MSAT: Multi-stage adaptive threshold for Deep Spiking Neural Networks

David S. Hippocampus*
Department of Computer Science
Cranberry-Lemon University
Pittsburgh, PA 15213
hippo@cs.cranberry-lemon.edu

Abstract

Spiking Neural Networks(SNNs) can do inference with low-power consumption natively because of its spike sparsity. Compared with the other two traning method: STDP and BP, Conversion from Artifical Neural Networks(ANNs), is a more easlier way to acheive deep SNNs and commly have the approximate performance compared with ANN. However, Conversion SNNs suffer from a accuracy degration and more latency at inference time. Lots of studys have tryed to make a trade-off between improving accuracy and reduce the latency. Here we analyze conversion loss layer-to-layer and point it out that membrane potential matters in both SNN accuracy and inference latency. Meanwhile, Different from current converson schemes which use the same and invariant threshold with inference time in a layer, we propose a multi-stage adaptive threshold inspired by biological model for deep spiking neural Networks and propose spike Confidence to further achieve higher accuracy and short latency. We examine the performance on CIFAR-10 and CIFAR-100 for classfication. Extensive experiments provide support on biological interpretability and present our method have a comparative performance and energy consumption with other state-of-the-art method.

1 Introduction

At present, Artificial Neural Network (ANN) is widely used in speech recognition, image processing and other fields. However, with the complexity of neural networks increasing progressively, running such deep networks often requires large amounts of computational resources, such as memory and power. In addition, current ANN's work mechanism differs from out brain. Actually, Neurons in the brain communicate by transmitting sequences (i.e. spike) generated by action potentials. Spiking Neuron network (SNN) works in a similar way. It also transmits the spike sequence to the downstream neurons. These spikes often carry a high amount of information, and the spike distribution is sparse, so it has the characteristics of low power consumption.

SNNs potentially offer an efficient way of doing inference when it conbines with neuroncomputing hardware, futuremore, SNN inherently shows efficiency on processing temporal and spatial data. Its diverse coding mechanisms, and events-driven characteristics are also promising. However, because the internal state variables of neurons do not satisfy the continuously differentiable requirement, it is difficult to be trained. To solve this problem, some algorithms based on the rules of gradient descent and spike-time dependent plasticity (STDP) were proposed, which had partly solved the problem of training SNNs. Frustratingly, It is still difficult to train deeper SNNs with complex network

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

Preprint. Under review.

structures, and results in a remaining of huge gap of performance between SNNs and CNNs in complex recognition or detection tasks.

To narrow the performance gap between SNNs and CNNs, methods of converting CNNs to SNNs had been proposed. In these methods, a CNN is firstly trained using the standard stochastic gradient descent and back propagation algorithm, and then the trained weights are mapped to an SNN with the same structure as the CNN. Inference is performed on the converted SNNs. The main idea is that the firing rates of spiking neurons can approximate the activations of their counterparts (RELU) in ANNs with sufficient time steps. This finding has become the fundamental principle underlying the conversion scheme. Converted SNNs often suffer from a accu- racy degration and more latency at inference time. Lots of studys have tryed to make a trade-off between improving accuracy and reduce the latency using method including adjust ANN topology when mapping ANN to SNN, using a more efficient fring mechanism et.al. Here we forms conversion loss formula and shows that residual membrane potential in each IF neuron increase the latency which mean firing rates approximate to activation value. We also find that most of current converson schemes, they use threshold invariant with inference time and are same and in a layer. This mechanism is inconsistent with a phenomenon which has been widely observed in the central nervous system, e.g. visual cortex , auditory midbrain, hippocampus, somatosensory cortex. It has been proposed that threshold variability reflects an adaptation of the spike threshold to the membrane potential. Inspired by this, we propose a multi-stage adaptive threshold for deep spiking neural Networks. For each neuron, its threshold vaires with its own membrane potential. We both do experimental on classfication tasks in non-trival datasets to prove proposed method is as well as efficiency with the current mainstream schemes when doing visual tasks.

Our major contribution can be summarized as:

- sufficient experimental on classfication tasks in non-trival datasets CIFAR10 and CIFAR100, shows that our proposed method is both efficiency and biological interpretability
- a formula on layer-by-layer conversion error, a new perspective diving existing method into three part
- a multi-stage adaptive threshold mechanism, which is widely existing in the center nervous system and thus more biological plausible. We use it for deep spiking neural Networks.
- a spike confidence in easlier timestep to determine probabilistic spike rather than all through spike

2 Preliminaries

Our conversion pipeline exploits the threshold balancing mechanism (Diehl et al., 2015; Sengupta et al., 2018) between ANN and SNN with modified ReLU function on the source ANN to reduce the consequential conversion error.

The main idea in ANN-to-SNN conversion is using mean firing rate $r_i^l(t)$ which indicates firing rate of neuron i in layer l for a total time t to approximate the activation value a_i^l in SNN. Here we give analytical explanation for the approximation.

In ANN, the neuron i activation value(after relu) in layer l a_i^l can be computed as:

$$a_i^l = \max\left(0, \sum_{j=1}^{M^{l-1}} W_{ij}^l a_j^{l-1} + b_i^l\right) \tag{1}$$

here $l \in \{1, \cdots, L\}$ indicates layer l in a network with L layers; W_{ij}^l indicates weight connection between neuron i in layer l and neuron j in layer $l-1$; b_i^l indicates neuron i bias in layer l; it is worth noting that a_i^l start from $l = 0$ and $a^0 = x$.

Neuron Model postsynaptic membrane potential(PSP) at timestep t, $V_i^l(t)$ is a sum of last timestep $t-1$ membrane potential and current input electric current. When PSP exceeds a certain voltage threshold, it emits an output spike and reset the membrane potential. One of the most widely adopted model is Integrate-and-Fire (IF) neuron, and membrane potential at the next time

step t would then be updated by soft-reset mechanism, which subtract threshold in PSP rather than reset the membrane potential to V_{reset}. The mathematical form is as follows.

$$V_i^l(t) = V_i^l(t-1) + V_{th}^l \left(\sum_j^{M^{l-1}} W_{ij}^l \Theta_{t,j}^{l-1} + b_i^l \right) - V_{th}^l \Theta_{t,i}^l \tag{2}$$

here $\Theta_{t,i}^l$ is a function indicating the neuron i in layer l occurrence of a spike at time t

$$\Theta_{t,i}^l = \Theta\left(V_i^l(t-1) + z_i^l(t) - V_{th}^l\right), with\ \Theta(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{else} \end{cases} \tag{3}$$

Here $z_i^l(t)$ is neuron i inputs in layer l and time t

$$z_i^l(t) = V_{th}^l \left(\sum_j^{M^{l-1}} W_{ij}^l \Theta_{t,i}^l + b_i^l \right) \tag{4}$$

3 Divding Conversion Error

Error comes from two part: one is converting ANN to SNN directly, result quantization error and clip error; the other is neuron transient dynamics and iiregular elicited spike, Spikes of Inactivated Neurons (SIN) error. show as fig 1

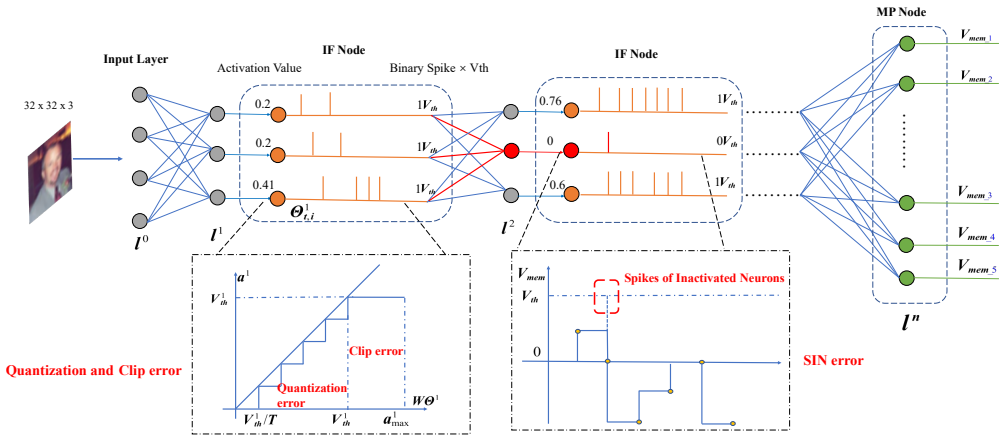


Figure 1: Two part error:one is quantization error and clip error result from converting ANN to SNN directly, the other is spikes of inactivated neurons error whoese existence result from iiregular elicited spike. A two layer multilayer perceptron network for demonstration.

3.1 Layer-By-Layer Quantization Error

For equation (2), cumulate the input over the simulation timestep T, we can derive the firing rate relationship layer-to-layer.

$$r_i^l(T) = \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} + b_i^l - \frac{V_i^l(T)}{T V_{th}^l} \tag{5}$$

Unfolding for each layer, relationship between mean fring rate and activation value can be shown as following:

$$
\begin{aligned}
r_i^l(T) &= \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} + b_i^l - \frac{V_i^l(T)}{TV_{th}^l} \\
&= \sum_j^{M^{l-1}} W_{ij}^l \left(\sum_k^{M^{l-2}} W_{jk}^{l-1} r_k^{l-2} + b_j^{l-1} - \frac{V_j^{l-1}(T)}{TV_{th}^{l-1}} \right) + b_i^l - \frac{V_i^l(T)}{TV_{th}^l} \\
&= \sum_j^{M^{l-1}} W_{ij}^l \left(\sum_k^{M^{l-2}} W_{jk}^{l-1} \left(\sum \cdots \underbrace{\sum_m^{M^1} W_{1m}^1 x_m + b_m^1}_{a_m^1} - \frac{V_i^1(T)}{TV_{th}^1} + \cdots + b - \frac{V(T)}{TV_{th}} \right) \right. \\
&\quad \left. + b_j^{l-1} - \frac{V_j^{l-1}(T)}{TV_{th}^{l-1}} \right) + b_i^l - \frac{V_i^l(T)}{TV_{th}^l}
\end{aligned} \tag{6}
$$

Use ΔV_i^l denotes $\frac{V_i^l(T)}{TV_{th}^l}$

$$
r_i^l(T) = a_i^l - \Delta V_i^l - \sum_j^{M^{l-1}} W_{ij} \Delta V_j^l - \cdots - \sum_j^{M^{l-1}} W_{ij}^l \cdots \sum_k^{M^1} W_{1k}^2 \Delta V_k^1 \tag{7}
$$

Note that the activation value is strictly fall in [0, 1] by using weight normalization and residual membrane potential $V_i^l(T)$ falls into $[0, V_{th}^l]$ so that it will enable us to estimate the activation function of SNNs ignoring the effect of unevenness error which degenerate to the quantization error as [3] mentioned. So the weights are not origin ANN weight and are scaled by V_{th}^{l-1}/V_{th}^l and bias are scaled by V_{th}^l and V_{th} is set to 1. It has the same form with threshold balancing, the different is that threshold balancing use postsynaptic neuron threshold times firing spike to compute mean firing rate Actually, Two method: weight normalization and threshold balancing are mathematically equivalent. We use threshold balancing in rest of paper for convenience.

Bias is a constant all the time so it does't affect the conversion error and we omit it and in threshold balancing, mean firing rate $r_i^l(t)$ is PSP average value, so equation (5) becomes

$$
r_i^l(T) = \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} - \frac{V_i^l(T)}{T} \tag{8}
$$

When V_{th}^l is larger than maximum of activation value, $V_i^l(T)$ will be less than V_{th} thus the residual membrane potential cannot be output thats why information transmition suffer a loss. There error is by nature because the discrete of timestep that the mean firing rate is a step function which cannot exactly approximate the source continuous RELU function, which is known as quantization error (flooring error), is can be expressed as

$$
r_i^l(T) = \frac{V_{th}^l}{T} \left\lfloor \frac{\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} T}{V_{th}^l} \right\rfloor \tag{9}
$$

3.2 Maximum Activation Clip Error

As mentioned in equation (9), if voltage threshold is set less than maximum activation value, then when PSP exceeds voltage threshold, the emited spike will not transmit efficient information to distinct above PSP. Set voltage threshold to maximum activation value can avoid this but suffer a huge latency. This is a trade-off, and in [32], choose quantile p for different datasets. (Li) propose a

Bayesian Optimization to find this p value. The Quantization error and Clip error and be expressed as

$$r_i^l(T) = \text{clip}\left(\frac{V_{th}^l}{T}\left\lfloor\frac{\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} T}{V_{th}^l}\right\rfloor, 0, V_{th}^l\right) \tag{10}$$

3.3 Spikes of Inactivated Neurons (SIN) Error

Spikes of Inactivated Neurons (SIN) is a group of neuron whoese activation value counterparts in ANN is negative. Theoretically they should not fire spike in total timestep to achieve ReLU filtering the negative value, however they fire spike that activation value a_i^l should zero while corresponding mean fring rate r_i^l is larger than zero. We use $\mathcal{R} = \left\{j \middle| \sum_{t=0}^T \Theta_{t,j}^l > 0, a_j^l(t) < 0\right\}$ to denote SIN, then SIN error can be expressed as

$$r_i^l(T) = \mathbf{0} - \frac{\sum_{j\in\mathcal{R}}\sum_{t=0}^T W_{ij}^l \Theta_{t,j}^l}{T} \tag{11}$$

The neurons with SIN usually fire early and then silence during the conversion process. We statistics SIN proportion in each layer, as shown in fig 2. It shows that SIN is prevalent and have larger proportion in deeper layer. What's more, It is clear that most of SIN appear in early 32 timestep, that's why early timestep accuracy suffer from more degration.

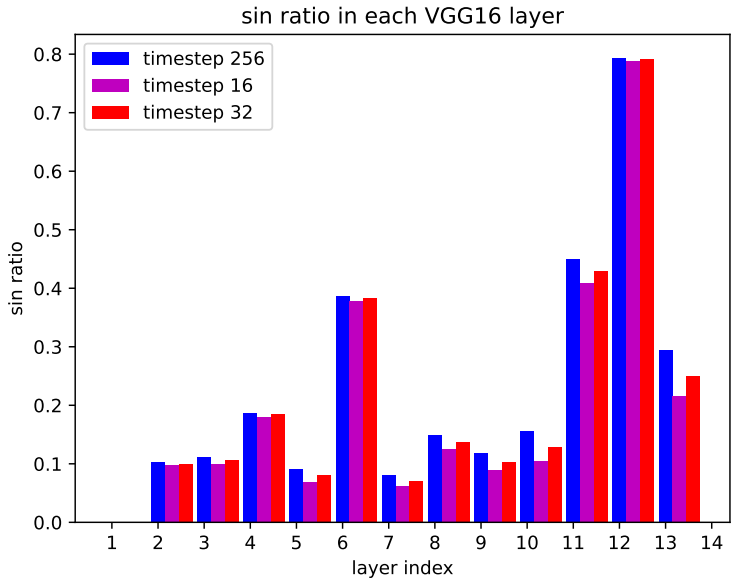


Figure 2: SIN ratio in each VGG16 layer. SIN have larger proportion in deeper layer early, account for early timestep accuracy suffer from more degration.

4 Method

4.1 adaptive threshold optimization error

Let's take a look at existing methods, no matter weight normalization or threshold balancing, they aim at zipping the gap between ANN and SNN. Though, we should be aware that the real advantage of SNN is its sparse spike which simultaneously low-power and brain-Inspired. Current method, however, set threshold voltage as the same in the same layer and these threshold will remain unchanged despite inference time increasing. It ignores a fact that neurons in different regions of brain

represent distinct dynamics and process information differently from other regions[25]. The threshold voltage of neurons is also known to have a broad range rather than a single value, known as homoeostasis[24]. Some neuronscience literature also indicates that threshold value is variable in the same neuron and threshold variability is a genuine feature of neurons[1, 13, 2, 26, 36]. Thus the voltage threshold should different from neurons and timestep. Here we demonstrate that voltage threshold is a function of timestep t and the transmits the equivalent information with the constant threshold.

obmitting the bias, the equation(5) can be rewrited as the following form

$$V_i^l(t) = V_i^l(t-1) + \sum_j^{M^{l-1}} V_{th,j}^l(t) W_{ij}^l \Theta_{t,j}^{l-1} - V_{th,i}^l(t)\Theta_{t,i}^l \tag{12}$$

firing rate during timestep T is computed as

$$r_i^l(T) = \frac{\sum_{t'=1}^{T} V_{th,i}^l(t')\Theta_{t',i}^l}{T} \tag{13}$$

the firing rate relationship in higher layer, it means equation 8 still satisfy so the conversion error form is the same. But note that the residual neuron membrane potential $V_i^l(T)$ can be appropriately adjusted, so the spike information could be more efficiency and thus shorten the conversion latency.

In threshold adaptation, the optimization target is

$$\min_{V_{th,i}^l(T)} \left(\text{clipfloor}\left(\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1}, T, V_{th,i}^l(T) \right) - \frac{\sum_{j\in\mathcal{R}} \sum_{t=0}^{T} W_{ij}^l \Theta_{t,j}^l}{T} - \text{ReLU}\left(\sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1} \right) \right) \tag{14}$$

There is no closed-form solution to above problem, [22] use gird search to find final solution, this is still a heuristic method. A trivial solution to this is $V_{th,i}^l = \sum_j^{M^{l-1}} W_{ij}^l r_j^{l-1}$, it means voltage threshold eaquals to input value for each neuron. With this solution, ANN can be converted to SNN only one timestep, however, this SNN elicit spike every timestep and lose spike sparsity, so it is not reasonable and makes no sense. In next subsection, we will give a more biologically rational method to finda a solution qualitatively.

4.2 Multi-stage adaptive threshold

In vivo, the spiking threshold displays large variability. Paper [9] show that this phenomenon has been widely observed in the central nervous system, e.g. visual cortex [1, 2], auditory midbrain [26], hippocampus [13], somatosensory cortex [36]. However, to our best knowledge, threshold varies in conversion is fewly used in conversion ANN to SNN. Work[17, 4, 22] while they only use two-stage or heuristic method and still cannot represent the homoeostasis well. The other work[8] use this biologically inspired dynamic thresholds mechanism to do robot obstacle avoidance and continuous control tasks.

Inspired by this, we propose a adaptive threshold, which is multi-stage and vaires with inference time. The method can be briefly sumed up as: **varies with firing history and input properties**, Specifically, spike threshold is positively correlated with the average V_i preceding spikes and negatively correlated with the rate of depolarization. It is consistent with neuroscience literature spike-threshold adaptation [28, 9] and [1] separately. some other threshold adaptation models such as the threshold increases after each spike and decreases if there is no spike[30, 6] is relatively simple and not fitted to this problem so we donot adopted. The relationship between threshold and membrane potential and rate of depolarization is described as

$$V_{th,i}^l(t+1) = \tau_{mp} V_{th\_mp,i}^l(t) + \tau_{rd} V_{th\_rd,i}^l(t+1) \tag{15}$$

Where τ_{mp} and τ_{rd} is the time constant of the dynamic tracking threshold $V_{th\_mp,i}^l(t)$ and dynamic evoked threshold $V_{th\_rd,i}^l(t+1)$ separately.

dynamic tracking threshold(DTT) DTT is a flection of spiking threshold vaires with firing history. It shows that spike threshold depends on preceding membrane potential and tracking the membrane potential at a short timescale due to inactivation of sodium channel[20, 16, 29] or the activation of potassium channels[14, 10]. in [9], the DTT is a similar first-order kinetic equation, we here use steady-state threshold for fitting our SNNs. we use $V^l_{m,i}(t)$ to denote the average membrane potential during timestep t in layer l neuron i, then DTT is following:

$$V^l_{th\_mp,i}(t) = \left(\alpha\left(V^l_i(t) - V^l_m(t)\right) + V^l_T + k_a ln\left(1 + e^{\frac{V^l_i(t) - V^l_m(t)}{k_i}}\right)\right) \tag{16}$$

here $\eta, k$is both time constant, V^l_T is the parameters to optimize. when PSP is less than average membrane potential $V^l_m(t)$, the slope is η on the left side of the knee. The slope on the right side is $\frac{k_a}{k_i} + \alpha$. The curvature C is determined by $\alpha, k_a, k_i, V^l_T, V^l_i(t)$.

the threshold increases with residual membrane potential and thus any voltage fluctuations that are slower than threshold adaptation should not have an impact on output spiking, this is indirectly relieve the spike of inactivated neuron error and clip error.

dynamic evoked threshold(DET) DET is a flection of spiking threshold vaires with input properties. Paper[1] show spike threshold vaires inversely with preceding rate of depolarization(rd) dV_m/dt by ploting scatter. And dependence of threshold dynamics on rd is due to decrease in the availability of sodium channels. In IF neuron, we use membrane potential before spike variation to express rd thus DET can be expressed as

$$V^l_{th\_rd,i}(t+1) = \tau_{rd} e^{-\frac{\left(\mathbf{v}^l_i(t+1) - \mathbf{v}^l_i(t)\right)}{C}} \tag{17}$$

Here use bold $\mathbf{V}^l_i(t)$ to denote membrane potential before spike and make a distinction with $V^l_i(t)$ which denote residual membrane potential after spike. C is time constant flecting the insensitive to input membrane potential variation. The threshold decreases with input value exponentially.

Interaction of DET and DTT Take together, the link between preceding spike membrane potential and negatively correlated with the rate of depolarization, shows that threshold adaptation neurons selective to fast input variations and remarkably insensitive to slow ones.

In other words, the slow voltage fluctuations are filtered out by threshold adaptation and will not elicit spike. The slow voltage fluctuations may come from unexpected spike of inactivated neuron so it relieve the SIN error partly and reduce the total spike number meanwhile promoting the energy efficiency.

On the other hand, the threshold variation by DTT ensure voltage threshold will not raise big gap with maximum activation value thus narrowed clip error; and the threshold variation by DET makes threshold not increase endlessly and reduce appropriately for fear of large quantization error.

It should be noted that DTT cannot guarantee positive value because the difference between current residual membrane potential and average membrane potential is may negative. Negative threshold is not we want so we let $V^l_{th,i}(t+1)$ pass a sigmoid function and map it to $[0, 1]$. We use this value multiply maximum activation value as truely threshold.

4.3 Spike Confidence

As shown in fig , the early spike elicited is not truely all and some part of them are raised by SIN, though threshold adaptive DTT can partly relieve this, SIN will not be distinguished with other normal neuron untill they keep silence in longer timestep. It hints us we should not totally believe the early spike. As many image cognition do, we also import a confidence to show how confident the elicited spike is that they are outcome of normal neuron so not SIN. So we give different spike confidence to different layer and in early timestep every spike which should elicit will not fire directly but use this spike confidence to determine spike or not. This is somewhat like dropout[34]

to normalize and relieve overfitting. Formula expression as follows

$$
\begin{aligned}
c_j^l &\sim Bernoulli(p) \\
\tilde{\Theta}_{t,i}^l &= c_j^l * \Theta_{t,i}^l \\
V_i^l(t) = V_i^l(t-1) + \sum_j^{M^{l-1}} V_{th,j}^l(t) W_{ij}^l \tilde{\Theta}_{t,j}^{l-1} &- V_{th,i}^l(t) \tilde{\Theta}_{t,i}^l
\end{aligned} \tag{18}
$$

Above equation ensures that all the output neurons are used and adjust the neurons, thresholds to the stimuli for which they become specialized. The pseudocodes for adaptive threshold algorithm are shown in Algorithm 1.

Algorithm 1 Conversion from ANN to SNN: Multi-stage adaptive threshold

Require: Pretrained ANN, training set, SNN's inference timestep T, spike confidence list C
Ensure: The converted SNN firing rate approximate ANN activation value with shorter latency

for s = 1 to # of samples **do**
  $a_l \leftarrow$ layer-wise activation value
  for l = 1 to L **do**
    $V_{th}^l \leftarrow max[V_{th}^l, max(a_l)]$
    $SNN.layer[l].V_{th} \leftarrow V_{th}^l$
  end for
end for
for t = 1 to timestep T **do**
  for l = 1 to L **do**
    spike confidence $c^l \leftarrow Bernoulli(C[l])$
    for j = 1 to neuron number of layer l **do**
      compute DTT and DET as formula (17)(19) to get adaptation value
      $SNN.layer[l].V_{th}[j] \leftarrow sigmoid(DTT + DET) * V_{th}^l$
      if fire spike and t < early timestep **then**
        spike = c^l * spike
      end if
    end for
  end for
end for

5 Related Work

The conversion use the pre-trained ANN and to map them to an equivalent SNN. the studies begin with [27], the main idea that mean firing rate in IF neuron can approximate RELU activation value is proposed in [7], and the mathematical formula derivation is expressed in [32]. Since [33], the converted SNN begin going deeper and do classfication task in larger datasets. [12] propose a more efficiency reset method named softreset which use reset by subtraction mechanism to get a better performance. [31] first use threshold optimization in deep SNN. [5] divide conversion into floor and clip error from a new quantization perspective, [22] is one step closer to optimizing the conversion error. [37] construct the deep SNN with double-threshold, [23] propose temporal separation to further zip gap between ANN and SNN. [21] use burst mechanism and propose LIPooling to solve the conversion error caused by the MaxPooling layer. [35] present a dual-phase converting algorithm to relieve clip and floor error.

6 Experiment

We validate effectiveness of our proposed method in this section. We test the classification task on the CIFAR10 [19], CIFAR100 [19] datasets. In input encoding, we choose the first network layer as the encoding layer and real-value input for higher performance.

Table 1: Summary of given hyperparameters on different network

Symbol	VGG16	ResNet20
α	0.03	0.3
k_a	1	1
k_i	1.0	1.0
C	5.0	5.0
τ_{mp}	1	0.5
τ_{rd}	1	0.5

As mentioned in Related work, some work train a modified ANN to get short latency when converting ANN-to-SNN. We donot want to do too much restriction on origin ANN, and we save the topology and use the original weights in target ANN for universal conversion. In this way, we compare the state-of-the-art approaches include P-Norm[32], Spike-Norm[33], Channel-Norm[18], TSC[11], RMP-SNN[12], Opt.[5], Calibration[22], Burst [21]. VGG16 and ResNet20 are used for target ANN as in previous work for comparison.

6.1 Ablation Study

There are six time constant in proposed adaptive threshold, our experiments use four group hyperparameters separately for VGG16 and ResNet20, And we give the best hyperparameters group which makes multi-stage adaptive threshold have the better performance on both CIFAR10 and CIFAR100 datasets in Table1

In our method, V_{th} are chosen as the maximum activation value. In figure3 The dotted lines indicate the target ANN accuracy, and we obverse the Dynamic Tracking Threshold(DTT) and Dynamic Evoked Threshold(DET) and fixed threshold impact on classfication accuracy. It is intuitive that with DET only we could achieve shorter latency and have a faster approach to target ANN classfication accuracy. While DET only sometimes have a better performance than $0.7V_{th}$ as figure3(d)and sometimes the close to $0.7V_{th}$ as figure3(c).

Adaptive threshold with both DTT and DET could have a faster inference time and need less timestep to achieve the target ANN accuracy, in figure3(c), it shows that it also have a better performance than ANN, this may because the adaptive threshold relieve the overfitting which exists in ANN. The result shows that the DTT and DET could have advantage on shorter latency and higher accuracy, it result from the adaptive threshold could find an appropriate threshold that reduce the quantization error and do not import larger clip error.

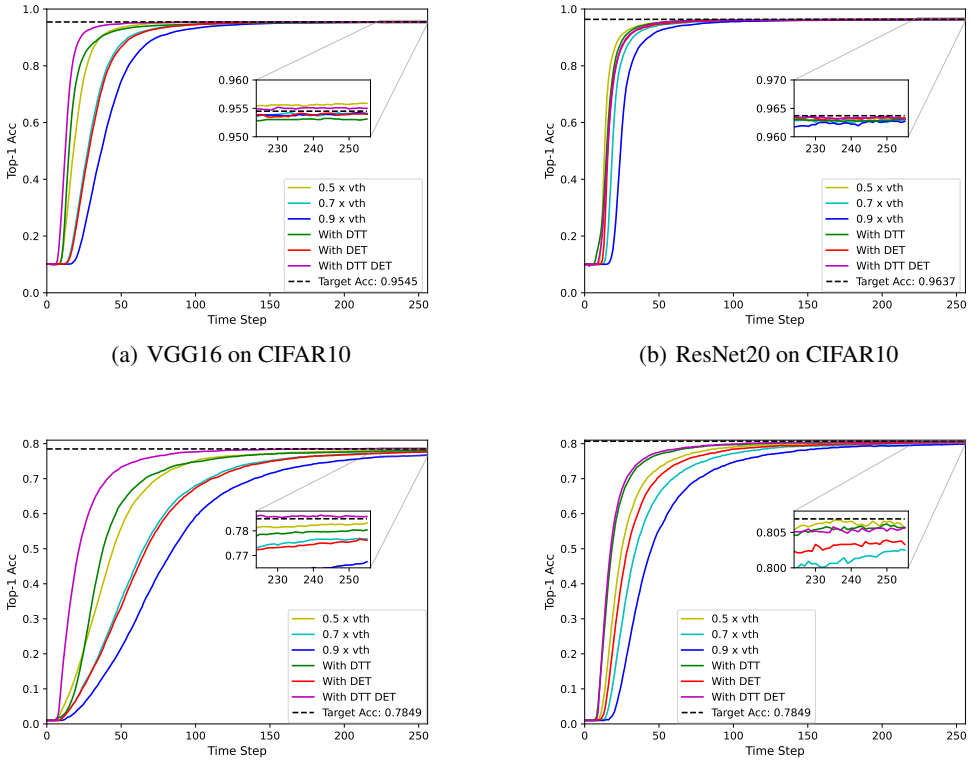
figure 4 shows SIN ratio in each VGG16 layer with dynamic threshold and spike confidence, it shows that the sin ratio could be remarkably degrated especially the penultimate IF layer.

6.2 Comparison With The State-Of-The-Art

We compare our method with current ANN-to-SNN conversion state-of-the-art method, we achieve the comparative performance with these SOTA method, and the nearly lossless conversion from target ANN in timestep 128. It is worth noting that though calibration also use dynamic threshold, our method with no need for search best voltage threshold which could may result more computing consumption. The proposed method voltage threshold is an adaptation to average membrane potential and current input variation, brings biological explanatory and energy-efficiency simultaneously.

6.3 Energy-Efficiency and Sparsity

In this section, we compute the firing rate each layer to evaluate our spike sparsity. We use the same way in [31] to compute energy consumption. Multiply and accumulation(MAC) in ANN needs 5.1x more energy cost than SNN accumulation(AC)[15]. 32-bit floating-point AC and MAC per operation consumpte 0.9pJ and 4.6pJ individually. And there is no operation in SNN if no spike elicit. We chosen the VGG16 on CIFAR10 dataset with timestep T=64, the firing rate on whole dataset each layer is shown in figure**??** we have 0.84x energy consumption compared to target ANN thus energy-efficiency.



(a) VGG16 on CIFAR10

(b) ResNet20 on CIFAR10

(c) VGG16 on CIFAR100

(d) ResNet20 on CIFAR100

Figure 3: Ablation study on DTT and DET

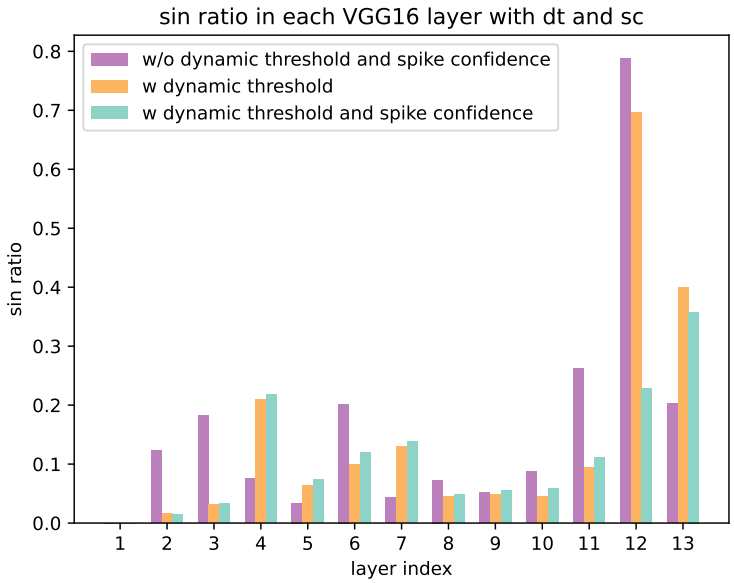


Figure 4: SIN ratio in each VGG16 layer with dynamic threshold and spike confidence.

Table 2: Experimental results on CIFAR10 and CIFAR100

Method	Use DT[1]	ANN	SNN Best	T=32	T=64	T=128	T=256
VGG16, CIFAR10							
p-Norm [32]	✗	95.74	94.61	93.40	93.98	94.40	94.61
Channel-Norm[18]	✗	95.74	94.49	87.12	91.85	93.81	94.49
Spike-Norm[33]	✗	91.7	91.55	-	-	-	-
Hybrid[31]	✗	92.81	92.48	-	-	91.13	-
TSC[11]	✗	93.63	93.63	-	-	-	-
RMP-SNN[12]	✗	93.63	93.63	60.30	90.35	92.41	93.04
Opt.[5]	✗	92.34	92.29	92.29	92.22	92.24	-
Burst [21]	✗	95.74	95.75	95.58	95.66	95.69	95.72
This Work(DT+SC)	✓	95.74	**95.52**	**93.33**	**95.14**	**95.42**	**95.50**
ResNet20, CIFAR10							
p-Norm [32]	✗	96.56	94.61	93.40	93.98	94.40	94.61
Channel-Norm[18]	✗	96.56	94.49	87.12	91.85	93.81	94.49
Spike-Norm[33]	✗	89.1	87.46	-	-	-	-
TSC[11]	✗	91.47	91.42	-	-	-	-
RMP-SNN[12]	✗	91.47	87.46	-	-	-	-
Opt.[5]	✗	93.61	93.58	93.30	93.55	93.56	-
Burst [21]	✗	96.56	96.59	96.11	95.49	95.45	96.36
This Work(DT+SC)	✓	96.56	**96.39**	**93.33**	**95.93**	**96.23**	**96.38**
VGG16, CIFAR100							
p-Norm [32]	✗	78.49	58.44	44.88	51.89	56.02	58.44
Channel-Norm[18]	✗	78.49	74.74	54.03	67.34	72.50	74.73
Spike-Norm[33]	✗	71.22	70.77	-	-	-	-
TSC[11]	✗	71.22	70.97	-	-	69.86	70.65
RMP-SNN[12]	✗	71.22	70.93	-	-	63.76	68.34
Opt.[5]	✗	77.89	77.71	7.64	21.84	55.04	73.54
Calibration[22]	✓	77.89	77.87	73.55	76.64	77.40	77.68
Burst [21]	✗	78.49	78.71	74.98	78.26	78.66	78.65
This Work(DT+SC)	✓	78.49	**78.55**	**67.73**	**76.09**	**78.08**	**78.50**
ResNet20, CIFAR100							
p-Norm [32]	✗	80.69	67.35	38.13	58.09	64.96	67.33
Channel-Norm[18]	✗	80.69	71.26	52.59	66.05	70.08	71.26
Spike-Norm[33]	✗	68.72	64.09	-	-	-	-
TSC[11]	✗	68.72	68.18	-	-	58.42	65.27
RMP-SNN[12]	✗	68.72	67.82	27.64	46.91	57.69	64.06
Opt.[5]	✗	77.16	77.22	51.27	70.12	75.81	77.22
Calibration[22]	✓	77.16	77.73	76.32	77.29	77.73	77.63
Burst [21]	✗	80.69	80.72	76.39	79.83	80.52	80.57
This Work(DT+SC)	✓	80.69	**80.61**	**71.92**	**78.94**	**80.28**	**80.58**

[1] Dynamic Threshold

Figure 5: firing in each VGG16 layer on CIFAR10 dataset with timestep T=16.

7 Discussion

we divide conversion error again and point out it that SIN take a larger proportion in deep SNN. Admittedly, our method exists small gap with the State-of-the-art methods, mainly because they modify the ANN topology for better conversion. However, our proposed method could achieve some accuracy with certain accuracy. biological inspired threshold and spike confidence, we believe it is a inspiring work for SNN and will have a better performance combined with other method.

References

[1] Rony Azouz and Charles M Gray. Dynamic spike threshold reveals a mechanism for synaptic coincidence detection in cortical neurons in vivo. *Proceedings of the National Academy of Sciences*, 97(14):8110–8115, 2000.

[2] Rony Azouz and Charles M Gray. Adaptive coincidence detection and dynamic gain control in visual cortical neurons in vivo. *Neuron*, 37(3):513–523, 2003.

[3] Tong Bu, Wei Fang, Jianhao Ding, PengLin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ann-snn conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*, 2021.

[4] Yunhua Chen, Yingchao Mai, Ren Feng, and Jinsheng Xiao. An adaptive threshold mechanism for accurate and efficient deep spiking convolutional neural networks. *Neurocomputing*, 469:189–197, 2022.

[5] Shikuang Deng and Shi Gu. Optimal conversion of conventional artificial neural networks to spiking neural networks. *arXiv preprint arXiv:2103.00476*, 2021.

[6] Peter U Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 9:99, 2015.

[7] Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)*, pages 1–8. ieee, 2015.

[8] Jianchuan Ding, Bo Dong, Felix Heide, Yufei Ding, Yunduo Zhou, Baocai Yin, and Xin Yang. Biologically inspired dynamic thresholds for spiking neural networks. *arXiv preprint arXiv:2206.04426*, 2022.

[9] Bertrand Fontaine, José Luis Peña, and Romain Brette. Spike-threshold adaptation predicted by membrane potential dynamics in vivo. *PLoS computational biology*, 10(4):e1003560, 2014.

[10] Ethan M Goldberg, Brian D Clark, Edward Zagha, Mark Nahmani, Alev Erisir, and Bernardo Rudy. K+ channels at the axon initial segment dampen near-threshold excitability of neocortical fast-spiking gabaergic interneurons. *Neuron*, 58(3):387–400, 2008.

[11] Bing Han and Kaushik Roy. Deep spiking neural network: Energy efficiency through time based coding. In *European Conference on Computer Vision*, pages 388–404. Springer, 2020.

[12] Bing Han, Gopalakrishnan Srinivasan, and Kaushik Roy. Rmp-snn: Residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13558–13567, 2020.

[13] DA Henze and G Buzsáki. Action potential threshold of hippocampal pyramidal cells in vivo is increased by recent spiking activity. *Neuroscience*, 105(1):121–130, 2001.

[14] Matthew H Higgs and William J Spain. Kv1 channels control spike threshold dynamics and spike timing in cortical pyramidal neurones. *The Journal of physiology*, 589(21):5125–5142, 2011.

[15] Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pages 10–14. IEEE, 2014.

[16] Wenqin Hu, Cuiping Tian, Tun Li, Mingpo Yang, Han Hou, and Yousheng Shu. Distinct contributions of nav1. 6 and nav1. 2 in action potential initiation and backpropagation. *Nature neuroscience*, 12(8):996–1002, 2009.

[17] Seijoon Kim, Seongsik Park, Byunggook Na, Jongwan Kim, and Sungroh Yoon. Towards fast and accurate object detection in bio-inspired spiking neural networks through bayesian optimization. *IEEE Access*, 9:2633–2643, 2020.

[18] Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11270–11277, 2020.

[19] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[20] Hiroshi Kuba, Yuki Oichi, and Harunori Ohmori. Presynaptic activity regulates na+ channel distribution at the axon initial segment. *Nature*, 465(7301):1075–1078, 2010.

[21] Yang Li and Yi Zeng. Efficient and accurate conversion of spiking neural network with burst spikes. *arXiv preprint arXiv:2204.13271*, 2022.

[22] Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ann: Towards efficient, accurate spiking neural networks calibration. In *International Conference on Machine Learning*, pages 6316–6325. PMLR, 2021.

[23] Fangxin Liu, Wenbo Zhao, Yongbiao Chen, Zongwu Wang, and Li Jiang. Spikeconverter: An efficient conversion framework zipping the gap between artificial neural networks and spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[24] Eve Marder and Jean-Marc Goaillard. Variability, compensation and homeostasis in neuron and network function. *Nature Reviews Neuroscience*, 7(7):563–574, 2006.

[25] Yasuhiro Mochizuki, Tomokatsu Onaga, Hideaki Shimazaki, Takeaki Shimokawa, Yasuhiro Tsubo, Rie Kimura, Akiko Saiki, Yutaka Sakai, Yoshikazu Isomura, Shigeyoshi Fujisawa, et al. Similarity in neuronal firing regimes across mammalian species. *Journal of Neuroscience*, 36(21):5736–5747, 2016.

[26] Jose Luis Pena and Masakazu Konishi. From postsynaptic potentials to spikes in the genesis of auditory spatial receptive fields. *Journal of Neuroscience*, 22(13):5652–5658, 2002.

[27] José Antonio Pérez-Carrasco, Bo Zhao, Carmen Serrano, Begona Acha, Teresa Serrano-Gotarredona, Shouchun Chen, and Bernabé Linares-Barranco. Mapping from frame-driven to frame-free event-driven vision systems by low-rate rate coding and coincidence processing–application to feedforward convnets. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2706–2719, 2013.

[28] Jonathan Platkiewicz and Romain Brette. A threshold equation for action potential initiation. *PLoS computational biology*, 6(7):e1000850, 2010.

[29] Jonathan Platkiewicz and Romain Brette. Impact of fast sodium channel inactivation on spike threshold dynamics and synaptic integration. *PLoS computational biology*, 7(5):e1001129, 2011.

[30] Damien Querlioz, Olivier Bichler, Philippe Dollfus, and Christian Gamrat. Immunity to device variations in a spiking neural network with memristive nanodevices. *IEEE transactions on nanotechnology*, 12(3):288–295, 2013.

[31] Nitin Rathi and Kaushik Roy. Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks. *arXiv preprint arXiv:2008.03658*, 2020.

[32] Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, 11:682, 2017.

[33] Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in neuroscience*, 13:95, 2019.

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[35] Ziming Wang, Shuang Lian, Yuhao Zhang, Xiaoxin Cui, Rui Yan, and Huajin Tang. Towards lossless ann-snn conversion under ultra-low latency with dual-phase optimization. *arXiv preprint arXiv:2205.07473*, 2022.

[36] W Bryan Wilent and Diego Contreras. Stimulus-dependent changes in spike threshold enhance feature selectivity in rat barrel cortex neurons. *Journal of Neuroscience*, 25(11):2983–2991, 2005.

[37] Qiang Yu, Chenxiang Ma, Shiming Song, Gaoyan Zhang, Jianwu Dang, and Kay Chen Tan. Constructing accurate and efficient deep spiking neural networks with double-threshold and augmented schemes. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1714–1726, 2021.

A Appendix

Optionally include extra information (complete proofs, additional experiments and plots) in the appendix. This section will often be part of the supplemental material.