Neurocomputing 365 (2019) 102–112

Contents lists available at ScienceDirect

Neurocomputing

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

Fast and robust learning in Spiking Feed-forward Neural Networks based on Intrinsic Plasticity mechanism

Anguo Zhang[a,b], Hongjun Zhou[c], Xiumin Li[a,*], Wei Zhu[b]

[a] College of Automation, Chongqing University, Chongqing, 400044 China
[b] Research Institute of Ruijie, Ruijie Networks Co., Lid, Fuzhou, 350002 China
[c] School of Economics and Business Administration, Chongqing University, Chongqing 400044, China

ARTICLE INFO

Article history:
Received 25 October 2018
Revised 2 May 2019
Accepted 14 July 2019
Available online 26 July 2019

Communicated by Prof. Zidong Wang

MSC:
00-01
99-00

Keywords:
Intrinsic Plasticity
Feed-forward Neural Network
Spiking neuron model
Fast and robust learning

ABSTRACT

In this paper, the computational performance of a Spiking Feed-forward Neural Network (SFNN) is investigated based on a brain-inspired Intrinsic Plasticity (IP) mechanism, which is a membrane potential adaptive tuning scheme used to change the intrinsic excitability of individual neuron. This learning rule has the ability of regulating neural activity in a relative homeostatic level even if the external input of a neuron is extremely low or extremely high. The effectiveness of IP on SFNN model has been studied and evaluated through the MNIST handwritten digits classification. The training of network weights starts from a conventional artificial neural network by backpropagation and then the rate-based neurons are transformed into spiking neuron models with IP learning. Our results show that both over-activation and under-activation of neuronal response which commonly exist during the computation of neural networks can be effectively avoided. Without loss of accuracy, the calculation speed of SFNN with IP learning is extremely higher than that of the other models. Besides, when the input intensity and data noise are taken into account, both of the learning speed and accuracy of the model can be greatly improved by the application of IP learning. This biologically inspired SFNN model is simple and effective which may give insights for the optimization of neural computation.

1. Introduction

In recent decade years, with the rapid development of theories and applications in brain science, spiking neural networks (SNNs) are proposed and have become increasingly attractive due to its advantages of high nonlinear approximation capability and low computation consumption. Although analog artificial neural networks with deep learning can achieve high classification accuracy on many computational applications, the huge number of neurons and complicate algorithms require massive computing power and energy. To overcome the large computational cost of deep network in artificial neural networks, spiking deep networks (SDNs), i.e. deep neural networks with spiking neurons have been proposed recently. SDNs present high performance for solving various problems, they not only achieve powerful computing capability and high accuracy, but also gain high processing speed and low computational cost [1–3]. Besides, SDNs have the potential to be directly applied to event-based computing systems, i.e. the neuromorphic circuits which have been shown to be significantly more energy-efficient than conventional ones [4]. It has become an increasingly attractive field for researchers and engineers [5–8]. In [9], Rossum et al. modeled the propagation of neural activities through a feed-forward network consisting of integrate-and-firing (I&F) neurons. They found that firing activities are transmitted linearly layer by layer, with a time delay proportional to the synaptic time constant. In [10] the authors discussed the convergence problem of leaky integrate-and firing (LIF) neuron model based on spiking recurrent networks with constant external inputs, and suggested that precise spatiotemporal sequences of spikes may be useful for information encoding and processing in biological neural networks. Further, in [11], Jin et al. also exhibited another spiking recurrent network that performed fast winner-take-all computations, which aimed to reducing the inter-spike intervals of the neurons.

Since traditional training algorithms (i.e. Error Back Propagation) of rate-based analog neural networks cannot be directly used for SNNs. Some specific algorithms have been proposed for obtaining a well-trained SNN. Similar to the BP learning rule, Bohte et al. derived a supervised learning method (SpikeProp) for spike response model (SRM) [12]. In [13], the authors introduced a new error-backpropagation mechanism based on SNN training

* Corresponding author.
E-mail addresses: anrial@live.cn (A. Zhang), hjzhou@cqu.edu.cn (H. Zhou), xmli@cqu.edu.cn (X. Li), ruilangzhu@icloud.com (W. Zhu).

https://doi.org/10.1016/j.neucom.2019.07.009
0925-2312/

algorithm, which treats the membrane potentials of spiking neurons as differentiable signals, where discontinuities at spike times are only considered as noise. Other backpropagation-inspired learning rules have also been proposed, e.g., [14–18]. On the other hand, unsupervised learning with respect to neural synaptic plasticity, especially the spike-timing-dependent-plasticity (STDP), has also been broadly studied. According to STDP, synapses through which a presynaptic spike arrived before (respectively after) a postsynaptic one are reinforced (respectively depressed) [19]. Inspired by the success and popularity of reinforcement learning, Kheradpisheh et al. proposed a reward modulated STDP (R-STDP) method to train the convolutional SNN for object recognition, and it demonstrated that R-STDP outperforms STDP on various image datasets [20].

However, the applications of the above SNNs models are very limited compared with the broadly used deep learning rate-based neural networks (ANNs). Directly converting the ANNs to be SNNs could be more effective and convenient for applications. But the conversion from analog neural networks to event-driven spiking neurons may cause the cost of performance losses. Errors are mostly caused by the over-activation of neural responses to input data. The probability of converting the well-trained rate-based ANNs to their SNN counterparts via directly mapping the connection weights was firstly proposed by [21,22]. OConnor et al. presented a mathematical framework and algorithm for converting a conventional deep belief model trained offline into the event-based spiking DBNs, and shows the application of the framework with such as real visual and auditory inputs from neuromorphic sensors [21]. By this approach, one can train the tailored ANN model through mature optimization algorithms, such as stochaotic gradient descent (SGD), Adam, and so on. And then, simply convert it to its corresponding SNN version. In [23] the authors extended the conversion method of [22] to reduce the performance loss during the conversion, further, they also proposed a set of optimization techniques for spiking convolutional neural networks (SCNNs) and spiking feed-forward neural networks (SFNNs), including using rectified linear units (ReLUs) with no bias during training and weight normalization method. Compared with previous approaches based on SNNs, these optimization techniques boost the computation performance that perform the state-of-art both in high accuracy and low latency. Despite the outstanding performance in [23], it should be noted that it is still time consuming and difficult to reach the optimal performance. Thus it is essential to find the right balance of spiking thresholds, input weight and input firing rates. More systematically, Rueckauer et al. introduced more conversion methods for some ANN operators to their SNN counterparts [24]. Further works of deep neural networks (DNNs) to spiking DNNs conversion have also been done, such as [25,26].

Actually, in neuroscience, it has been confirmed that a single neuron can adaptively change its intrinsic excitability to fit various synaptic inputs by the activity of their voltage-gated channels [27,28]. This adaption of neuronal intrinsic excitability, called intrinsic plasticity (IP), has been hypothesized to keep the mean firing activity of neuronal population in a homeostatic level which is effective to maximize the information entropy [29–31]. That is, IP can change the intrinsic excitability of individual neuron by adaptively turning the firing threshold. This adaptive adjustment of the neuronal input-output response online is crucial for efficient information processing. In [32], Marion et al. showed that IP is able to adapts reservoir computing with very constrained topologies and make the networks more robust. However, IP learning is not involved in most of current studies on computational modeling of neural network.

Therefore, inspired by this biologically discovered learning rule, we present a novel neural network model by combining the IP learning scheme with spiking feed-forward neural networks (SFNNs) in this paper. Experimental results tested on MNIST database show that compared with the networks without IP, both the learning speed and robustness of computation accuracy are obviously improved. A neuron with IP mechanism will weaken its excitability when the input is boosted and strengthen its excitability when the input is deprived. Thus, the neuron can stay at a homeostatic state no matter the synaptic input unexpected strongly increases or decreases, which protects the neuron activity against to an extremely high or low level. In this way, both over-activation and under-activation of neuronal response can be effectively avoided in time and ensure the accuracy of computation. This adaptive adjustment of spiking threshold is simple and effective, with no need to turn the right balance of parameters.

The remaining of this paper is organized as follows. Section 2 introduces the FNNs based on spiking neuron model (LIF model), and the corresponding IP mechanism for SNNs. Section 3 demonstrates the dataset benchmark used in this paper, as well as the SFNN construction method. Section 4 implements the experiment constructed on both normal MNIST dataset and MNIST dataset with different input intensity and noise intensity. Influences of key parameters are also fully investigated and analyzed. Section 5 discusses the findings and conclusions.

2. Model description

Based on the mathematical framework for converting a conventional deep neural networks into spiking neural networks in [21,23], training of spiking deep networks can start from a conventional ANN trained with back propagation and then transform the rate-based neuron model into spiking neuron model. In the following two subsections, feed-forward neural network for training process and spiking feed-forward neural network for testing process will be described in detail, which produce the rate-based FNN and spiking FNN (SFNN) respectively.

2.1. Feed-forward neural network for training process

In feed-forward neural networks (FNNs), neurons of the preceding layer are fully connected to the subsequent layer, while no intra-layer connections are established. FNNs have been extensively applied in image classification, trend prediction, scene labeling and other tasks. Nowadays, another most popular application of FNNs is acting as the final output layer of convolutional neural networks (CNNs), where higher-level feature extraction is achieved by the convolutional layers, pooling layers and normalization layers. In conventional FNNs, or named the rate-based FNNs, neurons are linear static units which directly sums the weighted input signals from previous layer and passes the obtained value to the following layer. For image classification problems, activation functions of rectified linear unit (ReLU) and softmax have been widely used for intermediate feature processing in hidden layers and image classification in output layer, respectively.

Nair and Hinton [33] presented that the stage function rectified linear units (ReLUs) greatly accelerate the convergence of stochastic gradient descent compared to the sigmoid/tanh functions due to its linear, non-saturating form. And considering the enhancement of ReLUs to computation performance for object recognition and face verification, we use ReLUs in our networks during the training process. A typical mathematical description is given as

$$x_i = max(0, \sum_j w_{ij} x_j) \tag{1}$$

where x_i is the activation of unit i, w_{ij} is the weight connecting unit j in the preceding layer to unit i in the current layer, and x_j is the activation of unit j in the preceding layer. It should be noted that the variable x_j in SFNNs represents the spike event of neuron

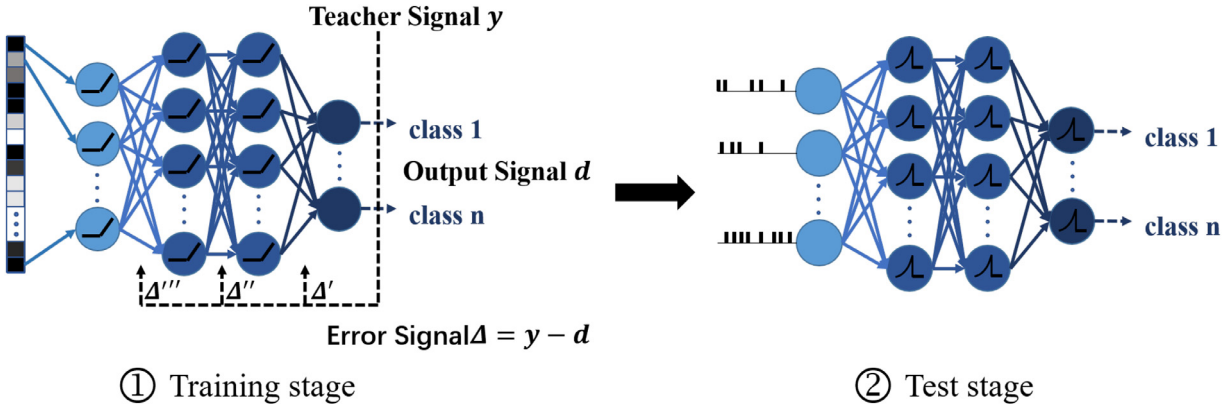

Fig. 1. Schematic diagram of converting an artificial neural network into a spiking neural network. ①: Training an analog FNN with ReLU activation function based on classical Back Propagation learning rule. ②: Test stage with the replacement of the ReLU neuron model with spiking neuron model, and input images are transferred into spike streams.

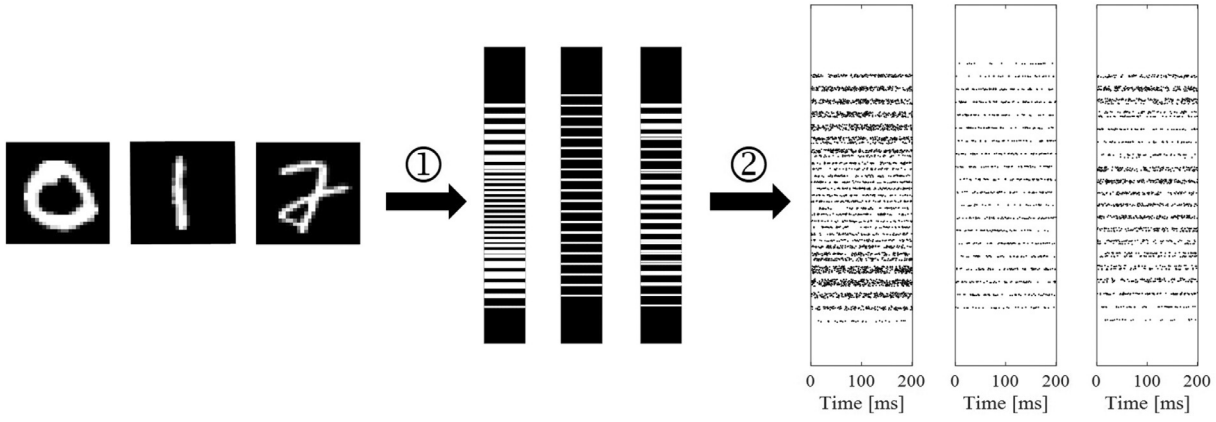

Fig. 2. Conversion from image pixels into spike streams. ①: Vectorize the 28*28 image pixels into 784 input neurons with two states of firing or no firing. The "white line" in spike streams denotes a spike event at current time step, while the "black line" denotes no spike generated. ②: Generate spike streams with Poisson distribution, where the firing frequency of spike stream is proportional to the image intensity.

j at a certain time step, which implies that $x_j = 1$ if a spike generated by neuron j, otherwise $x_j = 0$. ~~and~~ x_i is the synaptic current input produced by its presynaptic.

2.2. Spiking feed-forward neural network with intrinsic plasticity for testing process

Unlike the traditional FNNs, spiking feed-forward neural networks (SFNNs) only process signal in the format of spiking events. Fig. 1 shows a schematic diagram of image classification conducted by a SFNN framework. In the data preprocessing stage, image pixels are firstly reshaped to a vector, and then converted into a sequence of spike streams, which are used to feed into the input layer of SFNNs. Fig. 2 shows the conversion process where each image pixel is converted into a spike stream by using the Poisson distribution method [23,34]. The firing frequency of spike stream is proportional to the image intensity.

The Leaky Integrate-and-Fire (LIF) model [35] is used in this paper, which was described to be computationally effective for neural activities and widely deployed in artificial computational neuroscience. LIF model can be given as the differential equation,

$$\tau_m \frac{dV(t)}{dt} = V_{rest} - V(t) + R_m I(t) \tag{2}$$

where $V(t)$ denotes the membrane potential of a single neuron. V_{rest} denotes the resting membrane potential. The time constant τ_m is the product of membrane resistance R_m and membrane capacitance C_m. $I(t)$ is the sum of all inputs from pre-synaptic neurons. Each LIF neuron is stimulated by the activity of its pre-synaptic connected neurons.

$$I_j^{\phi}(t) = \sum_{i=1}^{N} S_i^{\phi-1}(t) \cdot w_{ji}^{\phi-1} \tag{3}$$

The variable ϕ means the ϕth layer of neural network. $S_i(t)$ represents the spiking of neuron i at time t. $S_i(t) = 1$ if this neuron generates a spike, otherwise $S_i(t) = 0$. When a neuron generates a spike, its membrane potential will be reset to V_{rest} in the next step.

A spiking IP learning rule for IF neuron model is proposed in [36]. This IP rule depends on Dirac delta function, which had the format of

$$\begin{aligned} \tau_{ip}\frac{drC}{dt} &= \frac{1}{rC} - yI + \beta(1-y)I \\ \tau_{ip}\frac{drR}{dt} &= -rR + y - \beta(1-y) \end{aligned} \tag{4}$$

where τ_{ip} denotes the relative integration resolution of the IF model and the IP mechanism, β is a scale factor. $rC = 1/C_m$, $rR = 1/R_m$, and the output in response to the input I of a neuron can be described as

$$y = \eta \sum_f \delta(t - t^{(f)}) \tag{5}$$

where $\delta(t - t^{(f)})$ is Dirac delta function representing a spike fired at time $t^{(f)}$ and η indicates the strength of spike. The tuning curve

of an IF neuron can be shifted due to the changing of two parameters rC and rR, which regulate the slope and threshold of the response curve.

The discrete version of synaptic current input $I_j^{\phi}(k)$ and neuron membrane potential update can be written as

$$I_j^{\phi}(k) = \sum_{i=1}^{N} S_i^{\phi-1}(k) \cdot w_{ji}^{\phi-1} \tag{6}$$

and

$$\Delta V(k) = \frac{1}{\tau_m(k-1)}(V_{rest} - V(k-1) + R_m(k-1)I(k)) \tag{7}$$

, the output at time step k, $y(k)$ is

$$y(k) = \eta \sum_f \delta(t(k) - t^{(f)}(k)) \tag{8}$$

According to (4), the parameters of $\tau_m(k)$ and $R_m(k)$ can be updated by the IP rule:

$$\begin{aligned} \tau_m(k) &= 1/rC(k)rR(k) \\ R_m(k) &= 1/rR(k) \end{aligned} \tag{9}$$

where

$$\begin{aligned} \Delta rC(k) &= \frac{1}{\tau_{ip}}\left(\frac{1}{rC(k-1)} - y(k)I(k) + \beta(1-y(k))I(k)\right) \\ \Delta rR(k) &= \frac{1}{\tau_{ip}}(-rR(k-1) + y(k) - \beta(1-y(k))) \end{aligned} \tag{10}$$

3. Material and method

3.1. Dataset

To verify the proposed IP method, a series of experiments on the pattern recognition have been conducted based on the MNIST benchmark. MNIST is a widely used digit character database for machine learning which consists of 60,000 training samples and 10,000 test samples. Each sample is a handwriting digit from 0 to 9 with the pixel size of 28×28, where the pixel is valued by gray value of 0–255.

Besides the normal MNIST dataset which is used to test the learning speed and accuracy, manually processed MNIST test datasets are also used to test the robustness of the proposed IP method. We perform two treatments on the test dataset by changing the exposure degree of the input test images or adding noise with different types and different intensities to the test images, including uniform noise, Gaussian noise, Rayleigh noise, gamma noise, and salt&pepper noise.

3.2. SFNN model construction

Based on the mathematical framework and experiment results for converting a conventional deep neural networks into spiking neural networks which proposed by [21,23–25], training of spiking deep networks can start from a conventional ANN trained with back propagation and then transform the rate-based neuron model into spiking neuron model. As shown in Fig. 1, we construct the SFNN by the following several steps:

1) Construct a deep fully connected network with 4 layers of scale 784-1200-1200-10, which consists of one input layer (784 neuron nodes), two hidden layers (1200 neuron nodes per layer), and an output layer (10 neuron nodes with each node represents one of the ten digit classes); Apply the ReLU model as the activation functions of neurons in hidden layers and softmax in the output layer;

2) Set the bias to be 0 and train the conventional analog FNN via BP training rule;

Table 1
List of parameter values.

Parameter	Value	Parameter	Value
V_{th}	1	V_{rest}	0
β	0.6	η	0.5
τ_{ip}	20	rR	2
rC	2		

3) Replace the ReLU units with the LIF model in the hidden layers and directly map the weights from the well-trained analog FNN network to the spiking network;

4) Add IP mechanism to the LIF units during the test stage.

The computational performance in terms of learning speed and classification accuracy are compared between SFNN with IP rule (SFNN-IP) and SFNN without IP rule (SFNN-noIP) under different input firing rates. Besides, we also study the computational performance of SFNN-noIP and SFNN-noIP on ill-exposed dataset and noisy dataset, respectively. It is known that ill-posed and noisy images are usually difficult to be accurately and reliably classified especially when the model is not been well-trained with these ill-posed datasets. Thus, it is meaningful to develop robust algorithms or models for recognizing disturbed images. Initial values of some important parameters are listed in Table 1, and the detailed investigations on parameters selection are discussed in Section 4.4.

4. Experimental results

4.1. Classification performance on normal dataset

4.1.1. Convergence speed

The calculation speed, as well as the learning speed, refers to the time taken by SNN to recognize the image pattern that carried by the input signal. It can be also considered as the convergence time for SNN to the homeostatic state of spiking activity. It is believed that shorter convergence time to homeostatic state means the faster learning speed, thus networks can respond more quickly to the input signals, which is critical for computational tasks with high real-time requirements. Normalization methods, such as batch normalization [37], are popular for accelerating training process and improving computing accuracy of analog neural networks. For spiking networks, In [23] the authors proposed model-based and data-based normalization methods for achieving fast classifying while maintaining high accuracy during the test (inference) stage. Similar as the experiments conducted in [23], in this paper the well-trained weights of ReLU network are directly transferred to a spiking IF network. However, instead of using the model-based or data-based normalization to avoid over-activation, here IP mechanism is applied to the IF units during the testing procedure to self-adjust the balance of input intensity and output firing rate. Fig. 3 compares the computational performance of several models, i.e. SFNN with or without IP learning, and SFNN normalized by the two methods proposed in [23]. We can see that without loss of accuracy, the time required to achieve the final accuracy of SFNN-IP is significantly shorter than SFNN-noIP and the models based on data and model normalization. The effects of firing rate of the input layer (called input firing rate in the following content) on performance of the spiking neural networks are also investigated in Fig. 4. We can see that SFNN-IP shows faster convergence speed than SFNN-noIP without loss of accuracy, especially for the case of low input firing rate.

In order to examine the influence of hidden layer numbers on the computational performance, we tested deeper SFNN models with three and four hidden layers under the structure of 784-1200-1200-800-10 and 784-1200-1200-800-800-10. The classification accuracy of FNNs with 2, 3, 4 hidden layers before converting

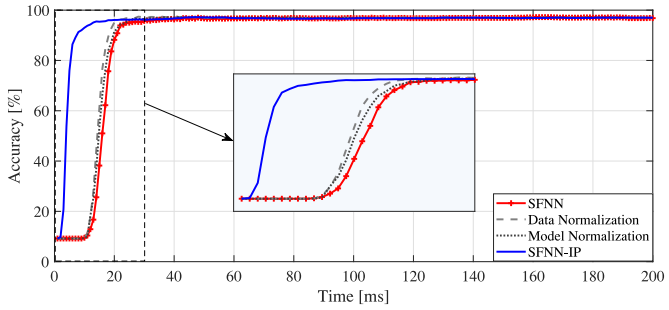

Fig. 3. Comparisons of classification performance on MNIST dataset between the models of SFNN-noIP, SFNN-IP and SFNN with data and model normalization method proposed in [23].

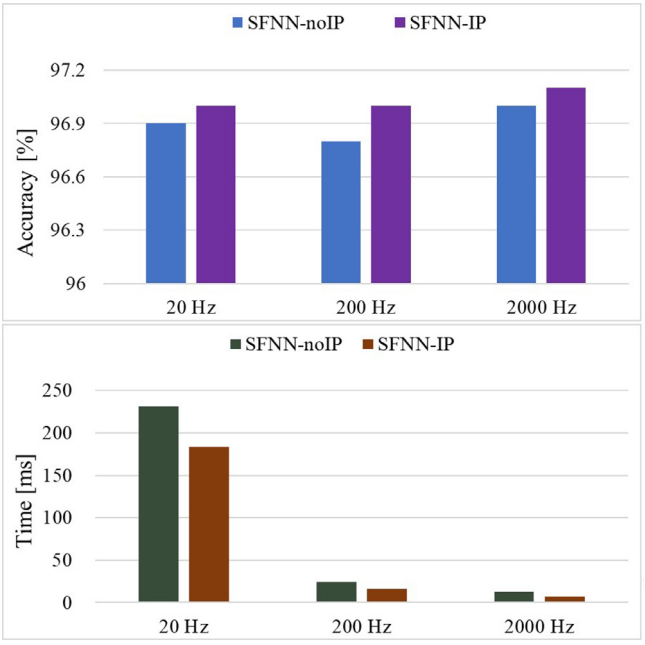

Fig. 4. Comparisons of the learning speed and accuracy between SFNN-noIP and SFNN-IP with the input rate of 20 Hz, 200 Hz and 2000 Hz.

Table 2
Computational performance comparisons between SFNN-IPs and SFNN-noIP with two, three and four hidden layers, respectively.

Network size	Model	Accuracy	Time
2 hidden layers	SFNN-noIP	96.8%	26 ms
	SFNN-IP	97.0%	14 ms
3 hidden layers	SFNN-noIP	98.2%	29 ms
	SFNN-IP	98.3%	18 ms
4 hidden layers	SFNN-noIP	98.4%	34 ms
	SFNN-IP	98.4%	21 ms

to the SFNN are 97.2%, 98.4% and 98.6%, respectively. As shown in Table 2, we can see that computational accuracy and convergence time for both SFNN-IP and SFNN-noIP increase slightly with the increase of hidden layers. Since the change is not obvious, SFNN with two hidden layers is enough for this computational task.

4.1.2. Firing rates

The spiking activities during the initial 100 ms in the hidden layers and output layer are shown in Fig. 5(a)–(c). It can be seen that the time of output response appears much earlier than the SFNN-noIP network, which means that the convergence speed is greatly accelerated (see the red vertical lines in Fig. 5). Due to the self-adjusting of firing threshold in the IP learning, single unit has the ability of adapting its intrinsic excitability to its synaptic inputs dynamically. Therefore during the initial stage, the whole population can be quickly triggered to make responses which contributing to the significant acceleration of recognition speed. Note that there is almost no neural response for the SFNN-noIP network when input rate is extremely weak (for example 20 Hz) within the initial 100ms, but IP learning can effectively avoid this non-activation of neural response to weak input.

Moreover, Fig. 4 implies that high input firing rate (for example 2000 Hz) can accelerate the learning speed of both SFNN-noIP and SFNN-IP models. However, as shown in the subsequent experiments for the ill-conditioned test data, the robustness of the spiking network with high input firing rate is reduced. With high input firing rates, SFNN turns to be more sensitive to image intensity and noise disturbance. To further confirm this point, the influence of input firing rate, image intensity and noise robustness on the recognition performance are investigated in the following content.

4.1.3. rR and rC of IP

Here we investigate the variation of the key variables rR and rC in the IP learning rule. Fig. 6 shows the rR and rC of a single neuron which is driven by a random sparse spiking trains and an external current input. It should be pointed out that event-triggered signal in Fig. 6 is an example for showing the dynamic process of rR and rC. The practical variations of rR and rC for MNIST image classification task are shown in Fig. 7, where the "Layer2" and "Layer3" denote the two hidden layers of SFNN (the network structure is 784-1200-1200-10, but only 400 hidden neuron states are represented), and "Output" is the output layer (10 neuron states). From Fig. 7 we can see that the values of rR decrease while the values of rC increases for both of the hidden layers and the output layer. The updating of rR and rC can be simplified by using the last state of rR and rC according to the degraded dynamics with no spiking response output y and no external input current I, which is formulated as

$$\tau_{ip}\frac{drC}{dt} = \frac{1}{rC}$$
$$\tau_{ip}\frac{drR}{dt} = -rR - \beta \tag{11}$$

Based on 11, it is known that if the initial values of rR and rC are greater than 0, and $\tau_{ip} > 0$, $\beta > 0$, then rR will converge to a fixed value, while rC will continuously increase over time. It should be noted that the variations of the rR, rC in Figs. 6 and 7 are consistent with the results in 11. On the other hand, it is important to notice that in the output layer, both of rR and rC of the output neuron which represented the image class evolves significantly different from the other neurons. It is the remarkable dynamical changes of the corresponding output neuron which significantly contribute to the quick convergence process during the test stage of the SFNNs.

4.2. Classification performance on ill-exposed dataset

The problem of inconsistency between the feature distributions of the test dataset and training dataset is a common phenomenon in some practical applications. For example, in the same scene, the overall pixel brightness of the image taken in the dark night is significantly lower than that taken in the strong sunlight. If a neural network model is trained for image recognition tasks using image samples taken from normal light brightness, it can not work well for recognizing images taken under different light conditions. In the vast majority of engineering applications, it is necessary to sample the training dataset so that to cover all the cases as much as possible, or use complex normalization techniques which

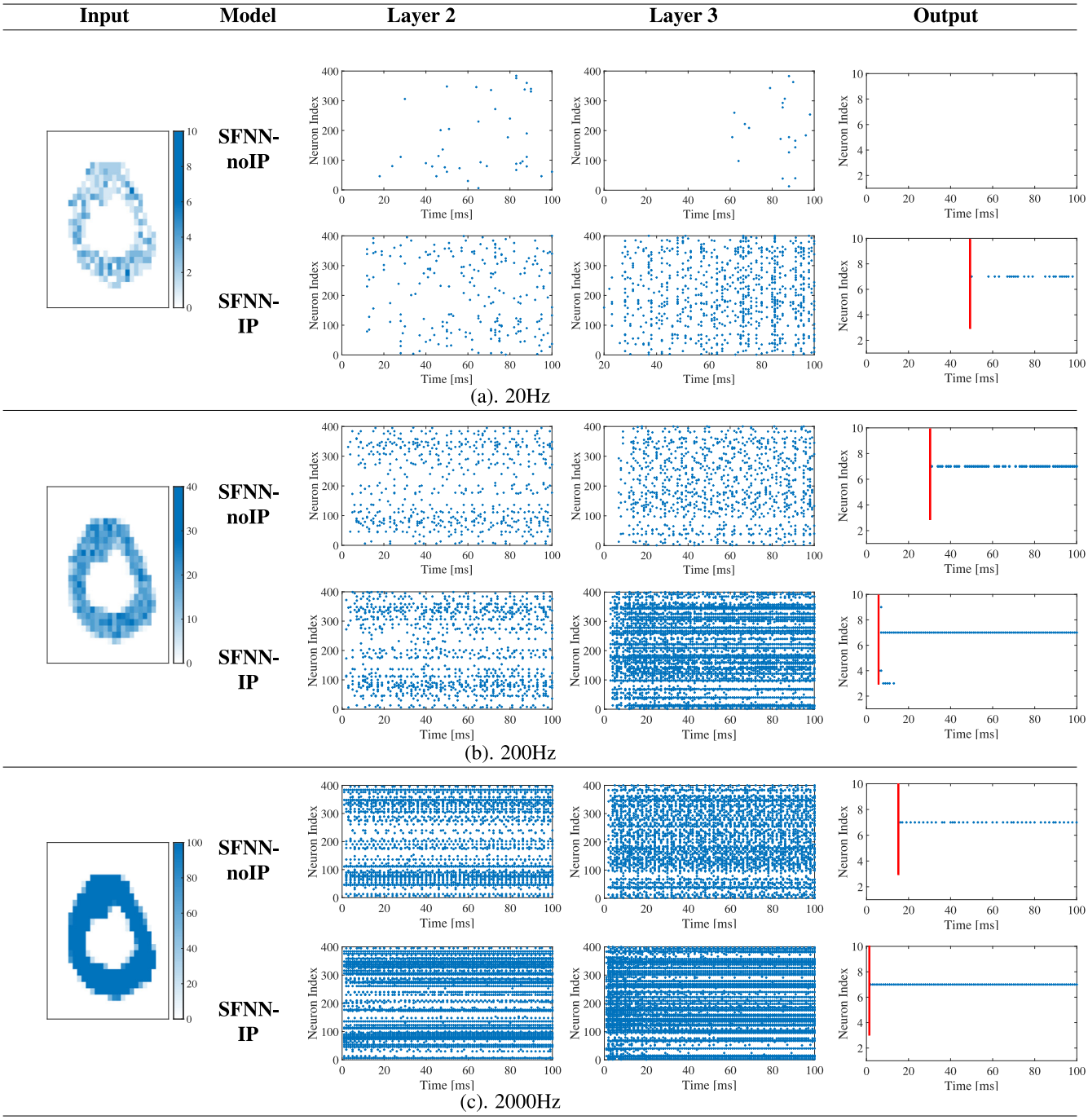

Fig. 5. Firing activities of the hidden layers and output layer of SFNN-IP and SFNN-noIP with different input firing rates (20 Hz, 200 Hz, 2000 Hz). The red vertical lines represent the start time of output response. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

based on prior statistical knowledge to regulate the input distribution [38–41]. But unfortunately, it is usually resource-costly or even impossible to achieve.

Under this situation, it is valuable to obtain an effective model or algorithm which has the ability to maintain a stable computational performance. Here we investigate the influence of IP learning on the computational performance of SFNN for image recognition with different exposure degrees. The normal "pure" MNIST dataset is used for the training process, while the test dataset is ill-exposed with different exposure degrees. Results are as shown in Fig. 8. Here the exposure degree or the image brightness is adjusted by changing the values of image pixels from −0.5 and 0.5. That is, if the image is underexposed, the values of all pixels after normalization are adjusted to be 0 if the value is less than 0.4915. For over exposure, the values of all pixels are adjusted to be 1 if it is larger than 0.7717. Examples of images with different exposure degree are also shown in Fig. 8. We can see that in all of the test cases, both of the learning speed and accuracy of SFNN can be significantly improved through the IP mechanism. This result further confirms that the IP learning can effectively avoid both over-activation and non-activation, thus broadening the input condition.

Further, it should be noted that unlike the effect of input firing rate shown in Fig. 4 which seems to be a positive correlation between the input firing rate and classification performance in

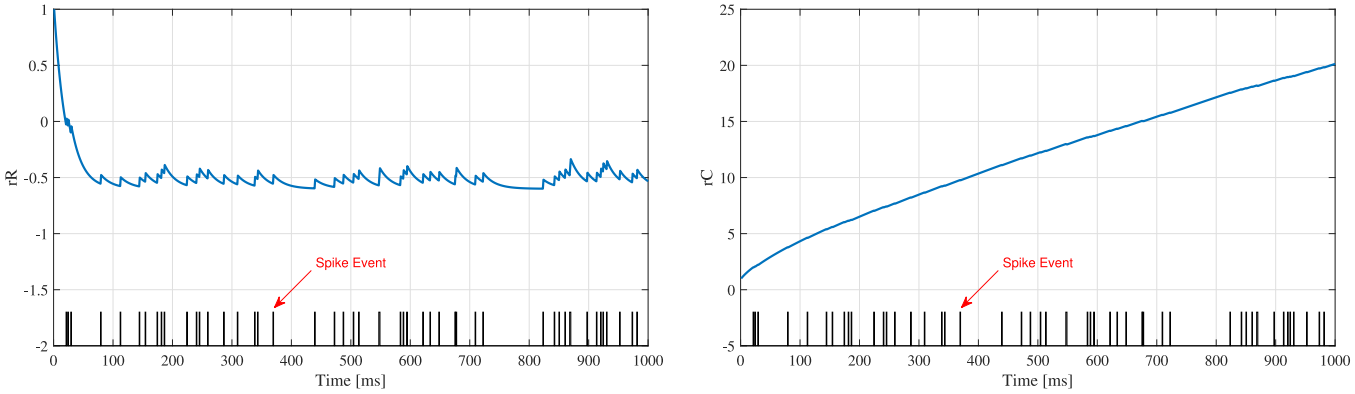

Fig. 6. Dynamical variations of *rR* (left) and *rC* (right) of a single neuron driven by a random spiking train and a fixed input current *I*.

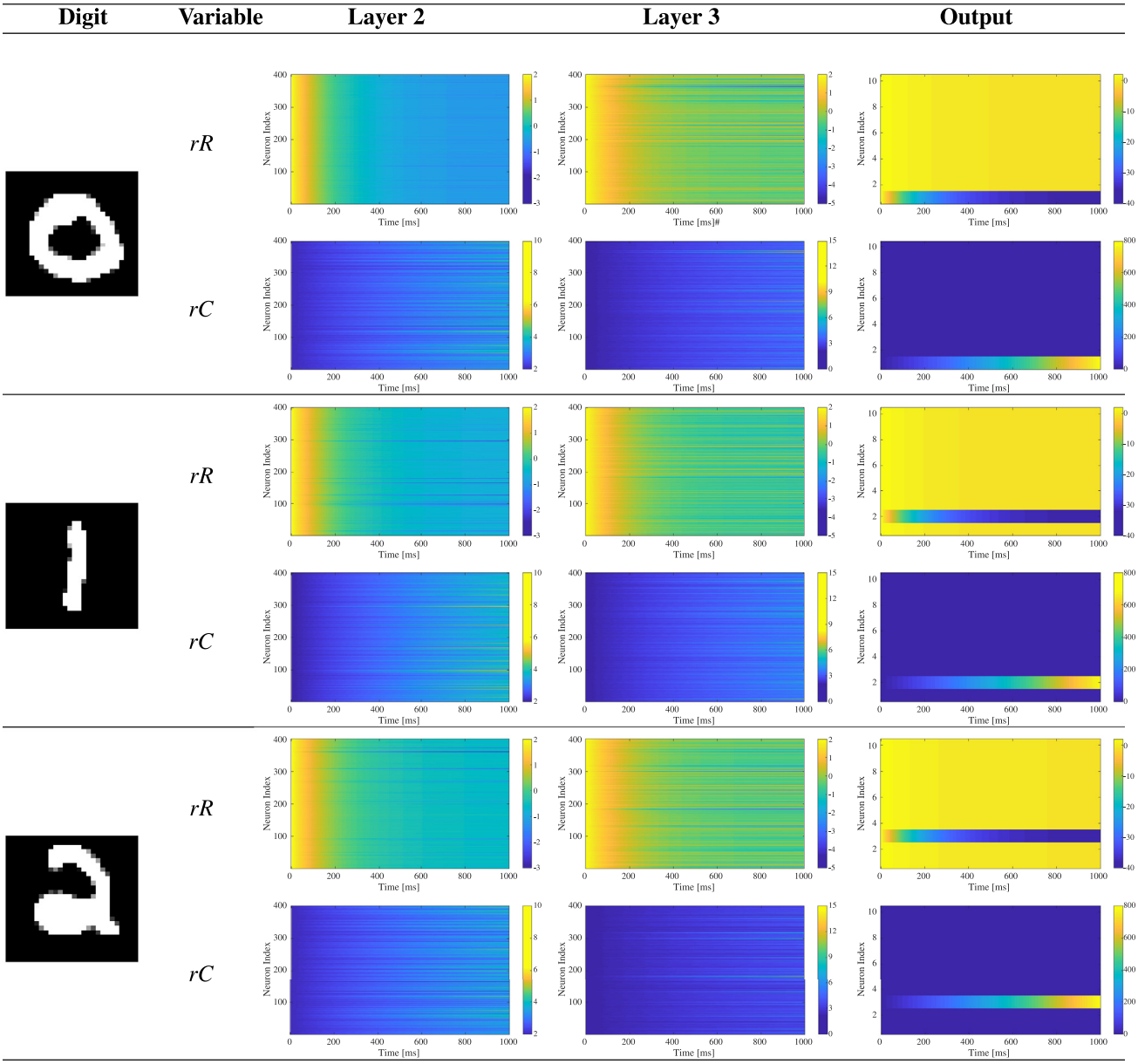

Fig. 7. The variations of *rR* and *rC* of the two hidden layers and the output layer during the IP learning in SFNN-IP for different input images (digits 0, 1, and 2).

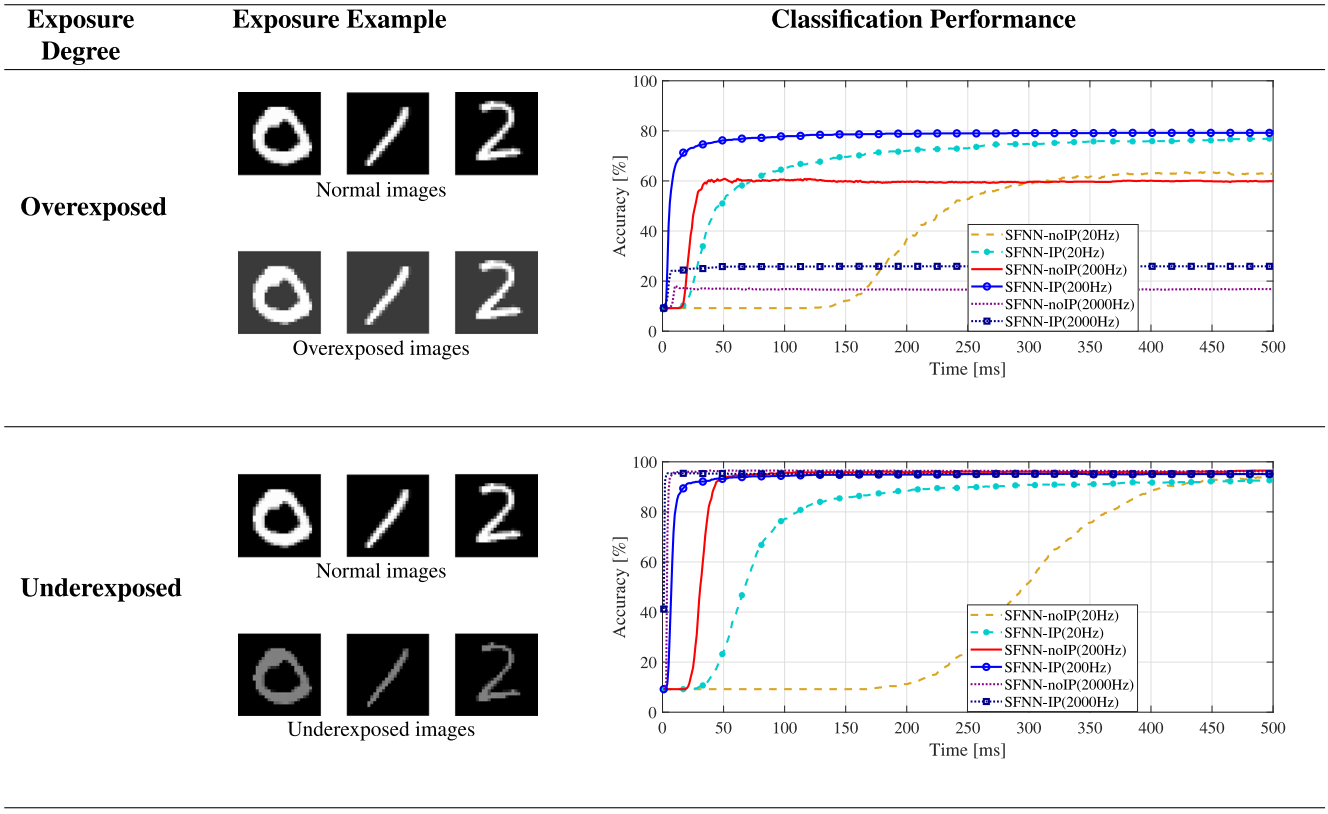

Fig. 8. Influence of image exposure degrees on the classification performance of SFNN-noIP and SFNN-IP with different input rates.

terms of both learning speed and accuracy, the computational performance on overexposed dataset at input firing rate of 2000 Hz is under-performed compared with the other cases. It indicates that too high input firing rate will cause over-activation and reduce classification accuracy. A suitable input firing rate seems to be necessary for the optimal performance.

4.3. Classification performance on noisy dataset

Noise commonly exists in sample data due to external disturbance or measurement error. Despite some literature demonstrated that appropriate intensity of noise is benefit for enhancing the generalization capability of the machine learning models, it is still an undeniable fact that too much noise will seriously weaken the models' performance. Thus, learning robustness has always been an important comprehensive indicator for the evaluation of computational capability.

Here, we use a well-trained SFNN model on normal "pure" MNIST training dataset and test its classification performance on noisy dataset. Five different types of noise are considered (as shown in Fig. 9), including gaussian noise, uniform noise, rayleigh noise, gamma noise and salt-and-pepper noise, which exist widely in engineering practice. Here image signal-to-noise ratio (SNR) is used for characterizing the intensity ratio between image and noise, which is calculated by

$$SNR(x,y) = 10log_{10}\left(\frac{\sum_{i=1}^{M}\sum_{j=1}^{N}x(i,j)^2}{\sum_{i=1}^{M}\sum_{j=1}^{N}(x(i,j)-y(i,j))^2}\right) \tag{12}$$

where $x(i, j)$, and $y(i, j)$ denote the original image and disturbed image respectively for the pixel (i, j) of image. M and N are the pixel numbers. In our experiments, classification performance of SFNN-noIP and SFNN-IP are investigated based on the ill-posed dataset with the SNR value of -3 db.

Gaussian noise. Gaussian noise is also known as normal noise which means the statistical property obeys the normal distribution with zero mean and variance θ^2. $\theta = 0.33465$ leads to the signal-to-noise ratio (SNR) value of -3 db.

Uniform noise. Uniform noise it also another common form of noise, it can be written as

$$z = \alpha + (\beta - \alpha)u(0,1) \tag{13}$$

where $u(0, 1)$ is uniformly distributed between 0 and 1. $\alpha = 0, \beta = 0.5793$ leads to the SNR value of -3 db.

Rayleigh noise. The mathematical formulation of rayleigh noise is

$$z = \alpha + \beta\sqrt{-ln[1-u(0,1)]} \tag{14}$$

and $\alpha = 0, \beta = 0.33417$ leads to the SNR value of -3 db.

Gamma noise. Gamma noise is superimposed by n exponentially distributed noises, it can be described by

$$z = z_1 + z_2 + \cdots + z_n \tag{15}$$

where $z_i = -\frac{1}{\alpha}ln[1-u(0,1)]$. And it should be noted that if $n = 1$, the gamma noise turns into exponential noise. $n = 4, \alpha = 18.78$ leads to the SNR value of -3 db.

Salt-and-Pepper noise. Salt-and-Pepper Noise is represented as sparsely occurring white and black pixels, which can be generated by sharp or sudden disturbances in the image signal. The noise probability of 6% of "salt" and "pepper" also leads to the SNR value of -3 db.

Classification performances on the above mentioned noisy datasets are presented in Fig. 10, where *Accuracy* and *Time* denote the final convergent classification accuracy and convergence time reaching to the final classification accuracy, respectively. From these figures we can see that the IP rule can also promote the robustness of spiking neural networks, which is independent of the noise type. This is significantly meaningful to real-time and noise

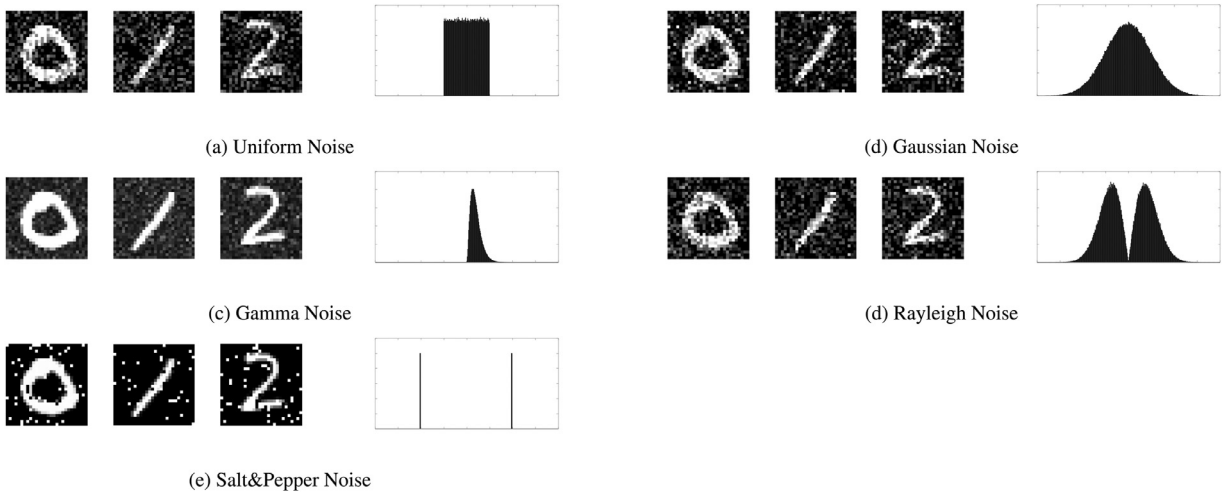

Fig. 9. Noisy images with different noise distribution.

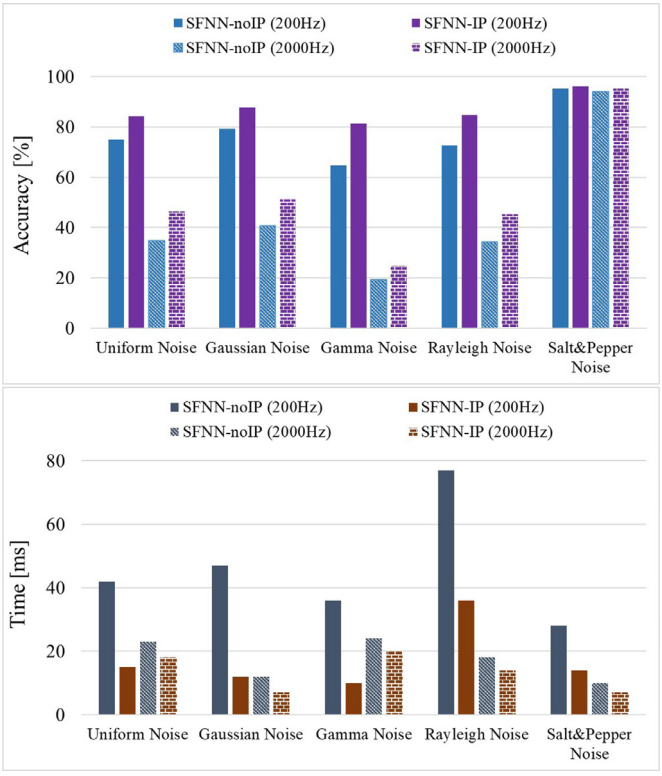

Fig. 10. The classification accuracy and convergence time of SFNN-noIP and SFNN-IP based on different noisy test datasets.

interfered applications. Note that for most cases, too high input firing rate i.e. 2000 Hz leads to low classification accuracy. Besides, the recognition performance of our model on MNIST dataset with different intensity of the salt&pepper noise is also compared with an event-driven feed-forward categorization model which is introduced in [42]. Comparison results shown in Table 3 demonstrate that our method can achieve higher classification accuracy than the model proposed in [42].

4.4. Parameters selection

The advantages of IP learning on speed acceleration and robustness of the computational performance of Spiking Feed-Forward Neural Networks (SFNN) have been investigated in detail, however, value selection of several key parameters can also significantly influence the computational performance. In the following, three important parameters, i.e. the relative integration resolution τ_{ip} and the scale factor β of the IP mechanism (see Eq. (4)), and the strength of spiking output η are investigated in detail.

Table 3
Classification accuracy comparison between our method and [42] on salt&pepper noise added MNIST test dataset.

Noise density	0	0.05	0.1	0.2	0.3	0.5
Method in [42](%)	91.3	89.6	87.4	81.2	73.8	47.1
Our Method(%)	97.2	96.1	95.0	89.5	77.2	55.8

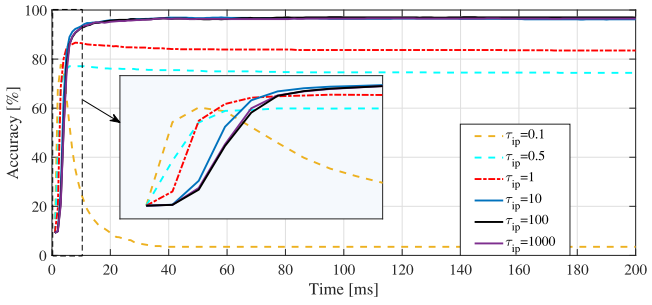

Fig. 11. Computational performance of SFNN-IPs with different τ_{ip}, where β and η are set to be 0.6 and 0.5, respectively.

4.4.1. The relative integration resolution τ_{ip} of IP learning

Fig. 11 shows the computational performance of SFNN-IP with different values of τ_{ip}. We can see that when τ is very small (i.e. 0.1), the classification accuracy drops quickly after reaching the peak value. As the value of τ increases, the classification accuracy can converge into a stable state. Besides, too large value of τ will dramatically slow down the updating process of IP learning. Fig. 12 shows that spiking frequencies of both hidden layers increases with the increase of τ_{ip}, then fall into a relative homeostatic state. Therefore, the parameter τ is set as 20 in this paper.

4.4.2. The scale factor β of IP learning

Fig. 13 shows that computational accuracy of SFNN-IP drops quickly when β is greater than 1, while the performance curves for β values in the range of (0, 1) are almost coincident. It should be noted that when beta is 0, the mathematical formulation (4) of

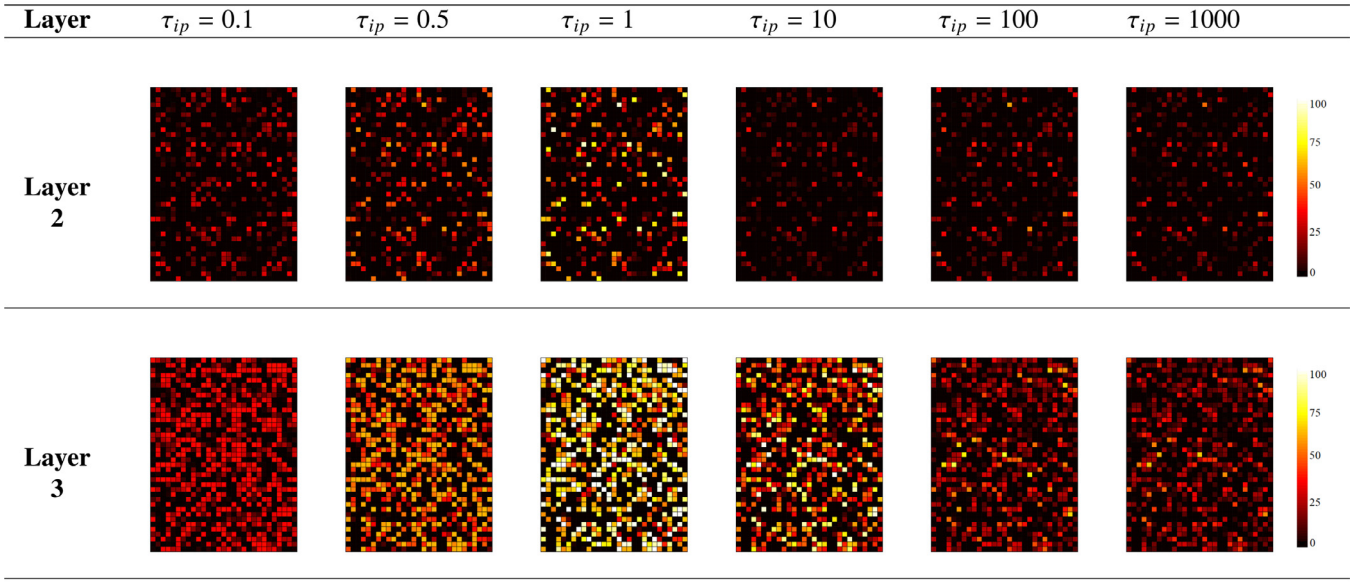

Fig. 12. Firing activities of the two hidden layers of SFNN-IP during the initial 100ms, with different values of τ_{ip}.

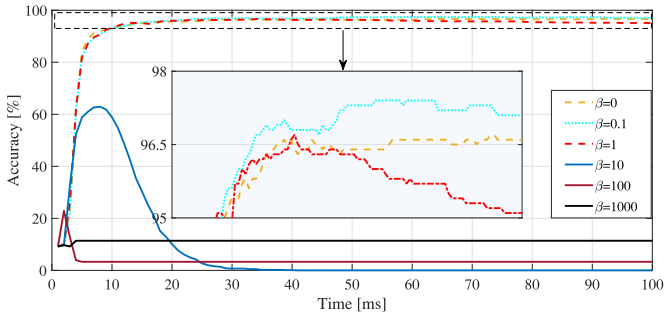

Fig. 13. Computational performance of SFNN-IPs with different β, where τ_{ip} and η are set to be 20 and 0.5, respectively.

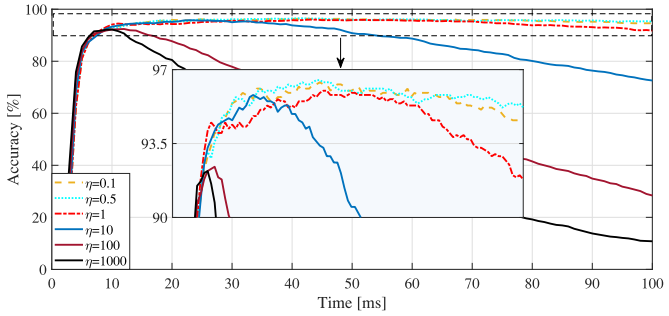

Fig. 14. Computational performance of SFNN-IPs with different η, where τ_{ip} and β are set to be 20 and 0.6, respectively.

IP can be simplified as

$$\tau_{ip}\frac{drC}{dt} = \frac{1}{rC} - yI$$

$$\tau_{ip}\frac{drR}{dt} = -rR + y \tag{16}$$

In this paper, the parameter β is set as 0.6.

4.4.3. The strength η of spiking output

In Fig. 14, we compare the computational performance of SFNN-IP for different η values. It can be seen that SFNN-IP performs well in terms of learning speed and classification accuracy when η is less than 1. In this paper the parameter η is set as 0.5.

5. Conclusion

Although analog deep learning networks can achieve high classification accuracy on computational tasks, they can not be directly applied to event-based computing systems, i.e. the neuromorphic circuits. Deep spiking neural networks, which have been shown to be significantly more energy-efficient than conventional ones [4], are trying to solve the problem. However, the conversion from analog neural networks to event-driven spiking neurons cause the cost of performance losses. Errors are mostly caused by the over-activation of neural responses to input data. Thus it is essential to find the right balance of spiking thresholds, input weight and input firing rates. In this paper, it is achieved by using the Intrinsic plasticity (IP) which is a self-adjusting mechanism of single unit with the ability of changing its intrinsic excitability to adapt its synaptic input dynamically. We focused on the effects of IP rule for spiking neural networks. By a series of experiments, we found that IP rule had the ability to guarantee the network classification accuracy at a high level, and meanwhile, the network convergence speed was dramatically improved. Besides, when the exposure degree and noise of input data are taken into account, both of the learning speed and accuracy of the model can be greatly improved by the application of IP learning. With the application of IP learning, the average firing rates of the whole network can be maintained in a reasonable scope even if the input is extremely weak or strong or interfered by noise. The robustness enhancement is quite important for achieving stable and high computational performance of neural networks. And the most attractive characteristic of this method is that the enhancement of both accuracy and computing speed can be achieved in a simple and effective way, without tuning the right balance of lots of parameters.

Declarations of interest

None.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China (no. 61473051), and Natural Science Foundation of Chongqing (no. cstc2016jcyjA0015).

References

[1] B. Han, A. Senqupta, K. Roy, On the energy benefits of spiking deep neural networks: A case study, in: International Joint Conference on Neural Networks, 2016, pp. 971–976.
[2] B. Han, A. Ankit, A. Ankit, K. Roy, Cross-layer design exploration for energy-quality tradeoffs in spiking and non-spiking deep artificial neural networks, IEEE Trans. Multi-Scale Comput. Syst. (2017). 1–1
[3] A.P. Johnson, J. Liu, A.G. Millard, et al., Homeostatic fault tolerance in spiking neural networks: a dynamic hardware perspective, IEEE Trans. Circuits Syst. I (2017) 1–13.
[4] M. Sharad, C. Augustine, G. Panagopoulos, K. Roy, Proposal for neuromorphic hardware using spin devices, Comput. Sci. (2012).
[5] C. Hausler, M.P. Nawrot, M. Schmuker, A spiking neuron classifier network with a deep architecture inspired by the pfactory system of the honeybee, in: Proceedings of International IEEE/EMBS Conference on Neural Engineering, IEEE, 2011, pp. 4361–4362.
[6] P. Panda, K. Roy, Unsupervised regenerative learning of hierarchical features in spiking deep networks for object recognition, in: Proceedings of International Joint Conference on Neural Networks, IEEE, 2016, pp. 885–900.
[7] M. Cernak, A. Lazaridis, A. Asaei, P.N. Garner, Composition of deep and spiking neural networks for very low bit rate speech coding, Sci. China Math. 22 (10) (1979) 1109–1113.
[8] A. Sengupta, M. Parsa, B. Han, K. Roy, Probabilistic deep spiking neural systems enabled by magnetic tunnel junction, IEEE Trans. Electron Dev. 22 (10) (2016) 2963–2970.
[9] M.C.W. van Rossum, G.G. Turrigiano, S.B. Nelson, Fast propagation of firing rates through layered networks of noisy neurons, J. Neurosci. 22 (5) (2002) 1956–1966.
[10] D.Z. Jin, Fast convergence of spike sequences to periodic patterns in recurrent networks, Phys. Rev. Lett. 89 (20) (2002).
[11] D.Z. Jin, H.S. Seung, Fast computation with spikes in a recurrent neural network, Phys. Rev. E - Stat. Phys. Plasmas Fluids Rel. Interdiscip. Top. 65 (5) (2002) 4.
[12] S.M. Bohte, J.N. Kok, H. La Poutré, Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing 48 (1–4) (2002) 17–37.
[13] J.H. Lee, T. Delbruck, M. Pfeiffer, Training deep spiking neural networks using backpropagation, Front. Neurosci. 10 (NOV) (2016) arXiv:1608.08782.
[14] B. Schrauwen, J. Van Campenhout, Backpropagation for Population-Temporal Coded Spiking Neural Networks, in: The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE, 2006, pp. 1797–1804.
[15] S. McKennoch, T. Voegtlin, L. Bushnell, Spike-Timing error backpropagation in theta neuron networks, Neural Comput. 21 (1) (2009) 9–45.
[16] S.M. Bohte, Error-Backpropagation in Networks of Fractionally Predictive Spiking Neurons, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 6791 LNCS, 2011, pp. 60–68.
[17] J. Yang, W. Yang, W. Wu, A remark on the error-backpropagation learning algorithm for spiking neural networks, Appl. Math. Lett. 25 (8) (2012) 1118–1120.
[18] Y. Wu, L. Deng, G. Li, J. Zhu, L. Shi, Spatio-temporal backpropagation for training high-performance spiking neural networks, Front. Neurosci. 12 (MAY) (2018) arXiv:1706.02609.
[19] S.R. Kheradpisheh, M. Ganjtabesh, S.J. Thorpe, T. Masquelier, STDP-based spiking deep convolutional neural networks for object recognition, Neural Netw. 99 (2018) 56–67 arXiv:1611.01421.
[20] M. Mozafari, S.R. Kheradpisheh, T. Masquelier, A. Nowzari-Dalini, M. Ganjtabesh, First-spike-based visual categorization using reward-modulated STDP, IEEE Trans. Neural Netw. Learn.Syst. 29 (12) (2018) 6178–6190 arXiv:1705.09132.
[21] P. O'Connor, D. Neil, S.C. Liu, T. Delbruck, M. Pfeiffer, Real-time classification and sensor fusion with a spiking deep belief network, Frontiers in Neuroscience 7 (7) (2013) 1–13.
[22] Y. Cao, Y. Chen, D. Khosla, Spiking deep convolutional neural networks for energy-Efficient object recognition, Int. J. Comput. Vis. 113 (1) (2015) 54–66 arXiv:1502.05777.
[23] P.U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, M. Pfeiffer, Fast-classifying, high--accuracy spiking deep networks through weight and threshold balancing, in: Proceedings of International Joint Conference on Neural Networks, IEEE, 2015, pp. 1109–1113.
[24] B. Rueckauer, I.A. Lungu, Y. Hu, M. Pfeiffer, S.C. Liu, Conversion of continuous–valued deep networks to efficient event-driven networks for image classification, Front. Neurosci. 11 (DEC) (2017) 1–12.
[25] B. Rueckauer, S.C. Liu, Conversion of analog to spiking neural networks using sparse temporal coding, Proc. IEEE Int. Symp.Circuits Syst. 2018-May (2018).
[26] S. Kim, S. Park, B. Na, S. Yoon, Spiking-YOLO: spiking neural network for real-time object detection, arXiv:1903.06530 (2019).
[27] N.S. Desai, L.C. Rutherford, G.G. Turrigiano, Plasticity in the intrinsic excitability of cortical pyramidal neurons, Nature Neurosci. 2 (1999) 515–520.
[28] W. Zhang, D.J. Linden, The other side of the engram: experience driven changes in neuronal intrinsic excitabilit, Nature Rev. Neurosci. 4 (2003) 885–900.
[29] B. Schrauwena, M. Wardermanna, D. Verstraetena, J.J. Steilb, D. Stroobandt, Improving reservoirs using intrinsic plasticity, Neurocomputing 13 (6) (2008) 1159–1171.
[30] P. Joshi, J. Triesch, Rules for information maximization in spiking neurons using intrinsic plasticity, in: Proceedings of International Conference on Machine Learning, 2009, pp. 1456–1461.
[31] P. Koprinkova-Histova, On effects of IP improvement of ESN reservoirs for reflecting of data strcture, in: Proceedings of International Conference on Machine Learning, 2015, pp. 26–30.
[32] M. Wardermann, J.J. Steil, D. Stroobandt, Improving reservoirs using intrinsic plasticity, Neurocomputing 4 (2008) 885–900.
[33] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proceedings of International Conference on Machine Learning, 2010, pp. 26–30.
[34] Y. Cao, Y. Chen, D. Khosla, Spiking deep convoloutional neural networks for energy-efficient object recognition, Int. J. Comput. Vis. 13 (6) (2014) 1–13.
[35] L.F. Abbott, Lapicque's introduction of the integrate-and-fire model neuron, Brain Res. Bull 13 (6) (1999) 303–304.
[36] C. Li, Y. Li, A review on synergistic learning, IEEE Access 4 (2016) 119–134.
[37] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, Int. Conf. Mach. Learn. 37 (2015) 448–456.
[38] G. Chakraborty, B. Chakraborty, A novel normalization technique for unsupervised learning in ANN, IEEE Trans. Neural Netw. 11 (2000) 253–257.
[39] H. Zhang, H. Lin, Y. Li, Impacts of feature normalization on optical and SAR data fusion for land use/land cover classification, IEEE Geosci. Remote Sens. Lett. 12 (2015) 1061–1065.
[40] Y.H. Kim, H. Kim, S.W. Kim, H.Y. Kim, S.J. Ko, Illumination normalisation using convolutional neural network with application to face recognition, Electron. Lett. 53 (2017) 399–401.
[41] P. Sane, R. Agrawal, Pixel normalization from numeric data as input to neural networks, in: International Conference on Wireless Communications, Signal Processing and Networking, 2017, pp. 2221–2225.
[42] B. Zhao, R. Ding, S. Chen, B. Linares-Barranco, H. Tang, Feedforward categorization on AER motion events using cortex-like features in a spiking neural network, IEEE Trans. Neural Netw. Learn. Syst. 26 (9) (2015) 1963–1978.

Anguo Zhang was born in Hefei city, Anhui province, China in 1990. He received his Bachelor degree and Master degree in control engineering from Chongqing University, Chongqing in 2012 and 2016, respectively. Since 2018, he has been a researcher and senior engineer in the Research Institute of Ruijie, Ruijie Networks Co., Ltd. His research interest includes machine learning, artificial neural networks, control theory and applications, and intelligent communication networks.

Hongjun Zhou received his Bachelor degree and Master degree in Mathematics from Jilin University, China, in 2003 and 2006, respectively. He got the Ph.D. degree from Hong Kong Polytechnic University, Hong Kong, in 2012. He is currently a full-time lecturer in the School of Economics and Business Administration, Chongqing University, Chongqing, China. His research interests include computational mathematics, financial applications of artificial neural networks.

Xiumin Li received the Bachelor degree in Automation from Taiyuan University of Technology, China, in 2005, the Master degree in Automation from Tianjin University, China, in 2007, and the Ph.D. degree from Hong Kong Polytechnic University, Hong Kong, in 2011. She is currently a full-time associate professor in the College of Automation, Chongqing University, Chongqing, China. Her research interests include computational neuroscience, neurodynamic analysis, neural network modeling, and their implications to intelligent computing.

Wei Zhu (JUI LANG CHU) received the B.S. degree in electromechanics from Harbin Institute of Technology in 2006, Harbin, China and the Ph.D. degree in Instrumentation Science from University of Science and Technology of China, Hefei, China in 2011. He continued his postdoctoral research on neurological disease assistant diagnosis in National University of Singapore, Singapore. He is currently a Senior Researcher in Ruijie Networks, Fuzhou, China and the team leader of Artificial Intelligence and Computer Vision. His current research interests include machine learning, computer vision and neurology.