LMUFormer: Low Complexity Yet Powerful Spiking Model With Legendre Memory Units

[1,*]Zeyu Liu, [1,2,*]Gourav Datta, [1]**Anni Li**, [1]Peter A. Beerel
[1]University of Southern California
[2]Currently employed at Amazon Inc.
*Equally contributing authors

USC Viterbi School of Engineering
Ming Hsieh Department of Electrical and Computer Engineering

Motivation

Transformers ❌

$$\text{Attention}(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

- **Quadratic** computational and memory complexities w.r.t. sequence length N.
- Global self-attention mechanism needs to process the entire sequence, which yields **high latency** for real-time **streaming** applications.
- **Stateless**

RNN ❌

- **Higher training time** because training must accommodate the long sequence of dependencies within the model, making parallelization more difficult.
- Traditionally suffer from **forgetting** due to having a limited memory horizon.

Background

Legendre Memory Unit (LMU)[1]

- A memory cell that efficiently captures and represent **temporal dependencies** in sequential data.
- Utilizes the mathematical properties of **Legendre polynomials**.
- Based on two state-space matrices (A, B) that approximate a linear transfer function in continuous time / discrete time:

$$\dot{\boldsymbol{m}}(t) = \boldsymbol{A}\boldsymbol{m}(t) + \boldsymbol{B}\boldsymbol{u}(t)$$
$$\boldsymbol{m}[t] = \bar{\boldsymbol{A}}\boldsymbol{m}[t-1] + \bar{\boldsymbol{B}}\boldsymbol{u}[t]$$

- To enable **parallelization**, we get u[t] and output of LMU as follows[2] to make the module a linear time-invariant (LTI) system:

$$\boldsymbol{u}[t] = Act_u\left(\boldsymbol{W}_u\boldsymbol{x}[t] + \boldsymbol{b}_u\right)$$
$$\boldsymbol{o}[t] = Act_o\left(\boldsymbol{W}_m\boldsymbol{m}[t] + \boldsymbol{W}_x\boldsymbol{x}[t] + \boldsymbol{b}_o\right)$$

Spiking Neural Network (SNN)

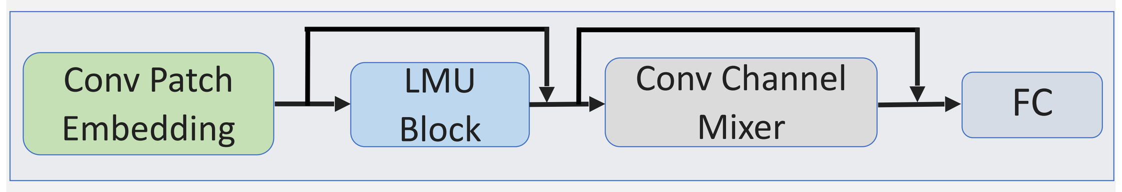
- Uses binary "spikes" to process and transmit information.
- We use Leaky Integrate-and-Fire (LIF) [3] neurons to get the membrane potentials:

$$\boldsymbol{u}_l^t = \lambda\boldsymbol{u}_l^{t-1} + \boldsymbol{w}_l\boldsymbol{o}_{l-1}^t - v_l^{th}\boldsymbol{o}_{l-1}^{t-1}$$
$$\boldsymbol{o}_l^{t-1} = \begin{cases} 1, & \text{if } \boldsymbol{u}_l^{t-1} \geq v_l^{th} \\ 0, & otherwise \end{cases}$$

$\boldsymbol{u}_l^t$: Membrane potential tensor of l^{th} layer at t^{th} time step
λ: Leak factor between [0, 1]
$\boldsymbol{w}_l$: The weight connecting layers $l-1$ and l
$\boldsymbol{o}_{l-1}^t$: Spike output of $(l-1)^{\text{th}}$ layer at t^{th} time step
$\boldsymbol{v}_l^{\text{th}}$: Constant threshold for layer l

LMUFormer

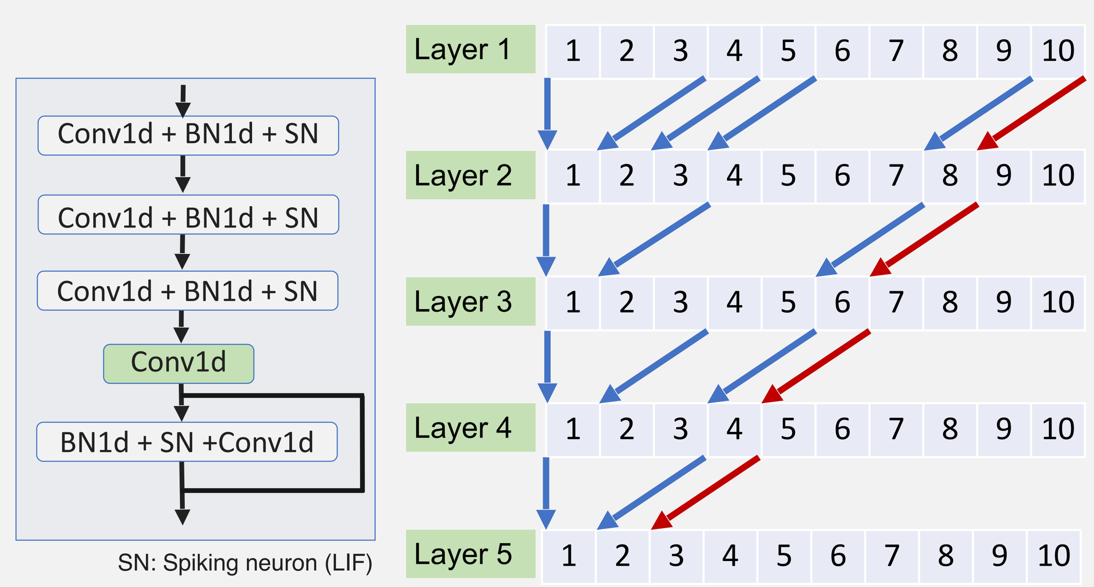
Structure



- Can process data in **real time** during inference.
- Can be trained in **parallel**.
- **Smaller** model size and FLOPs, **fewer** parameters.
- **SOTA performance** within the realm of SNN models on the Speech Commands dataset.

Spiking LMUFormer

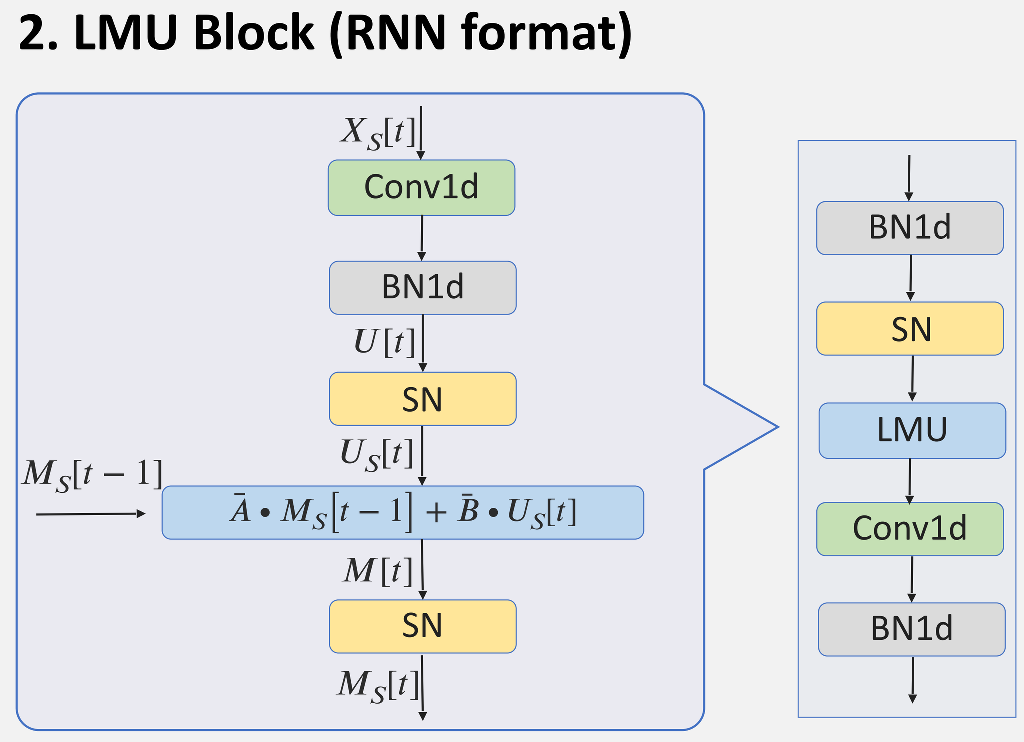
1. Convolutional Patch Embedding



- Apply Conv1d on **time** dimension: adds negligible delay, but enhances performance significantly.
- Latency analysis:

1. To get the 1st output of the 1st Conv layer we need to wait for **3** input samples.
2. To get the 1st output of the patch embedding we need to wait **8** extra input samples.
3. After **8** input samples, we can operate on the inputs sequentially.

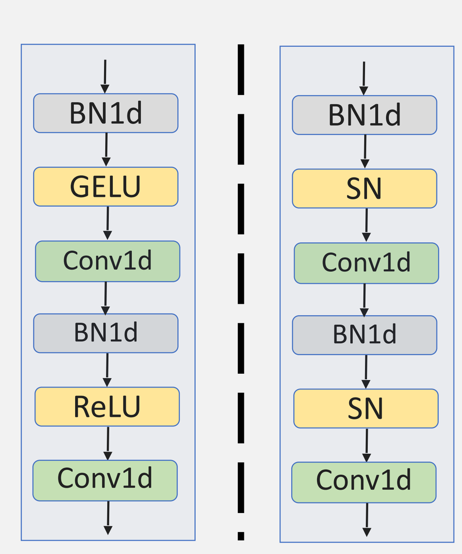
2. LMU Block (RNN format)



t : The time step t & sample index t
$X_S[t]$: Input spikes at time step t
$U[t]$: Input signal of the LMU memory cell
$U_S[t]$: Firing of the spikes of $U[t]$
$M[t]$: Memory vector at time step t
$M_S[t]$: Firing of the spikes of $M[t]$

3. Conv Channel Mixer

- Left: Non-Spiking
- Right: Spiking



Key Innovation

Merge SNN time step with the LMUFormer index, avoiding the need for an extra time dimension and enabling an efficient spiking architecture.

Experiments

Accuracy

Speech Commands V2 Dataset:

Model	Sequential Inference	Parallel Training	SNN	Accuracy (%)
RNN (Bittar & Garner, 2022)	Yes	No	No	92.09
Attention RNN (De Andrade et al., 2018)	No	No	No	93.9
liBRU (Bittar & Garner, 2022)	Yes	No	No	95.06
Res15 (Vygon & Mikhaylovskiy, 2021)	Yes	Yes	No	97.00
KWT2 (Berg et al., 2021)	No	Yes	No	97.74
AST (Gong et al., 2021)	No	Yes	No	98.11
LIF (Bittar & Garner, 2022)	Yes	Yes	Yes	83.03
SFA (Salaj et al., 2021)	Yes	No	Yes	91.21
Spikformer* (Zhou et al., 2022)	No	Yes	Yes	93.38
RadLIF (Bittar & Garner, 2022)	Yes	No	Yes	94.51
Spike-driven ViT* (Yao et al., 2023)	No	Yes	Yes	94.85
LMUFormer	Yes	Yes	No	**96.53**
LMUFormer (with states)	Yes	Yes	No	**96.92**
Spiking LMUFormer	Yes	Yes	Yes	**96.12**

Reduction in # of Params. :
$\mathbf{86.93 \div 1.62 \approx 53.66}$
Reduction in FLOPS:
$\mathbf{12.4 \div 0.189 \approx 65.61}$

Model	Params. (M)	OPs (G)
AST (Gong et al., 2021)	86.93	12.4
LMUFormer	1.62	0.189
Spiking LMUFormer	1.69	0.0309

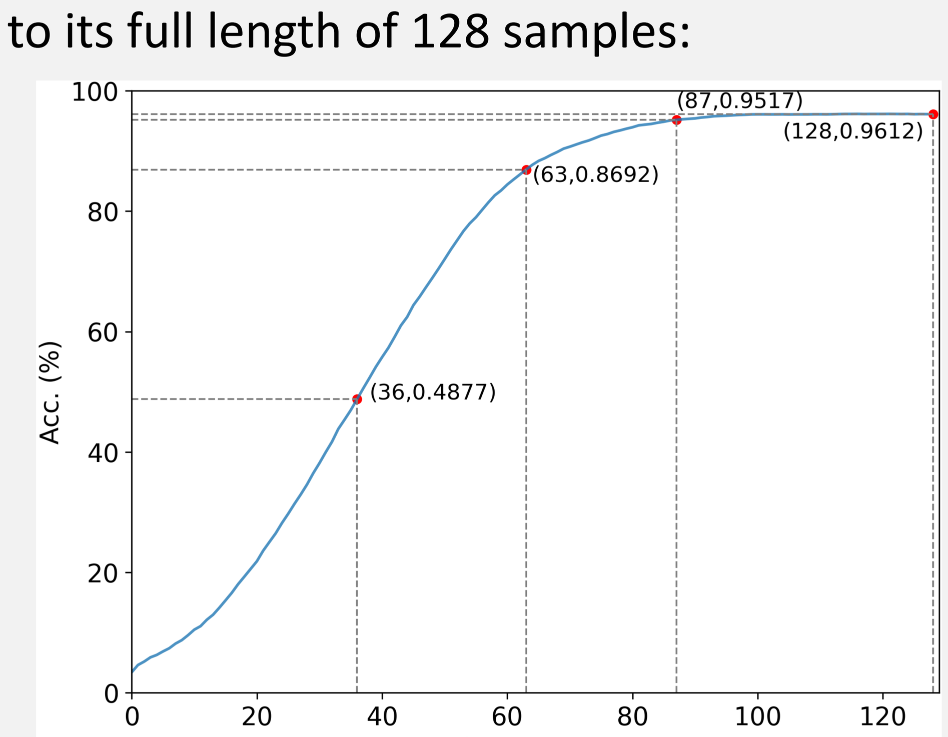
Our models achieve **SOTA accuracy** in SNN domain and achieve comparable performance to AST[4] with a significantly reduced **number of parameters** and lower **FLOPS**.

LRA (Long Range Arena) benchmark:

Model	ListOps (2K)	Text(4K)	Retrieval (4K)	Image(1K)	Pathfinder (1K)	Avg
S4	58.35	76.02	87.09	87.26	86.05	80.48
Linear Trans.	16.13	65.90	53.09	42.34	75.30	50.55
Linformer	35.70	53.94	52.27	38.56	**76.34**	51.36
Transformer	36.37	64.27	57.46	42.44	71.40	54.39
BigBird	36.05	64.02	59.29	40.83	74.87	55.01
Nystromformer	37.15	65.52	79.56	41.58	70.94	58.95
LMUFormer	34.43	**68.27**	78.65	54.16	69.9	61.08
Spiking LMUFormer	**37.30**	65.80	**79.76**	**55.65**	72.68	**62.24**

Energy-Efficiency

We evaluated the trained spiking LMUFormer on the Speech Command V2 test dataset, gradually increasing the sequence length from 0 to its full length of 128 samples:



Spiking LMUFormer achieves
99% (95.17% / 96.12%)
of its original performance,
while getting a
32.03% (1 - 87/128)
reduction in the sequence length!

References

Paper: Zeyu Liu, , Gourav Datta, Anni Li, Peter Anthony Beerel. "LMUFormer: Low Complexity Yet Powerful Spiking Model With Legendre Memory Units." *The Twelfth International Conference on Learning Representations*. 2024.

Code: https://github.com/zeyuliu1037/LMUFormer

[1] Aaron Voelker, Ivana Kajic, and Chris Eliasmith. Legendre memory units: Continuous time representation in recurrent neural networks. *Advances in neural information processing systems*, 32, 2019.

[2] Narsimha Reddy Chilkuri and Chris Eliasmith. Parallelizing legendre memory unit training. In *International Conference on Machine Learning*, pp. 1898–1907. PMLR, 2021.

[3] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

[4] Yuan Gong, et al. AST: Audio spectrogram transformer. arXiv preprint arXiv:2104.01778, 2021.