The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

MPD-AL: An Efficient Membrane Potential Driven Aggregate-Label Learning Algorithm for Spiking Neurons

Malu Zhang,[1,2] **Jibin Wu,**[2*] **Yansong Chua,**[3*]
Xiaoling Luo,[1] **Zihan Pan,**[2] **Dan Liu,**[4] **Haizhou Li**[2,3]
[1]School of Computer Science and Engineering, University of Electronic Science and Technology of China, China
[2]Department of Electrical and Computer Engineering, National University of Singapore, Singapore
[3]Institute for Infocomm Research, A*STAR, Singapore
[4]JD AI Research
{maluzhang, jibin.wu, panzihan}@u.nus.edu, chuays@i2r.a-star.edu.sg, cdliudan@jd.com, haizhou.li@nus.edu.sg

Abstract

One of the long-standing questions in biology and machine learning is how neural networks may learn important features from the input activities with a delayed feedback, commonly known as the temporal credit-assignment problem. The aggregate-label learning is proposed to resolve this problem by matching the spike count of a neuron with the magnitude of a feedback signal. However, the existing threshold-driven aggregate-label learning algorithms are computationally intensive, resulting in relatively low learning efficiency hence limiting their usability in practical applications. In order to address these limitations, we propose a novel membrane-potential driven aggregate-label learning algorithm, namely MPD-AL. With this algorithm, the easiest modifiable time instant is identified from membrane potential traces of the neuron, and guild the synaptic adaptation based on the presynaptic neurons' contribution at this time instant. The experimental results demonstrate that the proposed algorithm enables the neurons to generate the desired number of spikes, and to detect useful clues embedded within unrelated spiking activities and background noise with a better learning efficiency over the state-of-the-art TDP1 and Multi-Spike Tempotron algorithms. Furthermore, we propose a data-driven dynamic decoding scheme for practical classification tasks, of which the aggregate labels are hard to define. This scheme effectively improves the classification accuracy of the aggregate-label learning algorithms as demonstrated on a speech recognition task.

Introduction

Traditional artificial neural networks have made remarkable achievements in recent years as demonstrated in a wide range of applications (LeCun, Bengio, and Hinton 2015). These rate-based computational models despite brain inspired, are lacking in several other aspects of biological realism, one being the presence of spiking activities (Bengio et al. 2015). Although it is widely accepted that the graded activation of these analog neurons is equivalent to the firing rate of spiking neurons inside the biological neural network (Rueckauer et al. 2017), the view that precise spike timing encodes information in the temporal domain that is important for neural computation is just as prominent in the neuroscience community (Gerstner and Kistler 2002; Gütig and Sompolinsky 2006). Consequently, the spiking neuron models are proposed to describe the dynamic of spike generation process, such that the additional temporal information including precise spike timing and phase could be better captured (Kasabov et al. 2013).

Although spiking neural networks (SNNs) offer a computing paradigm with the promise of emulating the human brain, its potential in terms of computing powers and applications are yet to be exploited. Despite the exact learning mechanisms in the brain remain unknown, many spike-based learning algorithms have been proposed for pattern recognition tasks. Depending on the number of target spikes, these algorithms can be broadly categorized into single- or multi-spike based.

One well-known single-spike learning algorithm is the Tempotron (Gütig and Sompolinsky 2006), whereby the output neuron is trained to fire a single spike in response to the correct input patterns and remain silent otherwise. The Rank-Order learning algorithms (Kasabov et al. 2013) update synaptic weights based on the rank order of the arrival time of incoming spikes. Additionally, the time-to-first spike decoding scheme is employed at the output layer. The Spike-Prop and its improved variants (Bohte, Kok, and La Poutre 2002) are other prominent single-spike based learning algorithms, extending the traditional gradient descent algorithm to SNNs. The objective function is constructed from the distance between the actual and the desired firing time of the output neuron, such that the output neuron is trained to generate a spike at the desired time.

Although single spike learning algorithms have been applied in several applications (Thorpe, Delorme, and Van Rullen 2001), only one output spike limits the robustness and amount of information that may be transmitted across the SNN. Specifically, the presynaptic spikes arriving after the output spike are discarded, while making a decision only based on a single local temporal feature. In order to overcome the aforementioned limitations, multi-spike based learning algorithms have been proposed to train neurons to emit multiple spikes with precise timing.

---

[*] Jibin Wu and Yansong Chua are the corresponding authors

The Remote Supervised Method (ReSuMe) (Ponulak and Kasiński 2010) is one such example of multi-spike learning algorithms, in which synaptic weight updates are based on a combination of spike time-dependent plasticity (STDP) and anti-STDP rules; whereas, the Chronotron E-learning (Florian 2012) and the Spike Pattern Association Neuron (SPAN) (Mohemmed et al. 2012) learning rules transform the desired and the actual output spike trains into analog signals, and then apply standard learning methods to reduce the discrepancy between the desired and the actual spike trains. Recently, the membrane-potential driven algorithms are emerging with instances include the PBSNLR (Xu, Zeng, and Zhong 2013), HTP (Memmesheimer et al. 2014), EMPD (Zhang et al. 2018b) and MemPo-Learn (Zhang et al. 2018a), whereby the loss is computed based on the postsynaptic membrane potential rather than the spike timing. In general, the membrane-potential driven algorithms are more efficient in learning a precise spike train than the other algorithms. However, they require that the neurons reset their membrane potential at the desired output spike times, which run contrary to the neuronal dynamics during learning (Memmesheimer et al. 2014).

To forecast opportunities or dangers, a neural clue (feature) detector should fire whenever the clue occurs and remain quiescent otherwise (Gütig 2016). If the timing of the clues is known, the aforementioned multi-spike learning algorithms may be used to learn these desired responses. Typically, however, the timing of the clues is commonly unavailable; setting the desired output spike times hence becomes challenging. To resolve this problem, Gütig (Gütig 2016) has proposed an aggregate-label learning algorithm, namely Multi-Spike Tempotron (MST). This algorithm trains a neuron to output a desired number of spikes that is proportional to the number of available clues, without considering the precise timing of spikes. The MST employs the gradient descent strategy, which uses a threshold-driven method for each iteration of the synaptic update. Within each iteration, the difference between the actual and desired number of spikes is instead replaced with the distance between the fixed biological firing threshold ϑ and the closest hypothetical threshold ϑ^* (at which the neuron would fire a desired number of spikes). However, this transformation makes the learning process indirect and increases the computational complexity significantly. Despite Yu et. al. have simplified the recursive gradient computation of the MST (Yu, Li, and Tan 2018) (we refer to to this algorithm as TDP1 henceforth), the necessary computation of ϑ^* is still time-consuming.

In this paper, to improve the learning efficiency of the aggregate-label learning, we proposed a membrane-potential driven aggregate-label learning algorithm, namely MPD-AL. In contrast to the existing threshold-driven learning algorithms, the proposed MPD-AL algorithm constructs an error function based on the membrane potential trace and the fixed firing threshold ϑ of the neuron. The experimental results demonstrate that the learning speed of the proposed MPD-AL algorithm is faster than the TDP1 and the MST algorithms. In addition, we propose a dynamic decoding scheme for aggregate-label learning. Instead of enforcing a fixed number of desired output spikes across all classes, we propose to dynamically determine the number of desired output spikes for each class in a data-driven manner. Hence, it makes better use of the available input information to identify the underlying clues and improves the classification accuracy as demonstrated on a speech recognition task.

Neuron Model and Learning Algorithm

In this section, we begin by presenting the neuron model used in this work. Then, we describe the proposed MPD-AL learning algorithm. Finally, we compare the proposed MPD-AL algorithm with other existing aggregate-label learning algorithms and highlight their differences.

Neuron Model

In this work, we employ the current-based leaky integrate-and-fire neuron model (Gütig 2016). This model is biological realistic and yet mathematically tractable compared to other more detailed neuron models.

We consider a neuron connected with N spiking input afferents, whose postsynaptic membrane potential is represented by $V(t)$. The membrane potential $V(t)$ is initialized at the resting potential $V_{rest} = 0$ when there is no spike received from the presynaptic neurons. Each incoming spike from the presynaptic neurons will induce a postsynaptic potential (PSP) at the postsynaptic neuron, which are integrated over time. The postsynaptic neuron will fire a spike once its membrane potential $V(t)$ reaches the firing threshold ϑ from below. The post-synaptic membrane potential $V(t)$ of the neuron can be expressed as

$$V(t) = V_{rest} + \sum_{i}^{N} \omega_i \sum_{t_i^j < t} K(t - t_i^j) - \vartheta \sum_{t_s^j < t} \exp\big(-\frac{t - t_s^j}{\tau_m}\big) \tag{1}$$

where t_i^j denotes the spike time of the jth spike from the afferent i, and ω_i is the corresponding synaptic weight. Here, K denotes the normalized PSP kernel, which is defined as follows

$$K(t - t_i^j) = V_0\big(\exp(\frac{-(t - t_i^j)}{\tau_m}) - \exp(\frac{-(t - t_i^j)}{\tau_s})\big) \tag{2}$$

the shape of the PSP is jointly governed by the membrane and synaptic time constants τ_m and τ_s, respectively. $K(t - t_i^j)$ is a causal filter that only considers spikes arriving at $t_i^j \leq t$. The coefficient V_0 normalizes PSP so that the maximum value of the PSP kernel is 1. The last term in Eq. 1 is the refractory kernel which occurs at each post-synaptic spike. t_s^j denotes the time of the jth spike emitted by the postsynaptic neuron.

MPD-AL Learning Algorithm

The goal of the proposed learning algorithm is to modify the synaptic weights of the neuron so that the trained neuron fires the desired number of output spikes in response to a given input spike pattern. This goal is violated either when the actual output spike count N_o is less than the desired number N_d (i.e., $N_d > N_o$), or the neuron fires more spikes than the desired number (i.e., $N_o > N_d$). To address these two distinct scenarios, we propose two different learning mechanisms, which we describe as follows.

Firing less spikes than desired ($N_d > N_o$) When the desired output spike count is more than the actual output spike count, the synaptic weights should be strengthened to increase the current spike count iteratively till $N_d = N_o$. To achieve this goal, as shown in Fig. 1a, we identify the easiest modifiable time instant t^*, at which the membrane potential $V(t^*)$ is the maximum among all peaks of subthreshold membrane potential. Then, as shown in Eq. 3 and 4, the gradient descent method is applied to increase the membrane potential $V(t^*)$ towards the firing threshold ϑ with the aim of generating one more spike,

$$E = \frac{1}{2}(V(t^*) - \vartheta)^2 \tag{3}$$

$$\Delta\omega_i = -\lambda_1 \frac{dE}{d\omega_i} = -\lambda_1 \big(V(t^*) - \vartheta\big) \frac{dV(t^*)}{d\omega_i} \tag{4}$$

whereby λ_1 is the learning rate which defines the magnitude of synaptic updates. Since $V(t^*)$ depends on the synaptic weight also through previous output spike times $t_s^j < t^*, j \in \{1, 2, ..., m\}$. Therefore, $dV(t^*)/d\omega_i$ is given by

$$\frac{dV(t^*)}{d\omega_i} = \frac{\partial V(t^*)}{\partial \omega_i} + \sum_{j=1}^{m} \frac{\partial V(t^*)}{\partial t_s^j} \frac{\partial t_s^j}{\partial \omega_i} + \frac{\partial V(t^*)}{\partial t^*} \frac{\partial t^*}{\partial \omega_i} \tag{5}$$

The last term in Eq.5 has no contribution to the synaptic update since $V(t^*)$ is either a local maximum with $\partial V(t^*)/\partial t^* = 0$ or t^* is the time of an inhibitory input spikes whose arrival time does not depend on ω_i. From Eq. 1, the first term of Eq. 5 can be expressed as

$$\frac{\partial V(t^*)}{\partial \omega_i} = \sum_{t_i^j < t^*} K(t^* - t_i^j) \tag{6}$$

For the second term of Eq. 5, applying the chain rule, we get

$$\frac{\partial V(t^*)}{\partial t_s^j} \frac{\partial t_s^j}{\partial \omega_i} = \frac{\partial V(t^*)}{\partial t_s^j} \frac{\partial t_s^j}{\partial V(t_s^j)} \frac{\partial V(t_s^j)}{\partial \omega_i} \tag{7}$$

where $\partial V(t^*)/\partial t_s^j$ and $\partial V(t_s^j)/\partial \omega_i$ are evaluated as below

$$\frac{\partial V(t^*)}{\partial t_s^j} = -\frac{\vartheta}{\tau_m} \exp\left(-\frac{t^* - t_s^j}{\tau_m}\right) \tag{8}$$

$$\frac{\partial V(t_s^j)}{\partial \omega_i} = \sum_{t_i^j < t_s^j} K(t_s^j - t_i^j) \tag{9}$$

According to the linear assumption of threshold crossing (Bohte, Kok, and La Poutre 2002; Yu, Li, and Tan 2018), we get

$$\frac{\partial t_s^j}{\partial V(t_s^j)} = -\left(\frac{\partial V(t_s^j)}{\partial (t_s^j)}\right)^{-1} \tag{10}$$

where

$$\begin{aligned} \frac{\partial V(t_s^j)}{\partial t_s^j} &= \frac{\partial V(t)}{\partial t}\Big|_{t=t_s^{j-}} \\ &= \frac{1}{\tau_s} \sum_{i=1}^{N} \omega_i V_0 \sum_{t_i^j < t_s^j} \exp\left(-\frac{t_s^j - t_i^j}{\tau_s}\right) \\ &\quad - \frac{1}{\tau_m} \sum_{i=1}^{N} \omega_i V_0 \sum_{t_i^j < t_s^j} \exp\left(-\frac{t_s^j - t_i^j}{\tau_m}\right) \\ &\quad + \frac{\vartheta}{\tau_m} \sum_{t_s^m < t_s^j} \exp\left(-\frac{t_s^j - t_s^m}{\tau_m}\right) \end{aligned} \tag{11}$$

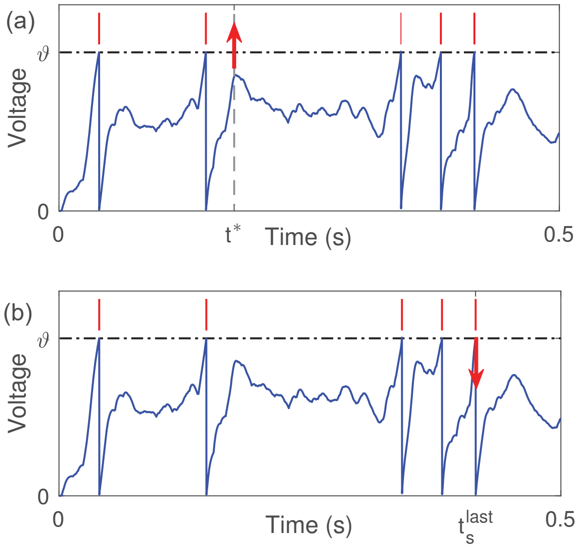


Figure 1: Demonstration of learning processes. (a) For $N_d > N_o$, the modification is performed at time t^* with the aim to generate one more spike. The subthreshold membrane potential $V(t^*)$ is the local maximum that closest to the firing threshold ϑ. (b) For $N_d < N_o$, the modification is performed at the last output spike time t_s^{last} with the aim to remove it.

Firing more spikes than desired ($N_o > N_d$) When the learning neuron fires more spikes than the desired spike count, the synaptic weights are weakened to reduce the output spike count. To minimize the changes to the membrane potential trace, as shown in Fig. 1b, the modification is performed at the last output spike time with the aim of removing it. This choice is desirable as a modification to intermediate spikes could cause catastrophic effects to the spikes emitted immediately after it. Assuming that the last output spike is emitted at t_s^{last}, the error function is constructed as

$$E = V(t_s^{last}) - \vartheta \tag{12}$$

Likewise, applying the gradient descent leads to the following update rule

$$\Delta\omega_i = -\lambda_2 \frac{dE}{d\omega_i} = -\lambda_2 \frac{dV(t_s^{last})}{d\omega_i} \tag{13}$$

where λ_2 is the learning rate. Since $V(t_s^{last})$ depends on the synaptic weight also through the previous spike times $t_s^j \leq t_s^{last}, j \in \{1, 2, ..., m\}$; its derivative can be expressed as

$$\frac{dV(t_s^{last})}{d\omega_i} = \frac{\partial V(t_s^{last})}{\partial \omega_i} + \sum_{j=1}^{m} \frac{\partial V(t_s^{last})}{\partial t_s^j} \frac{\partial t_s^j}{\partial \omega_i} \qquad (14)$$

The Eq. 14 is then solved using the same treatments as that for Eq. 5.

Comparison between the MPD-AL and other Aggregate-Label Learning Algorithms

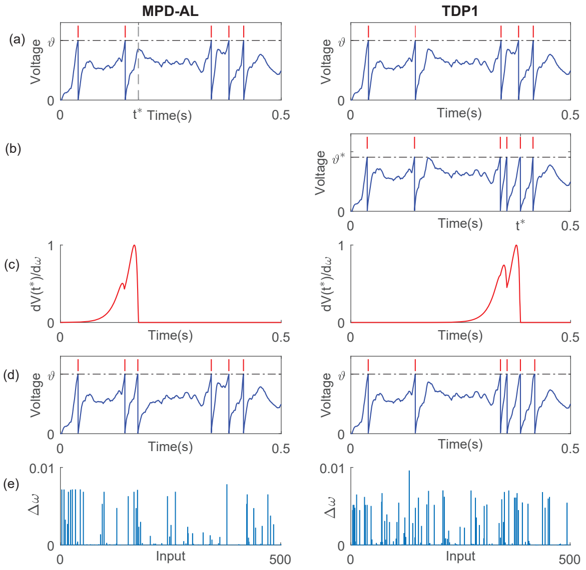


Figure 2: The comparison between MPD-AL and TDP1. (a) Membrane potential traces before learning. t^* denotes the time instant at which the subthreshold membrane potential is closest to the firing threshold. (b) The membrane potential trace of the neuron with spikes generated at the hypothetical threshold ϑ^*. Here, the t^* corresponds to the time instant at which the membrane potential reaches ϑ^*. (c) The shape of the learning curves, depicting the spike-timing dependence of a synapse's contribution to the $dV(t^*)/d\omega$. (d) Membrane potential traces after learning. (e) The accumulated weight modifications during the learning process.

The existing aggregate-label learning algorithms (MST and TDP1) are all threshold-driven. These algorithms modify the synaptic weights based on the error between the fixed firing threshold ϑ and the hypothetical threshold ϑ^*. Therefore, it is necessary to determine the ϑ^* for each iteration of the synaptic update. For example, as shown in Fig. 2a (right), the neuron fires five spikes before learning while the desired spike count is six. The threshold-driven algorithms will first numerically determine the critical threshold ϑ_6^*, with which the neuron fires exactly six spikes as shown in Fig. 2b (right). However, the ϑ^* can not be analytically obtained; therefore, ϑ^* is numerically obtained using interval halving, which is very time-consuming. In contrast, the proposed MPD-AL adopts a more direct approach in which the membrane potential (instead of threshold) is considered as the relevant signal for synaptic updates. As shown in Fig. 2b (left), MPD-AL need not numerically determine ϑ^*. Instead, the learning is directly driven by the error between the firing threshold ϑ and the membrane potential, whose value is easily accessible.

In Fig. 2c, the learning curves depict the spike-timing dependence of a synapse's contribution to the $dV(t^*)/d\omega$, which highlights the differences between the MPD-AL and the TDP1. In both sub-figures, the weight updates are governed by these learning curves with the aim to increase the membrane potential at t^*. According to the definition of t^* in the MPD-AL, $V(t^*)$ is the subthreshold membrane potential closest to the firing threshold, which makes the synaptic update a straight-forward process. In contrast, the TDP1 rule attempts to potentiate the synaptic weights at a different t^* that is typically further away from the threshold, hence requiring more changes. This is shown in Fig. 2e, where MPD-AL requires fewer modifications to allow the neuron to generate a desired number of spikes.

Experimental Results

Learning to fire a desired number of spikes

In this section, we first introduce a learning task to demonstrate the ability of the proposed MPD-AL algorithm to teach a neuron to fire a desired number of spikes. Next, we compare the learning efficiency of the MPD-AL against other competitive aggregate-label learning algorithms.

In the first experiment, a single output neuron with $N = 500$ afferents is trained to fire 10 spikes under different input firing rates. The input spike patterns are generated over a time window of $T = 500$ ms and each presynaptic neuron fires at a Poisson firing rate $r_{in} = 5$ Hz (Fig. 3a, 3b) and 20 Hz (Fig. 3c, 3d). The initial synaptic weights are drawn randomly from a Gaussian distribution with a mean of 0.01 and a standard deviation of 0.01.

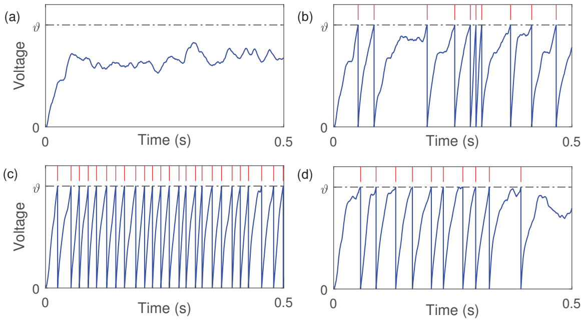


Figure 3: Neuron's membrane potential traces before (a, c) and after learning (b, d). The neuron is trained to emit 10 desired spikes under different initial firing rates, i.e., (a, b) $r_{in} = 5$ Hz and (c, d) $r_{in} = 20$ Hz.

The Fig. 3a depicts the neuron's membrane potential trace before learning, in which no output spike initially. After training, the neuron successfully emits the desired number of spikes, distributing throughout the simulation as shown in Fig. 3b. In contrast, Fig. 3c depicts responses of the neuron with an input firing rate of 20 Hz, in which the neuron exhibits bursting behavior at the beginning. Likewise, as shown in Fig. 3d, the MPD-AL algorithm adjusts the synaptic weights so that the trained neuron emits the desired number of spikes, much less than the initial bursting activities.

Furthermore, we compare the learning efficiency of different algorithms with the desired spike counts varying from 2 to 20 at an interval of 2. The input firing rate for presynaptic neurons is set to 5Hz. For each setup, 20 experiments are carried out with randomly generated input spike patterns and initial synaptic weights. The average number of learning epoch and CPU time used for each setup are calculated and reported in Fig. 4.

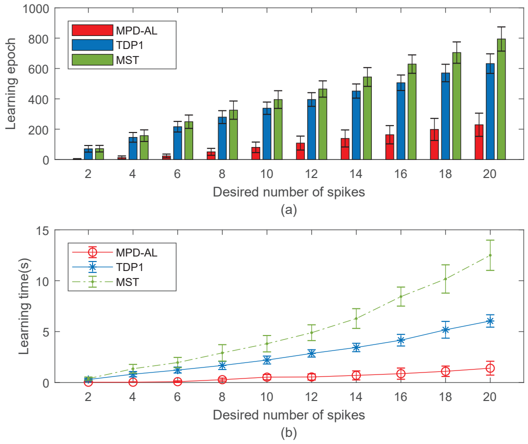


Figure 4: The comparison among MPD-AL, TDP1 and MST algorithms in terms of learning efficiency, i.e., (a) the required number of learning epochs and (b) the required CPU time.

As shown in Fig. 4a, due to the nature of the iterative optimization scheme employed, the required number of training epochs grow for all the algorithms with increasing spike count. The proposed MPD-AL algorithm consistently outperforms other algorithms and requires the lowest amount of training epochs as well as CPU time. Specifically, when the desired spike count is 20, the proposed MPD-AL algorithm requires about 250 learning epochs, while the TDP1 and the MST require more than 600 and 800 learning epochs, respectively. As shown in Fig. 4b, the average CPU time required for the MPD-AL algorithm is about 2s for the desired spike count of 20, while the TDP1 and the MST algorithms require 6s and 13s, respectively. These results could be explained as below: firstly, as illustrated in Fig. 4a, the MPD-AL algorithm requires fewer learning epochs to reach the desired spike count due to the more direct adaptation strategy taken. On the other hand, both the TDP1 and MST need to numerically determine the ϑ^*, which requires more computation times at each learning epoch. Consequently, the MPD-AL offers a better learning efficiency.

Learning Predictive Clues

To learn predictive sensory clues, an organism must bridge the gap between the time when a clue occurs and when the feedback arrives (Gütig 2016). The temporal credit-assignment problem is a core challenge in both cognition and machine learning. In this section, we will demonstrate that the temporal credit-assignment problem can be solved using the proposed MPD-AL algorithm.

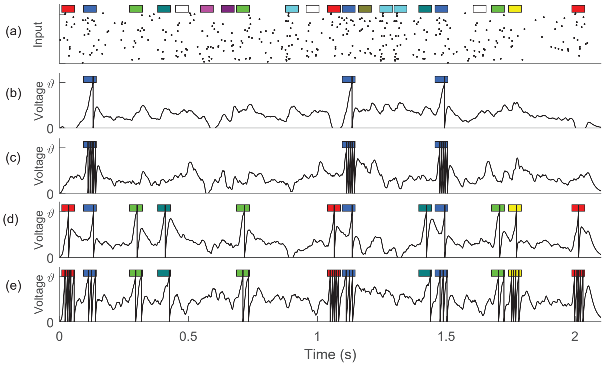


Figure 5: Learning predictive clues. (a) Input spike pattern, of which only 10% of the 500 synaptic afferents are shown. Colored rectangles represent 10 different sensory features (50 ms each) that are embedded within a stream of spontaneous background activities. (b) and (c) show the membrane potential traces after being trained to fire one or a burst of five spikes in response to a single clue (blue rectangle). (d) and (e) show the membrane potential traces after being trained to selectively respond to only five clues out of the total ten feature patterns. The neuron is expected to fire only a single spike (d) or distinct number of spikes $\{1, 2, 3, 4, 5\}$ (e) in response to each clue.

Similar to the tasks proposed in (Gütig 2016), ten brief (50ms) spiking patterns are constructed from 500 afferents with a firing rate of $r_f = 4$ Hz, which may represent spiking activities in response to perceived features from the environment. A certain number of these ten spike patterns will be designated as useful clues, while the rest as distractors. The single neuron connected to these sensory afferents is expected to signal all the useful features by firing a specific number of spikes. In contrast, the neuron should remain silent in the event of background activities or distractors. In each experimental trial, these ten spike patterns are sparsely embedded in background spike activity with duration T_b. The number of occurrence of each such pattern is drawn from a Poisson distribution with mean P_m. In order to better simulate non-stationarity of the environment, the background spike activities are generated according to a spontaneous firing rate varying from 0 to 4 Hz (with mean rate 2 Hz). During the training phase, we set the value of P_m and T_b to 0.1 and 500 ms, respectively. In the testing phase, these two values are set to 2 and 1,000 ms correspondingly to allow more exposure to both clues and distractors. For training, we generate 100 samples based on the above described method.

We first train the neuron to detect a single clue i among 9 distractors and background activities. In each trial, the desired spike count N_d is set to the number of clue event i ($N_d = c_i$). Whenever the neuron fires greater or fewer spikes than the desired count N_d, the proposed MPD-AL learning algorithm will weaken or potentiate the synaptic connections, respectively, so as to reach the desired spike count N_d. Fig. 5b illustrates the testing result, in which the neuron fires exactly one spike whenever the corresponding clue activity pattern (blue rectangle) occurs and remains silent otherwise. The N_d is not necessarily equal to the number of clue events since each clue may trigger more than one spike. To simulate this scenario, we design a task where the neuron is required to fire a burst of 5 spikes to the clue i ($N_d = 5c_i$) and remain silent otherwise as demonstrated in Fig. 5c. Intriguingly, the proposed learning algorithm enables the trained neuron to decompose the feedback signal and associate each clue with a distinct desired spike count, such that $N_d = \sum_i c_i d_i$, whereby c_i is the number of clue events i within a trial and d_i is the corresponding desired spike count belonging to that clue i. Fig. 5d and 5e show the testing results of these challenging scenarios, where the d_i values for the five clues are set as $\{1, 1, 1, 1, 1\}$ and $\{1, 2, 3, 4, 5\}$, respectively. The experimental results highlight the capabilities of the proposed MPD-AL learning algorithm to decompose the delayed feedback signal.

Application to Speech Recognition

SNNs transform spatiotemporal spike patterns into desired output patterns, and are hence well suited for processing temporally rich signals, for instance, motion and speech recognition. However, the lack of efficient learning algorithms limits application of SNN to mostly small toy problems that do not fundamentally involve spatiotemporal spike time computations (Zenke and Ganguli 2018). Although some promising learning algorithms (Lee, Delbruck, and Pfeiffer 2016; Neftci et al. 2017) have been proposed recently, they are applied to datasets of static images. Therefore, the temporal pattern classification tasks remain a challenging topic warranting further investigation.

To demonstrate the capability of the MPD-AL learning algorithm in a temporal pattern classification task, we apply it to solve a speaker-independent spoken digit classification task. The TIDIGITS corpus (Leonard and Doddington 1993) is investigated in this work, which is one of the most common datasets for benchmarking speech recognition algorithms (Wu, Chua, and Li 2018; Wu et al. 2018). This dataset consists of isolated spoken digit strings from a vocabulary of 11 words (i.e., ‘zero’ to ‘ nine” and ‘oh’) and speakers from 22 different dialectical regions. In this experiment, we use the standard training and testing sets, consisting of 2464 and 2486 speech utterances, respectively.

Firstly, the raw speech waveform is filtered by a Constant-Q Transform (CQT) cochlear filter bank (Pan et al. 2018) to extract the spectral information. Here, we use 20 cochlear filters ranging from 200 Hz to 8 KHz. Next, as shown in Fig. 6, the threshold-coding mechanism (Gütig and Sompolinsky 2009) has been applied to convert incoming speech signals into spike patterns.

The encoded spike patterns are transmitted to the next layer for classification; there are eleven groups of output neurons in this layer with each group corresponding to one class. Each group consists of ten neurons. In order to discriminate between different spoken digits, neurons are trained to generate the desired number of spikes (N_d) only when a spike pattern from their assigned class is presented, and remain silent otherwise. However, how the value of N_d is set remains an open question. On the one hand, each spoken digit is formed by different number of sub-patterns and hence one common N_d for all classes is undesirable. On the other hand, a small value of N_d limits the use of available local temporal features, while a large N_d value is prone to the overfitting problem. To resolve this problem, we propose a data-driven dynamic decoding scheme.

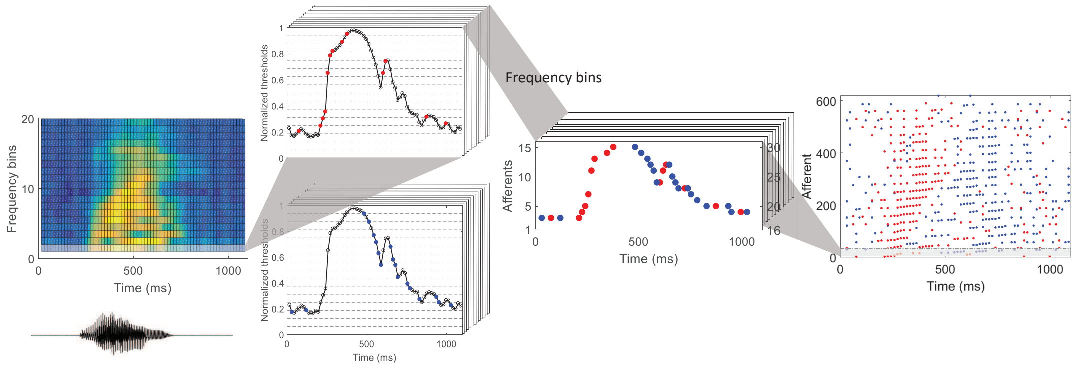
When a training spike pattern is presented, we observe the membrane potential trace of the corresponding output neuron. The N_d is decided based on the following three cases:

- If the membrane potential $V(t)$ remains lower than the firing threshold ϑ and no spike is generated (as shown in Fig. 7(a)), then $N_d \leftarrow 1$.
- If the neuron generates N_o spikes and $V(t^*)$ is above the pre-defined encoding threshold ϑ_e (as shown in Fig. 7(b)), then $N_d \leftarrow N_o + 1$.
- If the neuron generates N_o spikes and $V(t^*)$ is below the pre-defined encoding threshold ϑ_e (as shown in Fig. 7(c)), then $N_d \leftarrow N_o$.

As shown in Table.1, it is encouraging to note that the proposed MPD-AL algorithm with dynamic decoding achieves an accuracy of 97.52%, outperforming all other bio-inspired baseline systems. Moreover, to investigate the effectiveness of the proposed dynamic decoding strategy, we perform experiments with fixed $N_d \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ and the best accuracy of 95.35% is achieved when N_d = 3. This result highlights the effectiveness of the dynamic decoding strategy, which could be applied for all temporal classification tasks. In this work, the classification performance is improved by training with margins and spike time noise (Gütig 2016). It is worth mentioning that the accuracy of this work can still be improved when the model is scaled up with hidden layers using techniques proposed in (Lee, Delbruck, and Pfeiffer 2016; Neftci et al. 2017).

Table 1: Comparison of the proposed framework against other baseline frameworks.

Model	**Accuracy**
Single-layer SNN and SVM (Tavanaei and Maida 2017b)	91.00%
Spiking CNN and HMM (Tavanaei and Maida 2017a)	96.00%
AER Silicon Cochlea and SVM (Abdollahi and Liu 2011)	95.58%
Auditory Spectrogram and SVM (Abdollahi and Liu 2011)	78.73%
AER Silicon Cochlea and Deep RNN (Neil and Liu 2016)	96.10%
Liquid State Machine (Zhang et al. 2015)	92.30%
MPD-AL with N_d = 3	95.35%
MPD-AL with Dynamic Decoding	97.52%



(a) (b) (c) (d)

Figure 6: Illustration of the threshold coding mechanism. (a) The raw speech waveform and the spectrogram generated from the CQT cochlear filter bank. (b) The spectrogram is further encoded into spikes using the threshold coding. The top and bottom sub-figures depict the upward (red dots) and downward (blue dots) crossing events, respectively. For better visualization, only the output from the 1st cochlear filter is displayed. (c) The upward and downward events from (b) are merged to visualize the neuronal activation trajectory. The upward and downward crossing events for the 1st cochlear filter are represented by afferents 1-15 and 16-30, respectively. The threshold coding preserves temporal dynamics of the filtered spectral information. (d) The entire threshold-encoded spike pattern by concatenating the spike events from (c) vertically. The spike events that corresponds to the first filter in (c) is shaded in grey.

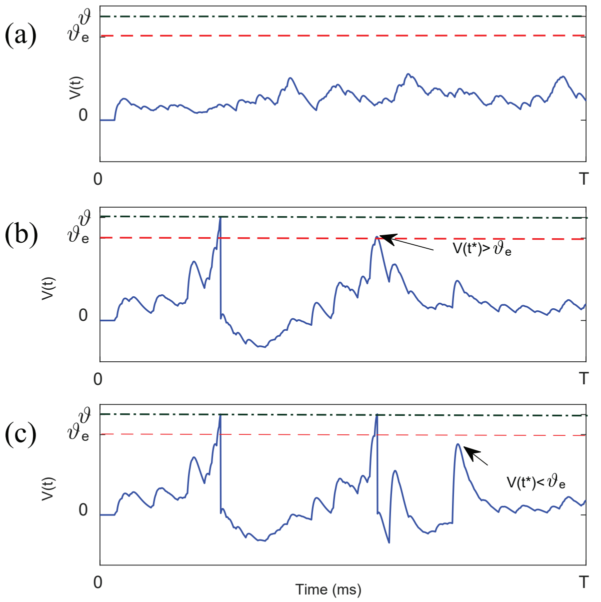


Figure 7: Illustration of the dynamic output decoding strategy.

Discussion and Conclusion

In this paper, we propose a novel aggregate-label learning algorithm MPD-AL to tackle the challenging temporal credit assignment problem efficiently. In contrast to existing aggregate-label learning algorithms including MST and TDP1, we adopt a membrane-potential driven approach instead of the threshold-driven one, where the easiest modifiable time instants are identified from the membrane potential traces and used to update the synaptic weights. This more direct training scheme has significantly improved the learning efficiency. The ability of the proposed learning algorithm to detect useful clues embedded among distractors and background spiking activities as well as to decompose an aggregate delayed feedback signal to its individual clues are demonstrated in our experiments. We further propose a data-driven dynamic decoding scheme and integrate it with our MPD-AL learning algorithm.

Although the proposed MPD-AL learning algorithm is simple and more efficient, there are areas to improve on. Firstly, MPD-AL increases the output spike count using t^* at which $V(t^*)$ is the maximum peak of the subthreshold membrane potential. However, when there is no such subthreshold peak between any two adjacent output spikes, then the learning mechanism of MPD-AL fails. Secondly, in the experiments of learning predictive clues, clues and distractors are embedded sparsely in training trials to reduce the difficulty of learning. We find that if clues and distractors are dense, the learning performance of MPD-AL decreases. Thirdly, we expect better learning performance and classification results when the proposed MPD-AL is scaled up to include multiple hidden layers. We will address these areas of improvement in future work.

Acknowledgments

This work was supported by in part by the Programmatic Grant No. A1687b0033 from the Singapore Government's Research, Innovation and Enterprise 2020 plan (Advanced Manufacturing and Engineering domain), and by the National Natural Science Foundation of China with Grant No. 61806040.

References

Abdollahi, M., and Liu, S.-C. 2011. Speaker-independent isolated digit recognition using an aer silicon cochlea. In *Biomedical circuits and systems conference (biocas), 2011 ieee*, 269–272. IEEE.

Bengio, Y.; Lee, D.-H.; Bornschein, J.; Mesnard, T.; and Lin, Z. 2015. Towards biologically plausible deep learning. *arXiv preprint arXiv:1502.04156*.

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.

Florian, R. V. 2012. The chronotron: a neuron that learns to fire temporally precise spike patterns. *PloS one* 7(8):e40233.

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., and Sompolinsky, H. 2009. Time-warp–invariant neuronal processing. *PLoS Biology* 7(7):e1000141.

Gütig, R. 2016. Spiking neurons can discover predictive features by aggregate-label learning. *Science* 351(6277):aab4113.

Kasabov, N.; Dhoble, K.; Nuntalid, N.; and Indiveri, G. 2013. Dynamic evolving spiking neural networks for on-line spatio-and spectro-temporal pattern recognition. *Neural Networks* 41:188–201.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521(7553):436.

Lee, J. H.; Delbruck, T.; and Pfeiffer, M. 2016. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience* 10:508.

Leonard, R. G., and Doddington, G. 1993. Tidigits speech corpus. *Texas Instruments, Inc.*

Memmesheimer, R.-M.; Rubin, R.; Ölveczky, B. P.; and Sompolinsky, H. 2014. Learning precisely timed spikes. *Neuron* 82(4):925–938.

Mohemmed, 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.

Neftci, E. O.; Augustine, C.; Paul, S.; and Detorakis, G. 2017. Event-driven random back-propagation: Enabling neuromorphic deep learning machines. *Frontiers in Neuroscience* 11:324.

Neil, D., and Liu, S.-C. 2016. Effective sensor fusion with event-based sensors and deep network architectures. In *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*, 2282–2285. IEEE.

Pan, Z.; Li, H.; Wu, J.; and Chua, Y. 2018. An event-based cochlear filter temporal encoding scheme for speech signals. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Ponulak, F., and Kasiński, A. 2010. Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. *Neural Computation* 22(2):467–510.

Rueckauer, B.; Lungu, I.-A.; Hu, Y.; Pfeiffer, M.; and Liu, S.-C. 2017. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience* 11:682.

Tavanaei, A., and Maida, A. 2017a. Bio-inspired multi-layer spiking neural network extracts discriminative features from speech signals. In *International Conference on Neural Information Processing*, 899–908. Springer.

Tavanaei, A., and Maida, A. S. 2017b. A spiking network that learns to extract spike signatures from speech signals. *Neurocomputing* 240:191–199.

Thorpe, S.; Delorme, A.; and Van Rullen, R. 2001. Spike-based strategies for rapid processing. *Neural Networks* 14(6-7):715–725.

Wu, J.; Chua, Y.; Zhang, M.; and Li, H. 2018. A spiking neural network framework for robust sound classification. *Frontiers in Neuroscience* 12:836.

Wu, J.; Chua, Y.; and Li, H. 2018. A biologically plausible speech recognition framework based on spiking neural networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Xu, Y.; Zeng, X.; and Zhong, S. 2013. A new supervised learning algorithm for spiking neurons. *Neural Computation* 25(6):1472–1511.

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*.

Zenke, F., and Ganguli, S. 2018. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation* 30(6):1514–1541.

Zhang, Y.; Li, P.; Jin, Y.; and Choe, Y. 2015. A digital liquid state machine with biologically inspired learning and its application to speech recognition. *IEEE Transactions on Neural Networks and Learning Systems* 26(11):2635–2649.

Zhang, M.; Qu, H.; Belatreche, A.; Chen, Y.; and Yi, Z. 2018a. A highly effective and robust membrane potential-driven supervised learning method for spiking neurons. *IEEE Transactions on Neural Networks and Learning Systems*.

Zhang, M.; Qu, H.; Belatreche, A.; and Xie, X. 2018b. Empd: An efficient membrane potential driven supervised learning algorithm for spiking neurons. *IEEE Transactions on Cognitive and Developmental Systems* 10(2):151–162.