Improved Integrate-and-Fire Neuron Models for Inference Acceleration of Spiking Neural Networks

Ying Han [1], Anguo Zhang [2], Qing Chen [1], and Wei Zhu [1]

[1]Affiliation not available
[2]Fuzhou University

October 30, 2023

Improved Integrate-and-Fire Neuron Models for Inference Acceleration of Spiking Neural Networks

Ying Han[a], Anguo Zhang[b,c,*], Qing Chen[d], Wei Zhu[c]

[a]*School of Public Health, Xiamen University, Xiamen, 361005 China.*
[b]*College of Physics and Information Engineering, Fuzhou University, Fuzhou, 350108 China.*
[c]*Research Institute of Ruijie, Ruijie Networks Co., Ltd, Fuzhou, 350002 China.*
[d]*College of Automation, Chongqing University, Chongqing, 400030 China.*

Abstract

This paper studies the effects of different bio-synaptic membrane potential mechanisms on the inference speed of both spiking feed-forward neural networks (SFNNs) and spiking convolutional neural networks (SCNNs). These mechanisms inspired by biological neuron phenomenon, such as electronic conduction in neurons, chemical neurotransmitter attenuation between presynaptic and postsynaptic neurons, are considered to be modeled in mathematical and applied to artificial spiking networks. In the field of spiking neural networks, we model some biological neural membrane potential updating strategies based on integrate-and-fire (I&F) spiking neuron, which includes spiking neuron model with membrane potential decay (MemDec), spiking neuron model with synaptic input current superposition at spiking time (SynSup) and spiking neuron model with synaptic input current accumulation(SynAcc). Experiment results show that compared with the general I&F model (one of the most commonly used spiking neuron models), SynSup and SynAcc can effectively improve the learning speed in the inference stage of SCNNs and SFNNs.

Keywords: Spiking Neural Network, Inference Acceleration, Neural Plasticity
2018 MSC: 00-01, 99-00

1. Introduction

Biologically inspired artificial intelligence has been an increasingly attractive topic during these decades, such as the particle swarm optimization (PSO) [1] which originates from the predation behavior of flocks, the ant colony algorithm which learns from the behaviors of ants finding paths during food search, the genetic algorithm (GA) which simulates the natural evolution of Darwin's biological evolution theory and the evolution process of genetic mechanism, and the artificial neural networks (ANNs) which refers the connection structure of animal neural systems and the way in which information is transmitted and processed, and so on.

Among these algorithms, ANNs have been considered to be the most promising one to realize "true" artificial intelligence, and they have also been widely applied in various applications, e.g. face recognition, object detection, self-driving car, data prediction, etc.. Currently, almost all these mature engineering applications are developed based on the second-generation of ANN models (also called rate-based neural networks, such as the traditional BP networks, Convolutional neural networks (CNNs), LSTM, and so on). However, although these above-mentioned ANNs are historically thought to be brain-inspired, there are fundamental differences in structure, computation and learning rule that compared with the brain.

Spiking neural networks (SNNs), a neural computational framework that more similar to the biological information encoding and neuronal information processing mechanism, have been proved to be a computationally effective framework which is firstly proposed by G. Maass [2] as the third-generation of ANNs, and have also shown their superiorities in rich neural plasticity and low energy consumption. SNNs based neuromorphic vision has become a more and more popular research field over the world. And further, there are many research results about effective computing frameworks of SNN that have been proposed in recent years. [3] derived a new solution method that allowed efficient simulation of Izhikevich spiking neuron model. In [4], the authors studied the necessary time steps and corresponding computational costs required to make the function approximation accurate of spiking neuron models, including Hodgkin-Huxley, Izhikevich, and leaky integrate-and-fire model. And they concluded that the leaky integrate-and-fire model needs the least number of computations and the least operations for a crude approximation. [5] proposed an evolutionary algorithms and graphics processing units (GPUs) based automated parameter tuning framework that capable of tuning SNNs quickly and efficiently. [6] presented a linear spiking decoding algorithm for computationally efficient implementation of the decoding joint model for the electrode spike counts and waveform features, which is reported to have low storage and computationally requirements.

*Corresponding author
Email addresses: 18050194992@qq.com (Ying Han), anrial@live.cn (Anguo Zhang), chenqing@cqu.edu.cn (Qing Chen), ruilangzhu@icloud.com (Wei Zhu)

One of the main drawbacks of SNNs is the lower real-time performance compared with the second generation of ANNs due to that SNNs take some time to reach the homeostatic firing state. [7] proposed a mode of spike information propagation through feedforward networks which consisting of layers of integrate-and-firing neurons, and the experimental results demonstrated that this mode allows for fast computation with population coding based on firing rates. [8] reported that the output delay involved in achieving acceptable classification accuracy, and the suitable trade-off between energy benefits and classification accuracy can be obtained by optimizing the input firing rate and output delay. In [8], Diehl et al. proposed two normalization methods named as Model Normalization and Data Normalization to obtain fast and accurate SNNs. Zhang et al. [9, 10] applied intrinsic plasticity, an unsupervised biological plausible mechanism, to spiking feedforward neural networks to accelerate the convergence speed during the inference stage.

Unlike the connection weights normalization methods in [8] or external neuronal parameters importation methods in [9, 10], in this paper, we proposed three novel biological plausible spiking neuron models which update their states of membrane potential only using local information. We constructed both spiking feedforward neural networks (SFNNs) and spiking convolutional neural networks (SCNNs) consisting of the proposed neuron models, respectively, and then compared their computational performance in terms of real-time inference with the conventional I&F spiking neuron model. The experimental results show that except the MemDec model, the inference speed of the other two proposed models (SynAcc and SynSup) is significantly better than the I&F model, while still achieve slightly higher classification accuracy.

The rest of this paper is organized as following. Sec.2 introduces some basic concept of spiking neural network. In Sec.3, three different inherent properties of spiking neuron model are proposed. The spiking neural network construction method, as well as the datasets are presented in Sec. 4. Experiment results are showed in Sec.5. At last, the conclusion has been drawn to end this paper in Sec.6.

2. Spiking Neural Network

Fig.1 shows the physical connection structure between two biological neurons and the signal transmission direction is also marked. The postsynaptic neuron (the larger one in the left) receives the signal from the presynaptic neuron (the smaller one in the right) by connecting its dendrites to the presynaptic neuron's axon terminals. In biological neural systems, signals are transmitted at a faster speed in the form of electrical current in neural bodies, while among neurons, signals are transmitted through chemicals (called neurotransmitters). Due to both the signal conversion between electrical current and neurotransmitters, and the time cost of spreading the neurotransmitters in the gap of presynaptic axon terminals and postsynaptic dendrites, signal transmission speed turns relative slower than through electrical current.

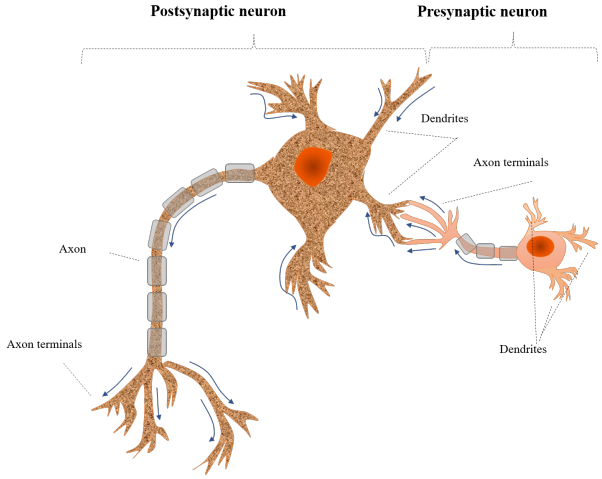


Figure 1: A simple presentation of biological neuron and information transmission among neurons.

In the long-term evolutionary process, animals have always tried to transmit the sensory signals of various parts of the limb to the brain in the least costly and most efficient way, and to transmit the command signals of the brain to various executing organs. Faster signal transmission helps animals to perceive the external environment and respond more quickly. Recently, researchers found that the event-driven mechanism

In conventional artificial neural networks (ANNs), input signal is feed into network at one time and processed layer-by-layer, then network produces the output value, while in SNNs, input signal processing flow of ANNs, in SNNs, inputs are typically transformed into streams of spike events at first, then the created spike streams are feed into SNNs and communicate information to subsequent layers over time.

2.1. Spiking Computational Operation

SNNs use spikes to transmit and process information instead of continuous numeric values, thus some conventional operations for continuous-valued neurons should be mapped into spiking ones before using them [8, 11].

1) For ReLU activation function, it is converted to

$$a_i = \max\left\{0, \sum_j w_{ji} s_j\right\} \tag{1}$$

where a_i denotes the activation of neuron i, w_{ji} is the connection weight from neuron j to i, s_j is the spike signal of j, and $s_j = 1$ only if neuron j fires, otherwise $s_j = 0$.

2) For convolutional computation, it is converted to

$$a^k = f\Big(\sum_l W^k * a^l + b^k\Big) \tag{2}$$

where $\{W^k, (k = 1, 2, ..., n)\}$ denotes a set of convolutional kernels, $\{a^k, (k = 1, 2, ..., n)\}$ denotes the resulting feature maps with the same number with convolutional kernels. f is an activation function, the symbol * is a 2D valid-region convolution, and b^k is a bias term.

3) For average pooling and max pooling computation

Pooling is a common operation to reduce the size of preceding feature maps, which often follows with convolutional layers. Both average pooling and max pooling have been the main choices in building CNNs. For averaging kernel in pooling layer, the activation can also be identical to Equ. (2), except that the kernel weights W^k are fixed to $1/size(W^k)$, where $size(W^k)$ represents the multiplication of the width and height of kernel W^k. While for max kernel in pooling layer, if any of the neurons within a pooling window is fired, then it outputs 1, otherwise it outputs 0.

4) For Softmax classification, it is converted to

$$c = \text{argmax}\Big(\{O_i(t), i = 1, 2, ..., P\}\Big) \tag{3}$$

where t denotes the time step from 0, P denotes the number of neuron in output layer, and $O_i(t)$ is the count of spike times of neuron i from time 0 to t. c is the practical output of label index.

2.2. Training SNNs

Several algorithms have been proposed to well train an SNN. The most popular one is spike-time-dependent plasticity (including related STDP-based algorithm), which is a bio-inspired unsupervised learning method found in the mammalian visual cortex [12, 13, 14]. By biological STDP mechanism, synapses through which a presynaptic spike arrived before (respectively after) a postsynaptic one are reinforced (respectively depressed), it brings benefit to primates, especially humans, can learn from far few examples while most of them are unlabelled. A simplified version of STDP used for training artificial SNNs was proposed by Masquelier in 2007, where a connection weight between two neurons depends on the exact spiking times of them, respectively, for more details, see [15].

Akin to conventional error-backpropagation training method, supervised learning rules using the output error backpropagation during the training procedure, like $SpikeProp$ and its extensions [16, 17, 18, 19], aiming to minimize the time difference between the target spike and the actual output spike. Tempotron, proposed by [20], is another gradient-descent learning approach to minimizing an energy cost function determined by the distance of the neuron membrane potential and its corresponding firing threshold.

Unlike the above-mentioned methods that train an SNN model using the exact signal of spiking time, [11] proposed an SCNN generating solution by directly converting from the corresponding well-trained ANN model. What should be paid attention to is the difficulties of representing the negative values and biases in conventional rate-based ANNs. To avoid this obstacle, rectified linear unit (ReLU) activation function and zero biases are set to the ANN before training it. [11] reported the method outperformed other previous approaches, and [8] extended it to spiking fully-connected feed-forward neural network (SFNN) conversion and presented several optimization tools for both SCNN and SFCN for faster classification based on fewer output spikes. Further, [21] developed a set of tools, as well as presented related theory for converting more other popular elements of CNN (e.g. max-pooling, batch normalization, softmax classification) into spiking form.

2.3. Inference Latency

In traditional rate-based neural networks, signals are transmitted from the input layer to the neural network at one time, and processed through layers, resulting in the final output by the output layer. However, in SNNs, signals are presented by streams of spike events, and flow layer by layer via spikes which created by neurons, ultimately, drive firing of output neurons that collect evidence over time. This mechanism gives SNN some advantages such as efficient processing of time-varying inputs [22] and high computational performance on specialized hardware [23].

However, it also implies that even for a time-invariant input, network output maybe varies over time, especially at the beginning of the spike signal input to the network because that sufficient spike evidence has not been collected by the output neurons. This phenomenon was studied by [24], which named pseudo-simultaneity, means that we can obtain a reliable or stable output immediately once the signal flows from the input layer to the output layer. To improve the real-time performance of SNN, [8] proposed two optimization methods to normalize the network weights, namely model-based normalization and data-based normalization, so that the neuron activations were sufficiently small to prevent from overestimating output activations. Retraining based layer-wise quantization method to quantize the neuron activation and pooling layer incorporation to reduce the number requirement of neurons were proposed in [25], the authors reported that these methods can build hardware-friendly SNNs with ultra-low-inference latency.

3. Spiking Neuron Model

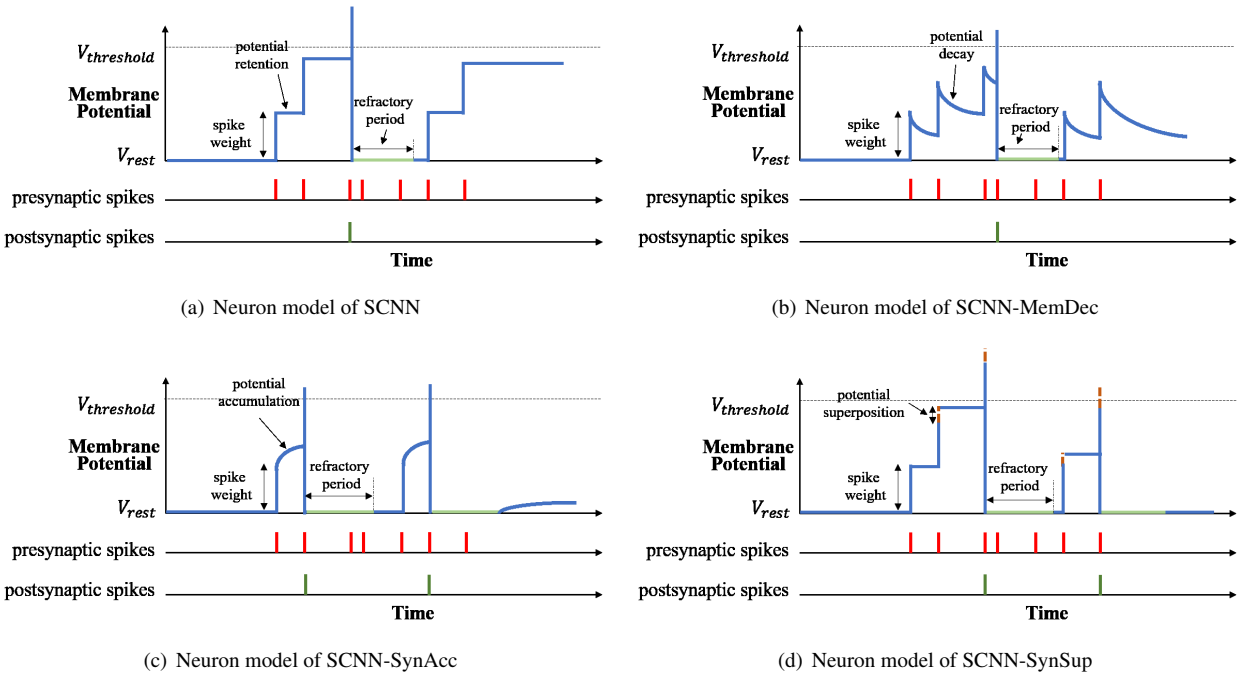
In this work, we proposed several spiking neuron models inspired by possible biological neural mechanisms, including spiking neuron model with membrane potential decay (MemDec), spiking neuron model with synaptic input current accumulation (SynAcc) and spiking neuron model with synaptic input current superposition at spiking time (SynSup). All these proposed models are studied whether they contribute to computational efficiency.

The membrane potential dynamics of a single IF neuron is defined by

$$\frac{dV_{mem}(t)}{dt} = I(t) \tag{4}$$

where $V_{mem}(t)$ denotes the membrane potential at time t, and if $V_{mem}(t)$ crosses the firing threshold $V_{threshold}$, a spike is generated and it will be reset to the rest potential V_{reset} instantaneously and then stay at V_{reset} for a time period t_{ref}, namely the refractory period. $I(t)$ presents the sum of presynaptic input current, and it can be simply calculated by

$$I(t) = \sum_{i \in N} w_i \delta(t - t_s^{(i)}), t_s^{(i)} \in T_S^{(i)} \tag{5}$$



(a) Neuron model of SCNN

(b) Neuron model of SCNN-MemDec

(c) Neuron model of SCNN-SynAcc

(d) Neuron model of SCNN-SynSup

Figure 2: Operation of four event-driven spiking neuron models. It should be noted that the input spike weight, refractory period after reset, threshold voltage $V_{threshold}$, rest voltage V_{reset} are the neuron operation parameters, while current membrane voltage is the neuron state parameter. (a) Operation diagram of general IF neuron model. (b) Operation diagram of IF neuron model with membrane potential decaying. (c) Operation diagram of IF neuron model with continuous synaptic input current accumulation. (d) Operation diagram of IF neuron model with synaptic input current superposition at spiking time.

where N is the presynapse set of the IF neuron. w_i is the weight of the ith presynapse, $T_S^{(i)} = \{t_1^{(i)}, t_2^{(i)}, ...\}$ denotes the set of spiking time instants of the ith presynapse, $\delta(t - t_s^{(i)})$ is a dirac-delta function, that is, $\delta(t - t_s^{(i)}) = 1$ if $t = t_s^{(i)}$, otherwise $\delta(t - t_s^{(i)}) = 0$. The neuron membrane potential update diagram is as shown in Fig.2(a).

3.1. IF Model with Membrane Potential Decay

Due to the ion permeation effect of the biological nerve cell membrane, the ions (for example, sodium ions, potassium ions and chloride ions both inside and outside the cell membrane of a neuron) spontaneously flow from the high concentration side to the low concentration side, thereby changing the membrane potential.

Motivated by this biological phenomenon, we also performed a simple model simulation, namely, the spiking neuron model with membrane potential decay (MemDec) of this mechanism. The MemDec neuron model is presented as Fig.2(b), what different with the general neuron model is that the membrane potential decays over time described by

$$\begin{aligned}\frac{dV_{mem}(t)}{dt} &= I(t) \\ &\quad - \lambda \int_{\hat{t}_s}^{t} \exp\left(-\frac{\tau - \hat{t}_s}{\tau_s}\right) d\tau, t \in [\hat{t}_s, \hat{t}_{s+1})\end{aligned} \tag{6}$$

where $\hat{t}_s$ is the spike time of this neuron itself and $\hat{t}_{s+1}$ is the next spike time, τ_s is a time constant, and λ is a coefficient.

3.2. IF Model with Synaptic Input Current Accumulation

Spiking neuron model with synaptic input current accumulation (SynAcc) mimics the biological neuron mechanism. Due to the capacitance and resistance effects of neurons, the ions inside the neurons do not flow out completely in an instant time, but flow out in an approximate exponential form over time. The SynAcc neuron model is designed as

$$\begin{aligned}\frac{dV_{mem}(t)}{dt} &= I(t) \\ &\quad + w_i \sum_i \int_{t_s^{(i)}}^{t} \exp\left(-\frac{\tau - t_s^{(i)}}{\tau_r}\right) d\tau, t \in [t_s^{(i)}, t_{s+1}^{(i+1)})\end{aligned} \tag{7}$$

where τ_r is a time constant, $t_s^{(i)}$ is the spike time of the ith presynaptic neuron, and $t_{s+1}^{(i)}$ denotes the next spike time. In Fig.2(c), a simple membrane potential update mechanism is given for a clear understanding of SynAcc.

3.3. IF Model with Synaptic Input Current Superposition at Spiking Time

The model with Synaptic Input Current Superposition at Spiking Time (SynSup) can be given by

$$\begin{aligned}\frac{dV_{mem}(t)}{dt} &= I(t) \\ &\quad + \sum_i I^{(i)}(t) \left(\exp\left(-\frac{t - t_{s-1}^{(i)}}{\tau_p}\right) - \exp\left(-\frac{t - t_{s-1}^{(i)}}{\tau_q}\right) \right)\end{aligned} \tag{8}$$

where $I^{(i)}(t)$ denotes the input current produced by the ith presynaptic neuron, and $\sum_i I^{(i)}(t) = I(t)$, τ_p and τ_q are time constants satisfying $\tau_p > \tau_q$.

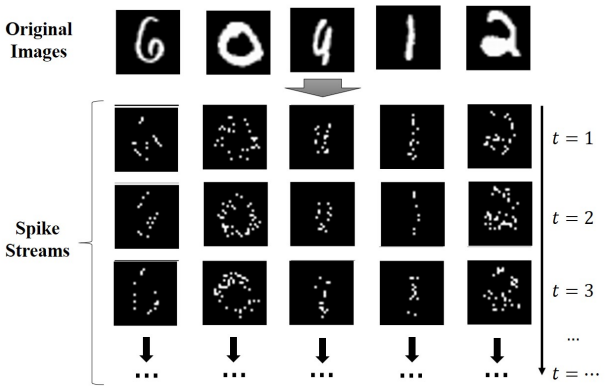


Figure 3: Transform original images to spike streams using Poisson sampling.

3.4. Comparison between These Models

All the spiking models can be implemented by the event-driven way, and they focus on regulating the presynaptic input current which received by the dendrites of postsynaptic neuron, when their membrane potential exceeds the threshold value, they are activated to fire and their membrane potential are then reset to V_{reset}. The normal IF neuron model only changes its membrane potential by receiving input current if some of the presynaptic neurons fire to generate spikes at a time step, otherwise, its membrane potential keeps unchanged. However, MemDec, SynAcc and SynSup continuously change their membrane potential based on themselves or external input current. Among them, the membrane potential of MemDec gradually decreases in the non-firing period due to the current decay of the neuron membrane. In the SynAcc mechanism, all presynaptic neurons that have fired will continue to deliver current to the postsynaptic neurons, besides the connection weights, the time interval between current time and the last firing time of the presynaptic neurons also affects the total amount of current delivered by presynaptic neurons to postsynaptic neuron. SynSup considers an input current enhancement mechanism, that is the shorter the time interval between pre- and post-synaptic neurons, the more obvious the subsequent output current enhancement effect.

The most significant difference between SynAcc and SynSup is that, in SynAcc mechanism, no matter a presynaptic generates a spike or not, the postsynaptic neuron always receives synaptic current from it. For a deeper understanding, one can compare the diagram Fig.2(d) of SynSup with Fig.2(c) of SynAcc.

4. Material and Method

4.1. Dataset

Two image classification oriented benchmarks, MNIST and Fashion-MNIST, are used to compare the performance between SNN, SNN-MemDec, SNN-SynAcc and SNN-SynSup. MNIST is a handwritten digit dataset that has been a ubiquitous benchmark in machine learning, and it is also chosen for our experiments. MNIST consists of 60000 labeled training samples and 10000 labeled test samples, each sample is organized as a 28×28 pixel grayscale image. Fashion-MNIST [26] is another benchmarking dataset which is intended to serve as a direct drop-in replacement for the original MNSIT dataset, and it is also consisting of the same number and pixel scale of a sample as MNIST. Fashion-MNIST contains 10 classes of samples which labeled "T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag" and " Ankle boot".

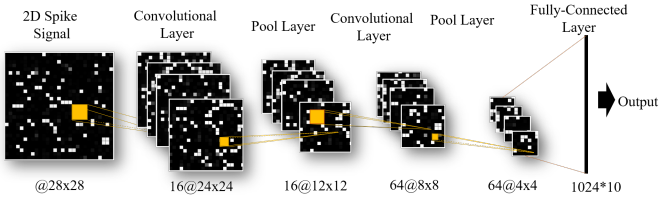


Figure 4: A diagram of general convolutional neural networks (CNNs) consisted of convolutional layers and pool layers.

It should be noted that the MNIST image is not directly inputted to the SFNN and SCNN, instead, the original image firstly converted into 2-dimension spike streams, and then input the spike signal to the input layer of SFNN or SCNN. In detail, as the spike conversion method proposed by [26], the intensity values of MNIST images are linearly normalized between 0 and 1, and the 2-dimension spike signal sequence is generated by Poisson distribution based on the image's intensity values, further, the probability of a spike generated for an image pixel is proportional to the input rates. which is as presented in Fig.3.

4.2. Network Model Construction

Two classical artificial neural network models, feed-forward neural network (FNN) and convolutional neural network (CNN), are used as the fundamental network frameworks. There are several types of training methods to get the spiking-version models of FNN and CNN, such as error backpropagation-like algorithms, Hebbain-like and reinforcement learning-based algorithms, direct conversion from ANNs, and so on. However, it should be noted that in this paper, we don't focus on how to get the well-trained spiking network models, but focus on the effects of the above mentioned synaptic mechanisms on spiking neurons.

The SFNN consists of an input layer, two hidden layers with 1200 neurons per layer, and an output layer. The structure of SCNN is as shown in Fig.4, which constructed by two convolutional layers, two average pool layers and a fully-connected layer. The input signal of 2-dimension spike is with the size of 28×28, convolved by 16 convolutional kernels of size 5×5, and then averagely pooled with the window size 2×2. The convolutional and pooling operations are repeated in a second stage with 64 maps, then flatted by a fully connected layer of size 1024×10, where 10 is the number of output nodes determined by the class number of MNIST labels.

5. Experiment Results

5.1. Parameter Setting

Some important model parameters are given in TABLE.1. It should be noted that since the connection weights of the SFNN and SCNN networks are obtained through the conversion of rate-based FNN and CNN which have been well trained before, parameters for training the rate-based networks need to be introduced here because they have no direct effects on the SFNN and SCNN.

5.2. Inference Speed and Accuracy on Normal Test Sets

Two key performance indicators, i.e., final accuracy (FA) and matching time (MT) are measured to evaluate the proposed spiking networks, where FA denotes the final classification accuracy when the spiking network achieves homeostatic state, and MT denotes the first time when the network achieves the accuracy that greater than 99% of FA.

Table 2 shows both the FA and MT values of different neuron updating strategies of SFNN and SCNN. The faster increase in classification accuracy implies that the spiking network has faster learning speed at the inference stage. It can be seen the network performance difference exhibited by different neuron updating strategies are particularly noticeable at low input rates. However, even at different input rates, the network performance under these neuron updating strategies remains consistently ordered.

SNN-SynSups (both SFNN-SynSup and SCNN-SynSup) present the best performance in terms of synaptic plasticity, in Fig.2(d), we can know that compared with SNNs (SFNN and SCNN), SNN-SynAccs (SFNN-SynAcc and SCNN-SynAcc) improve the learning speed at the beginning, however, it cannot be guaranteed that the network can achieve high classification accuracy in the subsequent time. Further, SNN-MemDecs (SFNN-MemDec and SCNN-MemDec) reduce the learning speed of SNNs in spite of remaining the same final classification accuracy. Thus, we can conclude that SCNN-SynSups get better performance than SNNs on learning speed and classification accuracy, while SNN-SynAccs and SNN-MemDecs both show their performance disadvantage especially at low input firing rates.

5.3. Inference Speed and Accuracy on Noisy Test Sets

We also compare the classification accuracy and inference speed between SNNs, SNN-MemDecs, SNN-SynAccs and SNN-SynSups on the test datasets with additional noises, while the original ANNs to be converted are trained on pure training sets without noises. To more thoroughly test the effects of noise, five different types of noise including Gaussian noise, Rayleigh noise, Uniform noise, Gamma noise as well as Salt&Pepper noise are considered, further, the mixture of these five types of noise are also tested. Fig. 5 shows the examples of pure training dataset and noisy test dataset of MNIST. Fig. 6 and Fig.7

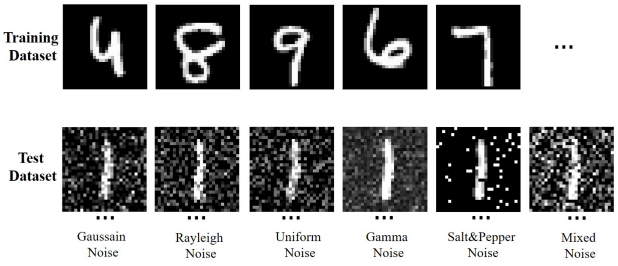


Figure 5

5.4. Spiking Activity

Fig.8 shows the spiking activities of six representative maps of the two convolutional layers in SCNNs of different neuron updating strategies within the initial 200ms at input firing rate of 200Hz, the spiking activities of the two average pool layers are omitted due to that their spiking activities are directly proportional to those of convolutional layers. In Fig.8, the spiking activities from 0 to 200ms are depicted once every 10ms period.

The spiking activities of the first convolutional layer of these strategies are similar, because their previous layer is the input layer, and the firing rate of their presynaptic neurons of the input layer is set to be the same, that is, 200 Hz. So only the difference in the update strategy of individual neurons has not caused a particularly significant difference in spiking activity. However, in the second convolutional layer, the spiking activity of the neurons in this layer shows a more significant difference due to the combination of the accumulative difference of spiking activity of the previous network layers and update strategy difference of membrane potential of this neural layer. Besides, the second average pooling layer which determined by the second convolutional layer directly affects the final classification result of the fully-connected layer. It means that the spiking activity of the second convolutional layer has a greater impact on the network output.

5.5. Input Firing Rate

The input firing rate has been proven to have an important impact on the spiking activity of SNN [8, 27, 28]. In this part, we study the detailed impact of the input firing rate, typically, we present the spiking activities within the initial 100ms of SCNN as shown in Fig.9.

It can be easily obtained that a higher input rate leads higher intensity of spiking activities, which is also consistent with the results in most other reports. Further, too low input rate will cause too low input stimulation to SNN, and results in the under-firing phenomenon of SNN due to the lack of sufficient input stimulation. On the other hand, because of the saturation of the input stimulation, there exists a marginal effect of the highest firing rate of SNN, an excessive input firing rate does not trigger infinitely high spiking activity of the network.

From the perspective of energy consumption and computational effectiveness, too low input rate leads few spiking events of neurons, thus SNN needs more time to reach a homeostatic firing state to get a high and stable output accuracy, which results in poor real-time performance. However, a lower input

Table 1: Parameter settings in our experiment

Strategy	Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
Global	time step	1ms	$V_{threshold}$	2	V_{rest}	0	refractory period	0ms
(SFNN/SCNN)-MemDec	λ	0.5	τ_s	0.001			9	
(SFNN/SCNN)-SynAcc	τ_r	0.004						
(SFNN/SCNN)-SynSup	τ_p	0.004	τ_q	0.002				

Table 2: Performance comparison of SNN, SNN-MemDec, SNN-SynAcc and SNN-SynSup models on FA and MT indicators

Model	**Metric**	**MNIST**			
		50Hz	**200Hz**	**500Hz**	**1000Hz**
SFNN	FA [%]	98.52	98.50	98.65	98.62
	MT [ms]	385	79	34	17
SFNN-MemDec	FA [%]	97.95	98.46	98.57	98.59
	MT [ms]	549	94	45	23
SFNN-SynAcc	FA [%]	98.52	98.59	98.68	98.65
	MT [ms]	147	62	29	16
SFNN-SynSup	FA [%]	**98.59**	**98.61**	**98.71**	**98.66**
	MT [ms]	**95**	**47**	**23**	**13**
SCNN	FA [%]	98.82	98.42	98.45	**98.84**
	MT [ms]	391	101	32	25
SCNN-MemDec	FA [%]	98.76	99.08	98.97	98.76
	MT [ms]	653	189	47	39
SCNN-SynAcc	FA [%]	**98.86**	**99.10**	99.03	98.34
	MT [ms]	249	83	27	17
SCNN-SynSup	FA [%]	98.62	99.05	**99.06**	98.65
	MT [ms]	**195**	**66**	**25**	**12**
Model	**Metric**	**Fashion-MNIST**			
		50Hz	**200Hz**	**500Hz**	**1000Hz**
SFNN	FA [%]	90.12	89.81	**90.21**	**90.22**
	MT [ms]	368	134	67	43
SFNN-MemDec	FA [%]	89.87	89.93	90.19	90.19
	MT [ms]	533	204	82	50
SFNN-SynAcc	FA [%]	**90.24**	89.96	90.13	89.94
	MT [ms]	219	120	59	37
SFNN-SynSup	FA [%]	90.21	**90.03**	90.18	90.13
	MT [ms]	**193**	**76**	**46**	**29**
SCNN	FA [%]	**92.13**	91.48	92.03	91.90
	MT [ms]	434	138	45	28
SCNN-MemDec	FA [%]	91.91	91.82	**92.20**	91.80
	MT [ms]	670	209	75	42
SCNN-SynAcc	FA [%]	92.00	**92.06**	92.12	**91.96**
	MT [ms]	298	95	31	19
SCNN-SynSup	FA [%]	91.95	**92.06**	92.00	91.91
	MT [ms]	**226**	**76**	**24**	**12**

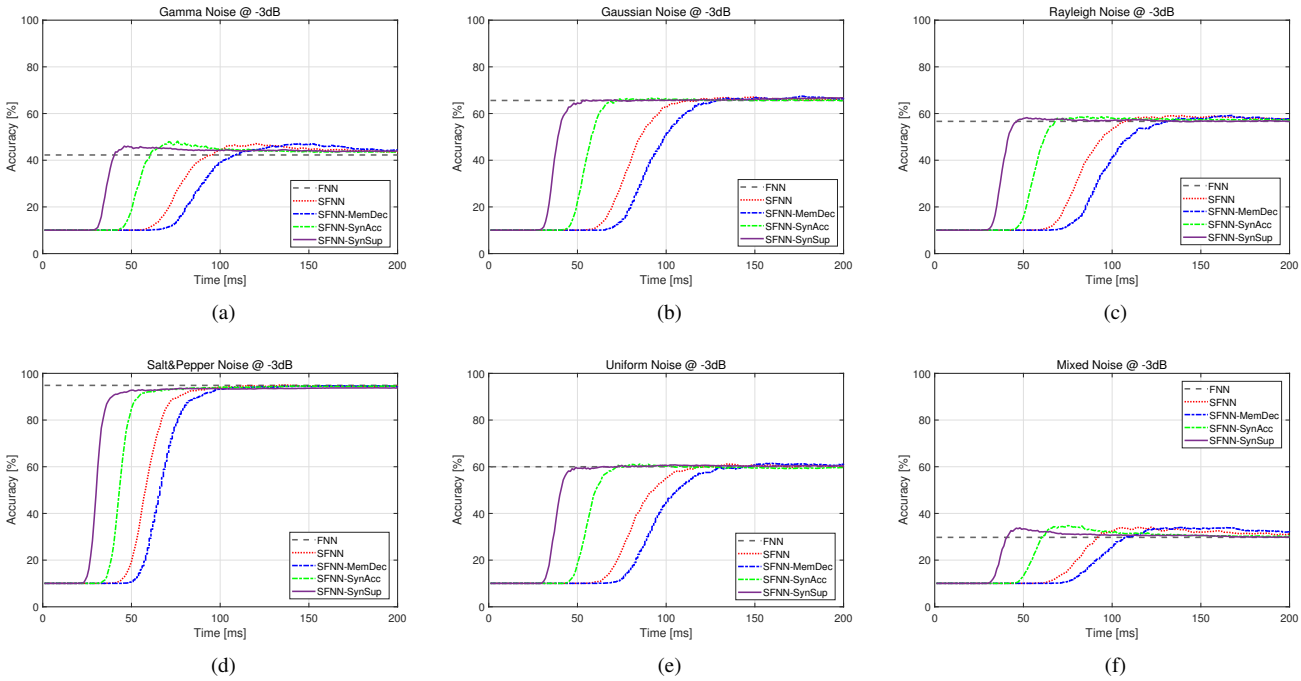


Figure 6: Comparison of accuracy and learning speed (convergence time) between spiking neuron models with other synaptic plasticity mechanisms of SFNN under different types of noisy MNIST test set.

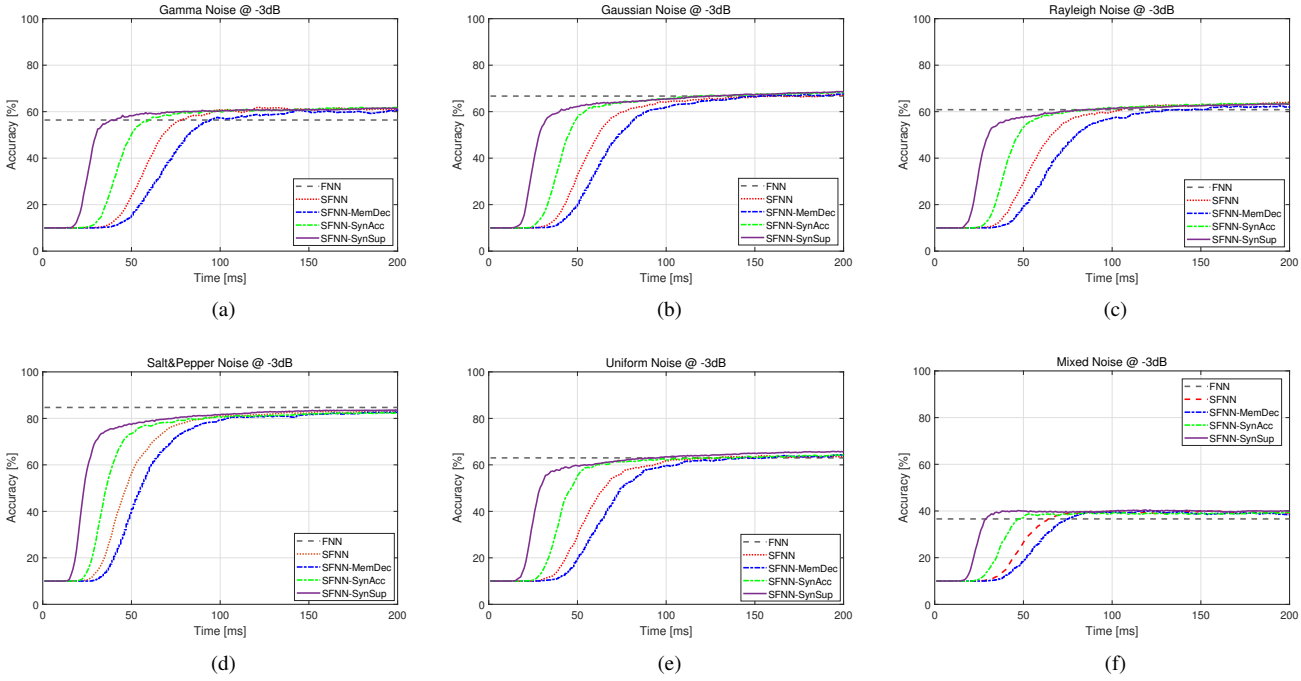


Figure 7: Comparison of accuracy and learning speed (convergence time) between spiking neuron models with other synaptic plasticity mechanisms of SFNN under different types of noisy Fashion-MNIST test set.

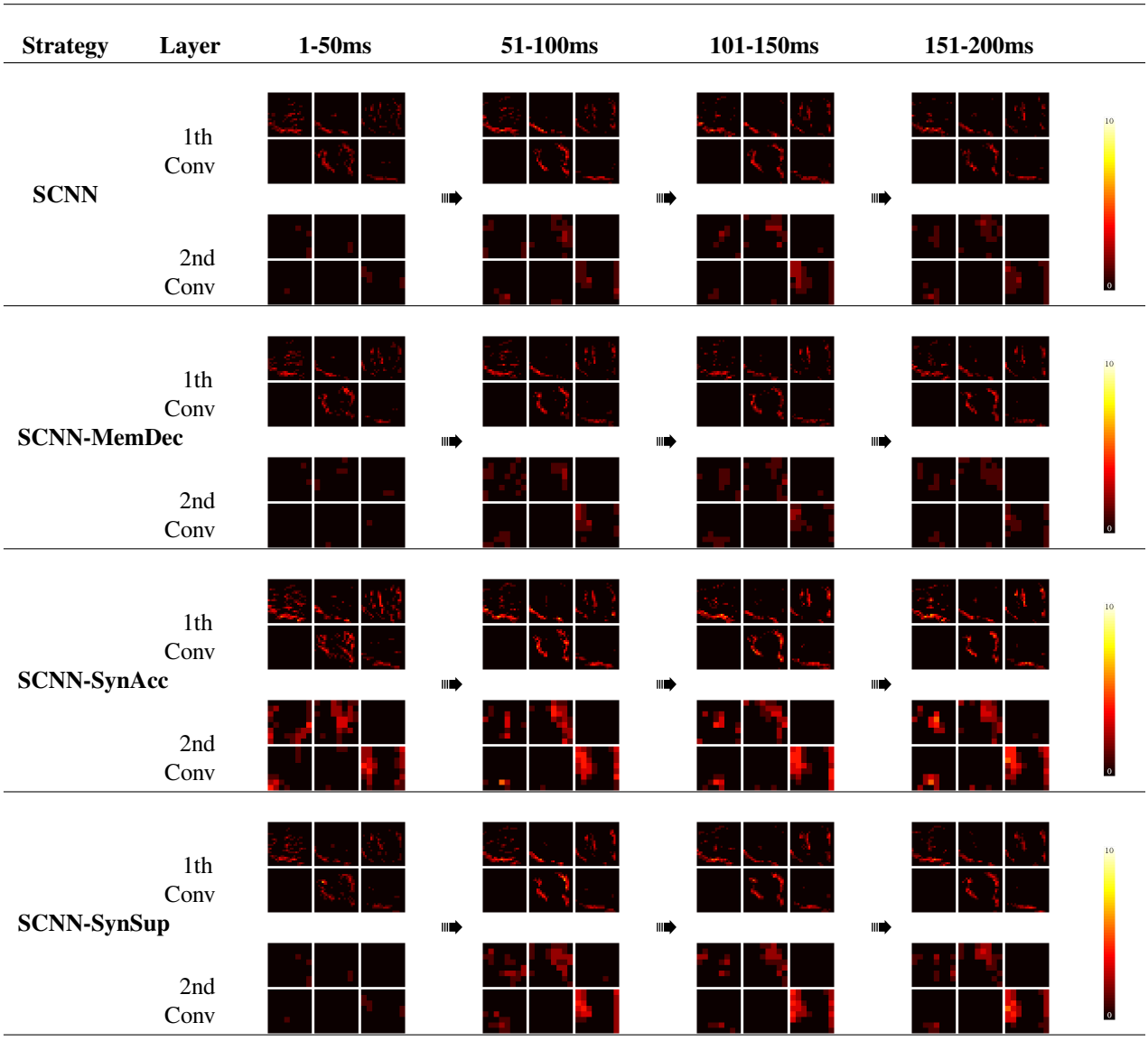


Figure 8: Spiking activities of different spiking neuron models at the input firing rate of 200Hz. From the beginning to 40ms, we divide the time into 4 segments, each segment has a period of 10ms, this table shows the spiking activities of six resulting maps of each convolutional layer.

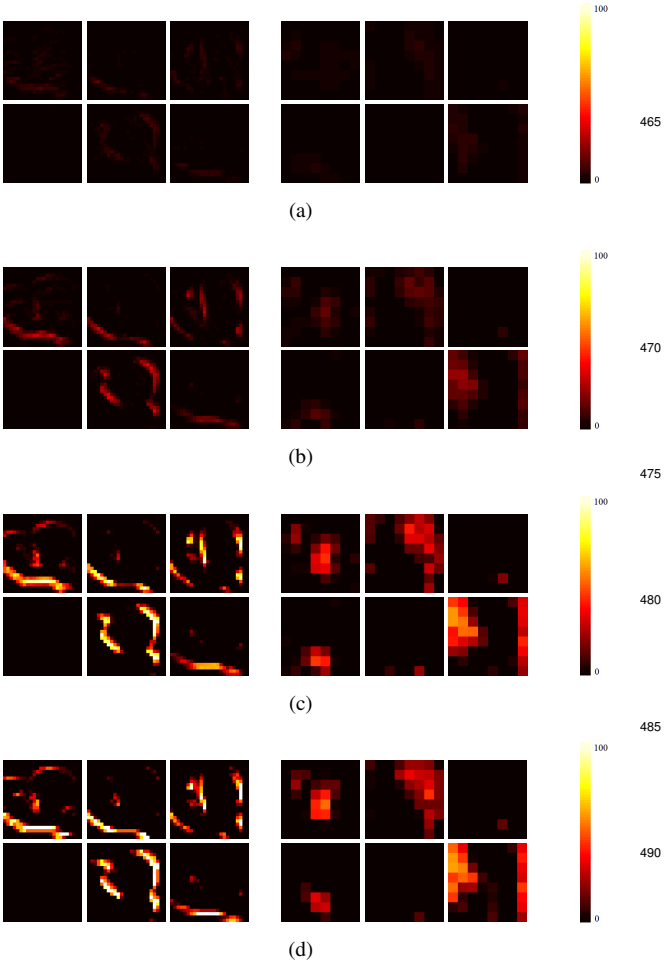


Figure 9: Spiking activities of the convolutional layers of SCNN under different input rates, the left part of each subfigure represents the 1st convolutional layer, and the right part represents the 2nd convolutional layer. (a) Input rate = 50Hz. (b) Input rate = 200Hz. (c) Input rate=1000Hz. (d) Input rate=5000Hz.

rate also makes less updating operations of neuron state triggered by software or hardware, which saves more computational energy during a certain period. The consequence of the high input firing rate is the opposite of the above.

Thus, we have to choose a suitable input firing rate to strike a trade-off between real-time performance and energy consumption, it is also meaningful to work on more effective methods that improve the real-time performance by reducing the time delay of reliable output under a low input firing rate.

6. Conclusion

In this paper, we mathematically model several different neuron membrane potential response mechanisms and construct them on conventional I&F neuron model. We built spiking feed-forward neural networks (SFNNs) and spiking convolutional neural networks (SCNNs) with different neuron models, respectively. It is found from the experiment results that whether it is on noise-free test data sets or on test data sets containing multiple types of additional noises, Synaptic Input Current Superposition at Spiking Time (SynSup) could greatly lift the learning speed as well as classification accuracy, especially under low input firing rate. The experimental results show that, unlike the network structure and connection weights adjustment methods proposed by other research works, our neuron membrane potential response mechanism provides a new perspective for improving the inference speed of the network.

References

[1] J. Kennedy, R. Eberhart, Particle swarm optimization, IEEE International Conference on Neural Networks 4.

[2] W. Maass, Networks of spiking neurons: The third generation of neural network models, Neural Networks 9 (10) (1997) 1659—1672. doi:10.1016/S0893-6080(97)00011-7.

[3] M. D. Humphries, K. Gurney, Solution methods for a new class of simple model neurons, Neural Computation 19 (12) (2007) 3216–3225. doi:10.1162/neco.2007.19.12.3216.

[4] M. J. Skocik, L. N. Long, On the capabilities and computational costs of neuron models, IEEE Transactions on Neural Networks and Learning Systems 25 (8) (2014) 1474–1483. doi:10.1109/TNNLS.2013.2294016.

[5] K. D. Carlson, J. M. Nageswaran, N. Dutt, J. L. Krichmar, An efficient automated parameter tuning framework for spiking neural networks, Frontiers in Neuroscience 8. doi:10.3389/fnins.2014.00010.

[6] V. Ventura, S. Todorova, A computationally efficient method for incorporating spike waveform information into decoding algorithms, Neural Computation 25 (5) (2015) 1033–1050. doi:10.1162/NECO_a_00731.

[7] V. Rossum, M. C. W., Turrigiano, G. G., S. B. Nelson, Fast Propagation of Firing Rates through Layered Networks of Noisy Neurons, The Journal of Neuroscience 22 (5) (2002) 1956–1966. doi:10.1523/JNEUROSCI.22-05-01956.2002.

[8] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, M. Pfeiffer, Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing, International Joint Conference on Neural Networksdoi:10.1109/IJCNN.2015.7280696.

[9] S. Zhang, A. Zhang, Y. Ma, W. Zhu, Intrinsic Plasticity Based Inference Acceleration for Spiking Multi-Layer Perceptron, IEEE Access 7 (2019) 73685–73693. doi:10.1109/ACCESS.2019.2914424.

[10] A. Zhang, H. Zhou, X. Li, W. Zhu, Fast and robust learning in Spiking Feed-forward Neural Networks based on Intrinsic Plasticity mechanism, Neurocomputing 365 (2019) 102–112. doi:10.1016/j.neucom.2019.07.009.

[11] Y. Cao, Y. Chen, Spiking deep convolutional neural networks for energy-efficient object recognition, International Journal of Computer Vision 113 (1) (2014) 54–66. doi:10.1007/s11263-014-0788-3.

[12] C. D. Meliza, Y. Dan, Receptive-field modification in rat visual cortex induced by paired visual stimulation and single-cell spiking, Neuron 49 (2) (2006) 183–189. doi:10.1016/j.neuron.2005.12.009.

[13] D. B. McMahon, D. A. Leoplod, Stimuls timing-dependent plasticity in high-level vision, Current Biology 22 (4) (2012) 332–337. doi:10.1016/j.cub.2012.01.003.

[14] S. Huang, C. Rozas, et al, Associate hebbian synaptic plasticity in primate visual cortex, The Journal of Neuroscience 34 (22) (2014) 7575–7579. doi:10.1523/JNEUROSCI.0983-14.2014.

[15] T. Masquelier, S. J. Thorpe, Unsupervised learning of visual features through spike timing dependent plasticity, PLoS Computational Biology 3 (2) (2007) e31. doi:10.1371/journal.pcbi.0030031.

[16] S. M. Bohte, J. N. Kok, H. L. Poutre, Error-backpropagation in temporally encoded networks of spiking neurons, Neurocomputing 48 (1) (2002) 17–37. doi:10.1016/S0925-2312(01)00658-0.

[17] B. Schrauwen, J. V. Campenhout, Improving spike-prop: Enhancements to an error-backpropagation rule for spiking neural networks, Proceedings of ProRISC Workshop 11 (2004) 301–305.

[18] P. Tino, A. J. S. Mills, Learning beyond finite memory in recurrent networks of spiking neurons, Neural Computation 18 (3) (2006) 591–613. doi:10.1162/neco.2006.18.3.591.

[19] H. Fang, Y. Wang, J. He, Spiking neural network for cortical neuronal spike train decoding, Neural Computation 22 (4) (2010) 1060–1085. doi:10.1162/neco.2009.10-08-885.

[20] R. Gutig, H. Sompolinsky, The tempotron: a neuron that learns spike timing-based decisions, Nature Neuroscience 9 (3) (2006) 420–428. doi:10.1038/nn1643.

[21] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, Theory and tools for the conversion of analog to spiking convolutional neural networks, arXiv:1612.04052 [cs, stat].

[22] Y. Bodyanskiy, A. Dolotov, I. Pliss, M. Malyar, A fast learning algorithm of self-learning spiking neural network, IEEE International Conference on Data Stream Mining & Processing (2016) 104–107doi:10.1109/DSMP.2016.7583517.

[23] P. Merolla, J. Arthur, F. Akopyan, et al, A digital neurosynaptic core using embedded crossbar memory with 45pj per spike in 45nm, IEEE Custom Integrated Circuits Conferencedoi:10.1109/CICC.2011.6055294.

[24] L. Camunas-Mesa, C. Zamarreno-Ramos, A. Linares-Barranco, et al, An event-driven multi-kernel convolution processor module for event-driven vision sensors, IEEE Journal of Solid-State Circuits 47 (2) (2012) 504–517. doi:10.1109/JSSC.2011.2167409.

[25] Z. Lin, J. Shen, D. Ma, J. Meng, Quantisation and pooling method for low-inference-latency spiking neural networks, Electronics Letters 53 (20) (2017) 1347–1348. doi:10.1049/el.2017.2219.

[26] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, M. Pfeiffer, Real-time classification and sensor fusion with a spiking deep belief network, Frontiers in Neuroscience 7. doi:10.3389/fnins.2013.00178.

[27] S. R. Lehky, Decoding poisson spike trains by gaussian filtering, Neural Computation 22 (5) (2010) 1245–1271. doi:10.1162/neco.2009.07-08-823.

[28] A. P. Johnson, J. Liu, A. Millard, et al, Homeostatic fault tolerance in spiking neural networks: A dynamic hardware perspective, IEEE Transactions on Circuits and Systems-I: Regular Papers (2017) 1–13doi:10.1109/TCSI.2017.2726763.