Spike Count Maximization for Neuromorphic Vision Recognition

Jianxiong Tang1, Jian-Huang Lai1,2,3*, Xiaohua Xie1,2,3 and Lingxiao Yang1,2,3

1School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
2Guangdong Province Key Laboratory of Information Security Technology, Guangzhou, China
3Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
tangjx6@mail2.sysu.edu.cn, {stsljh, xiexiaoh6, yanglx9}@mail.sysu.edu.cn

Abstract

Spiking Neural Networks (SNNs) are the promising models of neuromorphic vision recognition. The mean square error (MSE) and cross-entropy (CE) losses are widely applied to supervise the training of SNNs on neuromorphic datasets. However, the relevance between the output spike counts and predictions is not well modeled by the existing loss functions. This paper proposes a Spike Count Maximization (SCM) training approach for the SNN-based neuromorphic vision recognition model based on optimizing the output spike counts. The SCM is achieved by structural risk minimization (SRM) and a specially designed spike counting loss. The spike counting loss counts the output spikes of the SNN by using the $\ell_0$-norm, and the SRM maximizes the distance between the margin boundaries of the classifier to ensure the generalization of the model. The SCM is non-smooth and non-differentiable, and we design a two-stage algorithm with fast convergence to solve the problem. Experiment results demonstrate that the SCM performs satisfactorily in most cases. Using the output spikes for prediction, the accuracies of SCM are 2.12% ~ 16.50% higher than the popular training losses on the CIFAR10-DVS dataset. The code is available at https://github.com/TJXTT/SCM-SNN.

1 Introduction

Spiking Neural Networks (SNNs) [Tavanaei et al., 2019; Zhang et al., 2022] are bio-inspired models, and neuromorphic data [Amir et al., 2017; Li et al., 2017] is widely used for low-power vision sensing. Since the SNN transmits the information by the spike sequences, the feature maps of SNNs are binary. This advantage makes the SNN-based neuromorphic sensing techniques energy efficient on neuromorphic chips [Pei et al., 2019; Rahiminejad et al., 2022]. However, because the spiking features of SNNs are non-differentiable almost everywhere, the training of SNNs is more difficult than the Artificial Neural Networks (ANNs).

Two kinds of learning algorithms are widely used to train SNN: a) ANN-to-SNN (ANN2SNN) conversion; b) Spike-based BP training. The ANN2SNN [Bu et al., 2021; Ding et al., 2021; Deng and Gu, 2021] converts a well-trained ANN to its SNN version. Such a technique provides an efficient way to obtain a SNN from a well-trained ANN. However, the spatial-temporal context of the neuromorphic events is not well modeled by the converted SNN, making ANN2SNN mainly focus on the tasks of static images.

To improve the performance of the SNN-based neuromorphic vision tasks, the spike-based BP algorithms [Wu et al., 2018; Wu et al., 2019b; Lee et al., 2020; Deng et al., 2022a] that use the back-propagation (BP) are designed to directly train the SNNs. The surrogate gradient of the spike-based BP enables the gradient calculations of spikes w.r.t the membrane potential so that the BP algorithms can directly be applied for SNN training. Since the spike-based BP training captures the spatial-temporal dynamics of SNNs, the SNN can learn the context of the neuromorphic events. Many efficient BP-based algorithms are proposed to train the SNNs for better performance [Zheng et al., 2021; Li et al., 2021; Fang et al., 2021a; Deng et al., 2022a; Feng et al., 2022]. However, many of them use the mean square error (MSE) or cross-entropy (CE) loss on the membrane potentials of the logit layer to minimize the gap between the spatial-temporal outputs of SNN and the target spike sequences/ground truth labels. The relevance between spike counts and the correct predictions is not well described since optimizing the logit outputs does not always result in more output spike counts [Shrestha et al., 2022]. Although the spiking neurons can be applied to the output of the SNN to generate the spikes for recognition, the training error is updated based on the surrogate gradient. That is, the training process is to minimize the error between the target and differentiable surrogate, which smoothes the discreteness of the spiking output.

In this paper, we connect the output spike activities with the classification of SNNs. We assume that the output of SNN should activate as many spikes as possible for a correct prediction. Otherwise, the output neurons should be kept at rest. This assumption satisfies the spike counting strategy [Shrestha and Orchard, 2018]. On this basis, we address the SNN training problem by maximizing the number of output spikes for the correct predictions. Specifically, we design a spike counting loss to map each time step’s decision output
to a binary spike and achieve the training based on structural risk minimization (SRM). The spike counting loss maximizes the number of spikes through time, and the SRM guarantees that each output spike is activated with sufficient confidence. Our contributions can be summarized as follow:

- We propose the spike counting loss to handle the output spikes of the SNN.
- We propose the spike count maximization (SCM) approach for neuromorphic vision recognition based on the spike counting loss and structural risk minimization.
- We provide iterative solutions to the SCM and then design a two-stage SNN training algorithm.

Experiment results on the popular neuromorphic vision datasets demonstrate that the performance of the SCM is competitive with the popular training loss.

## 2 Background & Related Works

### 2.1 Neuromorphic Vision Recognition

The neuromorphic vision datasets are the event streams captured by the bio-inspired vision sensors. The Dynamic Vision Sensor (DVS) [Amir et al., 2017; Li et al., 2017; Bi et al., 2019] is the most popular neuromorphic sensor. It mimics the biological retina and generates a sparse event when a pixel value changes magnitude by a pre-setting threshold. The sparse event stream reduces the cost of energy and bandwidth for real-time transmission. In addition, the event stream’s high temporal resolution and dynamic range can provide abundant features for pattern recognition tasks. However, the discontinuous and sparse events make neuromorphic datasets much different from the CMOS-based sensing images. Many ANN-based techniques are designed for neuromorphic vision recognition [Bi et al., 2019; Wu et al., 2021; Deng et al., 2021; Deng et al., 2022b; Baldwin et al., 2022]. However, these methods’ good performance relies on the many floating point operations and real-value features, degenerating the energy efficiency on edge devices or neuromorphic chips.

### 2.2 Spiking Neural Networks

As shown in Fig. 1, the SNNs encode the input data to the spatial-temporal features. The spiking neuron is the basic component of SNN, and the Leaky Integrate-and-Fire (LIF) neuron is popular for SNN modeling. Given an input sequence \( \{o_t^j\}_{t=1}^T \), the dynamic of the LIF neuron is

\[
u^t = \tau u^{t-1} (1 - o^{t-1}) + w o^t + b, \tag{1}
\]

\[
o^t = \begin{cases} 1, & \text{if } u^t > V_{th}, \\ 0, & \text{otherwise}, \end{cases} \tag{2}
\]

where \( \tau \in (0, 1) \) is a decay factor, \( o^t \) denotes the spike and \( u^t \) is the membrane potential (MP) at \( t \), \( w \) is the weight, and \( b \) is the bias. The MP integrates the pre-synaptic inputs in a time direction, and the post-synaptic spikes are generated when the MP crosses \( V_{th} \). After that, the MP is reset to 0. Different from a ReLU-based ANN, the SNN has additional temporal dynamics. In each time step, the activation values are binary rather than a real value of the ReLU activation. The temporal dynamics of SNN enable the processes of the event stream, and the feature maps of every step are binaries. Therefore, the SNNs have more computation-efficient than the ANN for real-time neuromorphic vision recognition.

### 2.3 Loss Function for SNN Training

#### MP-based Loss

Supposing \( W \in \mathbb{R}^{d \times C} \) and \( b \in \mathbb{R}^C \) are the weights and biases of the classifier of the SNN, the Cross-Entropy (CE) and Mean Square Error (MSE) losses for the training of SNN are:

\[
\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log \left( \frac{e^{w_i^T \bar{o}_i + b_y}}{\sum_{j=1}^{C} e^{w_j^T \bar{o}_j + b_j}} \right), \tag{3}
\]

\[
\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=1}^{N} \|W^T \bar{o}_i + b - y_i\|_2^2, \tag{4}
\]

where \( w_j \in W, b_j \in b \) denote the class center and bias of \( j \), \( y_i \) is a \( C \) dimensions one-hot vector of \( y_i \), and \( \bar{o}_i = \frac{1}{T} \sum_{t=1}^{T} o^t_i \) is the average value of the feature over
time. Then, the weight of the SNN can be updated by utilizing the spike-based gradient techniques [Wu et al., 2018; Wu et al., 2019b]. The CE and MSE losses collect all decision outputs for prediction. To minimize the training error, Eq. (3) maximizes the predicted probability of sample \( i \), and Eq. (4) is to fit the targets directly. [Deng et al., 2022a] designs a Temporal Efficient Training (TET) loss that makes a decision output on each step, which re-weights the gradient of the synaptic weights to search for a flat local minimum.

**Spike-based Loss**

In neuromorphic hardware, the spiking outputs are more suitable than the MPs for the inference of SNN due to the binarity of spikes. Maximizing the logit outputs can result in more output spike counts, but it does not guarantee it always [Shrestha et al., 2022]. Some variants of Eq. (3) or (4) map the logit outputs to the binary spikes for model training. The spike rate loss (SRL) [Shrestha and Orchard, 2018; Kaiser et al., 2020] maps the logit outputs to the binary spikes for model training.

Mathematically, the spike-based cross entropy [Wu et al., 2019a; Meng et al., 2022] (SCE) applies the spiking neuron on the logit layer and maximizes the entropy value:

\[
\mathcal{L}_\text{SCE} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log \left( \frac{e^{f(w,b)\cdot y_i}}{\sum_{j=1}^{C} e^{f(w,b)\cdot y_j}} \right).
\]

Further, [Shrestha et al., 2022] proposes the SpikeMax loss, which divides the simulation time into several intervals to calculate the negative log-likelihood losses based on the probability interpretation of spikes.

Since the surrogate gradient optimizes the spike-based loss functions, the training of the output layer is to fit the target by the differentiable surrogate, smoothing the discrete spiking outputs in the training process.

### 3 Methodology

In this section, we introduce the principle of spike count maximization. Our idea is to train the classifier of SNN by maximizing the output spike counts for a correct prediction. First, we propose a spike counting loss to count the output spikes. Then, we propose the SCM based on structural risk minimization of this basis and provide its iterative solutions. Finally, we extend the SCM for multi-classification and propose a two-stage training algorithm.

#### 3.1 Spike Counting Loss

We consider the binary classification problem. Supposing \( \{w, b\mid w \in \mathbb{R}^d, b \in \mathbb{R}\} \) are the synaptic weights and bias of the classifier, and \( \{\sigma^t\}_{t=1}^{T} \) with label \( y \in \{-1, +1\} \) is a sequence of spatial-temporal features. For each step, we model the activation of the output neuron based on the hing loss with the \( \ell_0 \)-norm [Tang et al., 2018; Wang et al., 2022]:

\[
h(w,b;\sigma^t,y) = \| (V_{ih} - f(w,b;\sigma^t,y))_+ \|_0, t \in [1,T].
\]

where \( f(w,b;\sigma^t,y) = y(w^T \sigma^t + b) \), \( \| \cdot \|_0 \) is the \( \ell_0 \)-norm, \( V_{ih} > 0 \), and \((\cdot)_+ \) maps the negative values to 0. Based on Eq. (7) and Fig. 2, if \( f(w,b;\sigma^t,y) < V_{ih} \), the output neuron is rest and \( h(w,b;\sigma^t,y) = 1 \). Otherwise, the neuron fires a spike and \( h(w,b;\sigma^t,y) = 0 \).

Summarizing all \( h(w,b;\sigma^t,y) \) from \( t=1 \) to \( T \), we obtain the following Spike Counting Loss (SCL):

\[
\mathcal{H}(w,b;\{\sigma^t\}_{t=1}^{T},y) = \sum_{t=1}^{T} \| (V_{ih} - f(w,b;\sigma^t,y))_+ \|_0.
\]

It is obvious that the value of SCL (Eq. (8)) is integer and bounded by \( 0 \) and \( T \). The smaller \( \mathcal{H}(w,b;\{\sigma^t\}_{t=1}^{T},y) \), the more spikes for a prediction are fired. Ideally, a correct prediction should fire the spikes at every step, that is, \( \mathcal{H}(w,b;\{\sigma^t\}_{t=1}^{T},y) = 0 \), and the spike activities of the negative samples should keep at rest.

Different from Eq. (3), the SCL constrains the prediction of each step to satisfy \( y(w^T \sigma^t + b) \geq V_{ih} \) rather than maximizing a predicted probability or directly fitting the target. Compared to the existing spike-base loss functions, the value of SCL is discreteness.

#### 3.2 Spike Count Maximization

To ensure that SCL activates the output spikes with sufficient confidence. We model the SNN’s training problem by combining the structural risk minimization [Vapnik, 2000] with Eq. (8).

\[
\min_{w,b} \frac{\lambda}{2} \| w \|_2^2 + \mathcal{H}(w,b;\{\sigma^t\}_{t=1}^{T},y).
\]

Fig. 3 gives an example of Eq. (9) on spatial-temporal data, which has two dimension features and four inference steps. The blue hyperplanes is the classifier, the gray planes are the margin boundaries of the firing threshold \( y(w^T \sigma^t + b) = V_{ih} \). The training of regular term \( \| w \|_2^2 \) maximizes the margin \( \frac{2V_{ih}}{\| w \|_2^2} \), which separates positive and negative samples with enough confidence. \( \mathcal{H}(\cdot) \) minimizes the number of projections located in the negative direction of \( V_{ih} = y(w^T \sigma^t + b) \) (refer to the data marked by the dashed circles in Fig. 3) and is equivalent to maximizing the number of features satisfying \( y(w^T \sigma^t + b) \geq V_{ih}, t \in [1,T] \).
By replacing $A^t z$ with the potential $u^t$ and introducing the equality constraint, the optimization of the $\ell_0$-norm is independent of $z$, and the objective function of Eq. (14) can be relaxed to

$$
L(z, (u^t)_{t=1}^T) = \frac{\gamma}{2} z^\top D z + \sum_{t=1}^T g(u^t) + \sum_{t=1}^T \frac{\rho}{2} u^t - A^t z + \frac{\beta}{\rho} z^2, 
$$

(15)

where $\rho > 0$ and $\beta > 0$ are the penalty parameters. We can minimize Eq. (15) to learn the classifier $z$.

Based on the above analysis, the training of Eq. (9) is decomposed to optimize $z$ and $(u^t )_{t=1}^T$. Since the optimization of $u^t$ is non-differentiable, we minimize Eq. (15) by the block coordinate descent [Tseng, 2001; Tang et al., 2018] and have the following iterative scheme:

$$
\begin{align*}
  z^{k+1} &= \arg \min_z L(z, (u^t)^{k+1}_{t=1}^T) = \frac{\gamma}{2} z^\top D z + \frac{\lambda}{2} \|z - z^k\|^2, \\
  u^1_{t+1} &= \arg \min_{u^1} L(z^{k+1}, u^1; (u^t)^{k+1}_{t=2}^T), \\
  &\vdots \\
  u^T_{t+1} &= \arg \min_{u^T} L(z^{k+1}, (u^t)^{k+1}_{t=1}^{T-1}, u^T), 
\end{align*}
$$

(16)

where $k$ denotes the $k$-th training iteration and $\lambda > 0$ is a small enough value.

We present the solutions of Eq. (16) in Theorem 1. Compared with the surrogate gradient-based training of the existing spike-based loss functions, Theorem 1 learns the classifier by Eq. (17) and optimizes the output spikes based on Eq. (18).

**Theorem 1.** The solutions of Eq. (16) are

$$
\begin{align*}
  z^{k+1} &= (\gamma D + \rho \sum_{t=1}^T (A^t)^\top A^t + \lambda I)^{-1} \\
  &\quad \left(\sum_{t=1}^T (A^t)^\top \beta + \rho \sum_{t=1}^T (u^t)^k + \lambda z^k\right), 
\end{align*}
$$

(17)

where $a^t_i$ are $A^t z^{k+1} - \frac{\beta}{\rho}, 1 \leq i \leq N, 1 \leq t \leq T$.

We extend the SCM for multi-class classification by applying the “one-versus-all” strategy:

$$
\begin{align*}
  \min_{z, (u^t)_c} \frac{\gamma}{2} \sum_{c=1}^C z_c^\top D z_c + \sum_{c=1}^C \sum_{t=1}^T g(u^t_c), 
\end{align*}
$$

(19)

where $C$ denotes the class number, $z_c = [w^T_i b_c]^\top$, and $A^t_c = Y_c[(X^t)^\top \mathbf{1}]$. We solve each class’s $(z_c, u^t_c)$ based on Theorem 1.
Tab.1 shows the neuromorphic datasets: DVS128-GESTURE, CIFAR10-DVS, and ASL-DVS. We compare our SCM with the SNNs trained based on the population loss functions and spike-based BP algorithms. Details of the setting and results are presented in the following sections.

4 Experiments

This section estimates the SCM on the neuromorphic datasets: DVS128-GESTURE, CIFAR10-DVS, and ASL-DVS. We pre-train the SNN to train all models in Stage 1, the training epochs of DVS-G, C10-DVS, and ASL-DVS are 30, 30, and 3. For Stage 2 training, we set the iterations to 10, β = 0.01, ρ = 1, and γ ranges from 0.001 to 1000 with a step size of 10. The accuracy of all models is predicted by two metrics: a) MP Acc and b) Spike Acc. The MP Acc accumulates the MPs of the last layer for prediction, and the Spike Acc makes predictions by all output spikes. All results are averaged over 5 runs.

4.2 Performance

Comparison With Different Loss Functions

In this part, we compare the SCM with the SNNs supervised by the MP-based loss functions: CE, MSE, and TET [Deng et al., 2022a], and spike-based loss functions: SCE [Wu et al., 2019a; Meng et al., 2022], SRL [Shrestha and Orchard, 2018; Kaiser et al., 2020] with r_i = 1.0, and SpikeMax [Shrestha et al., 2022]. We use the PiecewiseLeakyReLU function [Wu et al., 2018; Wu et al., 2019b; Fang et al., 2020] as the surrogate gradient to train the SNNs. We initialize the MP-loss-based models by a SNN that is pre-trained on CIFAR10. Then, the spike-loss-based SNNs is initialized by the trained MP-loss-based SNNs. The accuracy results are shown in Tab. 2. Using output spikes for prediction degrades the performance of CE, MSE, and TET-based SNNs. Although the degeneration of the CE loss is lower than that of the MSE and TET, the accuracies are 1.82% and 8.42% lower than the MP Acc on C10-DVS and ASL-DVS, respectively. By applying the SCM to train the classifier of the CE-based SNNs, the Spike Acc on DVS-G, C10-DVS, and ASL-DVS are improved to 96.11%, 77.50%, and 90.23%, respectively.

The SCM improves the MP Acc of models trained with MP-based loss functions, although it does not directly optimize the MP. Both MP Acc and Spike Acc of the spike-based losses outperform the MP-based losses in most cases. Nevertheless, the performance of the MP-based SNNs with SCM is competitive. For C10-DVS, the SNN with “MSE+SCM” performs 1.34% better than the “SRL” on Spike Acc. The SCM also improves the performance of the SCE, SRL, and SpikeMax-based SNNs, with 0.07% ~ 0.57%, 2.74% ~ 3.36%, and 0.15% ~ 0.74% improvements over Spike Acc on DVS-G, C10-DVS, and ASL-DVS, respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Frame Size</th>
<th>Categories</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVS128-GESTURE</td>
<td>128 × 128 × 2</td>
<td>11</td>
<td>1,176</td>
</tr>
<tr>
<td>CIFAR10-DVS</td>
<td>128 × 128 × 2</td>
<td>10</td>
<td>10,000</td>
</tr>
<tr>
<td>ASL-DVS</td>
<td>120 × 240 × 2</td>
<td>24</td>
<td>211,392</td>
</tr>
</tbody>
</table>

Table 1: Details of the neuromorphic datasets.
Comparison With Difference Training Approaches

The surrogate gradient for SNN training is a potential factor affecting performance. To demonstrate that the SCM can adapt to different training approaches, we apply the two recently developed spike-based BP techniques: Gradient Rewiring (GR) [Chen et al., 2021] and Differentiable Spike (DS) [Li et al., 2021] with CE, MSE, and TET for the training of Stage 1. We initialize the models with the MP-loss-based SNNs given in Tab. 2. The experimental results are presented in Tab. 3. Compared to the MP-loss-based SNNs in Tab. 2, both GR and DS improve the Spike Acc of MP-based and Spike-based predictions in many cases. Nevertheless, the performance gap between MP-based and Spike-based predictions is still large, especially for MSE-based SNNs. All models achieve significant improvement by introducing SCM, similar to the SCM-based models in Tab. 2. For the MSE-based SNN with DS, SCM increases the average Spike and MP Acc from 62.72% and 77.14% to 79.22% and 79.74%.

4.3 Robustness & Generalization

The experiments in the following subsections are analyzed based on the C10-DVS, and MP-based prediction. We analyze the robustness and generalization of the SCM in two aspects: a) The improvement of the SCM for the baselines in every epoch; b) The performance of the SCM using different subsets of training samples. We apply the SCM to the spatial-temporal features of the SNNs of Stage 1 in every training epoch and accordingly train the classifiers for predictions. Experiment results are shown in Fig. 4. The training of Stage 1 requires many epochs to improve accuracy performance for classification. However, the SCM performs better than baseline models in every epoch. In early training epochs, the performance of the baselines is weak, but the SCM significantly improves the performance. For example, the average accuracy of the CE-based model in the first epoch is 51.18%, and the SCM increases the baseline accuracy to 62.88%, showing the robustness of the SCM.

To show the generalization of the SCM, we use the subsets of the C10-DVS with sample numbers given in {1000, 2000, 4000, 6000, 9000} to train the classifier. Experiment results are shown in Fig. 5. By using 1000/9000 sam-

Table 3: Comparison with Gradient Rewiring and Differentiable Spike training approaches on ResNet-18.
We show the convergence of the SCM based on the changing value of the penalty function, the objective function, the regular term of $z^k$, and the accuracy of each iteration. The baseline models are trained by GR, and the SCM is trained based on $\rho = 1$, $\gamma = 0.01$, and 9000 training samples. The results are shown in Fig. 7. All curves in Fig. 7(a)–7(c) decrease and converge. The decrease of the penalty function and the regularizer shows the convergence of Eq. (16) and shows that Eq. (9) can be approximated by solving Eq. (15). Based on Fig. 7(d), we find that the SCM converges fast, and the best accuracy scores of all models are 79.70% (MSE+SCM), 79.40% (CE+SCM), and 80.50% (TET+SCM).

4.5 Influence of Parameters $\gamma$ and $\rho$

We analyze the influence of $\gamma$ and $\rho$ based on the accuracy, and the results are given in Fig. 8. We fix $\rho = 1$ to show the influence of $\gamma$. For the analysis of the influence of $\rho$, we set $\gamma = 0.01$. The SCM stays stable in most cases. Based on Fig. 8(a), the fluctuations of all models are small if the $\gamma$ ranges from 0.001 to 1000. The MSE+SCM, CE+SCM, and TET+SCM achieve 79.90% ($\gamma = 1000$), 79.40% ($\gamma \leq 1000$), and 80.50% ($\gamma < 1000$), respectively. As shown in Fig. 8(b), all models keep stability if $\rho < 2$. By setting $\rho = 1.2$, the SCM+TET achieves 80.80% accuracy performance.

5 Conclusions

In this paper, we propose the SCM to train the SNN by optimizing the output spikes. We propose the spike counting loss to count the output spikes and design the two-stage algorithm for training. Experimental results on various neuromorphic datasets demonstrate the effectiveness of the SCM. Since the matrix multiplication complexity of the Stage 2 is proportional to the number of samples, the SCM is inefficient on large datasets. In addition, the two-stage strategy may be suboptimal for the SCM because the backbone is fixed to extract features for the Stage 2 training. Our future work will focus on improving the efficiency of the SCM.
Acknowledgements
This work was supported in part by the Key-Area Research and Development Program of Guangzhou (202007030004); in part by the National Natural Science Foundation of China (62076258).

References


[Tang et al., 2018] Jianxiong Tang, Na Zhang, and Qia Li. Robust binary classification via $\ell_0$-svm. In ICDM Workshops, 2018.

[Tseng, 2001] Paul Tseng. Convergence of a block co-
ordinate descent method for nondifferentiable minimiza-


[Wang *et al.*, 2022] Huajun Wang, Yuan-Hai Shao, Sheng-
long Zhou, Ce Zhang, and Naihua Xiu. Support vector
machine classifier via $l_{0/1}$ soft-margin loss. *IEEE Trans-

[Wu *et al.*, 2018] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu,
and Luping Shi. Spatio-temporal backpropagation for

[Wu *et al.*, 2019a] Jibin Wu, Yansong Chua, Malu Zhang,
Qu Yang, Guoqi Li, and Haizhou Li. Deep spiking neural
network with spike count based learning rule. In *IJCNN*,
2019.

[Wu *et al.*, 2019b] Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu,
Yuan Xie, and Luping Shi. Direct training for spiking neu-

[Wu *et al.*, 2021] Zhenzhi Wu, Hehui Zhang, Yihan Lin,
Guoqi Li, Meng Wang, and Ye Tang. Liaf-net: Leaky
integrate and analog fire network for lightweight and effi-
cient spatiotemporal information processing. *IEEE Trans-
actions on Neural Networks and Learning Systems*, pages
1–14, 2021.

[Zhang *et al.*, 2022] Duzhen Zhang, Shuncheng Jia, and
Qingyu Wang. Recent advances and new frontiers in spik-

[Zheng *et al.*, 2021] Hanle Zheng, Yujie Wu, Lei Deng, Yi-
fan Hu, and Guoqi Li. Going deeper with directly-trained