Studying on Improved Spiking Neural Network in Handwritten Digital Recognition

Lan Jia*, Hongxia Miao, Bensheng Qi
College of Internet of Things Engineering Hohai University. Changzhou, 213000, China
*12257758@163.com

Abstract. In this paper, an unsupervised learning algorithm of spiking neural network (SNN) by using biologically plausible mechanisms is presented for achieving handwritten digital recognition. Firstly, the spike-timing-dependent plasticity (STDP) model was established based on pre- and postsynaptic trace learning rule to determine the connection between neurons. Secondly, genetic algorithm (GA) was used to optimize the initial presynaptic weights and axon delay in the neural network. Finally, the Mixed National Institute of Standards and Technology (MNIST) dataset was trained and tested. Experimental results show that the proposed method can effectively improve the recognition rate of handwritten numbers and realize unsupervised learning of handwritten digital recognition. The accuracy in the MNIST benchmark test is increased by using this improved unsupervised SNN learning scheme. The computational complexity is greatly reduced; therefore, the calculation speed is increased. Simulation result is better than the implementation of the previous unsupervised SNNs.

1. Introduction
With the rapid development of artificial intelligence, especially the extensive application of neural network technology, handwritten digital recognition technology based on neural networks has drawn more and more attention from the academic circles. The accuracy of handwritten digital recognition has been greatly improved by Support Vector Machine (SVM) based on statistical model [1,2]. However too many training parameters and computational complexity make it takes a long time to finish the identification process, moreover the general neurons used in the internal neurons greatly limits application breadth. Convolutional neural networks (CNNs), as a research hotspot in the field of image recognition, achieve great success in a large number of visual processing tasks. Although high accuracy is achieved in classification problems, CNN involves significant computing resources [3]. Furthermore, the phenomenon of over fitting occurs during training process [4,5]. In practice, there is less data with labels, and it is very time-consuming and boring to manually label each data. Besides that, a large number of parameters need to be set and adjusted in convolutional neural network, which restrict the development of CNN.

Therefore, spiking neural networks (SNNs) are proposed to resolve the problem. SNNs are highly bio-realistic and computationally powerful in solving difficult problems in complex networks. The over-fitting phenomenon can be well solved by SNN, and a large number of tags and parameters need not to be set and adjusted. Therefore, in recent years, more and more attention has been paid on how to use SNN to perform complex calculations or to solve pattern recognition tasks. However, designing a SNN that uses a bionic mechanism (especially for learning a new paradigm) remains a challenging
Both the Hodgkin-Huxley (HH) model [6] and Leaky-Integrate-and-Fire (LIF) model [7] achieved supervised learning of handwritten characters, but the training and testing set was too small and the recognition rate was not high.

Previous studies have shown that spike-timing-dependent plasticity (STDP) can be used in SNN to extract visual features of low or intermediate complexity in an unsupervised manner [8,9]. Therefore, the pre- and postsynaptic spiking weights of SNN can be calculated by pre- and postsynaptic trace learning rule with a better training effect. The genetic algorithm (GA) is an efficient parallel global search algorithm, which has good robustness and succeeds in solving the global optimization problem [10,11]. Therefore, GA can be applied to the learning process of SNN. The recognition accuracy and the training speed of SNN are improved by using the GA to optimize the initial weight and axon delay of SNN. Therefore, the STDP model based on pre- and postsynaptic trace learning rule is built and the initial presynaptic weights and axon delay are optimized by GA in the SNN. Three different learning rules were simulated in this paper. What can be obtained through the simulation results is that the training speed is improved by 22.5% and the recognition accuracy is increased by 10.6%.

2. SNN model
   The SNN model is built mainly by neuronal and synaptic model.

2.1. Neuronal and synaptic model
   In order to establish a neuron dynamic model, we chose the leaky integrate-and-fire model. The membrane voltage \( V \) is described as:
   \[
   \tau \frac{dV}{dt} = (E_r - V) + c_e (E_e - V) + c_i (E_i - V)
   \]  
   (1)

   where \( E_r \) is the resting membrane potential, \( E_e \) and \( E_i \) are the equilibrium potentials of excitatory and inhibitory synapses respectively, \( c_e \) and \( c_i \) are the conductance of excitatory and inhibitory synapses respectively, and \( \tau \) is the time constant. When the neuronal membrane potential crosses its membrane threshold \( v_t \), the neuron fires and its membrane potential is reset to \( v_r \). Within a few milliseconds after the resetting, the neuron is in its refractory phase and cannot be stimulated again.

   Synapses are modeled by changes of conductance. If a presynaptic spike reaches the synapse, the synaptic increases its conductance by synaptic weights instantaneously, otherwise the conductance shows an exponential decay. If the presynaptic neuron is excitatory, the expression for the conductance of excitatory synapses \( c_e \) is:
   \[
   \tau_{e} \frac{dc_e}{dt} = -c_e
   \]  
   (2)

   where \( \tau_{e} \) is the time constant of the excitatory postsynaptic potential. Similarly, if the presynaptic neuron is inhibitory, then the conductance of inhibitory synapses \( c_i \) is updated by using the same equation, but with a time constant of the inhibitory post-synaptic potential \( \tau_{i} \):
   \[
   \tau_{i} \frac{dc_i}{dt} = -c_i
   \]  
   (3)

3. Improved SNN based on genetic algorithm

3.1. System flowchart of improved SNN
   The system flowchart of the improved SNN using GA is shown in Figure 1. Firstly, the network structure should be built, then weight and connection value should be set, secondly, the IiPe, IiPi, PePi,
PiPe matrices (Ii represents input neurons of the input layer, Pe represents excitatory neurons of the processing layer, Pi represents inhibitory neurons of the processing layer) should be created, then MNIST dataset (image metadata, tag metadata) should be downloaded, thirdly, parameters and equations should be set, then presynaptic weights and axon delay are optimized by GA, then network population, recurrent connections and input population should be created, after that, the simulation should be simulated, finally, the result should be saved and output.

The network structure is shown in Figure 2.
The first layer containing $28 \times 28$ neurons (each neuron represents an image pixel) is the input layer, the second layer containing 400 excitatory neurons and the same number of inhibitory neurons is the processing layer. The input of the network is a series of Poisson distribution sequences obtained by the input encoding. The excitatory neurons of the second layer are connected in a one-to-one fashion to inhibitory neurons. Each inhibitory neuron connects with all excitatory neurons except the one that it receives the spike. This connectivity provides lateral inhibition and leads to competition between excitatory neurons. This means that once a neuron selected will prevent other neurons from firing.

3.2. Improved learning rule

Based on the presynaptic trace learning rule, the pre- and postsynaptic trace learning rule was proposed. The equations are set by using improved learning rule (pre- and postsynaptic trace learning rule).

Pre- and postsynaptic trace learning rule is as follows:

The weight of presynaptic spike of the SNN for handwritten digital recognition is determined by

$$\Delta w = -\eta_1 x_2 w^\mu$$

(4)

where $\eta_1$ is learning-rate for a presynaptic spike, $x_2$ is the postsynaptic trace value, and $\mu$ determines the dependence of the update on the previous weight.

The weight for postsynaptic spike of the SNN for handwritten digital recognition is

$$\Delta w = \eta_2 (x_1 - x_3)(w_{\max} - w)^\mu$$

(5)

where $\eta_2$ is learning-rate for a postsynaptic spike, $x_1$ is the presynaptic trace value, $x_3$ is the target average value of the presynaptic trace at the moment of a postsynaptic spike, and $w_{\max}$ is the maximum weight.

In training process, in order to achieve better convergence, when a presynaptic spike arrives at the synapse every time, the trace is increased by 1, otherwise $x_i$ decays exponentially. The expression for the presynaptic trace value $x_i$ is:

$$\tau_{\delta i} \frac{dx_i}{dt} = -x_i$$

(6)

where $\tau_{\delta i}$ is the time constant of the presynaptic trace value. The expression for the postsynaptic trace value $x_2$ is:

$$\tau_{\delta 2} \frac{dx_2}{dt} = -x_2$$

(7)

where $\tau_{\delta 2}$ is the time constant of the postsynaptic trace value. The expression for the target average value of the presynaptic trace at the moment of a postsynaptic spike $x_3$ is:

$$\tau_{\delta 3} \frac{dx_3}{dt} = -x_3$$

(8)

where $\tau_{\delta 3}$ is the time constant of the target average value of the presynaptic trace at the moment of a postsynaptic spike.
3.3. Improvement of SNN by GA
Presynaptic weight and axon delay are optimized by GA. The initial random population of organisms is generated to randomly select presynaptic weight values and axon delays. SNN is encoded as a chromosome. As shown in Figure 3, each chromosome consists of two parts: X1 and X2. The weight between the corresponding presynaptic neurons and their postsynaptic neurons is shown in each fragment of X1; the connection delay between the corresponding presynaptic neuron and its postsynaptic neuron is represented by each fragment of X2. Each next generation is made up of the elite (15%), crossover (55%) and mutation (30%) of the previous generation. The elite are the best chromosomes, so they are moved directly to the next generation. Due to the longer chromosome length, 5 cut points were randomly selected from each parent for exchange. A typical crossover with two cut points is shown in Figure 4.

![Fig. 3 Composition of a typical chromosome](image)

![Fig. 4 Typical cross-process diagram with two cut points](image)

The initial presynaptic weights are optimized in SNN by using GA, which is divided into seven steps, as shown in Figure 5. Implementation of the algorithm is as follows:

1) The initial random populations of SNNs for handwritten digital recognition are generated and encoded as chromosomes;
2) Each chromosome is divided into two parts: Each fragment in X1, X2, X1 represents the connection weight between the pre- and postsynaptic neurons of the corresponding SNN for handwritten digital recognition. Each fragment of X2 represents the axon delay between the pre- and postsynaptic neurons of the corresponding SNNs used for handwritten digital recognition, which is shown in Figure 4;
3) The optimal values of the current pre- and postsynaptic neuron connection weight and axon delay are calculate;
4) Determine whether the biological population algebra of SNN, which is currently used for handwritten digital recognition, is less than the set maximum evolutionary algebra, if yes do (5), otherwise do (7);
5) The GA operation of the corresponding SNN for handwritten digital recognition is done by three steps:
   5.1) The chromosomes representing the connection weight and the axon delay between the pre- and postsynaptic neurons of the SNN used for handwritten digital recognition are crossed with a crossover probability $P_c$ (55%);
(5.2) The chromosomes representing the connection weight and the axon delay between the pre- and postsynaptic neurons of the SNN used for handwritten digital recognition are mutated with a mutation probability $P_2$ (30%);

(5.3) The chromosomes representing the connection weight and the axon delay between the pre- and postsynaptic neurons of the SNN used for handwritten digital recognition are chosen by using elitist selection, the fitness function is

$$f_i = 1 - \frac{E_i}{S}, i = 1, 2, \cdots, N$$

$E_i$ is the individual error function of each neuron in the SNN; $S$ is total error of the SNN; $N$ is the number of the SNN individuals, the chromosomes with higher fitness are copied directly to the next generation without crossover and mutation, elitist selection probability is $P_3$ (15%);

(6) Do the operations of the next generation, and return to do (3);

(7) End the optimization of pre- and postsynaptic initial weight and axon delay in SNN by GA and save the final result.

As shown in Figure 5, the iteration is stopped, when the evolutionary computation of GA has reached a certain number of iterations. The optimal chromosome (optimal value of presynaptic initial weight of SNN) is transferred to SNN, and MNIST handwritten digital set is input to the network for unsupervised learning training.

4. Simulation

To simulate the three SNNs of different rules we used Python in Linux of VMware Workstation. A network of 400 excitatory neurons and 400 inhibitory neurons are trained and tested by inputting MNIST dataset containing 60,000 training samples to the network three times. The MNIST dataset is The MNIST data set contains 28 × 28 pixel images of the digits 0–9.

Poisson distribution of the spiking sequence during 350ms input to the network, the firing rate of the spiking sequence is proportional to the pixel intensity of the MNIST dataset image. A two-dimensional matrix is converted into a one-dimensional array of 1×784 where white indicates a pixel point, otherwise black. The same number in the one-dimensional array representation has a certain similarity. Divide the image intensity factor (0-255) of 784 (28×28) pixels in the image by 4 to get the firing rate (0-63.75Hz) of the input pulse sequence.
As shown in Figure 6, spiking situations of excitatory neurons and inhibitory neurons are recorded by spike monitors of excitatory neurons and spike monitors of inhibitory neurons respectively. In Figure 6, the above figure shows the number of excitatory neurons spiked at each time and the following figure shows the number of inhibitory neurons spiked. It can be seen that the response phase of neurons is 350ms and the refractory period is 150ms. The data is displayed in the form of images. Only the data of last 1000ms is shown in the figure for it will be updated every 1000ms.

As shown in Figure 7, weights (from 784 to $28 \times 28$) of the connections from input to excitatory neurons of the network with 400 excitatory neurons in a $20 \times 20$ grid are rearranged. Each row and column has 560 ($20 \times 28$) pixels. The spiking sequences input to the network excite or inhibit excitatory neurons and inhibitory neurons and then the training is completed. The spiking threshold of each neuron is modified, and according to the highest response of each neuron to the ten digits of 0-9, a category is assigned to it, and the response for each neuron in each category is averaged. Finally, select the category with the highest average spiking rate to determine the number displayed by the image. So the number displayed on each grid represents the number of the highest response times of this excitatory neuron. Different colors of numbers represent different weights. The darker color means the higher weight the number represents.
As shown in Figure 8, the recognition rate of presynaptic learning rule is blue dash-and-dot line, the recognition rate of pre- and postsynaptic trace learning rule presented in this paper is red line with star, and the GA-optimized recognition rate is green solid line.

Compared with the presynaptic trace learning rule, the recognition rate of handwritten digits can be improved and the training time can be reduced by pre- and postsynaptic trace learning rule. Using GA-optimized SNN can further improve the recognition accuracy to 96% and reduce the training time to 14 hours, as shown in Table I.
Table 1. Performance Comparison of Different SNN.

| Different SNNs                  | Training time(hour) | Test accuracy |
|--------------------------------|---------------------|---------------|
| Presynaptic trace learning rule| 18.2                | 86.8          |
| Pre- and postsynaptic trace learning rule | 16.7                | 90.0          |
| GA optimization               | 14.1                | 96.0          |

What we can see from the table I is that compared with the SNN based on presynaptic trace learning rule, the SNN based on pre- and postsynaptic learning rule shortened one and a half hours of training time and increase 3.2% test accuracy. The SNN based on pre- and postsynaptic trace learning rule and GA optimization effectively improves the training time and the test accuracy, completing unsupervised learning more efficiently and accurately than using presynaptic trace learning rule.

Table 2. Classification Accuracy of Other SNNs.

| Architecture                     | (Un-) supervised | Learning-rule                          | Accuracy   |
|----------------------------------|------------------|----------------------------------------|------------|
| Spiking RBM [12]                 | Supervised       | Contrastive divergence, linear classifier | 89.0%      |
| Spiking convolutional neural network [13] | Supervised       | Tempotron rule                          | 91.3%      |
| Multi-layer hierarchical network [14] | Supervised       | STDP with calcium variable              | 91.6%      |
| Two-layer network [15]           | Unsupervised     | Rectangular STDP                        | 95.0%      |
| Two-layer network (this paper)   | Unsupervised     | GA optimization                         | 96.0%      |

What we can see from the table II is that compared with other approaches to (un-)supervised learning with SNNs, our method shows a higher classification accuracy.

5. Conclusion
In order to solve the problem that there are training samples based on unknown (unmarked) categories in pattern recognition and perform unsupervised learning on MNIST handwritten digital set, the imitational biological neural network is proposed in this paper using pre- and postsynaptic trace learning rule and GA. Unsupervised learning of MNIST handwritten digit sets is implemented in the MNIST benchmark test. The three SNNs using different rules (presynaptic trace learning rule, pre- and postsynaptic trace learning rule, and GA optimization) were simulated. The training speed and the recognition accuracy are improved. At the same time the running time and the parameters that need to be set are reduced and the calculation space is saved greatly. Finally, simulation example is conducted to verify the effectiveness of the proposed method.

Acknowledgments
This work was supported by Science and Technology Support Program (industry) of Changzhou under Grant No. CE20160068.

References
[1] Tuba E, Bacanin N. An algorithm for handwritten digit recognition using projection histograms and SVM classifier [C] Telecommunications Forum Telfor. IEEE, 2016:464-467.
[2] Abbas N, Chibani Y, Nemmour H. Handwritten Digit Recognition Based on a DSmT-SVM Parallel Combination [C] International Conference on Frontiers in Handwriting Recognition. IEEE, 2015:241-246.
[3] Han B, Sengupta A, Roy K. On the energy benefits of spiking deep neural networks: A case
study [C] International Joint Conference on Neural Networks. IEEE, 2016:971-976.
[4] Shopon M, Mohammed N, Abedin M A. Image augmentation by blocky artifact in Deep Convolutional Neural Network for handwritten digit recognition [C] IEEE International Conference on Imaging, Vision & Pattern Recognition. IEEE, 2017:1-6.
[5] Niu X X, Suen C Y. A novel hybrid CNN–SVM classifier for recognizing handwritten digits [J]. Pattern Recognition, 2012, 45 (4):1318-1325.
[6] Bhuiyan M A, Jalasutram R, Taha T M. Character recognition with two spiking neural network models on multicore architectures [C] IEEE Symposium on Computational Intelligence for Multimedia Signal & Vision Processing. IEEE, 2009:29-34.
[7] Rice K L, Bhuiyan M A, Taha T M, et al. FPGA Implementation of Izhikevich Spiking Neural Networks for Character Recognition [C] International Conference on Reconfigurable Computing and Fpgas. IEEE Computer Society, 2009:451-456.
[8] Kheradpisheh S R, Ganjtabesh M, Thorpe S J, et al. STDP-based spiking deep convolutional neural networks for object recognition [J]. Neural Networks, 2017, 99:56.
[9] Skorheim S, Lonjers P, Bazhenov M. A spiking network model of decision-making employing rewarded STDP [J]. Plos One, 2014, 9 (3): e90821.
[10] Goldberg D E. Genetic Algorithm in Search, Optimization, and Machine Learning [J]. 1989, xiii (7):2104–2116.
[11] Bagde P R, Gurjar A A. A handwritten recognition for free style Marathi script using genetic algorithm [C] International Conference on Global Trends in Signal Processing, Information Computing and Communication. IEEE, 2017:43-48.
[12] Merolla P, Arthur J, Akopyan F, et al. A digital neuromorphic core using embedded crossbar memory with 45pJ per spike in 45nm [C] Custom Integrated Circuits Conference. IEEE, 2011:1-4.
[13] Diehl P U, Neil D, Binas J, et al. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing [C] International Joint Conference on Neural Networks. IEEE, 2015:1-8.
[14] Beyeler M, Dutt N D, Krichmar J L. Categorization and decision-making in a neurobiologically plausible spiking network using a STDP-like learning rule [J]. Neural Netw, 2013, 48 (10):109-124.
[15] Querlioz D, Dollfus P, Bichler O, et al. Learning with memristive devices: How should we model their behavior? [C] Ieee/acm International Symposium on Nanoscale Architectures. IEEE, 2011:150-156.