



A biologically plausible supervised learning method for spiking neural networks using the symmetric STDP rule

Yunzhe Hao^{a,b,1}, Xuhui Huang^{a,1,*}, Meng Dong^a, Bo Xu^{a,b,c,**}

^a Research Center for Brain-inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, 100190 Beijing, China

^b University of Chinese Academy of Sciences, 100049 Beijing, China

^c CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, 100190 Beijing, China

ARTICLE INFO

Article history:

Available online 27 September 2019

Keywords:

Spiking neural networks
Dopamine-modulated spike-timing
dependent plasticity
Pattern recognition
Supervised learning
Biologically plausibility

ABSTRACT

Spiking neural networks (SNNs) possess energy-efficient potential due to event-based computation. However, supervised training of SNNs remains a challenge as spike activities are non-differentiable. Previous SNNs training methods can be generally categorized into two basic classes, i.e., backpropagation-like training methods and plasticity-based learning methods. The former methods are dependent on energy-inefficient real-valued computation and non-local transmission, as also required in artificial neural networks (ANNs), whereas the latter are either considered to be biologically implausible or exhibit poor performance. Hence, biologically plausible (bio-plausible) high-performance supervised learning (SL) methods for SNNs remain deficient. In this paper, we proposed a novel bio-plausible SNN model for SL based on the symmetric spike-timing dependent plasticity (sym-STDP) rule found in neuroscience. By combining the sym-STDP rule with bio-plausible synaptic scaling and intrinsic plasticity of the dynamic threshold, our SNN model implemented SL well and achieved good performance in the benchmark recognition task (MNIST dataset). To reveal the underlying mechanism of our SL model, we visualized both layer-based activities and synaptic weights using the t-distributed stochastic neighbor embedding (t-SNE) method after training and found that they were well clustered, thereby demonstrating excellent classification ability. Furthermore, to verify the robustness of our model, we trained it on another more realistic dataset (Fashion-MNIST), which also showed good performance. As the learning rules were bio-plausible and based purely on local spike events, our model could be easily applied to neuromorphic hardware for online training and may be helpful for understanding SL information processing at the synaptic level in biological neural systems.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Due to the emergence of deep learning technology and rapid growth of high-performance computing, artificial neural networks (ANNs) have achieved various breakthroughs in machine learning tasks (LeCun, Bengio, & Hinton, 2015; Schmidhuber, 2015). However, ANN training can be energy inefficient as communication among neurons is generally based on real-valued activities and many real-valued weights and error signals need to be transmitted in the basic error backpropagation (BP) training algorithm (Rumelhart, Hinton, Williams, et al., 1988). This can consume considerable energy for computations on central processing units (CPUs) and information transportation between

CPUs and random-access memory (RAM) in traditional high-performance computers. As such, ANNs are potentially highly energy consumptive. However, information processing in the human brain is very different to that in ANNs. Neurons in biological systems communicate with each other via spikes or pulses (i.e., event-based communication), which allow for the asynchronous update of states only when a spike is incoming. This has the potential advantage of energy efficiency compared to the synchronous update of real-valued states at each step in ANNs. In addition, learning rules for modifying synaptic weights of realistic neurons can also be based on local spike events, e.g., spike-timing dependent plasticity (STDP) (Bi & Poo, 1998; Caporale & Dan, 2008; Song, Miller, & Abbott, 2000), where the weight of a synapse is changed based on the spike activities of its pre- and post-synaptic neurons. This local event-based learning requires no additional energy for non-local transmission, which is required during the ANN training process. Therefore, spiking neural networks (SNNs), which can closely mimic the local event-based computations and learning of biological neurons, may be more energy efficient than ANNs, especially if

* Corresponding author.

** Corresponding author at: Research Center for Brain-inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, 100190 Beijing, China.

E-mail addresses: xuhui.huang@ia.ac.cn (X. Huang), xubo@ia.ac.cn (B. Xu).

¹ These two authors contributed equally to this work.

SNNs are implemented onto neuromorphic platforms. Moreover, SNNs exhibit the natural ability of spatiotemporal coding of an input, and thus hold the potential advantage of efficient coding through sparse activities, particularly for continuous spatiotemporal inputs (e.g., optic flow), whereas ANNs require specially designed architectures to deal with temporal input (e.g., long short-term memory, LSTM) (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2016; Hochreiter & Schmidhuber, 1997) as their individual neurons generally have no intrinsic temporal characteristics and only generate output responses based on current inputs at each step. Thus, for these reasons, SNNs are considered as third generation neural network models and have attracted growing interest for exploring their functions in real-world tasks (Maass & Markram, 2004; Tavanaei, Ghodrati, Kheradpisheh, Masquelier, & Maida, 2018).

Unlike that of ANNs, SNN training is highly challenging due to the non-differentiable properties of the spike-type activity. Hence, the development of an efficient training algorithm for SNNs is of considerable importance. Much effort has been expended in the past two decades on this issue (Tavanaei et al., 2018), with the subsequently developed approaches generally characterized as indirect supervised learning (SL), direct SL, or plasticity-based training (Tavanaei et al., 2018; Wu, Deng, Li, Zhu, & Shi, 2017). For the indirect SL method, ANNs are first trained and then mapped to equivalent SNNs by different conversion algorithms that transform real-valued computing into spike-based computing (Cao, Chen, & Khosla, 2015; Diehl et al., 2015; Diehl, Zarella, Cassidy, Pedroni, & Neftci, 2016; Esser, Appuswamy, Merolla, Arthur, & Modha, 2015; Esser et al., 2016; Hu, Tang, Wang, & Pan, 2018; Hunsberger & Eliasmith, 2015; Neil & Liu, 2016); however, this method does not incorporate SNN learning and therefore provides no heuristic information on how to train a SNN. The direct SL method is based on the BP algorithm (Lee, Delbruck, & Pfeiffer, 2016; Samadi, Lillicrap, & Tweed, 2017; Tavanaei & Maida, 2017; Wu et al., 2017; Zhang, Zeng, Zhao, & Shi, 2018), e.g., using membrane potentials as continuous variables for calculating errors in BP (Lee et al., 2016; Zhang et al., 2018) or using continuous activity function to approximate neuronal spike activity and obtain differentiable activity for the BP algorithm (Tavanaei & Maida, 2017; Wu et al., 2017). However, such research must still perform numerous real-valued computations and non-local communications during the training process; thus, BP-based methods are as potentially energy inefficient as ANNs and also lack bio-plausibility. For plasticity-based training, synaptic plasticity rules (e.g., STDP) are used to extract features for pattern recognition in an unsupervised learning (USL) way (Diehl & Cook, 2015). Due to the nature of spontaneous unsupervised clustering of synaptic plasticity, this method requires an additional supervised module for recognition tasks. Three supervised modules have been used in previous studies: (1) a classifier (e.g., support vector machine, SVM) (Kheradpisheh, Ganjtabesh, Thorpe, & Masquelier, 2018), (2) a label statistical method outside the network (Diehl & Cook, 2015), and (3) an additional supervised layer (Beyeler, Dutt, & Krichmar, 2013; Hu, Wang, & Hu, 2017; Mozafari, Ganjtabesh, Nowzari-Dalini, Thorpe, & Masquelier, 2018; Shrestha, Ahmed, Wang, & Qiu, 2017). In our opinion, a neural network model with biological plausibility must meet the following basic characteristics. Firstly, the single neuron model must integrate temporal inputs and generate pulses or spikes as outputs. Secondly, the computation processes of training and inference must be completely spike-based. Finally, all learning rules must be based on experiments, and should not be violated (obviously contrary to) by experiments or artificially designed. The first two supervised modules are bio-implausible due to the need of computation outside the SNNs (Diehl & Cook, 2015; Kheradpisheh et al., 2018). The last supervised module has the

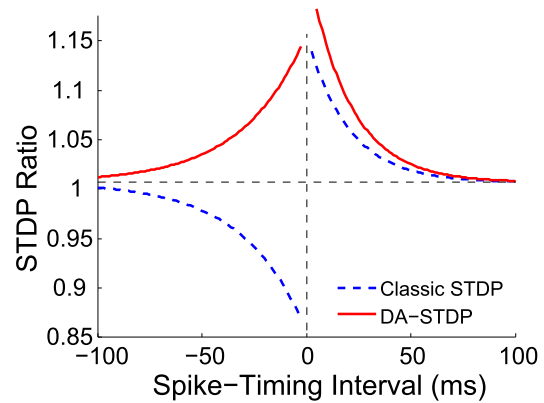


Fig. 1. Schematic of the classic STDP (Bi & Poo, 1998) and DA-STDP (Brzosko et al., 2015; Zhang et al., 2009).

potential of bio-plausibility, but existing supervised SNN models have either adopted artificially modified STDP rules (Beyeler et al., 2013; Hu et al., 2017; Mozafari et al., 2018; Shrestha et al., 2017) or exhibited poor performance (Beyeler et al., 2013). Currently, therefore, truly bio-plausible SNN models that can accomplish SL and achieve high-performance pattern recognition are lacking. Moreover, although great progress has been made in understanding the physiological mechanisms for the modification of synapses at the microscopic level (Bi & Poo, 1998; Caporale & Dan, 2008), how teacher learning at the macroscopic behavioral level is realized by changes in synapses at the microscopic level, i.e., the mechanism of SL processing in the brain, is still far from clear.

In this study, we proposed a novel bio-plausible SL method for training SNNs based on biological plasticity rules. We introduced the dopamine-modulated STDP (DA-STDP) rule, a new type of symmetric STDP (sym-STDP), for pattern recognition. The DA-STDP rule has been observed in several different experiments in the hippocampus and prefrontal cortex (Brzosko, Schultz, & Paulsen, 2015; Ruan, Saur, & Yao, 2014; Zhang, Lau, Bi, & Stevens, 2009), where the modification of synaptic weight is always incremental if the interval between the pre- and post-synaptic spike activities is within a narrow time-window when dopamine (DA) is present. The differences between the sym- or DA-STDP and classic STDP rules are shown in Fig. 1 (Bi & Poo, 1998; Brzosko et al., 2015; Zhang et al., 2009). While the sym-STDP rule has been used previously (Masuda & Kori, 2007; Mishra, Kim, Guzman, & Jonas, 2016; Serrano-Gotarredona, Masquelier, Prodromakis, Indiveri, & Linares-Barranco, 2013; Tanaka, Morie, & Aihara, 2009), this is the first time it has been applied for SL. In our proposed model, a three-layer feedforward SNN was trained by DA-STDP combined with synaptic scaling (Effenberger, Jost, & Levina, 2015; Turrigiano, 2008; Turrigiano, Leslie, Desai, Rutherford, & Nelson, 1998) and dynamic threshold, which are two homeostatic plasticity mechanisms for stabilizing and specializing the network response under supervised signals. Two different training methods were used in our SNN model, i.e., training two-layer input synaptic weights simultaneously and training the SNN layer-by-layer. Our model was tested in the benchmark handwritten digit recognition task (MNIST dataset) and achieved high performance under the two training methods. We also evaluated the model using the MNIST-like fashion product dataset (Fashion-MNIST), which also showed good classification performance. These results thus highlighted the robustness of our model.

2. Network architecture and neuronal dynamics

We constructed a three-layer feedforward spiking neural network for SL, which included an input layer, hidden layer, and SL

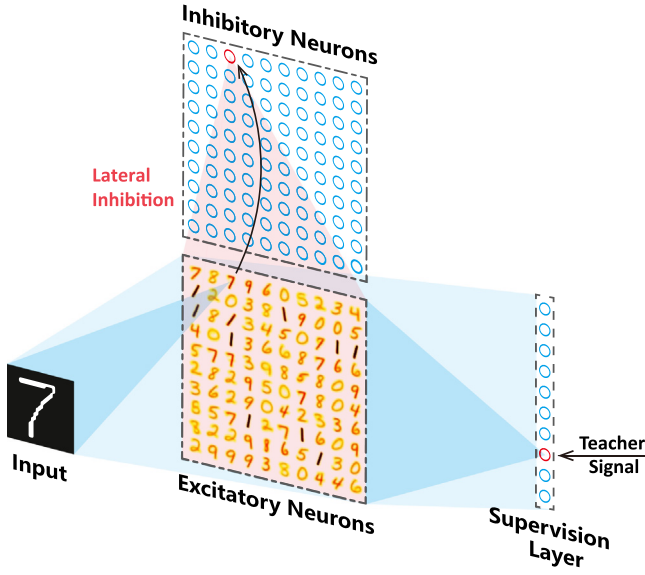


Fig. 2. Network structure.

layer (Fig. 2). The structure of the first two layers was inspired by the USL model of Diehl and Cook (2015). Input patterns were coded as Poisson spike processes with firing rates proportional to the intensities of the corresponding pixels. The Poisson spike trains were then fed to the excitatory neurons in the hidden layer with all-to-all connections. The dark blue shaded area in Fig. 2 shows the input connection to a specific neuron. The connection from the excitatory to inhibitory neurons was one-to-one. An inhibitory neuron only received input from the corresponding excitatory neuron at the same position in the map and inhibited the remaining excitatory neurons. All excitatory neurons were fully connected to the SL layer. In the SL layer, neurons fired with two different modes during the training and testing processes. During the SL training period, the label information of the current input pattern was converted to a teacher signal in a one-hot coding scheme by the 10 SL neurons. Only one SL neuron was pushed to fire as a Poisson spike process, with the remaining SL neurons maintained in the resting state. In the testing mode, all SL neurons fired according to inputs from the hidden layer.

We used the leaky integrate-and-fire (LIF) model and set the parameters within bio-plausible ranges. The resting membrane potential E_{rest} was set at -65 mV and the equilibrium potentials of the excitatory synapses E_E and inhibitory synapses E_I were set to 0 mV and -100 mV, respectively. The time constant of membrane potential damping τ was equal to 100 ms. The membrane potential V dynamics of the neurons can be described by Diehl and Cook (2015) and Vogels, Sprekeler, Zenke, Clopath, and Gerstner (2011):

$$\tau \frac{dV}{dt} = (E_{\text{rest}} - V) + g_E(E_E - V) + g_I(E_I - V) \quad (1)$$

where g_E and g_I are the total excitatory and total inhibitory conductances, respectively.

Both g_E and g_I have a similar dynamic equation. They are dependent on the number of excitatory and inhibitory synapses (N_E and N_I) and input synapses (in-synapses) weights (w_i^E and w_i^I) of excitatory and inhibitory synapses, respectively. Both of the time constants of synapse conductance damping τ_{g_E} and τ_{g_I} were equal to 1 ms. Thus, g_E and g_I in Eq. (1) can be described by the following equations (Diehl & Cook, 2015; Vogels et al., 2011):

$$\tau_{g_E} \frac{dg_E}{dt} = -g_E + \sum_{i=1}^{N_E} \sum_k w_i^E \delta(t - t_i^k) \quad (2)$$

$$\tau_{g_I} \frac{dg_I}{dt} = -g_I + \sum_{i=1}^{N_I} \sum_k w_i^I \delta(t - t_i^k) \quad (3)$$

where t_i^k is the k th spike time from the i th neuron.

In our neuron model, once the membrane potential exceeds the threshold voltage, the membrane potential is returned to the reset voltage E_{reset} (-65 mV), and the neuron will not fire again in the refractory time $T_{\text{refractory}}$ (2 ms). The firing threshold is not static, with a dynamic threshold (i.e., homeostatic plasticity) mechanism instead adopted. The dynamic threshold is the intrinsic plasticity of a neuron, and is found in different neural systems (Cooper & Bear, 2012; Pozo & Goda, 2010; Sun, 2009; Yeung, Shouval, Blais, & Cooper, 2004; Zhang & Linden, 2003). Here, it was introduced to generate a specific response to a class of input patterns for each excitatory neuron in the hidden layer (Diehl & Cook, 2015; Pozo & Goda, 2010; Zhang & Linden, 2003), otherwise single neurons can dominate the response pattern due to their enlarged input synaptic (in-synaptic) weights and lateral inhibition. The membrane potential threshold V^{th} was composed of two elements (Eq. (4)), i.e., the constant ($V_{\text{const}}^{\text{th}} = -72$ mV) and dynamic variable (θ) parts. θ increases slightly when the neuron fires, otherwise it decays exponentially. As a neuron will not (or barely) fire when θ is too large, which can negatively impact model performance, we adopted a dynamical increment to slow θ growth gradually. Therefore, V^{th} can be described as:

$$V^{\text{th}} = V_{\text{const}}^{\text{th}} + \theta \quad (4)$$

$$\tau_{\theta} \frac{d\theta}{dt} = -\theta + \frac{\theta_{\text{initial}}}{|2\theta - \theta_{\text{initial}}|} \sum_k \alpha \delta(t - t_k) \quad (5)$$

where τ_{θ} is the time constant of θ , with the initial value of θ (θ_{initial}) set to 20 mV, α is the maximum value of the increment and $\frac{\theta_{\text{initial}}}{|2\theta - \theta_{\text{initial}}|}$ is its dynamical scaling factor, and t_k is the k th firing time.

Synaptic weights were modified according to the two biological plasticity rules DA-STDP and synaptic scaling. Dopamine is an important neuromodulator and plays a critical role in learning and memory processes (Tritsch & Sabatini, 2012). Here, inspired by the DA-STDP found in different brain areas, such as the hippocampus and prefrontal cortex (Brzosko et al., 2015; Ruan et al., 2014; Zhang et al., 2009) (Fig. 1), we hypothesized that DA can modulate changes in synaptic weights according to the DA-STDP rule during the SL process. The weights increment ΔW under the phenomenological model of DA-STDP can be expressed as Zhang et al. (2009) and Brzosko et al. (2015):

$$\Delta W = \begin{cases} A_+ \exp(-\frac{\Delta t}{\tau_+}), & \Delta t > 0 \\ A_- \exp(\frac{\Delta t}{\tau_-}), & \Delta t < 0 \end{cases} \quad (6)$$

where Δt is the time difference between the pre- and post-synaptic spikes, τ_+ and τ_- are the time constants of positive and negative phases for Δt , respectively, and A_+ and A_- are learning rates.

As DA-STDP can only increase synapse strength, the synaptic scaling plasticity rule was introduced to generate a competition mechanism among all in-synapses of a neuron in the hidden layer (only for excitatory neurons) and SL layer. Synaptic scaling is a homeostatic plasticity mechanism observed in many experiments (Davis, 2006; Pozo & Goda, 2010; Turrigiano, 2008), especially in visual systems (Hengen, Lambo, Van Hooser, Katz, & Turrigiano, 2013; Keck et al., 2013; Maffei & Turrigiano, 2008) and the neocortex (Turrigiano et al., 1998). Here, synaptic scaling was conducted after the pattern was trained and synapse strength was normalized according to the following equation (Diehl & Cook, 2015):

$$w' = w \frac{\beta N_{\text{in}}}{\sum w} \quad (7)$$

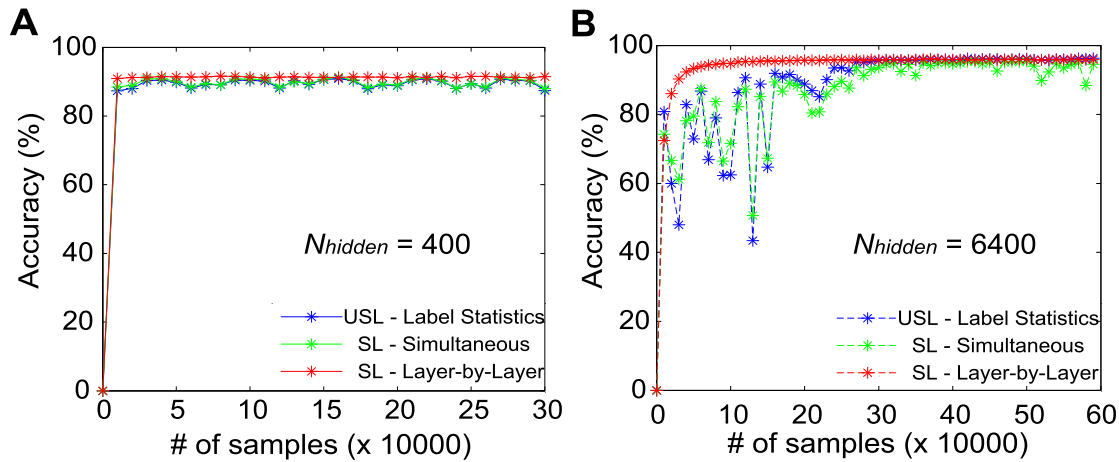


Fig. 3. Convergence property of networks with sizes $N_{\text{hidden}} = 400$ (A) and 6400 (B). ‘Layer-by-Layer’ curve in each figure represents SL performance under layer-by-layer training using STDP to train in-synaptic weights of the hidden layer in an unsupervised manner and then train out-synaptic weights in a supervised manner.

where N_{in} is the number of all in-synapses of a single neuron and β with $\beta \in (0, 1)$ is the scaling factor.

3. Recognition performance for the MNIST task

Our SNN model was trained on the MNIST dataset (training set: 60 000 samples; test dataset: 10 000 samples) (<http://yann.lecun.com/exdb/mnist/>) using two training methods, i.e., simultaneous and layer-by-layer training. For simultaneous training, the in-synapses of the hidden and SL layers were updated simultaneously during the training process, whereas for layer-by-layer training, the hidden layer was trained first, then the SL layer was trained with all in-synaptic weights of the hidden layer fixed. In the inference stage, the most active SL layer neuron was regarded as the inference label of the input sample. Error rate on the test dataset was used to evaluate performance. No data preprocessing was conducted, and we used a SNN simulator (GeNN) (Yavuz, Turner, & Nowotny, 2016) to simulate all experiments.

For the model parameters, we set the simulation time step to 0.5 ms and the presentation time of a single input sample to 350 ms, followed by a resting period of 150 ms. Both τ_{θ} and α were tuned according to network size. The in-synaptic and output synaptic (out-synaptic) weights of the excitatory neurons in the hidden layer were in the ranges of $[0, 1]$ and $[0, 8]$, respectively. All initial weights were set to corresponding maximum weights multiplied by uniform distributed values in the range $[0, 0.3]$. The factor for synaptic scaling β was set to 0.1. The firing rates of the input neurons were proportional to the intensity of the pixels of the MNIST images (Diehl & Cook, 2015). We set maximum rates to 63.75 Hz after dividing the maximum pixel intensity of 255 by 4. When less than five spikes were found in the excitatory neurons of the hidden layer during 350 ms, the maximum input firing rates were increased by 32 Hz. The firing rates of the SL layer neurons were 200 Hz or 0 Hz according to the one-hot code in the SL training period.

To demonstrate the power of the proposed SL method for different network sizes, we compared the results of our SL algorithm to the ‘Label Statistics’ algorithm used in previous study (Diehl & Cook, 2015). In this algorithm, an additional module first calculates the most likely representation of a neuron for a special class of input patterns, labels the neuron to the class during the training process, and uses the maximum mean firing rates among all classes of labeled neurons for inference in the test process.

We trained the model with different epochs of the training dataset for different network sizes (3, 5, 7, 10, and 20 epochs for the excitatory neuron number of the hidden layer $N_{\text{hidden}} = 100$,

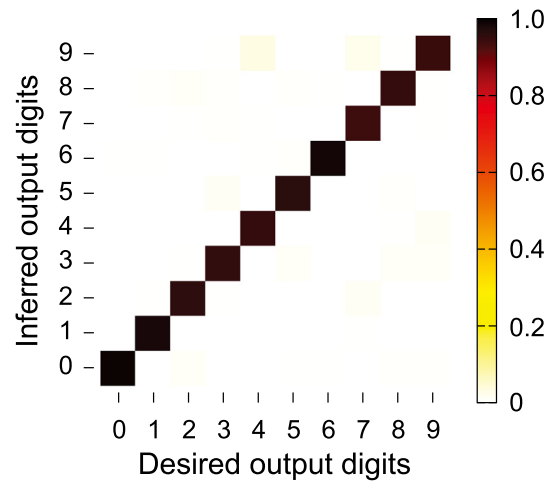


Fig. 4. Confusion matrix of test dataset results in the SL layer for the network with size $N_{\text{hidden}} = 6400$. Darker pixel indicates stronger consistency between desired (label) and inferred (real) outputs. Data were obtained for re-trained SL, as shown in Fig. 3B.

400, 1600, 6400, and 10 000, respectively). During the training process, the network performances for each training set of 10 000 samples were estimated by the test dataset. Taking network sizes $N_{\text{hidden}} = 400$ and 6400 as examples, classifying accuracies converged quickly under the two training methods (Fig. 3). Fig. 4 shows very high consistency between the desired (label) and inferred (real) outputs in the SL layer.

The results for the different learning methods are summarized in Table 1. Our two SL methods outperformed the ‘Label Statistics’ method for all small-scale networks ($N_{\text{hidden}} = 100, 400$, and 1600). In addition, the ‘Layer-by-Layer’ training method outperformed ‘Label Statistics’ for all network scales. The best performance of our SL model (96.73%) was achieved in the largest network under the ‘Layer-by-Layer’ training method. These results indicate that a SNN equipped with biologically realistic plasticity rules can achieve good SL by pure spike-based computation.

4. Visualization of model clustering ability

To demonstrate the underlying mechanisms of our SL model in pattern recognition tasks, we adopted the t-distributed Stochastic Neighbor Embedding (t-SNE) method (Maaten & Hinton, 2008) to reveal the model’s clustering ability. The t-SNE is a popular nonlinear dimensionality reduction method widely used for

Table 1

Performance for different sized networks with different classification and training methods. ‘Simultaneous’ and ‘Layer-by-layer’ are two training methods for SL. ‘Label Stat.’ represents classification by the ‘Label Statistics’ method in the hidden layer, for details please refer to Diehl and Cook (2015). We conducted five trials for each case and reported average accuracy in each case. For network size $N_{\text{hidden}} = 100, 400, 1600, 6400$, and 10000 , in Eq. (5), $\tau_\theta = 6 \times 10^6, 6 \times 10^6, 8 \times 10^6, 2 \times 10^7$ and 2×10^7 , $\alpha = 8.4 \times 10^5, 8.4 \times 10^5, 1.12 \times 10^6, 2 \times 10^6$ and 2×10^6 , respectively.

Network size (N_{hidden})	100	400	1600	6400	10000
Simultaneous SL	83.67% \pm 0.30%	91.13% \pm 0.05%	91.22% \pm 0.13%	95.89% \pm 0.14%	96.35% \pm 0.09%
Layer-by-layer SL	83.57% \pm 0.33%	91.41% \pm 0.15%	91.82% \pm 0.41%	96.28% \pm 0.10%	96.73% \pm 0.11%
Label stat.	83.00% \pm 0.50%	91.02% \pm 0.23%	91.19% \pm 0.32%	96.02% \pm 0.08%	96.52% \pm 0.13%

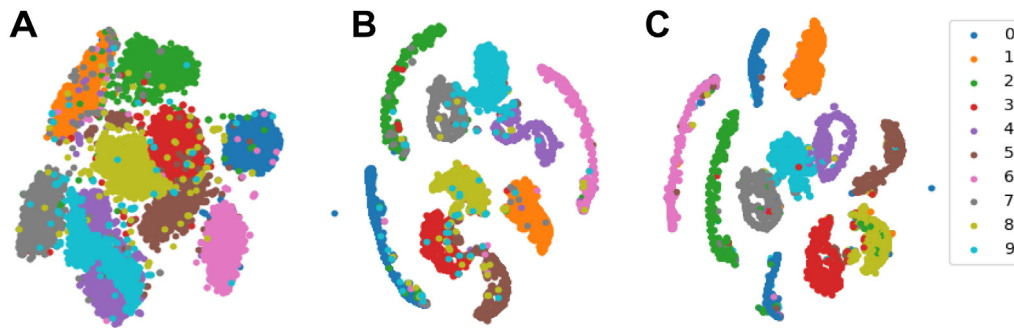


Fig. 5. Visualization of original digit patterns (A), spike activities of hidden layer (B), and spike activities of SL layer (C) for the MNIST test dataset using the t-SNE method. Each dot represents a digit sample and is colored by its corresponding label information. Activities were obtained from the case of the SL-Layer-by-Layer for $N_{\text{hidden}} = 6400$ (shown in Fig. 3).

visualizing high-dimensional data in low-dimensional space (e.g., two or three dimensions). We visualized the original digit patterns (Fig. 5A), spike activities of the hidden layer (Fig. 5B), and spike activities of the SL layer (Fig. 5C) for all samples in the test dataset. The separability of the output information of the three layers in our model increased from the input to SL layer, indicating that the SL layer served as a good classifier after training. To demonstrate why our SL method achieved effective clustering for the hidden layer outputs, we also applied t-SNE to reduce the dimensions of the out-synaptic weights of the excitatory neurons. As shown in Fig. 6, the clustering of the out-synaptic weights of the excitatory neurons was highly consistent with the clustering of their label information using the ‘Label Statistics’ method. This explains why our SL method achieved comparatively good performance as the ‘Label Statistics’ method for the classification task, although our model did not require ‘Label Statistics’ computation outside the network to calculate the most likely representation of a hidden neuron (Diehl & Cook, 2015), instead realizing SL based solely on computation within the network.

5. Comparison with other SNN models

Current SNN models for pattern recognition can be generally categorized into three classes: that is, indirect training (Cao et al., 2015; Diehl et al., 2015, 2016; Esser et al., 2015, 2016; Hu et al., 2018; Hunsberger & Eliasmith, 2015; Neil & Liu, 2016), direct SL training with BP (Beyeler et al., 2013; Lee et al., 2016, 2018; Samadi et al., 2017; Tavanaei & Maida, 2017; Wu et al., 2017; Zhang et al., 2018), and plasticity-based unsupervised training with supervised modules (Diehl & Cook, 2015; Kheradpisheh et al., 2018; Querlioz et al., 2013). Table 2 summaries several previous SNN models trained and tested using the full training and testing sets of the MNIST dataset.

Comparison with SNN models trained using BP. In previous studies with indirect training, ANNs were trained using the BP algorithm based on activity rates and transformed to corresponding equivalent SNNs based on firing rates. Although their performances were very good, they ignored the temporal evolution of SNNs and spike-based learning processes. Thus, indirect



Fig. 6. Visualization of the clustering ability of output synapses (out-synapses) of excitatory neurons in the hidden layer ($N_E = 6400$) to SL layer using the t-SNE method. Each dot represents an excitatory neuron in the hidden layer and is colored by its label using ‘Label Statistics’. Clustering of the out-synaptic weights of the excitatory neurons is highly consistent with the clustering of their labels. Synaptic weights used here were from the case in Fig. 5.

training provides very little enlightenment on how SNNs learn and encode different features of inputs. For other studies using direct SL training, most adopted the BP algorithm and calculated errors based on continuous variables, e.g., membrane potentials (voltage), currents, or activity rates, to approximate spike activities and achieve SL (Lee et al., 2016, 2018; Samadi et al., 2017; Tavanaei & Maida, 2017; Wu et al., 2017; Zhang et al., 2018). For example, Zhang et al. proposed a voltage-driven plasticity-centric SNN for SL (Zhang et al., 2018), with four learning stages required for training, i.e., equilibrium learning and voltage-based STDP-like learning rule for USL as well as voltage-based BP for SL; however, this resulted in the model being highly dissimilar to biological neuronal systems. Lee et al. pre-trained a multi-layer SNN system by STDP in an USL way for optimal initial weights and then used current-based BP to re-train all-layer weights in a supervised way (Lee et al., 2018); however, this also resulted in the model being bio-implausible due to the use of the BP algorithm.

Comparison with STDP-based models without BP. Several studies have also proposed STDP-based training methods without BP. These previous models adopted STDP-like plasticity rules for

Table 2

Comparison of classification performance between our SNN model and others on the MNIST task. All models reported here were trained based on the full training dataset and tested by the full testing dataset. LIF: Leaky integrate-and-fire; BP: Backpropagation; CSNN: Convolutional spiking neural network; SDRN: Spiking deep residual network; MLHN: Multi-layer hierarchical network; STBP: Spatio-temporal backpropagation; VDPC: Voltage-driven plasticity-centric; BA: Broadcast alignment; SDNN: Spiking deep neural network; SVM: Support vector machine; R-STDP: Reward-modulated STDP.

Network model	Learning method	Training type	(Un-)Supervised	Accuracy
In-direct training				
Deep LIF SNN (Hunsberger & Eliasmith, 2015)	BP-ANN conversion	Rate-based	Supervised	98.37%
CSNN (Diehl et al., 2015)	BP-ANN conversion	Rate-based	Supervised	99.1%
Chip-based SNN (Esser et al., 2015)	BP-ANN conversion	Rate-based	Supervised	99.42%
SDRN (Hu et al., 2018)	BP-ANN conversion	Rate-based	Supervised	99.59%
Direct training				
BP-STDP SNN (Tavanaei & Maida, 2017)	BP-STDP	Rate-based	Supervised	96.6%
Deep LIF-BA SNN (Samadi et al., 2017)	Broadcast alignment	Rate-based	Supervised	97.05%
STBP SNN (Wu et al., 2017)	Spatio-temporal BP	Rate-based	Supervised	98.89%
VDPC SNN (Zhang et al., 2018)	Equilibrium learning + STDP	Voltage-based	Supervised	98.52%
Deep SNN (Lee et al., 2016)	BP	Voltage-based	Supervised	98.88%
DCSNN (Lee, Panda, Srinivasan, & Roy, 2018)	STDP + BP	Spike-based & Current-based	Supervised	99.28%
Two-layer SNN (Querlioz, Bichler, Dollfus, & Gamrat, 2013)	Rectangular STDP	Spike-based	Unsupervised	93.5%
Two-layer SNN (Diehl & Cook, 2015)	Exponential STDP	Spike-based	Unsupervised	95.0%
SDNN (Kheradpisheh et al., 2018)	STDP + SVM	Spike-based	Unsupervised	98.4%
Three-layer SNN (Shrestha et al., 2017)	Specially modified STDP	Spike-based	Supervised	89.7%
MLHN (Beyeler et al., 2013)	STDP with Calcium variable	Spike-based	Supervised	91.6%
Bidirectional SNN (Hu et al., 2017)	Specially modified STDP	Spike-based	Supervised	96.8%
DCSNN (Mozafari et al., 2018)	STDP + R-STDP	Spike-based	Supervised	97.2%
sym-STDP SNN (Ours)	sym-STDP (DA-STDP)	Spike-based	Supervised	96.73%

USL and required a special supervised module for SL, e.g., a classifier (SVM) (Kheradpisheh et al., 2018), artificial label statistics outside the network (Diehl & Cook, 2015; Querlioz et al., 2013), or additional supervised layer (Beyeler et al., 2013; Hu et al., 2017; Mozafari et al., 2018; Shrestha et al., 2017). However, the first two supervised modules are bio-implausible because their computing modules are outside the SNNs, resulting in the SNNs having no direct relationship with SL (Diehl & Cook, 2015; Kheradpisheh et al., 2018; Querlioz et al., 2013). For example, Beyeler et al. adopted a calcium concentration-based STDP for SL (Beyeler et al., 2013), which showed considerably poorer performance than that of our model. Hu et al. used an artificially modified STDP with a special temporal learning phase for SL (Hu et al., 2017); however, their STDP rule was artificially designed and its internal mechanism was not well explained. Shrestha et al. also adopted a specially modified STDP rule with exponential weight change and extended depression window for SL in a SNN, with a similar supervised module as ours, but performance was relatively poor (less than 90%) (Shrestha et al., 2017). Mozafari et al. used a mixed classic STDP and anti-STDP rule to generate reward-modulated STDP with a remote supervised spike for SL, but they were not able to provide biological evidence to explain this type of learning (Mozafari et al., 2018).

Detail comparison with STDP-based SNN model by Diehl and Cook. Our model was inspired by the STDP-based SNN method proposed by Diehl and Cook (2015). In their two-layer SNN, STDP is used to extract features by USL and an additional computing module outside the SNN is used for label statistics in the training process and classification in the testing process. In our model, we achieved the same algebraic computation and reasoning using an additional layer of spiking neurons instead of the outside-network computations, thus achieving considerable progress in STDP-based SNN models for SL due to the completely spike-based computations. Moreover, there were two other improvements in the USL process in our model compared to that of Diehl and Cook (2015). The first improvement was the novel sym-STDP rule rooted in DA-STDP, with DA-STDP able to give a potential explanation for the SL processes occurring in the

brain. That is, we speculated that DA may be involved in the SL process and that local synaptic plasticity could be changed to sym-STDP during the whole training process. With the aid of the forced firing of a supervised neuron by the incoming teacher signal, sym-STDP could establish the relationship between the input and its teacher information after sufficient training. The second improvement was the new dynamic threshold rule in Eq. (5), in which a decay factor for α was introduced, which could significantly improve performance.

It should be noted that several other SNN training models have performed classification tasks in other ways, but their recognition performance was relatively poor (Lin, Ma, Meng, & Chen, 2018; Sporea & Grüning, 2013). Recently, Xu et al. constructed a novel convolutional spiking neural network (CSNN) model using the tempotron as the classifier and attempted to take advantage of both convolutional structure and SNN temporal coding ability (Xu, Qi, Yu, Shen, Tang, & Pan, 2018). However, their model only achieved a maximum accuracy of 88% on a subset of the MNIST dataset (Training samples: 500; Test samples: 100) when the network size equaled 1200; in contrast, our model achieved an accuracy of 91.41% with N_{hidden} equal to 400 on the full MNIST dataset. This indicates that our model could also work very well under small network size constraints.

Thus, for the above reasons, our proposed sym-STDP based SNN model could solve the lack of bio-plausible and high-performance SNN methods for spike-based SL.

6. Robustness of our SL model

To demonstrate the robustness of our SL model, we also tested its performance on Fashion-MNIST (Xiao, Rasul, & Vollgraf, 2017), a MNIST-like fashion product dataset with 10 classes. Fashion-MNIST shares the same image size and structure of the training and testing splits as MNIST but is considered more realistic as its images are generated from front look thumbnail images of fashion products on Zalando's website via a series of conversions. Therefore, Fashion-MNIST poses a more challenging classification task than MNIST. We preprocessed the data by normalizing the

Table 3

Performance on Fashion-MNIST dataset for different sized networks with different training methods. For network size $N_{\text{hidden}} = 400$ and 6400, $\tau_{\theta} = 5 \times 10^7$ and 2×10^7 , $\alpha = 5 \times 10^6$ and 2×10^6 , $\beta = 0.05$ and 0.025, respectively.

Network size (N_{hidden})	400	6400
Simultaneous SL	77.61% \pm 0.40%	84.70% \pm 0.24%
Layer-by-layer SL	78.68% \pm 0.27%	85.31% \pm 0.16%
Label stat.	77.73% \pm 0.50%	84.89% \pm 0.21%

sum of a single sample gray value because of high variance among examples. We then made necessary parameter adjustments to the model mentioned earlier. Specifically, some class examples in Fashion-MNIST have more non-zero pixels than MNIST, such as T-shirt, pullover, shirt, and coat. We decreased β in Eq. (7) so as to reduce weights and offset the impact of excessive spike quantity. We trained our model under different network sizes ($N_{\text{hidden}} = 400$ and 6400). The same evaluation criteria were applied, as shown in Table 3, our model also performed well on the Fashion-MNIST task under both two SL training methods. For example, the Layer-by-Layer training methods achieved accuracies of 78.68% and 85.31% for network sizes 400 and 6400, respectively. The best performance of our model is comparable with traditional machine learning methods, such as SVM with linear kernel (83.9%) and multilayer perceptron (the highest accuracy reported was 87.1%) (Xiao et al., 2017). These results further confirm the robustness of our SL model.

7. Discussion

A neural network model with biological plausibility must meet three basic characteristics, i.e., the ability to integrate temporal input and generate spike output, spike-based computation for training and inference, and all learning rules rooted in biological experiments. Here, we used the LIF neuron model, with all learning rules (e.g., sym-STDP, synaptic scaling, and dynamic threshold) rooted in experiments and computation based on spikes. Thus, the proposed SNN model meets all the above requirements and is a true biologically plausible neural network model.

However, how did our model obtain good pattern recognition performance? This was mainly because the three learning rules worked synergistically to achieve good feature extraction and generate the appropriate mapping from input to output. The sym-STDP rule demonstrated a considerable advantage by extracting the relationship of spike events, regardless of their temporal order, in two connected neurons, with synaptic scaling able to stabilize total in-synaptic weights and create weight competition among in-synapses of a neuron to ensure that the suitable group of synapses became strong. Furthermore, the dynamic threshold mechanism compelled a neuron to fire for matched patterns but rarely for unmatched ones, which generated neuron selectivity to a special class of patterns. By combining the three bio-plausible plasticity rules, our SNN model established a strong relationship between the input signal and supervised signal after sufficient training, ensuring effective SL implementation and good performance in the benchmark pattern recognition task (MNIST). The proposed model also obtained good performance by training two layers synchronously, whereas many previous SNN models require layer-by-layer or multi-phase/multi-step training (Hu et al., 2017; Kheradpisheh et al., 2018; Lee et al., 2016; Samadi et al., 2017; Zhang et al., 2018).

To explore the effect of different deep network structures, we also tested the performance of our SL algorithm when adding or removing a hidden layer in our network model. Taking $N_E = 100$ as an example, the network model with no hidden layer achieved a performance of only 57.74%, whereas the network

model with two hidden layers reached 81.65%, slightly lower than that achieved in the network with one-hidden layer (83.57%). In deep ANNs, classification performance is usually improved by increasing the number of hidden layers; however, this phenomenon was not found in our model. In our model, each hidden layer excitatory neuron has a global receptive field, which differs from that in deep convolutional neural networks (CNNs), where each convolutional layer neuron has a common local receptive field. A CNN needs more convolutional layers to generate larger receptive fields and thus achieve a global receptive field in the last layer; however, this is not required in our model. After training, the global receptive field of a hidden-layer neuron resulted in a maximum response of the neuron to a special category of input patterns, thus allowing hidden-layer neurons to be easily divided into different classes. That is, unsupervised clustering can be realized using just one hidden layer. A subsequent supervised layer can then directly implement classification after training based on supervised label information. Therefore, in the framework of our network model, it would be useless to promote performance by increasing hidden layers, and also would harm performance by removing the hidden layer as the model can only generate 10 special global receptive fields (by 10 supervised neurons) for all 60000 training patterns. Therefore, in our model, we increased the number of hidden-layer neurons to improve performance rather than the number of hidden layers. As no more than 10 neurons usually responded to an input pattern in the hidden layer, activity was so sparse that simple linear mapping from the high-dimension activities in the hidden layer to the low-dimension activities in the classification layer was possible, again indicating no need for further hidden layers. Nevertheless, additional hidden layers may be necessary to improve performance if the convolutional structure (local receptive fields) is adopted for the network. We will explore convolutional SNNs to further verify the universality of our SL algorithm in the future.

In our SNN model, DA was found to be a key factor for achieving SL. Dopamine plays a critical role in different learning processes and can serve as a reward signal for reinforcement learning (Dayan & Balleine, 2002; Glimcher, 2011; Holroyd & Coles, 2002; Wise, 2004). A special form of DA-modulated STDP, different to the symmetric one used here, has been applied for reinforcement learning in SNNs previously (Izhikevich, 2007); however, no direct experimental evidence for this kind of STDP rule has yet been reported. Here, we assumed that DA may also be involved in the SL process, with the symmetric DA-STDP rule found in experiments to modify synaptic weights during SL. Our work further indicated the potentially diverse functions of DA in regulating neural networks for information processing. Moreover, even if this DA-involved assumption is shown to be biologically unsound in the future, direct experimental evidence, i.e., identification of a sym-STDP without the need of DA in the hippocampal CA3-CA3 synapses under slow-frequency stimulation (Mishra et al., 2016), supports the bio-plausibility of the sym-STDP rule used in our model. It is worth noting that the sym-STDP rule is a spiking version of the original Hebbian rule, that is, ‘cells that fire together wire together’, and a rate-based neural network with the Hebbian rule, synaptic scaling, and dynamic bias could be expected to have similar classification abilities as our model. However, the performance of the rate model may not be as high as that reported here. Further exploration is needed using the simplified rate model with the original Hebbian rule.

Several SNN models have been developed to provide a general framework for SNN learning, e.g., liquid state machine (LSM) (Maass, Natschlager, & Markram, 2002) and NeuCube (Kasabov, 2014, 2018). The LSM is a classic model of reservoir computing (RC) (Jaeger, 2001; Jaeger & Haas, 2004; Lukoševičius & Jaeger, 2009; Lukoševičius, Jaeger, & Schrauwen, 2012; Maass et al.,

2002), which contains an input module, a reservoir module consisting of relatively large spiking neurons with randomly or fully connected structure, and a readout module. In a typical LSM, the low dimensional input, usually possessing spatiotemporal information, is fed into the high-dimensional space of the reservoir to optimally maintain a high-dimensional, dynamic representation of information as a short-term memory when the state of the reservoir is at the edge of chaos. The reservoir is then projected to the readout module where synaptic weights are modified by learning methods. NeuCube is also a general SNN model similar to a RC model, with both a similar network structure and ability to encode spatiotemporal input as LSM (Kasabov, 2014, 2018). In NeuCube, the reservoir is a three-dimensional-based structure and can dynamically evolve its weight structure using unsupervised learning rules, which allows it to encode an input as a special activity trajectory in the high-dimension activity space of the reservoir. In the readout module of NeuCube, an evolving SNN (eSNN) is usually adopted for classification (Kasabov, 2018), which can dynamically create a new output neuron for clustering by evaluating the similarity between a new input and encoded inputs in Euler space. Our model is a general SNN model for SL. There are some significant differences between our model and the two models. Firstly, our learning rule (sym-STDP) is consistent for different layers across the whole model, whereas NeuCube generally uses different learning rules for synapses in the reservoir and out-synapses reservoir and the classic LSM only uses a learning rule to adjust the weights of readout synapses from the reservoir with fixed inner connections. Secondly, our model contains biologically plausible receptive fields, which can be learned during training, whereas the concept of receptive fields is ambiguous in NeuCube and LSM. Compared to the two models, however, our model lacks the ability to process complex temporal information. Therefore, in the future, it would be useful to construct a model possessing the advantages of the above models to improve the ability to process complex spatiotemporal information.

As the plasticity rules used here were based purely on local spike events, in contrast with the BP method, our model not only has the potential to be applied to other machine learning tasks under the SL framework but may also be suitable for online learning on programmable neuromorphic chips. Moreover, our hypothesis regarding the function of DA in SL processing may serve as a potential mechanism for synaptic information processing of SL in the brain, which will need to be verified in future experiments.

Reference data

The code is available at <https://github.com/haoyz/sym-STDP-SNN>.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 11505283), Beijing Brain Science Project, China (Grant No. Z181100001518006), and Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32070000). We also gratefully acknowledge the support of the NVIDIA Corporation, United States of America for the donation of the Titan X Pascal GPU used in this research.

References

Beyeler, M., Dutt, N. D., & Krichmar, J. L. (2013). Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule. *Neural Networks*, 48(10), 109–124.

- Bi, G.-q., & Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24), 10464–10472.
- Brzosko, Z., Schultz, W., & Paulsen, O. (2015). Retroactive modulation of spike timing-dependent plasticity by dopamine. *Elife*, 4.
- Cao, Y., Chen, Y., & Khosla, D. (2015). Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 113(1), 54–66.
- Caporale, N., & Dan, Y. (2008). Spike timing-dependent plasticity: A Hebbian learning rule. *Annual Review of Neuroscience*, 31, 25–46.
- Cooper, L. N., & Bear, M. F. (2012). The BCM theory of synapse modification at 30: interaction of theory with experiment. *Nature Reviews Neuroscience*, 13(11), 798.
- Davis, G. W. (2006). Homeostatic control of neural activity: from phenomenology to molecular design. *Annual Review of Neuroscience*, 29, 307–323.
- Dayan, P., & Balleine, B. W. (2002). Reward, motivation, and reinforcement learning. *Neuron*, 36(2), 285–298.
- Diehl, P. U., & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, 9.
- Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S.-C., & Pfeiffer, M. (2015). Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE.
- Diehl, P. U., Zarella, G., Cassidy, A., Pedroni, B. U., & Neftci, E. (2016). Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware. In *IEEE International Conference on Rebooting Computing (ICRC)* (pp. 1–8). IEEE.
- Effenberger, F., Jost, J., & Levina, A. (2015). Self-organization in balanced state networks by STDP and homeostatic plasticity. *PLoS Computational Biology*, 11(9).
- Esser, S. K., Appuswamy, R., Merolla, P. A., Arthur, J. V., & Modha, D. S. (2015). Backpropagation for energy-efficient neuromorphic computing. In *Advances in Neural Information Processing Systems (NIPS)* (pp. 1117–1125).
- Esser, S. K., Merolla, P. A., Arthur, J. V., Cassidy, A. S., Appuswamy, R., Andreopoulos, A., et al. (2016). Convolutional networks for fast, energy-efficient neuromorphic computing. *Proceedings of the National Academy of Sciences of the United States of America*, 113(41), 11441–11446.
- Glimcher, P. W. (2011). Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. *Proceedings of the National Academy of Sciences*, 108(Suppl. 3), 15647–15654.
- Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10), 2222–2232.
- Hengen, K. B., Lambo, M. E., Van Hooser, S. D., Katz, D. B., & Turrigiano, G. G. (2013). Firing rate homeostasis in visual cortex of freely behaving rodents. *Neuron*, 80(2), 335–342.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- Holroyd, C. B., & Coles, M. G. (2002). The neural basis of human error processing: reinforcement learning, dopamine, and the error-related negativity. *Psychological Review*, 109(4), 679.
- Hu, Y., Tang, H., Wang, Y., & Pan, G. (2018). Spiking deep residual network. arXiv preprint arXiv:1805.01352.
- Hu, Z., Wang, T., & Hu, X. (2017). An STDP-based supervised learning algorithm for spiking neural networks. In *International Conference on Neural Information Processing (ICONIP)* (pp. 92–100). Springer.
- Hunsberger, E., & Eliasmith, C. (2015). Spiking deep networks with LIF neurons. arXiv preprint arXiv:1510.08829.
- Izhikevich, E. M. (2007). Solving the distal reward problem through linkage of STDP and dopamine signaling. *Cerebral Cortex*, 17(10), 2443–2452.
- Jaeger, H. (2001). The “echo state” approach to analysing and training recurrent neural networks—with an erratum note: Technical report, Bonn, Germany: German National Research Center for Information Technology GMD, 148 (34) 13.
- Jaeger, H., & Haas, H. (2004). Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667), 78–80.
- Kasabov, N. K. (2014). NeuCube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data. *Neural Networks*, 52, 62–76.
- Kasabov, N. K. (2018). *Time-space, spiking neural networks and brain-inspired artificial intelligence* (vol. 7). Springer.
- Keck, T., Keller, G. B., Jacobsen, R. I., Eysel, U. T., Bonhoeffer, T., & Hübner, M. (2013). Synaptic scaling and homeostatic plasticity in the mouse visual cortex in vivo. *Neuron*, 80(2), 327–334.
- Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., & Masquelier, T. (2018). STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99, 56–67.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

- Lee, J. H., Delbruck, T., & Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 10.
- Lee, C., Panda, P., Srinivasan, G., & Roy, K. (2018). Training deep spiking convolutional neural networks with STDP-based unsupervised pre-training followed by supervised fine-tuning. *Frontiers in Neuroscience*, 12, 435.
- Lin, Z., Ma, D., Meng, J., & Chen, L. (2018). Relative ordering learning in spiking neural network for pattern recognition. *Neurocomputing*, 275, 94–106.
- Lukoševičius, M., & Jaeger, H. (2009). Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3), 127–149.
- Lukoševičius, M., Jaeger, H., & Schrauwen, B. (2012). Reservoir computing trends. *KI-Künstliche Intelligenz*, 26(4), 365–371.
- Maass, W., & Markram, H. (2004). On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences*, 69(4), 593–616.
- Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11), 2531–2560.
- Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2579–2605.
- Maffei, A., & Turrigiano, G. G. (2008). Multiple modes of network homeostasis in visual cortical layer 2/3. *Journal of Neuroscience*, 28(17), 4377–4384.
- Masuda, N., & Kori, H. (2007). Formation of feedforward networks and frequency synchrony by spike-timing-dependent plasticity. *Journal of Computational Neuroscience*, 22(3), 327–345.
- Mishra, R. K., Kim, S., Guzman, S. J., & Jonas, P. (2016). Symmetric spike timing-dependent plasticity at CA3–CA3 synapses optimizes storage and recall in autoassociative networks. *Nature Communications*, 7, 11552.
- Mozafari, M., Ganjtabesh, M., Nowzari-Dalini, A., Thorpe, S. J., & Masquelier, T. (2018). Combining STDP and reward-modulated STDP in deep convolutional spiking neural networks for digit recognition. arXiv preprint arXiv:1804.00227.
- Neil, D., & Liu, S.-C. (2016). Effective sensor fusion with event-based sensors and deep network architectures. In *IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 2282–2285). IEEE.
- Pozo, K., & Goda, Y. (2010). Unraveling mechanisms of homeostatic synaptic plasticity. *Neuron*, 66(3), 337–351.
- Querlioz, D., Bichler, O., Dollfus, P., & Gamrat, C. (2013). Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE Transactions on Nanotechnology*, 12(3), 288–295.
- Ruan, H., Saur, T., & Yao, W.-D. (2014). Dopamine-enabled anti-Hebbian timing-dependent plasticity in prefrontal circuitry. *Frontiers in Neural Circuits*, 8(April), 1–12.
- Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive Modeling*, 5(3), 1.
- Samadi, A., Lillicrap, T. P., & Tweed, D. B. (2017). Deep learning with dynamic spiking neurons and fixed feedback weights. *Neural Computation*, 29(3), 578–602.
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.
- Serrano-Gotarredona, T., Masquelier, T., Prodromakis, T., Indiveri, G., & Linares-Barranco, B. (2013). STDP and STDP variations with memristors for spiking neuromorphic learning systems. *Frontiers in Neuroscience*, 7, 2.
- Shrestha, A., Ahmed, K., Wang, Y., & Qiu, Q. (2017). Stable spike-timing dependent plasticity rule for multilayer unsupervised and supervised learning. In *International Joint Conference on Neural Networks (IJCNN)* (pp. 1999–2006). IEEE.
- Song, S., Miller, K. D., & Abbott, L. F. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9), 919.
- Sporea, I., & Grüning, A. (2013). Supervised learning in multilayer spiking neural networks. *Neural Computation*, 25(2), 473–509.
- Sun, Q.-Q. (2009). Experience-dependent intrinsic plasticity in interneurons of barrel cortex layer IV. *Journal of Neurophysiology*, 102(5), 2955–2973.
- Tanaka, H., Morie, T., & Aihara, K. (2009). A CMOS spiking neural network circuit with symmetric/asymmetric STDP function. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 92(7), 1690–1698.
- Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. S. (2018). Deep learning in spiking neural networks. arXiv preprint arXiv:1804.08150.
- Tavanaei, A., & Maida, A. S. (2017). BP-STDP: Approximating backpropagation using spike timing dependent plasticity. arXiv preprint arXiv:1711.04214.
- Tritsch, N. X., & Sabatini, B. L. (2012). Dopaminergic modulation of synaptic transmission in cortex and striatum. *Neuron*, 76(1), 33–50.
- Turrigiano, G. G. (2008). The self-tuning neuron: Synaptic scaling of excitatory synapses. *Cell*, 135(3), 422–435.
- Turrigiano, G. G., Leslie, K. R., Desai, N. S., Rutherford, L. C., & Nelson, S. B. (1998). Activity-dependent scaling of quantal amplitude in neocortical neurons. *Nature*, 391(6670), 892.
- Vogels, T. P., Sprekeler, H., Zenke, F., Clopath, C., & Gerstner, W. (2011). Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062), 1569–1573.
- Wise, R. A. (2004). Dopamine, learning and motivation. *Nature Reviews Neuroscience*, 5(6), 483.
- Wu, Y., Deng, L., Li, G., Zhu, J., & Shi, L. (2017). Spatio-temporal backpropagation for training high-performance spiking neural networks. arXiv preprint arXiv:1706.02609.
- Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.
- Xu, Q., Qi, Y., Yu, H., Shen, J., Tang, H., & Pan, G. (2018). Csn: an augmented spiking based framework with perceptron-inception. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18* (pp. 1646–1652).
- Yavuz, E., Turner, J., & Nowotny, T. (2016). GeNN: a code generation framework for accelerated brain simulations. *Scientific Reports*, 6, 18854.
- Yeung, L. C., Shouval, H. Z., Blais, B. S., & Cooper, L. N. (2004). Synaptic homeostasis and input selectivity follow from a calcium-dependent plasticity model. *Proceedings of the National Academy of Sciences*, 101(41), 14943–14948.
- Zhang, J. C., Lau, P. M., Bi, G. Q., & Stevens, C. F. (2009). Gain in sensitivity and loss in temporal contrast of STDP by dopaminergic modulation at hippocampal synapses. *Proceedings of the National Academy of Sciences of the United States of America*, 106(31), 13028–13033.
- Zhang, W., & Linden, D. J. (2003). The other side of the engram: experience-driven changes in neuronal intrinsic excitability. *Nature Reviews Neuroscience*, 4(11), 885.
- Zhang, T., Zeng, Y., Zhao, D., & Shi, M. (2018). A plasticity-centric approach to train the non-differential spiking neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.