Reproduction of Paper: Spike-driven Transformer

Project of AI3610 Brain-inspired Intelligence, 2023 Fall, SJTU

Xiangyuan Xue, Shengmin Yang, Yi Ai

December 26, 2023

Group 5

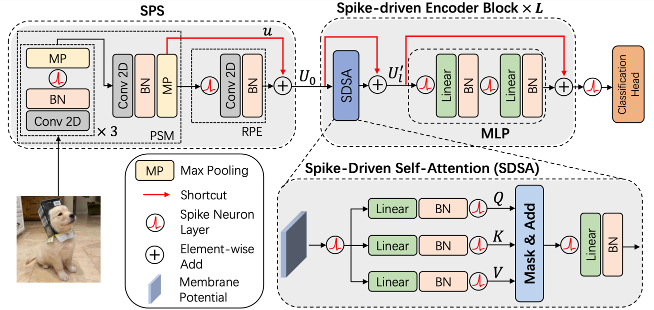
Introduction

Paper Highlight

- Spikformer proposes a spiking version of transformer, but it fails to exploit the full energy-efficiency potential of the spike-driven paradigm combined with self-attention.
- Spike-driven Transformer proposes a novel attention mechanism which only consists of mask and sparse addition operations with little energy consumption.
- Spike-driven Transformer is claimed to achieve the state-of-the-art performance on multiple common datasets and be friendly to neuromorphic hardware.

Method

Network Architecture: Overview



- Spike-driven Transformer includes 3 parts: spiking patch splitting, spike-driven encoder and classification head.

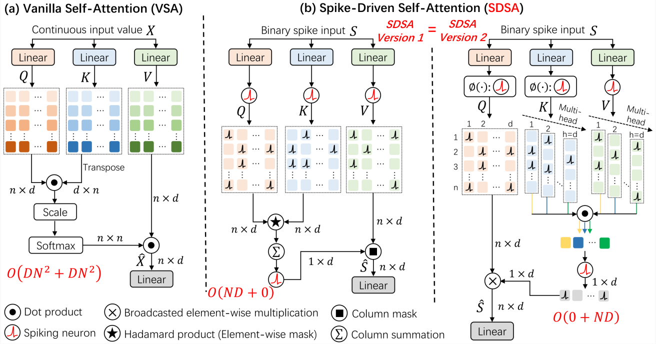
Network Architecture: Spiking Patch Splitting (SPS)

- The spiking patch splitting converts a 2D image sequence $I \in \mathbb{R}^{T\times C\times H\times W}$ into a membrane potential embedding.

$$
\begin{aligned}
u &= \mathrm{PSM}\left(I\right), && u \in \mathbb{R}^{T\times N\times D} \\
s &= \mathcal{SN}(u), && s \in \mathbb{R}^{T\times N\times D} \\
\mathrm{RPE} &= \mathrm{BN}(\mathrm{Conv2d}(s)), && \mathrm{RPE} \in \mathbb{R}^{T\times N\times D} \\
U_0 &= u + \mathrm{RPE}, && U_0 \in \mathbb{R}^{T\times N\times D}
\end{aligned}
$$

- PSM represents the patch splitting module. RPE represents the relative position embedding. $\mathcal{SN}(\cdot)$ denotes the spike neuron layer.

Network Architecture: Spike-Driven Self-Attention (SDSA)



- SDSA is different from the vanilla self-attention (VSA).

Method

Network Architecture: Spike-Driven Self-Attention (SDSA)

- VSA is formulated as:

$$\mathrm{VSA}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V$$

- SDSA is formulated as:

$$\mathrm{SDSA}(Q, K, V) = Q_S \otimes \mathcal{SN}\left(\mathrm{SUM_c}\left(K_S \otimes V_S\right)\right)$$

where $\otimes$ represents the Hadamard product. $\mathrm{SUM_c}(\cdot)$ represents the column-wise summation.

Method

Network Architecture: Spike-driven Encoder Block (SEB)

- The spike-driven encoder block follows the classical structure.

$$
\begin{aligned}
S_0 &= \mathcal{SN}(U_0), & S_0 &\in \mathbb{R}^{T\times N\times D} \\
U_l' &= \mathrm{SDSA}(S_{l-1}) + U_{l-1}, & U_l' &\in \mathbb{R}^{T\times N\times D} \\
S_l' &= \mathcal{SN}(U_l'), & S_l' &\in \mathbb{R}^{T\times N\times D} \\
S_l &= \mathcal{SN}(\mathrm{MLP}(S_l') + U_l'), & S_l &\in \mathbb{R}^{T\times N\times D}
\end{aligned}
$$

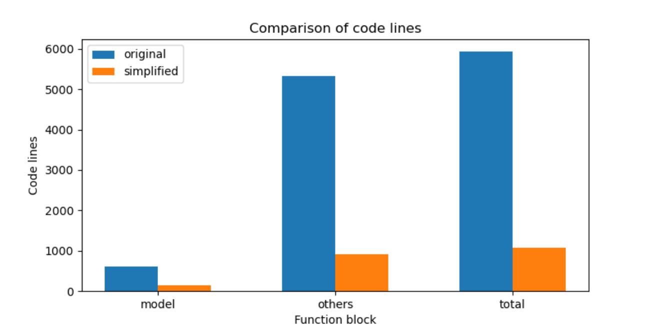
- The classification head utilizes a global average-pooling and a fully connected layer to output the prediction.

Method Innovation: Simplified Implementation

- The original implementation contains lots of redundant code and applies an outdated framework, lacking modern features.
- We use the latest version of `spikingjelly`, which supports multi-step propagation, thus significantly improving the computing efficiency.
- We utilize `einops` for powerful and flexible tensor operations to simplify the code and enhance the readability.

Method

Method Innovation: Simplified Implementation



- We implement all the function blocks within 1/6 code lines.

Method

Method Innovation: Transfer Learning

- Training SNNs on a GPU-based platform is extremely expensive compared to ANNs.
- We pretrain a model on the large-scale dataset to capture generic features and patterns, and then finetune it on the smaller datasets, which allows the model to adapt quickly.
- Transfer Learning can slightly improve the performance and significantly raise the training efficiency.

Experiments

Experiment Setup: Dataset

- We evaluate our model on both static datasets including CIFAR-10 and CIFAR-100 and neuromorphic datasets including CIFAR-10-DVS and DVS-128-Gesture.
- Considering the limited time and computing resources, we use the ImageNet-1k dataset only for a few rounds of pretraining instead of complete training.

Experiments

Experiment Setup: Implementation Details

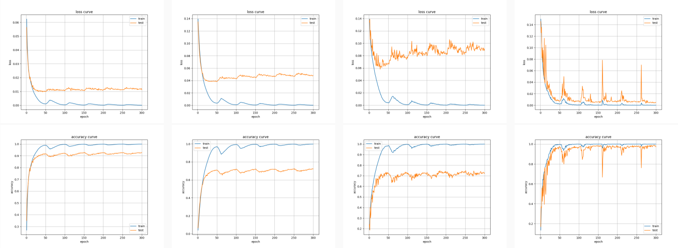
- We mainly use Spike-driven Transformer with 4 encoder blocks, 8 heads and 512 channels. The model is trained for 300 epochs on a single NVIDIA Tesla V100 GPU.
- We use the AdamW optimizer with a learning rate of 0.0005. The learning rate is warmed up for 10 epochs and then decayed by cosine annealing for 40 epochs.
- The batch size is 32 for static datasets and 16 for neuromorphic datasets. The number of timesteps is 4 for static datasets and 16 for neuromorphic datasets.

Result Analysis

- With only simple data augmentation employed, the metrics on all the datasets are close to those in the paper.

Dataset	#Class	#Timestep	Accuracy
CIFAR-10	10	4	92.90%
CIFAR-100	100	4	72.44%
CIFAR-10-DVS	10	16	75.00%
DVS-128-Gesture	11	16	98.96%

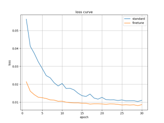
Result Analysis

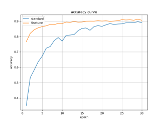


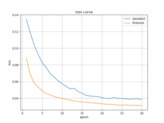
- The training curves on the CIFAR-10, CIFAR-100, CIFAR-10-DVS and DVS-128-Gesture datasets show stable trends influenced by the adjustment of the learning rate.

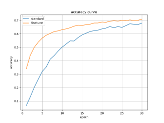
Experiments

Transfer Learning









- With transfer learning, the model can achieve satisfactory performance within several epochs of finetuning.

Dataset	Standard	Finetune
CIFAR-10	89.56%	**91.25%**
CIFAR-100	67.98%	**70.90%**

Conclusion

Conclusion

- We successfully reproduce the paper and verify the correctness of the paper.
- We utilize the latest frameworks to simplify the implementation and introduce the method of transfer learning to improve training efficiency.
- We conduct experiments to prove the effectiveness of our method innovation.

Thank You!