

CIRM-SNN: Certainty Interval Reset Mechanism Spiking Neuron for Enabling High Accuracy Spiking Neural Network

Li-Ye Niu

Northeastern University

Ying Wei (✉ weiyiing@ise.neu.edu.cn)

Northeastern University

Research Article

Keywords: Spiking neural network, continuous normalization, certainty interval reset, firing rate, modulation factor

Posted Date: July 22nd, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1860157/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

CIRM-SNN: Certainty Interval Reset Mechanism Spiking Neuron for Enabling High Accuracy Spiking Neural Network

Li-Ye Niu^a, Ying Wei^{a, b, c *}

^a College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

^b Information Technology R&D Innovation Center of Peking University, Shaoxing, China

^c Changsha Hisense Intelligent System Research Institute Co., Ltd.

*Corresponding Author Email: weiyi@ise.neu.edu.cn

Abstract: Spiking neural network (SNN) based on a sparse trigger and event-driven information processing has the advantages of ultra-low power consumption and hardware friendliness. As a new generation of the neural network, SNN is widely concerned. At present, the most effective way to realize deep SNN is through artificial neural network (ANN) conversion. Compared with the original ANN, the converted SNN will suffer performance loss. This paper adjusts the spike firing rate of spiking neurons to minimize the performance loss of SNN in the conversion process. We map the ANN weight to the corresponding SNN after continuous normalization, which ensures that the spike firing rate of neurons is in the normal range. We propose a certainty interval reset mechanism (CIRM), which effectively reduces the loss of membrane potential and avoids the problem of neuronal over-activation. In the experiment, we added a modulation factor (MF) to the CIRM to further adjust the spike firing rate of neurons. The accuracy of the converted SNN on CIFAR-10 is 1.026% higher than that of the original ANN. The algorithm not only realizes the lossless conversion of ANN but also reduces the network energy consumption. Our algorithm also effectively improves the accuracy of SNN (VGG-15) on CIFAR-100 and reduces the network delay. The work of this paper is of great significance for developing high-precision depth SNN.

Keywords: Spiking neural network; continuous normalization; certainty interval reset; firing rate; modulation factor.

1. Introduction

Image processing plays an important role in military, astrophysics, and aerospace. High-resolution images contain a lot of information, and processing such images requires a powerful computing platform [1]. Due to the limitation of hardware, developing a hardware-friendly and low-power neuromorphic chip is an effective solution to this problem [2]. SNN is different from the current ANN and closer to the brain information processing mode [3]. As a new generation of ANN, SNN has a more powerful information processing ability than the traditional ANN based on frequency coding information [4]. SNN uses a discrete spike train to transfer information instead of analog quantity, which is more suitable for hardware

implementation and rapid information processing [5]. Therefore, image processing instruments and systems based on SNN neuromorphological chips will show excellent performance in complex task processing.

The research of SNN is the basis of neuromorphological chip design. SNN-based neuromorphological chip needs a high-performance SNN algorithm as a guarantee [6]. This paper mainly focuses on the improvement of the performance of the SNN algorithm. The output of ANN neurons is the analog quantity in the interval, while the output and input of neurons in the biological nervous system are discrete spike information [7]. Scientific research shows that the information coding and processing mechanism based on spike train is very important for the study of brain-like intelligence [8]. Therefore, SNN composed of spiking neurons based on biological interpretability has become an indispensable tool for the study of brain-like systems [9].

ANN only processes the image in the spatial dimension, while SNN uses a spike train to represent and process all kinds of information [10]. It integrates multiple dimensions of input information, such as time, space, frequency, and phase. Because SNN processes discrete spike information, it can not train the network through back propagation (BP) like ANN [11]. Although SNN has advantages in processing perceptual information, the training of its network model has not been unified [12].

In the past few years, SNN has developed rapidly, and its model training is still an active research field. At present, the algorithms used for SNN training are mainly divided into direct training and indirect training. Direct training includes local learning rules based on spike time-dependent plasticity (STDP) and spike-based approximate error BP [13]. The SNN based on STDP rules is limited to shallow networks (generally no more than 5 layers), and its accuracy is much lower than ANN in complex tasks [14]. The internal training of SNN based on the spike BP algorithm is very complex, which limits its application in deep SNN. The indirect training method is to convert the ANN into the corresponding SNN [15]. This method is an effective way to realize the deep SNN. Hunsberger and Eliasmith used the biologically more reliable LIF neuron model instead of the previous IF to convert SNN [6, 15]. They found that the high discharge rate of neurons can produce performance equivalent to that of the original network, while the low discharge rate will cause a great loss of performance [6]. In 2015, Cao et al. Successfully converted a deep SNN with two orders of magnitude lower power consumption than the original network by training and tailoring CNN [15]. Compared with the original network, the accuracy of the converted SNN network is reduced by 1.69% on the CIFAR-10. Han Bing et al. proposed using soft reset (It can also be called residual membrane potential (RMP)) spiking neurons, appropriate hierarchical threshold initialization, and constrained ANN training to achieve almost lossless ANN-SNN conversion [16]. The conversion from ANN to SNN will cause the loss of network performance. The focus of researchers' work is how to reduce the loss in the conversion process [17].

ANN-SNN conversion has been proved to be an effective way to realize deep SNN. The transformed

SNN can obtain high enough accuracy in complex tasks [18]. Based on this, we propose a weight continuous normalization (WCN) algorithm for the network to ensure the normal spike emission of neurons. The certainty interval reset mechanism (CIRM) proposed in this paper solves the problems existing in hard reset and soft reset mechanisms. By adding a modulation factor (MF) to the CIRM, the spike firing rate of neurons is further adjusted to ensure the performance of the network. For different tasks, the MF only needs to be adjusted properly to realize the lossless conversion of the network. By analyzing the spike firing rate of neurons, the advantages of our algorithm are explained.

The specific contributions of our work are as follows:

1. We monitor and analyze the spiking firing rate of the model and propose a modulation method to adjust it to improve the network performance.
2. We propose the WCN algorithm of ANN to ensure that the spike firing rate of spiking neurons is in the normal range.
3. Aiming at the deficiency of hard reset and soft reset of spiking neurons, the CIRM proposed by us effectively reduces the loss of membrane potential of hard reset and avoids the problem of over activation of soft reset neurons.
4. We added the MF to the certainty interval reset mechanism to further adjust the spike firing rate of neurons. The algorithm not only reduces the energy consumption of the model but also improves the network delay.
5. In the experiment, the algorithm in this paper realizes the lossless conversion of SNN. The classification accuracy of the network on MNIST and CIFAR-10 is higher than that of the original ANN. The algorithm also has good applicability in the deep network (VGG-15) and CIFAR-100.

The first part introduces the background and development of SNN. The second part of the article introduces the relevant theories of ANN-SNN. On this basis, the optimization algorithm of the SNN model is proposed. Then, experiments are carried out on our proposed algorithm. Finally, the algorithm is discussed and analyzed in detail.

2. ANN to SNN

The conversion of ANN to SNN requires a series of conversion operations, mainly including neuron replacement, weight normalization, threshold allocation, and selection of appropriate reset mechanisms [19]. The purpose of these conversion operations is to improve the performance of SNN and reduce the performance loss in the process of network conversion.

2.1 Neuron Model

The spiking neuron is the basic unit of SNN. Its main function is to integrate and transmit the spike train information of the network. The integrated and fire (IF) model has been widely used in the field of neural

computing for a long time [20]. The reason why the IF neuron is selected is that it has a simple linear model. The analytical expression of the membrane potential of the IF model can not only quantitatively study the properties of neurons but also accurately simulate SNN by using an event-driven method [21]. The membrane potential of IF neuron at time t is calculated by equation (1).

$$V_m(t) = V_m(t-1) + \sum_j W_{j,i} S_j(t) \quad , \quad (1)$$

where $V_m(t-1)$ represents the membrane potential of neuron at time $t - 1$. $W_{j,i}$ represents the weight of the connection between the current layer neuron i and the upper layer neuron j . S_j represents the spike train information of neuron j (If the neuron has a spike at time t , $S_j(t) = 1$, otherwise, $S_j(t) = 0$). The membrane potential of neuron at time t is the sum of the membrane potential at time $t - 1$ and the spike input received at time t . When the membrane potential at time t exceeds the threshold V_{thr} , the neuron will generate a spike and the neuron membrane potential will be reset [22].

The training of the ANN converted to SNN uses ReLU as the activation function, and the nonlinear description of the function is as equation (2).

$$y = \max(0, \sum_j W_{ij} x_j + b) \quad , \quad (2)$$

where y is the output of the activation function ReLU. W_{ij} is the weight of the connection between the current layer neuron i and the upper layer neuron j . x_j is the activation value of neuron j . b represents the bias term of neurons[23]. The activation function ReLU varies linearly with positive input, which can be roughly simulated using IF neurons. Therefore, when ANN is converted to SNN, the neuron nodes in ANN are replaced by IF neurons. To make the network conversion more effective, the b is usually set to 0.

2.2 Reset Mechanism

When the neuron membrane potential exceeds its threshold, the neuron will emit spikes to trigger the corresponding reset mechanism. Next, we introduce the most commonly used hard reset and soft reset. On this basis, we propose a certainty interval reset mechanism.

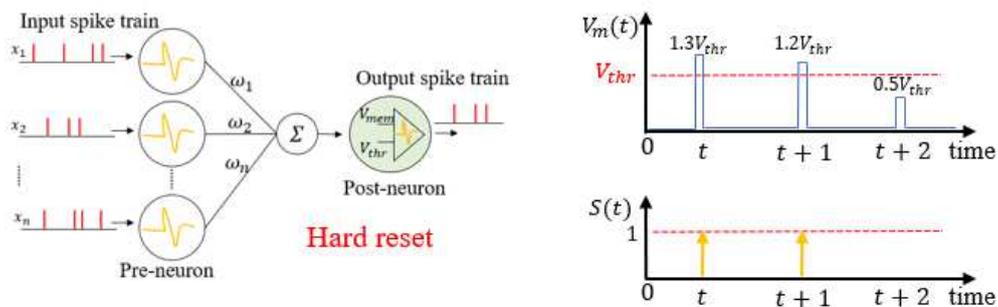


Fig. 1. Hard reset mechanism of neuronal membrane potential.

2.2.1 Hard Reset

At present, the hard reset is a common method in the research of SNN. At one spike instant, the membrane potential is "hard reset" to 0, regardless of how much the membrane potential exceeds the threshold V_{thr} . If

the membrane potential does not exceed V_{thr} , then its value remains unchanged [24]. Equation (3) shows the specific way of the neuron hard reset method.

$$V_m'(t) = \begin{cases} V_m(t) & V_m(t) < V_{thr} \\ 0, S(t) = 1 & V_m(t) \geq V_{thr} \end{cases} \quad (3)$$

Ignoring the residual membrane potential above V_{thr} will affect the expected linear relationship between input and output. We assume that the weighted input sum received by an IF neuron (as shown in Fig. 1) in three consecutive time steps is $1.3V_{thr}$, $1.2V_{thr}$, and $0.5V_{thr}$, respectively. The weighted input sum of the three consecutive time steps is $3V_{thr}$. Neurons need to generate three spikes to maintain a precise linear relationship between input and output. Because hard reset ignores the residual potential above the threshold at the trigger moment, the neuron produces only two spikes in three consecutive time steps. The hard reset will cause the loss of neuronal membrane potential.

2.2.2 Soft Reset

The soft reset mechanism of spiking neurons effectively solves the problem of membrane potential loss of the hard reset mechanism [16]. Equation (4) shows the reset mode of the soft reset mechanism.

$$V_m'(t) = \begin{cases} V_m(t) & V_m(t) < V_{thr} \\ V_m - V_{thr}, S(t) = 1 & V_m(t) \geq V_{thr} \end{cases} \quad (4)$$

At time t , the neuronal membrane potential $V_m(t) < V_{thr}$, the membrane potential remained unchanged, $V_m'(t) = V_m(t)$; $V_m(t) \geq V_{thr}$, the membrane potential was reset, $V_m'(t) = V_m - V_{thr}$ and $S(t) = 1$. Similarly, we assume that the sum of weighted inputs received by an IF neuron (as shown in Fig. 2) in three consecutive time steps is $1.3V_{thr}$, $1.2V_{thr}$, and $0.5V_{thr}$, respectively. The neuron soft reset method retains the membrane potential beyond the threshold, and the accumulation of membrane potential generates a spike in three consecutive time steps (as shown in Fig. 2(b)). If the neuron membrane potential $V_m(t) \geq 2V_{thr}$ at time t , the reset neuron membrane potential $V_m'(t)$ is still greater than the threshold V_{thr} . When neurons do not receive any input at $t + 1$ time, they will also generate a spike, resulting in the problem of over-activation of neurons (as shown in Fig. 2(c)).

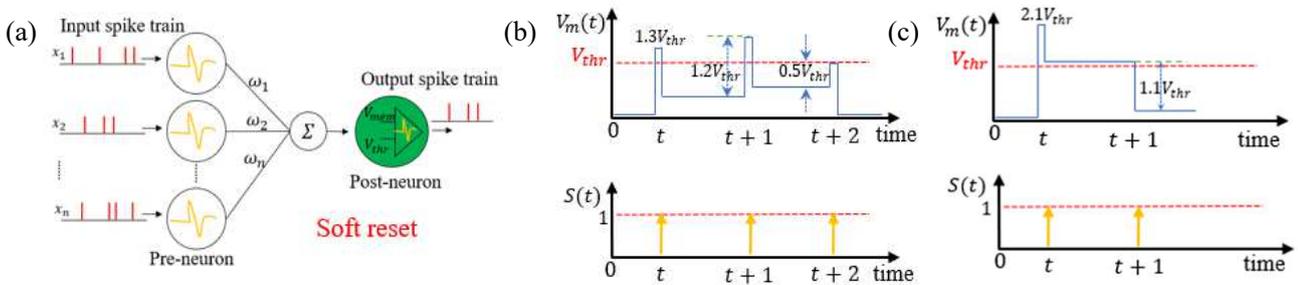


Fig. 2. Soft reset mechanism of neuronal membrane potential.

2.2.3 Certainty Interval Reset

To solve the problems of membrane potential loss caused by hard reset and neuron over-activation caused by soft reset, we propose a CIRM, which effectively solves the problems caused by hard reset and soft reset.

The membrane potential reset mode of CIRM is shown in equation (5).

$$V_m'(t) = \begin{cases} V_m(t) & V_m(t) < V_{thr} \\ V_m(t) - V_{thr}, S(t) = 1 & V_{thr} \leq V_m(t) < 2V_{thr} \\ 0.99V_{thr}, S(t) = 1 & V_m(t) \geq 2V_{thr} \end{cases} \quad (5)$$

Assuming that the membrane potential of neurons at time t is V_{thr} , the adjustment of membrane potential under the CIRM will be as follows:

1. $V_m(t) < V_{thr}$, the neuronal membrane potential remained unchanged, $V_m'(t) = V_m(t)$.
2. $V_{thr} \leq V_m(t) < 2V_{thr}$, the neuronal membrane potential was reset, $V_m'(t) = V_m(t) - V_{thr}$ and $S(t) = 1$. In this case, CIRM solves the problem of membrane potential loss caused by hard reset.

$V_m(t) \geq 2V_{thr}$, the membrane potential was reset, $V_m'(t) = 0.99V_{thr}$ and $S(t) = 1$. At time t , the membrane potential exceeded the threshold, and $V_m'(t)$ was reset to $0.99V_{thr}$. The membrane potential was reset to $0.99V_{thr}$, which effectively reduced the loss of membrane potential caused by hard reset. When the neuron does not receive any input at time $t + 1$, the neuron membrane potential $V_m(t + 1) = 0.99V_{thr} < V_{thr}$, the neuron does not emit a spike. Assuming that the weighted input sum of the neuron at time t is $2.1V_{thr}$, the neuron emits a spike and resets the membrane potential to $0.99V_{thr}$. At time $t + 1$, if the neuron does not receive any input, the neuron will not generate a spike, as shown in Fig. 3.

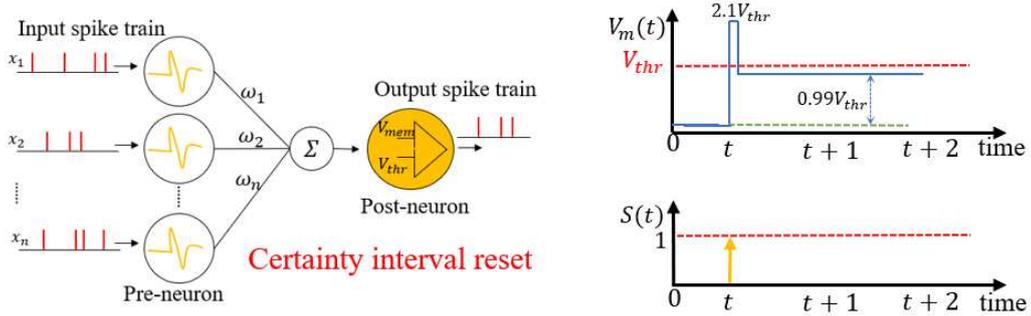


Fig. 3. CIRM of neuronal membrane potential.

2.3 Pooling Operation

The pooling layer is generally behind the convolution layer of the convolution neural network to reduce the size of the convolution output map. Maximum pooling and average pooling are the two most popular methods to implement the pooling mechanism. SNN deals with spike train information rather than analog values. If the maximum is performed in the network, it will lead to a serious loss of information. Therefore, our pre-trained network considers average pooling as a pooling mechanism [25].

2.4 Normalized Weight

ANN saves the weight of its network after full training. If the ANN weights are directly mapped to the corresponding SNN without any operation, it will cause a great loss of network performance [26]. The first possible cause of network performance loss is that the weight is too high, and its size exceeds the threshold of spiking neurons, resulting in multiple spikes in one timestep. The second possible reason is that the neuron

does not have enough input or the neuron threshold is too high so that the input cannot exceed its threshold, resulting in the spike firing rate lower than its normal rate. For the conversion loss, we propose a WCN algorithm to minimize the loss to improve the network performance. The pseudo-code of the WCN method is shown in **Algorithm 1**. This method adjusts the weight twice continuously according to the maximum activation value and weight value of neurons to avoid the loss of network performance caused by too high weight. After the weight is adjusted by WCN, we set the threshold of all spiking neurons in the network to 1 to avoid the abnormal spike emission caused by too high a threshold. This normalization method is suitable for the case of short delay and high precision. The results show that this method is very effective to improve network performance.

Algorithm 1: Weight continuous normalization

```

1 for layer in net.layers:
2   max_activation = 0
3   max_weight = 0
4   normalization_factor = 0
5   m = 0
6   for m in layers:
7     for neuron in layers.neurons:
8       weight = max( Wij )
9     end
10    max_weight = max(weight)
11    max_activation = max(neurons_activation)
12    normalization_factor = max (max_weight, max_activation)
13    for neuron in layers.neurons:
14      Wij = Wij/normalization_factor
15    end
16  end
17 end

```

2.5 Input Encoding for SNN

We use Poisson coding to encode the input image of the network. A Poisson event generation process is used to generate the input spike train [27]. Each time step of the SNN operation is associated with the generation of a random number whose value is compared with the corresponding input amplitude (image pixel value). If the generated random number is less than the value of the corresponding pixel intensity, a spike is triggered. The average number of spikes transmitted to the network as an input in a sufficiently large time window is approximately proportional to the size (pixel intensity) of the original ANN input. The time window size of Poisson coding is very important for the coding process of neural information. The time window cannot be selected arbitrarily but depends on the dynamic characteristics of the stimulus signal.

We randomly select 8 samples in CIFAR-10 (as shown in Fig. 4 (a)) to compare the impact of different time windows on image coding. In the experiment, different time windows (Blue dashed box, window sizes: 64, 128, 256, 374, 512, 768, and 1024) are selected to encode the selected samples respectively. The coding frequency of all our experiments is fixed at 1000Hz. Fig. 4 (b) shows the coding of the sample at a certain time (black dashed box), and Fig. 4 (c) shows the image reconstructed after the Poisson coding of the sample

(red dashed box). By comparing the reconstructed coded images, we can know that the small Poisson coding time window will lead to the large loss of reconstructed image information and the performance loss of SNN. On the contrary, a large coding time window will enrich the image information and will not cause serious loss of image information. However, the larger the time window, the amount of calculation of the corresponding network will increase significantly, and the calculation delay of the network will also be prolonged. The algorithm we designed should not only reduce the amount of computation but also shorten the network delay under the condition of ensuring network performance.

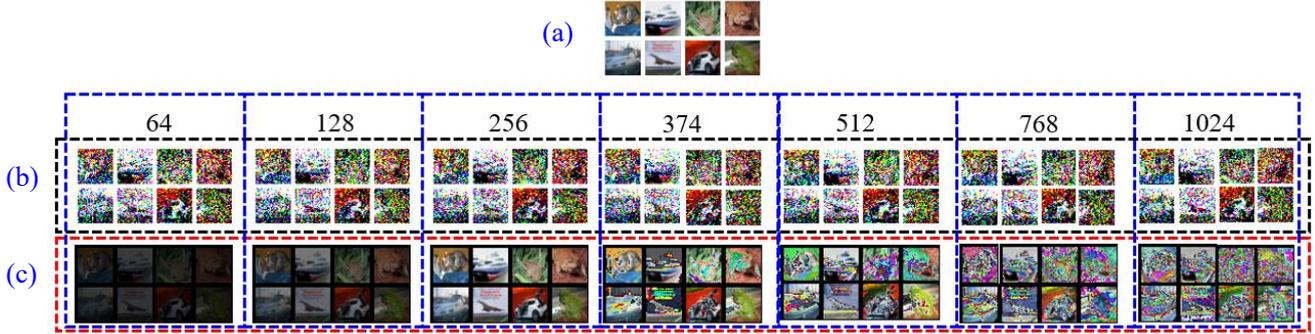


Fig. 4. Poisson coding of image. Different time windows (Blue dashed box) are selected to encode the selected samples respectively. (a) Original image. (b) the coding of the sample at a certain time (black dashed box). (c) the image reconstructed after Poisson coding of the sample (red dashed box).

3. Spike Firing Rate Modulation

The firing rate of spiking neurons is an important factor affecting the performance of SNN. In this section, we analyze the spike firing rate of neurons from the perspective of reducing the conversion error. On this basis, we give the mathematical expression of the spike firing rate and the theoretical explanation of the conversion error.

3.1 Analysis of Neuronal Membrane Potential

The too high or too low spike output rate of neurons will affect the performance of SNN. An appropriate spike output rate is very important to improve the performance of the network. [28] adjust the spike output rate of the network by changing the size of the neuron threshold. There are two problems with this adjustment method:

1. The increase of the threshold value of spiking neurons will reduce the spike firing rate. In this case, it will be more difficult for inactive neurons to emit spikes.
2. Decreasing the threshold can increase the firing rate of spiking neurons. However, active spiking neurons may be over-activated.

We avoid the above problems by adding modulation factor η to the CIRM to adjust the neuron spike firing rate. The specific adjustment method of η to spike firing rate is shown in equation (6).

$$V_m'(t) = \begin{cases} V_m(t) & V_m(t) < V_{thr} \\ V_m(t) - \eta * V_{thr}, & S(t) = 1 \quad V_m(t) \geq V_{thr} \end{cases} \quad (6)$$

Assuming that the membrane potential of neurons at time t is V_{thr} , the adjustment of membrane potential under the MF will be as follows:

1. $V_m(t) < V_{thr}$, the neuronal membrane potential remained unchanged, $V_m'(t) = V_m(t)$.
2. $V_m(t) \geq V_{thr}$, the neuronal membrane potential was reset, $V_m(t) - \eta * V_{thr}$ and $S(t) = 1$.

According to different modulation factors, the reset membrane potential $V_m'(t)$ is divided into three cases, as shown in equation (7).

$$V_m'(t) = \begin{cases} 0 & V_m'(t) < 0 \\ V_m(t) - \eta * V_{thr} & 0 \leq V_m'(t) < V_{thr} \\ 0.99V_{thr} & V_m'(t) \geq V_{thr} \end{cases} . \quad (7)$$

After MF was added, the reset neuronal membrane potential was $V_m'(t)$, The value of MF affects the membrane potential reset, including the following three cases:

1. $\eta > 1$ may cause $V_m'(t)$ to be less than 0. If $V_m'(t) < 0$, to avoid the loss of membrane potential, we reset $V_m'(t)$ to 0.
2. If $0 \leq V_m'(t) < V_{thr}$, $V_m'(t)$ remains unchanged.
3. If $V_m'(t) \geq V_{thr}$, to avoid over activation of neurons at $t + 1$ time, reset $V_m'(t)$ to $0.99V_{thr}$.

Choosing a larger MF can reduce the spike firing rate of easily activated neurons, which has little effect on the spike firing rate of difficult activated neurons. Smaller MF can increase the spike firing rate of difficult to activate neurons, and it does not affect the spike firing rate of easily activated neurons. The addition of η can flexibly adjust the spike firing rate of neurons and avoid the problem caused by changing the threshold value of neurons.

3.2 Error Analysis

There is an approximately proportional relationship between the firing rate and the activation value of neurons. According to the relationship between the spike firing rate $r_i^1(t)$ and the activation value a_i^1 of the neurons in the first layer of the network under hard reset and soft reset [29], we derive equation (8). The equation represents the relationship between the spike firing rate $r_i^1(t)$ and the activation value a_i^1 of the CIRM.

$$r_i^1(t) = a_i^1 r_{max} - \frac{V_m^1(t)}{t \times V_{thr}} , \quad (8)$$

where r_{max} represents the maximum spike firing rate, a_i^1 represents the activation value of ANN neurons, $V_m^1(t)$ represents the membrane potential of neuron i in the first layer after reset. The relationship between the spike firing rate $r_i^1(t)$ and the activation value a_i^1 is not strictly proportional, but there is an error term $\frac{V_m^1(t)}{t \times V_{thr}}$. When we add an MF (η) to the membrane potential $V_m^1(t)$, the relationship between $r_i^1(t)$ and a_i^1 is shown in equation (9).

$$r_i^1(t) = a_i^1 r_{max} - \frac{V_m^1(t)}{t \times V_{thr}} = \begin{cases} a_i^1 r_{max} & V_m^1(t) < 0 \\ a_i^1 r_{max} - \frac{V_m^1(t) - \eta * V_{thr}}{t \times V_{thr}} & 0 \leq V_m^1(t) < V_{thr} \\ a_i^1 r_{max} - \frac{0.99 V_{thr}}{t \times V_{thr}} & V_m^1(t) \geq V_{thr} \end{cases} \quad (9)$$

where $V_m^1(t)$ represents the membrane potential of neurons after MF modulation. After adding η , the error term $\frac{V_m^1(t)}{t \times V_{thr}}$ becomes $\frac{V_m^1(t)}{t \times V_{thr}}$. According to the size of $V_m^1(t)$, there are three cases. In these three cases, $V_m^1(t) < V_m(t)$, so $\frac{V_m^1(t)}{t \times V_{thr}} < \frac{V_m(t)}{t \times V_{thr}}$. In either case, the error term is reduced. Therefore, the MF (η) can effectively reduce the error in the conversion model.

4. Experiment

In the previous chapters, we theoretically derived and analyzed the algorithm. In this section, the algorithm is tested experimentally.

4.1 Network structure

Here, we use a network similar to Fig. 5 to test our algorithm with MNIST and CIFAR-10 data sets. The MNIST dataset trains 100 epochs in the network, the batch size is 256, and the learning rate is 1. The CIFAR-10 data set is trained in the network for 240 epochs, and the batch size is also 256. The initial learning rate of the network is 0.1. The learning rates of epoch 81 and epoch 142 are divided by 10 respectively. All experiments are run on a GPU. The regularized dropout parameter is uniformly set to 0.5, and the momentum attenuation and momentum are 0.0001 and 0.9 respectively. The size of all convolution kernels in the network is 3×3 . All the experimental results in this paper are the average of the results of five independent runs of the network. We trained the network according to the above method. The classification accuracy of the network is 99.39% on MNIST and 86.00% on CIFAR-10.

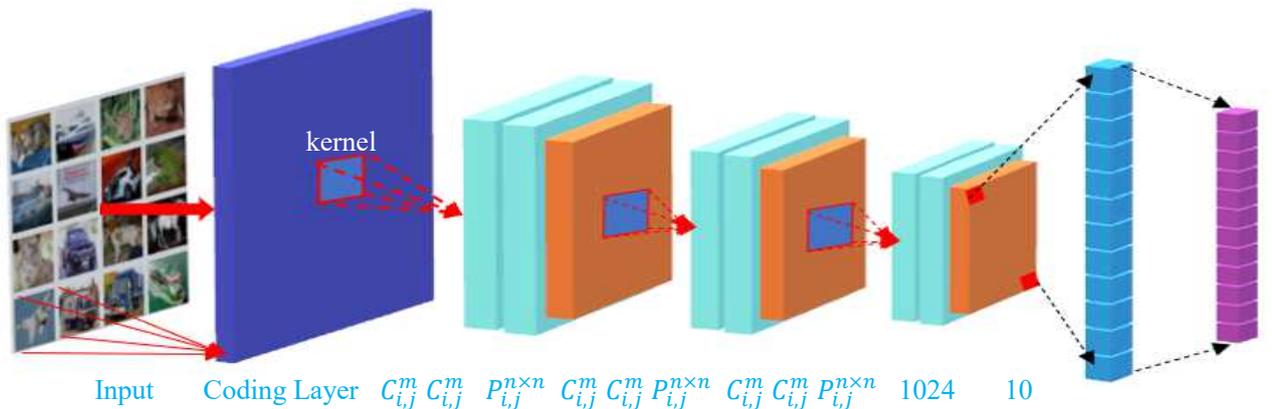


Fig. 5. Network structure. The network includes one input coding layer, six convolution layers, three pooling layers and two full connection layers. $C_{i,j}^m$ indicates that there are m neurons in the i th convolution layer, and the step size is j .

$P_{i,j}^{n \times n}$ indicates that the step size of the i th pool layer is j and the kernel size is $n \times n$.

4.2 Result

Next, the above networks are converted into corresponding SNNs. We first save the weight of the network

and prepare the conversion operation of the network. Then, the above-saved weights are continuously normalized. Replace all nodes of the network with IF neurons (except the output layer). A coding layer is added at the network input to Poisson code the input image. In this way, the ANN network is successfully converted to SNN. We test its performance on the data set.

4.2.1 MNIST

Based on different coding time windows (time window sizes: 32, 64, 128, 256, 374, and 512), we tested the classification performance of SNN on the MNIST dataset under hard reset, soft reset, and CIRM respectively (the results are shown in Fig. 6). In our experiment, when the coding time window of the network is 32, the accuracy of our proposed CIRM is 99.1%, and the accuracy of hard reset and soft reset methods is only 64.89% and 98.79%. When the coding time window is 512, SNN uses hard reset, soft reset, and CIRM to achieve the best performance, and the accuracy rates are 99.38%, 99.27%, and 99.36% respectively.

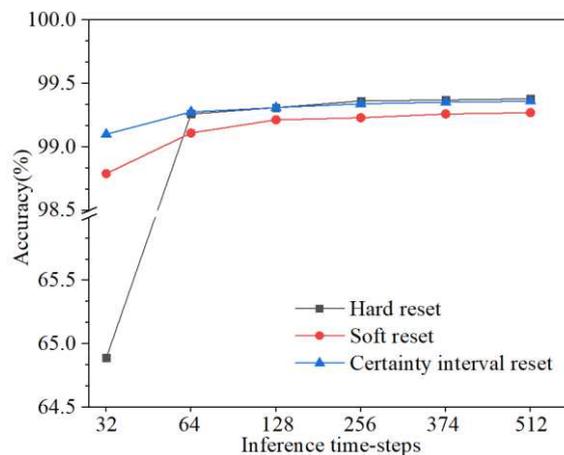


Fig. 6. The classification performance of SNN on MNIST under hard reset, soft reset and CIRM respectively.

Under the above three reset mechanisms, the classification accuracy of SNN increases gradually with the increase of the coding time window. However, the increase in the coding time window will greatly increase the amount of calculation and delay of the network. By comparing the experimental results, we find that SNN has good configurability. If the network computing time is very important, we can choose the interval reset mechanism. This mechanism can make the network performance reach the same level as ANN with a small time window. If accuracy is important, a large time window can be used to improve the performance of the network.

The CIRM proposed by us shows good performance in the above experiments. Next, we add modulation factor η to the CIRM to adjust the spike firing rate of neurons, to further improve the network performance. The impact of η of different sizes on network performance is shown in Fig. 7. When the time window size is 512 and the modulation factor $\eta = 1.5$, the performance of the SNN network reaches the best classification accuracy of 99.48%. In this case, the accuracy of SNN is improved by 0.09% compared with the original ANN, which exceeds the performance of the ANN network. When the time window is 64, the SNN network performance with $\eta = 1.5$ has exceeded the accuracy of the original ANN. Choosing the appropriate η can not

only improve the performance of the network but also improve the delay of the network.

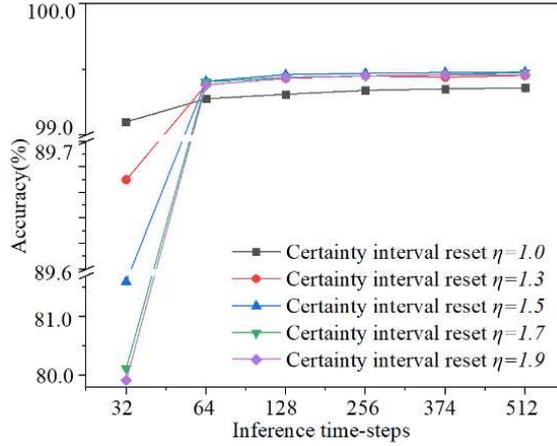


Fig. 7. The impact of η of different sizes on network performance.

Table 1 compares the best performance of SNN under different reset mechanisms and also compares it with previous work. By comparison, we can know that the addition of MF in the CIRM effectively improves the performance of SNN.

Table 1. Network performance comparison (MNIST).

Model	Reset	Accuracy (%)	Ref.
SNN(BP)	/	98.71	[30]
SNN(STDP)	Hard Reset	95.00	[31]
SNN(BP)	/	97.66	[32]
Spiking RBM(STDP)	Hard Reset	93.16	[33]
SNN(IF)	Hard Reset	98.48	[25]
SNN(Data-Norm)	Hard Reset	98.64	[25]
SNN(Model-Norm)	Hard Reset	98.61	[25]
SNN(WCN)	Hard Reset	99.38	This work
SNN(WCN)	Soft Reset	99.27	This work
SNN(WCN)	CIRM	99.36	This work
SNN(WCN)	CIRM(MF=1.5)	99.48	This work

4.2.2 CIFAR-10

With the above experiments, we tested the classification performance of SNN on CIFAR-10 under hard reset, soft reset, and CIRM respectively (time window size: 64, 128, 256, 374, 512, 768, 1024, and 2048). The results are shown in Fig. 8. Under the three reset mechanisms, the classification accuracy of SNN increases gradually with the increase of the coding time window. When the coding time window size is 1024, SNN uses hard reset, soft reset, and CIRM, and its network accuracy is 85.88%, 85.18%, and 85.70% respectively. When the coding time window size is 2048, SNN uses hard reset, soft reset, and CIRM, and its network achieves the best performance, with accuracy rates of 86.49%, 86.63%, and 86.13% respectively. Compared with ANN, the accuracy of the three reset mechanisms are better than ANN, which shows that the weight normalization method proposed by us is effective in network conversion and has good applicability.

Next, we add η to the interval reset mechanism to adjust the spike firing rate of neurons. The impact of η of different sizes on network performance is shown in Fig. 9. When the input coding time window size is 2048 and $\eta = 1.5$, the classification accuracy of the SNN network reaches 87.01%. When $\eta = 1.1$, the

performance of the SNN network reaches the best, and the classification accuracy is 87.03%. The accuracy of SNN with $\eta = 1.1$ is 1.03% higher than that of the original ANN. In this paper, the optimal MF of the network in MNIST is $\eta = 1.5$, while the optimal MF of the network in CIFAR-10 is $\eta = 1.1$. The optimal modulation factors for different data sets are different. The experimental results show that we can flexibly select the best η for different tasks to improve the performance of SNN.

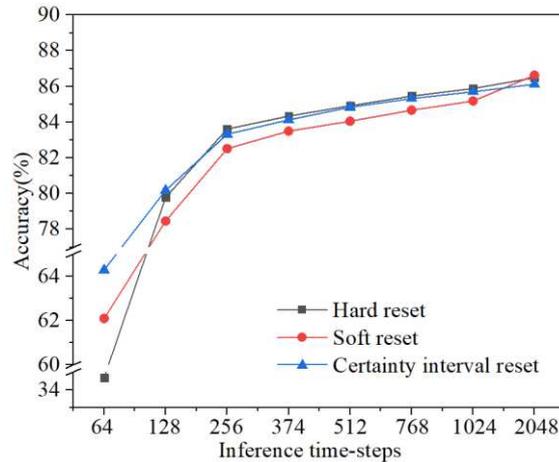


Fig. 8. The classification performance of SNN on CIFAR-10 under hard reset, soft reset and CIRM respectively.

The focus of the research on the conversion from ANN to SNN is to reduce the loss of network conversion. The WCN algorithm proposed in this paper shows excellent performance in SNN of three reset methods. The WCN algorithm adjusts the spike firing rate of neurons to an appropriate range so that the classification accuracy of the converted SNN exceeds that of the original ANN. [25] in the four-layer fully connected network, the data-based normalization algorithm has achieved good results on the MNIST data set. We compare the spike firing rate of this algorithm with the WCN algorithm proposed in this paper at each neuron layer of the network (network time window size: 256, neuron reset mode: hard reset), as shown in Fig. 10.

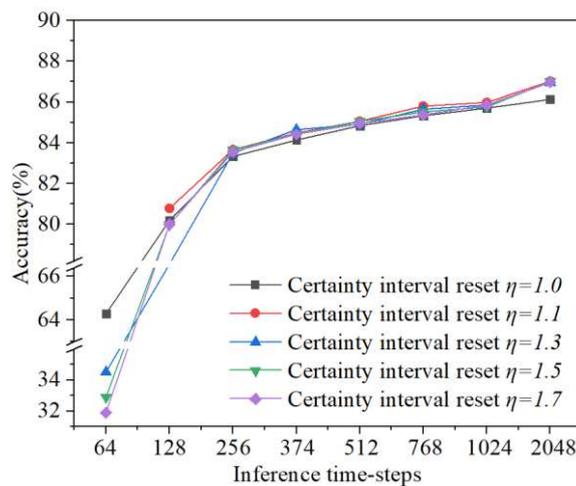


Fig. 9. The impact of η of different sizes on network performance.

As can be seen from the figure, we do not deal with the weight of ANN and directly convert the network. The neurons in the front layer of the network are over-activated, and the neurons in the subsequent layer are under-activated. Our network algorithm based on [25] caused serious loss of spikes in the neuron layer behind

the network, resulting in a great loss of network performance. Both over-activation and under-activation of neurons will cause the loss of SNN performance. The spike firing rate of neurons is the fundamental reason affecting SNN performance. The spike emission of each layer of the network of the WCN is better than the above two methods. The WCN algorithm proposed in this paper ensures that the neuron spike firing rate is in the normal range, and the performance of the converted SNN network is not reduced but improved.

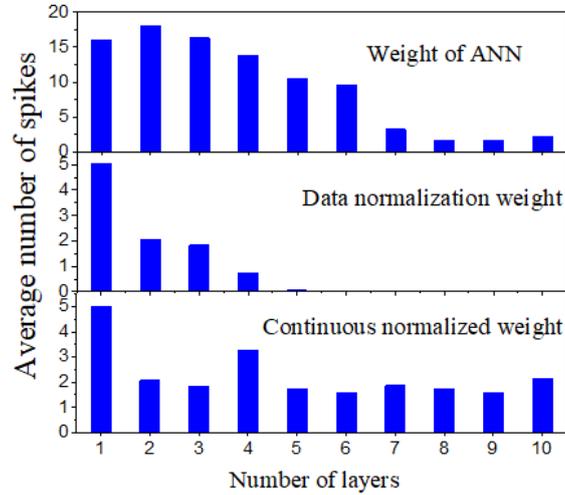


Fig. 10. The spike firing rate.

The WCN algorithm proposed by us preliminarily adjusts the spike firing rate to ensure the normal firing of neuron spikes. Aiming at the shortcomings of the hard reset and soft reset mechanism, we propose CIRM. The spike firing rate was further adjusted by adding MF to CIRM. The spike emission of neurons in each layer of the SNN network with MF added to CIRM is shown in Fig. 11.

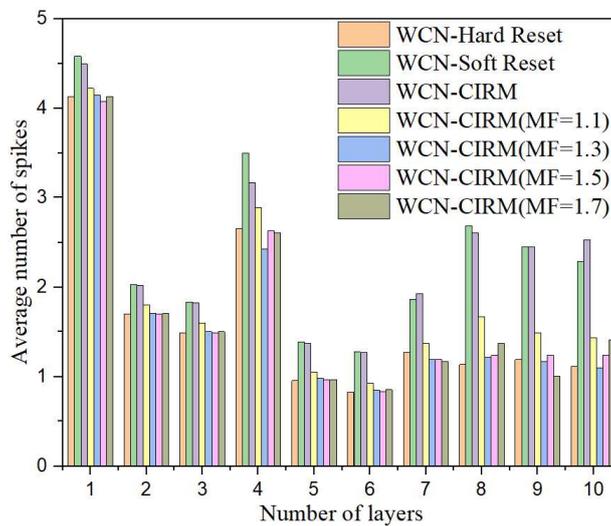


Fig. 11. The spike emission of neurons in each layer of SNN network with MF added to CIRM.

CIRM resets the membrane potential of spiking neurons to a reasonable range, which solves the problems of membrane potential loss of hard reset and over activation of soft reset neurons. MF was added to CIRM to further adjust the reset membrane potential. Choose the appropriate value of MF according to our needs. We experimentally selected $MF = 1.1$ to achieve the optimal performance of SNN (time window: 2048, CIFAR-10), and the accuracy is improved by 1.026% compared with ANN. The MF with appropriate value finally controls the spike firing rate in a better range, to improve the performance of SNN.

Table 2 compares the best performance of SNN on CIFAR-10 under different reset mechanisms and also compares it with previous work. Through comparison, we can know that the previous SNN conversion algorithms have caused the loss of network performance. The classification performance of SNN based on our algorithm is improved compared with the original network. Our algorithm has obvious advantages in the conversion of SNN networks.

Table 2. Network performance comparison (CIFAR-10).

Model	Reset	Loss (%)	Ref.
VGG-16(SPIKE-NORM)	Hard Reset	-0.15	[28]
VGG-16(ANN-SNN)	Hard Reset	-0.24	[28]
ResNet-20(IF)	Hard Reset	-1.64	[16]
ResNet-20(RMP)	Soft Reset	-0.11	[16]
VGG-16(IF)	Hard Reset	-0.15	[16]
VGG-16(RMP)	Soft Reset	-(<0.01)	[16]
BinaryNet Sign	Hard Reset	-0.72	[29]
BinaryNet Heav	Hard Reset	-0.97	[29]
BinaryConnect, binarized at infer.	Hard Reset	+0.19	[29]
BinaryConnect, full prec. at infer.	Hard Reset	-1.06	[29]
SNN(WCN)	Hard Reset	+0.492	This work
SNN(WCN)	Soft Reset	+0.627	This work
SNN(WCN)	CIRM	+0.127	This work
SNN(WCN)	CIRM(MF=1.1)	+1.026	This work

4.2.3 Average firing rate

SNN processes information only when the spike appears, resulting in energy consumption. Therefore, the event-driven characteristic of SNN improves the energy efficiency of the neural morphological network. We propose the average firing rate (R_{ASR}) to indirectly reflect the energy consumption of the SNN model, which is defined in equation (8).

$$R_{ASR} = \frac{N_{spikes}}{N_{neurons} \times time - steps} \times 100\% , \quad (8)$$

where, R_{ASR} represents the average percentage of spikes produced by spiking neurons in each time step. N_{spikes} represents the total number of spikes in the network. $N_{neurons}$ indicates the number of spiking neurons in the network. $time - steps$ indicates the length of the encoding train.

The higher the average firing rate, the higher the energy consumption of the algorithm model. According to the above definition, we calculate the average firing rate of the SNN algorithm on CIFAR-10. The calculation results are shown in Fig. 12. According to the resulting graph, it can be found that the average firing rate ($\sim 1\%$) presents a disproportionate benefit with the doubling of reasoning $time - steps$. Although the energy consumption of the hard reset model is low, the accuracy can not meet the requirements. The soft reset mechanism solves the problem of membrane potential loss of hard reset and improves the recognition accuracy of the network. However, the average firing rate increases significantly, which means that the energy consumption of the soft reset model is higher than that of the hard reset. Our CIRM reduces the loss of membrane potential caused by hard reset and reduces the energy consumption of soft reset.

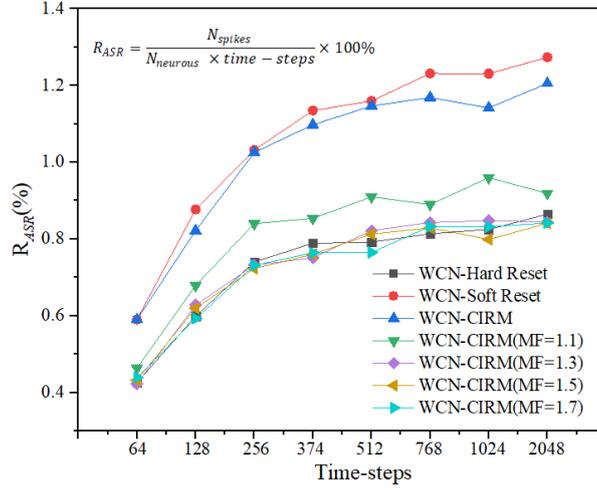


Fig. 12. Average firing rate curve of SNN.

As can be seen from Fig. 12, the average firing rate of CIRM is not much smaller than that of soft reset. While ensuring accuracy, to further reduce the energy consumption of the model, we add an MF to CIRM. The average firing rate curves of different modulation factors are drawn in Fig. 12. When MF = 1.1, the performance of SNN reaches the best (time window: 2048), and the accuracy is improved by 1.026% compared with ANN. And the energy consumption of the model is further reduced compared with CIRM. With the increase of MF value, although the average firing rate of the model gradually decreases (the corresponding energy consumption also decreases), the accuracy cannot meet the requirements. The algorithm not only reduces the energy consumption but also ensures its accuracy without loss. Adding appropriate MF into CIRM not only improves the accuracy but also reduces the energy consumption. It can be seen that our algorithm can flexibly and effectively improve the performance of SNN.

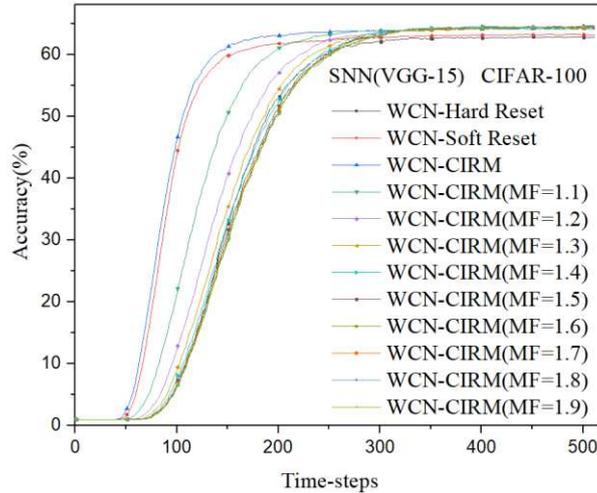


Fig. 13. The test results of SNN models based on different algorithms on CIFAR-100.

4.2 CIFAR-100

The above content makes an empirical analysis and optimization of the algorithm proposed in this paper on CIFAR-10. We apply the conclusions and settings to the simulation of CIFAR-100 by deep layer network (VGG-15). To verify the effectiveness and generalization of our algorithm, the settings of VGG-15 and training parameters used in this experiment are consistent with [34]. CIFAR-100 trains 200 epochs, the batch size is

256, the initial learning rate is 0.05, and the learning rates of epoch 81 and epoch 122 are divided by 10 respectively. All experiments are run on a GPU. The regularized dropout parameter is uniformly set to 0.1, and the momentum attenuation and momentum are 0.0001 and 0.9 respectively. The classification accuracy of VGG-15 on the CIFAR-100 test set is 64.9%.

Table 3. SNN (VGG-15) performance comparison (CIFAR-100).

Model (Algorithm)	Reset	Accuracy/Loss (%)	Time-steps(about)
VGG-15 [25]	Hard Reset	62.69/2.21	362
VGG-15 [34]	Soft Reset	63.16/1.74	402
VGG-15 [34]	B-Soft Reset	62.07/2.83	345
VGG-15(WCN) (This work)	CIRM	64.13/0.77	340
VGG-15(WCN) (This work)	CIRM(MF=1.5)	64.63/0.27	334

Next, we convert the trained VGG-15 to the corresponding SNN and test its performance on CIFAR-100. The test results of SNN models based on different algorithms on CIFAR-100 are shown in Fig. 13. From the test curve, we get the performance comparison table of different algorithms (Table 3). Compared with the hard reset and soft reset methods, the SNN (VGG-15) model based on WCN and CIRM algorithm not only improves the network accuracy but also improves the network delay. Adding MF to the above model further improves the network performance. When $MF = 1.5$, the classification accuracy reaches 64.63%. Compared with the original ANN, the classification accuracy of SNN decreased by only 0.27%. Compared with other methods, the accuracy loss is the smallest, and the reasoning time-steps required by the network is also the smallest. It can be seen that CIRM and MF also have good applicability in deep networks and large data sets.

5. Conclusion

In this work, we propose a weight continuous normalization technique for ANN to SNN conversion. This technology ensures the normal spike emission of spiking neurons, and there is almost no loss of network performance after conversion. We also propose a certainty interval reset mechanism of membrane potential. This mechanism solves the problem of membrane potential loss of hard reset neurons and the problem of over activation caused by a soft reset. In the interval reset method, we introduce a modulation factor to further regulate the spike firing rate of neurons. For different tasks, the lossless conversion of the network can be realized by appropriately adjusting the modulation factor. In our experiment, the performance of the network with modulation factor on MNIST and CIFAR-10 exceeds that of the original ANN. Moreover, the algorithm proposed in this paper also has good practicability in deep network and large data set.

In the follow-up research, a general SNN algorithm is developed in close combination with the design requirements of the SNN neuromorphological chip. Image processing systems and instruments based on SNN have great advantages in both speed and power consumption.

Acknowledgements

This work is supported by the National Nature Science Foundation of China (grant No.61871106 and

No.61370152), Key R & D projects of Liaoning Province, China (grant No. 2020JH2/10100029), and the Open Project Program Foundation of the Key Laboratory of Opto-Electronics Information Processing, Chinese Academy of Sciences (OEIP-O-202002).

Conflict of interest

The authors declare that they have no conflict of interest.

References

- [1] J. G. Lee *et al.*, "Deep Learning in Medical Imaging: General Overview," *Korean J Radiol*, vol. 18, no. 4, pp. 570-584, Jul-Aug 2017.
- [2] S. S. Beauchemin, M. A. Bauer, T. Kowsari, and C. Ji, "Portable and Scalable Vision-Based Vehicular Instrumentation for the Analysis of Driver Intentionality," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 2, pp. 391-401, 2012.
- [3] P. Pouladzadeh, S. Shirmohammadi, and R. Al-Maghrabi, "Measuring Calorie and Nutrition From Food Image," *IEEE Transactions on Instrumentation & Measurement*, vol. 63, no. 8, pp. 1947-1956, 2014.
- [4] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: the rising trend of vision based measurement," *IEEE Instrumentation & Measurement Magazine*, vol. 17, no. 3, pp. 41-47, 2014.
- [5] J. Wu, Y. Chua, M. Zhang, Q. Yang, G. Li, and H. Li, "Deep Spiking Neural Network with Spike Count based Learning Rule," *IEEE*, 2019.
- [6] E. Hunsberger and C. Eliasmith, "Spiking Deep Networks with LIF Neurons," *Computer ence*, 2015.
- [7] Z. Hu, T. Wang, and X. Hu, "An STDP-Based Supervised Learning Algorithm for Spiking Neural Networks," in *Neural Information Processing (Lecture Notes in Computer Science, 2017*, pp. 92-100.
- [8] H. Mostafa, "Supervised Learning Based on Temporal Coding in Spiking Neural Networks," *IEEE Trans Neural Netw Learn Syst*, vol. 29, no. 7, pp. 3227-3235, Jul 2018.
- [9] E. O. Neftci, C. Augustine, S. Paul, and G. Detorakis, "Event-Driven Random Back-Propagation: Enabling Neuromorphic Deep Learning Machines," *Front Neurosci*, vol. 11, p. 324, 2017.
- [10] J. Li, W. Hu, Y. Yuan, H. Huo, and T. Fang, "Bio-Inspired Deep Spiking Neural Network for Image Classification," in *Neural Information Processing (Lecture Notes in Computer Science, 2017*, pp. 294-304.
- [11] S. K. Esser *et al.*, "Convolutional networks for fast, energy-efficient neuromorphic computing," *Proc Natl Acad Sci U S A*, vol. 113, no. 41, pp. 11441-11446, Oct 11 2016.
- [12] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Netw*, vol. 99, pp. 56-67, Mar 2018.
- [13] H. Hazan, D. Saunders, D. T. Sanghavi, H. Siegelmann, and R. Kozma, "Unsupervised Learning with Self-Organizing Spiking Neural Networks," 2018, pp. 1-6.
- [14] T. Masquelier and S. J. Thorpe, "Unsupervised learning of visual features through Spike Timing Dependent Plasticity," *PLoS Computational Biology*, vol. preprint, no. 2007, 2005.
- [15] Y. Cao, Y. Chen, and D. Khosla, "Spiking Deep Convolutional Neural Networks for Energy-Efficient Object Recognition," *International Journal of Computer Vision*, vol. 113, no. 1, pp. 54-66, 2014.
- [16] B. Han, G. Srinivasan, and K. Roy, "RMP-SNN: Residual Membrane Potential Neuron for Enabling Deeper High-Accuracy and Low-Latency Spiking Neural Network," *IEEE*, 2020.
- [17] D. Neil and S.-C. Liu, "Effective sensor fusion with event-based sensors and deep network architectures," presented at the 2016 IEEE International Symposium on Circuits and Systems (ISCAS), 2016.
- [18] D. Neil, M. Pfeiffer, and S.-C. Liu, "Learning to be efficient," presented at the Proceedings of the 31st Annual ACM Symposium on Applied Computing, 2016.

- [19] P. O'Connor, D. Neil, S. C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Front Neurosci*, vol. 7, p. 178, 2013.
- [20] C. Panchev and S. Wermter, "Spike-timing-dependent synaptic plasticity: from single spikes to spike trains," *Neurocomputing*, vol. 58-60, pp. 365-371, 2004.
- [21] D. Neil and S.-C. Liu, "Minitaur, an Event-Driven FPGA-Based Spiking Network Accelerator," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 12, pp. 2621-2628, 2014.
- [22] L. Zhang, S. Zhou, T. Zhi, Z. Du, and Y. Chen, "TDSNN: From Deep Neural Networks to Deep Spike Neural Networks with Temporal-Coding," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 1319-1326, 2019.
- [23] W. Tan, D. Patel, and R. Kozma, "Strategy and Benchmark for Converting Deep Q-Networks to Event-Driven Spiking Neural Networks," 2020.
- [24] R. Xiao, Q. Yu, R. Yan, and H. Tang, "Fast and Accurate Classification with a Multi-Spike Learning Algorithm for Spiking Neurons," in *Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019.
- [25] P. U. Diehl, D. Neil, J. Binas, M. Cook, and S. C. Liu, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *International Joint Conference on Neural Networks*, 2015.
- [26] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193-202, 1980.
- [27] A. Tavanaei and A. S. Maida, "Multi-layer unsupervised learning in a spiking convolutional neural network," in *International Joint Conference on Neural Networks*, 2017.
- [28] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going Deeper in Spiking Neural Networks: VGG and Residual Architectures," *Front Neurosci*, vol. 13, p. 95, 2019.
- [29] B. Rueckauer, I. A. Lungu, Y. Hu, M. Pfeiffer, and S. C. Liu, "Conversion of Continuous-Valued Deep Networks to Efficient Event-Driven Networks for Image Classification," *Front Neurosci*, vol. 11, p. 682, 2017.
- [30] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training Deep Spiking Neural Networks Using Backpropagation," *Frontiers in Neuroscience*, vol. 10, 2016.
- [31] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Front Comput Neurosci*, vol. 9, p. 99, 2015.
- [32] G. K. Cohen, O. Garrick, S. H. Leng, T. Jonathan, R. B. Benosman, and V. André, "Skimming Digits: Neuromorphic Classification of Spike-Encoded Images," *Frontiers in Neuroscience*, vol. 10, no. 184, 2016.
- [33] E. Neftci, S. Das, B. Pedroni, K. Kreutz-Delgado, and G. Cauwenberghs, "Event-Driven Contrastive Divergence for Spiking Neuromorphic Systems," *Frontiers in Neuroscience*, vol. 7, no. 8, p. 272, 2013.
- [34] S. Lu and A. Sengupta, "Exploring the Connection Between Binary and Spiking Neural Networks," *Frontiers in Neuroscience*, vol. 14, p. 535, 2020.