A Progressive Training Framework for Spiking Neural Networks with Learnable Multi-hierarchical Model

Zecheng Hao, Xinyu Shi, Zihan Huang, Tong Bu, Zhaofei Yu, Tiejun Huang
Peking University

The representation capability of the LIF & LM-H model

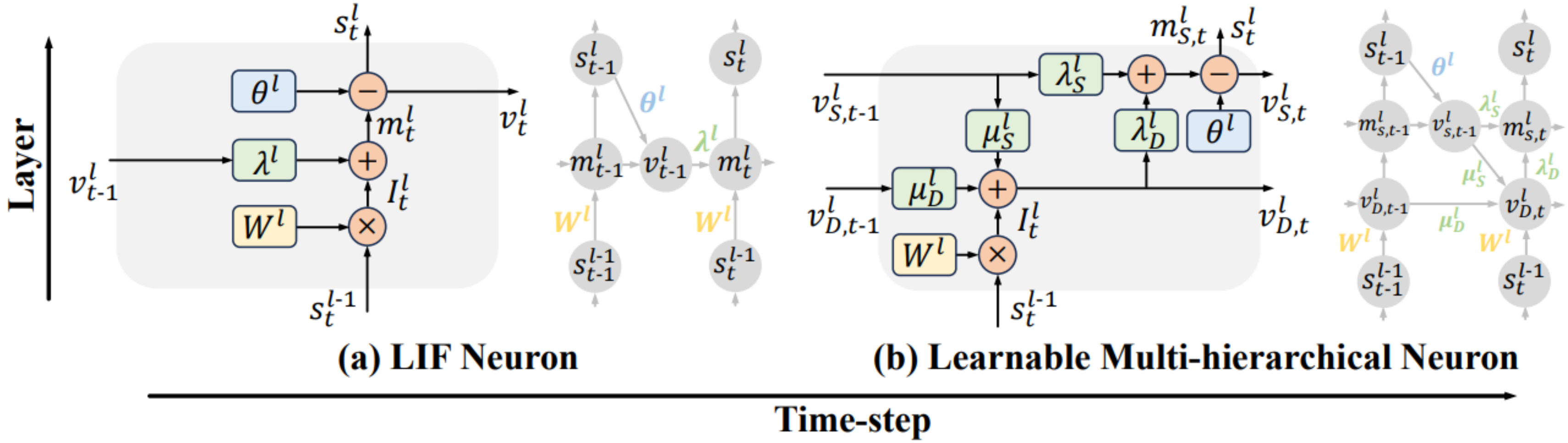


Figure 1: The structural description for vanilla LIF and LM-H models.

- Representation defects of the LIF model
  - The gradient vanishing & exploding problem in deep residual architectures. $\prod_{l=1}^{L} H'(\boldsymbol{s}^{l-1}[t] - \theta^l) \to 0$ or $\prod_{l=1}^{L} H'(\boldsymbol{s}^{l-1}[t] - \theta^l) \to +\infty$
  - The inability to differentiate the current response through extracting past information.

The representation capability of the LIF & LM-H model

$$\boldsymbol{v}_D^l[t] = \mu_D^l \boldsymbol{v}_D^l[t-1] + \mu_S^l \boldsymbol{v}_S^l[t-1] + \boldsymbol{I}^l[t].$$

$$\boldsymbol{v}_S^l[t] = \boldsymbol{m}_S^l[t] - \boldsymbol{s}^l[t]\theta^l, \quad \boldsymbol{m}_S^l[t] = \lambda_S^l \boldsymbol{v}_S^l[t-1] + \lambda_D^l \boldsymbol{v}_D^l[t].$$

$$\boldsymbol{I}^l[t] = \boldsymbol{W}^l \boldsymbol{s}^{l-1}[t]\theta^{l-1}, \quad \boldsymbol{s}^l[t] = H(\boldsymbol{m}_S^l[t] - \theta^l) = \begin{cases} 1, & \text{if } \boldsymbol{m}_S^l[t] \geq \theta^l \\ 0, & \text{otherwise} \end{cases}$$

transform the LM-H model into a single-layer form

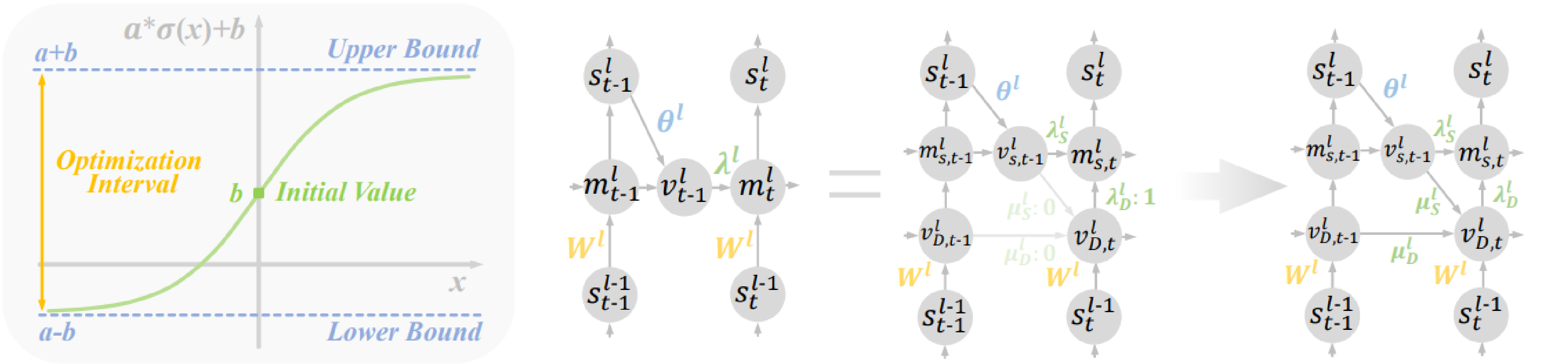
$$\begin{aligned}
\boldsymbol{v}_D^l[t] &= \mu_S^l \boldsymbol{v}_S^l[t-1] + \boldsymbol{I}^l[t] + \mu_D^l \boldsymbol{v}_D^l[t-1] \\
&= (\mu_S^l \boldsymbol{v}_S^l[t-1] + \boldsymbol{I}^l[t]) + \mu_D^l(\mu_S^l \boldsymbol{v}_S^l[t-2] + \boldsymbol{I}^l[t-1]) + (\mu_D^l)^2 \boldsymbol{v}_D^l[t-2] \\
&= \sum_{k=1}^{t} (\mu_D^l)^{t-k} (\mu_S^l \boldsymbol{v}_S^l[k-1] + \boldsymbol{I}^l[k]), \\
\boldsymbol{m}_S^l[t] &= \lambda_S^l \boldsymbol{v}_S^l[t-1] + \lambda_D^l \boldsymbol{v}_D^l[t] \\
&= \underbrace{\sum_{k=1}^{t-1} \lambda_D^l (\overset{1}{\boxed{\mu_D^l}})^{t-k} (\overset{2}{\boxed{\mu_S^l}} \boldsymbol{v}_S^l[k-1] + \boldsymbol{I}^l[k])}_{\text{historical representation}} + \underbrace{((\overset{3}{\boxed{\lambda_S^l}} + \lambda_D^l \mu_S^l) \boldsymbol{v}_S^l[t-1] + \overset{4}{\boxed{\lambda_D^l}} \boldsymbol{I}^l[t])}_{\text{current representation}}.
\end{aligned}$$

μ_D^l: historical information
μ_S^l: historical membrane potential
λ_S^l: current membrane potential
λ_D^l: input current & the calculation of residual architecture

$\prod_{l=1}^{L} \lambda_D^l H'(\lambda_D^l \boldsymbol{s}^{l-1}[t] - \theta^l)$

Conclusion: the LM-H model can overcome the representation defects of vanilla LIF model, the LIF model is actually a special case of the LM-H model.

A progressive STBP training for the LM-H model



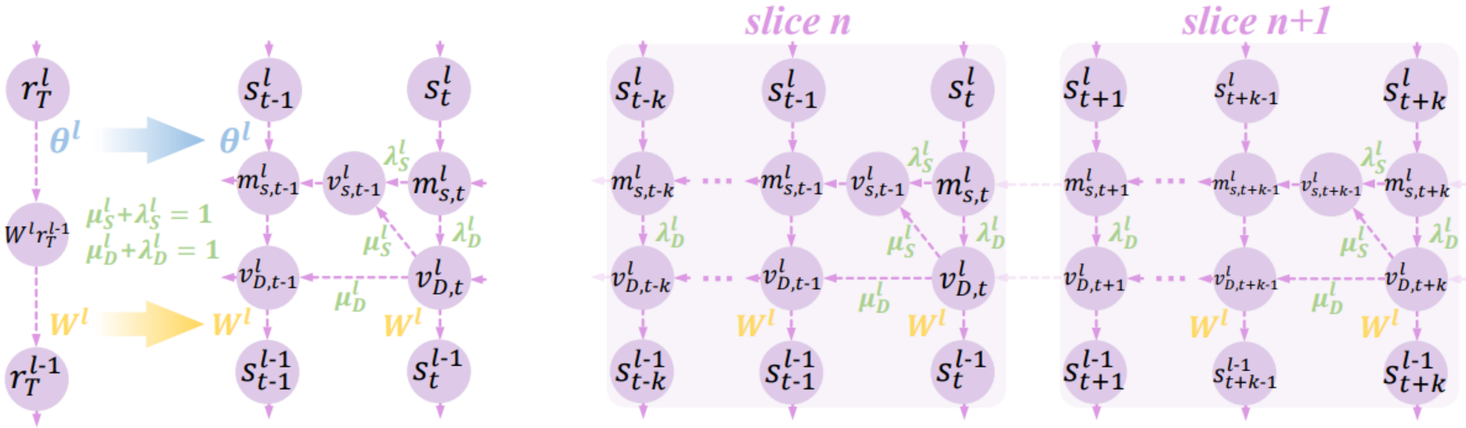
(a) Learning Function for Membrane Parameter

(b) From Initial State (IF/LIF Neuron) to Optimal State (LM-H Neuron)

Figure 3: Progressive STBP Training framework for the LM-H Neuron.

- Set the LM-H model as vanilla LIF model under the initial state
- Dynamically optimize the membrane-related parameters during the learning process to achieve more reasonable extraction of historical and current information

Efficient training based on the LM-H model



(a) Hybrid Training (b) Time-slicing Online Training

1. **Hybrid Training:** ANN-SNN Conversion + STBP Training ($\leq$ 30 epochs)
2. **Time-slicing Online Training:** Each k steps form a slice and update the gradients once

Figure 4: Efficient Training framework for LM-H Neuron.

$$\frac{\sum_{t=1}^{T} \boldsymbol{s}^l[t]\theta^l}{T} = \frac{\sum_{t=1}^{T} \boldsymbol{W}^l \boldsymbol{s}^{l-1}[t]\theta^{l-1}}{T} - \left(\frac{\sum_{t=1}^{T} \boldsymbol{\delta}_D^l[t]}{T} + \frac{\sum_{t=1}^{T} \boldsymbol{\delta}_S^l[t]}{T} \right),$$

$$\boldsymbol{\delta}_D^l[t] = (1-\lambda_D^l)\boldsymbol{v}_D^l[t] - \mu_D^l \boldsymbol{v}_D^l[t-1], \quad \boldsymbol{\delta}_S^l[t] = \boldsymbol{v}_S^l[t] - (\lambda_S^l + \mu_S^l)\boldsymbol{v}_S^l[t-1].$$

$$\boldsymbol{r}^l[T] = \boldsymbol{W}^l \boldsymbol{r}^{l-1}[T] - \left(\frac{(1-\lambda_D^l)\boldsymbol{v}_D^l[T] - \mu_D^l \boldsymbol{v}_D^l[0]}{T} + \frac{\boldsymbol{v}_S^l[T] - \boldsymbol{v}_S^l[0]}{T} \right).$$

$$\boldsymbol{r}^l[T] = \boldsymbol{W}^l \boldsymbol{r}^{l-1}[T] - \frac{\boldsymbol{v}_S^l[T] - \boldsymbol{v}_S^l[0]}{T}$$

Experiments: STBP training

Table 1: Comparison with previous SOTA works. * denotes an improved version of the network structure.

Dataset	Method	Architecture	Time-steps	Accuracy(%)
CIFAR-10	STBP-tdBN (Zheng et al., 2021)	ResNet-19	4	92.92
	Dspike (Li et al., 2021)	ResNet-18	4	93.66
	TET (Deng et al., 2022)	ResNet-19	4	94.44
	GLIF (Yao et al., 2022)	ResNet-18	4, 6	94.67, 94.88
		ResNet-19	4, 6	94.85, 95.03
	Ours	**ResNet-18**	**4**	**95.62**
		ResNet-19	**4**	**96.36**
CIFAR-100	Dspike (Li et al., 2021)	ResNet-18	4	73.35
	TET (Deng et al., 2022)	ResNet-19	4	74.47
	GLIF (Yao et al., 2022)	ResNet-18	4, 6	76.42, 77.28
		ResNet-19	4, 6	77.05, 77.35
	TEBN (Duan et al., 2022)	ResNet-19*	4, 6	78.71, 78.76
	Ours	**ResNet-18**	**4**	**78.58**
		ResNet-19	**4**	**80.31**
		ResNet-19*	**4**	**81.65**
ImageNet-200	DCT (Garg et al., 2020)	VGG-13	125	56.90
	Online-LTL (Yang et al., 2022)	VGG-13	16	54.82
	Offline-LTL (Yang et al., 2022)	VGG-13	16	55.37
	ASGL (Wang et al., 2023)	VGG-13	4, 8	56.57, 56.81
	Ours	**VGG-13**	**4**	**59.93**
	Ours (radical version)	**VGG-13**	**4**	**60.37**
DVS-CIFAR10	STBP-tdBN (Zheng et al., 2021)	ResNet-19	10	67.80
	RecDis-SNN (Guo et al., 2022)	ResNet-19	10	72.42
	MPBN (Guo et al., 2023)	ResNet-19	10	74.40
	Ours	**ResNet-19**	**10**	**79.10**

Experiments: efficient training

Table 2: Performance of hybrid training for LM-H model.

Dataset	Method	Architecture	Time-steps	Accuracy(%)
CIFAR-10	QCFS (Bu et al., 2022)	ResNet-18	4	93.66
		ResNet-20	4	83.75
	Ours	**ResNet-18**	**4**	**94.02**
		ResNet-20	**4**	**87.56**
CIFAR-100	QCFS (Bu et al., 2022)	VGG-16	4	69.62
		ResNet-20	4	34.14
	Ours	**VGG-16**	**4**	**73.11**
		ResNet-20	**4**	**57.12**
ImageNet-200	QCFS (Bu et al., 2022)	VGG-13	4	45.15
	Ours	**VGG-13**	**4**	**49.09**

Table 3: Performance of time-slicing online training on ResNet-18.

Dataset	Method	Time-steps	Accuracy(%)
CIFAR-10	SLTT (Meng et al., 2023)	6	94.44
	3 time-steps per slice, 2 slices	**4, 6, 8**	**95.05, 95.42, 95.49**
CIFAR-100	SLTT (Meng et al., 2023)	6	74.38
	2 time-steps per slice, 2 slices	4, 6, 8	76.27, 77.10, 77.56
	2 time-steps per slice, 3 slices	4, 6, 8	75.99, 77.35, 77.81
	2 time-steps per slice, 4 slices	4, 6, 8	74.81, 76.28, 77.01
	3 time-steps per slice, 2 slices	**4, 6, 8**	**77.28, 78.21, 78.66**
	4 time-steps per slice, 2 slices	4, 6, 8	77.23, 78.30, 78.59

Discussion & Conclusion

- We identify the limitations of the vanilla LIF model in terms of its representation capabilities and propose the LM-H model with a wider calculation scope. We mathematically demonstrate that our proposed model can effectively extracting global information along the time dimension and propagate gradients in deep networks.
- We systematically analyze the specific roles of parameters on the dendrite and soma layers, and further develop a progressive STBP training algorithm for the LM-H model, which can dynamically optimize the membrane-related parameters during the learning process.
- To enhance the energy efficiency of SNN learning, we propose an efficient training framework specifically designed for the LM-H model, which includes hybrid training and time-slicing online training.
- Experimental results validate the significant advantages of the LM-H model in the field of SNN supervised learning. Our proposed method achieves state-of-the-art performance on multiple datasets with various scales and data types.

Thanks for Listening!