An SNN Construction Method Based on CNN Conversion and Threshold Setting

Ying Shang, Yongli Li, Feng You
Department of Computer Science, Beijing University of Chemical Technology, Beijing, China
shangy@mail.buct.edu.cn, Liaibm@inbox.worktile.com, youf@mail.buct.edu.cn

***Abstract*—We present a method to converse the Convolutional Neural Network (CNN) to the Spiking Neural Network (SNN), and we post a threshold adjustment algorithm for SNN. First, the adjustment strategy for CNN is introduced. Then after training, the weight parameters in the model are extracted, which is the corresponding synaptic weight in the layer of the SNN. Finally, a new threshold-setting algorithm based on feedback is proposed to solve the critical problem of the threshold setting of neurons in the SNN. We evaluate our method on the Cifar10 data sets released by Hinton's team. The experimental results show that the image classification accuracy of the SNN is more than 98% of that of CNN, and the theoretical value of power consumption per second is 3.9 *mW*.**

Keywords—Convolutional Neural Network; Spiking Neural Network; Threshold Setting Algorithm

I. INTRODUCTION

CNN is an artificial neural network with human visual processing as the model. It is the most successful network for image classification, and it has been widely used in the fields of target recognition, target detection, target positioning, and so on. However, due to the limitations of CNN, it requires a lot of computational power and training data. [1-3]

SNN is the neural network model closest to biological neural network at present, with strong computing power, which makes the application of SNN gradually increase, such as robot control, pattern recognition, speech recognition [4-7], image recognition [8-10], and target detection [11] and so on. SNN has similar supervised learning strategies with the traditional artificial neural network [12]. Still, the network model transmits the spike-timing signal, and the error function is not continuously differentiable, so it cannot use a monitoring algorithm for network training like a traditional neural network. [13,14]

Therefore, using the mature algorithm and framework of CNN to get the network model to solve related problems, and then converse CNN to SNN, this has become a direction that can be explored. In this way, both CNN and SNN can be considered, and the training can be carried out with the help of the mature CNN algorithm method, which also provides a solution for the application of SNN.

The main contributions of this paper are summarized below. First, we address a method to transfer the weight parameters in the CNN model to the synaptic weight in the SNN and realize the conversion from CNN to SNN. Second, a threshold-setting algorithm based on feedback adjustment is proposed to set the threshold of the neurons in SNN. Finally, experiment results based on benchmarks show that the proposed method can converse CNN to SNN, therefore to reduce the power consumption and the threshold adjustment algorithm is effective, which can improve the SNN classification accuracy.

The rest of this paper is organized as follows. In Section II, related work is discussed. Section III introduces our proposed method. In Section IV, we perform a comprehensive set of experiments on image classification. Finally, we conclude our work and suggest directions for future work.

II. RELATED WORK

SNN is a third-generation neural network, which can more truly simulate the information transmission mechanism between biological neurons. SNN is modeled using the synaptic transmission mechanism of biological neurons. This mechanism can describe various neuronal action potential triggers and transitions, as well as electroweak sequence coding operations. Based on this unique information transmission mechanism, we can imitate the mechanism of neurons in the brain to process information. Researchers proved that the SNN model has a stronger sense of calculating think than the first generation and second generation of the neural network. [15,16] Common SNN models include HH model [17], IF model, SRM model [18], Izhikevich model [19], etc.

$$\begin{cases} V(t) = V(t\text{-}1) + L + X(t) \\ IFV(t) \geq \theta \text{ , } produce\ pulse\ and\ V(t) = 0 \\ IFV(t) \prec \quad set\ V(t) = V_{\min} \end{cases} \tag{1}$$

In the above equation, L is a constant parameter, and X(t) is the sum of the inputs of all the synapses connected to the neuron in time. Once V(t) exceeds the threshold, this neuron is activated and generates a spike, and its membrane potential V(t) is reset to 0. The membrane potential of the neuron is not allowed to be set lower than Vmin, and Vmin is usually set to 0, which conforms to the biological characteristics of the neuron. Setting the threshold is an important issue.

For a neuron in the convolutional layer, X(t) can be defined as follows:

$$X(t)=\sum A(t)\ K \tag{2}$$

A(t) is the input spike from the previous layer, and it has a value of 1 or 0; 1 represents spike input, 0 represents no spike input; K is the weight of the convolution kernel connected to the neuron.

DOI reference number: 10.18293/SEKE2020-112

In the SNN model, the spike-timing signal is transmitted, and the error function is not continuously differentiable, which makes it impossible to train the network with a supervisory algorithm like the traditional neural network. However, CNN's mature algorithm and framework are used to get the network model to solve related problems, and then CNN is mapped to SNN, which becomes a direction to be explored.

Some researchers have studied the conversion of CNN to SNN. Cao and Grossberg [20] studied how the frequency-based perceptual neural network model could be transformed into an SNN model without losing performance, but they did not give a specific implementation of this method. The pulse-like HMAX method is the basic implementation of the original HMAX algorithm [21], but it does not explain how to transform the general HMAX structure into the pulse and CNN structure. Masquelier and Thorpe used unsupervised STDP method to learn unclassified features [22] and then used non-impulse classifier to classify these features, which resulted in the fact that the recognition results could not reach the accuracy of CNN similar to them. Perez-Carrasco [23] USES kernel projection to calculate convolution, but this method requires a lot of hardware resources. Xing F's method can be used to large-scale employment of spiking networks [24].

To analyzing the neurons, network structure, and network input of the two neural networks, the simulation of visual processing can be realized on SNN. This paper proposes a method to converse CNN to SNN. After CNN conversed to SNN, the threshold value of the neuron group in SNN needs to be set so that SNN can achieve the classification effect of CNN. In this paper, cifar10's CNN recognition is taken as an example, and a threshold setting method is proposed. After experimental verification, the SNN achieves the classification result close to that of CNN after applying the threshold setting method.

III. SNN CONSTRUCTION METHOD

In this section, we introduce our method to construct an SNN model. First, build a CNN model, and obtain the number of convolution kernels in the model, the combination of pooling layers, the number of iterations, and the learning rate. To skipping the difficult training problem of SNN, we map the CNN model weight parameters to SNN to achieve CNN to SNN conversion. To improving the accuracy of SNN classification, the threshold adjustment algorithm based on feedback adjustment is designed by using the idea of feedback adjustment algorithm (BP) in SNN training to set the threshold of neuron groups in impulsive neural networks. Create SNNs with high computational power and high image classification accuracy.

A. Converting to a Spiking Neural Network

The differences between SNN and CNN are reflected in the following three aspects:

1. Neuron: CNN's model is designed with biological neurons as the model. The main operation on CNN is convolution operation. The SNN model is the simulation of biological characteristics of biological neurons and the real simulation of biological neurons. The neurons participate in the complex integration operation and can realize the simple operation of convolution neurons.

2. Network structure: the structure of the CNN model is a typical multilayer feedforward network structure, which is similar to the feedforward network structure on the SNN model. Therefore, a similar network structure with CNN can be realized in the SNN model.

3. Input: CNN takes the pixel value of the image as the input of the network, while SNN is the spike converted from the image. A large number of studies have shown that the spike train following the Poisson distribution can well simulate the discharge of neurons, and the resulting spike train can be used as the input of the feedforward neural network.

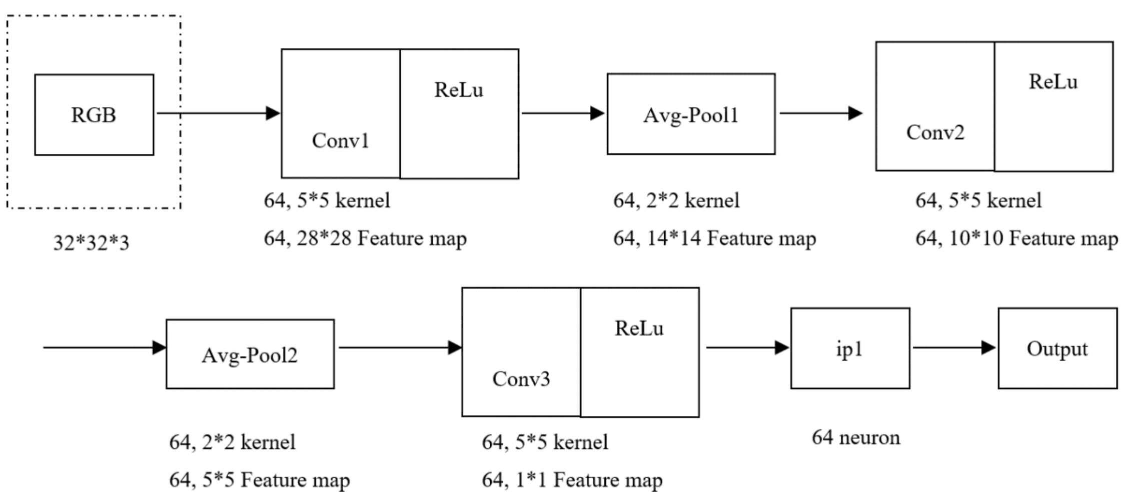


Figure 1. *detailed structure of the adjusted CNN*

Therefore, converse CNN to SNN, the parameters of the CNN model must be applied to SNN. Since SNN processes spike information, if the spike signal transmitted from the previous layer is received, the synaptic weight coefficient of the neuron is 1; otherwise, it is 0, so no negative value will appear in the output layer of SNN. Therefore, the LRelu activation function used by the convolution layer is replaced by the Relu function of non-negative conversion. Then, no bias term is used in each layer of CNN, and the value of bias is set to 0. Finally, to reduce the errors caused by the increase in the number and complex structure of neurons, the maximum pooling that requires the two-layer group of neurons is realized through average pooling. The adjustment of CNN is summarized as follows. Figure 1 is the structure of the adjusted CNN.

B. Spike generation layer

In SNN, images need to be converted into spike signals in order to be used as network input. Therefore, before the input layer of SNN, there is the spike generation layer, whose function is to convert images into spike signals.

The response function of the spiking neuron is defined as follows:

$$\rho(t) = \sum_{i=1}^{k} \delta(t - t_i) \tag{3}$$

K is the number of spikes in the spike train, and t is the arrival time of the spike. According to the properties of the spike function, the number of received spikes in a minor interval can be calculated by the formula (4):

$$n = \int_{t1}^{t2} \rho(t) dt \tag{4}$$

The instantaneous discharge frequency of the spiking neuron is expressed by the expectation of the neuron response function.

$$\gamma(t) = \frac{dn(t)}{dt} = E(\rho(t)) \tag{5}$$

The mean value of the neuron response function within a short time interval can be approximated as the firing frequency of the neuron, as shown in formula (6):

$$\gamma_M(t) = \frac{1}{M} \sum_{j=1}^{M} \rho_j(t) \tag{6}$$

If the spike generation is assumed to be independent of each other, and the instantaneous discharge frequency is a constant. Set the number of spikes in the period as k, and then the probability of containing n spikes in the period can be expressed by formula (7):

$$P(n, t_1, t_2) = \frac{k!}{(k-n)!n!} p^n q^{k-n} \tag{7}$$

$p = \frac{(t_1 - t_2)}{T}$, $q = 1 - p$, while $k \to +\infty$ and the average discharge frequency is a constant, which can be substituted into equation 4-5 to get formula 4-6:

$$P(n, t_1, t_2) = e^{-\gamma\Delta t \frac{\gamma\Delta t^n}{n!}} \tag{8}$$

The spike generation layer converts the input image into the Poisson spike-timing signal, which is described as follows: I_{ijk} *(k=1,2,3)* is the image map of the input to the spike production layer. At time t, if the spike production function *spike( )<c* I_{ijk}*(k=1,2,3)* Where, the function of *spike(j,j)* is to generate a number following the distribution *(0,1),* and c is a constant, which is used to scale the spike frequency generated. Each simulation time can be set to milliseconds. After passing through the spike generation layer, the image is transformed into the spike signal, and the gray images of R, G, and B can be obtained after processing.

The adjusted CNN was trained several times to obtain a better detection model, which was used as a model of the SNN. The convolution kernel weight was mapped to the neuron synapse value of the SNN. Figure 2 is the detailed network structure diagram of the SNN.

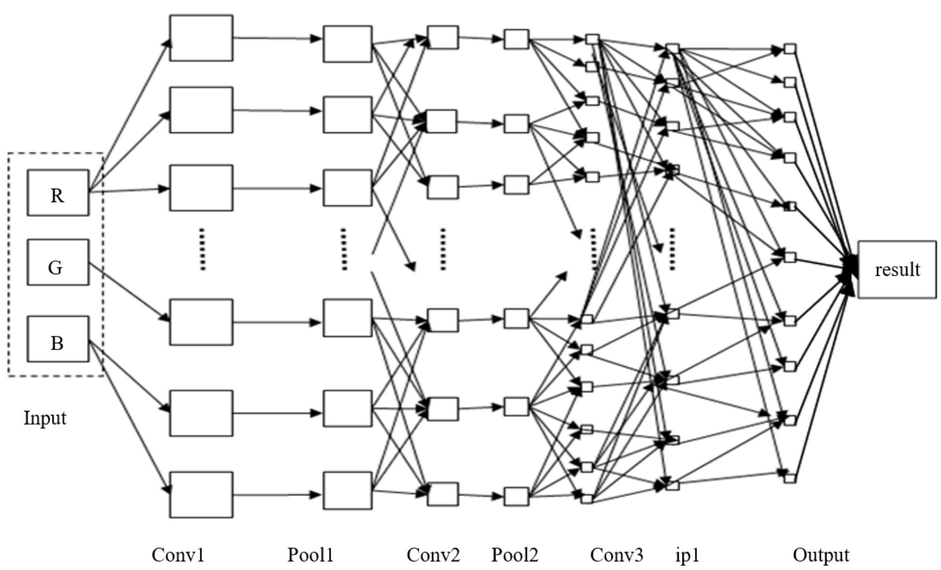


Figure 2. *detailed block diagram of the SNN*

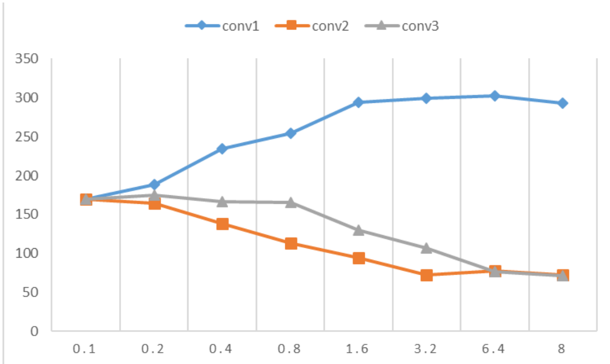


Figure 3. *Eeffect of threshold values on neuron groups conv1, conv2 and conv3 on output*

C. *Threshold Setting Algorithm*

In SNN, the threshold determines the spike generation of neurons. Only when the value of the neuron is greater than the threshold value, the neuron will generate a spike and transmit it to the lower neuron group. Therefore, the setting of the threshold will also have an impact on the characteristics extracted from SNN.

In Figure3, the horizontal axis is the threshold value, and the vertical axis represents the number of spikes received by the output neurons corresponding to the input. The higher the value is, the more accurate the classification is. It shows that, for the neuron group in SNN, when the threshold is too small, many features will be transferred to the next layer, which will include many features that have not been extracted, resulting in low accuracy of the final classification results. However, when the threshold is too large, the threshold, resulting in insufficient feature information and low accuracy of classification results, will filter out the features that can be used as classification.

Therefore, it is necessary to select an appropriate threshold for the neuron group so that the classification can be completed to the maximum extent, and the classification accuracy of the network can be improved.

We apply the idea of the back-propagation algorithm to the SNN threshold setting and propose a threshold adjustment algorithm based on feedback. This algorithm combines the threshold with the final classification effect. First, each layer is initialized with a threshold, and the initial value is set to 0, then set the maximum value for the neuron group to be adjusted, and then use the binary search method to set the threshold. The method to determine whether a threshold is optimal is:

Through the output feature map and convolution of the current neuron group The neural network corresponds to the output feature map, and observes whether the features of the feature map in the CNN and the SNN are consistent, that is, whether the brighter parts of the feature map are consistent;

Calculate the number of pulses of output neurons corresponding to the image in the output result, and determine whether the value is the maximum value;

Algorithm Threshold Setting Algorithm Based on Feedback

Input: a category picture in the classification, the threshold pi of the neuron group

Output: The ratio of the number of spikes of the corresponding neuron to the total number of output spikes

function threshold setting (*p1, p2, p3*)
Initialize the three-layer threshold in the SNN to *0.1, 0.1, 0.1*
Store the output of the network at max and n is the step size of the threshold change
repeat
    for all (*p1, p2, p3*) do
        $i = 1,2,3;$ // represents the number of neurons
        Calculate the output result of the network *max1* when it is the *conv* layer *conv[i]* and the threshold $p[i] = p[i] + n$
        if *max1> max*
            Update threshold $p[i] = p[i] + n$
        else $p[i] = p[i] - n$
    end for
until the stop condition is reached
Output the threshold of the neuron group
end function

When the above two conditions are met, the threshold value is determined to be the threshold value of the threshold group, and then the next group of threshold group debugging until the final debugging is completed. The algorithm is as fs.

IV. Experiment and Analysis

A. *Experimental Design*

We carried out experiments on the Cifar10 data set, which is provided by the Hinton team. It is a common data set for classification problems. It consists of 10 categories: aircraft, cars, birds, cats, deer, dogs, frogs, horses, boats, and trucks. There are 60,000 images in the data set, which are divided into 50,000 train image samples, 5,000 for each category. Test 10,000 image samples, 1000 for each category. The size of the images in the data set is 32*32, and they are all color images. Table list the data set and the CNN classified result.

The experimental hardware environment was CPU Intel(R) 3.6GHz, RAM 32.00GB, the operating system was Ubuntu16.04, and the program compilation environment was Python2.7. The framework of CNN is Caffe.

B. *Experimental Results and Analysis*

This paper conducts experiments and analysis of the following two research questions (RQ):

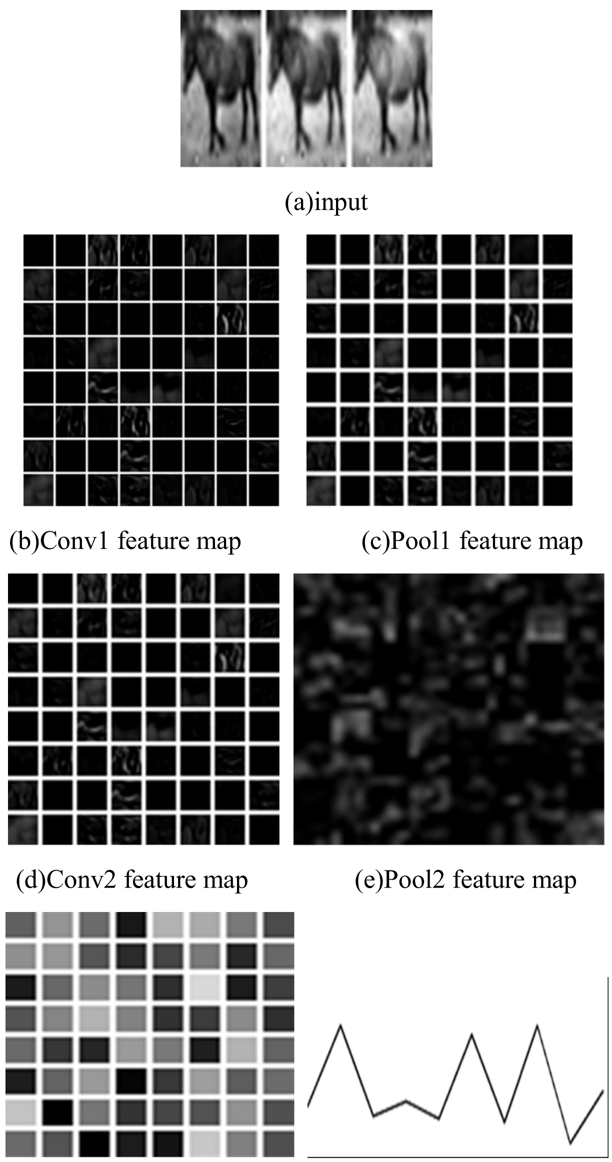
RQ1:In this paper, CNN conversed to SNN. Whether the SNN consistent with CNN?

RQ2:What is the influence of the threshold-setting algorithm based on feedback on the classification accuracy of SNN?

1) RQ1

The consistency of the network model refers to the statistical analysis of the output results of the network model under the same data set and the analysis of the number of correctly classified and wrongly classified pictures on the data set. The higher the number is, the higher the consistency of the model will be. This section verifies the consistency of the unadjusted CNN model, the adjusted CNN model, and the SNN model on the test in the cifar10 data set.

Input a 32*32 RGB color image, when changing the image into a spike signal, the RGB separately coded into three ones. Figure 4 shows the feature map in the SNN. For the limitation of length, this paper only lists the feature graphs and output results of each layer in SNN. The characteristics of each layer feature graph corresponding to CNN are consistent, and the prediction trend of SNN results is consistent with that of CNN.



(a)input

(b)Conv1 feature map (c)Pool1 feature map

(d)Conv2 feature map (e)Pool2 feature map

(f) Conv3 feature map (g)output

Figure 4. *(a) ~ (g) are the output characteristics and results of each layer of the SNN*

2) RQ2

Through the threshold-setting algorithm, the threshold value of the network is obtained, as shown in Table 1 below.

In order to verify the effectiveness of the threshold-setting algorithm proposed in this paper, we randomly selected 100 pictures of each category in the test data set for the experiment. The classification accuracy is shown in Figure 5, where the threshold-setting algorithm is identified as C_SNN.

We found that after applying the threshold adjustment algorithm, the classification accuracy of ship and truck was slightly lower than that of no threshold setting, and the classification accuracy of other categories was improved.

TABLE I. THRESHOLDS FOR EACH NEURON GROUP OF SNN

Neuron	**Threshold**
Conv1	6.8
Pool1	0.75
Conv2	0.1
Pool2	0.75
Conv3	0.1
ip1	0.7
ip2	0.7

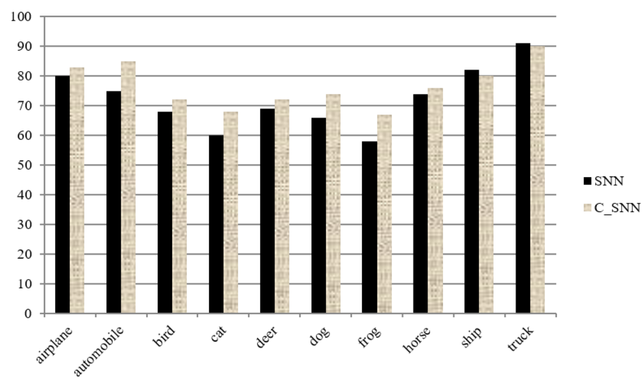


Figure 5. *Accurate Classification of SNN before and after Threshold Setting*

TABLE II. CLASSIFICATION ACCURACY OF ALL THE MODELS

	CNN (%)	**Adjust CNN (%)**	**C_SNN (%)**	**C_SNN/CNN (%)**
airplane	83.5	82.3	82	98.2
automobile	85.8	85.2	84.7	98.7
bird	71.7	71.5	70.6	98.5
cat	67.5	66.9	65.8	97.5
deer	84.2	83.6	82.8	98.3
dog	73.2	71.8	71.5	97.7
frog	77.1	76.9	75.4	97.8
horse	79.5	79.3	78.9	99.2
ship	86.7	85.7	84.9	97.9
truck	94.7	92.8	91.5	96.6
Ave	**80.39**	**79.6**	**78.8**	**98.04**

Table2 shows the statistical accuracy of each network model. The accuracy of SNN can reach 98% of that of CNN. The SNN obtained by CNN conversion can achieve the target classification problem and achieve good results.

Compared with CNN, the power consumption of SNN is lower. The theoretical analysis of SNN energy consumption shows that there are about 14200,000 synapses in the SNN model in this paper, and each synapse consumes a small amount of focus. Assuming that this model can process 742 pictures per second, the power consumed per second of this model can be calculated by the following formula:

$$P_{SNN}=1.42\times10^{7}\times742\times\alpha\ W \tag{9}$$

According to the data provided by Cruz-Albrecht, the energy consumption of each neuromorphic circuit isα= 0.37 mJ, so

$$P_{SNN} \approx 3.9\ mW \tag{10}$$

Under the same hardware system, compared with the CNN, the SNN consumes less energy and has higher efficiency.

V. Conclusion and Future Work

This paper studies the problem of image classification using CNN and SNN. A method of conversion CNN to SNN is proposed, and an adjustment strategy is designed for CNN to reduce the errors caused by the conversion. Through these adjustments, the spike generation layer, convolution layer, pooling layer, and output layer are constructed by using the feedforward network structure. The weight parameters in the adjusted CNN model are transferred to the synaptic weight in the SNN to complete the construction of the network model and realize the simulation of image classification. Finally, according to the relationship between the threshold and the classification results, a threshold-setting algorithm based on feedback is proposed, which can improve the classification accuracy of the pulse neural network.

In future work, the relationship between the threshold value of neurons and the characteristic graph of the network in SNN is deeply analyzed. The threshold value determines whether the neuron is activated to achieve feature selection. Therefore, the research on the relationship between the threshold value and feature value can provide more guidance that is effective for the threshold setting.

Acknowledgment

The National Natural Science Foundation of China under Grant No. 61672085 supports the work described in this paper.

References

[1] Cireșan D C, Meier U, Gambardella L M, et al. Deep, big, simple neural nets for handwritten digit recognition[J]. Neural computation, 2010, 22(12): 3207-3220.

[2] Farabet C, Couprie C, Najman L, et al. Learning hierarchical features for scene labeling[J]. IEEE transactions on pattern analysis and machine intelligence, 2013, 35(8): 1915-1929.

[3] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in neural information processing systems. 2012: 1097-1105.

[4] Morri A L, Gomez F, Jimenez F Z. A spiking neural network for real-time spanish vowel phonemes recognition [J]. Neurocomputing, 2017, 226: 249-261.

[5] Silva M, Vellasco M, Cataldo E. Evolving spiking neural networks for recogition of aged voices [J]. J of Voice,2017, 31(1): 24-33.

[6] Dominguez M J, Jimenez A, Rois N A, et al. Multilayer spiking neural network for audio samples classification using SpiNNaker[C]. The 25th Int Conf on Artificial Neural Networks. Barcelona: IEEE, 2016: 45-53.

[7] Cerezuela E, Jimenez A, Paz R, et al. Sound recognition systems using spiking and MLP neural networks[C]. The 25th Int Conf on Artificial Neural Networks. Barcelona:IEEE, 2016: 363-371.

[8] Stromatias E, Soto M, Serrano G. An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data[J]. Frontiers in Neuroscience,2017, 11(28): 350.

[9] Sun Q, Wu Q, Wang X, et al. A spiking neural network for extraction of features in colour opponent visual pathways and FPGA implementation [J]. Neucocomputing, 2017,228: 119-132.

[10] Ltaief M, Bezine H, Alimi A M. Training a spiking neural network to generate online handwriting movements[C].The 16th Int Conf on Intelligent Systems Design and Applications. Porto: ISEP, 2016: 289-298.

[11] Matsubara T, Torikai H. An asynchronous recurrent network of cellular automation-based neurons and its production of spiking neural network activities [J]. IEEE Trans on Neural Networks and Learning Systems, 2016,27(4): 836-852.

[12] Knudsen E I. Supervised learning in the brain[J]. J of Neuroscience, 1994, 14(7): 3985-3997.

[13] Bohte S M. The evidence for neural information processing with precise spike-times: A survey[J]. Natural Computing, 2004, 3(2): 195-206.

[14] Filp Ponulak and Andrzej. Introduction to spiking neural networks: Information processing, learning and applications [J]. Polish Neuroscience Society, 2011, 71: 409-433.

[15] Maass W. Networks of spiking neurons: The third generation of neural network models [J]. Neural Networks, 1997, 10(9):1659 -1671.

[16] Hodgkin A L, Huxley A F. A quantitative description of membrane current and its application to conduction and excitation in nerve [J]. Bulletin of Mathematical Biology, 1989, 52(l - 2):25-71.

[17] Hodgkin A L, Huxley A F. A quantitative description of membrane current and its application to conduction and excitation in nerve[J]． J Physiol，1952，117(4) : 500-544.

[18] Gersterner w, Kistler W. Spiking neuron models [M]． Cambridge University Press，2002.

[19] Izhikevich E M.Which model to use for cortical spiking neurons?" Neural Networks [J]． IEEE Trans Neural Netw，2004，15(5) : 1063-1070.

[20] Cao Y , Grossberg S , Markowitz J . How does the brain rapidly learn and reorganize view-invariant and position-invariant object representations in the inferotemporal cortex?[J]. neural networks the official journal of the international neural network society, 2011, 24(10):1050-1061.

[21] Folowosele, F., Vogelstein,R.J.,&Etienne-Cummings, R.Towards a cortical prosthesis: implementing a spike-based HMAX model of visual object recognition in silico. IEEE Journal on Emerging and Selected Topics in Circuits and Systems.516-525.

[22] Masquelier,T.,&Thorpe,S.J. Unsupervised learning of visual features through spike timing dependent plasticity. PLoS Computational Biology,3,0247-0257.

[23] Perez-Carrasco,J.A.,Serrano,C.,Acha,B.,Serrano-Gotaeerdona,T.,& Linares-Barranco,B.Spike –based convolutional network for real-time processing. In 2010 International Conference on Pattern Recognition (pp:3085-3088).

[24] Xing F , Yuan Y , Huo H , et al. Homeostasis-Based CNN-to-SNN Conversion of Inception and Residual Architectures[M]. Neural Information Processing, 2019, 173-184.