Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks [3]

January 6, 2021

Content

Introduction

The spiking neural network (SNN) is emerging as a promising candidate for the third generation of neural networks. Compared with the previously-proposed artificial neural networks (ANNs), SNN has the advantage of temporal information processing capability, low power consumption, and high biological plausibility.

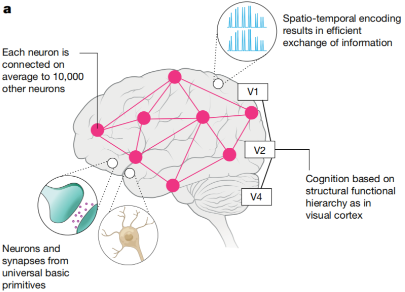


Figure 1: An overview to spiking neural networks[7]

Background

Spiking Neural Network

The spiking neural network are have an additional axis of time, compared with traditional ANNs such as CNN. As long as the input, output and hidden states of a neural network are all presented in the form of spikes, it can called as a spiking neural network.

The spiking neural network are usually composed of an encoder to transform the input into spikes and a neuron model to receive the spikes in the previous layer and emit spikes to the next layer.

Background I

Spiking Neural Network - Encoders

1. **Naive encoder** At each timestamp, a spike with magnitude of the input serves as the input of the first layer. The inputs are identical for each timestamp.

$$I(t) = \sum_k i \times \delta(t - k),$$

   where $I(t)$ is the input spike train, i is the real-valued input, $\delta(x)$ is the spike function: $\delta(x) = 0,\ \forall x \neq 0,\ \int_{\mathbb{R}} \delta(x)\mathrm{d}x = 1$.

2. **Poisson encoder** At each timestamp, there is a possibility proportional to the real-valued input that emits a unit spike.

$$I(t) = \sum_k p_k \times \delta(t - k),$$

   where p_k are sampled from Bernoulli distributions, $p_k \sim \mathrm{Bernoulli}(i)$.

Background II

Spiking Neural Network - Encoders

3. **Gaussian Harmonic encoder** The time of spike occurs are given by the following rule: given the number m of harmonic curves used, m curves are evenly distributed in the range of inputs, and the spike time are determined by the intersection of verticle line $x = i$ and all the m curves.

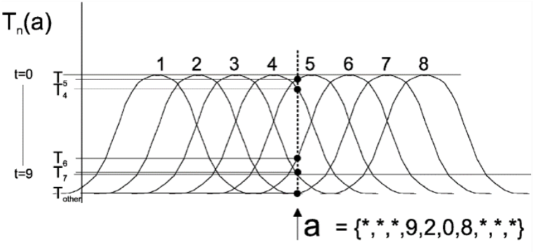


Figure 2: Illustration of Gaussian Harmonic Encoder

Background

Spiking Neural Network - Neurons

The neuron are responsible for receive spikes from previous layer and fire spikes to the next layer. The most most commonly used model is leaky integrate-and-fire (LIF) neurons.

In the terms of differential equation, the membrane potential $V(t)$ (a hidden value saved in LIF neuron) can be expressed as

$$\tau \frac{\mathrm{d}V(t)}{\mathrm{d}t} = -(V(t) - V_{reset}) + I(t) \tag{1}$$

where $V(t)$ is the membrane potential at time t, τ is the membrane constant, $I(t)$ is the input spike train and V_{reset} is the resting voltage, a steady point of the differential function if no input is fed.

Background

Spiking Neural Network - Neurons

Expanding the differential equation 1 into a recursive one, we have

$$\tau(H_{t+1} - V_t) = -(V_t - V_{reset}) + I_t \tag{2}$$

After that, we build the firing function

$$S_t = \Theta(V_t - V_{threshold}), \tag{3}$$

where $\Theta(x)$ is the Heaviside function

$$\Theta(x) = \begin{cases} 1 & x > 0, \\ 0 & x \leqslant 0, \end{cases} \tag{4}$$

and the membrane voltage in the next timestamp V_{t+1} can be expressed as

$$V_{t+1} = V_{reset} \cdot S_t + H_{t+1}(1 - S_t) \tag{5}$$

Traditional Training Methods

Unsupervised Training

The main theorem used in unsupervised learning of spiking neural networks is the spike-time dependent plasticity (STDP) [2]. The principle is derived from Hebbian Learning that the connection between two nodes which is closely correlated should be strengthened [4]. An simple explanation can be seen from the following figure:

Traditional Training Methods

Unsupervised Training

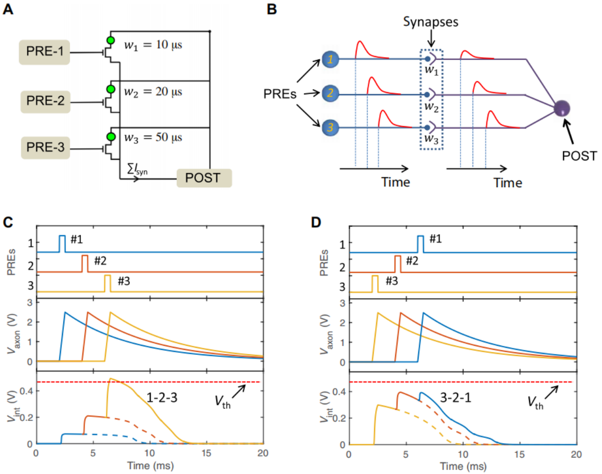


Figure 3: An Illustration of STDP

Traditional Training Methods

Unsupervised Training

From figure 3, we can see that the magnitude of the last spike before the postsynaptic fires have the most impact on the postsynaptic voltage and thus we should strengthen the connection between these two neurons, according to Hebbian's rule. For example, the synaptic weight can be updated according to the following figure

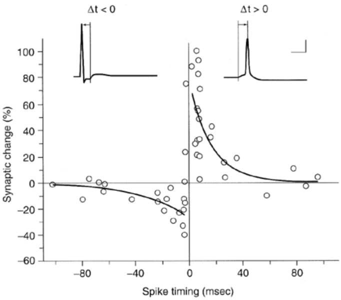


Figure 4: Update using STDP

Traditional Training Methods

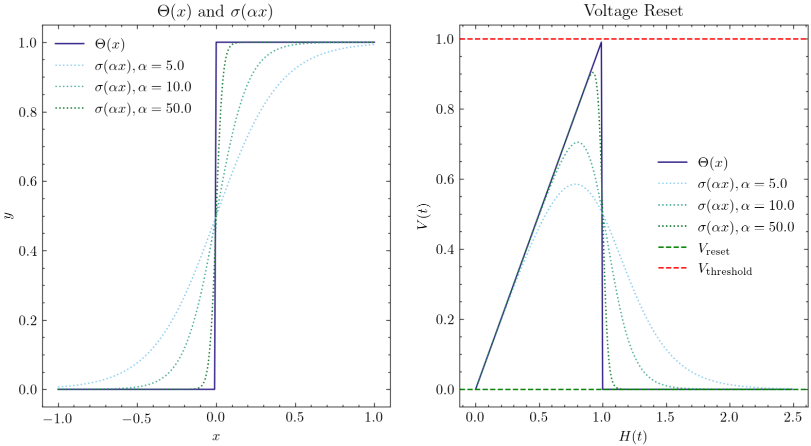
Supervised Training

The supervised training method is still based on backpropagation. The loss function are usually set as the cross-entropy of the firing rate of output neurons and the one-hot coding of the label. Revisiting equation 2 - 5, we can see that only the Heaviside function can not be differentiated.

Surrogate gradient is frequently used to calculate the gradient in backpropagation. Sigmoid function $\sigma(x;\alpha) = \dfrac{1}{1+\exp(\alpha x)}$ is a good candidate for the surrogate function of the Heaviside function.

Traditional Training Methods

Supervised Training



Methodology

While most of the previous works set the membrane constant τ as a fixed hyper-parameter or a trainable parameter that is shared all over the whole layer, the paper claims that every neuron should have different membrane time constant and call the LIF neuron with trainable τ as parametric LIF (PLIF).

Except for the medical excuse that there exists different membrane time constants for spiking neurons across brain regions, an important explanation that PLIF can boost the accuracy is that the time constant can make the membrane behave differently from those whose weights are the only trainable parameter.

Methodology

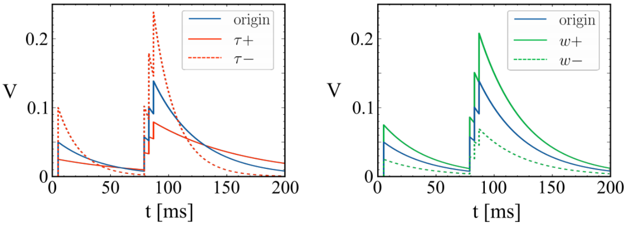


Figure 5: Effect of changing membrane time constant τ and weight

From the figure, we can see changing w can only stretch the voltage proportionally. A larger w will lead to a higher voltage at all the time and vice versa. However, changing τ behave differently. A larger τ will make the neuron charges slower and decharges slower. This leads the neuron with larger τ will have a lower voltage at first and a higher voltage later, largely enriches the behaviour of the post-synaptic voltage and thus lead to a better ability to fit.

Methodology

Since the time constant lays in the denominator, calculating the gradient for it may lead to numerical instability. Therefore, the paper use an agent parameter a to calculate the gradient by substituting $\tau = 1 + \exp(-a)$.

The process of calculating the gradient is similar to ordinary back-propagation with the help of surrogate function:

$$\frac{\partial L}{\partial a^i} = \sum_{t=0}^{T-1} \frac{\partial L}{\partial H_t^i} \frac{\partial H_t^i}{\partial \alpha^i} \tag{6}$$

where a^i is the agent parameter for time constant of the i^{th} neuron and H_t^i is the hidden state of the neuron before firing.

Experiment

Model	Method	Accuracy MNIST	Accuracy Fashion-MNIST	Accuracy CIFAR-10	Accuracy N-MNIST	Accuracy CIFAR10-DVS	Accuracy DVS128 Gesture	Accuracy Type
[24]	ANN2SNN	98.37%	-	82.95%	-	-	-	Unknow
[51]	ANN2SNN	99.44%	-	88.82%	-	-	-	Unknow
[52]	ANN2SNN	-	-	91.55%	-	-	-	A
[19]	ANN2SNN	-	-	**93.63%**	-	-	-	Unknow
[34]	Spike-based BP	99.31%	-	-	98.74%	-	-	A
[58]	Spike-based BP	99.42%	-	-	98.78%	50.7%	-	Unknow
[55]	Spike-based BP	99.36%	-	-	99.2%	-	93.64%	Unknow
[26]	Spike-based BP	-	-	-	96%	-	**95.54%**	Sub-Dataset
[25]	Spike-based BP	99.49%	-	-	98.84%	-	-	Unknow
[65]	Spike-based BP	**99.62%**	90.13%	-	-	-	-	Unknow
[59]	Spike-based BP	-	-	90.53%	**99.53%**	**60.5%**	-	A
[33]	Spike-based BP	99.59%	-	90.95%	99.09%	-	-	A
[8]	Spike-based BP	99.5%	**92.07%**	-	99.45%	-	-	Unknow
[36]	Spike-based BP	-	-	-	96.3%	32.2%	-	Unknow
[61]	Spike-based BP	-	-	-	-	-	92.01%	A
[12]	Spike-based BP	99.46%	-	-	99.39%	-	96.09% (10 classes)	A
[21]	Spike-based BP	-	-	-	98.28%	-	93.40%	Unknow
[48]	ANN2SNN and Spike-based BP	-	-	92.64%	-	-	-	Unknow
[56]	HATS	-	-	-	99.1%	52.4%	-	Unknow
[4]	GCN	-	-	-	99.0%	54.0%	-	Unknow
Ours								
Without Validation	Spike-based BP	99.72%	94.38%	93.50%	99.61%	74.80%	97.57%	A
15% Validation	Spike-based BP	99.63%	93.85%	92.58%	99.57%	69.00%	96.53%	B

Table 2. Performance comparison between the proposed method and the state-of-the-art methods on different datasets.

Figure 6: Accuracy type A: Seperating the dataset into training set and validation set, save the best result on validation set. Type B: Separating the dataset into training set, validation set and test set, use the model with best accuracy on validation set, save the accuracy on test set for only once.

Dataset

In the experiment, there are several datasets, such as N-MNIST [6], CIFAR10-DVS [5], DVS128 Gesture [1], that are modified from exist dataset or created to meet with the need of training neuromorphic models. They are mostly captured by using dynamic vision sensor (DVS) cameras, and a spike input is fired when the change in a pixel exceeds threshold. Remarkably, the positive changes and negative changes are saved in separate matrices.

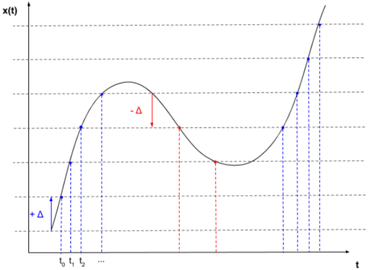


Figure 7: The firing criteria of event-based datasets

Reference I

[1] Arnon Amir et al. "A low power, fully event-based gesture recognition system". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 7243–7252.

[2] Yoshua Bengio et al. "STDP as presynaptic activity times rate of change of postsynaptic activity". In: *arXiv preprint arXiv:1509.05936* (2015).

[3] Wei Fang et al. "Incorporating learnable membrane time constant to enhance learning of spiking neural networks". In: (2020).

[4] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[5] Hongmin Li et al. "Cifar10-dvs: an event-stream dataset for object classification". In: *Frontiers in neuroscience* 11 (2017), p. 309.

Reference II

[6] Garrick Orchard et al. "Converting static image datasets to spiking neuromorphic datasets using saccades". In: *Frontiers in neuroscience* 9 (2015), p. 437.

[7] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. "Towards spike-based machine intelligence with neuromorphic computing". In: *Nature* 575.7784 (2019), pp. 607–617.