fouda, M.E., Eltawil, A.M., Salama, K.N.: Neural coding in spiking neural networks: A comparative study for robust neuromorphic systems. Frontiers in Neuroscience **15**, 1–21 (2021)
10. Guo, Y., Wu, H., Gao, B., Qian, H.: Unsupervised learning on resistive memory array based spiking neural networks. Frontiers in Neuroscience **13**, 1–16 (2019)
11. Heeger, D.: Poisson model of spike generation. New York University **1**, 1–13 (2000)
12. 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)
13. 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)
14. 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)
15. 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)
16. Masquelier, T., Thorpe, S.J.: Unsupervised learning of visual features through spike timing-dependent plasticity. PLoS Computational Biology **3**, 247–257 (2007)
17. Morrison, A., Aertsen, A., Diesmann, M.: Spike-timing-dependent plasticity in balanced random networks. Neural Computation **19**, 1437–1467 (2007)

18. 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)
19. Pfister, J.P., Gerstner, W.: Triplets of spikes in a model of spike timing-dependent plasticity. The Journal of Neuroscience pp. 9673–9682 (2006)
20. 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)
21. Shears, O., Yazdani, A.H.: Spiking neural networks for image classification, `https://oshears.github.io/adv-ml-2020-snn-project/pages/motivation.html`
22. 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)