Deep Residual Learning in Spiking Neural Networks

Wei Fang\textsuperscript{1,2}, Zhaofei Yu\textsuperscript{1,2*}, Yanqi Chen\textsuperscript{1,2},
Tiejun Huang\textsuperscript{1,2}, Timothée Masquelier\textsuperscript{3}, Yonghong Tian\textsuperscript{1,2*}

\textsuperscript{1}Department of Computer Science and Technology, Peking University
\textsuperscript{2}Peng Cheng Laboratory, Shenzhen 518055, China
\textsuperscript{3}Centre de Recherche Cerveau et Cognition, UMR5549 CNRS - Univ. Toulouse 3, Toulouse, France

Abstract

Deep Spiking Neural Networks (SNNs) present optimization difficulties for gradient-based approaches due to discrete binary activation and complex spatial-temporal dynamics. Considering the huge success of ResNet in deep learning, it would be natural to train deep SNNs with residual learning. Previous Spiking ResNet mimics the standard residual block in ANNs and simply replaces ReLU activation layers with spiking neurons, which suffers the degradation problem and can hardly implement residual learning. In this paper, we propose the spike-element-wise (SEW) ResNet to realize residual learning in deep SNNs. We prove that the SEW ResNet can easily implement identity mapping and overcome the vanishing/exploding gradient problems of Spiking ResNet. We evaluate our SEW ResNet on ImageNet, DVS Gesture, and CIFAR10-DVS datasets, and show that SEW ResNet outperforms the state-of-the-art directly trained SNNs in both accuracy and time-steps. Moreover, SEW ResNet can achieve higher performance by simply adding more layers, providing a simple method to train deep SNNs. To our best knowledge, this is the first time that directly training deep SNNs with more than 100 layers becomes possible. Our codes are available at \url{https://github.com/fangwei123456/Spike-Element-Wise-ResNet}.

1 Introduction

Artificial Neural Networks (ANNs) have achieved great success in many tasks, including image classification \cite{28,52,55}, object detection \cite{9,44,44}, machine translation \cite{2}, and gaming \cite{37,51}. One of the critical factors for ANNs’ success is deep learning \cite{29}, which uses multi-layers to learn representations of data with multiple levels of abstraction. It has been proved that deeper networks have advantages over shallower networks in computation cost and generalization ability \cite{3}. The function represented by a deep network can require an exponential number of hidden units by a shallow network with one hidden layer \cite{38}. In addition, the depth of the network is closely related to the network’s performance in practical tasks \cite{52,55,27,52}. Nevertheless, recent evidence \cite{13,53,14} reveals that with the network depth increasing, the accuracy gets saturated and then degrades rapidly. To solve this degradation problem, residual learning is proposed \cite{14,15} and the residual structure is widely exploited in “very deep” networks that achieve the leading performance \cite{22,59,18,57}.

Spiking Neural Networks (SNNs) are regarded as a potential competitor of ANNs for their high biological plausibility, event-driven property, and low power consumption \cite{45}. Recently, deep learning methods are introduced into SNNs, and deep SNNs have achieved close performance as ANNs in some simple classification datasets \cite{56}, but still worse than ANNs in complex tasks, e.g., classifying the ImageNet dataset \cite{47}. To obtain higher performance SNNs, it would be natural to

*Corresponding author

explore deeper network structures like ResNet. Spiking ResNet [25, 60, 21, 17, 49, 12, 30, 64, 48, 42, 43], as the spiking version of ResNet, is proposed by mimicking the residual block in ANNs and replacing ReLU activation layers with spiking neurons. Spiking ResNet converted from ANN achieves state-of-the-art accuracy on nearly all datasets, while the directly trained Spiking ResNet has not been validated to solve the degradation problem.

In this paper, we show that Spiking ResNet is inapplicable to all neuron models to achieve identity mapping. Even if the identity mapping condition is met, Spiking ResNet suffers from the problems of vanishing/exploding gradient. Thus, we propose the Spike-Element-Wise (SEW) ResNet to realize residual learning in SNNs. We prove that the SEW ResNet can easily implement identity mapping and overcome the vanishing/exploding gradient problems at the same time. We evaluate Spiking ResNet and SEW ResNet on both the static ImageNet dataset and the neuromorphic DVS Gesture dataset [1], CIFAR10-DVS dataset [32]. The experiment results are consistent with our analysis, indicating that the deeper Spiking ResNet suffers from the degradation problem — the deeper network has higher training loss than the shallower network, while SEW ResNet can achieve higher performance by simply increasing the network’s depth. Moreover, we show that SEW ResNet outperforms the state-of-the-art directly trained SNNs in both accuracy and time-steps. To the best of our knowledge, this is the first time to explore the directly-trained deep SNNs with more than 100 layers.

2 Related Work

2.1 Learning Methods of Spiking Neural Networks

ANN to SNN conversion (ANN2SNN) [20, 4, 46, 49, 12, 11, 6, 54, 33] and backpropagation with surrogate gradient [40] are the two main methods to get deep SNNs. The ANN2SNN method firstly trains an ANN with ReLU activation, then converts the ANN to an SNN by replacing ReLU with spiking neurons and adding scaling operations like weight normalization and threshold balancing. Some recent conversion methods have achieved near loss-less accuracy with VGG-16 and ResNet [12, 11, 6, 33]. However, the converted SNN needs a longer time to rival the original ANN in precision as the conversion is based on rate-coding [46], which increases the SNN’s latency and restricts the practical application. The backpropagation methods can be classified into two categories [26]. The method in the first category computes the gradient by unfolding the network over the simulation time-steps [31, 19, 58, 50, 30, 40], which is similar to the idea of backpropagation through time (BPTT). As the gradient with respect to the threshold-triggered firing is non-differentiable, the surrogate gradient is often used. The SNN trained by the surrogate method is not limited to rate-coding, and can also be applied on temporal tasks, e.g., classifying neuromorphic datasets [58, 8, 16]. The second method computes the gradients of the timings of existing spikes with respect to the membrane potential at the spike timing [5, 39, 24, 65, 63].

2.2 Spiking Residual Structure

Previous ANN2SNN methods noticed the distinction between plain feedforward ANNs and residual ANNs, and made specific normalization for conversion. Hu et al. [17] were the first to apply the residual structure in ANN2SNN with scaled shortcuts in SNN to match the activations of the original ANN. Sengupta et al. [49] proposed Spike-Norm to balance SNN’s threshold and verified their method by converting VGG and ResNet to SNNs. Existing backpropagation-based methods use nearly the same structure from ResNet. Lee et al. [30] evaluated their custom surrogate methods on shallow ResNets whose depths are no more than ResNet-11. Zheng et al. [64] proposed the threshold-dependent batch normalization (td-BN) to replace naive batch normalization (BN) [23] and successfully trained Spiking ResNet-34 and Spiking ResNet-50 directly with surrogate gradient by adding td-BN in shortcuts.

3 Methods

3.1 Spiking Neuron Model

The spiking neuron is the fundamental computing unit of SNNs. Similar to Fang et al. [8], we use a unified model to describe the dynamics of all kinds of spiking neurons, which includes the following
discrete-time equations:

\[ H[t] = f(V[t - 1], X[t]), \quad (1) \]
\[ S[t] = \Theta(H[t] - V_{th}), \quad (2) \]
\[ V[t] = H[t] (1 - S[t]) + V_{reset} S[t], \quad (3) \]

where \( X[t] \) is the input current at time-step \( t \), \( H[t] \) and \( V[t] \) denote the membrane potential after neuronal dynamics and after the trigger of a spike at time-step \( t \), respectively. \( V_{th} \) is the firing threshold, \( \Theta(x) \) is the Heaviside step function and is defined by \( \Theta(x) = 1 \) for \( x \geq 0 \) and \( \Theta(x) = 0 \) for \( x < 0 \). \( S[t] \) is the output spike at time-step \( t \), which equals 1 if there is a spike and 0 otherwise. \( V_{reset} \) denotes the reset potential. The function \( f(\cdot) \) in Eq. (1) describes the neuronal dynamics and takes different forms for different spiking neuron models. For example, the function \( f(\cdot) \) for the Integrate-and-Fire (IF) model and Leaky Integrate-and-Fire (LIF) model can be described by Eq. (4) and Eq. (5), respectively.

\[ H[t] = V[t - 1] + X[t], \quad (4) \]
\[ H[t] = V[t - 1] + \frac{1}{\tau} (X[t] - (V[t - 1] - V_{reset})), \quad (5) \]

where \( \tau \) represents the membrane time constant. Eq. (2) and Eq. (3) describe the spike generation and resetting processes, which are the same for all kinds of spiking neuron models. In this paper, the surrogate gradient method is used to define \( \Theta'(x) \triangleq \sigma'(x) \) during error back-propagation, with \( \sigma(x) \) denoting the surrogate function.

### 3.2 Drawbacks of Spiking ResNet

The residual block is the key component of ResNet. Fig. 1(a) shows the basic block in ResNet [14], where \( X^l, Y^l \) are the input and output of the \( l \)-th block in ResNet, Conv is the convolutional layer, BN denotes batch normalization, and ReLU denotes the rectified linear unit activation layer. The basic block of Spiking ResNet used in [64, 17, 30] simply mimics the block in ANNs by replacing ReLU activation layers with spiking neurons (SN), which is illustrated in Fig. 1(b). Here \( S^l[t], O^l[t] \) are the input and output of the \( l \)-th block in Spiking ResNet at time-step \( t \). Based on the above definition, we will analyze the drawbacks of Spiking ResNet below.

**Spiking ResNet is inapplicable to all neuron models to achieve identity mapping.** One of the critical concepts in ResNet is identity mapping. He et al. [14] noted that if the added layers implement the identity mapping, a deeper model should have training error no greater than its shallower counterpart. However, it is unable to train the added layers to implement identity mapping in a feasible time, resulting in deeper models performing worse than shallower models (the degradation problem). To solve this problem, the residual learning is proposed by adding a shortcut connection (shown in Fig. 1(a)). If we use \( F^l \) to denote the residual mapping, e.g., a stack of two convolutional layers, of the \( l \)-th residual block in ResNet and Spiking ResNet, then the residual block in Fig. 1(a)
and Fig. 1(b) can be formulated as

\[
Y^l = \text{ReLU}(\mathcal{F}^l(X^l) + X^l),
\]

(6)

\[
O^l[t] = \text{SN}({\mathcal{F}^l}^l(S^l[t])) + S^l[t].
\]

(7)

The residual block of Eq. (6) make it easy to implement identity mapping in ANNs. To see this, when \( \mathcal{F}^l(X^l) \equiv 0, Y^l = \text{ReLU}(X^l) \). In most cases, \( X^l \) is the activation of the previous ReLU layer and \( X^l \geq 0 \). Thus, \( Y^l = \text{ReLU}(X^l) = X^l \), which is identity mapping.

Different from ResNet, the residual block in Spiking ResNet (Eq. (7)) restricts the models of spiking neuron to implement identity mapping. When \( \mathcal{F}^l(S^l[t]) \equiv 0, O^l[t] = \text{SN}(S^l[t]) \neq S^l[t] \). To transmit \( S^l[t] \) and make \( \text{SN}(S^l[t]) = S^l[t] \), the last spiking neuron (SN) in the \( l \)-th residual block needs to fire a spike after receiving a spike, and keep silent after receiving no spike at time-step \( t \). It works for IF neuron described by Eq. (4). Specifically, we can set \( 0 < V_{ih} \leq 1 \) and \( V[t-1] = 0 \) to ensure that \( X[t] = 1 \) leads to \( H[t] \geq V_{ih} \), and \( X[t] = 0 \) leads to \( H[t] < V_{ih} \). However, when considering some spiking neuron models with complex neuronal dynamics, it is hard to achieve \( \text{SN}(S^l[t]) = S^l[t] \). For example, the LIF neuron used in [66, 8,61] considers a learnable membrane time constant \( \tau \), the neuronal dynamics of which can be described with Eq. (5). When \( X[t] = 1 \) and \( V[t-1] = 0 \), \( H[t] = \frac{1}{\tau} \). It is difficult to find a firing threshold that ensures \( H[t] > V_{ih} \) as \( \tau \) is being changed in training by the optimizer.

**Spiking ResNet suffers from the problems of vanishing/exploding gradient.** Consider a spiking ResNet with \( k \) sequential blocks to transmit \( S^l[t] \), and the identity mapping condition is met, e.g., the spiking neurons are the IF neurons with \( 0 < V_{ih} \leq 1 \), then we have \( S^l[t] = S^{l+1}[t] = ... = S^{l+k-1}[t] = \Theta^{l+k-1}[t] \). Denote the \( j \)-th element in \( S^l[t] \) and \( O^l[t] \) as \( S^l_j[t] \) and \( O^l_j[t] \) respectively, the gradient of the output of the \((l+k-1)\)-th residual block with respect to the input of the \( l \)-th residual block can be calculated layer by layer:

\[
\frac{\partial O^{l+k-1}[t]}{\partial S^l_j[t]} = \prod_{i=0}^{k-1} \frac{\partial O^{l+i}[t]}{\partial S^l_j[t]} = \prod_{i=0}^{k-1} \Theta'(S^{l+i}_j[t] - V_{ih}) \rightarrow \begin{cases} 
0, & \text{if } 0 < \Theta'(S^{l+i}_j[t] - V_{ih}) < 1 \\
1, & \text{if } \Theta'(S^{l+i}_j[t] - V_{ih}) = 1 \\
+\infty, & \text{if } \Theta'(S^{l+i}_j[t] - V_{ih}) > 1 
\end{cases},
\]

(8)

where \( \Theta(x) \) is the Heaviside step function and \( \Theta'(x) \) is defined by the surrogate gradient. The second equality hold as \( O^{l+k}[t] = \text{SN}(S^{l+k}[t]) \). In view of the fact that \( S^{l+k}_j[t] \) can only take 0 or 1, \( \Theta'(S^{l+k}_j[t] - V_{ih}) = 1 \) is not satisfied for commonly used surrogate functions mentioned in [40]. Thus, the vanishing/exploding gradient problems are prone to happen in deeper Spiking ResNet.

Based on the above analysis, we believe that the previous Spiking ResNet ignores the highly nonlinear caused by spiking neurons, and can hardly implement residual learning. Nonetheless, the basic block in Fig. 1(b) is still decent for ANN2SNN with extra normalization [17, 49], as the SNN converted from ANN aims to use firing rates to match the origin ANN’s activations.

### 3.3 Spike-Element-Wise ResNet

Here we propose the Spike-Element-Wise (SEW) residual block to realize the residual learning in SNNs, which can easily implement identity mapping and overcome the vanishing/exploding gradient problems at the same time. As illustrated in Fig. 1(c), the SEW residual block can be formulated as:

\[
O^l[t] = g(\text{SN}(\mathcal{F}^l(S^l[t])), S^l[t]) = g(A^l[t], S^l[t]),
\]

(9)

where \( g \) represents an element-wise function with two spikes tensor as inputs. Here we use \( A^l[t] \) to denote the residual mapping to be learned as \( A^l[t] = \text{SN}(\mathcal{F}^l(S^l[t])) \).

**SEW ResNet can easily implement identity mapping.** By utilizing the binary property of spikes, we can find different element-wise functions \( g \) that satisfy identity mapping (shown in Tab. 1). To be specific, when choosing \( \text{ADD} \) and \( \text{IAND} \) as element-wise functions \( g \), identity mapping is achieved by setting \( A^l[t] \equiv 0 \), which can be implemented simply by setting

<table>
<thead>
<tr>
<th>Name</th>
<th>Expression of ( g(A^l[t], S^l[t]) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{ADD} )</td>
<td>( A^l[t] + S^l[t] )</td>
</tr>
<tr>
<td>( \text{IAND} )</td>
<td>( A^l[t] \wedge S^l[t] )</td>
</tr>
<tr>
<td>( \text{IAND} )</td>
<td>( (\neg A^l[t]) \wedge S^l[t] )</td>
</tr>
</tbody>
</table>

Table 1: List of element-wise functions \( g \).
When the identity mapping is implemented, the gradient of the output of the block with respect to the input of the block will regulate the output to be no larger than 2 when \( g \) is \( IAND \) or \( ADD \), rather than the \( 0 \) or \( 1 \) as \( X \). When choosing \( ADD \) as \( g \), the infinite outputs problem can be alleviated, and the output of \( k \) sequential SEW blocks will be no larger than \( k \) in addition. In a downsample SEW block, the output will be no larger than \( 2 \) when \( g \) is \( ADD \).

When the identity mapping is implemented, the gradient of the output of the \((l + k - 1)\)-th SEW block with respect to the input of the \( l \)-th SEW block can be calculated layer by layer:

\[
\frac{\partial O^{l+k-1}[t]}{\partial S^l_j[t]} = \prod_{i=0}^{k-1} \frac{\partial g(A^l_{i}[t], S^l_{i}[t])}{\partial S^l_{i}[t]} \cdot 1 = \begin{cases} 
\prod_{i=0}^{k-1} \frac{\partial (0+S^l_{i}[t])}{\partial S^l_{i}[t]}, & \text{if } g = \text{ADD} \\
\prod_{i=0}^{k-1} \frac{\partial (1+S^l_{i}[t])}{\partial S^l_{i}[t]}, & \text{if } g = \text{AND} \\
\prod_{i=0}^{k-1} \frac{\partial ((1-0)+S^l_{i}[t])}{\partial S^l_{i}[t]}, & \text{if } g = I\text{AND} 
\end{cases} \quad (11)
\]

The second equality holds as identity mapping is achieved by setting \( A^{l+i}[t] = 1 \) for \( g = \text{AND} \), and \( A^{l+i}[t] = 0 \) for \( g = \text{ADD|IAND} \). Since the gradient in Eq. (11) is a constant, the SEW ResNet can overcome the vanishing/exploding gradient problems.

4 Experiments

4.1 ImageNet Classification

As the test server of ImageNet 2012 is no longer available, we can not report the actual test accuracy. Instead, we use the accuracy on the validation set as the test accuracy, which is the same as [17, 54].
He et al. [14] evaluated the 18/34/50/101/152-layer ResNets on the ImageNet dataset. For comparison, we consider the SNNs with the same network architectures, except that the basic residual block (Fig. 1(a)) is replaced by the spiking basic block (Fig. 1(b)) and SEW block (Fig. 1(c)) with $g$ as $ADD$, respectively. We denote the SNN with the basic block as Spiking ResNet and the SNN with the SEW block as SEW ResNet. The IF neuron model is adopted for the static ImageNet dataset. During training on ImageNet, we find that the Spiking ResNet-50/101/152 cannot converge unless we use the zero initialization [10], which sets all blocks to be an identity mapping at the start of training. Thus, the results of Spiking ResNet-18/34/50/101/152 reported in this paper are with zero initialization.

**Spiking ResNet vs. SEW ResNet.** We first evaluate the performance of Spiking ResNet and SEW ResNet. Tab. 2 reports the test accuracy on ImageNet validation. The results show that the deeper 34-layer Spiking ResNet has lower test accuracy than the shallower 18-layer Spiking ResNet. As the layer increases, the test accuracy of Spiking ResNet decreases. To reveal the reason, we compare the training loss, training accuracy, and test accuracy of Spiking ResNet during the training procedure, which is shown in Fig. 3. We can find the degradation problem of the Spiking ResNet — the deeper network has higher training loss than the shallower network. In contrast, the deeper 34-layer SEW ResNet has higher test accuracy than the shallower 18-layer SEW ResNet (shown in Tab. 2). More importantly, it can be found from Fig. 3 that the training loss of our SEW ResNet decreases and the training/test accuracy increases with the increase of depth, which indicates that we can obtain higher performance by simply increasing the network’s depth. All these results imply that the degradation problem is well addressed by SEW ResNet.

**Comparisons with State-of-the-art Methods.** In Tab. 3 we compare SEW ResNet with previous Spiking ResNets that achieve the best results on ImageNet. To our best knowledge, the SEW ResNet-101 and the SEW ResNet-152 are the only SNNs with more than 100 layers to date, and there are no other networks with the same structure to compare. When the network structure is the same, our SEW ResNet outperforms the state-of-the-art accuracy of directly trained Spiking ResNet, even with fewer
<table>
<thead>
<tr>
<th>Network</th>
<th>Methods</th>
<th>Accuracy(%)</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEW ResNet-34</td>
<td>Spike-based BP</td>
<td>67.04</td>
<td>4</td>
</tr>
<tr>
<td>Spiking ResNet-34(large)† with td-BN</td>
<td>Spike-based BP</td>
<td>67.05</td>
<td>6</td>
</tr>
<tr>
<td>Spiking ResNet-34 with td-BN [64]</td>
<td>Spike-based BP</td>
<td>63.72</td>
<td>6</td>
</tr>
<tr>
<td>Spiking ResNet-34 [12] ANN2SNN</td>
<td>ANN2SNN</td>
<td>69.89</td>
<td>4096</td>
</tr>
<tr>
<td>Spiking ResNet-34 [49] ANN2SNN</td>
<td>ANN2SNN</td>
<td>74.61</td>
<td>256</td>
</tr>
<tr>
<td>Spiking ResNet-34 [33] ANN2SNN and Spike-based BP</td>
<td>ANN2SNN and Spike-based BP</td>
<td>61.48</td>
<td>250</td>
</tr>
<tr>
<td>SEW ResNet-50</td>
<td>Spike-based BP</td>
<td>67.78</td>
<td>4</td>
</tr>
<tr>
<td>Spiking ResNet-50 with td-BN [64]</td>
<td>Spike-based BP</td>
<td>64.88</td>
<td>6</td>
</tr>
<tr>
<td>Spiking ResNet-50 [17] ANN2SNN</td>
<td>ANN2SNN</td>
<td>72.75</td>
<td>350</td>
</tr>
<tr>
<td>SEW ResNet-101</td>
<td>Spike-based BP</td>
<td>68.76</td>
<td>4</td>
</tr>
<tr>
<td>SEW ResNet-152</td>
<td>Spike-based BP</td>
<td>69.26</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3: Comparison with previous Spiking ResNet on ImageNet. † has the same network structure as the standard Spiking ResNet-34, but uses four times as many the number of convolution kernels.

Figure 4: Firing rates of $A_l$ in SEW blocks on ImageNet.

time-steps $T$. The accuracy of SEW ResNet-34 is slightly lower than Spiking ResNet-34 (large) with td-BN (67.04% v.s. 67.05%), which uses 1.5 times as many simulating time-steps $T$ (6 v.s. 4) and 4 times as many the number of parameters (85.5M v.s. 21.8M), compared with our SEW ResNet. The state-of-the-art ANN2SNN methods [33, 17] have better accuracy than our SEW ResNet, but they respectively use 64 and 87.5 times as many time-steps as ours.

Analysis of spiking response of SEW blocks. Fig. 4 shows the firing rates of $A_l$ in SEW ResNet-18/34/50/101/152 on ImageNet. There are 7 blocks in SEW ResNet-18, 15 blocks in SEW ResNet-34 and SEW ResNet-50, 33 blocks in SEW ResNet-101, and 50 blocks in SEW ResNet-152. The downsample SEW blocks are marked by the triangle down symbol ▽. As we choose ADD as element-wise functions $g$, a lower firing rate means that the SEW block gets closer to implementing identity mapping, except for downsample blocks. Note that the shortcuts of downsample blocks are not identity mapping, which is illustrated in Fig. 2(b). Fig. 4 shows that all spiking neurons in SEW blocks have low firing rates, and the spiking neurons in the last two blocks even have firing rates of almost zero. As the time-steps $T$ is 4 and firing rates are no larger than 0.25, all neurons in SEW ResNet-18/34/50 fire on average no more than one spike during the whole simulation. Besides, all firing rates in SEW ResNet-101/152 are not larger than 0.5, indicating that all neurons fire on average not more than two spikes. In general, the firing rates of $A_l$ in SEW blocks are at a low level, verifying that most SEW blocks act as identity mapping.

Gradients Check on ResNet-152 Structure. Eq. (8) and Eq. (11) analyze the gradients of multiple blocks with identity mapping. To verify that SEW ResNet can overcome vanishing/exploding gradient, we check the gradients of Spiking ResNet-152 and SEW ResNet-152, which are the deepest standard ResNet structure. We consider the same initialization parameters and with/without zero initialization. As the gradients of SNNs are significantly influenced by firing rates (see Sec.A.4), we analyze the firing rate firstly. Fig. 5(a) shows the initial firing rate of $O_l$’s output $O_l$. The indexes of downsample blocks are marked by vertical dotted lines. The blocks between two adjacent dotted lines represent the identity mapping areas, and have inputs and outputs with the same shape. When using zero initialization, Spiking ResNet, SEW AND ResNet, SEW IAND ResNet, and SEW ADD ResNet have the same firing rates (green curve), which is the zero init curve. Without zero initialization, the silence problem happens in the SEW AND network (red curve), and is relieved by the SEW IAND network (purple curve). Fig. 5(b) shows the firing rate of $A_l$, which represents the output of last SN in $l$-th block. It can be found that although the firing rate of $O_l$ in SEW ADD ResNet increases linearly in the identity mapping areas, the last SN in each block still maintains a stable firing rate. Note that when $g$ is ADD, the output of the SEW block is not binary, and the firing
rate is actually the mean value. The SNs of SEW IAND ResNet maintain an adequate firing rate and decay slightly with depth (purple curve), while SNs in deep layers of SEW AND ResNet keep silent (orange curve). The silence problem can be explained as follows. When using AND, \( O^l[t] = SN(F^l(O^{l-1}[t])) \land O^{l-1}[t] \leq O^{l-1}[t] \). Since it is hard to keep \( SN(F^l(O^{l-1}[t])) = 1 \) at each time-step \( t \), the silence problem may frequently happen in SEW ResNet with AND as \( g \). Using IAND as a substitute of AND can relieve this problem because it is easy to keep \( SN(F^l(O^{l-1}[t])) = 0 \) at each time-step \( t \).

The surrogate gradient function we used in all experiments is \( \sigma(x) = \frac{1}{2} \arctan(\frac{\alpha}{2} ax) + \frac{1}{2} \), thus \( \sigma'(x) = \frac{\alpha}{2(1 + (\frac{\alpha}{2} ax)^2)} \). When \( V_{th} = 1, \alpha = 2 \), the gradient amplitude \( \| \frac{\partial L}{\partial S^l} \| \) of each block is shown in Fig. 6. Note that \( \alpha = 2 \), \( \sigma'(x) \leq \sigma'(0) = \sigma'(1 - V_{th}) = 1 \) and \( \sigma'(0 - V_{th}) = 0.092 < 1 \). It can be found that the gradients in Spiking ResNet-152 decay from deeper layers to shallower layers in the identity mapping areas without zero initialization, which is caused by \( \sigma'(x) \leq 1 \). It is worth noting that the decay also happens in Spiking ResNet-152 with zero initialization. The small convex \( \land \) near the dotted lines is caused by the vanishing gradients of those \( S^l_j[t] = 0 \). After these gradients decays to 0 completely, \( \| \frac{\partial L}{\partial S^l} \| \) will be a constant because the rest gradients are calculated by \( S^l_j[t] = 1 \) and \( \sigma'(1 - V_{th}) = 1 \), which can also explain why the gradient-index curve is horizontal at some areas. When referring to SEW ResNet-152 with zero initialization, it can be found that all gradient-index curves are similar no matter what \( g \) we choose. This is caused by that in the identity mapping areas, \( S^l \) is constant for all index \( l \), and the gradient also becomes a constant as it will not flow through SNs. Without zero initialization, the vanishing gradient happens in the SEW AND ResNet-152, which is caused by the silence problem. The gradients of SEW ADD, IAND network increase slowly when propagating from deeper layers to shallower layers, due to the adequate firing rates shown in Fig. 5.

When \( V_{th} = 0.5, \alpha = 2 \), \( \sigma'(0 - V_{th}) = \sigma'(1 - V_{th}) = 0.288 < 1 \), indicating that transmitting spikes to SNs is prone to causing vanishing gradient, as shown in Fig. 7. With zero initialization, the decay in Spiking ResNet-152 is more serious because gradient from \( F^l \) can not contribute. The SEW ResNet-152 will not be affected no matter what \( g \) we choose. When \( V_{th} = 1, \alpha = 3 \), \( \sigma'(1 - V_{th}) = 1.5 > 1 \), indicating that transmitting spikes to SNs is prone to causing exploding gradient. Fig. 8 shows the gradient in this situation. Same with the reason in Fig. 6, the change of surrogate function will increase gradients of all networks without zero initialization, but not affect SEW ResNet-152 with zero initialization. The Spiking ResNet-152 meets exploding gradient, while this problem in SEW ADD, IAND ResNet-152 is not serious.

### 4.2 DVS Gesture Classification

The origin ResNet, which is designed for classifying the complex ImageNet dataset, is too large for the DVS Gesture dataset. Hence, we design a tiny network named 7B-Net, whose structure is c32k3s1-BN-PLIF-[SEW Block-MPk2s2]*7-FC11. Here c32k3s1 means the convolutional layer with channels 32, kernel size 3, stride 1. MPk2s2 is the max pooling with kernel size 2, stride 2. The symbol \([ \) *7 denotes seven repeated structure, and PLIF denotes the Parametric Leaky-Integrate-and-Fire Spiking Neuron with a learnable membrane time constant, which is proposed in [8] and can be described by Eq. 3. See Sec. A.1 for AER data pre-processing details.
Spiking ResNet vs. SEW ResNet. We first compare the performance of SEW ResNet with ADD element-wise function (SEW ADD ResNet) and Spiking ResNet by replacing SEW blocks with basic blocks. As shown in Fig. 9 and Tab. 4, although the training loss of Spiking ResNet (blue curve) is lower than SEW ADD ResNet (orange curve), the test accuracy is lower than SEW ADD ResNet (90.97% vs. 97.92%), which implies that Spiking ResNet is easier to overfit than SEW ADD ResNet.

Evaluation of different element-wise functions and plain block. As the training cost of SNNs on the DVS Gesture dataset is much lower than on ImageNet, we carry out more ablation experiments on the DVS Gesture dataset. We replace SEW blocks with the plain blocks (no shortcut connection) and test the performance. We also evaluate all kinds of element-wise functions $g$ in Tab. 1. Fig. 9 shows the training loss and training/test accuracy on DVS Gesture. The sharp fluctuation during early epochs is caused by the large learning rate (see Sec.A.1). We can find that the training loss is SEW IAND$<$Spiking ResNet$<$SEW ADD$<$Plain Net$<$SEW AND. Due to the overfitting problem, a lower loss does not guarantee a higher test accuracy. Tab. 2 shows the test accuracy of all networks. The SEW ADD ResNet gets the highest accuracy than others.

Comparisons with State-of-the-art Methods. Tab. 5 compares our network with SOTA methods. It can be found that our SEW ResNet outperforms the SOTA works in accuracy, parameter numbers, and simulating time-steps.

Table 4: Test accuracy on DVS Gesture. The networks’ order is ranked by accuracy.

<table>
<thead>
<tr>
<th>Network</th>
<th>Element-Wise Function $g$</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEW ResNet</td>
<td>ADD</td>
<td>97.92</td>
</tr>
<tr>
<td>SEW ResNet</td>
<td>IAND</td>
<td>95.49</td>
</tr>
<tr>
<td>Plain Net</td>
<td></td>
<td>91.67</td>
</tr>
<tr>
<td>Spiking ResNet</td>
<td></td>
<td>90.97</td>
</tr>
<tr>
<td>SEW ResNet</td>
<td>AND</td>
<td>70.49</td>
</tr>
</tbody>
</table>

Table 5: Comparison with the state-of-the-art (SOTA) methods on DVS Gesture dataset.

<table>
<thead>
<tr>
<th>Network</th>
<th>Accuracy (%)</th>
<th>Parameters</th>
<th>$T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>c32k3s1-BN-PLIF-[SEW Block (c32)-MPk2s2]*7-FC11 (7B-Net)</td>
<td>97.92</td>
<td>0.13M</td>
<td>16</td>
</tr>
<tr>
<td>[c128k3s1-BN-PLIF-MPk2s2]*5-DP-FC512-PLIF-DP-FC110-PLIF-APk10s10</td>
<td>97.57</td>
<td>1.70M</td>
<td>20</td>
</tr>
<tr>
<td>Spiking ResNet-17 with td-BN [64]</td>
<td>96.87</td>
<td>11.18M</td>
<td>40</td>
</tr>
<tr>
<td>MPk4-c64k3-PLIF-c128k3s1-BN-PLIF-c128k3s1-APk2s2-FC256-PLIF-APk10s10 [6]</td>
<td>93.40</td>
<td>23.23M</td>
<td>60</td>
</tr>
</tbody>
</table>

4.3 CIFAR10-DVS Classification

We also report SEW ResNet on the CIFAR10-DVS dataset, which is obtained by recording the moving images of the CIFAR-10 dataset on a LCD monitor by a DVS camera. As CIFAR10-DVS is more complicated than DVS Gesture, we use the network structure named Wide-7B-Net, which is similar to 7B-Net but with more channels. The structure of Wide-7B-Net is c64k3s1-BN-PLIF-[SEW Block (c64)-MPk2s2]*4-c128k3s1-BN-PLIF-[SEW Block (c128)-MPk2s2]*3-FC10.
In Tab[6] we compare SEW ResNet with the previous Spiking ResNet. One can find that our method achieves better performance (70.2% v.s. 67.8%) and fewer time-steps (8 v.s. 10) than the Spiking ResNet [64]. We also compare our method with the state-of-the-art (SOTA) supervised learning methods on CIFAR10-DVS. The accuracy of our Wide-7B-Net is slightly lower than the current SOTA method [8] (74.4% v.s. 74.8%), which uses 1.25 times as many simulation time-steps T (20 v.s. 16) and 14.6 times as many the number of parameters (17.4M v.s. 1.19M). Moreover, when reducing T shapely to T = 4, our Wide-7B-Net can still get the accuracy of 64.8%.

5 Conclusion

In this paper, we analyze the previous Spiking ResNet whose residual block mimics the standard block of ResNet, and find that it can hardly implement identity mapping and suffers from the problems of vanishing/exploding gradient. To solve these problems, we propose the SEW residual block and prove that it can implement the residual learning. The experiment results on ImageNet, DVS Gesture, and CIFAR10-DVS datasets show that our SEW residual block solves the degradation problem, and SEW ResNet can achieve higher accuracy by simply increasing the network’s depth. Our work may shed light on the learning of “very deep” SNNs.

6 Acknowledgment

This work is supported by grants from the National Natural Science Foundation of China under contracts No.62027804, No.61825101, and No.62088102.
References


13


