

New Efficient Multi-Spike Learning for Fast Processing and Robust Learning

Shenglan Li,¹ Qiang Yu^{1*}

¹Tianjin Key Laboratory of Cognitive Computing and Application,
College of Intelligence and Computing, Tianjin University, Tianjin, China
{lishenglan, yuqiang}@tju.edu.cn

Abstract

Spiking neural networks (SNNs) are considered to be more biologically plausible and lower power consuming than traditional artificial neural networks (ANNs). SNNs use discrete spikes as input and output, but how to process and learn these discrete spikes efficiently and accurately still remains a challenging task. Moreover, most existing learning methods are inefficient with complicated neuron dynamics and learning procedures being involved. In this paper, we propose efficient alternatives by firstly introducing a simplified and efficient neuron model. Based on it, we develop two new multi-spike learning rules together with an event-driven scheme being presented to improve the processing efficiency. We show that, with the as-proposed rules, a single neuron can be trained to successfully perform challenging tasks such as multi-category classification and feature extraction. Our learning methods demonstrate a significant robustness against various strong noises. Moreover, experimental results on some real-world classification tasks show that our approaches yield higher efficiency with less requirement on computation resource, highlighting the advantages and potential of spike-based processing and driving more efforts towards neuromorphic computing.

Introduction

Due to the development of deep learning, artificial neural networks (ANNs) have prospered in many fields, such as computer vision and speech recognition (LeCun, Bengio, and Hinton 2015). However, current deep learning methods are resource consuming, data depending and lack biological plausibility as compared to high efficiency of the brain. Therefore, efficient and biologically plausible spiking neural networks (SNNs) can thus be more favorable (Maass 1997). Unlike artificial neurons that communicate with each other by numerical values, biological ones use discrete all-or-nothing response, i.e. spikes (Kandel et al. 2000). It is believed that spikes play an essential role in efficient processing in biological systems, but how to model the neurons in a way such that they can efficiently learn to process

spikes while resembling biological counterparts to certain extent still remains challenging.

There are many spiking neuron models that have been proposed for emulating the ability of biological neurons, such as Hodgkin-Huxley model (Hodgkin and Huxley 1952), spike response model (SRM) (Gerstner and Kistler 2002), and leaky integrate-and-fire (LIF) model (Burkitt 2006). Among them, LIF and SRM models are most widely used due to their simple forms and computational effectivity (Ponulak 2005).

Based on these spiking neuron models, different learning algorithms have been developed. The tempotron (Gütig and Sompolinsky 2006) is an efficient rule that trains neurons to make decisions by binary response of firing or not. However, the binary response of the tempotron constrains its ability to utilize the temporal structure of its output (Yu, Li, and Tan 2018). To overcome this limitation, a series of learning rules were proposed to train neurons to fire at desired times, such as ReSuMe (Ponulak 2005), SPAN (Mohammed et al. 2012) and PSD (Yu et al. 2013). With association between different output spike times and categories, neuron can perform the challenging task of multi-category classification, but how biological systems establish such a precise instruction signal is still unclear. Moreover, most of these rules are designed for a temporal code where information is carried by the precise arrival time of input spikes (Panzeri et al. 2010), limiting their generalization to other schemes such as a rate code where information is encoded by firing frequencies (Brette 2015).

Recently, a new multi-spike tempotron (MST) rule (Gütig 2016) has been proposed to train neurons to fire desired number of spikes. This kind of multi-spike learning enables neurons to process both rate- and temporal-based spike patterns, and thus provides a new way for processing information under a broad range of coding schemes. Inspired by it, two multi-spike learning rules (Yu, Li, and Tan 2018), namely TDP1 and TDP2, were developed to improve simplicity and efficiency by using a linear assumption for threshold crossing. These rules are successfully applied to sound recognition and demonstrate advanced performance. (Yu et al. 2019). In a later work, another method was developed (Miao, Tang, and Pan 2018). It simplifies the expres-

*Corresponding author

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

sion of membrane potential by assuming an infinite threshold and update synaptic weights by local maximum point. However, a common disadvantage of these rules exists in the complexity of neuron dynamics, computation scheme and spike learning, as a result of which would limit their implementations in both software and hardware.

A recent variant was proposed to improve the processing speed by combining the advantages of both tempotron and PSD (Yu, Wang, and Dang 2018). Similarly, another efficient method was developed (Xiao Rong and Tang 2019) by quickly finding a local maximum value as the signal to update synaptic weights. In this paper, we continue to contribute to this family of multi-spike learning rules with efficiency mainly considered for both processing and learning. Firstly, we introduce a simplified LIF neuron model for efficient processing of spikes, where membrane potential is modeled with an impulse function. In addition, we take an event-driven scheme to further improve the processing efficiency. Based on these, we propose two new rules for multi-spike learning and evaluate their performance on various tasks such as multi-category classification, feature extraction and real-world classifications. Experimental results show that our method has higher performance and lower computational request than other baseline learning rules, making our work significantly valuable to spike-based and neuromorphic computing.

Method

In this section, the neuron model and two multi-spike learning algorithms are described in detail, together with an event-driven scheme.

Neuron Model

The biologically inspired LIF model has been widely used in many neuroscience tasks due to its simplicity and computational effectivity (Ponulak and Kasinski 2011). Therefore, we build our simplified neuron model based on it, as follows:

$$\tau \frac{dV}{dt} = -V(t) + I_{in}(t) + I_{out}(t) \quad (1)$$

where τ is the time constant of neuron's membrane potential V . I_{in} and I_{out} model the input currents from presynaptic neurons and firing reset dynamics, respectively. The two currents can be set in a simple form as Eq. 2 and 3:

$$I_{in}(t) = \sum_{i=1}^N w_i \sum_{t_i^j \leq t} \delta(t - t_i^j) \quad (2)$$

Here, N and w_i denote the number of the presynaptic afferents and the corresponding synaptic efficacy, respectively. t_i^j is the arrival time of the j -th presynaptic spike of the i -th afferent.

$$I_{out}(t) = \vartheta \sum_{t_s^j < t} \delta(t - t_s^j) \quad (3)$$

where ϑ denotes the firing threshold. Whenever neuron's membrane potential crosses it, an output spike is elicited. t_s^j represents the time of the j -th output spike. $\delta(t)$ is a unit

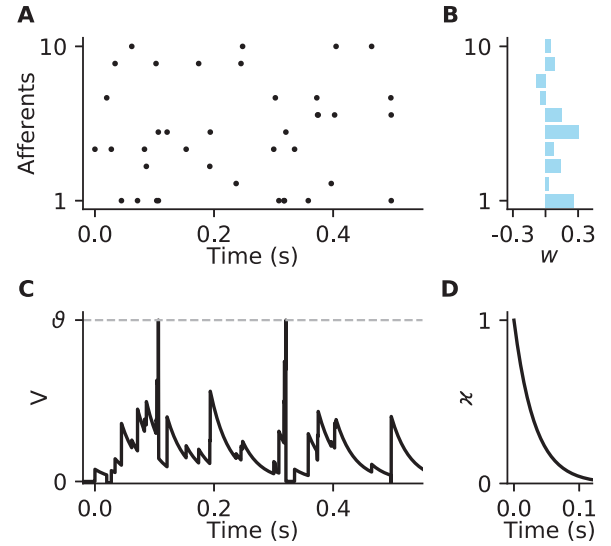


Figure 1: Dynamics of the spiking neuron model. **A.** input spike pattern where x and y axis denotes the time and afferent ids, respectively. Each dot represents a spike. **B.** synaptic weights of corresponding afferents in **A.** **C.** membrane potential trace of the neuron in response to the pattern in **A.** Gray dashed line denotes the firing threshold. **D.** The post-synaptic potential kernel, \varkappa .

impulse function where its value is equal to zero everywhere except for t at 0. Its integral over the entire domain is one. $\delta(t)$ describes the occurrence of neuron's firings from both pre-synaptic neurons (t_i^j) and its own output (t_s^j).

Substituting Eq. 2 and 3 into Eq. 1, the final form of the neuron model can be given as:

$$V(t) = \sum_{i=1}^N w_i \sum_{t_i^j \leq t} \varkappa(t - t_i^j) - \vartheta \sum_{t_s^j < t} \varkappa(t - t_s^j) \quad (4)$$

Here $\varkappa(t) = \exp(-t/\tau)$ is an exponential decay kernel describing the influence of spikes on postsynaptic membrane potential.

As is shown in Fig. 1, neuron's membrane potential evolves by integrating synaptic currents from afferent neurons. Each afferent spike contributes a postsynaptic potential (PSP), whose peak amplitude is determined by w , to the neuron. When its membrane potential crosses the firing threshold, an output spike is elicited, followed by a reset dynamics. When there is no input spike, the membrane potential will gradually decay to the rest level.

In most other multi-spike learning rules, a double-exponential kernel is adopted for PSP as:

$$K(t) = V_0[\exp(-t/\tau_m) - \exp(-t/\tau_s)] \quad (5)$$

where the maximum value of the kernel function is normalized by a factor V_0 . The membrane time constant τ_m and the synaptic time constant τ_s are generally set to 20 ms and 5 ms, respectively. We find the effects of different kernels on

neuron's membrane potential can be approximated by following $\int K(t)dt = \int \varkappa(t)dt$, and thus we set our parameter τ according it.

Compared to the double exponential kernel, our model is more simple, yet without affecting its performance on various tasks like recognition and feature extraction that we will demonstrate later in this paper.

Event-Driven Scheme

An efficient event-based computation scheme is adopted to improve the processing in this study. This event-driven approach is superior to a time-based one because it does not depend on a step size for simulation, thus reducing computational operations to be linearly related to the total number of input spikes (Yu, Li, and Tan 2018). In this approach, computations are driven by spikes one after another in order of time. So, for a stream input spikes $t_1 \leq t_2 \leq \dots \leq t_n$ with their corresponding synaptic weights w_1, w_2, \dots, w_n , the voltage at the k -th spike is given as:

$$V(t_k) = \sum_{i=1}^k w_i \exp\left(-\frac{t_k - t_i}{\tau}\right) - \vartheta \sum_{t_s^j < t_k} \exp\left(-\frac{t_k - t_s^j}{\tau}\right) \\ := D_k - E_k \quad (6)$$

The states of D_k and E_k can be expressed recursively as follows:

$$D_k = D_{k-1} \exp\left(-\frac{\Delta_{k-1}}{\tau}\right) + w_k \quad (7)$$

$$E_k = E_{k-1} \exp\left(-\frac{\Delta_{k-1}}{\tau}\right) \quad (8)$$

where $\Delta_{k-1} = t_k - t_{k-1}$ denotes the interspike interval before the k -th input spike. Our computation method is detailed in Algorithm 1.

Algorithm 1 Event-driven computation scheme

Input: Sorted input spike times $S = \{t_k | k=1,2,\dots,n\}$

Output: Output spike number n_o

```

1: function RESPONSE(S)
2:   for each input spike with time  $t_k$  do
3:      $\Delta_{k-1} = t_k - t_{k-1}$ 
4:     Update  $D_k$  by Eq. 7
5:     Update  $E_k$  by Eq. 8
6:     Update membrane potential  $V(t_k)$  by Eq. 6
7:     while  $V(t_k) \geq \vartheta$  do
8:        $V(t_k) \leftarrow V(t_k) - \vartheta$ 
9:        $E_k \leftarrow E_k + \vartheta$ 
10:      Elicit an output spike  $n_o \leftarrow n_o + 1$ 
11:    end while
12:  end for
13: return  $n_o$ 
14: end function

```

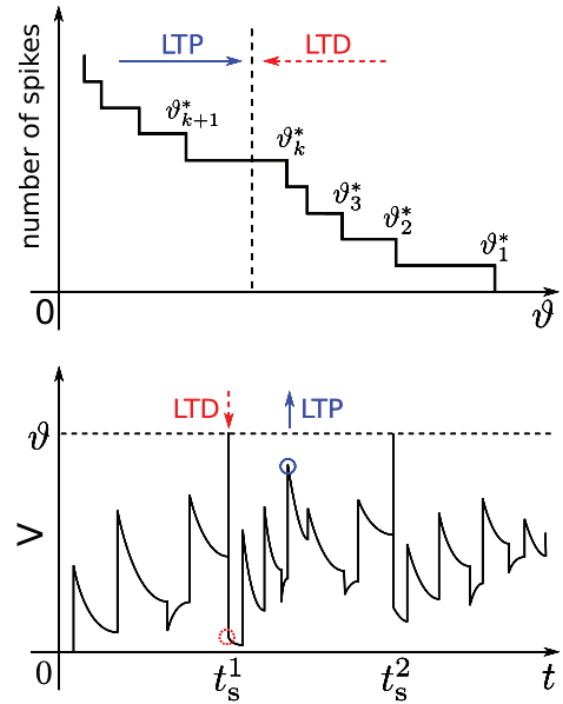


Figure 2: Illustration of EML and EMLC learning rule including the long-term potentiation (LTP) and long-term depression (LTD) process. Top, both LTP and LTD depends on critical thresholds ϑ_k^* . Vertical dashed line denotes the threshold of the neuron. Bottom, both LTP and LTD depends on the current state of the neuron in response to a spike pattern. The blue circle represents the maximum value of the subthreshold voltage, and the red circle denotes the minimum value of the reset potentials at all output spikes, t_s^j .

Learning Rule

Recently, MST (Gütig 2016) is proposed to train neurons to elicit spikes according to a given target spike number. Following this work, efficient variants are developed such as TDP1 and TDP2 (Yu, Li, and Tan 2018) where a linear assumption for threshold crossing is applied to simplify the processing. Further these studies, we propose two new ones based on our neuron model in this study.

A. The EML Rule

Like MST and TDP, our first learning rule named efficient multi-spike learning (EML), is also based on the spike-threshold-surface (STS) (Gütig 2016). STS describes the relation between the number of output spikes and the neuron's threshold. A small value of threshold will normally result in a high number of output spikes (see Fig. 2). STS is characterized by a sequence of critical threshold values ϑ_k^* at which n_{out} jumps from $k - 1$ to k . Therefore, for a given input spike pattern, the output spike number can be changed by manipulating the critical threshold.

Each critical threshold value ϑ_k^* corresponds to a voltage value described by Eq. 4, and thus it is a function of synap-

tic weights \vec{w} and differentiable with respect to them. For a given threshold ϑ , we define the maximum of all the peaks of subthreshold voltage as $v_{max}(\vartheta)$. Consider a given ϑ^* as the threshold, we assume there exists a t^* such that $V(t^*) = v_{max}(\vartheta^*) = \vartheta^* = \vartheta$. We denote the derivative of ϑ^* with respect to the i th synaptic efficacy as $\vartheta_i^{* \prime}$. In MST and TDP1, the influence of output spikes t_s^j ($j \in \{1, 2, \dots, m\}$) before t^* are considered, so the derivative is given as:

$$\vartheta_i^{* \prime} = \frac{\partial V(t^*)}{\partial w_i} + \sum_{j=1}^m \frac{\partial V(t^*)}{\partial t_s^j} \frac{\partial t_s^j}{\partial w_i} \quad (9)$$

Differently, in our EML method, we find the effect through previous output spikes can be neglected. The reason is that for any previous output spike j , $\exists \xi > 0$ such that $V(t_s^j) - \vartheta > \xi$. A sufficiently small change on w will hardly affect t_s^j . So the last component in Eq. 9 is a vanishing part. The derivative of our methods can thus be expressed as:

$$\vartheta_i^{* \prime} = \frac{\partial V(t^*)}{\partial w_i} = \sum_{t_s^j \leq t^*} \kappa(t^* - t_s^j) \quad (10)$$

According to Eq. 10, changing synaptic efficacies can modify critical threshold values, and thus results in a different output spike number. We choose a simple form of learning to modify synaptic weights whenever there is a mismatch between the neuron's actual (o) and desired (d) number of spikes, as:

$$\Delta w = \begin{cases} -\lambda \frac{d\vartheta_o^*}{dw} & \text{if } o > d \\ \lambda \frac{d\vartheta_{o+1}^*}{dw} & \text{if } o < d \end{cases} \quad (11)$$

where $\lambda > 0$ is the learning rate that controls the step size of each update. As is shown Fig. 2, LTP will occur when the neuron elicits less spikes than desired, while LTD will lead the learning if it fires too many.

B. The EMLC Rule

The above mentioned rules, such as MST, TDP and EML, are all based on STS. A common disadvantage is that they depend on evaluations of critical thresholds as well as their derivatives with respect to synaptic efficacy, resulting in relatively low efficiency. Here, we propose our second efficient multi-spike learning rule where only neuron's current states of response are considered. We name it as EMLC.

For a given input spike pattern, the number of output spikes can be adjusted by directly modifying the voltage closest to the threshold. We define the time points for adaptation as t_{LTP} and t_{LTD} . For LTP process, t_{LTP} is the time point corresponding to the maximum value of the subthreshold voltage. For LTD process, t_{LTD} is selected as the time point corresponding to the minimum value of the voltage after firing reset among all output spikes. We use these time points to adjust synaptic efficacies (see Fig. 2), and the EMLC rule can thus be formalized as:

$$\Delta w = \begin{cases} -\lambda \frac{\partial V(t_{LTD})}{\partial w} & \text{if } o > d \\ \lambda \frac{\partial V(t_{LTP})}{\partial w} & \text{if } o < d \end{cases} \quad (12)$$

In this way, neurons can thus learn to fire a desired number of spikes only based on its current state of response rather than a complicated STS function, resulting in a simplified computation.

Experiment

In this section, experiments are conducted to evaluate the performance of our methods. Firstly, we give the details of our default settings. Next, we test the effects of different initial setups, followed by experiments on multi-classification and feature extraction. Finally, we examine performance of our learning rules on some real-world datasets.

Experimental Setup

The neuron is connected with N afferents, and each one fires at a Poisson rate of $r_{in} = 4$ Hz over a time window T . We set $N = 500$ and $T = 500$ ms. The initial weights are drawn from a random Gaussian distribution with both mean and standard deviation being set as 0.01. Additionally, we set $\vartheta = 1$ and $\lambda = 0.0001$. Parameter setups different from the default would be stated otherwise. All experiments were conducted on a platform of Intel E5-2620@2.10GHz

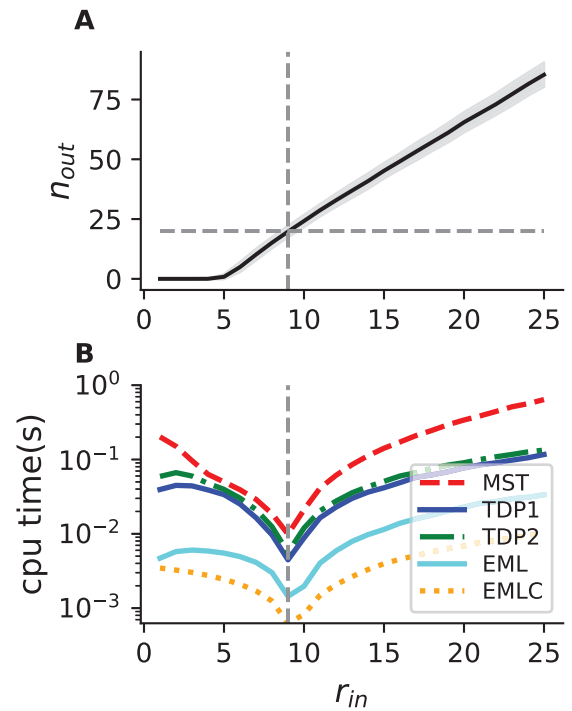


Figure 3: Effects of different afferent firing rates r_{in} . **A.** initial actual output spike number n_{out} versus r_{in} , solid line and shaded area denote the mean and standard deviation, respectively. The horizontal dotted line shows the desired output spike number while the vertical dotted line denotes the minimum cpu time in **B.** **B.** training times of different methods.

Here, we measure the effects of r_{in} and initial weight distribution w_{mean}^{init} on the processing and learning since these

two have a direct influence on neuron’s output response. Our learning efficiency is compared with baseline methods of MST, TDP1 and TDP2.

Effects of Initial Setups

Neuron’s output response varies under different initial conditions: some setups lead to a high firing rate output while others would result in a low or even silent response. How neurons respond under different setups and whether they can adapt to proper states effectively and efficiently with our learning rules are studied in this section.

In Fig. 3, patterns are generated by a Poisson process with different r_{in} . The target is to train neurons to fire 20 desired output spikes. Following setups in (Yu, Li, and Tan 2018), the learning rate is scaled as $\lambda r_{in}^0 / r_{in}$ with $r_{in}^0 = 4$ Hz. Fig. 3A shows neuron’s response to different setups of r_{in} : it starts to fire if r_{in} is greater than a certain amount (5 Hz here), and its output spike number increases linearly after that with r_{in} . Fig. 3B shows the learning efficiency of different multi-spike rules. For all the learning rules, the closer the distance between initial response and the desired one is, the less running time is required. Both our proposed learning rules, EML and EMLC, are faster than other baselines. EMLC is the fastest one among all.

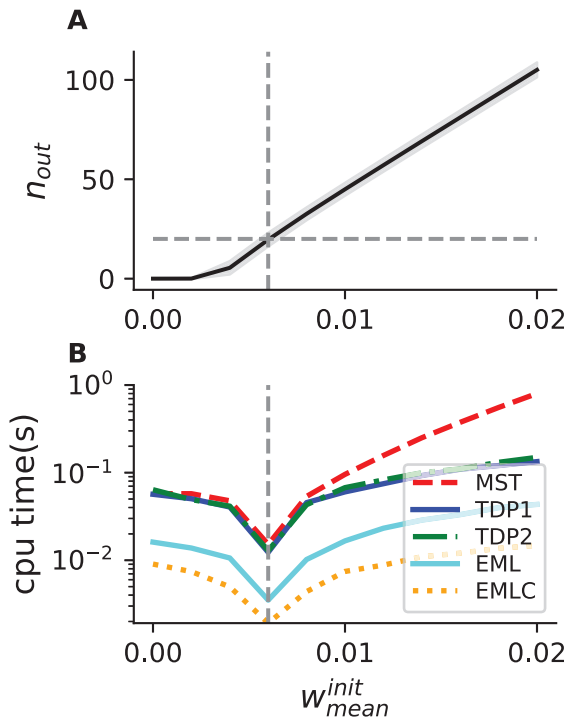


Figure 4: Effects of different initial mean weight w_{init}^{mean} . **A.** initial response n_{out} versus w_{init}^{mean} . **B.** execution times of learning.

Fig. 4 shows the effects of w_{init}^{mean} on neuron’s initial response and the performance of different learning methods. Again, our methods are more efficient than others as suggested by our experimental results.

Multi-category Classification

The learning ability of spiking neurons to classify different spike patterns have been widely studied (Brader, Senn, and Fusi 2007; Memmesheimer et al. 2014). Some of which focus on rate-coded patterns, while others based on precise timings. Here, we design several tasks to evaluate the learning performance of our method.

In the first task, we use our EML rule to train a single neuron to perform multi-category classification by firing different spike numbers in response to each. In this experiment, both the rate and temporal codes are considered. The experimental setups are similar to those in (Yu, Li, and Tan 2018). Under the rate coding scheme, we randomly generate three firing-rate templates, where a random half of the afferents have a low firing rate of 2 Hz while the other half with a high one of 10 Hz. The patterns are generated according to the Poisson process every time. For the temporal-based experiment, three spike pattern templates are randomly generated with the default setup and then fixed. New spike patterns are constructed by adding two types of noises to the templates: spike jitter noise (σ_{jit}) and random deletion noise (p_{del}). We set $\sigma_{jit} = 2$ ms and $p_{del} = 0.1$.

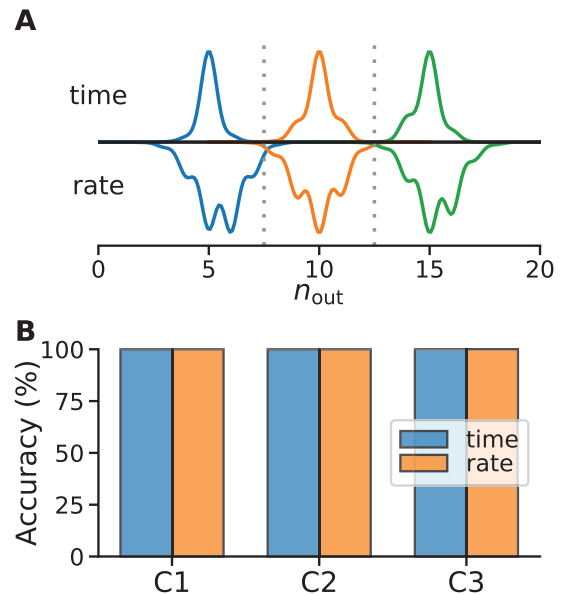


Figure 5: Multi-category classification of single neurons. **A.** Gaussian kernel evaluation of the output spike numbers, with a bandwidth of 0.35. The time- and rate-based experiments are separated by the horizontal line. Vertical dotted lines denotes the decision boundaries used in **B.** **B.** classification accuracy of each category.

A single neuron is trained to fire different output spike numbers in response to each category as : 5 (Class1), 10 (Class2) and 15 (Class3). After learning, the neuron can successfully discriminate different categories where information can be coded by either time or rate (see Fig . 5). Moreover, our learning rule achieves remarkable performance with accuracies of 100% on the given task.

In the second task, we use three neurons to learn the multi-category classification task, with each neuron being trained for one category. A temporal-based task is used to evaluate the robustness of our learning methods. Again, spike patterns of each category are constructed by adding two types of noises to the template: spike jitter noise σ_{jit} and spike deletion noise p_{del} . After training, these two noises are used to evaluate the learning performance. Multi-spike learning rules are applied to train the three neurons to elicit more than 20 spikes responding to their target category and to keep silent for the others. In the decision phase, we consider the classification is correct only when the corresponding neuron fires more than 10 spikes.

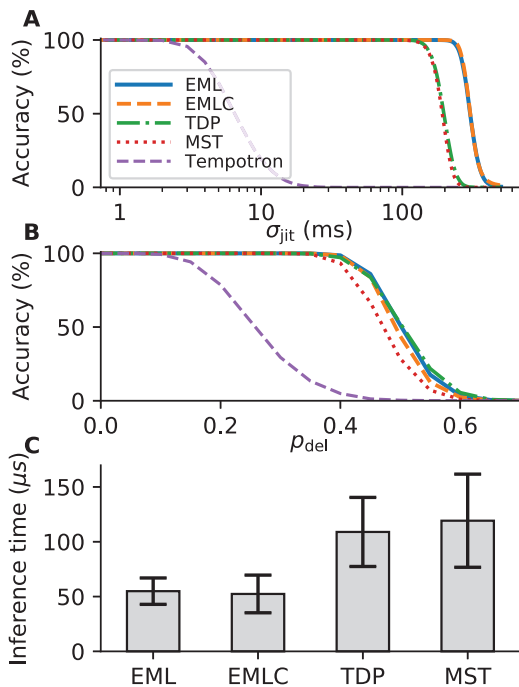


Figure 6: Learning performance of different rules on the classification task. **A.** learning accuracy under the spike jitter noise σ_{jit} . **B.** learning accuracy under the spike deletion noise p_{del} . **C.** inference time of the neurons in **A.** Data were averaged over 100 independent simulations.

Fig. 6 shows that the tempotron rule is inferior to all multi-spike learning rules in term of robustness, indicating the advantages of exploiting output temporal structure with multiple spikes. Notably, our EML and EMLC learning rules outperform the other multi-spike rules in terms of robustness against different noises. Moreover, the inference time shows that our methods are more efficient (over twice) than the others.

Learning for Feature Extraction

Useful information is often hidden in distracting streams of unrelated sensory activities, making it difficult to be detected

by neurons, a common challenging task referred as ‘temporal credit-assignment’ problem (Gütig 2016). In this part, we will show the ability of our EML rule for solving this problem.

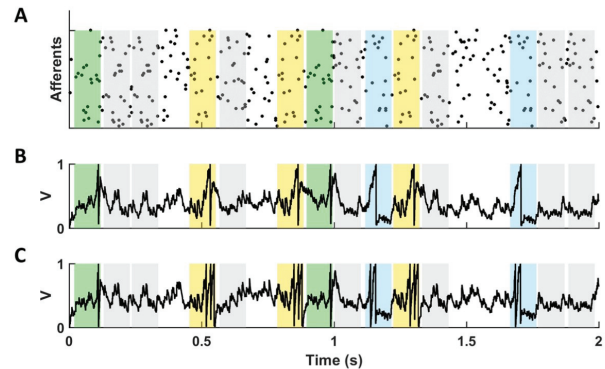


Figure 7: Learning for feature extraction. **A.** a spike stream where features (shaded green, yellow and blue) and distractors (shaded gray) are embedded in a background activity. **B** and **C** are the membrane potential trace of the neuron in response to the spike pattern in **A** after training.

Similar to the task in (Yu, Li, and Tan 2018), we first randomly generate six activity patterns with the background firing statistics and then fixed them. Half of the activity patterns are assigned as feature patterns while the others are referred as distractors. In each trial, the occurrence number of each activity pattern is generated by a Poisson process with a mean of 3, and then the activity patterns are randomly embedded to a random background over $T = 2s$. The task is to train the neuron to fire different desired spike numbers at the presence of different feature patterns while to keep silent for distractors as well as background. The total desired output spike number n_{out}^* in response to a trial pattern is given by $n_{out}^* = \sum_{i=1}^{n_f} c_i d_i$ where c_i is the number of the i -th feature pattern and d_i is the desired output spike number of the i -th feature. n_f is the total number of feature patterns.

Fig. 7 shows that our method can successfully learn the two different temporal credit-assignment tasks where d_i for three feature patterns are set as $\{1,1,1\}$ and $\{1,2,3\}$, respectively.

Real-world Classification Tasks

In this section, we evaluate the performance of our EML and EMLC rules for solving some more realistic recognition tasks. Different encoding schemes are adopted to convert external stimuli into spike patterns. We set the number of classification neurons to be the same as that of categories. The neurons are trained to fire a desired output spike numbers n_d for the corresponding category while keep silent for the others. We use a winner-take-all (WTA) scheme (Oster, Douglas, and Liu 2009) for our readout where the predicted category is represented by the neuron with most number of spikes. In addition, a momentum scheme (Gütig and Sompolinsky 2006) with $\mu = 0.9$ is applied to accelerate the learning.

Table 1: Performance of different multi-spike learning rules on UCI datasets.

Dataset	MST			TDP1 \ TDP2			Yu's work ¹			EML \ EMLC (Ours)		
	Accuracy	Epoch	Time(s)	Accuracy	Epoch	Time(s)	Accuracy	Epoch	Time(s)	Accuracy	Epoch	Time(s)
BC	95.68	100	4.59	95.80 \ 95.75	100 \ 100	4.18 \ 4.15	96.18	100	2.65	96.01 \ 91.24	100 \ 100	1.50 \ 1.48
Iris	95.80	200	3.17	95.52 \ 95.55	200 \ 200	2.82 \ 2.78	95.88	200	1.91	96.16 \ 91.58	200 \ 200	1.12 \ 1.08
Zoo	82.65	85	2.23	78.75 \ 80.25	88 \ 91	1.25 \ 1.32	80.90	82	1.16	80.00 \ 82.30	82 \ 76	0.69 \ 0.64

¹ This work in (Yu, Wang, and Dang 2018).

Table 2: The performance comparison on MNIST dataset.

Network Model	Neurons	Training size	Test size	Accuracy(%)
CNN+EML(Ours)	800+10	500	100	89
CSNN (Xu et al. 2018)	800+400	500	100	88
CNN+EML (Ours)	800+10	10000	5000	91.81
Dendritic Neurons (Hussain, Liu, and Basu 2014)	5000+10	10000	5000	90.26
CNN+EML (Ours)	800+10	60000	10000	92.52
Unsupervised STDP SNN (Diehl and Cook 2015)	784+6400+6400	60000	10000	95

A. UCI Benchmarking

In this part, three datasets are selected from UCI repositories (Asuncion and Newman 2007) and are used in this experiment. We use a Gaussian receptive field method (Eurich and Wilke 2000; Bohte, Kok, and La Poutre 2002) to encode input variables. 60% samples are used as training while the rest as the test. The target output spike number for each neuron is denoted as n_d which is set to 10 for the Breast Cancer and Iris datasets while set to 1 for the Zoo dataset.

- **Breast Cancer (BC)**. A binary classification dataset for breast cancer recognition which contains 683 samples with 9 features each. In our experiment, every input feature is encoded by 8 neurons with Gaussian receptive fields.
- **Iris**. It contains 3 categories with 50 samples in each. Every sample consists of 4 features. We use 12 neurons to encode each input feature.
- **Zoo**. This dataset is for animal classification where there are 7 classes with 16 features. The total number of samples is 101. We use 10 neurons to encode each feature.

Table 1 shows the recognition results of different multi-spike learning rules under the same experimental setups. For both Breast Cancer and Iris dataset, the training epochs are fixed, while an early-stop criterion is used to test convergence speed for the Zoo dataset. As is shown in the table, our methods are more efficient than all other baselines, with a leading speed across all dataset. The recognition accuracies of our methods are mostly at the top ranges or comparable to the best.

B. MNIST Dataset

Here, a more complex dataset, MNIST, is used to evaluate the performance of our EML rule. The MNIST dataset contains a large number of handwritten digits from 0 to 9, where each example has an image size of 28×28 pixels (Larochelle et al. 2007).

We adopt the encoding method of (Xu et al. 2018) that uses CNN to extract features. Then the activation values are linearly mapped to spiking times. We choose 800 encoding neurons and use 10 neurons to learn the patterns. In order to make a fair comparison with different methods, different numbers of samples used for training and test are set accordingly. The target output spike number n_d is set to 10 and time window T is set to 100 ms.

A comparison of different SNN-based models is shown in Table 2. The CSNN model (Xu et al. 2018) uses 400 tempotron neurons in the output layer with each 40 to represent a same class. With a number of 500 training samples, it achieves an accuracy of 88%. Remarkably, our method reaches 89% but with fewer neurons, thanks to the multi-spike characteristic. When more training data are used, the accuracy of our method increases as expected. With a size of 10000 training samples, our method outperforms the dendritic neurons (Hussain, Liu, and Basu 2014) and with a lighter structure. A complex network structure with more neurons could further improve the performance accuracy as that in (Diehl and Cook 2015), but a rate-based coding is normally the key. Both these can limit the efficiency of the network. Differently, our work demonstrates the advantages of our learning rule that enables light network structure under a temporal code to achieve a comparable accuracy.

Conclusion

In this work, two new multi-spike learning methods are proposed for fast processing and robust learning. We have demonstrated the advantages of our method with challenging tasks including multi-category classification and feature extraction. Our learning rules can successfully perform the task and show a strong robustness to noise. Furthermore, we evaluated the performance of our rule on several real-world dataset. Experimental results show that our method has higher performance and lower computation cost than the others, making it beneficial for fast and efficient neuromorphic computing.

Acknowledgments

This work was supported by the Natural Science Foundation of China (No. 61806139), and the Natural Science Foundation of Tianjin (No. 18JCYBJC41700).

References

- Asuncion, A., and Newman, D. 2007. Uci machine learning repository.
- Bohte, S. M.; Kok, J. N.; and La Poutre, H. 2002. Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing* 48(1-4):17–37.
- Brader, J. M.; Senn, W.; and Fusi, S. 2007. Learning real-world stimuli in a neural network with spike-driven synaptic dynamics. *Neural Computation* 19(11):2881.
- Brette, R. 2015. Philosophy of the spike: rate-based vs. spike-based theories of the brain. *Frontiers in systems neuroscience* 9:151.
- Burkitt, A. N. 2006. *A review of the integrate-and-fire neuron model: II. Inhomogeneous synaptic input and network properties*.
- Diehl, P. U., and Cook, M. 2015. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience* 9:99.
- Eurich, C. W., and Wilke, S. D. 2000. Multidimensional encoding strategy of spiking neurons. *Neural Computation* 12(7):1519–1529.
- Gerstner, W., and Kistler, W. M. 2002. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press.
- Gütig, R., and Sompolinsky, H. 2006. The tempotron: a neuron that learns spike timing-based decisions. *Nature neuroscience* 9(3):420.
- Gütig, R. 2016. Spiking neurons can discover predictive features by aggregate-label learning. *Science* 351(6277):aab4113.
- Hodgkin, A. L., and Huxley, A. F. 1952. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology* 117(4):500–544.
- Hussain, S.; Liu, S.-C.; and Basu, A. 2014. Improved margin multi-class classification using dendritic neurons with morphological learning. In *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2640–2643. IEEE.
- Kandel, E. R.; Schwartz, J. H.; Jessell, T. M.; of Biochemistry, D.; Jessell, M. B. T.; Siegelbaum, S.; and Hudspeth, A. 2000. *Principles of neural science*, volume 4. McGraw-hill New York.
- Larochelle, H.; Erhan, D.; Courville, A.; Bergstra, J.; and Bengio, Y. 2007. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, 473–480. ACM.
- LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature* 521(7553):436.
- Maass, W. 1997. Networks of spiking neurons: the third generation of neural network models. *Neural networks* 10(9):1659–1671.
- Memmesheimer, R.-M.; Rubin, R.; Ölveczky, B. P.; and Sompolinsky, H. 2014. Learning precisely timed spikes. *Neuron* 82(4):925–938.
- Miao, Y.; Tang, H.; and Pan, G. 2018. A supervised multi-spike learning algorithm for spiking neural networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–7. IEEE.
- Mohammed, A.; Schliebs, S.; Matsuda, S.; and Kasabov, N. 2012. Span: Spike pattern association neuron for learning spatio-temporal spike patterns. *International journal of neural systems* 22(04):1250012.
- Oster, M.; Douglas, R.; and Liu, S.-C. 2009. Computation with spikes in a winner-take-all network. *Neural computation* 21(9):2437–2465.
- Panzeri, S.; Brunel, N.; Logothetis, N. K.; and Kayser, C. 2010. Sensory neural codes using multiplexed temporal scales. *Trends in neurosciences* 33(3):111–120.
- Ponulak, F., and Kasinski, A. 2011. Introduction to spiking neural networks: Information processing, learning and applications. *Acta neurobiologiae experimentalis* 71(4):409–433.
- Ponulak, F. 2005. Resume-new supervised learning method for spiking neural networks. institute of control and information engineering, poznań university of technology. Technical report, Tech. rep.
- Xiao Rong, Yu Qiang, Y. R., and Tang, H. 2019. Fast and accurate classification with a multi-spike learning algorithm for spiking neurons. In *IJCAI*, 1445–1451.
- Xu, Q.; Qi, Y.; Yu, H.; Shen, J.; Tang, H.; and Pan, G. 2018. Csn: An augmented spiking based framework with perceptron-inception. In *IJCAI*, 1646–1652.
- Yu, Q.; Tang, H.; Tan, K. C.; and Li, H. 2013. Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns. *Plos one* 8(11):e78318.
- Yu, Q.; Yao, Y.; Wang, L.; Tang, H.; Dang, J.; and Tan, K. C. 2019. Robust environmental sound recognition with sparse key-point encoding and efficient multi-spike learning. *arXiv preprint arXiv:1902.01094*.
- Yu, Q.; Li, H.; and Tan, K. C. 2018. Spike timing or rate? neurons learn to make decisions for both through threshold-driven plasticity. *IEEE transactions on cybernetics* 49(6):2178–2189.
- Yu, Q.; Wang, L.; and Dang, J. 2018. Efficient multi-spike learning with tempotron-like ltp and psd-like ltd. In *International Conference on Neural Information Processing*, 545–554. Springer.