Self-Evolutionary Neuron Model for Fast-Response Spiking Neural Networks

Anguo Zhang, Yuzhen Niu, Yueming Gao*, Ying Han, Qing Chen, Wei Zhu

***Abstract*—In this paper, we proposed simple but effective spiking neuron models for improving the fast-response ability of spiking neural networks (SNNs). The proposed neuron models can adaptively tune the presynaptic input current according to the input received from its presynapses and the firing events of itself. Experimental results of spiking feedforward neural network (SFNN) and spiking convolutional neural network (SCNN) on both MNIST handwritten digits and Fashion-MNIST classification tasks showed that compared with the plain SNNs, the proposed neuron models based networks significantly accelerate the response speed to input signal. Besides the experiments, theoretical analysis about homeostatic state convergence ability of firing activity has been presented to illustrate the reliability of our proposed methods. Experiment codes are available on https://github.com/anvien/Evol-SNN.**

***Index Terms*—Spiking Neural Network, Synaptic Plasticity, Fast Response Speed, Information Maximization.**

I. Introduction

ARTIFICIAL intelligence, especially artificial neural networks (ANNs), have been an attractive topic in recent years. Conventional ANNs, also namely rate-based neural networks, has a significant disadvantage in terms of huge energy consumption even though they have been extensively applied in various fields, and it limits the use of ANNs for embedded devices due to the limited energy storage. In many ways that try to solve this problem, spiking neural networks (SNNs) are considered to be energy-friendly at the chip hardware level, and plenty of works have been done on embedding the SNNs on chips like GPU, FPGA, VLSI, etc [1]–[6]. SNNs, the third generation of neural networks, are firstly proposed by W. Maass [7]. Unlike the rate-based neural networks, the spiking neuron will continuously change its membrane potential according to the external input or its own neuronal state. And when the membrane potential exceeds a certain threshold, a spike signal will be generated to transmit to the post-synaptic neuron through the axon, thus complete the information transfer. [8] has proven that networks of spiking neurons can simulate arbitrary feed-forward sigmoidal neural networks and can thus approximate any smooth nonlinear function. Further, in [9], it has been demonstrated that spiking neurons which encode information by individual spike times are computationally more powerful than neurons with sigmoidal activation function.

Nowadays, SNNs are regarded as an effective computing toolkit which has been used widely, such as object recognition [10], [11], image classification [12], series data process [13]–[15], and so on.

High input firing rate is beneficial to the rapid response ability of SNNs due to that more synaptic current may be input to SNNs at a single time step. However, high input rate also leads to high computation power cost and memory access [16]–[18]. On the other hand, firing threshold is another important factor that directly influences the network response speed to the input signal. If high firing threshold, neuron need to collect much excitatory current to boost its membrane potential to generate a spike signal, thus, the neuron takes more time to respond to the input. Further, as demonstrated in [18], there is a trade-off between the response speed and classification accuracy by tuning the value of firing threshold. High firing threshold helps to improve the accuracy but decreases the response speed, while low firing threshold ensures low latency after spike signal input to the network but may lead to a relatively low accuracy. So, we find that simply tuning the input firing rate or neuronal firing threshold may be no longer a viable or effective way to optimize an SNN with respect to both computational accuracy and response speed.

Some contributions have been taken to solve this problem. [18] proposed a weight normalization method named "model-based normalization" and "data-based normalization" to help regulate the firing rates of spiking feed-forward networks and spiking convolutional networks, and their pattern recognition experiments showed that the normalization technique boosts the convergence speed of the firing activity of spiking neurons, which improves the real-time performance. In [19] the authors proposed an information transmission method with burst spikes and a layer-wise hybrid neural coding scheme for deep SNNs, and their experiment results of image classification tasks proved that the proposed method substantially improves the inference efficiency in terms of speed and energy, while also maintains reasonable accuracy. In [20], Zambrano et al. presented an adaptive spiking neurons based network, where the neurons encode information in spike-trains using a form of Asynchronous Pulsed Sigma-Delta coding, and the authors demonstrated that the proposed neuron models based

Anguo Zhang is with the College of Physics and information Engineering, Fuzhou University, Fuzhou 350108, China and the Research Institute of Ruijie, Ruijie Networks Co., Ltd, Fuzhou 350002, China.

Yuzhen Niu is with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China. (Corresponding author: yuzhenniu@gmail.com)

Yueming Gao is with the College of Physics and information Engineering, Fuzhou University, Fuzhou 350108, China.

Ying Han is with the School of Public Health, Xiamen University, Xiamen 361102, China.

Qing Chen is with the College of Automation, Chongqing University, Chongqing 400030, China.

Wei Zhu is with the Research Institute of Ruijie, Ruijie Networks Co., Ltd, Fuzhou 350002, China.

Manuscript received April 19, 2019; revised August 26, 2019.

network responds an order of magnitude faster and uses an order of magnitude fewer spikes. In the structural plasticity mechanism which proposed in [21] that demonstrated this plasticity improves the learning speed of SNNs, the authors also took the neural conductance into account, and designed the conductance variation models for excitatory neurons and inhibitory neurons, respectively. In [21], the conductance variation models are defined as negative exponential relationships with time only, and the models are nonlinearly fused the spiking neuron model (leaky integrate-and-fire model, LIF model).

However, both the "model-based normalization" and "data-based normalization" methods proposed by [18] have to use global information of network layers to normalize the input weights of each neuron. The information transmission method and layer-wise neural coding scheme proposed in [19] need to be well designed and carefully implemented for a relative desired performance. In [20], the authors consider the firing threshold plasticity based adaptive spiking neuron model, but as the paper demonstrated, the presented networks are specific in straightforward neural networks without additional operations like pooling, softmax, etc., specific SNN variations need to be further developed. On the other hand, the structural plasticity proposed in [21] is to consider the network performance optimization problem from the aspect of connection creation or elimination between neurons, rather than the plasticity of some factors or variables of the neurons themselves.

In this paper, we proposed effective but very simple optimization methods for fast-response SNNs, two integrate-and fire (IF) neuron based novel models, namely evolutionary spiking neuron (Evol) and adaptive evolutionary spiking neuron (Adap-Evol), are used in our optimization methods. The methods can be implemented asynchronously and locally in computational, where the "asynchronously" denotes event-triggered computation mechanism of spiking neurons, and the "locally" means every neuron updates its state by only using its own presynaptic input current and its own neuronal dynamics. Based on these advantages, the proposed neuron optimization methods can be easily applied to engineering without any external complex implementation, and can also be distributed deployed with low computation cost. Series of experiment results on both MNIST and Fashion-MNIST datasets prove the effectiveness of our proposed methods.

The rest of this paper is structured as follows. Sec. II introduces our proposed spiking neuron models with Evol and Adap-Evol mechanisms. Sec. IV and Sec. V describe the test benchmarks and network model construction procedure, as well as the experimental results, respectively. Finally, we draw a conclusion in Sec. VI to end this paper.

II. Adaptive Presynaptic Input Current

A. Spiking Neuron Model

Spiking neural networks are considered to be the third generation of artificial neural networks, have shown their outstanding advantages in terms of low computational cost, high computational power, and rich neural plasticity [7]–[9], [22], [23]. In the conventional ANNs, input data is fed into the network at one time, then processed layer by layer, and finally, the network produces the results from the output layer. However, in the computing framework of SNNs, the raw input data is first converted into spike streams of event signals, neurons record these signals and create spikes to communicate information with other connected neurons, ultimately, the output layer neurons collect the firing evidence driven by the incoming information and make decisions over time.

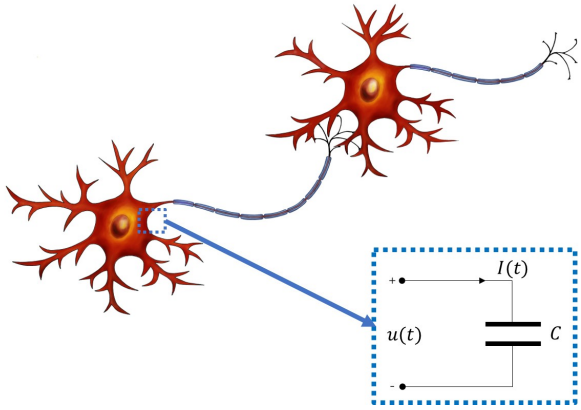


Fig. 1: The membrane of biological neuron is mainly a semi-permeable biofilm composed of phospholipid molecules. Except the protein channels of specific ions, the neuronal membrane is non-conductive and its physical properties are similar to parallel-plate capacitor. A simple schematic diagram of parallel-plate capacitor, where $u(t)$ represents the voltage across the capacitor, $I(t)$ denotes the current flowing to the capacitor, and C is the capacitance of the capacitor.

Spiking neuron models mimic some important physiological properties of biological neurons, such as membrane potential v, membrane resting potential v_{rest}, membrane firing potential threshold v_{thr}, membrane resistance R_m and capacitance C_m, refractory period t_{ref}, etc.. When the membrane potential v exceeds the membrane firing potential threshold, the neuron generates a current spike and transmit it to the dendrites of other neurons, meanwhile, the membrane potential immediately drops to the membrane resting potential and the neuron enters the refractory period. During the refractory period, the neuron does not receive any external current signal from its synapse.

B. Proposed Method

Fig.1 shows a simple schematic diagram of a parallel-plate capacitor, its physical model can be written by

$$U(t) = \frac{1}{C}\int I(t)dt \tag{1}$$

where C denotes the capacity. A well-known physical fact is that when the distance between the two plates of a parallel-plate capacitor increases or decreases, the capacitance of the capacitor is changed, therefore, the capacitor can be adaptively discharged or charged by dynamically changing its capacitance

value according to a certain demand. Considering this case, a the derivative of $U(t)$ is derived as

$$\dot{U}(t) = \frac{1}{C(t)} I(t) \tag{2}$$

where $C(t)$ turns to be a dynamically changing capacitance value over time t.

Among different kinds of spiking neuron models, the leaky integrate-and-fire (LIF) neuron may have a similar mathematical formulation to Eq.(2), which is simply given by

$$\frac{dv(t)}{dt} = -v(t) + \sum_{i}^{\Omega} \omega_i I_i(t) \tag{3}$$

where Ω denotes the set of presynapses of the I&F neuron, $v(t)$ denotes the membrane potential (voltage) of the neuron, $w_i, i \in \Omega$ is the connection weight between the neuron and its presynapse i, $I_i(t)$ is the synaptic current that input from its presynapses. It should be noted that Eq.(3) can be regarded as a special case of

$$\frac{dv(t)}{dt} = -v(t) + \frac{1}{C(t)} \sum_{i}^{\Omega} \omega_i I_i(t) \tag{4}$$

if $C(t)$ is set fixedly to be 1.

In this paper, we use the extended I&F model that presented by Eq.(4) as the basic neurons of SNNs.

Define

$$R(t) = \Theta(t - \hat{t} - t_{ref}) \tag{5}$$

where $\Theta(\cdot)$ denotes a Heaviside function that takes a value of one for positive arguments and vanishes otherwise. $\hat{t}$ is the last firing time of the neuron, and t_{ref} is the refractory period.

Let the membrane conductance $E(t) = \frac{1}{C(t)}$, and the total external input current $I(t) = \sum_i^{\Omega} \omega_i I_i(t)$, where $E(t)$ is constrained to be greater than 0, we design the following evolutionary tuning law for neuron Eq.(4):

$$\begin{aligned} \frac{dE(t)}{dt} &= \nabla E(t) \\ &= \eta \left(\frac{\epsilon}{E(t)} + \frac{I(t)R(t)}{v_{thr} - v_{rest}} \big(\gamma - (1+\gamma) O(t)\big) \right) \end{aligned} \tag{6}$$

where η is the update rate, ϵ represents a positive proportion factor, γ is a positive design constant, and $O(t)$ is the spike output in response to the total presynaptic input current which denoted by an impulsive function as

$$O(t) = \sum_{f} \delta\left(t - t^{(f)}\right) \tag{7}$$

where $t^{(f)}$ denotes the firing time that spike signal generated by the neuron, $\delta(\cdot)$ is a Dirac-Delta function meaning that $\delta\left(t - t^{(f)}\right) = 1$ if $t = t^{(f)}$, otherwise $\delta\left(t - t^{(f)}\right) = 0$. η indicates the strength of the generated spike signal.

In Eq.(6), the scale factor ϵ is set to be a fixed positive value, however, we can observe that a higher ϵ value can effectively increase the rate of change of $E(t)$, and vice versa. Therefore, we try to design a time-varying ϵ by

$$\epsilon(t) = 1 - \frac{1}{1 + exp(-\frac{2t}{\tau} + 1)} \tag{8}$$

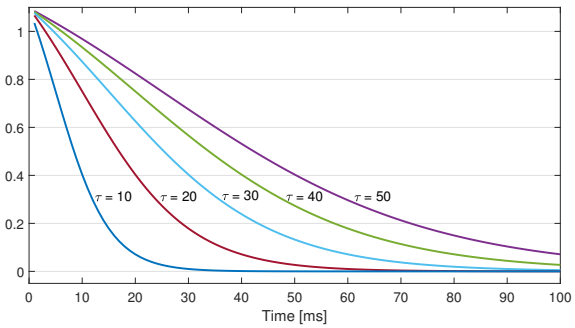


Fig. 2: The variation of $\epsilon(t)$ with time under different time constants τ.

where t is the relative time since the new signal is inputing to SNN, τ denotes the time constant. Under different time constants of τ, the variation of $\epsilon(t)$ with time is shown in Fig. 2.

The method by which we use Eq.(6) to regulate the neuron model Eq.(4) independently is named "Evol", while the neuron regulation method using both Eq.(6) and Eq.(8) is named "Adap-Evol". The description of the "Evol" and "Adap-Evol" methods have been presented in Algorithm 1 and Algorithm 2.

Algorithm 1: Algorithm of evolutionary spiking neuron model (Evol)

Input: Inference Images $\mathcal{X} = \{x_i\}_{i=1}^n$.
Output: Predicted label $C = \{c\}_{i=1}^n$ for each image.
Setting the parameters k, β and η, initialize C;
for *time* t ***in*** $[1, 2, \ldots, T]$ **do**
  for *layer* ***in*** *layers* **do**
    for *neuron* ***in*** *layer.neurons* **do**
      Calculate input current $I(t) = \sum_i^{\Omega} \omega_i I_i(t)$;
      Calculate $\nabla v(t)$ by Eq.(4);
      Update the membrane potential $v(t) = v(t-1) + \nabla v(t)$;
      Calculate $\nabla E(t)$ by Eq.(6);
      Update $E(t) = E(t-1) + \nabla E(t)$, $C(t) = \frac{1}{E(t)}$;
    end
  end
  Collect the spike signal from the neurons of output layer;
  Obtain the classification result at time t.
end

III. Theoretical Analysis

A. Mutual Information Maximum between Input Current and Output Spike

The firing probability ρ at time t is a function related to the membrane potential v and refractory state R,

$$\rho(t) = 1 - \exp\left(- \frac{v(t) - v_{rest}}{v_{thr} - v_{rest}} R(t) \right) \tag{9}$$

Algorithm 2: Algorithm of adaptive evolutionary spiking neuron model (Adap-Evol)

Input: Inference Images $\mathcal{X} = \{x_i\}_{i=1}^n$.
Output: Predicted label $C = \{c\}_{i=1}^n$ for each image.
Setting the parameters k, β, η and the time constant τ, initialize C;
for *time* t ***in*** $[1, 2, \ldots, T]$ **do**
  for *layer* ***in*** *layers* **do**
    for *neuron* ***in*** *layer.neurons* **do**
      Calculate input current $I(t) = \sum_i^{\Omega} \omega_i I_i(t)$;
      Calculate $\nabla v(t)$ by Eq.(4);
      Update the membrane potential $v(t) = v(t-1) + \nabla v(t)$;
      Update $\epsilon(t)$ by Eq.(8);
      Calculate $\nabla E(t)$ by Eq.(6);
      Update $E(t) = E(t-1) + \nabla E(t)$, $C(t) = \frac{1}{E(t)}$;
    end
  end
  Collect the spike signal from the neurons of output layer;
  Obtain the classification result at time t.
end

where $v_{thr} - v_{rest} > 0$.

Thus, we have the following derivative of ρ with respect to v,

$$\frac{\partial \rho}{\partial v} = \frac{R(t)}{v_{thr} - v_{rest}}\big(1-\rho\big) \tag{10}$$

The probability density function of the two-parameter Weibull distribution is

$$f_{Weib}(y;\alpha,k) = \frac{k}{\alpha}\big(\frac{y}{\alpha}\big)^{k-1} exp\left(-(y/\alpha)^k\right) \tag{11}$$

The Kullback-Leiber divergence (KLD) of the firing probability density functions $f_\rho(\rho)$ and the desired Weibull-like distribution function $f_{Weib}(\rho;\alpha,k)$ can be calculated by

$$\begin{aligned} D &= d_{KL}(f_\rho || f_{Weib}) \\ &= \int f_\rho(\rho)\log(f_\rho(\rho))d\rho - \ln(\frac{k}{\alpha^k}) \\ &\quad -(k-1)\int f_\rho(\rho)\ln(\rho)d\rho + \frac{1}{\alpha^k}\int f_\rho(\rho)\rho^k d\rho \\ &= -H(\rho) + \frac{1}{\alpha^k}\Xi(\rho^k) - (k-1)\Xi(\ln(\rho)) \\ &\quad -\ln(\frac{k}{\alpha^k}) \end{aligned} \tag{12}$$

where $\Xi(\cdot)$ denotes the mathematical expectation.

According to the neuronal dynamics of Eq.(4), we have

$$\begin{aligned} \frac{\partial v}{\partial E} &= I \\ \frac{\partial v}{\partial t} &= -v + EI \\ \frac{\partial E}{\partial t} &= \frac{1}{\partial v/\partial E}\cdot\frac{\partial v}{\partial t} = -\frac{1}{I}(v - EI) \end{aligned} \tag{13}$$

So the derivative of ρ respect to the membrane conductance$E$ is

$$\begin{aligned} \frac{\partial \rho}{\partial E} &= \frac{\partial \rho}{\partial v}\cdot\frac{\partial v}{\partial t}\cdot\frac{1}{\partial E/\partial t} \\ &= \frac{R}{v_{thr}-v_{rest}}\big(1-\rho\big)(-v+EI)\frac{-I}{v-EI} \\ &= \frac{IR}{v_{thr}-v_{rest}}\big(1-\rho\big) \end{aligned} \tag{14}$$

Considering that [24]

$$\begin{aligned} H(y) &= -\int f_\rho(\rho)\log(f_\rho(\rho))d\rho \\ &= \Xi\Big(\ln\big(\frac{\partial\rho}{\partial v}\big)\Big) - \Xi\Big(\ln\big(f_v(v)\big)\Big) \end{aligned} \tag{15}$$

We calculate the derivative of the KLD with respect to the variable of E,

$$\begin{aligned} \frac{\partial D}{\partial E} &= -\frac{1}{E} + \Xi\left[\left(\frac{1}{1-\rho} + \frac{k}{\alpha^k}\rho^{k-1} - \frac{k-1}{\rho}\right)\frac{\partial\rho}{\partial E}\right] \\ &= -\frac{1}{E} + \Xi\left[\frac{IR}{v_{thr}-v_{rest}}\Big(1 + (1-\rho) \right. \\ &\quad \left. \big(\frac{k}{\alpha^k}\rho^{k-1} - \frac{k-1}{\rho}\big)\Big)\right] \end{aligned} \tag{16}$$

By setting $k = 1$ and defining $\gamma = -1 - \frac{1}{\alpha}$, Eq.(16) can be simplified by

$$\begin{aligned} \frac{\partial D}{\partial E} &= -\frac{1}{E} + \frac{1}{v_{thr}-v_{rest}}\Xi\Big[IR\Big(-\gamma + (\gamma+1)\rho\Big)\Big] \\ &= -\frac{1}{E} - \frac{1}{v_{thr}-v_{rest}}\Xi\Big[IR\Big(\gamma - (\gamma+1)\rho\Big)\Big] \end{aligned} \tag{17}$$

Finally, gradient descent learning rule for adjusting E is given by

$$\begin{aligned} E &\leftarrow E - \Delta E = E - \eta\left(\frac{\partial D}{\partial E}\right) \\ &= E + \eta\left(\frac{\epsilon}{E} + \frac{IR}{v_{thr}-v_{rest}}\big(\gamma - (\gamma+1)\rho\big)\right) \end{aligned} \tag{18}$$

where η is the learning rate, the parameter ϵ can be set as 1 or other positive values for regulation, in this paper, ϵ is set to 1 for ''Evol" mechanism, and set to a time-varying value for "Adap-Evol" mechanism. Noting that if the refractory period t_{ref} is set to be 0, $R(t)$ will be fixed as 1, thus Eq.(18) can be rewritten by

$$E \leftarrow E + \eta\left(\frac{\epsilon}{E} + \frac{I}{v_{thr}-v_{rest}}\big(\gamma - (\gamma+1)\rho\big)\right) \tag{19}$$

On the other hand, the firing probability $\rho(t)$ is a continuous variable which takes a value between 0 and 1, we simply substitute it by using the firing output $O(t)$ that defined in Eq.(7), thus we have

$$E \leftarrow E + \eta\left(\frac{\epsilon}{E} + \frac{IR}{v_{thr}-v_{rest}}\big(\gamma - (\gamma+1)O\big)\right) \tag{20}$$

which presented the proposed "Evol" mechanism proposed by Eq.(6).

Combining Eq.(20) with time-varying $\epsilon(t)$ of Eq.(8), we get the proposed "Adap-Evol" mechanism that mentioned before.

Through the self-evolutionary rules of the membrane conductance E derived above, we find that the mutual information entropy between the firing event of the spiking neuron and the input current can be maximized by adjusting E at a suitable learning rate η. So the mutual information entropy between the input current and output spikes of the whole spiking neural networks is also maximized since that all neurons are self-evolving.

B. Convergence of Homeostatic State

According to Eq.(4), we can know that ignoring the change of the presynaptic current input, the larger the absolute value of $C(t)$, the smaller the change of $V_{mem}(t)$. In Eq.(6), under the setting of a positive initial value of $E(t)$, two extreme cases are considered. One is the neuron output $O(t) = 1$, which means that the neuron fires at each discrete time step. The other one is $O(t) = 0$, which means that the neuron stay inactive at all discrete time steps.

Case 1: In this case, $O(t) = 1$, due to the fact that the presynaptic input current $I_i(t), i \in \Omega$ is either zero or positive, we have $I(t) \geq 0$, thus $(\gamma - (1+\gamma)O(t))\, I(t)R(t) \leq 0$. By Eq.(6), it derives

$$\nabla E(t) > 0, \text{ if } \quad E(t) < \frac{\epsilon(v_{thr} - v_{rest})}{(-\gamma + (1+\gamma)O(t))\, I(t)R(t) + \sigma}$$

$$\nabla E(t) < 0, \text{ if } \quad E(t) > \frac{\epsilon(v_{thr} - v_{rest})}{(-\gamma + (1+\gamma)O(t))\, I(t)R(t) + \sigma}$$

with the positive value of parameter γ, and the σ in the denominator is a sufficiently small positive number to prevent the denominator from being 0. So, in this case, $E(t)$ will converge to (or track) the value of term $\frac{\epsilon(v_{thr}-v_{rest})}{(-\gamma+(1+\gamma)O(t))I(t)+\sigma}$. Fig. 3 shows the variation of $E(t)$ over time with different values of the presynaptic input current $I(t)$, where Fig.3(a) presents the $E(t)$ curves under constant values of $I(t)$, and where Fig. 3(b) presents the curves under random $I(t)$ whose average values of 0.5, 1 and 2, respectively.

Thus, we can know that if the presynaptic input current of a neuron has a fixed mean, and the neuron keeps firing at each time step of the early stage, it will continue to fire during the subsequent time steps.

Case 2: In this case, $O(t) = 0$, we have $(\gamma - (1+\gamma)O(t))\, I(t)R(t)/(v_{thr} - v_{rest}) \geq 0$, so the gradient $\nabla E(t)$ will be always greater than 0, which indicates that $E(t)$, i.e., $\frac{1}{C(t)}$ is monotonically increases over $[0, +\infty)$. Based on Eq.(4), compared to setting $C(t)$ to a fixed value, the rule of Eq.(6) will increase the membrane potential of neuron more rapidly, causing it to generate firing behavior more quickly, thereby responding to the input signal earlier.

In summary, if a neuron does not produce a spike (i.e, $O(t) = 0$), then the Evol and Adapt-Evol rules increase the sensitivity of the neuron's membrane potential to the input current. When the neuron produces a spike (i.e, $O(t) = 1$), it returns to the discussion of case 1. It should be noted that even if the sensitivity of the neuron membrane potential to the input current increases, it does not mean that the neuron will fire in a short period of time, it is because neuron requires presynaptic input currents that is not zero or too small.

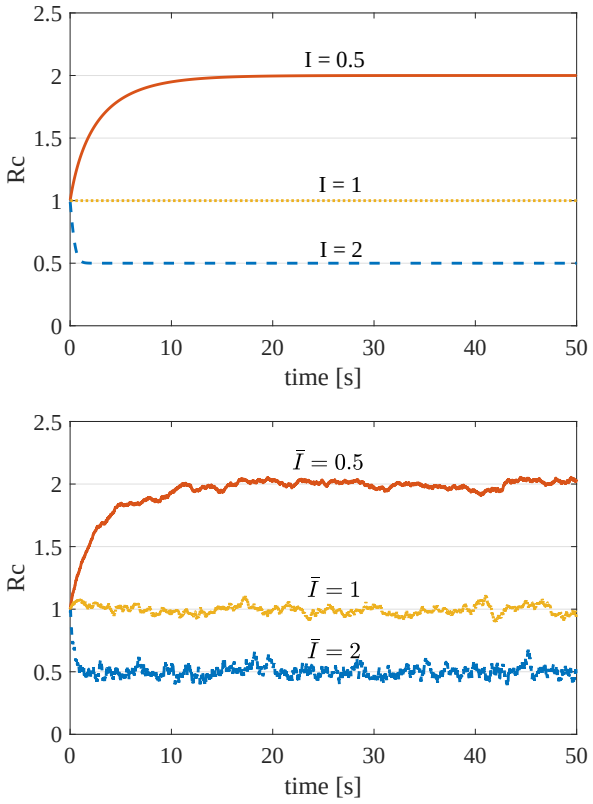


Fig. 3: Variation of $E(t)$ over time with different values of the presynaptic input current $I(t)$.

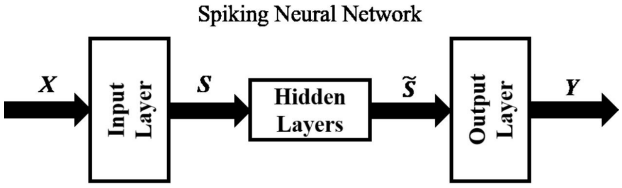


Fig. 4: Information transmission through an SNN network.

C. Convergence of Computational Accuracy

As shown in Fig. 4, one can obtain the output Y under a given input X, if there is no information loss during the phases of $X \to S$ and $\tilde{S} \to Y$. The biggest challenge is how to design a low error rate and high noise tolerance channel $S \to \tilde{S}$. When information transmits from S to $\tilde{S}$, it should be encoded to flow through the channel. In the information communication systems the channel may be physical things like fiber optic or wireless radio, while in the SNNs, the channel is consisted to be the hidden layers of neural networks. So it is feasible to use the information theory to analyze the properties of SNNs.

The mutual information between the variable X and Y is

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{21}$$

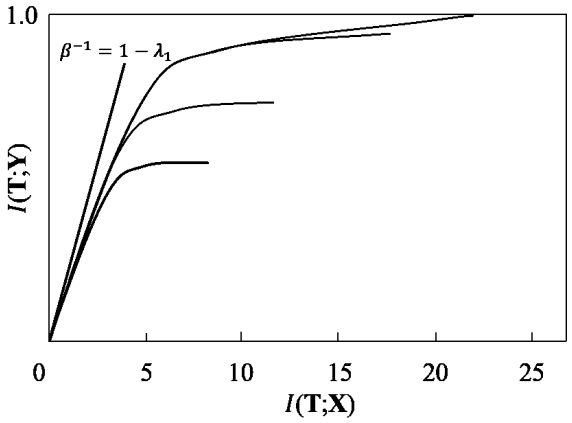


Fig. 5: GIB information curve obtained in [25].

Since T is a compressed representation of X, its distribution completely determined given X alone, that is, $q(T|X, y) = q(T|X)$, or

$$q(X, Y, T) = p(X, Y)q(T|X) \tag{22}$$

Tishby et.al [25] formulated the optimization problem as the minimization of the following IB functional,

$$J(q(T|X)) = I(T; X) - \beta I(T; Y) \tag{23}$$

where β is a positive Lagrange multiplier of the trade-off between the compression (minimal representation) and predictability (information preservation).

For the Gaussian framework problem, Tishby plotted the curve of the mutual information $I(T; X)$ and $I(T; Y)$ under different parameter β, as shown in Fig.5 [25], it can be found that the curve is concave everywhere. At each value of the mutual information $I(T; X)$, the curve is bounded by a tangent with a slope defined by the function $\beta^{-1}(I(T; X))$. At the original point, $I(T; X) = 0$, the slope $\beta^{-1}(0) = 1 - \lambda_1$, where λ_1 is the first eigenvalue of the canonical correlation analysis of the original random vector X and its compressed version Y. It also should be noted that the asymptotic slope of the information curve is zero, which means $\beta \to \infty$. And it simply reflects that the addition of more bits of information to the description of the original random vector X does not provide increased accuracy for the bottleneck vector T.

In SNNs, it takes a certain time period to achieve a relatively reliable output, and at every time step, the source signal is imposed on the input. By the description above, it can be analogized that the addition of more time steps does not increase the network computational accuracy, thus it will converge to the upper bound.

Even there exists a large volume of results demonstrate that SNNs are promising next generation of artificial neural networks, it is still a lack of sufficient theoretical analysis to prove the computational effectiveness of SNNs. Unlike the rate-based neural networks in which the output can be calculated immediately within one-time step if the input signal is given, SNNs take a period of time to get the converged output. Therefore, a very serious question is whether the computation result will converge to a certain value.

D. Speed Acceleration of Inference Learning

Mutual information is a measure of the amount of information that one random variable constants about another random variable. It is the reduction in the uncertainty of one random variable due to the knowledge of the other.

It has been demonstrated that the proposed "Evol" and "Adap-Evol" rules are beneficial to mutual information maximization between S and $\tilde{S}$, on the other hand,

$$\begin{aligned} I(S; \tilde{S}) &= \sum_{x,y} p(s, \tilde{s}) \log \frac{p(s, \tilde{s})}{p(s)p(\tilde{s})} \\ &= \sum_{x,y} p(s, \tilde{s}) \log \frac{p(\tilde{s}|s)}{p(\tilde{s})} \\ &= -\sum_{x,y} p(s, \tilde{s}) \log p(\tilde{s}) + \sum_{x,y} p(s, \tilde{s}) \log p(\tilde{s}|s) \\ &= H(\tilde{S}) - H(\tilde{S}|S) \end{aligned} \tag{24}$$

Thus, it can be obtained that the mutual information $I(S; \tilde{S})$ is the reduction in the uncertainty of $\tilde{S}$ due to the knowledge of S. Therefore, the maximization of $I(S; \tilde{S})$ means the uncertainty of $\tilde{S}$ is reduced. In SNNs, if we want to achieve a relatively stable and high computational result when a source signal is input, it should be guaranteed that the network has sufficiently learned enough feature information from the spiking signals passing through the channel. Maximizing mutual information minimizes the impact of uncertainty on feature information, thus, fewer time steps are required to wash out the uncertainty that the inputs bring to the network, which means less time is cost to learn from the source signals for SNNs.

IV. Material and Methods

A. Test Datasets

We tested the effects of our proposed evolutionary spiking neuron models on the classification performance of SFNN and SCNN on both the MNIST and Fashion-MNIST datasets.

Fashion-MNIST [26] is a benchmarking dataset for machine learning algorithms, which is intended to serve as a direct drop-in replacement for the original MNSIT dataset, and it is also consisting of 60000 training examples and 10000 test examples, where each example is a 28x28 grayscale image associated with a label from 10 classes. The 10 classes contain the label "T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag" and " Ankle boot". Some example images of Fashion-MNIST dataset are shown in Fig. 6.

It should be noted that we use the ANN-SNN conversion method to obtain the network hyperparameter of the expected final SNN, that is, the connection weights. Therefore, the pixel values of the image can be directly input into the corresponding ANN after being normalized. However, SNN can only receive discrete spike signals as network inputs, thus the image data needs to be first converted into spike signals by certain rules. In this paper, Poisson-distributed spike encoding [18], [27] is used to produce spikes for SNN input.

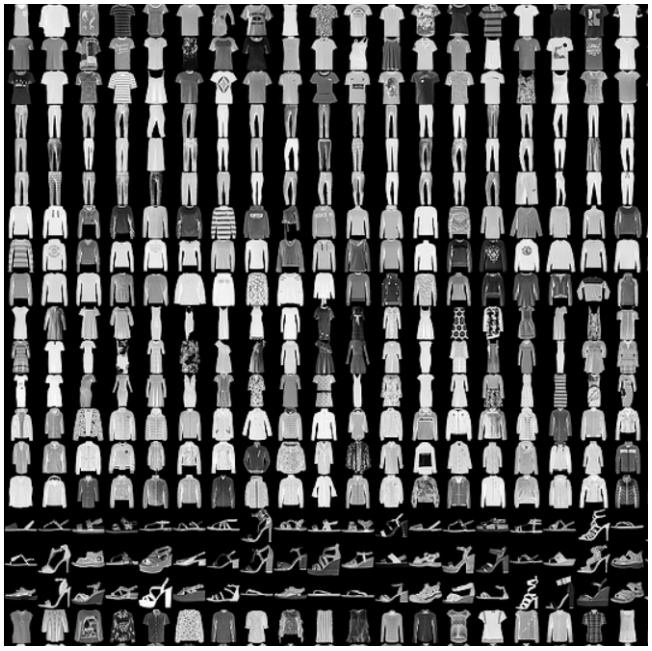


Fig. 6: Example images of fashion mnist dataset.

B. Spiking Networks with Evolutionary Neuron Model

We constructed two typical neural networks, namely feed-forward neural networks (FNNs) and convolutional neural networks (CNNs), as the fundamental computing frameworks for our proposed evolutionary neuron models. The two types of networks, especially convolutional neural networks, have been widely used in image classification tasks, so not only the conventional rate-based FNNs and CNNs, but also the spike event-based FNNs and CNNs, have attracted lots of research interest over the world.

However, direct training a spike event-based network (spiking neural network) can be complex and difficult due to the undifferentiable nature of the discrete and asynchronous neuronal spike events. Therefore, [16]–[18], [28], [29] studied the probability of converting the trained ANN to the SNN counterpart, and proposed some conversion methods for network computing operations such as convolution, average pooling, max pooling, fully-connection propagation, ReLU activation function, softmax activation function, etc..

In this paper, we also use the ANN-to-SNN conversion method to get the fundamental network weights, then replace the rate-based neural activation function by the proposed evolutionary and adaptive evolutionary spiking neuron models. Similar to [18], the detail steps of constructing the SFNN and SCNN are:

1) Implement FNN and CNN with the activation function of ReLU, and set no bias;

2) Train with error backpropagation algorithm;

3) Directly map the trained ANN weights to the corresponding SNNs with the same structure, and set the SNN neurons to the integrate-and-fire (IF) or our proposed evolutionary neuron models.

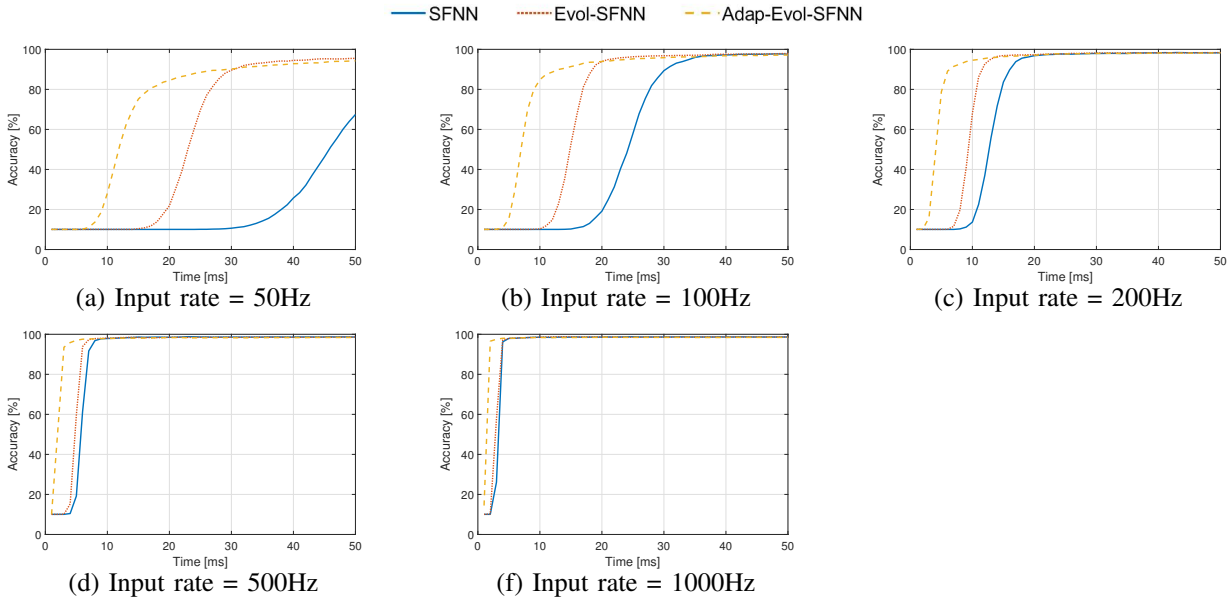
To prove the efficacy of our evolutionary neuron models, we used two main network architectures. For the feed-forward neural network, we construct a 784-1200-1200-10 network, it means that the network is with the input dimension of 784 (28x28, same with the pixel number of input image data) and the output dimension of 10 (same with the number of image classes), further, the network has two hidden layers with 1200 neurons per layer. For the convolutional neural network, we construct a 28x28-12c5-2s-64c5-2s-10o network, it means that 12 5x5 convolutional kernels follow the input layer, and the generated feature maps continue to be convoluted by 64 5x5 convolutional kernels after 2x2 average pooling, and then again by the 2x2 average pooling. The resulting 64 feature maps are flattened and fully linked to the final 10 output neurons. We performed simple training on FNN and CNN models on the MNIST and Fashion MNIST datasets respectively. FNN achieved 98.84% and 90.75% test accuracy, respectively, and CNN achieved 99.14% and 91.87% test accuracy. In addition, what we are required to point out is that whether these test accuracies are state-of-the-art does not affect our experimental conclusions.

V. Experiment Results

During the inference process of SNN to perform image classification tasks, the spike streams generated by original image are continuously input to the network input layer, and then the neurons in each layer of SNN may generate spike signal to the neurons in the subsequent layer, output layer collects the spike response to determine the class of the original image, after that, the classification task is complete. It is worth noting that the network spike response collected by the output layer may be different at different time steps, the classification accuracy presented by SNN may change with time. Only when the neuronal state of SNN achieves homeostatic, the accuracy will eventually converge. Thus, as demonstrated in [30], not only the classification accuracy, but also the response time (latency) and the total number of synaptic events (spikes) are the main concerns when evaluating the recognition performance of an SNN. In this paper, we tested these performance indexes of SFNN/Evol-SFNN/Adap-Evol-SFNN and SCNN/Evol-SCNN/Adap-Evol-SCNN on both MNIST and Fashion MNIST datasets.

A. Results on MNIST dataset

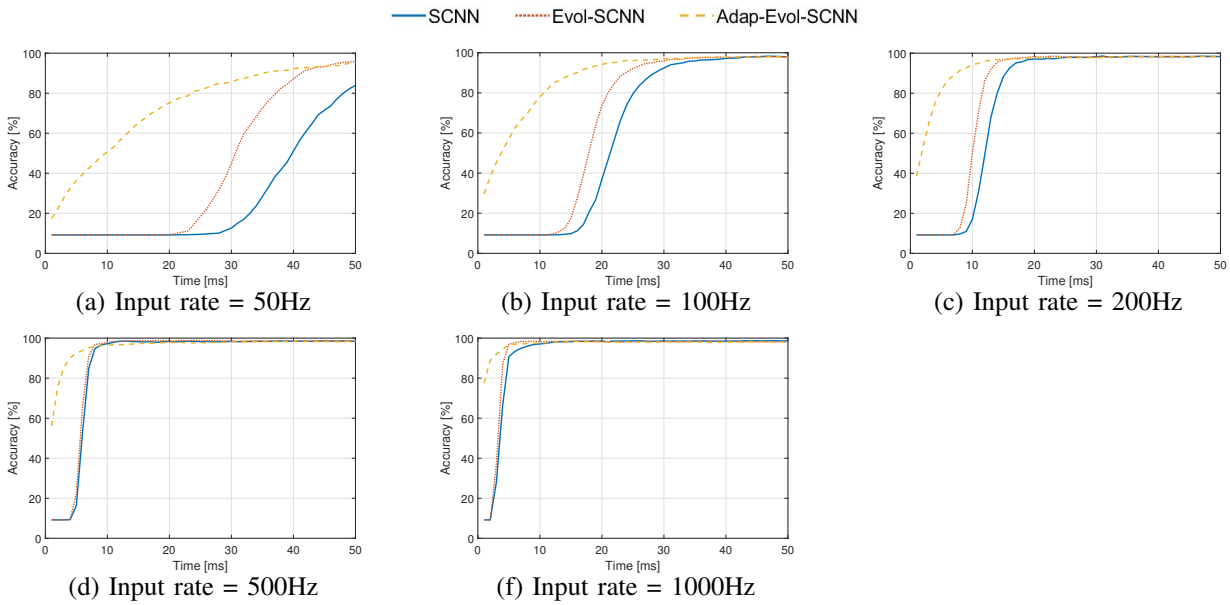
1) Classification Accuracy and Convergence Time: Fig.7 and Fig.8 show the classification accuracy of SFNN/Evol-SFNN/Adap-Evol-SFNN and SCNN/Evol-SCNN/Adap-Evol-SCNN over time, respectively. The running time step is set as 1 ms, we can see that the error rates of classification accuracy are approximately 90% at the initial state, after a short time to collect the output signal of spikes from the output layer, the error rates drop rapidly, and eventually stabilizes at a relatively small value. The stable classification error rate means the neurons of SNN converge to a homeostatic state. The faster convergence speed indicates that the network takes less time to reaching the homeostatic state and produces a stable output. It is beneficial to the low-latency performance of spiking networks for real-time tasks.



(a) Input rate = 50Hz (b) Input rate = 100Hz (c) Input rate = 200Hz

(d) Input rate = 500Hz (f) Input rate = 1000Hz

Fig. 7: Classification accuracy comparison among SFNN, Evol-SFNN and Adap-Evol-SFNN over time. Performance curves of networks are tested under different input firing rates, include 50Hz, 100Hz, 200Hz, 500Hz and 1000Hz, respectively.



(a) Input rate = 50Hz (b) Input rate = 100Hz (c) Input rate = 200Hz

(d) Input rate = 500Hz (f) Input rate = 1000Hz

Fig. 8: Classification accuracy comparison among SCNN, Evol-SCNN and Adap-Evol-SCNN over time. Performance curves of networks are tested under different input firing rates, include 50Hz, 100Hz, 200Hz, 500Hz and 1000Hz, respectively.

Each of the subgraphs in Fig.7 and Fig.8 compares the classification accuracy of the SFNN or SCNN with the same input frequency. It can be easily seen that both Evol and Adap-Evol mechanisms can effectively improve the response speed of the network while ensuring the classification accuracy is not reduced, whether it is low input rate (50 Hz) or high input rate (1000 Hz). , shorten the response time to the input. Compared with Evol, Adapt-Evol is able to significantly improve the network's response ability, especially at low input rate. In addition, it should be noted that in the spiking convolutional neural networks, even at the initial time step, the classification accuracy of the Adap-Evol-SCNN has exceeded 10%, which indicates that for some input signals (spike signals obtained by Poisson sampling), Adap-Evol-SCNN has been able to achieve zero-latency response.

2) Firing Activity of Neurons: Fig.9 and Fig.10 show the statistics of the firing activity of neurons in the main network layers of SFNN and SCNN during the process of recognizing an image. It should be noted that for each network, we only count the firing activity of the network from the initial reception of the input signals to the time when the classification accuracy begins to converge. Therefore, the statistical time

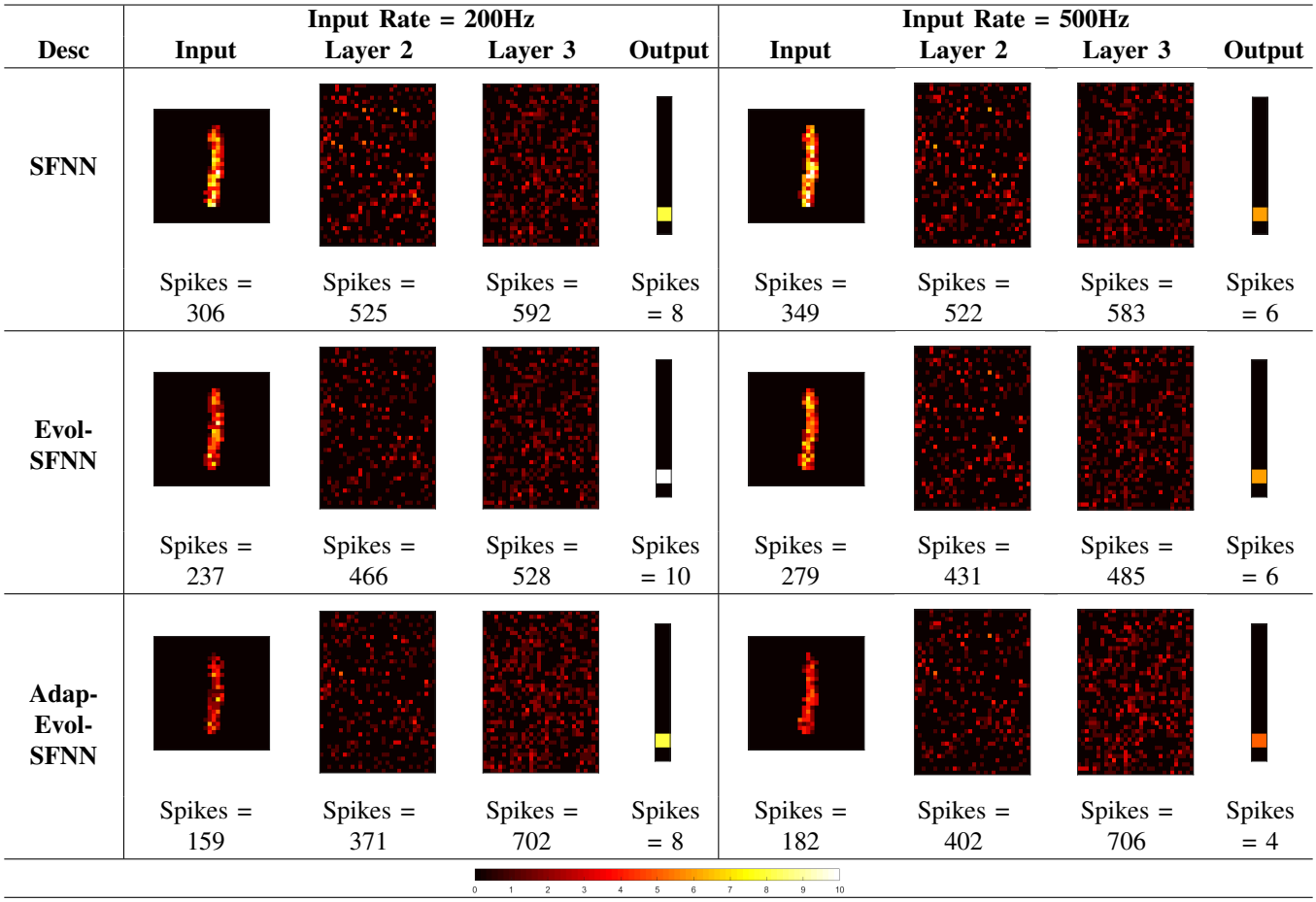


Fig. 9: Firing activities of SFNN, Evol-SFNN and Adap-Evol-SFNN, where the "Layer 2" and "Layer 3" note the 1st and 2nd hidden layer, respectively.

may be different for different networks, for example, SFNN and SCNN have the longest statistical time, Evol-SFNN and Evol-SCNN are slightly shorter, and Adap-Evol-SFNN and Adap-Evol-SCNN are with the shortest statistical time.

The firing activity means how many spike events generated during a certain period of time. At the same input rate, the spike densities of input signal are set to be approximately the same, however, the input spikes amount in the "Input" column of Fig.9 and Fig.10 is different because of the different statistical time length, Apap-Evol-SFNN and Adapt-Evol-SCNN having the least amount of spikes, while SFNN and SCNN are the most.

However, in Fig.9, the spike events of inner-layer neurons of spiking feed-forward networks do not show a significant difference. We can see that for the input rate of both 200Hz and 500Hz, no matter SFNN, or Evol-SFNN and Adap-Evol-SFNN, the summed numbers of spike events of "Layer 2", "Layer 3" and "Output" are approximative, where those of Evol-SFNN may be a few lower than the other two, comparatively. Further, an interesting result is that compared to SFNN and Evol-SFNN, Adap-Evol-SFNN inhibits the firing activity of the previous hidden layer (Layer 2), but enhances the firing activity of the latter hidden layer (Layer 3). This phenomenon brings an advantage that more spike events directly stimulate the neurons of output layer, thus, the output neurons can respond faster to the input signals.

As showed in Fig.10, in spiking convolutional networks, the firing activities of SCNN, Evol-SCNN and Adap-Evol-SCNN are similar to the spiking feed-forward networks mentioned above, i.e., the number of neuronal spike events in the inner layers of Evol-SCNN is smaller than in the SCNN. On the other hand, Adap-Evol-SCNN seems to "move" the spike events of the previous layer to the next layer, in order to speed up the input response of the output neurons.

B. Results on Fashion-MNIST dataset

We quantitatively evaluate the performance of SNN, Evol-SNN and Adap-Evol-SNN on the Fashion-MNIST dataset, and similar with [20], we note the simulation time to which 101% of the minimum classification error is reached (Matching Time, MT). Further, the final classification accuracy (final accuracy, FA) is also computed, where FA represents the stable precision when an SNN reaches homeostatic state. The lower value of MT means that the SNN produces a relatively reliable output faster, which is beneficial to the real-time performance of SNN. While the higher value of FA means that the SNN produces a more accurate output, which is beneficial to the accuracy performance of SNN.

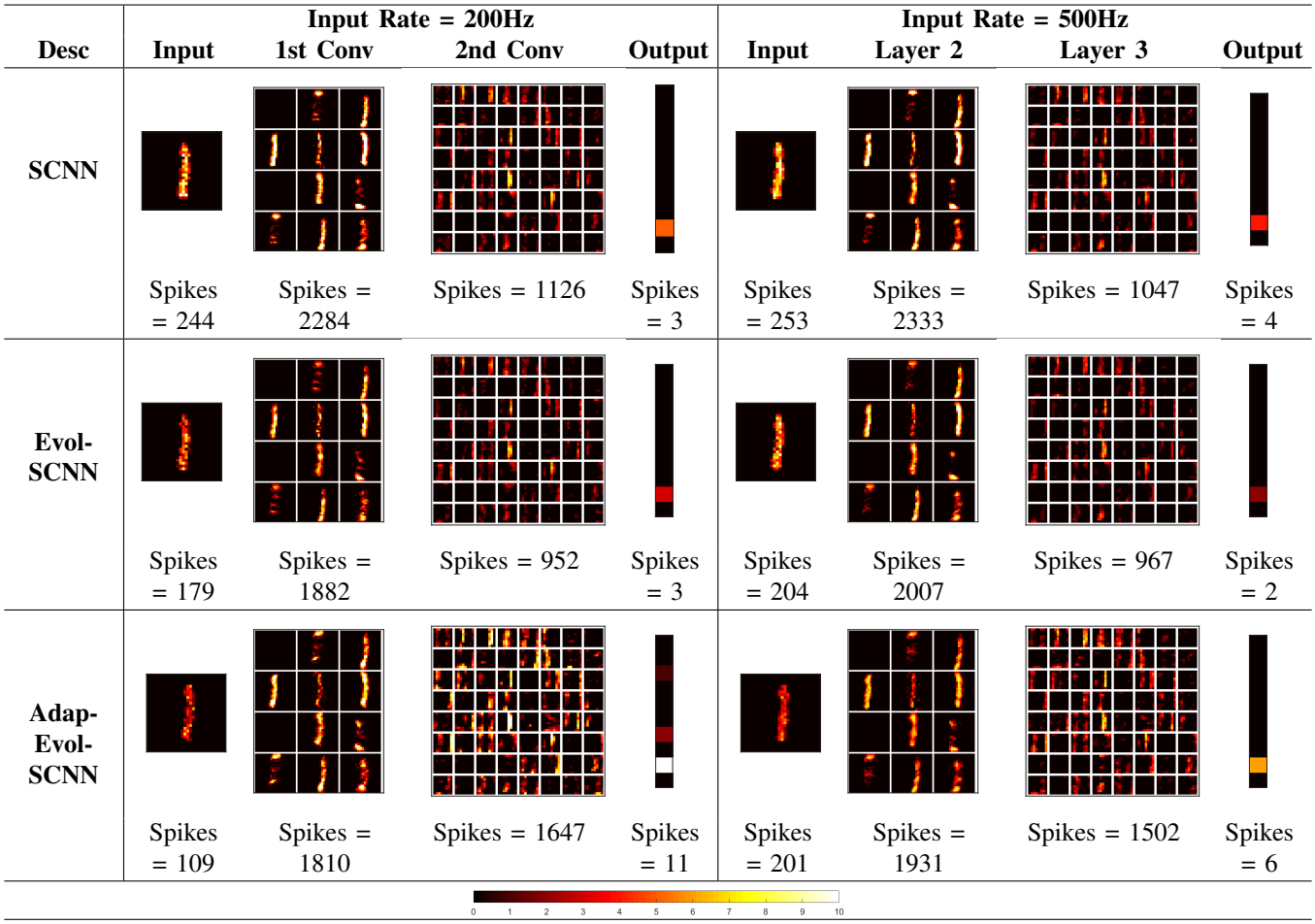


Fig. 10: Firing activities of SCNN, Evol-SCNN, and Adap-Evol-SCNN, where the "1st Conv" and "2nd Conv" denote the 1st and 2nd convolutional layers of SCNN, respectively.

In TAB.I, we present the performance (FA, MT) comparison between SNN, Evol-SNN and Adap-Evol-SNN under different input firing rates. Both the two network models, spiking feed-forward neural network and spiking convolutional neural networks, are taken into account for experiment test.

From TAB.I, we can see that for all values of input firing rate, Adap-Evol-SNN, i.e., Adap-Evol-SFNN and Adap-Evol-SCNN have the significantly minimal MT, it suggests that the proposed Adap-Evol mechanism accelerates the response speed of SNN. Further, compared to SFNN and SCNN, the values of MT of both Evol-SFNN and Evol-SCNN are still smaller.

On the other hand, under the majority of the values of input firing rates, Evol-SNN or Adap-Evol-SNN achieve the highest FA scores, which mean that while gaining the response speed, Evol and Adap-Evol, especially Adap-Evol mechanism can also slightly improve the classification accuracy.

C. Evolution of Variables $dv(t)$ And $C(t)$ over Time

According to Eq.(4), we know that the variable $dv(t)$ determines the firing activity of an IF neuron, and the larger the $dv(t)$, the more frequent the neuronal spike events. We use SFNN, Evol-SFNN and Adap-Evol-SFNN to infer a handwritten digit 2 of MNIST, respectively, Fig.11 compares the difference of the variation of neuronal $dv(t)$ over time, the input firing rate is set as 200Hz. The table column titles "Layer 2" and "Layer 3" denote the 1st and 2nd hidden layer respectively, and "Output" denotes the neurons of output layer. From the column "Layer 2" and "Layer 3" of Fig.11, we can see that the $dv(t)$ values of almost all the hidden neurons of SFNN are approximately 0, Adap-Evol-SFNN significantly increases the difference of $dv(t)$ among hidden neurons, while the increase of Evol-SFNN is less obvious. What should be noted is that both in Evol-SFNN and Adap-Evol-SFNN, the difference between the $dv(t)$ of neurons aB56re more and more obvious over time, and there are not only positive values of $dv(t)$, but also negative values of $dv(t)$.

On the other hand, from the column "Output" of Fig.11, we can find that for all the SFNN, Evol-SFNN and Adap-Evol-SFNN, the $dv(t)$ values of neuron index 3 are greater than that of other 9 neurons, which due to the neuron index 3 is the expected index of digit 2. Further, as time goes by, the $dv(t)$ of the output layer neurons of Adapt-Evol-SFNN is much larger than that of Evol-SFNN and SFNN (approximately 5000 vs. 100 and 1), which also means that in Adap-Evol-SFNN, the expected output neuron is able to acquire more spike stimuli, i.e., more synaptic currents are input into this neuron.

Model	Metric	50Hz	100Hz	200Hz	500Hz	1000Hz
SFNN	FA [%]	88.5	89.4	90.5	91.2	90.6
	MT [ms]	385	252	167	76	42
Evol-SFNN	FA [%]	88.4	89.6	90.6	**91.3**	90.5
	MT [ms]	247	163	116	66	38
Adap-Evol-SFNN	FA [%]	**89.9**	**89.7**	**91.2**	91.2	**91.4**
	MT [ms]	**172**	**125**	**78**	**45**	**28**
SCNN	FA [%]	**90.2**	89.9	**91.2**	90.5	91.2
	MT [ms]	391	288	151	66	49
Evol-SCNN	FA [%]	89.4	**90.6**	90.4	91.1	91.0
	MT [ms]	233	142	109	53	34
Adap-Evol-SCNN	FA [%]	90.1	90.3	**91.0**	91.3	**91.6**
	MT [ms]	**184**	**119**	**82**	**52**	**26**

TABLE I: Performance comparison between SNN, Evol-SNN and Adap-Evol-SNN, under different input firing rates.

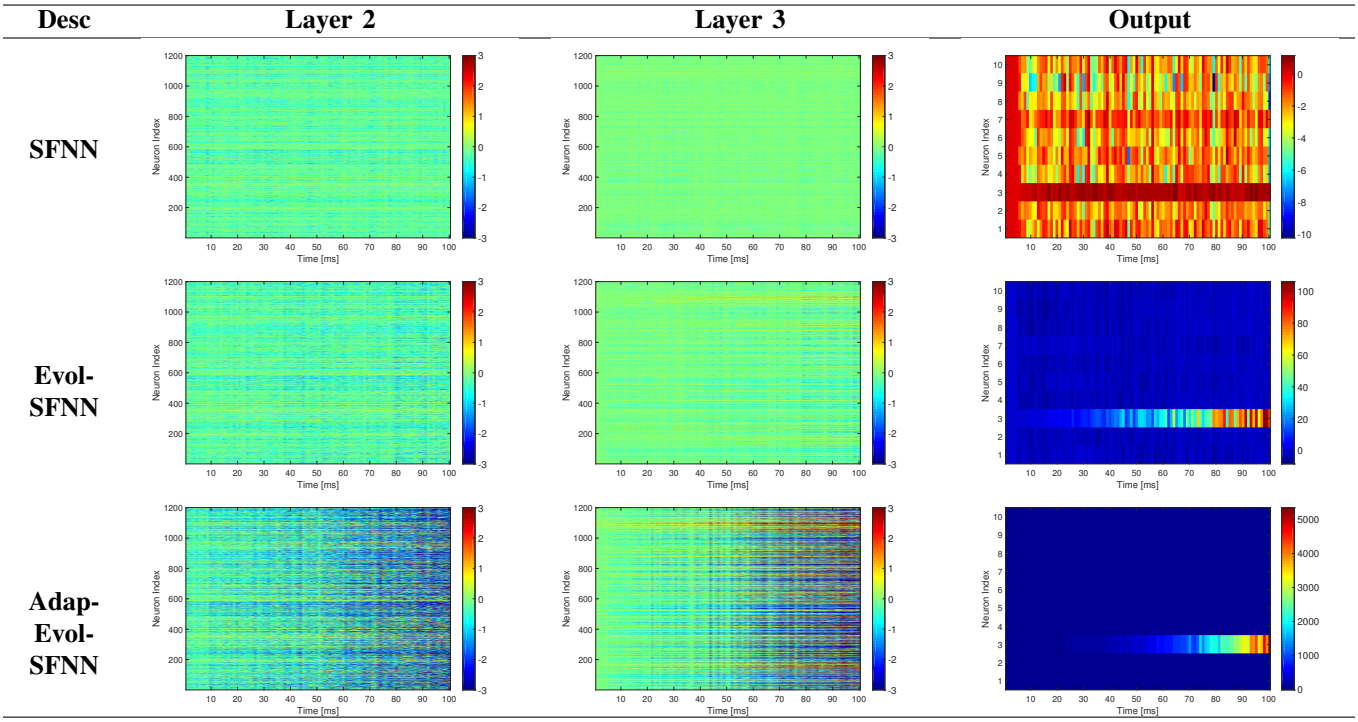


Fig. 11: Comparison of the $dv(t)$ of Eq.(4) in SFNN, Evol-SFNN, and Adap-Evol-SFNN.

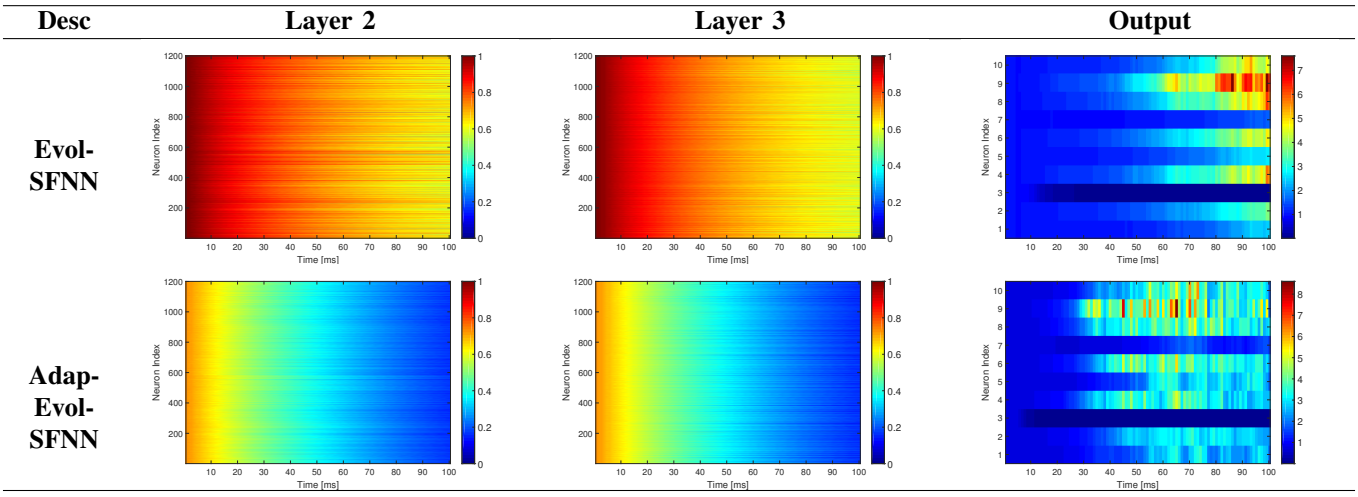


Fig. 12: Comparison of the $C(t)$ of Eq.(4) in SFNN, Evol-SFNN, and Adap-Evol-SFNN.

Therefore, compared to SFNN and Evol-SFNN, the expected output neuron will collect enough information evidence to classify the input signal significantly earlier.

For deeply understanding the effect of the proposed "Evol"

and "Adap-Evol" mechanisms, in Fig.12, we present the variation of $C(t)$ values of Evol-SFNN as well as Adap-Evol-SFNN while inferring a handwritten digit 2. From column "Layer 2" and "Layer 3", we can see that the change range of $C(t)$ vales of Evol-SFNN and Adap-Evol-SFNN is not the same, and the $C(t)$ of both shows a downward trend from 0 to 100ms, however in contrast, the $C(t)$ of Adap-Evol-SFNN drops faster than that of Evol-SFNN. In the column "Output" of Fig.12, the $C(t)$ value of the neuron index 3 corresponding to the handwritten digit 2 is the smallest of all, which supports the fact showed in the column "Output" of Fig.11 that the membrane potential change $dv(t)$ of neuron index 3 is the largest of all output neurons.

VI. Conclusion

Different with the traditional second generation of artificial neural networks (rate-based ANNs), spiking neural networks use the streams of spike events to process information, and even during the inference stage, SNNs also take time to collect enough evidence of spike information to produce reliable output, which results in the output delay of network. In this paper, we firstly proposed a self-evolutionary spiking neuron model (Evol), and further proposed an adaptive self-evolutionary model (Adap-Evol) for SNNs. Both Evol and Adap-Evol focus on regulating the conductivity of neuron membrane dynamically according to the spiking activity of the neuron and its external current input. Once an input signal is first input into an SNN, our proposed methods can activate the neuronal state rapidly by increasing the response of neurons to the input signal. When the output of the network tends to be stable, the spiking activity of each neuron also tends to be stable and regular, during this stage, the proposed method keeps the conductivity relatively unchanged. Finally, two different neural network frameworks, namely spiking multi-layer perceptron and spiking convolutional neural network, are utilized to validate the effectiveness of MNIST and Fashion-MNIST datasets. Theoretical analysis and experiment results show the improvement in terms of reducing the output delay, while the number of spike events at the time of accuracy convergence does not increase, and even by Evol method, the number of spike events decreases.

References

[1] S. Fusi, P. Del Giudice, and D. Amit, "Neurophysiology of a VLSI spiking neural network: LANN21," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*. IEEE, 2002, pp. 121–126 vol.3. [Online]. Available: http://ieeexplore.ieee.org/document/861291/

[2] J. Schemmel, A. Grubl, K. Meier, and E. Mueller, "Implementing Synaptic Plasticity in a VLSI Spiking Neural Network Model," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2008, pp. 1–6.

[3] A. Afifi, A. Ayatollahi, and F. Raissi, "Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits," *ECCTD 2009 - European Conference on Circuit Theory and Design Conference Program*, pp. 563–566, 2009. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/5275035/

[4] R. Wang, G. Cohen, K. M. Stiefel, T. J. Hamilton, J. Tapson, and A. van Schaik, "An FPGA Implementation of a Polychronous Spiking Neural Network with Delay Adaptation." *Frontiers in neuroscience*, vol. 7, no. February, p. 14, 2013. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3570898{\&}tool=pmcentrez{\&}rendertype=abstract

[5] K. Minkovich, C. M. Thibeault, M. J. O'Brien, A. Nogin, Y. Cho, and N. Srinivasa, "HRLSim: A high performance spiking neural network simulator for GPGPU clusters," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 2, pp. 316–331, feb 2014.

[6] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, aug 2014. [Online]. Available: http://www.sciencemag.org/cgi/doi/10.1126/science.1254642

[7] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[8] ——, "Fast sigmoidal networks via spiking neurons," *Neural Computing*, vol. 9, no. 2, pp. 279–304, 1997.

[9] ——, "Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons," *Advances in Neural Information Processing Systems*, vol. 9, p. 211, 1997.

[10] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "STDP-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, 2018.

[11] 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, 2015.

[12] E. Stromatias, M. Soto, T. Serrano-Gotarredona, and B. Linares-Barranco, "An event-driven classifier for spiking neural networks fed with synthetic or dynamic vision sensor data," *Frontiers in Neuroscience*, vol. 11, no. JUN, jun 2017. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2017.00350/full

[13] N. Kasabov and E. Capecci, "Spiking neural network methodology for modelling, classification and understanding of EEG spatio-temporal data measuring cognitive processes," *Information Sciences*, vol. 294, pp. 565–575, feb 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025514006562

[14] J. P. Dominguez-Morales, Q. Liu, R. James, D. Gutierrez-Galan, A. Jimenez-Fernandez, S. Davidson, and S. Furber, "Deep Spiking Neural Network model for time-variant signals classification: A real-time speech recognition approach," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2018-July. IEEE, jul 2018, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8489381/

[15] A. Zhang, W. Zhu, and J. Li, "Spiking echo state convolutional neural network for robust time series classification," *IEEE Access*, vol. 7, pp. 4927–4935, 2019.

[16] B. Rueckauer and S. C. Liu, "Conversion of analog to spiking neural networks using sparse temporal coding," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2018-May, 2018.

[17] 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," *Frontiers in Neuroscience*, vol. 7, no. 7 OCT, pp. 1–13, 2013. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2013.00178/abstract

[18] P. U. Diehl, D. Neil, J. Binas, M. Cook, S. C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2015-September, 2015. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7280696/

[19] S. Park, S. Kim, H. Choe, and S. Yoon, "Fast and Efficient Information Transmission with Burst Spikes in Deep Spiking Neural Networks," no. 1, sep 2018. [Online]. Available: http://arxiv.org/abs/1809.03142

[20] D. Zambrano, "Fast and Efficient Asynchronous Neural Computation with Adapting Spiking Neural Networks," no. December, 2016.

[21] R. Spiess, R. George, M. Cook, and P. U. Diehl, "Structural Plasticity Denoises Responses and Improves Learning Speed," *Frontiers in Computational Neuroscience*, vol. 10, 2016. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fncom.2016.00093

[22] M. Galarreta and S. Hestrin, "A network of fast-spiking cells in the neocortex connected by electrical synapses," *Nature*, vol. 402, no. 6757, pp. 72–75, 1999. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/10573418{\%}5Cnhttp://www.nature.com/doifinder/10.1038/47029

[23] A. Aertsen, M. Diesmann, and M.-O. Gewaltig, "Stable propagation of synchronous spiking in cortical neural networks

: Abstract : Nature," *Nature*, vol. 402, no. 6761, pp. 529–533, 1999. [Online]. Available: http://www.nature.com/doifinder/10.1038/990101{\%}5Cnpapers2://publication/doi/10.1038/990101

[24] E. T. Bell and T. J. Sejnowski, "An information-maximization ap- proach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[25] G. Chechik, A. Globerson, N. Tishby, and Y. Weiss, "Information bottleneck for gaussian variables," *Journal of Machine Learning Research*, vol. 6, pp. 165–188, 2005.

[26] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

[27] S. Roy and A. Basu, "An online unsupervised structural plasticity algorithm for spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 4, pp. 900–910, 2017. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7508492/

[28] 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," *Frontiers in Neuroscience*, vol. 11, no. DEC, pp. 1–12, 2017.

[29] S. Kim, S. Park, B. Na, and S. Yoon, "Spiking-YOLO: Spiking Neural Network for Real-time Object Detection," *Arxiv*, mar 2019. [Online]. Available: http://arxiv.org/abs/1903.06530

[30] Q. Liu, G. Pineda-García, E. Stromatias, T. Serrano-Gotarredona, and S. B. Furber, "Benchmarking Spike-Based Visual Recognition: A Dataset and Evaluation," *Frontiers in Neuroscience*, vol. 10, no. November, nov 2016. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2016.00496/full

Anguo Zhang was born in Hefei city, Anhui province, China in 1990. He received his bachelor's degree and master's degree in control engineering from Chongqing University, Chongqing in 2012 and 2016, respectively. Since 2018, he has been a researcher and senior engineer in the Research Institute of Ruijie, Ruijie Networks Co., Ltd. He is also pursuing the Ph.D in Communication and Information Systems at Fuzhou University. His research interest includes machine learning, artificial neural networks, control theory and applications.

Yuzhen Niu received her PhD in Computer Science from Shandong University of China in 2010. She was a Post-Doctoral Researcher with the Department of Computer Science, Portland State University, Portland, OR. She is currently a Professor with the College of Mathematics and Computer Science, Fuzhou University, China. Her current research interests include image and video processing, computer vision, artificial intelligence, and multimedia.

Yueming Gao received the Ph.D degree in Electric Engineering from Fuzhou University, Fuzhou Fujian, China in 2010. He is now a professor in College of Physical and Information Engineering of Fuzhou University, vice director of International S&T Cooperative Base of Healthy Medical Instrumentation of Chinese MOST. His research interest are detection and processing of biomedical signal, biomedical instrumentation, and engineering of Chinese traditional medicine.

Ying Han (1992-) received the B.S. degree in Medicine from the Fujian Medical University, Fuzhou, in 2015 and M.S. degree in Nursing from Fujian Medical University in 2018. She is currently a doctoral student in Xiamen University majoring in biologics. Her previous research was in nursing of post- operation for renal transplantation patients and management in manpower resources of hospital.

Qing Chen received the M.S. degree from the School of Automation, Chongqing University, Chongqing, China, in 2017, where she is currently pursuing the Ph.D. degree.

Her current research interests include artificial neural networks, intelligent control, robust adaptive control.

Wei Zhu (JUI LANG CHU) received the B.S. degree in electromechanics from Harbin Institute of Technology in 2006, Harbin, China and the Ph.D. degree in Instrumentation Science from University of Science and Technology of China, Hefei, China in 2011. He continued his post-doctoral research on neurological disease assistant diagnosis in National University of Singapore, Singapore.

He is currently a Senior Researcher in Ruijie Networks, Fuzhou, China and the team leader of Artificial Intelligence and Computer Vision. His current research interests include machine learning, computer vision and neurology.