TEAS: Exploiting Spiking Activity for Temporal-wise Adaptive Spiking Neural Networks

Authors: Fangxin Liu=, Haomin Li=, Ning Yang, Zongwu Wang, Tao Yang , and **Li Jiang***

Shanghai Jiao Tong University

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

先进计算机体系结构实验室
Advanced Computer Architecture Laboratory

Outline

- Background and motivation
- Design and implementation details
- Experiment setting &results
- Conclusion

Background and motivation

Spiking Neural Network

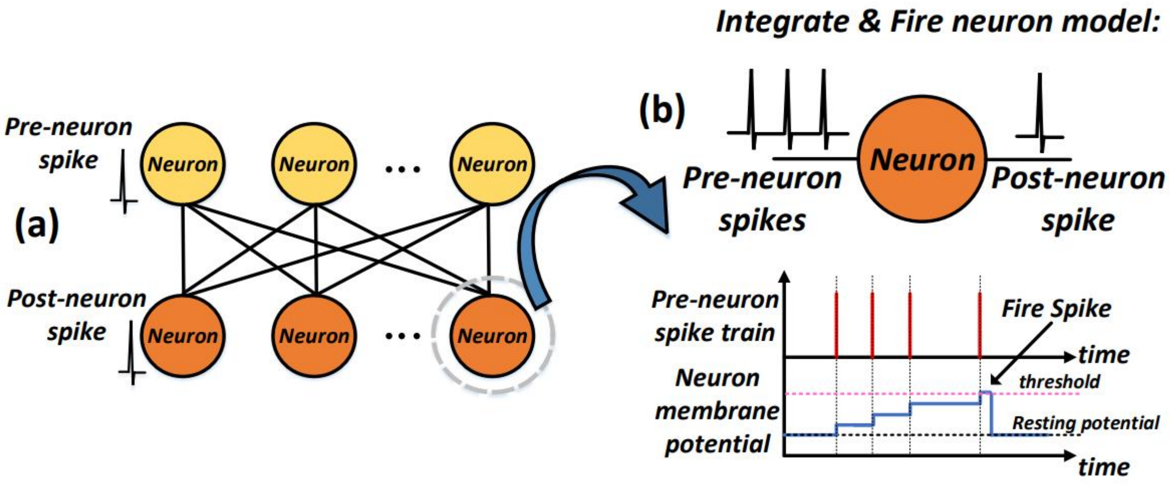
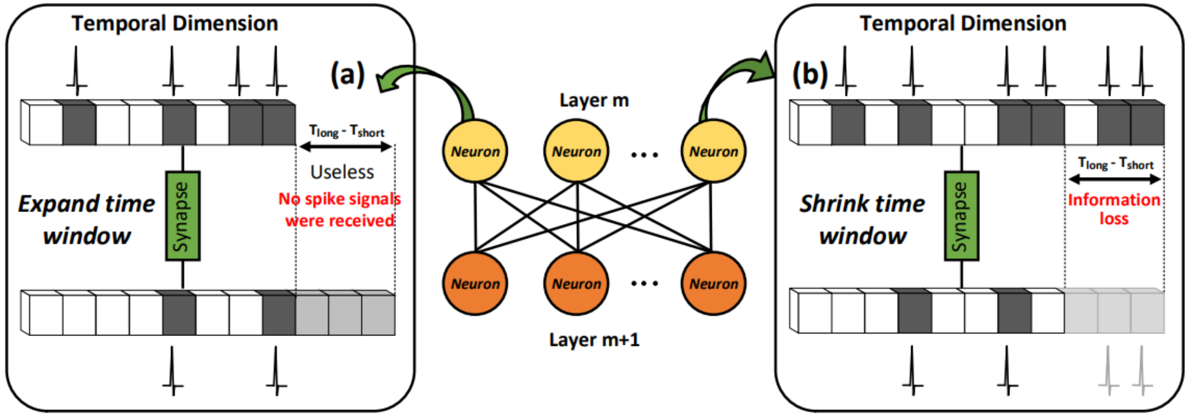


Illustration of Linear-Integrate and Fire (IF) neuron dynamics

- The spiking neuron is the fundamental computing unit of SNN.
- An IF neuron receives train of spikes, over certain time window. Input spikes (‘0’ or ‘1’ spike signals) from pre-synaptic neurons multiply the synaptic weight.

Background and motivation

Motivation



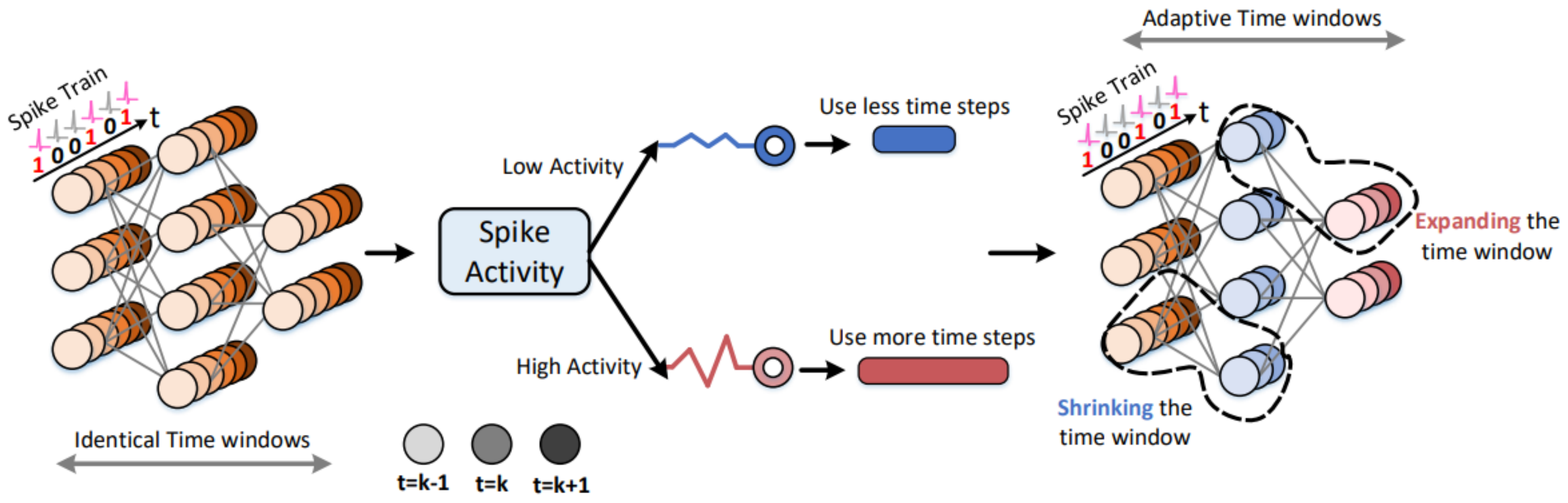
An example of temporal alignment.

- **Limitations of previous works:** These previous SNNs are developed based on the temporal alignment for the network.

- **Temporal-wise adaptive SNNs:** The key to achieving efficient SNNs in trading-off the processing efficiency.

Design and implementation details

Identifying Vulnerable Layers in the SNN



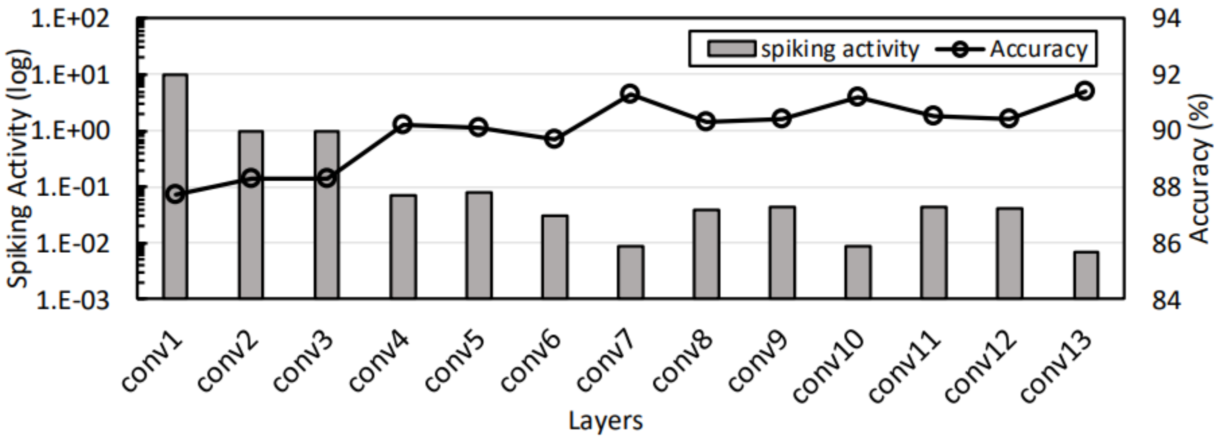
The overview of TEAS

- **Two questions**: 1) whether different layers have different degrees of sensitivity to the reduced time steps and 2) how to utilize the difference to maintain the accuracy while shortening the time step.
- **Studying**: Shortening the time steps of the spike train affects the accuracy of the SNN, and certain layers, termed as vulnerable layers, require more temporal information to maintain accuracy, while invulnerable layers can have different time steps without accuracy loss.

ASIA SOUTH PACIFIC DAC DESIGN AUTOMATION CONFERENCE

Design and implementation details

Identifying Vulnerable Layers in the SNN



Analysis of Spiking Activity and Accuracy of a SNN with VGG-16 structure trained on CIFAR-10 dataset, under input noise (u = 0.1 that represents the magnitude of noise).

- With the circle curve representing network accuracy affected by noise in specific layers, while the bars indicate the spiking activity (average spike count over time) in those layers.
- The impact of layers on the accuracy of SNN can vary, with layers having higher spiking activity having a greater impact, and by limiting the temporal impact on vulnerable layers, the accuracy of the SNN can be preserved while allowing for more relaxed.

ASIA SOUTH PACIFIC DAC DESIGN AUTOMATION CONFERENCE

Design and implementation details

Adaptive Conversion in the Temporal Dimension

How to conduct the spike train scaling at runtime?

Combing the relationship between the firing time or spike count with the strength of the information, we have:

$$\tilde{x}_T = x + \frac{V_{th}}{V_{fire}}$$

$$\tilde{x}_R = x + 1 - \frac{V_{th}}{V_{fire}}$$

Following this adjustment, we can then apply a linear mapping to determine the input spike time for the subsequent layer.

Design and implementation details

Training Process

☐ Objective of TEAS training

- ***Regularization term***: To encourage fewer time steps of layers, we introduce a regularization term into the training loss.

$$\ell_r = \sum_{l=1}^{L} \frac{\sum_{l=1}^{L} \#Act(l) \times \#\,\text{Para}\,(l)}{\#Act(l)}$$

- ***The cross-entropy loss of the SNNs***:

$$\mathcal{L}_{CE} = -\sum_{i} y_i \log\left(\frac{e^{u_i^T}}{\sum_{k=1}^{N} e^{u_k^T}}\right)$$

- **The total loss function**: The total loss function is the combination of cross-entropy loss and regularization:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \ell_r$$

☐ Temporal Adjustment.

Parameters	Descriptions	Configurations
N_p	# Epochs for pretraining	150
N_r	# Epochs for retraining	30
Batch Size	-	$16 \sim 64$
V_{th}	Threshold Potential	$0.7 \sim 10$
V_{reset}	Rest Potential	0.0
decay coefficient	Adam Decay	0.99
lr	Learning Rate	5e-4/1e-4

- The temporal adjustments for each layer in the SNN are made by calculating the spiking activity during the forward pass and trimming time steps based on the activity level.
- The regularization strength allocated to each layer varies with the temporal dimension.
- After determining the temporal dimensions, fine-tuning is performed to improve the network accuracy, starting from a pre-trained model.

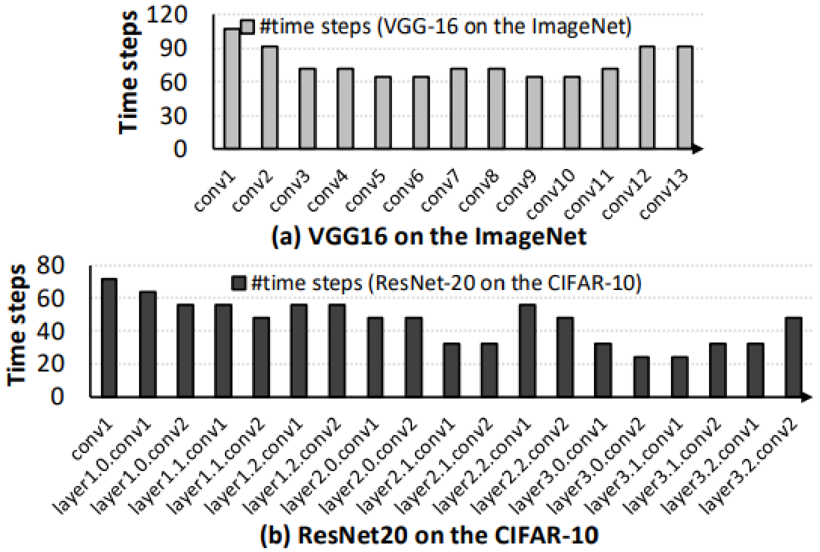
ASIA SOUTH PACIFIC DESIGN AUTOMATION CONFERENCE

Experiment Results

Inference performance

SNN	Structure	Dataset	ANN Acc.	SNN Acc.	Ori. T	Reduced Acc.	Reduced T
Directly Trained SNN [15]	7 Conv, 2 FC	CIFAR-10	-	91.31	16	91.31	6.2
Directly Trained SNN [5]	6 Conv, 2 FC	CIFAR-10	-	92.84	8	92.80	5.3
ANN-Converted SNN [4]	ResNet-20	CIFAR-100	68.72	68.18	2048	68.15	78.2
ANN-Converted SNN [4]	VGG-16	CIFAR-100	71.22	70.97	2048	70.95	71.3
ANN-Converted SNN [4]	ResNet-34	ImageNet	70.64	69.93	4096	69.87	83.1
ANN-Converted SNN [4]	VGG-16	ImageNet	73.49	73.46	2560	73.41	76.9

Classification Accuracy (%) on CIFAR10, CIFAR100, and ImageNet.



Layer-wise time steps of SNNs with VGG16 structure on ImageNet (a) and ResNet20 structure on CIFAR-10 (b).

- For ANN converted SNNs with the VGG-16 network structure, we reduce the averaged time steps to 40.5, 71.3, and 77.3 on CIFAR-10, CIFAR-100, and ImageNet, respectively, achieving 0.03% accuracy loss on average, where the maximum accuracy loss is 0.05%.
- TEAS can easily find better tradeoff points with both accuracy and the size of temporal dimension.

ASIA SOUTH PACIFIC DAC DESIGN AUTOMATION CONFERENCE

Experiment Results

Comparison to other competing methods.

Method	Structure	ANN-SNN Conversion (T=2500)	Temporal Comp.	Acc. (%)	Time Steps
		CIFAR-10			
[13]	11 layer CNN	-	312.5×	89.53	8
this work	**7 Conv, 2 FC**	**-**	403.2×	**91.31**	**6.2**
[18]	VGG16	92.48%	25×	91.29	100
[19]	VGG16	92.48%	25×	90.2	100
[19]	ResNet20	92.94%	10×	91.12	250
[20]	VGG9	-	52×	89.94	48
[21]	VGG16	92.48%	3.7×	91.40	680
this work	**VGG16**	**92.48%**	54.9×	**91.43**	**45.5**
this work	**ResNet20**	**92.94%**	61.7×	**93.66**	**40.5**
		CIFAR-100			
[18]	VGG11	70.94%	20.8×	64.98	120
[19]	VGG11	70.94%	20×	65.52	125
[20]	VGG11	-	52×	68.30	48
[21]	VGG16	70.97%	3.7×	68.80	680
this work	**VGG16**	**70.97%**	35.1×	**70.95**	**71.3**
		ImageNet			
[19]	VGG16	68.12%	10×	56.87	250
[19]	ResNet34	65.1%	10×	62.73	250
this work	**VGG16**	**73.46%**	32.5×	**73.41**	**76.9**
this work	**ResNet34**	**69.93%**	30.1×	**69.87**	**83.1**

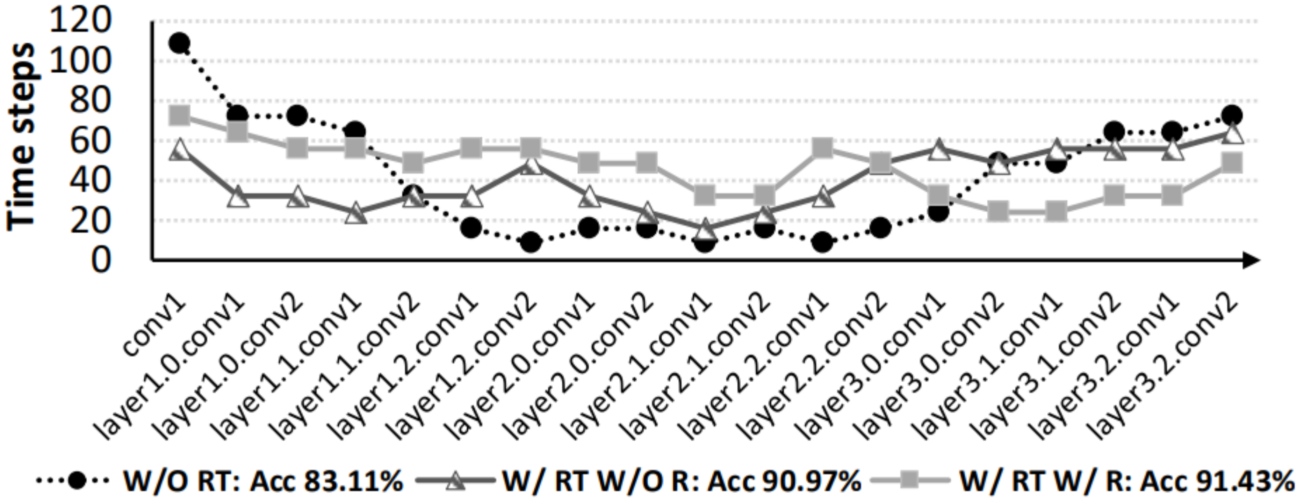
Comparion of our work with other SNN models on CIFAR-10, CIFAR-100 and ImageNet datasets

- Regarding ResNet-20 on the CIFAR-10, TEAS achieves 61.7× time steps reduction with nearly no accuracy loss compared to the original SNN (2500 time steps).
- For CIFAR-100, TEAS algorithm shows a 0.02% accuracy loss compared to the original SNN and a significant 6.3% accuracy improvement over the state-of-art temporal compression method with 40% fewer time steps.
- F or the more complex task ImageNet, the adaptive time window among layers makes our TEAS more robust for preserving SNN accuracy than the SNNwith the small windows .

The proposed method outperforms previous works

Experiment Results

Effect of retraining



Layer-wise time steps comparison under TEAS schemes achieved with or without retraining. W/O RT, W RT W/O R and WRT W R indicate without retraining, retraining without regularization term and retraining with regularization term, respectively.

- Training without the term will lead to over-penalization on earlier layers with higher spiking activities (vulnerable layers), while later layers with fewer spiking activities (invulnerable layers) are not compressed enough.

Experiment Results

Computation efficiency of SNNs

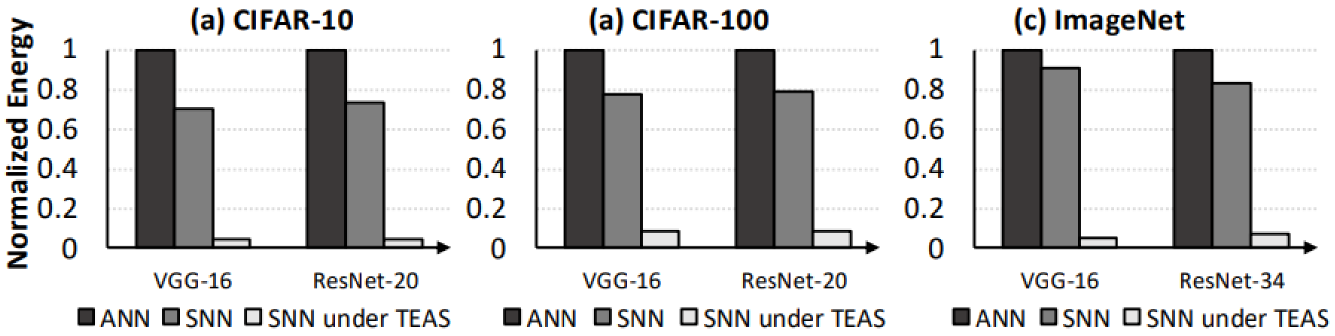


Fig. 7: Comparison of ANN, SNN and SNN under TEAS in terms of normalized energy efficiency with VGG and ResNet network structure on CIFAR-10 (a), CIFAR-100 (b) and ImageNet (c).

Comparison of ANN, SNN and SNN under TEAS in terms of normalized energy efficiency with VGG and ResNet network structure on CIFAR-10 (a), CIFAR-100 (b) and ImageNet (c).

- For VGG-16, the SNN under TEAS provides 11.5× higher energy efficiency compared to ANN and SNN with the identical temporal dimension.

The energy efficiency can reach up to 22.9× on average, as opposed to ANN models having similar parameters

Conclusion

- TEAS is the very first work for dynamic conversion on the temporal dimension of SNNs, and sheds early lights on why the control of the temporal dimension for SNNs needs more attention.
- Our method uses spiking activities of layers, which can be easily integrated and trained in an end-to-end fashion, to guide the temporal dimension optimization.
- We propose temporal conversion to skip the redundant temporal dimension at run-time.

Thank you !

TEAS: Exploiting Spiking Activity for Temporal-wise Adaptive Spiking Neural Networks

Fangxin Liu=, Haomin Li=, Ning Yang, Zongwu Wang, Tao Yang , and Li Jiang*

Shanghai Jiao Tong University

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

ASIA SOUTH PACIFIC DAC DESIGN AUTOMATION CONFERENCE