THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 23, Number 2/2022, pp. 189–198

AN ONLINE SUPERVISED LEARNING ALGORITHM BASED ON FEEDBACK ALIGNMENT FOR MULTILAYER SPIKING NEURAL NETWORKS

Xianghong LIN, Jia HU, Donghao ZHENG, Tiandou HU, Xiangwen WANG

Northwest Normal University, College of Computer Science and Engineering, Lanzhou, China
Corresponding author: Xianghong LIN, E-mail: linxh@nwnu.edu.cn

Abstract. The feedback alignment provides a biologically plausible learning mechanism, which can directly transmit error signals with a random weight matrix to multiple layers of a neural network. This paper proposes an online supervised learning algorithm based on the feedback alignment mechanism for multilayer spiking neural networks, named Multi-OSLFA, which can support real-time learning for the spatio-temporal pattern of spike trains. The online learning rule is represented by the kernel function of spike trains and adjusts the synaptic weights when the output neuron fires a spike during the running process of spiking neural networks. The Multi-OSLFA algorithm is successfully applied to spike train learning tasks and nonlinear pattern classification problems on two UCI datasets. Simulation results indicate that the proposed algorithm can improve learning accuracy in comparison with other supervised learning algorithms. It shows that the proposed learning algorithm is effective for solving spatio-temporal pattern learning problems.

***Key words**:* spiking neural network, supervised learning, online learning, feedback alignment.

1. INTRODUCTION

Spiking neural network (SNN) is a new brain-inspired computing model, which encodes and transmits neural information through precisely timed spike trains [1]. SNN has been proved to be a suitable tool for processing spatio-temporal information [2]. Supervised learning of SNN refers to adding the multiple input spike trains to the network and comparing the corresponding desired output spike trains with the actual output spike trains to obtain the error signal, so as to find the appropriate synaptic weights to minimize the error after multiple training [3]. According to the running mode of supervised learning algorithms of SNNs, they can be divided into two categories: online learning and offline learning [4]. For the entire actual and desired output spike trains, offline learning can only adjust the parameters once after running the network. However, the synapses of biological neural networks are updated in a real-time manner. Online learning can adjust parameters many times when the output neuron fires spikes in the running of the network. Therefore, online learning is a biologically plausible algorithm, and suitable for solving real-time problems.

In recent years, some online supervised learning algorithms of SNNs have been proposed [5]. Based on the spike-timing-dependent plasticity (STDP) mechanism, Ponulak and Kasinski [6] proposed a remote supervised method (ReSuMe) for training single layer SNNs in an online manner, where the adjustment of synaptic weights is represented as a combination of STDP and anti-STDP. Sporea and Grüning [7] extended this work to multilayer SNNs, called Multi-ReSuMe. Wang et al. [8] proposed an online hybrid learning method for feedforward SNNs with an adaptive structure that combines unsupervised and supervised learning rules. Lin et al. [9, 10] reported the online supervised learning algorithm based on spike train kernels for single layer networks. Combining the mechanisms of error backpropagation and spike train convolution, Lin et al. [11] have proposed a supervised learning algorithm based on the operations of spike train inner products for multilayer feedforward SNNs, named Multi-STIP. The Multi-STIP algorithm adjusts the synaptic weights in an offline manner. In order to better compare online and offline algorithms, we extend Multi-STIP to an online learning algorithm, and call it Multi-OSTIP.

The traditional backpropagation algorithm transmits the error by multiplying the error signal with all the synaptic weights on the axon of each neuron and passing it further down. This process involves a precise and symmetrical connection pattern, but in fact, this pattern is impossible to achieve in the brain. In other words, in backpropagation, all synaptic connections represent two types of weights: the weights in the forward path and the weights in the feedback path. In the brain, the connections in the forward and feedback paths are different, and the weights in the feedback path are changing in the weight adjustment process. Grossberg [12] introduced this problem and called this phenomenon the weight transfer problem. In recent years, Lillicrap et al. [13] proposed a feedback alignment mechanism, in which the weight matrix for propagating network error does not need to be symmetrical with the weight for forwarding propagation, but can be randomly generated and then stay the same. Feedback alignment is considered to be more in line with the biological mechanism.

In this paper, on the basis of the spike train kernel function, an online supervised learning algorithm based on the feedback alignment mechanism is proposed for multilayer feedforward SNNs, named Multi-OSLFA. The characteristics of Multi-OSLFA are as follows: (1) The derivation of the weight learning rule only depends on the convolution of spike trains, so it can be applied to various spiking neuron models. (2) By constructing real-time network error, synaptic weights in all layers are updated many times in the running process of the network. (3) The online learning algorithm based on the feedback alignment mechanism has good biological plausibility. In addition, in order to compare with the proposed learning algorithm, the Multi-OSLFA can also be extended to offline learning manner, and called Multi-SLFA.

2. MULTI-OSLFA: ONLINE SUPERVISED LEARNING ALGORITHM

2.1. Spike train kernel representation and network error

In SNNs, the input and output signals of spiking neurons are encoded into discrete spike trains. The spike train $s(t)=\{t^f \in \Gamma : f=1,2,\cdots,F\}$ represents the ordered sequence of spikes fired by the spiking neuron in the interval Γ. The spike train can be formally expressed as follows:

$$s(t)=\sum_{f=1}^{F}\delta(t-t^f)\,, \tag{1}$$

where t^f is the f-th spike time, and F is the number of spikes. $\delta(\bullet)$ is the Dirac delta function, $\delta(x)=1$ if $x=0$ and $\delta(x)=0$ otherwise. Using the convolution operation with a symmetric and positive definite kernel function $\kappa(x)$, the spike train can be converted to a unique continuous function [14]:

$$f_s(t)=s(t)*\kappa(t)=\sum_{f=1}^{F}\kappa(t-t^f)\,. \tag{2}$$

When the spike trains are converted the continuous functions, a linear relationship between the inputs of presynaptic neurons and the output of postsynaptic neurons can be expressed as [15]:

$$f_{s_o}(t)=\sum_{i=1}^{N}w_{oi}(t)f_{s_i}(t)\,, \tag{3}$$

where $f_{s_o}(t)$ and $f_{s_i}(t)$ are convolved continuous functions corresponding to the postsynaptic spike train $s_o(t)$ and the presynaptic spike train $s_i(t)$, respectively. w_{oi} is the synaptic weight between the presynaptic neuron i and the postsynaptic neuron o, and N is the number of presynaptic neurons. In fact, the spike train linear summation relationship has been used for deducing the corresponding learning rule in the SNNs [16].

For the multilayer SNN, assuming that $s_o^d(t)$ and $s_o^a(t)$ represent the desired and actual output spike trains of output neuron o, the error $e_o(t)$ of each output neuron is defined in terms of the square difference between the convolved continuous functions $f_{s_o^d}(t)$ and $f_{s_o^a}(t)$. The total instantaneous error of the network at time t can be formally defined as:

$$E(t) = \frac{1}{2}\sum_{o=1}^{N_O} e_o(t) = \frac{1}{2}\sum_{o=1}^{N_O}\left[f_{s_o^a}(t) - f_{s_o^d}(t) \right]^2 , \tag{4}$$

where N_O is the number of neurons in the output layer.

2.2. Learning rules based on feedback alignment mechanism

Error backpropagation is an extremely important technique for designing supervised learning algorithms and has been widely applied to train neural networks. In the traditional backpropagation algorithm, the connection weight in the feedback path must be equal to the weight in the corresponding forward path. In this process, the error of the upstream neuron needs to be multiplied by a weight matrix that is completely symmetrical with the feedforward connection. Therefore, during the backpropagation process, the error signal matrix of upstream neurons is multiplied by the weight matrix $\boldsymbol{W}^{\mathrm{T}}$, which is the transpose of the weight matrix $\boldsymbol{W}$ of forward synaptic connections. Backpropagation means that feedback neurons must know all synaptic weights in the forwarding path. It is considered biologically unreasonable and difficult to implement neural circuits in the brain. Therefore, the feedback alignment mechanism is an effective and biologically feasible alternative method and exists in the backpropagation algorithm for training neural networks [17]. The feedback alignment replaces $\boldsymbol{W}^{\mathrm{T}}$ with a fixed random weight matrix $\boldsymbol{B}$. The concept diagram of the error backpropagation with feedback alignment mechanism is shown in Fig. 1. The feedforward SNN architecture is used in this paper, where the neurons are fully connected. The real-time error signal $E(t)$ is the sum of $e_o(t)$ of each output neuron in the output layer, it is directly used to update the synaptic weights between the hidden and output layers, and propagated to the hidden layer through the fixed random matrix $\boldsymbol{B}$ to update the synaptic weights between the input and hidden layers.

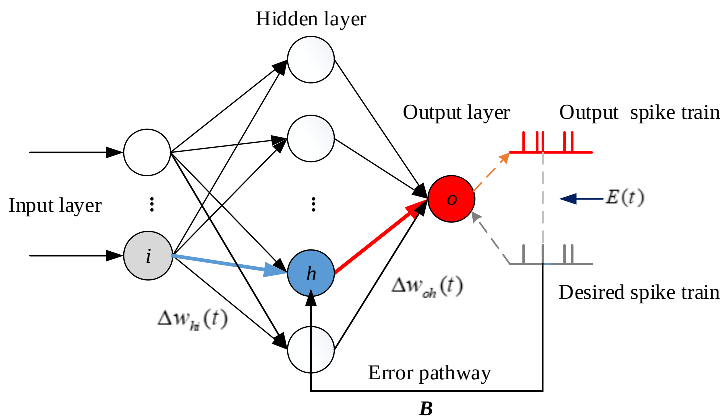

Fig. 1 – The diagram of error backpropagation with feedback alignment mechanism for a feedforward SNN with one hidden layer.

According to the gradient descent rule, the change of synaptic weight $\Delta w(t)$ at time t is computed as:

$$\Delta w(t) = -\eta \nabla E(t) = -\eta \frac{\partial E(t)}{\partial w(t)} , \tag{5}$$

where η is the learning rate and $\nabla E(t)$ represents the gradient calculation value of the total spike train error $E(t)$. The gradient can be expressed as the derivative of the instantaneous error $E(t)$ with respect to synaptic weight $w(t)$ at the time t.

We firstly deduce the learning rule for synaptic weights between the hidden and output layers. For a connection weight w_{oh} in the SNN, the gradient value is computed using the chain rule:

$$\nabla E_{oh}(t) = \frac{\partial E(t)}{\partial w_{oh}(t)} = \frac{\partial E(t)}{\partial f_{s_o^a}(t)} \frac{\partial f_{s_o^a}(t)}{\partial w_{oh}(t)} . \tag{6}$$

According to Eq. 4, the first partial derivative term of Eq. 6 is computed as:

$$\frac{\partial E(t)}{\partial f_{s_o^a}(t)} = f_{s_o^a}(t) - f_{s_o^d}(t) . \tag{7}$$

Using the linear relationship of spike trains in Eq. 3, the second partial derivative term of Eq. 6 is computed as:

$$\frac{\partial f_{s_o^a}(t)}{\partial w_{oh}(t)} = f_{s_h}(t) , \tag{8}$$

where $f_{s_h}(t)$ is the continuous function corresponding to the spike train $s_h(t)$ fired by the neuron h in the hidden layer.

Therefore, the online adjustment rule for synaptic weights between the hidden and output layers can be expressed as follows:

$$\Delta w_{oh}(t) = -\eta \nabla E_{oh}(t) = \eta \left[f_{s_o^d}(t) - f_{s_o^a}(t) \right] f_{s_h}(t) . \tag{9}$$

In the learning process, when the output neuron fires a desired spike or actual spike, the synaptic weight is updated, the adjustment value is determined by $f_{s_h}(t)$ and the difference between $f_{s_o^d}(t)$ and $f_{s_o^a}(t)$ at time t. If $f_{s_o^d}(t) - f_{s_o^a}(t) > 0$, the synaptic weight is strengthened; if $f_{s_o^d}(t) - f_{s_o^a}(t) < 0$, the synaptic weight is weakened; if $f_{s_o^d}(t) - f_{s_o^a}(t) = 0$, the synaptic weight is not changed.

Furthermore, we deduce the learning rule for synaptic weights between the input and hidden layers. The derivative of the instantaneous error $E(t)$ is expressed as:

$$\nabla E_{hi}(t) = \frac{\partial E(t)}{\partial w_{hi}(t)} = \frac{\partial E(t)}{\partial f_{s_h}(t)} \frac{\partial f_{s_h}(t)}{\partial w_{hi}(t)} . \tag{10}$$

Using the chain rule, the first partial derivative term of Eq. 10 can be computed as:

$$\frac{\partial E(t)}{\partial f_{s_h}(t)} = \sum_{o=1}^{N_o} \frac{\partial E(t)}{\partial f_{s_o^a}(t)} \frac{\partial f_{s_o^a}(t)}{\partial f_{s_h}(t)} . \tag{11}$$

Using the feedback alignment mechanism for the error signal backpropagation, the feedback weight matrix $\boldsymbol{W}^{\mathrm{T}}$ can be replaced by a fixed random matrix $\boldsymbol{B}$. The second partial derivative term of Eq. 11 is expressed as:

$$\frac{\partial f_{s_o^a}(t)}{\partial f_{s_h}(t)} = b_{oh} , \tag{12}$$

where b_{oh} is an element in $\boldsymbol{B}$. By combining Eqs. 7 and 12, Eq. 11 can be rewritten as:

$$\frac{\partial E(t)}{\partial f_{s_h}(t)} = \sum_{o=1}^{N_o} \left[f_{s_o^a}(t) - f_{s_o^d}(t) \right] b_{oh} . \tag{13}$$

Similarly, using the linear summation relationship the spike trains between all input neurons and a hidden neuron, the second partial derivative term of Eq. 10 is computed as:

$$\frac{\partial f_{s_h}(t)}{\partial w_{hi}(t)} = f_{s_i}(t) , \tag{14}$$

where $f_{s_i}(t)$ is the continuous function corresponding to the spike train $s_i(t)$ fired by the input neuron i.

Therefore, the online adjustment rule for synaptic weights between the input and hidden layers can be expressed as follows:

$$\Delta w_{hi}(t) = -\eta \nabla E_{hi}(t) = \eta \sum_{o=1}^{N_o} \left[f_{s_o^d}(t) - f_{s_o^a}(t) \right] f_{s_i}(t) b_{oh} \,. \tag{15}$$

In the learning process of synaptic weight from the input neuron to the hidden neuron, the updating value of synaptic weight is determined by $f_{s_i}(t)$ and the difference between $f_{s_o^d}(t)$ and $f_{s_o^a}(t)$ at time t.

The synaptic weights in the feedforward SNNs can be adjusted according to Eqs. 9 and 15 in an online manner. We integrate Eqs. 9 and 15, the offline supervised learning algorithm can be obtained. The learning rules of Multi-SLFA are expressed by the inner products of the corresponding spike trains:

$$\Delta w_{oh} = \eta \left[F(s_o^d, s_h) - F(s_o^a, s_h) \right] = \eta \left[\sum_{m=1}^{N_o^d} \sum_{n=1}^{N_h} \kappa(t_m^d - t_n^h) - \sum_{m=1}^{N_o^a} \sum_{n=1}^{N_h} \kappa(t_m^a - t_n^h) \right] \tag{16}$$

$$\Delta w_{hi}(t) = \eta \sum_{o=1}^{N_o} \left[F(s_o^d, s_i) - F(s_o^a, s_i) \right] b_{oh} = \eta \sum_{o=1}^{N_o} \left[\sum_{m=1}^{N_o^d} \sum_{n=1}^{N_i} \kappa(t_m^d - t_n^i) - \sum_{m=1}^{N_o^a} \sum_{n=1}^{N_i} \kappa(t_m^a - t_n^i) \right] b_{oh} \,, \tag{17}$$

where $F(s_k, s_l) = \left\langle f_{s_k}(t), f_{s_l}(t) \right\rangle = \int_\Gamma f_{s_k}(t) f_{s_l}(t) \mathrm{d}t$ is the inner product of the corresponding functions $f_{s_k}(t)$ and $f_{s_l}(t)$ [9]. t_m^d, t_m^a, t_n^h and t_n^i are the spike times in the corresponding spike trains, and N_o^d, N_o^a, N_h and N_i are the numbers of spikes in the spike trains.

3. EXPERIMENTS AND RESULTS

3.1. Learning of spike trains

The spike train learning tasks are used to demonstrate the learning performance of the Multi-OSLFA algorithm. The algorithm is applied to train SNN that output the given target spike trains. The clock-driven strategy is used to simulate SNNs with time step $\mathrm{d}t = 0.1$ ms. The three-layer SNN contains 40 input neurons, 100 hidden neurons, and one output neuron. The synaptic weights are generated in the interval [0, 0.2] that satisfies the uniform distribution, and the feedback elements in the fixed random matrix $\boldsymbol{B}$ are generated in the interval [0, 0.05]. The learning rate of the algorithm is $\eta = 0.005$. In this paper, the spiking neuron uses the spike response model (SRM) [11], the spike firing threshold is $\theta = 1$, the time constant of the postsynaptic potential is $\tau = 2$, the time constant of refractory period is $\tau_R = 50$, and the absolute refractory period is $t_R = 1$ ms. The Laplacian kernel function $\kappa(s) = \exp(-|s|/\sigma)$ with parameter $\sigma = 5$ is used in all simulations. In the biological nervous system, it is usually observed that the spikes fired by neurons satisfy the Poisson distribution of a certain frequency. For spike train learning tasks, the spike trains in the input layer are generated randomly by a homogeneous Poisson encoding method [18] with spike firing rate $r = 40$ Hz in the time interval $\Gamma = [0, 200]$ ms. The outputs of randomly initialized SNN are used as the desired spike trains. To quantitatively evaluate the learning performance, the spike train kernel is used to define a correlation-based measure C [3]. The value of C is within the range of [0, 1]. When the two spike trains are identical, the value of C is 1, and gradually tends to 0 as their similarity becomes lower and lower. The results are averaged over 20 trials, and each trial of the learning algorithm is applied for a maximum of 200 learning epochs.

In the following experiments, we show the learning process and the influence of different parameters on the Multi-OSLFA algorithm. Figure 2 shows the spike train learning process of Multi-OSLFA to reproduce the desired spatio-temporal spike pattern. Figure 2a shows the complete learning process in the time interval Γ. It shows that the actual output spike train gradually approaches the desired spike train. The evolution of learning accuracy with measure C during the learning process is presented in Fig. 2b. The learning accuracy increases gradually with the increase of the learning epoch. After 135 learning epochs, the learning accuracy C reaches 1.0. As shown in Figure 2c, synaptic weights from the input layer to the hidden layer change over time during the 50th learning epoch. The results from top to bottom show the changes of synaptic weights at 50 ms, 100 ms, 150 ms and 200 ms respectively. Figure 2d shows the real-time changes of the synaptic weights between the hidden and output layers during a learning epoch. When the output neuron fires a spike, the synaptic weights change over time, which indicates that the Multi-OSLFA algorithm is learning in a real-time manner.

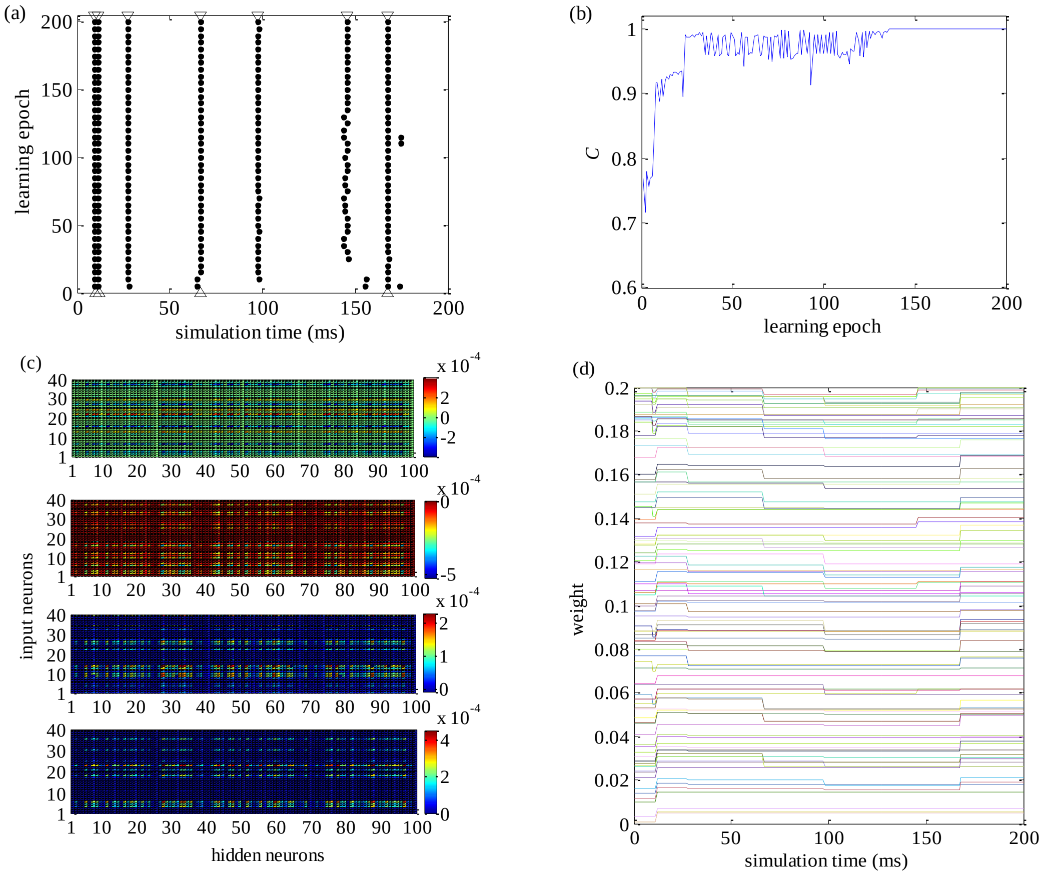

Fig. 2 – The spike train learning process of the proposed Multi-OSLFA algorithm: a) the complete learning process (△, the initial actual output spike train before learning; ▽, desired output spike train; •, actual output spike trains during the learning process); b) the evolution of learning accuracy with measure *C*; c) the changes of the synaptic weights between the input and hidden layers during a learning epoch; d) the changes of the synaptic weights between the hidden and output layers the interval [0, 200] ms.

Figure 3 shows the learning results of the Multi-OSLFA, Multi-OSTIP, Multi-SLFA and Multi-STIP algorithms with different learning rates. As shown in Fig. 3a, with the increase of the learning rate, the learning accuracy C of the four algorithms increases first and then decreases. When the learning rate is less than 0.01, the learning accuracy of the two online learning algorithms is higher than that of the two offline algorithms. The two online algorithms have the highest learning accuracy when the learning rate is 0.005. The average learning accuracy C of the Multi-OSLFA and Multi-OSTIP algorithms is 0.9999 and 0.9996, respectively. Multi-SLFA achieves the best learning result when the learning rate is 0.05, and the learning accuracy C is 0.9961. When the learning rate is 0.005, the Multi-STIP algorithm achieves the best learning result and the learning accuracy C is 0.9946. Figure 3b shows the learning epoch when the learning accuracy C reaches the maximum value. The learning epochs required for the four algorithms to achieve maximum learning accuracy are gradually reduced with the increase of the learning rate. When the learning rate is 0.005, the mean learning epochs of the four algorithms are 68.2, 36.9, 160.95 and 160.55, respectively. Therefore, in the following spike train learning experiments, we set the learning rates of the Multi-OSLFA, Multi-OSTIP, Multi-SLFA and Multi-STIP algorithms as 0.005, 0.005, 0.05 and 0.005, respectively.

Figure 4 shows the results of the four algorithms with different spike train firing rates in the input layer of SNNs. The firing rate of input spike trains is increased from 20 Hz to 100 Hz in steps of 10 Hz while the other parameter settings remain the same. As shown in Fig. 4a, the learning accuracy of these four algorithms is decreased gradually with the increase of the spike train firing rate, and the learning accuracy of the Multi-OSLFA algorithm is higher than that of the other three algorithms. In addition, the learning accuracy of the online algorithms is higher than that of the offline algorithms. For example, when the spike train firing rate is

60 Hz, the learning accuracy C of the Multi-OSLFA, Multi-OSTIP, Multi-SLFA and Multi-STIP algorithms is 0.9962, 0.9933, 0.9329 and 0.9852, respectively. Figure 4b shows the learning epochs when the learning accuracy C reaches the maximum value. It shows that the learning epochs of these four algorithms increase first and then decrease. This simulation indicates that the proposed Multi-OSLFA algorithm can well learn with different firing rates of the spike trains.

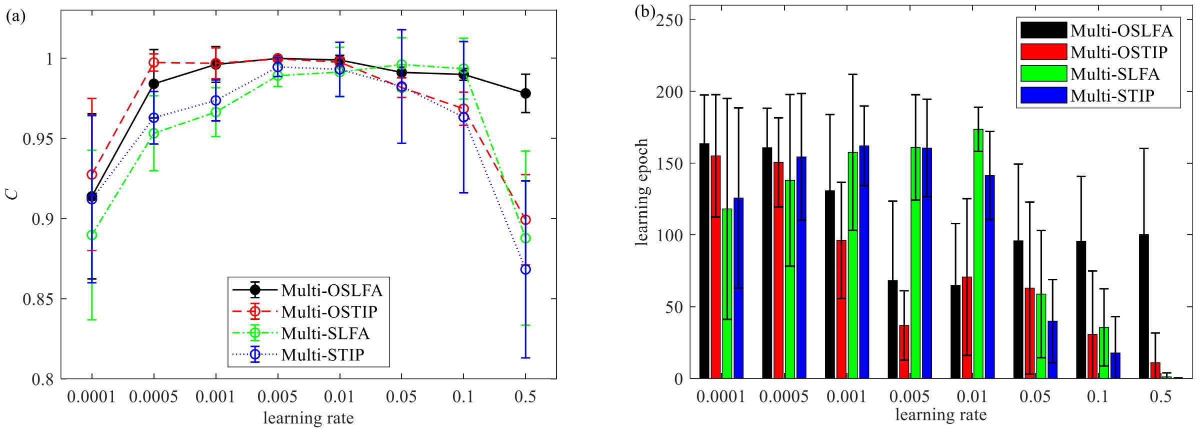

Fig. 3 – Learning results with different learning rate: a) the learning accuracy C; b) the learning epochs when the learning accuracy C reaches the maximum value.

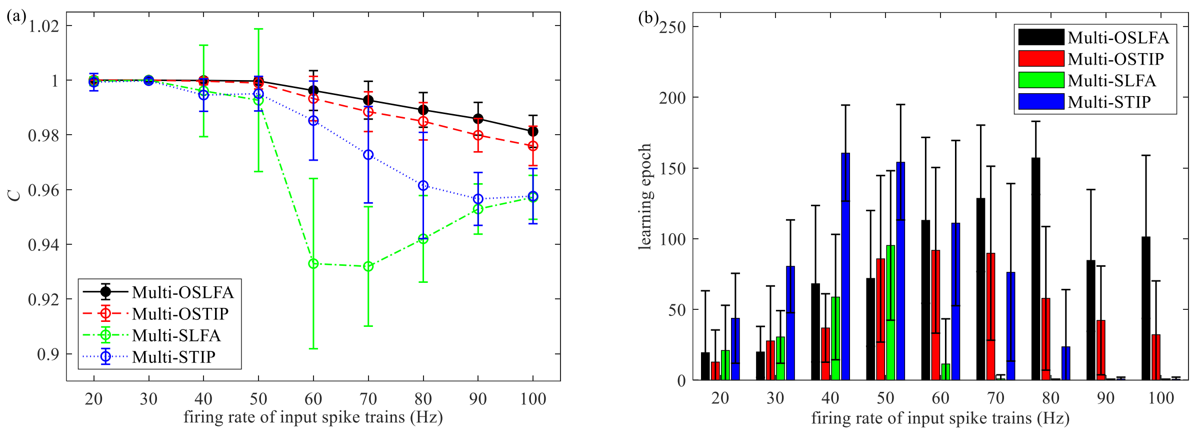

Fig. 4 – Learning results with different firing rates of input spike trains: a) the learning accuracy C; b) the learning epochs when the learning accuracy C reaches the maximum value.

Finally, the learning performance of the proposed Multi-OSLFA algorithm with different lengths of spike trains is tested and compared with the Multi-OSTIP, Multi-SLFA and Multi-STIP algorithms. The length of spike trains is increased from 100 ms to 1000 ms in steps of 100 ms while the other parameter settings remain the same. The corresponding learning results are shown in Fig. 5. Figure 5a shows the learning accuracy of the four learning algorithms with different spike train lengths after 200 learning epochs. The learning accuracy C of these four algorithms is decreased slowly with the increase of the length of spike trains, and the standard deviation of C gradually increases. In addition, the learning accuracy of the Multi-OSLFA algorithm is higher than that of the other three algorithms. It can be seen from the results that the learning accuracy of the online algorithms is higher than that of the offline algorithms, and the learning accuracy of the algorithms with the feedback alignment mechanism is higher than that of the algorithms that propagate error using weight matrix $\boldsymbol{W}^{\mathrm{T}}$. When the length of spike trains is 800 ms, the learning accuracy C of the Multi-OSLFA, Multi-OSTIP, Multi-SLFA and Multi-STIP algorithms is 0.9913, 0.9845, 0.8937 and 0.9733, respectively. As shown in Fig. 5(b), the learning epochs of the four algorithms increase gradually with the increase of the length of the spike trains. For example, when the length of spike trains is 400 ms, the mean learning epochs for the four algorithms are 100.4, 103.75, 102.1 and 155.35, respectively. This simulation indicates that the proposed Multi-OSLFA algorithm can well learn in a large range of spike train lengths.

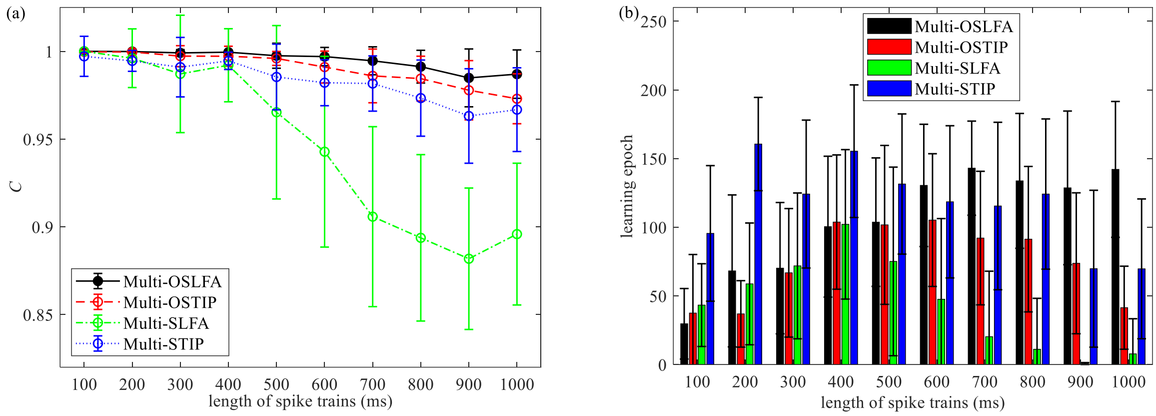

Fig. 5 – Learning results with different spike train lengths: a) the learning accuracy *C*; b) the learning epochs when the learning accuracy *C* reaches the maximum value.

3.2. Nonlinear pattern classification

In this experiment, we apply the proposed online learning algorithm to practical nonlinear pattern classification problems. Figure 6 shows the computational process of pattern classification using the SNN. Firstly, the features of samples in the dataset are normalized, and the linear encoding method is used to convert the samples into the spike trains within a given simulation duration [19, 20]. The number of input neurons depends on the number of features of the sample. Then, the encoded spike trains of each sample are input into the SNN, and the proposed online learning algorithm is used for training. The instantaneous network error is calculated according to the actual output spike train and the desired output spike train representing the input sample label information given in advance. Synaptic weights are updated in real-time by error backpropagation. Finally, the classification information of input sample is obtained according to the similarity of spike trains.

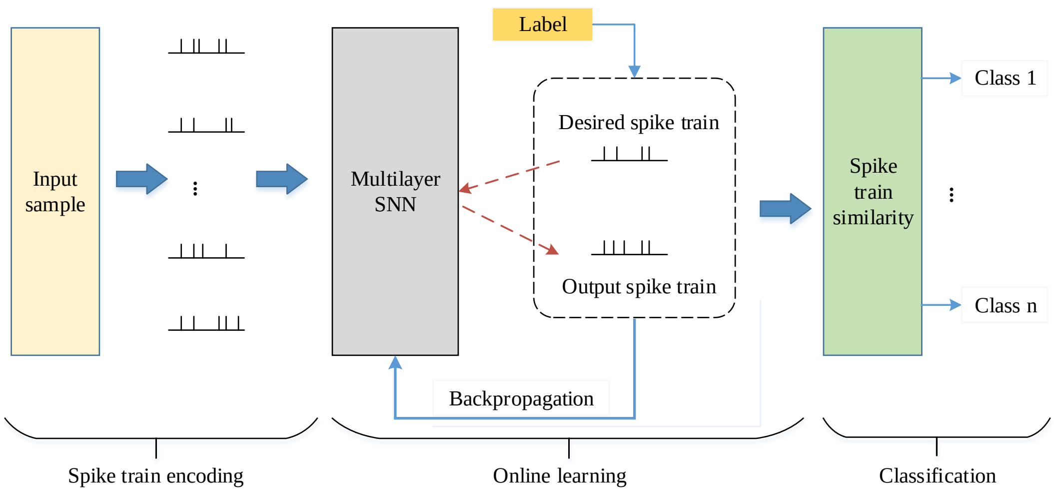

Fig. 6 – The pattern classification process using SNNs with the online learning method.

Two benchmark datasets are selected from the UCI machine learning repository, including Wisconsin breast cancer (WBC) and Pima Indians diabetes (PIMA) (The programs and datasets of this experiment are available at http://121.42.213.146:8080/SNNLea3.0.zip). The WBC dataset contains 683 samples with 9 features, and the PIMA dataset contains 768 samples with 8 features. The samples in the dataset are divided equally into the training set and testing set. In the following experiments, the simulation duration is 100 ms. The values of time decay constants of the SRM model are $\tau = 5$ and $\tau_R = 50$. Table 1 shows the parameter settings for different datasets, where the range of the initial synaptic weights and the fixed random matrix $\boldsymbol{B}$ before learning is different. The label information of each sample in the two datasets is encoded into desired spike trains with different frequencies using a linear encoding scheme. The classification results are averaged over 20 trials.

Table 1

Parameter settings for different datasets

Dataset	Weight range	Weight range of ***B***	Frequency range of input spike train (Hz)	Frequency of desired spike trains (Hz)	Upper limit of learning epochs
WBC	[0, 2.0]	[0, 0.05]	[15, 50]	40, 50	150
PIMA	[0, 1.0]	[0, 0.005]	[10, 45]	30, 50	200

To further evaluate the learning performance of the Multi-OSLFA algorithm, we compare the classification accuracy of the Multi-OSLFA algorithm against some other supervised learning algorithms for SNNs for the datasets of WBC and PIMA. The SpikeProp [21], SWAT [20], Multi-ReSuMe [7] and Multi-STIP [11] are chosen, the classification accuracy of the two datasets refer to the results described in [20] and [22]. The classification accuracy of different learning algorithms for the datasets of WBC and PIMA is shown in Table 2. It can be seen that for the two datasets, the SpikeProp algorithm achieves the highest classification accuracy on both the training set and the testing set. However, the SpikeProp algorithm is a single-spike learning method, which adopts a more complex data encoding method and network structure. For the multi-spike learning methods, the classification accuracy of the Multi-OSLFA algorithm is 97.2% and 96.9% on the training set and the testing set respectively, which is higher than that of the SWAT, Multi-ReSuMe, Multi-STIP and Multi-SLFA algorithms on the WBC dataset. For the PIMA dataset, the classification accuracy of the Multi-OSLFA algorithm is 72.7% and 72.2% on the training set and the testing set, respectively. In general, our proposed Multi-OSLFA algorithm can well solve the nonlinear pattern classification problems and achieve high classification accuracy on both the training set and the testing set for different datasets.

Table 2

The classification accuracy of different supervised algorithms on two datasets

Dataset	Algorithm	Architecture	Training accuracy (%)	Testing accuracy (%)	Epochs
WBC	SpikeProp [20]	64-15-2	97.6 ± 0.2	97.0 ± 0.6	500
	SWAT [20]	9-117-2	96.2 ± 0.4	96.7 ± 2.3	500
	Multi-ReSuMe	9-40-1	95.5 ± 1.2	94.8 ± 2.1	150
	Multi-STIP	9-40-1	96.1 ± 0.8	96.1 ± 0.5	150
	Multi-SLFA	9-40-1	96.9 ±0.6	96.4 ± 0.8	150
	Multi-OSLFA	9-40-1	97.2 ± 0.7	96.9 ± 0.6	150
PIMA	SpikeProp [22]	49-20-2	78.6 ± 2.5	76.2 ± 1.8	3000
	SWAT [22]	48-624-2	77.0 ± 2.1	72.1 ± 1.8	500
	Multi-ReSuMe	8-30-1	69.3 ± 3.0	68.4 ± 2.5	200
	Multi-STIP	8-30-1	72.1 ± 1.3	71.1 ± 1.1	200
	Multi-SLFA	8-30-1	73.2 ± 1.7	71.3 ± 2.3	200
	Multi-OSLFA	8-30-1	72.7 ± 1.1	72.2 ± 1.5	200

4. CONCLUSIONS

In this paper, we use the feedback alignment mechanism to design a new online supervised learning algorithm for multilayer feedforward SNNs. The discrete spike trains are converted to continuous functions using convolution operation. Then the continuous functions are used to construct the error function and derive the online learning rule of synaptic weights. The learning rule only depends on the precisely timed spike trains, so it can be theoretically applied to various spiking neuron models. It can be seen from the spike train learning experiments that the proposed Multi-OSLFA algorithm can successfully learn the spatio-temporal pattern of spike trains. Compared with the Multi-OSTIP, Multi-SLFA and Multi-STIP algorithms, the Multi-OSLFA algorithm can obtain higher learning accuracy. Since the instantaneous error is backpropagated through the feedback alignment mechanism and synaptic weight is updated in a real-time manner, it has a biologically plausible mechanism for learning. In addition, the two UCI benchmark datasets of WBC and PIMA are used to test the classification performance of the proposed algorithm. The Multi-OSLFA learning algorithm is compared with different supervised learning algorithms for multilayer feedforward SNNs. Classification results show that the proposed method can effectively solve the nonlinear pattern classification problems and achieve high classification accuracy on both the training set and the testing set for different datasets.

ACKNOWLEDGEMENTS

This work was supported by the Key Research and Development Project of Gansu Province under grant no. 20YF8GA049, the Youth Science and Technology Fund Project of Gansu Province under grant no. 20JR10RA097, and the Lanzhou Municipal Science and Technology Project under grant no. 2019-1-34.

REFERENCES

1. S. GHOSH-DASTIDAR, H. ADELI, *Spiking neural networks,* International Journal of Neural Systems, **19**, pp. 295–308, 2009.
2. C. TAN, M. SARLIJA, N. KASABOV, *NeuroSense: Short-term emotion recognition and understanding based on spiking neural network modelling of spatio-temporal EEG Pattern,* Neurocomputing, **433**, pp. 137–148, 2021.
3. X. LIN, M. ZHANG, X. WANG, *Supervised learning algorithm for multilayer spiking neural networks with long-term memory spike response model*, Computational Intelligence and Neuroscience, art. 8592824, 2021.
4. X. WANG, X. LIN, X. DANG, *Supervised learning in spiking neural networks: A review of algorithms and evaluations*, Neural Networks, **125**, pp.258–280, 2020.
5. J.L. LOBO, J. DELSER, A. BIFET, N. KASABOV, *Spiking neural networks and online learning: An overview and perspectives*, Neural Networks, **121**, pp. 88–100, 2020.
6. F. PONULAK, A. KASINSKI, *Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting,* Neural Computation, **22**, pp. 467–510, 2010.
7. I. SPOREA, A. GRÜNING, *Supervised learning in multilayer spiking neural networks,* Neural Computation, **25**, pp. 473–509, 2013.
8. J. WANG, A. BELATRECHE, L. MAGUIRE, T.M. MCGINNITY, *An online supervised learning method for spiking neural networks with adaptive structure*, Neurocomputing, **144**, pp. 526–536, 2014.
9. X. LIN, N. ZHANG, X. WANG, *An online supervised learning algorithm based on nonlinear spike train kernels*, In Proceedings of the International Conference on Intelligent Computing, pp. 106–115, 2015.
10. X. WANG, X. LIN, X. DANG, *A delay learning algorithm based on spike train kernels for spiking neurons*. Frontiers in Neuroscience, **13**, art. 252, 2019.
11. X. LIN, X. WANG, Z. HAO, *Supervised learning in multilayer spiking neural networks with inner products of spike trains*, Neurocomputing, **237**, pp. 59–70, 2017.
12. S. GROSSBERG, *Competitive learning: From interactive activation to adaptive resonance*, Cognitive Science, **11**, pp. 23–63, 1987.
13. T.P. LILLICRAP, D. COWNDEN, D.B. TWEED, C.J. AKERMAN, *Random synaptic feedback weights support error backpropagation for deep learning*, Nature Communications, **7**, art. 13276, 2016.
14. I.M. PARK, S. SETH, M. RAO, J.C. PRINCIPE, *Strictly positive-definite spike train kernels for point-process divergences*, Neural Computation, **24**, pp. 2223–2250, 2012.
15. R. GTIG, R. AHARONOV, S. ROTTER, H. SOMPOLINSKY, *Learning input correlations through nonlinear temporally asymmetric Hebbian plasticity*, The Journal of Neuroscience, **23**, pp. 3697–3714, 2003.
16. A. CARNELL, D. RICHARDSON, *Linear algebra for time series of spikes*, Proceedings of the 13th European Symposium on Artificial Neural Networks, 2005, pp. 363–368.
17. B. CRAFTON, A. PARIHAR, E. GEBHARDT, A. RAYCHOWDHURY, *Direct feedback alignment with sparse connections for local learning*. Frontiers in Neuroscience, **13**, art. 525, 2019.
18. Y. XU, X. ZENG, L. HAN, J. YANG, *A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks*, Neural Networks, **43**, pp. 99–113, 2013.
19. C. GLACKIN, L.J. MCDAID, L. P. MAGUIRE, H.M. SAYERS, *Implementing fuzzy reasoning on a spiking neural network*, Proceedings of the 15th International Conference on Artificial Neural Networks, 2008, pp. 258–267.
20. J.J. WADE, L.J. MCDAID, J.A. SANTOS, H.M. SAYERS, *SWAT: a spiking neural network training algorithm for classification problems*, IEEE Transactions on Neural Networks, **21**, pp. 1817–1830, 2010.
21. S.M. BOHTE, J.N. KOK, H. POUTRE, *Error-backpropagation in temporally encoded networks of spiking neurons*, Neurocomputing, **48**, pp. 17–37, 2002.
22. A. JEYASOTHY, S. SUNDARAM, N. SUNDARARAJAN, *SEFRON: a new spiking neuron model with time-varying synaptic efficacy function for pattern classification*, IEEE Transactions on Neural Networks and Learning Systems, **30**, pp. 1231–1240, 2019.

Received November 8, 2021