SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2017, Vol. 60 052201:1–052201:11
doi: 10.1007/s11432-016-0439-4

# Improving multi-layer spiking neural networks by incorporating brain-inspired rules

Yi ZENG[1,2*†], Tielin ZHANG[1*†] & Bo XU[1,2]

[1]*Institute of Automation, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai* 200031*, China*

Received June 28, 2016; accepted August 28, 2016; published online February 24, 2017

**Abstract** This paper introduces seven brain-inspired rules that are deeply rooted in the understanding of the brain to improve multi-layer spiking neural networks (SNNs). The dynamics of neurons, synapses, and plasticity models are considered to be major characteristics of information processing in brain neural networks. Hence, incorporating these models and rules to traditional SNNs is expected to improve their efficiency. The proposed SNN model can mainly be divided into three parts: the spike generation layer, the hidden layers, and the output layer. In the spike generation layer, non-temporary signals such as static images are converted into spikes by both local and global feature-converting methods. In the hidden layers, the rules of dynamic neurons, synapses, the proportion of different kinds of neurons, and various spike timing dependent plasticity (STDP) models are incorporated. In the output layer, the function of classification for excitatory neurons and winner take all (WTA) for inhibitory neurons are realized. MNIST dataset is used to validate the classification accuracy of the proposed neural network model. Experimental results show that higher accuracy will be achieved when more brain-inspired rules (with careful selection) are integrated into the learning procedure.

**Keywords** brain-inspired rules, spiking neural network, plasticity, classification task

**Citation** Zeng Y, Zhang T L, Xu B. Improving multi-layer spiking neural networks by incorporating brain-inspired rules. Sci China Inf Sci, 2017, 60(5): 052201, doi: 10.1007/s11432-016-0439-4

## 1 Introduction

Recent advancements of brain-inspired neural networks, such as deep neural networks (DNNs), have been successful in many domains, including visual information processing, speech recognition, etc. [1,2]. These advancements encourage efforts that borrow more brain-inspired mechanisms and rules to improve and extend the ability of neural networks.

Spiking neural network (SNN) is considered as the third generation of neural network models [3], which is biologically more consistent with the brain neural network, and has shown its uniqueness in temporal plasticity, etc. compared to other state-of-the-art models of artificial neural networks (such as DNNs). It has already been widely used and successful in various cognitive tasks, such as image classification, working memory, reasoning, robot navigation and motion learning, etc. [4–8].

* Corresponding author (email: yi.zeng@ia.ac.cn, tielin.zhang@ia.ac.cn)
† Equal contribution.

 

Several computational models have been proposed to refine SNNs, which attempt to combine biological plausible rules to improve the performance, such as higher accuracy and shorter training time. Some achievements have been made on the models of spiking neurons, learning strategies for synaptic weights, rank order coding and STDP learning rules for online or off-line spatial-temporal pattern recognition.

From the performance of accuracy perspective, Wade et al. [9] proposed a synaptic weight association and training algorithm which combined the Bienenstock Cooper Munro (BCM) learning rule with STDP. Both excitatory and inhibitory facilitating synapses which showed frequency routing capability were used to route the learning of hidden layer neurons. The work shows that the combination of different kinds of neurons can improve the network performance for classification. Beyeler et al. [10] used the Izhikevichs neuron model [11] and conductance based synaptic-dynamic rules, instead of single LIF model and traditional STDP rules, to form a hierarchical large scale SNN model to classify the hand written digits.

From the performance of convergence time perspective, Iakymchuk et al. [12] proposed an efficient model which integrated LIF, STDP, spike frequency coding, and receptive field coding into an SNN architecture. Their major concern is on efficiency and think that different neuron models with high computational complexity and memory requirement would limit their realization in real-time learning applications. The new model is with up to 20 times computing speed-up compared to other traditional SNNs. Spiking neural P systems (SN P systems) integrate biological rules and are successful on time-coding related tasks [13, 14]. Compared to SNN, although most of the variables in the system are well predefined and not based on self-organization mechanisms or dynamically tuned during task processing, the systems still give us many inspirations for better learning networks by incorporating more biological rules.

In this paper, we will introduce seven brain-inspired rules and add them into the SNN training procedure. These rules include the dynamics of neurons and the synapses, various kinds of STDPs, specific functions of the excitatory and inhibitory neurons, and the background noises. Experimental results indicate that higher accuracy can be achieved when more brain-inspired rules (that are carefully selected) introduced in this paper are incorporated into the multi-layer SNN model.

In Section 2, a multi-layer SNN architecture will be introduced which contains both excitatory and inhibitory neurons. Section 3 will introduce the models of neurons and synapses. Section 4 will make a detailed description on the seven learning rules for SNN training. In Section 5, a golden standard experiment will be conducted to test the accuracy and efficiency of our proposed architecture. Section 6 will conclude the paper by highlighting major contributions.

## 2 A multi-layer SNN architecture

Here we define a multi-layer SNN, which is shown in Figure 1. It is a feed forward network and all the variables are dynamic. The numbers of neurons in each layer and the synaptic weights between layers will increase or decrease based on the seven rules described in Section 4. The abilities of the proposed multi-layer SNN should contain static multi-dimensional classification (e.g. image classification), temporal prediction (i.e. regression analysis), or the combination of these two. In this paper, we focus on the classification task. The architecture contains three parts: the spikes generation layer, the hidden layers, and output layer. In the hidden layers and output layer, the proper percentage of inhibitory neurons are assigned for realizing sparsity and winner take all (WTA) mechanism respectively.

### 2.1 The spike generation layer

In spike generation layer, when the inputs are 2D static image data (with width and height coordinates) without temporary information, two methods are proposed (as shown in Figures 2 and 3) to convert these data into spike signals (with the number of neurons and spike timing coordinates).

The global feature converting method contains two steps: (1) Convert the 2-D gray images into 1-D spike information (with only spatial information left) based on the comparison between gray values and

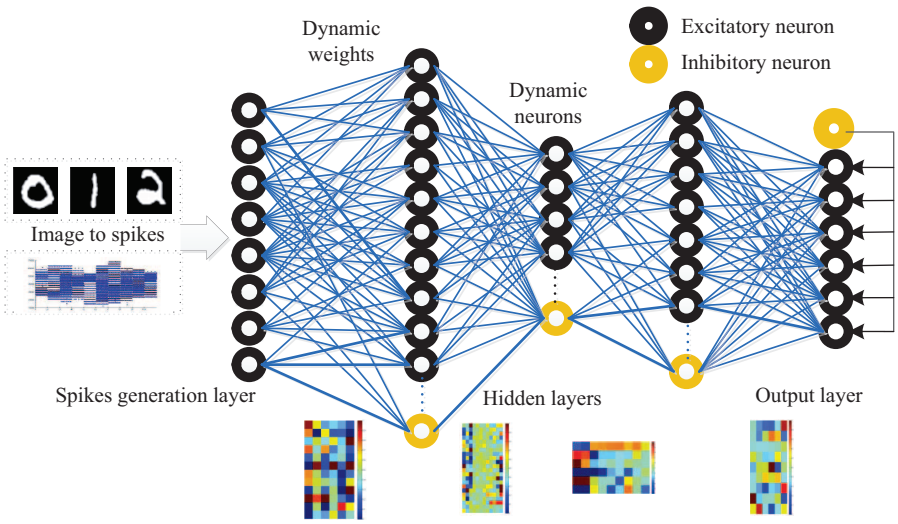


**Figure 1** (Color online) The architecture of multi-layer SNN.

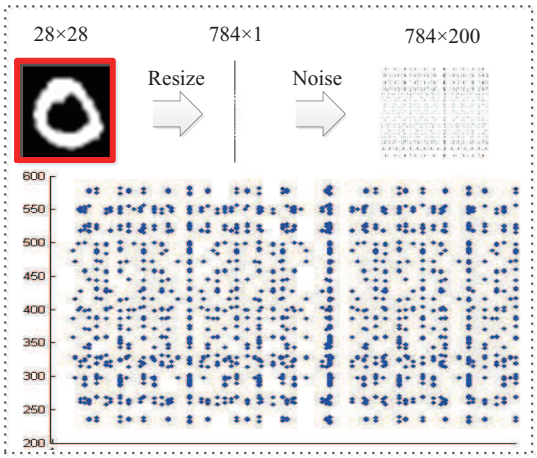


**Figure 2** (Color online) The global feature converting method.

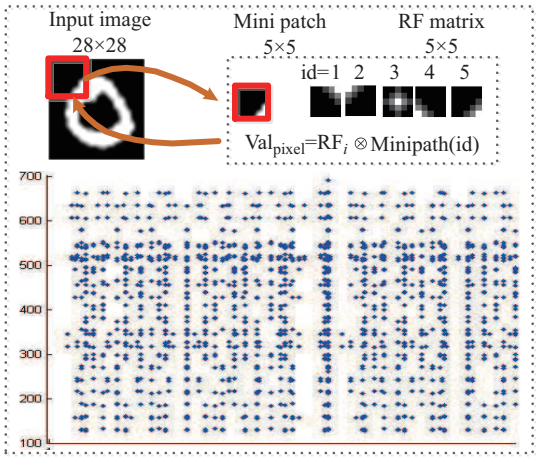


**Figure 3** (Color online) The local feature converting method.

the static or dynamic threshold. (2) The 1-D spike information are repeated (here is 200 times) and mixed with Poisson noises to form new 2-D signals in which temporary information is added into the spatial information.

The local feature converting method is similar with the convolutional procedure where the convolutional computation is made by receptive field kernels in an image. The kernels are sensitive to different kinds of features (e.g. point, corner or edge features) [15]. The time of each step for convolution procedure is consistent with the time of different spiking sequences.

The global feature converting method is more robust to the noise of the images but needs more input neurons. While the local feature converting method has clearer descriptions on the inner characteristics of images, but are more sensitive to the size of kernels.

Since the global features and local features usually describe different scales of characteristics of the image, in this paper, we make a combination of these two methods by making the OR operation to local and global features to represent the original image. The effectiveness of this combination will be introduced in Subsection 5.1.

### 2.2 The hidden layers

The hidden layers are the keys for dynamic learning on the number of task neurons and the weights of synapses based on the learning rules. The proportion of inhibitory neurons will be changed based on the learning rules. Three hidden layers are constructed in the auto-encoder style (in which the second layer has fewest neurons and connections compared to the other two layers which is inspired by the early stage of visual pathway [16]).

One of the problems in learning synaptic weights is that when the weights in lower layers are changed, some characteristics that the network has learnt will be erased by the new data. In order to solve this problem, we set $\mathrm{mw}_{L_{i,j}}$ as the meta weight among layer $i$ and layer $j$, which is essential for synaptic weight update, and will be discussed in more detail in Subsection 4.2.

### 2.3 The output layer

The output layer is for the final classification using excitatory neurons. The number of excitatory neurons in this layer is the same with the number of output classes. One inhibitory neuron is used for inhibiting other candidate classes in the WTA manner.

## 3 Neuron and synapse models in the multi-layer SNN architecture

### 3.1 The neuron model

There are various kinds of neurons in the human brain. Different neurons have very different morphologies and functions, while they all play various important roles for realizing complex cognitive functions. In this paper, we select two kinds of typical neurons, pyramidal cell and basket cell as the models of excitatory neurons and inhibitory neurons in the multi-layer SNN architecture.

Different kinds of neuron models have been proposed, such as the Hodgkin-Huxley (H-H) model [17], the Leaky Integrate and Fire (LIF) model [18], and the Izhikevich model [11]. For the purpose of efficient computation, we use LIF model in this paper. A standard membrane potential equation of LIF proposed in [18] is shown in (1) and (2):

$$C_m \frac{\mathrm{d}V}{\mathrm{d}t} = -g^L (V - V_L) + I_{\mathrm{syn}}, \tag{1}$$

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = -(V - V_L) + \frac{I_{\mathrm{syn}}}{g^L}, \tag{2}$$

where $C_m$ is the membrane capacitance of the neuron, $I_{\mathrm{syn}}$ is the input current from the pre-synapses, $V$ is the membrane potential, $g^L$ is the conductance of membrane, $V_L$ is the steady-state leaky potential. $\tau_m = C/g^L$ stands for the period of voltage decay, and for different kinds of neurons, $\tau_m$ may be very different [18]. The reason why we do not select H-H model (only leak conductance $g^L$ is kept, other ion channels such as $g^{\mathrm{Na}^+}$, $g^{\mathrm{K}^+}$ and $g^{\mathrm{Ca}^{2+}}$ are ignored), is that the LIF model is with the minimal computational complexity and has shown the majorities of the neuron reactions to the stimulus.

In this paper, the total received excitatory conductance is $g^E$ and received inhibitory conductance is $g^I$. And the reversal potential for corresponding channels are $V^E$ and $V^I$. The model for each neuron in SNN is described by (3)–(5). When $V > V_{\mathrm{th}}$, a spike is generated and then the potential value is reset. Following the long-term potentiation (LTP) mechanism, $V_{\mathrm{reset}}$ will be a little higher than $V_L$ for both excitatory and inhibitory neurons. $g^E$ and $g^I$ are dynamic updated based on the synapse models. Here $V_L = -70$ mV, $V_{\mathrm{th}} = -50$ mV, $V_{\mathrm{reset}} = -55$ mV, $\tau_m = 20$ ms, $V_L = -70$ mV, $V_E = 0$ mV, $V_I = -70$ mV and $g^L = 20$ nS.

$$I_i^{\mathrm{syn}} = -g^E(V - V_E) - g^I(V - V_I), \tag{3}$$

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = -(V - V_L) - \frac{g^E}{g^L}(V - V_E) - \frac{g^I}{g^L}(V - V_I), \tag{4}$$

$$V \to V_{\text{reset}} \quad \text{if} \quad V > V_{\text{th}}. \tag{5}$$

### 3.2 The synapse model

The dynamics of the synapses are the keys for neural network training [19]. There are many models for synaptic connections, and one of the most widely used models for spiking neurons is the conductance-based synapse model [20]. It assumes that the pre-synaptic spikes affect post-synaptic membrane potential by opening particular ion channels. Excitatory synapse will release glutamate, which binds AMPA receptors (For the purpose of simplification, we only consider AMPA in this paper). Inhibitory synapse will release neurotransmitter GABA which binds GABA receptor. Ion channels of these receptors will be opened after binding of neurotransmitter, allowing ions flux through the channels, thus increasing the conductance of the membrane. We can eventually have one value for representing the total excitatory and inhibitory conductance onto a neuron. AMPA receptor and GABA receptor channels allow for different types of ions to pass through, so the reversal potential to $V_E$ and $V_I$ are different. Without any pre-synaptic spikes, conductance will decrease to zero with time constant $\tau_E$ and $\tau_I$ for excitatory and inhibitory conductance respectively, as shown in (6) and (7):

$$\frac{\mathrm{d}g_i^E}{\mathrm{d}t} = -\frac{g_i^E}{\tau_E}, \tag{6}$$

$$\frac{\mathrm{d}g_i^I}{\mathrm{d}t} = -\frac{g_i^I}{\tau_I}. \tag{7}$$

Whenever a pre-synaptic neuron $j$ spikes, it increases the conductance $g_{i,j}$ by an amount of $w_{i,j}$. $t_k^j$ is the spiking time to neuron $j$. The membrane potential of neuron $j$ reaches $V_{\text{peak}}$ if neuron $j$ fires, or become resting membrane potential when neuron $j$ did not fire for a long time. Finally we can get (8) and (9) in which $\tau_{(\text{syn}_E)} = 2$ ms, $\tau_{(\text{syn}_I)} = 5$ ms. The connectivity weight of the synapses $w_{i,j}$ will follow the STDP mechanism introduced in Section 4.

$$\Delta g_i^E(t) = -\frac{g_i^E(t)}{\tau_E}\Delta t + \sum_{j \in C_E} w_{i,j} \sum_k \delta_{t,t_k^j}, \tag{8}$$

$$\Delta g_i^I(t) = -\frac{g_i^I(t)}{\tau_I}\Delta t + \sum_{j \in C_I} w_{i,j} \sum_k \delta_{t,t_k^j}. \tag{9}$$

## 4 Brain-inspired learning rules in the multi-layer SNN

### 4.1 Learning rules

In order to improve the performance of traditional SNN models, here we introduce and incorporate seven learning rules to guide the training procedure of SNN, as shown in Table 1.

(1) $R1$ and $R2$: Allocating new neurons or deleting extra neurons based on task complexity. In many neural network models, the number of neurons for the network is static and cannot be changed. While in the human brain, different number of neurons can be allocated for different information processing procedures and tasks [21]. Based on this observation, we propose to dynamically allocate or delete neurons based on the task requirement for achieving better computation performance. In this paper, the number of neurons will increase or decrease automatically based on the complexity of the tasks (or the number of learning classes) and the neurons will be removed while no active synapses are with them.

(2) $R3$ and $R4$: Synapse formation and elimination. Different from some of the existing artificial neural network models in which neurons are fully connected to neighborhood layers, synapses are very dynamic in the biological brain, with synaptic formation and elimination based on biological learning rules such as STDP. In this paper, we introduce this mechanism to our multi-layer SNN model.

**Table 1** Seven learning rules used in the training procedure

| | |
|---|---|
| $R1$ | Allocating new Neurons. |
| $R2$ | Deleting extra Neurons. |
| $R3$ | Synapse formation. |
| $R4$ | Synapse elimination. |
| $R5$ | Background noise.<br>a: Uniform noise.<br>b: Possion noise. |
| $R6$ | STDP rules.<br>a: bi-phasic STDP.<br>b: tri-phasic STDP.<br>c: voltage STDP. |
| $R7$ | Introducing inhibitory neurons in SNN.<br>a: the anti-excitatory-type neurons.<br>b: the lock-excitatory-type neurons.<br>c: the percentage of the inhibitory neurons. |

For synapse formation, let $n_i$ denotes an arbitrary neuron in the lower layer and $n_j$ denotes another neuron in a higher neighbor layer, when the spikes appearance time for $n_j$ after $n_i$ spikes reach a threshold, at least 5 consecutive times (without post-synaptic neuron fires before pre-synaptic neuron during these consecutive times), then a new synapse will be formatted from $n_i$ to $n_j$.

For synapse elimination, synaptic weights may decrease based on the STDP rule, and if the weight of synapse is weak enough (with strength of $10^{-4}$), the specific synapse will be signed as eliminated.

The weights of synapses will be dynamically changed based on two considerations: the synapse model introduced in Subsection 3.2 and the rules of $R6$ and $R7$. The initial weights of synapses are random and the number of synapses is 50% full connections. Each inhibitory neuron in SNN plays a role on the detection of the active level of excitatory neurons, and it will be combined with $R5$b to guide the network for a balanced activity of each excitatory neuron. For example, some neurons are more active for the class named $\text{class}_A$, and when there is a new class named $\text{class}_B$, 50% of the neuron activities for $\text{class}_A$ will be inhibited so that new neurons which are inactive for $\text{class}_A$ could start to learn from $\text{class}_B$.

In order to simulate the procedure of information processing of biological neurons, we also set different refractory periods for excitatory and inhibitory neurons respectively in the multi-layer SNN model.

(3) $R5$a and $R5$b: Network background noise. For SNNs, the background noise will help network training. Two kinds of noises ($R5$a and $R5$b) are tested separately in the whole process of training procedure. The first one is uniform distribution random noise. Another one is Possion distribution noise. The Possion distribution noise shows more biological characteristics [22] and indeed shows better performance in the experimental task. The Possion distribution noise added into the network makes the neurons more probable to fire at the beginning of the training which will contribute to a higher convergence speed. In this paper, the background Possion noise is shown in (10). The $\text{mw}_{L_{i,j}}$ is the meta weight among different neighborhood layers, which is defined in detail in Subsection 4.2. $C_{\text{sp}}(q)$ is recorded by inhibitory neurons and works as the spike counter for the frequency of spikes of the $q$th excitatory neuron in the $i$th layer:

$$P(t=k)=\frac{\lambda^k}{k!}\mathrm{e}^{-\lambda}\text{mw}_{L_{i,j}}\tanh^{-1}\left(C_{\text{sp}}(q)\right),\quad k\in T. \tag{10}$$

(4) $R6$: STDP learning rules. Many implementations of the STDP rules have been proposed based on the biological experiments [23]. In this paper, the Tri-phasic STDP, Bi-Phasic STDP and voltage-dependent plasticity model are used to change the weight of synapses in the training procedure of the SNNs. Detailed descriptions about STDP models will be discussed in Subsection 4.2.

(5) $R7$: The proportion of excitatory and inhibitory neurons. Many experimental results have shown the great values of the excitatory neurons for feed forward network training [24], while there are even more

types of inhibitory neurons in the brain. These inhibitory neurons play very important roles in higher cognitive functions such as attention and control. Besides excitatory neurons, two types of inhibitory neurons will be added into the proposed network: one is anti-excitatory neurons (anti-E [25]) which will decrease the membrane potential of the post-synaptic excitatory neurons when the inhibitory neurons fire, and the other is lock-excitatory neurons (lock-E [26]) which will stop the target excitatory neurons from updating the synapse weights for a refractory time (so more synapses of unlocked excitatory neurons will be updated based on $R3$) instead of just weaken them. The designed two kinds of specific inhibitory neurons will be built to record the spike frequency of each excitatory neurons, and the plasticity level of inhibitory neurons will be one third of the excitatory neurons in SNN for the quick convergence. The analysis of proper proportions of inhibitory neurons in SNN will be introduced in Section 5.

### 4.2 STDP models and synaptic weight update

STDP is a kind of temporal learning rule in which the synapse weight will be updated based on timing of spikes generated by pre-synaptic and post synaptic neurons. Many STDP models have been proposed for network learning, such as Tri-phasic STDP, Bi-phasic STDP and voltage-dependent plasticity model. The descriptions of them are shown in (11) [27], (12) [28] and (13) [29], respectively.

(1) Tri-phasic STDP is modeled as [27]

$$\Delta w_i = A_+ \exp\left(\frac{-(\Delta t_i - t_m)^2}{\tau_+}\right) - A_- \exp\left(\frac{-(\Delta t_i - t_m)^2}{\tau_-}\right), \tag{11}$$

where $A_+$ and $A_-$ are the learning rates. $\Delta t_i$ is the delay from pre-synaptic spike to post-synaptic spike. $\tau_+$and $\tau_-$ are delays which control the rates of exponential decrease or potentiation. $t_m$ is the time corresponding to the maximum value of $\Delta w_i$. Relevant parameters are set as the following: $\tau_+ = 100$, $\tau_- = 1000$ and $t_{\max} = 15$ [27].

(2) Bi-phasic STDP is modeled as [28]

$$\Delta w_i = \begin{cases} A_+ \exp\left(\dfrac{\Delta t_i}{\tau_+}\right), & \text{if } \Delta t_i < 0; \\ -A_- \exp\left(\dfrac{-\Delta t_i}{\tau_-}\right), & \text{if } \Delta t_i > 0, \end{cases} \tag{12}$$

where $A_+$ and $A_-$ are learning rates, and $\Delta t_i$ is the delay time from pre-synaptic spike to post-synaptic spike.

(3) Voltage-dependent plasticity is modeled as [29]

$$\Delta w_i = -A_{\text{LTD}} x_i \left[\bar{u} - \bar{\theta}\right] + A_{\text{LTP}} \bar{x}_i \left[u - \theta\right] \left[\bar{u} - \bar{\theta}\right], \tag{13}$$

$x_i$ denotes the pre-synaptic spike train. $u$ denotes the post-synaptic voltage. $\theta$ denotes a threshold for synaptic weight change to occur, $\bar{x}$, $\bar{u}$ and $\bar{\theta}$ are the low-pass filtered values of $x$, $u$ and $\theta$, respectively.

Based on our experimental study, we find that for the purpose of improving the learning efficiency, synaptic weight updates are not only based on the upper equations. Consideration of meta weight among layers will be very helpful for speeding up the learning process. We define meta weight among layers, denoted as $\text{mw}_{L_{m,n}}$, to represent an additional weight to be multiplied to every synaptic changes among layer $m$ and layer $n$. For example, $i$ and $j$ are two arbitrary neurons from layer $m$ and layer $n$, respectively, and $i$ is the pre-synaptic neuron. By STDP learning algorithms introduced above, their synaptic weight should have a change represented as $\Delta w_i$. $\text{mw}_{L_{m,n}}$ is multiplied to $\Delta w_i$ for the final actual change on the synaptic weights between $i$ and $j$, denoted as $\Delta w_i' = \text{mw}_{L_{m,n}} \times \Delta w$.

Starting from the input layer to the output layer, $\text{mw}_{L_{m,n}}$ is set to $1:1:4:9$ among each neighborhood layers. The ratios are inspired by the fact that the lower layers describe basic features (e.g. edges or angles) [30] while the higher layers are mainly for describing more complex features (e.g. the shapes features), so meta weights in lower layers are lower compared to that of higher layers.

**Table 2** The comparison of classification results based on rules $R1$ to $R6$

| Method | $R6$a | | $R6$b | | $R6$c | |
|---|---|---|---|---|---|---|
| | Acc (%) | Time (s) | Acc (%) | Time (s) | Acc (%) | Time (s) |
| $R125$a | 68 | 304 | 72 | 264 | 65 | 453 |
| $R345$a | 64 | 140 | 67 | 114 | 46 | 165 |
| ***R*12345a** | **72** | **492** | **78** | **367** | **51** | **843** |
| $R125$b | 64 | 285 | 71 | 195 | 55 | 657 |
| $R345$b | 78 | 192 | 75 | 184 | 61 | 920 |
| ***R*12345b** | **84** | **177** | **87** | **141** | **69** | **586** |

## 5 Experimental results and analysis

### 5.1 The experiment settings and procedures

In this paper, MNIST hand writing digit dataset [1)] is used to train and test the performance of proposed multi-layer SNN. Pybrain [31] is used to simulate the models of neurons and synapses. The dataset contains 10 classes of digit images from zero to nine and each image is with the size of $28 \times 28$. The total number of images is 60000 for training and 10000 for testing.

The experiment procedure can be divided in to three phases, namely, initialization phase, training phase, and test phase.

During the initialization phase, the number of neurons in each layer is minimized (here 20 excitatory neurons are assigned for initialization). The number and weights of synapses are crucial for the learning performance. We recommend at least 100 tries for random distribution of these two parameters and picking the best one as random seed. The proportion of excitatory and inhibitory neurons is specifically designed based on learning rules. The maximum training time is set as 1000 s. In this phase, we convert gray-scale values to spike trains in the following ways: Global converting method, in which all the values are compared with a static threshold; Local converting method, in which the new values are formed after convolution and static threshold variable processing. For each image, 200 iterations (1 ms between two neighborhood iterations) are presented to the network.

During the training phase, the number of neurons and synapses, the weights of synapses will be changed based on the integration of different learning rules. The training procedure will be stopped when the changes of the weights (entropy) in the network is less than 5% per second or the maximum training time has reached.

During the test phase, the structure and the variables of the network will not be changed. New data which are not trained will be tested for the performance of the well trained classification network.

### 5.2 Results for integrating rule *R*1 to *R*6

In this paper, seven learning rules are incorporated into the network for a better performance on accuracy. The rules are shown in Table 1, and the results of integrating $R1$ to $R6$ are shown in Table 2. For different kind of background noises ($R5$a or $R5$b), the combinations of four rules ($R1$ to $R4$) show better performance than the combination of two rules ($R1$ to $R2$ or $R3$ to $R4$). Among the rules $R6$a, $R6$b, and $R6$c, the Tri-phasic STDP ($R6$b) shows the best accuracy and convergence time compared with the bi-phasic STDP rule ($R6$a) and voltage-dependent STDP rules ($R6$c), respectively. The input spikes are generated through the OR operations on the results from the local and global features generation procedures introduced in Subsection 2.1.

### 5.3 Results for incorporating rule *R*7

The function of inhibitory neurons in output layer is similar with WTA procedure in artificial networks, in which the neuron spikes in the output layers will give inhibition to other neurons. Two kinds of inhibitory neurons (e.g. anti-E and lock-E types) are introduced into the hidden layers, and their different

1) LeCun Y, Cortes C, Burges C J. The MNIST database of handwritten digits. 1998.

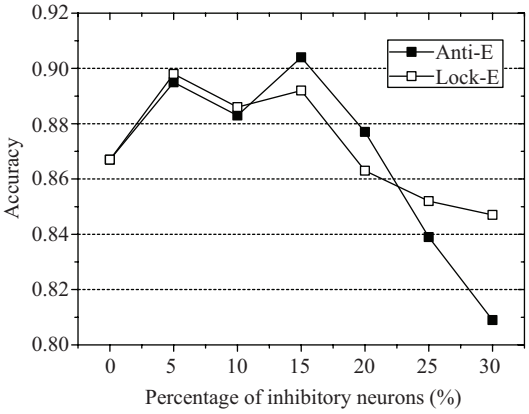


**Figure 4** The accuracies of different proportions and types of inhibitory neurons. Initial synaptic connection is 50%. All the results are based on the integrated rules of $R1$, $R2$, $R3$, $R4$, $R5$b and $R6$b.

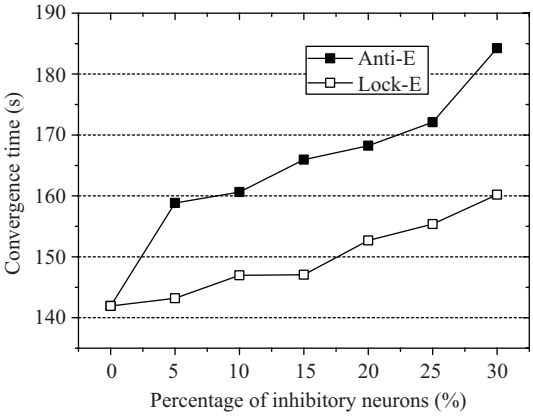


**Figure 5** The convergence time of different proportions and types of inhibitory neurons.

contributions to the learning results are observed. Different proportions (from 0% to 30%) and types (anti-E and lock-E types)of the inhibitory neurons are tested in the experiments and the results are shown in Figure 4. The best accuracy without inhibitory neurons is 86.7% (as shown in Table 2), and when incorporating 5% to 20% inhibitory neurons, the accuracy for classification will be better. With the increase percentage of inhibitory neurons involved in the SNN, the accuracies increased at first, but decreased very obviously later. This shows that proper percentage of inhibitory neurons will help the network achieve better performance on the accuracy. Finally the proportion of 85% excitatory neurons and 15% inhibitory neurons obtains the highest accuracy (i.e. 90.4%) than other groups.

With more inhibitory neurons in the SNN model, the convergence time of the SNNs will be longer, as shown in Figure 5. In addition, the network with lock-E type neurons shows better performance on the convergence time than the network with anti-E type neurons. Take both convergence time and accuracy into consideration, 5% to 15% of inhibitory neurons will be appropriate for good performances.

### 5.4 Higher performances for combining more rules and mechanisms

In the upper experiments, spikes are generated based on the combination of local and global features converting method (based on the OR operation, as introduced in Subsection 2.1). In this paper, the performances of three kinds of strategies are shown in Figure 6. As the result shows, the combination method has better performance than the other two methods (only considering local or global features) on both accuracy and convergence time (the learning rules here is the combination of $R1$, $R2$, $R3$, $R4$, $R6$, $R7$b, and 15% anti-E inhibitory neurons). This result indicates that the proposed spike generation method based on OR operation is rational and effective.

We attempt to validate the hypothesis that gradually incorporating more brain-inspired rules will help to improve the learning accuracy. Figure 7 presents the accuracies based on the gradual integrations of different learning rules. As can be observed from the experimental results, with more brain-inspired learning rules (with careful selection) integrated into the learning procedure, higher accuracies can be achieved.

## 6 Conclusion

Efforts for developing learning mechanisms of SNNs are mostly focusing on spike timing dependent plasticity (STDP), while the actual learning process is supported by even more complex learning rules in the biological brain. Borrowing more brain-inspired rules for SNN learning is expected to be effective for improving its performance.

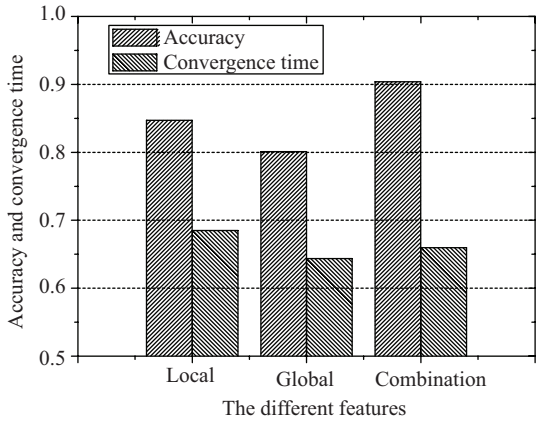


**Figure 6** The accuracy and convergence time of classification results based on different features detection method (the convergence time is normalized).

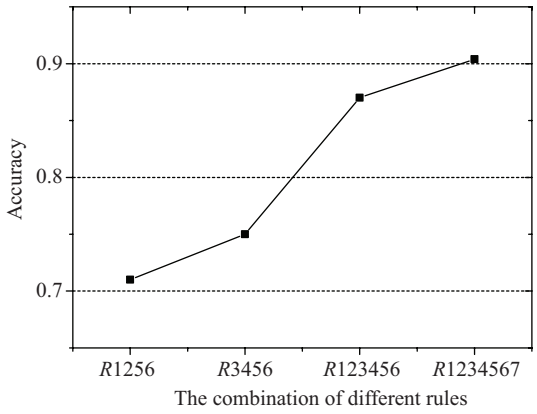


**Figure 7** The improving accuracies based on gradually incorporating more brain-inspired learning rules.

In this paper, with careful consideration and selection, we incorporate seven brain-inspired learning rules into a multi-layer SNN model. The results indicate a potential that with more carefully selected brain-inspired rules integrated into the SNN model, higher accuracies can be achieved. Hence, many learning rules from the brain really can add useful contributions for computational models of cognitive tasks, and each of them have their own effects. In the future, with deeper understanding of the learning mechanisms of the brain, more rules will be found and can be integrated together for developing a better learning model and better performances of SNNs.

**Acknowledgements** This work was supported by Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDB02060007), and Beijing Municipal Commission of Science and Technology (Grant Nos. Z151100000915070, Z161100000216124).

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

1 Hinton G, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets. Neural Comput, 2006, 18: 1527–1554
2 He K, Zhang X, Ren S, et al. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE International Conference on Computer Vision, Santiago, 2015. 1026–1034
3 Maass W. Networks of spiking neurons: the third generation of neural network models. Neural Networks, 1997, 10: 1659–1671
4 Eliasmith C, Stewart T, Choo X, et al. A large-scale model of the functioning brain. Science, 2012, 338: 1202–1205
5 Zenke F, Agnes E, Gerstner W. Diverse synaptic plasticity mechanisms orchestrated to form and retrieve memories in spiking neural networks. Nat Commun, 2015, 6: 6922
6 Song H F, Yang G R, Wang X J. Training excitatory-inhibitory recurrent neural networks for cognitive tasks: a simple and flexible framework. PLoS Comput Biol, 2016, 12: e1004792
7 Beyeler M, Oros N, Dutt N D, et al. A GPU-accelerated cortical neural network model for visually guided robot navigation. Neural Networks, 2015, 72: 75–87
8 Maffei G, Santos-Pata D, Marcos E, et al. An embodied biologically constrained model of foraging: from classical and operant conditioning to adaptive real-world behavior in DAC-X. Neural Networks, 2015, 72: 88–108
9 Wade J J, McDaid L J, Santos J, et al. Swat: a spiking neural network training algorithm for classification problems. IEEE Trans Neur Net, 2010, 21: 1817–1830
10 Beyeler M, Dutt N D, Krichmar J L. Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule. Neural Networks, 2013, 48: 109–124
11 Izhikevich E M. Simple model of spiking neurons. IEEE Trans Neur Net, 2003, 14: 1569–1572
12 Iakymchuk T, Rosado-Muñoz A, Guerrero-Martínez J F, et al. Simplified spiking neural network architecture and stdp learning algorithm applied to image classification. EURASIP J Image Vide Process, 2015, 2015: 1–11
13 Ionescu M, Paun G, Yokomori T. Spiking neural P systems. Fund Inform, 2006, 71: 279–308
14 Zhao Y, Liu X, Wang W. Spiking neural P systems with neuron division and dissolution. PLoS ONE, 2016, 11: e0162882

15 Jia Y, Huang C, Darrell T. Beyond spatial pyramids: receptive field learning for pooled image features. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, 2012. 3370–3377
16 Kravitz D J, Saleem K S, Baker C I, et al. The ventral visual pathway: an expanded neural framework for the processing of object quality. Trend Cogn Sci, 2013, 17: 26–49
17 Häusser M. The hodgkin-huxley theory of the action potential. Nat Neurosci, 2000, 3: 1165
18 Brette R, Gerstner W. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. J Neurophysiology, 2005, 94: 3637–3642
19 Sharkey N E, Jackson S A. An internal report for connectionists. Comput Arc Integrat Neural Symb Proc, 1995, 292: 223–244
20 Destexhe A. Conductance-based integrate-and-fire models. Neural Comput, 1997, 9: 503–514
21 Heeger D J, Ress D. What does fMRI tell us about neuronal activity. Nat Rev Neurosci, 2002, 3: 142–151
22 Sokal R R, Rohlf F J. Biometry: the Principles and Practice of Statistics in Biological Research. New York: WH Freeman and Company, 1969
23 Chrol-Cannon J, Jin Y. Computational modelling of neural plasticity for self-organization of neural networks. Biosystems, 2014, 125: 43–54
24 Ghosh-Dastidar S, Adeli H. Spiking neural networks. Int J Neural Syst, 2009, 19: 295–308
25 Seress L, Ribak C E. Direct commissural connections to the basket cells of the hippocampal dentate gyrus: anatomical evidence for feed-forward inhibition. J Neurocytology, 1984, 13: 215–225
26 Lytton W W, Sejnowski T J. Inhibitory Interneurons Can Rapidly Phase-lock Neural Populations, Chapter for Induced Rhythms in the Brain. New York: Springer, 1992. 357–366
27 Waddington A, Appleby P A, De Kamps M, et al. Triphasic spike-timing-dependent plasticity organizes networks to produce robust sequences of neural activity. Front Comput Neurosci, 2012, 6: 88
28 Song S, Miller K D, Abbott L F. Competitive hebbian learning through spike-timing-dependent synaptic plasticity. Nat Neurosci, 2000, 3: 919–926
29 Clopath C, Büsing L, Vasilaki E, et al. Connectivity reflects coding: a model of voltage-based stdp with homeostasis. Nat Neurosci, 2010, 13: 344–352
30 Rolls E T, Deco G. Computational Neuroscience of Vision. New York: Oxford University Press, 2002
31 Schaul T, Bayer J, Wierstra D, et al. Pybrain. J Mach Learn Res, 2010, 11: 743–746