Neural Networks 154 (2022) 68–77

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

BackEISNN: A deep spiking neural network with adaptive self-feedback and balanced excitatory–inhibitory neurons[☆]

Dongcheng Zhao [a], Yi Zeng [a,b,c,d,*], Yang Li [a,b]

[a] Research Center for Brain-Inspired Intelligence, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[b] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[c] Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China
[d] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 9 August 2021
Received in revised form 6 May 2022
Accepted 28 June 2022
Available online 11 July 2022

Keywords:
Spiking neural networks
Adaptive self-feedback connections
Dynamically balanced
excitatory–inhibitory neurons

ABSTRACT

Spiking neural networks (SNNs) transmit information through discrete spikes that perform well in processing spatial–temporal information. Owing to their nondifferentiable characteristic, difficulties persist in designing SNNs that deliver good performance. SNNs trained with backpropagation have recently exhibited impressive performance by using gradient approximation. However, their performance on complex tasks remains significantly inferior to that of deep neural networks. By taking inspiration from autapses in the brain that connect spiking neurons with a self-feedback connection, we apply adaptive time-delayed self-feedback to the membrane potential to regulate the precision of the spikes. We also strike a balance between the excitatory and inhibitory mechanisms of neurons to dynamically control the output of spiking neurons. By combining these two mechanisms, we propose a deep SNN with adaptive self-feedback and balanced excitatory and inhibitory neurons (BackEISNN). The results of experiments on several standard datasets show that the two modules not only accelerate the convergence of the network but also increase its accuracy. Our model achieved state-of-the-art performance on the MNIST, Fashion-MNIST, and N-MNIST datasets. The proposed BackEISNN also achieved remarkably good performance on the CIFAR10 dataset while using a relatively light structure that competes against state-of-the-art SNNs.

1. Introduction

In the past few years, deep neural networks (DNNs) have made tremendous progress in various machine learning tasks in computer vision (Krizhevsky, Sutskever, & Hinton, 2017), natural language processing (Collobert & Weston, 2008), and speech recognition (Amodei et al., 2016). However, DNNs mimic only the hierarchical topological structure of the flow of information in the brain and process data in a real-valued form, but this is far removed from the information processing mechanisms of the brain. Spikes play a crucial role in efficient information processing, and spiking neural networks (SNNs) have been developed to emulate information processing in the brain. SNNs are considered to be third-generation neural networks (Maass, 1997). The discrete spike-driven communication between spiking neurons, multiple brain-inspired learning rules, and intricate connections make these networks biologically plausible and energy efficient.

However, owing to the lack of an effective training algorithm, the development of SNNs has stalled for a while, and they have not yet demonstrated performance comparable to that of DNNs. Many researchers have taken inspiration from the process of learning by synapses in the brain to introduce such mechanisms as spike-timing-dependent plasticity (Bi & Poo, 1998) and short-term facilitation or depression to the learning of weights of the SNNs (Diehl & Cook, 2015; Falez, Tirilly, Bilasco, Devienne, & Boulet, 2019; Tavanaei & Maida, 2016, 2017; Zhang, Zeng, Zhao and Shi, 2018; Zhang, Zeng, Zhao and Xu, 2018). Although such an approach is suitable for biological interpretation, most of these mechanisms are local learning rules, and it becomes increasingly difficult to achieve global convergence as the number of network layers increases. GLSNN (Zhao, Zeng, Zhang, Shi, & Zhao, 2020) introduces global feedback layers combined with local learning rules for optimization to alleviate this problem, but still performs poorly on complex datasets.

The success of deep learning comes from the backpropagation algorithm; however, the characteristics of discontinuity and nondifferentiability of SNNs have impeded the implementation of backpropagation based on gradient descent. It is not feasible to apply the backpropagation algorithm to directly train SNNs. An

[☆] Dongcheng Zhao and Yi Zeng contributed equally to this work.
[*] Correspondence to: Institute of Automation, Chinese Academy of Sciences, Beijing, China.

E-mail addresses: zhaodongcheng2016@ia.ac.cn (D. Zhao), yi.zeng@ia.ac.cn (Y. Zeng), liyang2019@ia.ac.cn (Y. Li).

https://doi.org/10.1016/j.neunet.2022.06.036

alternative is to convert well-trained DNNs into SNNs through additional adjustments of their parameters (Diehl et al., 2015; Hu, Tang, & Pan, 2018; Sengupta, Ye, Wang, Liu, & Roy, 2019; Xu et al., 2018). Such methods of conversion have achieved accuracy comparable to the state of the art on large datasets such as ImageNet and complex architectures such as VGG and ResNet. However, their impressive performance is due to well-trained DNNs, and this does not solve the problem of adequately training SNNs. Moreover, their DNN-based training does not take full advantage of the temporal information in SNNs. When SNNs are deep, the time window should be large enough to enable the mean firing rates to approach the values obtained by DNNs. With the recent proposal of the approximation of the gradient of spiking threshold functions, the backpropagation algorithm can now be directly applied to train SNNs (Jin, Zhang, & Li, 2018; Lee, Delbruck, & Pfeiffer, 2016; Wu, Deng, Li, Zhu, & Shi, 2018; Wu et al., 2019). In this process, the discontinuous derivative of the spiking neurons is approximated by a continuous function. Although they perform well on some simple datasets, their performance on the whole still lags significantly behind that of traditional DNNs.

In addition to typical spiking neurons and feedforward network structures, many complicated mechanisms support the brain's learning and inference. We draw on some of these to further improve backpropagation-based SNNs. The typical network structure is based on a simple forward structure. When the spiking neuron's membrane reaches a threshold, it releases a spike to the postsynaptic neuron. There are many other complex structures in the brain, such as feedback connections (Felleman & Van Essen, 1991; Sporns & Zwi, 2004). Cross-layer feedback connections take the information predicted by the higher cortex to early cortical areas to help with inference and learning. In particular, the autapse connected to the soma (Ikeda & Bekkers, 2006; Wang et al., 2017; Yin et al., 2018) applies time-delayed feedback on the neuron's membrane potential to regulate the precision of the spike and network activity.

Excitatory spiking neurons are used in most current SNNs. Once the membrane potential of the presynaptic neuron reaches a certain threshold, it releases the spike to enhance the membrane potential and render the postsynaptic neurons easier to fire. However, there exist both excitatory and inhibitory neurons in the brain, and the dynamic balance between them is crucial to healthy cognition and behavior (Dehghani et al., 2016; Rubin, Abbott, & Sompolinsky, 2017).

By taking inspiration from the two mechanisms described above, we introduce an adaptive time-delayed self-feedback mechanism (SFBM) to help regulate the membrane potential to improve backpropagation-based SNNs. We also introduce inhibitory neurons to backpropagation-based SNNs. The dynamic balance between the excitatory and inhibitory neurons accelerates the convergence of the neural networks and improves their performance. We use the combination of the two mechanisms to propose a deep SNN with adaptive self-feedback and balanced excitatory–inhibitory neurons (BackEISNN). The results of experiments on several commonly used datasets verified the performance of our BackEISNN. It achieved state-of-the-art performance on the MNIST, N-MNIST, and Fashion-MNIST datasets. Our contributions here can be summarized as follows:

- We introduce the adaptive SFBM to apply time-delayed feedback to the membrane potential of spiking neurons to regulate the precision of spikes.
- We introduce a balanced excitatory and inhibitory neuron mechanism (BEIM) to control the balanced firing of spikes, which enables the network to converge more quickly and thus perform better.
- We combined the SFBM and BEIM, and subjected this combination to extensive experiments on the MNIST, Fashion-MNIST, N-MNIST, and CIFAR10 datasets. The results indicate that the proposed mechanisms can significantly improve the performance of backpropagation-based SNNs and accelerate the training of SNNs. It achieved state-of-the-art performance on the MNIST, N-MNIST, and Fashion-MNIST datasets, and delivered impressive performance on the CIFAR10 dataset while using a relatively light structure.

2. Related work

This section reviews several SNNs developed in recent years, including those that contain biological mechanisms for better training.

The SpikeProp algorithm (Bohte, Kok, & La Poutré, 2000) can be viewed as the first attempt to train SNNs by using discontinuous spiking activities. However, the restriction of a single spike output limits its performance on real-world tasks. O'Connor and Welling (2016) use a spiking version of backpropagation to train a deep SNN with rectified linear units, Wu et al. (2019) treat the number of spikes as the surrogate for gradient backpropagation. Lee et al. (2016) treat the membrane potentials of the spiking neurons as differentiable signals and the instances of spiking as noise.

However, none of these methods explicitly considers the temporal correlation among neural activities. SLAYER (Shrestha & Orchard, 2018) handles the problem of nondifferentiability of the spike function by considering the temporal dependence between the input and output signals. The STBP algorithm (Wu et al., 2018) uses spatial–temporal backpropagation to train SNNs, and an improved version of it (Wu, Deng et al., 2019) uses NeuNorm and decoding voting mechanisms to improve performance on various datasets. The HM2-BP algorithm of Jin et al. (2018) uses hybrid macro-level/micro-level backpropagation. The micro level is used to capture the temporal effects, and both levels are used to compute rate-coded errors. The ST-RSBP algorithm of Zhang and Li (2019) uses a spike-level backpropagation algorithm to help train a recurrent SNN, and the long short-term memory SNN of Bellec, Salaj, Subramoney, Legenstein, and Maass (2018) introduces adaptive neurons in the recurrent SNN to help it acquire knowledge in a learning-to-learn scheme. The TSSL-BP method of Zhang and Li (2020) introduces interneuron and intraneuron dependencies in error backpropagation to increase the precision of temporal learning. However, the performance of backpropagation-based SNNs still lags far behind that of DNNs owing to the special form of spike transmission in them.

Others have taken inspiration from the structures and rules of learning of the brain to improve network learning. LISNN (Cheng, Hao, Xu, & Xu, 2020) introduces lateral interaction to SNNs to enhance the performance of the network and its robustness against noise. The distributed coding SNN of Machingal, Thousif, Dora, and Sundaram (2020) introduces the distributed coding layer with inhibitory interconnections, and the weight is updated with a self-regulated learning algorithm. In addition to the rules of learning of the spike-timing-dependent plasticity, Zhang, Zeng, Zhao and Xu (2018) introduce short-term plasticity and a balance between excitatory and inhibitory neurons based on a voltage-driven plasticity-centric SNN (Zhang, Zeng, Zhao and Shi, 2018), but the extent of excitability has been tested by only manually trying different ratios. Moreover, the winner-takes-all (Diehl & Cook, 2015) and population coding (Pan, Wu, Zhang, Li, & Chua, 2019) schemes have been used for training SNNs. A gap in performance between SNNs and DNNs persists, however, owing to the special form of spike transmission in the former. Inspired by the autapses as well as the balance between excitatory and inhibitory neurons in the brain, we introduce an adaptive SFBM as well as a mechanism to balance excitatory and inhibitory neurons in this study to improve the performance of backpropagation-based SNNs.

3. Methods

In this section, we first introduce the basic model of the neuron used here, and then provide detailed descriptions of our adaptive SFBM and BEIM. Finally, the entire pipeline of our BackEISNN is presented.

3.1. Basic leaky integrate-and-fire neuron model

The leaky integrate-and-fire model is the model most commonly used to describe the dynamic neural activities of spiking neurons, including dynamic changes in the membrane potential and the firing process of the spikes. It can be formulated as a differential formula, as in Eq. (1):

$$\tau \frac{dV(t)}{dt} = -V(t) + RI(t), \tag{1}$$

where $V(t)$ is the membrane potential, R is the resistance of the membrane, $\tau = CR$ denotes the time constant, and $I(t) = \sum_j^N w_{j,i}\delta_j$ denotes the total input generated by synaptic currents triggered by the arrival of spikes of presynaptic neurons. When the membrane potential reaches a threshold V_{th}, the neuron fires a spike.

To improve the computational model, we modify Eq. (1) to produce a discrete form as shown in Eq. (2):

$$V_t = (1 - \frac{1}{\tau})V_{t-1} + \frac{1}{C}I_t. \tag{2}$$

When the neuron fires a spike, the membrane potential is reset to V_{reset}. By assuming $V_{reset} = 0$, and $C = 1$ as usual, we can write the final equation as

$$V_t = (1 - \frac{1}{\tau})V_{t-1}(1 - \delta_{t-1}) + I_t, \tag{3}$$

where δ_{t-1} represents the spikes at time $t - 1$.

$$\delta_t = S(V_t) = \begin{cases} 1 & \text{if} \quad V_t \geq V_{th}, \\ 0 & \text{if} \quad V_t < V_{th}. \end{cases} \tag{4}$$

When backpropagation is being performed in SNNs, spike activation cannot be derived, and this is a critical issue that restricts the development of backpropagation-trained SNNs. Here we use the surrogate gradient used in Wu, Deng et al. (2019) and Wu et al. (2018) as shown in Eq. (5):

$$\frac{\partial S(V)}{\partial V} = \begin{cases} 1 & \text{if} \quad \frac{-1}{2} \leq V - V_{th} \geq \frac{1}{2}, \\ 0 & \text{otherwise}. \end{cases} \tag{5}$$

3.2. Adaptive self-feedback mechanism

In addition to the traditional mode of information transmission in the brain, presynaptic neurons emit spikes along the axons to the postsynaptic neurons. In particular, the autapse connects the soma to a self-feedback loop that transmits information to itself. As shown in Fig. 1, the autapses in the brain act on the membrane potential to help regulate spike precision and network activity in a time-delayed self-feedback.

Inspired by the autapses in the brain, we introduce an adaptive SFBM for convolutional SNNs. Each convolutional layer can be regarded as a large neuron model, and the autapses are modeled as self-feedback connections to control the input current. To reflect the modeling of the delay, we use spikes fired at the previous timestep to regulate the input to the neurons.

As shown in Fig. 1, the spikes sent by the layer at the previous timestep are input to the convolution operation with a sigmoid activation function, which takes advantage of the temporal information of SNNs. The size of the output is identical to that of the input current received by the layer, with a value between 0 and 1. It can thus adjust the input adaptively to control the input to the membrane potential so that the spikes fire more accurately.

A detailed description of the adaptive SFBM is shown in Eq. (6), where f denotes the convolution operation, σ denotes the sigmoid function, the self-feedback gate SFB_t is multiplied by the input, and $\odot$ represents element-wise multiplication:

$$\begin{cases} SFB_t = \sigma(f(\delta_{t-1})), \\ V_t = (1 - \frac{1}{\tau})V_{t-1}(1 - \delta_{t-1}) + SFB_t \odot I_t. \end{cases} \tag{6}$$

With the introduction of the SFBM, first, the postsynaptic neurons are no longer passively receiving the input from the presynaptic neurons. The SFBM will help to select and control the scale of the input to the membrane potential. As a result, the precision of the spikes can be better controlled. Second, as can be seen in Eq. (6), the temporal dependency of V_t and V_{t-1} relies mainly on the leaky term $1 - \frac{1}{\tau}$. As $SFB_t = \sigma(f(\delta_{t-1}))$ is a function of δ_{t-1}, the interaction of information from different timesteps of the SNN is enhanced, thereby increasing the temporal dependency of the SNN.

3.3. Mechanism of balance between excitatory and inhibitory neurons

Both excitatory and inhibitory neurons exist in the brain. Their balanced combination helps with learning and inference. As shown in Fig. 2, excitatory neurons exhibit excitatory postsynaptic potential, which makes it easier for the postsynaptic neurons to fire. By contrast, inhibitory neurons exhibit inhibitory postsynaptic potential, making postsynaptic neurons less likely to fire. We define excitatory neurons as neurons that release positive spikes and inhibitory neurons as neurons that release negative spikes to simplify their modeling. The combination of positive and negative spikes was first used in SpikeGrad (Thiele, Bichler, & Dupret, 2019), shown in Eq. (7):

$$S(V) = \begin{cases} 1 & \text{if} \quad V \geq V_{th}, \\ -1 & \text{if} \quad V \leq -V_{th}, \\ 0 & \text{otherwise}. \end{cases} \tag{7}$$

The rules for sending positive or negative spikes are determined in advance. A positive spike occurs when the membrane potential exceeds the positive threshold, and a negative spike occurs when it is below the negative threshold. In the biological brain, the neuron does not fire a spike when the membrane potential does not reach the threshold. In this article, we introduce a mechanism to determine whether the neurons are dynamically excitatory.

Zhu, Jiang, Yang, Hou, and Shu (2011) argued that the membrane potential dynamically maintains the balance of excitatory and inhibitory neurons. We use the membrane potential here to guide the generation of this balance. As shown in Fig. 2, before being sent to the threshold function, the membrane potential is sent to a convolutional layer with a sign activation function to determine whether the neurons are excitatory. The gradient of this challenging function is processed in the same manner as the hard threshold function mentioned above. When we obtain the neuron's positive and negative attributes, and the membrane potential reaches the threshold, we can decide whether to send a positive or a negative spike.

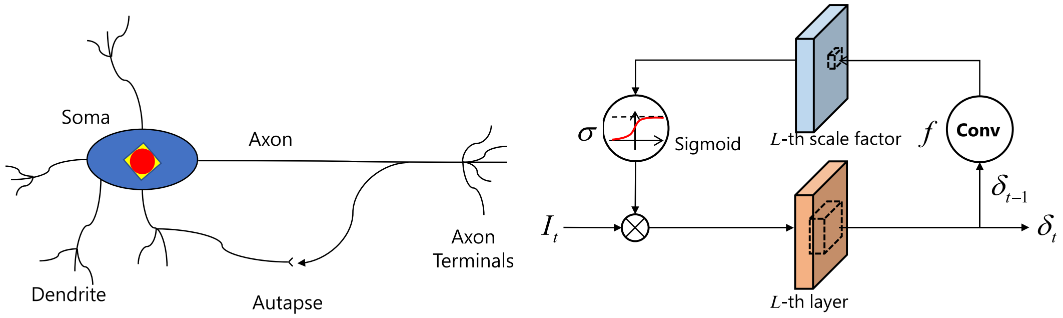

Fig. 1. An autapse in the brain that connects to the soma for self-feedback (left), and the corresponding computational model (right).

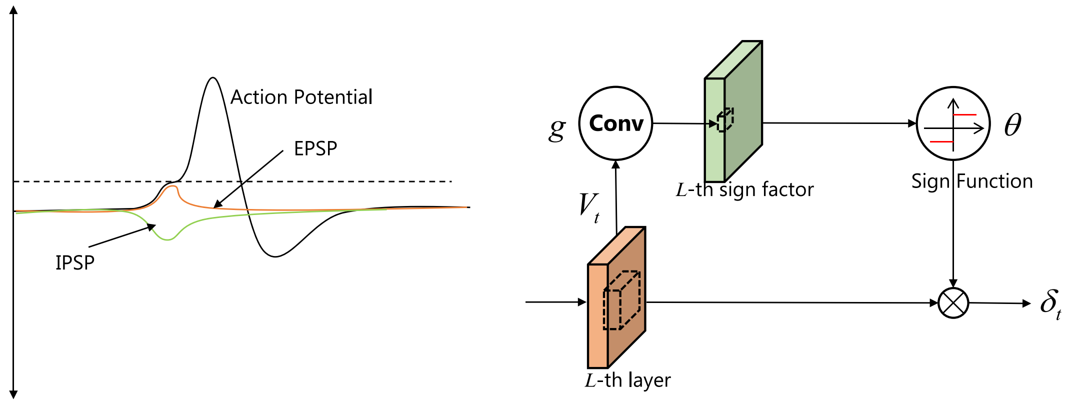

Fig. 2. The balance between excitatory and inhibitory neurons in the brain (left), and the corresponding computational model (right). EPSP, excitatory postsynaptic potential; IPSP, inhibitory postsynaptic potential.

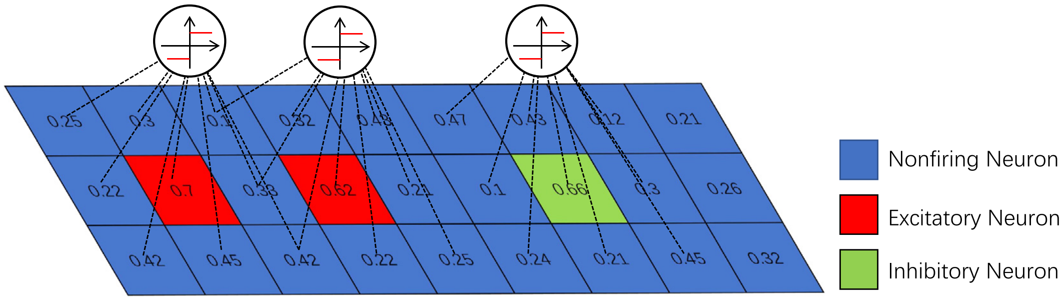

Fig. 3. The membrane potential of neurons is involved in determining whether they are excitatory.

The details are shown in Eq. (8), where g denotes the convolution operation, θ denotes the sign function, and the gate EI_t, which determines whether the given neuron is excitatory or inhibitory, is multiplied by the output:

$$\begin{cases} EI_t = \theta(g(V_t)), \\ \delta_t = EI_t \odot S(V_t). \end{cases} \tag{8}$$

The membrane potential contains a wealth of information. When the membrane potential is greater than the threshold, the information transmitted is the same regardless of its specific value. Neurons whose membrane potentials are not higher than the threshold do not transmit information, leading to a waste of information. As shown in Fig. 3, in addition to the membrane potential of this neuron, the membrane potentials of peripheral neurons are involved in the calculation of a neuron's attributes, which further increases the information transmission efficiency of the SNNs.

3.4. BackEISNN model

The entire training process is shown in Fig. 4. The input is sent to the spiking convolutional networks, and the two proposed mechanisms are combined. The process of updating the leaky integrate-and-fire neuron is shown in Eq. (9):

$$\begin{cases} SFB_t = \sigma(f(\delta_{t-1})), \\ V_t = (1 - \frac{1}{\tau})V_{t-1}(1 - \delta_{t-1}) + SFB_t \odot I_t, \\ EI_t = \theta(g(V_t)), \\ \delta_t = EI_t \odot S(V_t). \end{cases} \tag{9}$$

Assuming that the output of the last layer is O_t at time t, for a given time window T the mean firing rate is $1/T \sum_{t=1}^{T} O_t$, y_i is the true label, and M is the number of samples, we use the mean-squared error loss function as shown in Eq. (10):

$$L = \frac{1}{M} \sum_{i=1}^{M} \|y_i - 1/T \sum_{t=1}^{T} O_t\|_2^2. \tag{10}$$

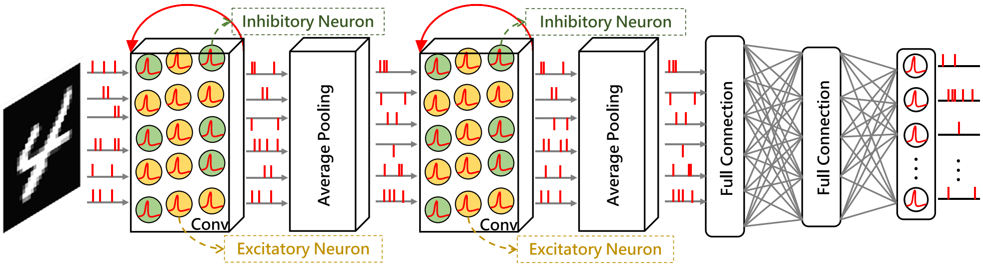

Fig. 4. The entire pipeline of our BackEISNN. The adaptive self-feedback mechanism is used to control the precise spiking of the membrane. The balanced excitatory and inhibitory neuron mechanism is used to help the layer release spikes in a balanced manner.

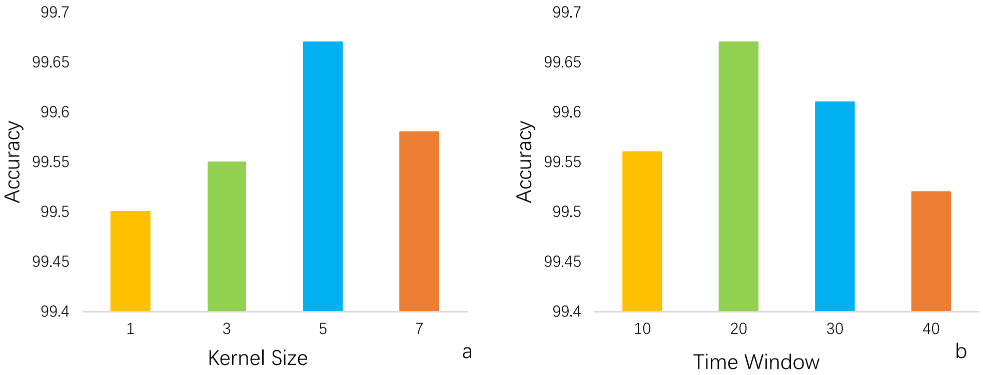

Fig. 5. (a) The test accuracy on the MNIST dataset with different sizes of the kernels of the two modules. (b) The test accuracy on the MNIST dataset with different time windows.

4. Experiments

In this section, we report experiments performed on a TITAN A100 GPU with the PyTorch framework (Paszke et al., 2019) to verify our proposed method. We initialized the network weights using the default method of PyTorch. The Adam optimizer (Kingma & Ba, 2014) was used, the batch size was set to 100, and the number of epochs was set to 200. The learning rate lr was set to 0.001, and was reduced by $0.1lr$ after every 40 epochs. We set $V_{th} = 0.5$. To illustrate the improvement brought about by them in the capability of the network for representation, we applied only the two modules mentioned above in the convolutional layers. We conducted the experiments on the MNIST (LeCun, 1998), N-MNIST (Orchard, Jayawant, Cohen, & Thakor, 2015), Fashion-MNIST (Xiao, Rasul, & Vollgraf, 2017), and CIFAR10 datasets to illustrate the superiority of our model.

4.1. MNIST

The MNIST dataset is the most widely used classification dataset to assess the performance of deep learning models. It consists of 60,000 training examples and 10,000 test samples describing handwritten digits from 0 to 9. The shape of each sample was 28 × 28. For the static MNIST dataset, we first converted the samples into spike trains. The normalized pixels were converted into 1 or 0 by our comparing each pixel value with a random number between 0 and 1. If the value of the pixel was greater than the random number, then a spike was triggered. The structure of our network was identical to that of ST-RSBP of Zhang and Li (2019), which uses two convolutional layers and two linear layers. However, in our experiments, we did not use the elastic distortion method for data augmentation. We fixed the time window to 20 and explored the impact of different sizes of kernels of the two modules on the performance of the network.

As shown in Fig. 5a, when the kernel size was set to 5, the network achieved the best performance. We then explored the impact of the time window on network performance with the kernel size fixed at 5. As shown in Fig. 5b, the network delivered the best performance when the time window T was 20.

The experiment was performed five times with different random number seeds by use of the above-mentioned hyperparameters. The test accuracy is shown in Fig. 6. Each category was well classified.

As shown in Table 1, several backpropagation-based SNNs were compared with our BackEISNN. Compared with the STBP algorithm, which does not use the SFBM and the BEIM, our method yielded an increase of 0.25 percentage points in terms of accuracy. It outperformed all the other methods as well.

4.2. N-MNIST

The N-MNIST dataset is a neuromorphic version of the MNIST dataset, obtained by mounting a dynamic version sensor in front of static images of digits on a computer screen. It contains 60,000 training samples and 10,000 test samples, like the MNIST dataset. The dynamic version sensor is moved in the direction of the three sides of an isosceles triangle in turn to collect the two-channel spike event triggered by a pixel change that lasts for 300 ms. Because of relative shifts in each image, each sample of the N-MNIST dataset is a spatial–temporal pattern with 34 × 34 × 2 spike sequences. As the time resolution was 1 μs, there were a total of 300,000 timesteps.

To speed up the simulation, we reduced it to 100 timesteps, which means that we recorded only one spike regardless of the number of events occurring in the 3000 intervals. We also tested the influence of different kernel sizes on performance. The network achieved its best performance when the kernel size was set to 3. The details are shown in Fig. 7.

The experiment was first conducted for 100 timesteps. As shown in Table 2, our BackEISNN-100 achieved an accuracy of 99.57%, delivering state-of-the-art performance. Moreover, we used the first 30 timesteps for another experiment and denote

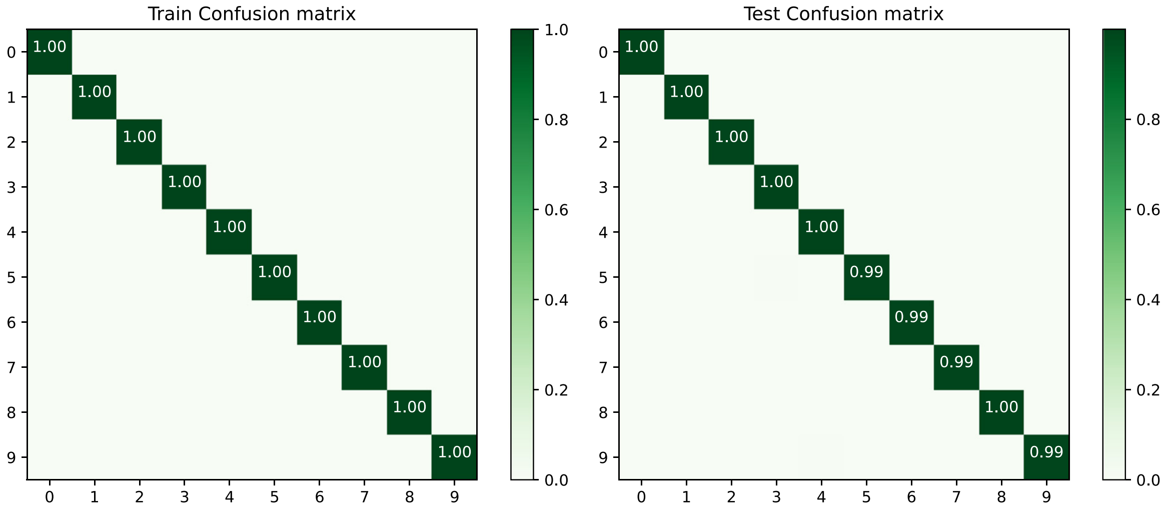

Fig. 6. Confusion matrix of BackEISNN on the MNIST dataset for MNIST handwritten digits 0 through 9 (as marked on the x-axis and y-axis scales). The left is for the training dataset, and the right is for the test dataset.

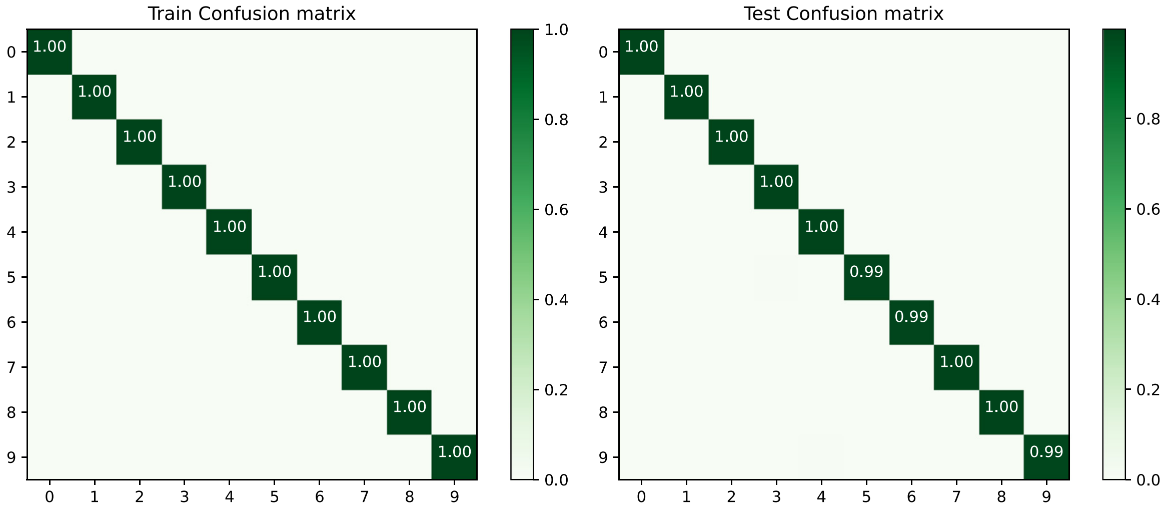

Fig. 7. Confusion matrix of BackEISNN on the N-MNIST dataset. The left is for the training dataset, and the right is for the test dataset.

Table 1
Performance of backpropagation-based spiking neural networks on the MNIST dataset.

Model	Structure	Mean accuracy (%)	Standard deviation	Best accuracy (%)
SLAYER (Shrestha & Orchard, 2018)	Net 1	99.36	0.05	99.41
STBP (Wu et al., 2018)	Net 2	–	–	99.42
HM2-BP (Jin et al., 2018)	Net 2	99.42	0.11	99.49
LISNN (Cheng et al., 2020)	Net3	–	–	99.5
TSSL-BP (Zhang & Li, 2020)	Net 2	99.5	0.02	99.53
ST-RSBP (Zhang & Li, 2019)	Net 2	99.57	0.04	99.62
BackEISNN (ours)	Net 2	99.58	0.06	99.67

Net 1 is 12C5-P2-64C5-p2. Net 2 is 15C5-P2-40C5-p2-300. Net 3 is 32C3-P2-32C3-P2-128.

the result by "BackEISNN-30". The proposed method maintained a high level of performance in this case as well.

4.3. Fashion-MNIST

The Fashion-MNIST dataset is a complicated challenge compared with the MNIST dataset. It consists of grayscale images of clothing. As in the case of the MNIST dataset, this dataset was processed for spike trains.

We also analyzed the impact of different kernel sizes on network performance. It delivered the best performance when the kernel size was 5. We then analyzed the influence of different time windows when the kernel size was fixed to 5. Our BackEISNN achieved the best performance in this case when the time window T was 30.

For the TSSL-BP model, the real-value input current was directly fed to the network. For comparison purposes, we also used the same input to demonstrate the influence of the kernel

Table 2
Performance of backpropagation-based spiking neural networks on the N-MNIST datset.

Model	Structure	Mean accuracy (%)	Standard deviation	Best accuracy (%)
HM2-BP (Jin et al., 2018)	400–400	–	–	98.88
SLAYER (Shrestha & Orchard, 2018)	Net 1	–	–	99.2
TSSL-BP 30 (Zhang & Li, 2020)	Net 1	99.23	0.05	99.28
TSSL-BP 100 (Zhang & Li, 2020)	Net 1	99.35	0.03	99.4
STBP (Wu, Deng et al., 2019)	Net 2	–	–	99.44
LISNN (Cheng et al., 2020)	Net 3	–	–	99.45
STBP NeuNorm (Wu, Deng et al., 2019)	Net 2	–	–	99.53
BackEISNN-30 (ours)	Net 1	99.47	0.05	99.56
BackEISNN-100 (ours)	Net 1	99.5	0.06	99.57

Net 1 is 12C5-P2-64C5-P2. Net 2 is 128C3-128C3-P2-128C3-256C3-P2-1024-Voting. Net 3 is 32C3-P2-32C3-P2-128.

Table 3
Performance of backpropagation-based spiking neural networks on the Fashion-MNIST dataset.

Model	Structure	Mean accuracy (%)	Standard deviation	Best accuracy (%)
HM2-BP (Jin et al., 2018)	400–400	–	–	88.99
GLSNN (Zhao et al., 2020)	256 × 8	–	–	89.02
ST-RSBP (Zhang & Li, 2019)	400-R400	–	–	90.13
LISNN (Cheng et al., 2020)	Net 1	–	–	92.07
BackEISNN-E (ours)	Net 2	92.56	0.08	92.7
TSSL-BP (Zhang & Li, 2020)	Net 2	92.69	0.09	92.83
BackEISNN-D (ours)	Net 2	93.04	0.31	93.45

Net 1 is 32C3-P2-32C3-P2-128. Net 2 is 32C5-P2-64C5-p2-1024.

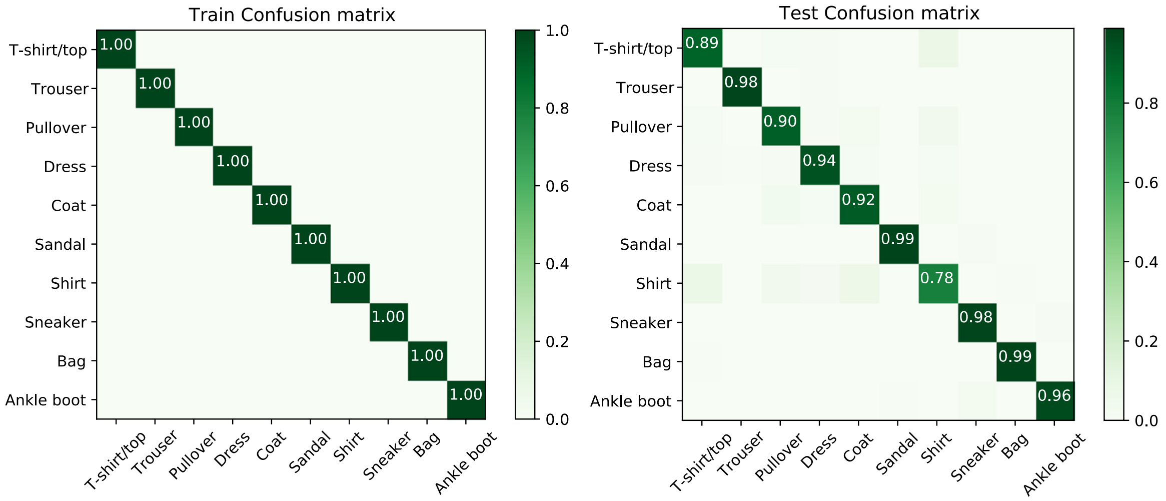

Fig. 8. Confusion matrix of BackEISNN on the Fashion-MNIST dataset. The left is for the training dataset, and the right is for the test dataset.

size and the time window on performance. We repeated the experiment as mentioned above. When the kernel size was 3 and the time window T was 20, the network achieved its best performance.

We performed the experiments five times using the parameters mentioned above as shown in Table 3. The result of the encoding input is denoted by BackEISNN-E and that of the direct input by BackEISNN-D.

To better analyze the results, we plotted a confusion matrix. As shown in Fig. 8, our BackEISNN could clearly identify each category in the training dataset. However, it was unable to distinguish between images of T-shirts and shirts on the test dataset because they look very similar, and even humans have trouble telling them apart.

4.4. CIFAR10

We also applied our BackEISNN to the CIFAR10 dataset. Compared with the MNIST and Fashion-MNIST datasets, the CIFAR10 dataset is a more challenging color image dataset, in which each image has a size of 32 × 32 × 3 pixels. It has 60,000 color images in 10 classes: 50,000 for training and 10,000 for testing. The dataset was normalized, randomly cropped, and horizontally flipped for data augmentation, which is common for preprocessing CIFAR10 images. We also used dropout after each layer, which is also used in the STBP and TSSL-BP methods. The network structure was 128C3-P2-256C3-P2-512C3-P2-1024, the kernel size of the two modules was set to 3, and the time window T was set to 20.

Table 4 shows that BackEISNN recorded an accuracy of 90.93%, which exceeded the accuracies of SNNs trained with backpropagation, decoded voting, and NeuNorm. The warm-up mechanism was used for the TSSL-BP method when very low levels of firing activity were observed. The network used the continuous sigmoid function of the membrane potential to approximate the activation so that errors could be propagated back when there was no spike to obtain a deeper SNN. The warm-up mechanism was not

Table 4
Performance of backpropagation-based spiking neural networks on the CIFAR10 dataset.

Model	Method	Accuracy (%)
STBP (Wu, Deng et al., 2019)	BP and Voting	89.83
STBP (Wu, Deng et al., 2019)	BP, Voting, and NeuNorm	90.53
BackEISNN (ours)	BP, SFBM, and BEIM	90.93
TSSL-BP (Zhang & Li, 2020)	Interneuron and Intraneuron BP	91.41

BP, backpropagation; BEIM, balanced excitatory and inhibitory neuron mechanism; SFBM, self-feedback mechanism.

Table 5
Accuracy of BackEISNN in ablation studies on different datasets.

Dataset	Baseline (%)	SFBM (%)	BEIM (%)	Both (%)
MNIST	99.42	99.56	99.59	99.67
N-MNIST	99.09	99.26	99.46	99.55
Fashion-MNIST	92.44	92.78	93.05	93.45
CIFAR10	82.6	88.86	89.32	90.93

BEIM, balanced excitatory and inhibitory neuron mechanism; SFBM, self-feedback mechanism.

used in our method. It also had a lighter structure, with three convolutional layers and two linear layers. By contrast, the TSSL-BP and STBP methods used deep networks with five convolutional layers each.

5. Discussion

To clarify the contributions of the two modules of our algorithm, we conducted ablation studies on the datasets mentioned above. Table 5 shows the performance of the baseline, the model with the adaptive SFBM introduced, the model with the BEIM introduced, and the model with both the adaptive SFBM and the BEIM introduced. From the MNIST dataset to the CIFAR10 dataset, as the complexity of the dataset increases, there is a gradual improvement in performance. For the CIFAR10 dataset, the accuracy of the baseline was only 82.6%. With the introduction of the SFBM, it increased to 88.86%, and rose to 89.32% with the introduction of the BEIM. With both mechanisms in place, our BackEISNN reached an accuracy of 90.93%, which shows that the combination of the two mechanisms improved the performance of the network.

We also plotted the test accuracies of the baseline, the baseline with the SFBM, the baseline with the BEIM, and the baseline with both the SFBM and the BEIM. Fig. 9 shows that with the introduction of the two mechanisms, the accuracy as well as the speed of convergence of our network increased.

To better illustrate the comparative improvement in performance due to the two modules in our method, we plotted the confusion matrix of the four models. As shown in Fig. 10, the original model without the SFBM and BEIM modules tended to confuse similar objects, such as an airplane and a bird, a cat and a dog, and a horse and a deer. However, when the two modules were introduced, the network was able to distinguish them more clearly. BackEISNN also recorded a higher accuracy of identification on images of other objects that are easy to distinguish. Thus, the introduction of the adaptive self-feedback module and the dynamically balanced excitatory and inhibitory neurons increased the accuracy of the network not only on categories of objects that are easy to distinguish but also on categories that are easy to confuse.

In addition, we randomly selected five samples from the CIFAR10 test dataset. They were cat, ship, automobile, airplane, and frog, representing the third, eighth, first, zeroth, and sixth categories, respectively. We visualized the spikes of the output layer for them. Fig. 11 shows that with the introduction of the SFBM and the BEIM, the spikes of the output layer were relatively accurate and stable at the beginning. However, for the network without these modules, the spikes of the output layer changed significantly. This shows that the introduction of the two modules helped control the stability of the network and the precise release of the spikes.

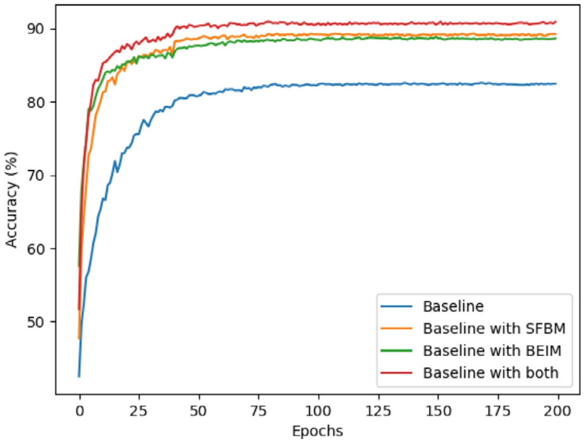

Fig. 9. The test accuracy of the proposed method on the CIFAR10 dataset with and without the self-feedback mechanism (SFBM) and the balanced excitatory and inhibitory neuron mechanism (BEIM).

6. Conclusion

A difference in performance persists between backpropagation-based SNNs and DNNs owing to the particular mode of data transmission of SNNs. This suggests that using only backpropagation is not sufficient to adequately train the network. Inspired by autapses in the brain that connect the soma in a self-feedback manner, we proposed applying time-delayed feedback to the neuron's membrane potential in this study. The input current was gated with information on the spikes at the previous timestep to regulate their precision and fully use the temporal information of the SNN. We also used a mechanism to balance excitatory and inhibitory neurons to dynamically control the form of the output of the spiking neurons. The introduction of the two mechanisms to SNNs trained with backpropagation accelerated the convergence of the network and increased its accuracy. Our proposed BackEISNN achieved state-of-the-art performance on the MNIST, Fashion-MNIST, and N-MNIST datasets. It also performed competitively against state-of-the-art SNNs on the CIFAR10 dataset in spite of its light structure.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program (Grant No. 2020AAA0104305), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB32070100).

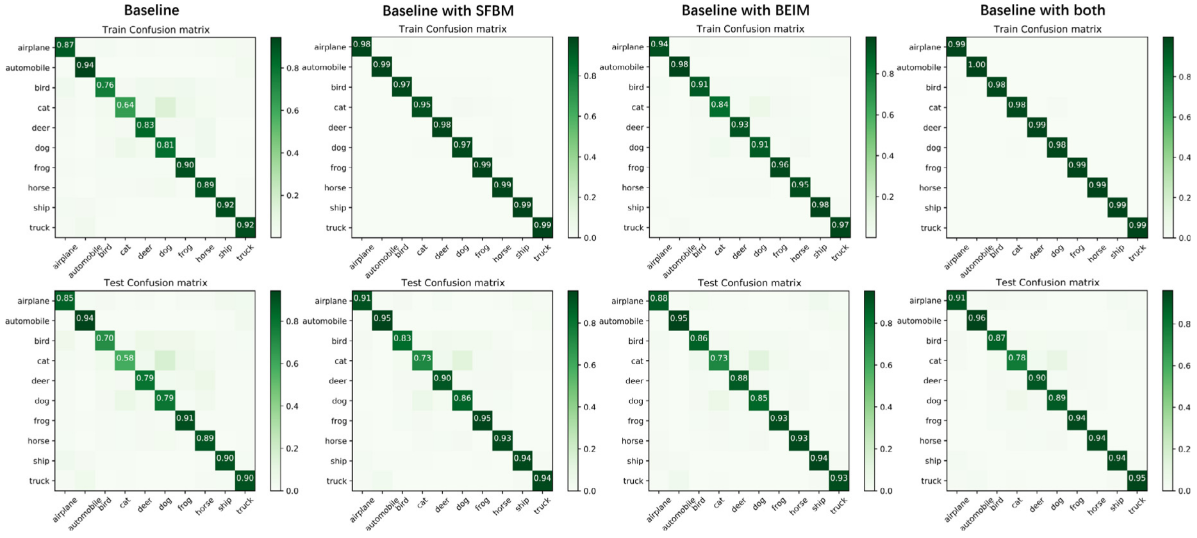

Fig. 10. Confusion matrix of BackEISNN with the self-feedback mechanism (SFBM), the balanced excitatory and inhibitory neuron mechanism (BEIM), and both mechanisms on the CIFAR10 dataset. The top row is for the training dataset, and the bottom row is for the test dataset.

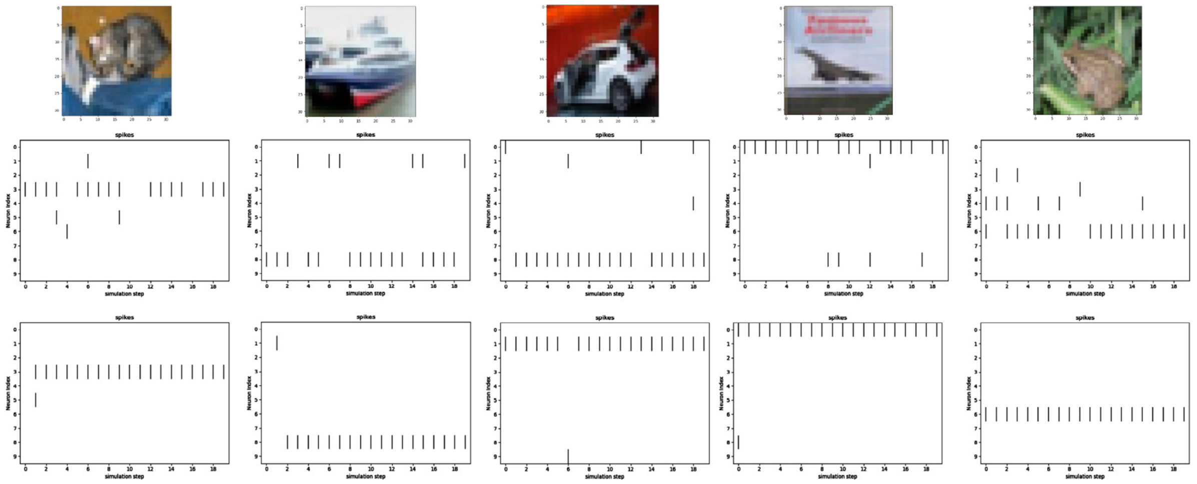

Fig. 11. The spikes of the output layer for five randomly selected samples from the CIFAR10 test dataset. The second row indicates the results of the network without the two modules. The third row indicates the results with the two modules.

References

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., et al. (2016). Deep speech 2: End-to-end speech recognition in English and Mandarin. In *International conference on machine learning* (pp. 173–182). PMLR.

Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., & Maass, W. (2018). Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in Neural Information Processing Systems*, *31*.

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.

Bohte, S. M., Kok, J. N., & La Poutré, J. A. (2000). SpikeProp: backpropagation for networks of spiking neurons. In *ESANN* (pp. 419–424).

Cheng, X., Hao, Y., Xu, J., & Xu, B. (2020). LISNN: Improving spiking neural networks with lateral interactions for robust object recognition. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence* (pp. 1519–1525).

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on machine learning* (pp. 160–167).

Dehghani, N., Peyrache, A., Telenczuk, B., Le Van Quyen, M., Halgren, E., Cash, S. S., et al. (2016). Dynamic balance of excitation and inhibition in human and monkey neocortex. *Scientific Reports*, *6*(1), 1–12.

Diehl, P. U., & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in Computational Neuroscience*, *9*, 99.

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 *2015 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Falez, P., Tirilly, P., Bilasco, I. M., Devienne, P., & Boulet, P. (2019). Multi-layered spiking neural network with target timestamp threshold adaptation and STDP. In *2019 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Felleman, D. J., & Van Essen, D. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cerebral Cortex (New York, NY: 1991)*, *1*(1), 1–47.

Hu, Y., Tang, H., & Pan, G. (2018). Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Ikeda, K., & Bekkers, J. M. (2006). Autapses. *Current Biology*, *16*(9), R308.

Jin, Y., Zhang, W., & Li, P. (2018). Hybrid macro/micro level backpropagation for training deep spiking neural networks. In *Advances in neural information processing systems* (pp. 7005–7015).

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90.

LeCun, Y. (1998). The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

Lee, J. H., Delbruck, T., & Pfeiffer, M. (2016). Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, *10*, 508.

Maass, W. (1997). Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, *10*(9), 1659–1671.

Machingal, P., Thousif, M., Dora, S., & Sundaram, S. (2020). Self-regulated learning algorithm for distributed coding based spiking neural classifier. In *2020 international joint conference on neural networks (IJCNN)* (pp. 1–7). IEEE.

O'Connor, P., & Welling, M. (2016). Deep spiking networks. arXiv preprint arXiv:1602.08323.

Orchard, G., Jayawant, A., Cohen, G. K., & Thakor, N. (2015). Converting static image datasets to spiking neuromorphic datasets using saccades. *Frontiers in Neuroscience*, *9*, 437.

Pan, Z., Wu, J., Zhang, M., Li, H., & Chua, Y. (2019). Neural population coding for effective temporal classification. In *2019 international joint conference on neural networks (IJCNN)* (pp. 1–8). IEEE.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*.

Rubin, R., Abbott, L., & Sompolinsky, H. (2017). Balanced excitation and inhibition are required for high-capacity, noise-robust neuronal selectivity. *Proceedings of the National Academy of Sciences*, *114*(44), E9366–E9375.

Sengupta, A., Ye, Y., Wang, R., Liu, C., & Roy, K. (2019). Going deeper in spiking neural networks: VGG and residual architectures. *Frontiers in Neuroscience*, *13*.

Shrestha, S. B., & Orchard, G. (2018). SLAYER: Spike layer error reassignment in time. *Advances in Neural Information Processing Systems*, *31*.

Sporns, O., & Zwi, J. D. (2004). The small world of the cerebral cortex. *Neuroinformatics*, *2*(2), 145–162.

Tavanaei, A., & Maida, A. S. (2016). Bio-inspired spiking convolutional neural network using layer-wise sparse coding and STDP learning. arXiv preprint arXiv:1611.03000.

Tavanaei, A., & Maida, A. S. (2017). Multi-layer unsupervised learning in a spiking convolutional neural network. In *2017 international joint conference on neural networks (IJCNN)* (pp. 2023–2030). IEEE.

Thiele, J. C., Bichler, O., & Dupret, A. (2019). SpikeGrad: An ANN-equivalent computation model for implementing backpropagation with spikes. In *International conference on learning representations*.

Wang, C., Guo, S., Xu, Y., Ma, J., Tang, J., Alzahrani, F., et al. (2017). Formation of autapse connected to neuron and its biological function. *Complexity*, *2017*, Article 5436737.

Wu, J., Chua, Y., Zhang, M., Yang, Q., Li, G., & Li, H. (2019). Deep spiking neural network with spike count based learning rule. In *2019 international joint conference on neural networks (IJCNN)* (pp. 1–6). IEEE.

Wu, Y., Deng, L., Li, G., Zhu, J., & Shi, L. (2018). Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, *12*.

Wu, Y., Deng, L., Li, G., Zhu, J., Xie, Y., & Shi, L. (2019). Direct training for spiking neural networks: Faster, larger, better. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 1311–1318).

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). CSNN: An augmented spiking based framework with perceptron-inception. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence* (pp. 1646–1652).

Yin, L., Zheng, R., Ke, W., He, Q., Zhang, Y., Li, J., et al. (2018). Autapses enhance bursting and coincidence detection in neocortical pyramidal cells. *Nature Communications*, *9*(1), 1–12.

Zhang, W., & Li, P. (2019). Spike-train level backpropagation for training deep recurrent spiking neural networks. *Advances in Neural Information Processing Systems*, *32*.

Zhang, W., & Li, P. (2020). Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Advances in Neural Information Processing Systems*, *33*, 12022–12033.

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*.

Zhang, T., Zeng, Y., Zhao, D., & Xu, B. (2018). Brain-inspired balanced tuning for spiking neural networks. In *Proceedings of the twenty-seventh international joint conference on artificial intelligence* (pp. 1653–1659).

Zhao, D., Zeng, Y., Zhang, T., Shi, M., & Zhao, F. (2020). GLSNN: A multi-layer spiking neural network based on global feedback alignment and local STDP plasticity. *Frontiers in Computational Neuroscience*, *14*, Article 576841.

Zhu, J., Jiang, M., Yang, M., Hou, H., & Shu, Y. (2011). Membrane potential-dependent modulation of recurrent inhibition in rat neocortex. *PLoS Biology*, *9*(3), Article e1001032.