Efficient Training of Spiking Neural Networks with Multi-Parallel Implicit Stream Architecture (Supplementary Material)

Anonymous ECCV 2024 Submission

Paper ID #5068

1 More Details for the Experiments

The model architecture is denoted as $xFl*(aCbSd)-FCn$, where xFl indicates the number of implicit flows is x and the fusion position at layer l, $aCbSd$ represents a convolutional layer with output channels a, kernel size b, and stride d, and FCn stands for a fully connected layer with n output nodes.

1.1 Comparing with BPTT

The detailed hyperparameter settings for this section are shown in Tab. 1.

Table 1: Experimental details for the comparison with BPTT.

	Method	Size	Architecture	T	Bath Size	Environment	Optimizer
Fashion-MNIST	BPTT	133K	16C3S1-2F1*(32C3S2-48C3S1)-FC10	30 / 100	256	Windows 11, RTX 4060 Ti, Pytorch 2.1.	SGD
	MPIS	133K	16C3S1-2F1*(32C3S2-48C3S1)-FC10	30 / 100			
N-MNIST	BPTT	213K	2F1*(32C3S1-32C3S2-64C3S1)-FC10	30 / 100	512		
	MPIS	213K	2F1*(32C3S1-32C3S2-64C3S1)-FC10	30 / 100			

The calculation formula for the firing rate r_l for each layer is

$$r_l = \frac{s_l}{T \cdot n_l} \tag{1}$$

where r_l represents the firing rate for the $l-$th layer, T denotes the simulation time steps, n_l is the number of neurons in the $l-$th layer, and s_l is the number of spikes emitted by the $l-$th layer neuron within T time steps.

1.2 Comparing with the General Equilibrium SNNs

The detailed hyperparameter settings for this section are shown in Tab. 2.

Table 2: Experimental details for comparison with the general equilibrium SNNs.

	Method	Size	Architecture	T	Bath Size	Environment	Optimizer
CIFAR-10	IDE-Nets	11.8M	128C3S2-256C3S1-512C3S2-1024C3S1-512C3S1-FC10	30 / 100	128	Windows 11, RTX 3060 Ti, Pytorch 2.1.	SGD
	MPIS-SNNs	11.8M	120C3S1-120C3S2-2F3*(240C3S1-360C3S1-480C3S2)-FC10	30			
		28.5M	128C3S1-128C3S2-2F3*(256C3S1-512C3S1-1024C3S2)-FC10	30			
CIFAR-100	IDE-Nets	14.8M	128C3S2-256C3S1-512C3S2-1024C3S1-512C3S1-FC100	30 / 100			
	MPIS-SNNs	14.8M	120C3S1-120C3S2-2F3*(240C3S1-480C3S1-570C3S2)-FC100	30			
		30.0M	128C3S1-128C3S2-2F3*(256C3S1-512C3S1-1024C3S2)-FC10	30			

1.3 Comparing with the Latest Efficient Training Methods

The detailed hyperparameter settings for this section are shown in Tab. 3.

Table 3: Experimental details for comparison with the latest efficient training methods.

	Architecture	T	Bath Size	Environment	Optimizer
N-MNIST	2F1*(32C3S1-32C3S2-64C3S1)-FC10	30	128	Windows 11, RTX 3060 Ti, Pytorch 2.1.	SGD
Fashion-MNIST	128C3S1-2F1*(128C3S1-256C3S2)-FC10	1			
CIFAR-10	128C3S1-128C3S2-2F3*(256C3S1-512C3S1-1024C3S2)-FC10	10			
CIFAR-100	128C3S1-128C3S2-2F3*(256C3S1-512C3S1-1024C3S2)-FC100	5			

2 Limitations and Prospects

In this work, we have proposed an efficient training method for SNNs based on the theory of deep equilibrium models. Our method requires the calculation of the equivalent firing rate for a single step based on the equilibrium state of the input data, thus the dynamic nature of the input data affects the accuracy of the single-step equivalence. Specifically, when two sample categories are different but share the similar equilibrium state, the model may become confused. The occurrence of another scenario arises when the dynamics of the samples are excessively strong, leading to a failure in convergence of the input. Consequently, this results in an amplification of error in single-step equivalence. One potential solution is to treat time as an additional channel dimension for input, but this approach would lead to an increase in training memory cost as the length of time increases. Accomplishing precise single-step substitution for inputs with varying dynamics is a crucial task in applying equilibrium model theory to diverse types of SNNs. Our model offers the flexibility to adjust the resolution of feature maps according to specific requirements. Therefore, exploring the adaptation of our model to accommodate various visual tasks is a promising direction for further investigation.