SPIKING MIXERS FOR ROBUST AND ENERGY-EFFICIENT VISION-AND-LANGUAGE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal learning is a fundamental challenge in artificial intelligence, with applications spanning computer vision, speech recognition, and natural language processing. This paper presents the pioneering incorporation of Spiking Neural Networks (SNNs) into the Vision-and-Language domain, introducing MLP-Mixer as a unified backbone and adapting mixture of experts approach to effectively fuse different modalities. The Mixer is directly trained using surrogate gradients and has small timesteps. We propose a SNN specific adversarial training technique, combined with the mixture of experts framework, leads to improvements in adversarial robustness. We hope these findings will shed light on future research in the field of Multimodal Spiking Neural Network and adversarial robustness of Multimodal Learning.

1 INTRODUCTION

Recently, MixerTolstikhin et al. (2021) based models has witnessed rapid development in computer vision(CV)Trockman & Kolter (2022), time seriesChen et al. (2023) and natural language processing(NLP)Fusco et al. (2023) domain as a basic backbone.The broad applicability of mixers mainly comes from the lack of inductive biasTouvron et al. (2021) and Transformer-like global information extraction capability. Although the mixers is more computation efficient than Transformers because there's only MLPs instead of self-attention with $O(N^2d)$ computation complexity in mixers, the computation cost is still very high.

The Spiking Neural Networks(SNN)Maass (1997) utilize discrete and event-based information processing neurons embedded into modern Neural Network as underlying computation mechanism to save computation costs.SNN has been demonstrated effective in CNN backbonesFeng et al. (2023) and TransformersZhou et al. (2022) Li et al. (2022b) with strong prior, but the role of spiking neuron for backbones with less prior information still remains largely unexplored.

In this work, we propose Spiking Mixer model with dedicate encoding method and computation blocks for both vision tasks and language processing tasks.For CV tasks, we view images as high dimensional words with short sentence length while for language processing, we use the inherent order of words as temporal input.To effectively capture different input we use MLP to encode images into patches as Dosovitskiy et al. (2021) and for NLP tasks Fusco et al. (2023)empirical demonstrate the effectiveness of min-Hash in encoding sub-word inputs after standard tokenizers for mixers. Inspired by the recent RWKV modelPeng et al. (2023) we propose to use bio-plausible multi-compartmental neuron model to further improve the accuracy. The feedback loops inside multicompartment model combined with timeshift enhance the temporal connection between consecutive time-steps like in RWKV Peng et al. (2023). We use generalized LIF model with learnable threshold directly trained with surrogate gradient based methods.We find without strong inductive bias, SNN can consistently improve the accuracy, computation costs and robustness of original model. To summarize, our contributions in this paper are as follows:

- A Spiking Mixer model is proposed for both vision and NLP tasks and shows improved accuracy, robustness and computation costs.
- A generalized LIF model and RWKV-like is proposed to improve the model accuracy and save computation cost.

2 RELATED WORKS

Our model is the temporal extension of Mixers, For the development of Mixers, Google in Tolstikhin et al. (2021) probed into the necessity of using computation-extensive attention mechanism and use the 'Patch Mixing' and 'Channel Mixing' Operator for vision task. Later they use similar blocks in Chen et al. (2023) for time-series predictions. a-MLP in Liu et al. (2021) use attention for spatial mixer and achieve good performance in vision and language.Rajagopal & Nirmala (2021); Touvron et al. (2021); Trockman & Kolter (2022) use different variant of spatial mixer and can achieve better performance on smaller datasets.Li et al. (2023b) use factorized TS-Mixer for multivariate timeseries forecasting. There're also some attempts to design Mixers targeting NLP tasks, for instance, P-NLP Fusco et al. (2023) use minhash and MLP-Mixer for lightweight text classification.Hyper-Mixer Mai et al. (2023) use dynamic-generated token mixing. TCA-Mixer Liu et al. (2023) use a variant of attention with Mixer for NLP tasks.

2.0.1 SNN BEYOND VISION TASKS OR CNN BACKBONE.

For the SNN parts, researchers try to extend spiking neural network to beyond Convolutional Neural Network backbone or beyond vision tasks.Some very recent worksZhou et al. (2022) Zhou et al. (2023) use LIF neurons and delete the SoftMax for SNN-compatible Transformer.Che et al. (2023) propose to use neural architecture search(NAS) for efficient spiking transformer. Li et al. (2022a) views the input image patches as temporal input and use horizontal and vertical LIF groups for different division of patches, but their implementation is in a SNN-DNN hybrid fashion and is not fully compatible with neuromorphic chips.Li et al. (2023a) is another spiking mixer model but their dedicate model is only optimized for vision tasks and the effective extension method for NLP tasks remains unknown. Yao et al. (2021)exploit the spatial-temporal attention mechanism inside specific neuron design methods and Cai et al. (2023) consider temporal attention of SNN. Lv et al. (2023) use conversion-based method and trained a convolutional SNN for NLP tasks and shows potential of spike tokenizer and robustness of SNN in NLP tasks.Zhu et al. (2023) is the first large scale directly trained SNN model for NLP tasks, they use token-shift and binary embedding as well as modified RWKV operation for effectively training large models. However, their model is not fully-SNN compatible with the computation of RWKV in real-value and cannot be directly applied to modern neuromorphic chips.

In this work, we successfully use the gernalized LIF as a building block and extend spiking neural network to the Mixer backbone. Our model is directly trained using surrogate gradients and can be applied ti multiple different tasks.We empirical prove the efficiency and potential of this architecture to beyond vision tasks.

3 THE PROPOSED SPIKING MIXER FRAMEWORK

The proposed Spiking Mixer model is composed of temporal-token mixing block and channel mixing block. In standard RWKV based Transformer models, the standard attention is firstly replaced by the temporal shift of key and value. Linear attention is then applied with temporal-specific weight across timestep and a receptance field is applied. Inspired by RWKV, we view the max sentence of length N or the patched image sequence as a temporal-feature input with size T * F, where T is the Temporal dimension size and F is the spatial feature size. For the case of NLP tasks, T is equal to the sequence length N and F is set to 1. While for the case of images, for static dataset like CIFAR-10, the input is repeated for T times to simulate temporal data input and for neuromorphic dataset, the temporal dimension correspond to sampled frames. In this case T is identical to the temporal dimension of the images and the feature dimension is correspond to the actual feature size. We empirically found the direct mixing of vision datasets in temporal-token dimension will degrade the performance because image data is sensitive to the inverse of temporal dimension, so in practice we apply temporal-wise fully connected layer instead of standard Linear layer.

The generalized LIF model is formulated as:

$$\mathbf{c_m}\frac{d\mathbf{v}}{dt} = \mathbf{g_m}\cdot\mathbf{v} + \mathbf{v^T}\mathbf{M}_{m\times m}\mathbf{v} + \mathbf{I}$$
$$o = h(\sum_{w,v\in Neuron_i} \mathbf{w}\cdot\mathbf{v} - v_{th})$$

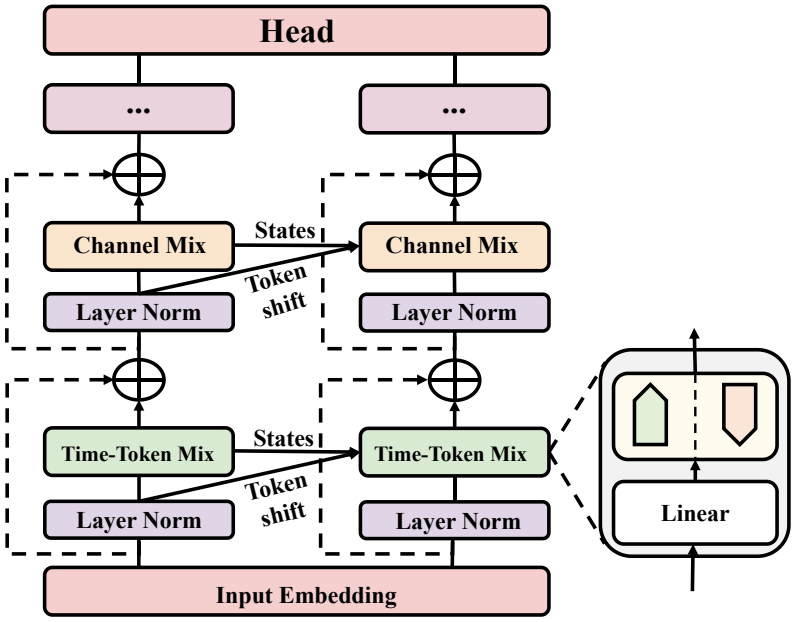

Figure 1: The high-level schematic of Spiking-Mixer model, which contains Time-feature mixing block and channel mixing block. For compatibility with neuromorpjic chips, a multicompartment LIF model is attached after each linear layer.Inspired by RWKV, we add a token shift operator for mixing channel information in nearby timesteps.

where $\mathbf{c_m}$ is the conductance vector, $\mathbf{v}$ is the membrane potential, the $\mathbf{M}$ is the inter-compartment term representing the relationship between different compartments.$\mathbf{M}$ can be divide into intra-neuron parts with the shape of block diagonal and inter-neuron parts.$\mathbf{I}$ is the input vector to neuron. $\mathbf{w}$ is the importance for different compartment within the same neuron, h is the spike function. For traditional LIF based SNN,the $\mathbf{M}$ and $\mathbf{w}$ term vanished, and the input is $\sum w_{ij} z_i$ where z_i is the output from the prvious layer.

For the most generalized form of using m neuron as a group, each neuron has n compartment, and there's connection between inter-neuron compartments and intra-neuron compartments, so there's at most m spikes per timestep, the problem is difficult.In this paper we consider the case of per neuron group has only one neuron, and each neuron has two compartments, the input of two compartments are formulated as

$$I_1 = \sum w_{ij} z_i$$

$$I_2 = \sum -\beta O_i$$

where O_i is output from the first compartment.

For the weighted addition of different timesteps, we use generalized LIF shares a similar idea as in Zhang et al. (2023) . Because in standard RWKV, the weight in different timesteps are encoded like position embedding, so for the case without receptance field, we can approximate such multiple loops with the linear combination of the weight of single-loop feedback generalized neuron, we empirically find such implementation is more efficient compared to use one feed back layer. [tb]Pseudo-code for the Spiking Mixer[language=python] def $spiking_m ixer_b lock(x)$: $shortcut = xx = norm(x)x = tokenshift(x)x = proj(x)x = multi_c omp(x)returnx + shortcutdefmulti_c omp(x) : initu1, u2, tau1, tau2, vth, TfortinT : ift > 0 : u2 = tau1 * u2 + o(t-1)u1 = tau2 * (u1 + u2) * (1 - s(u1 + u2 - vth)) + x(t)o = s(u1 - vth)$ For the temporal shift to provide a mixing of channel information in nearby timesteps, we apply standard padding in *Pytorch* Library as *nn.ZeroPad2d((0,0,1,-1))*.

Benefit from the mechanism of Spiking Neural Network, the computation of mixing can be parallelized and can benefit from the RNN-like sequential readout. The Spiking Mixer is trained directly

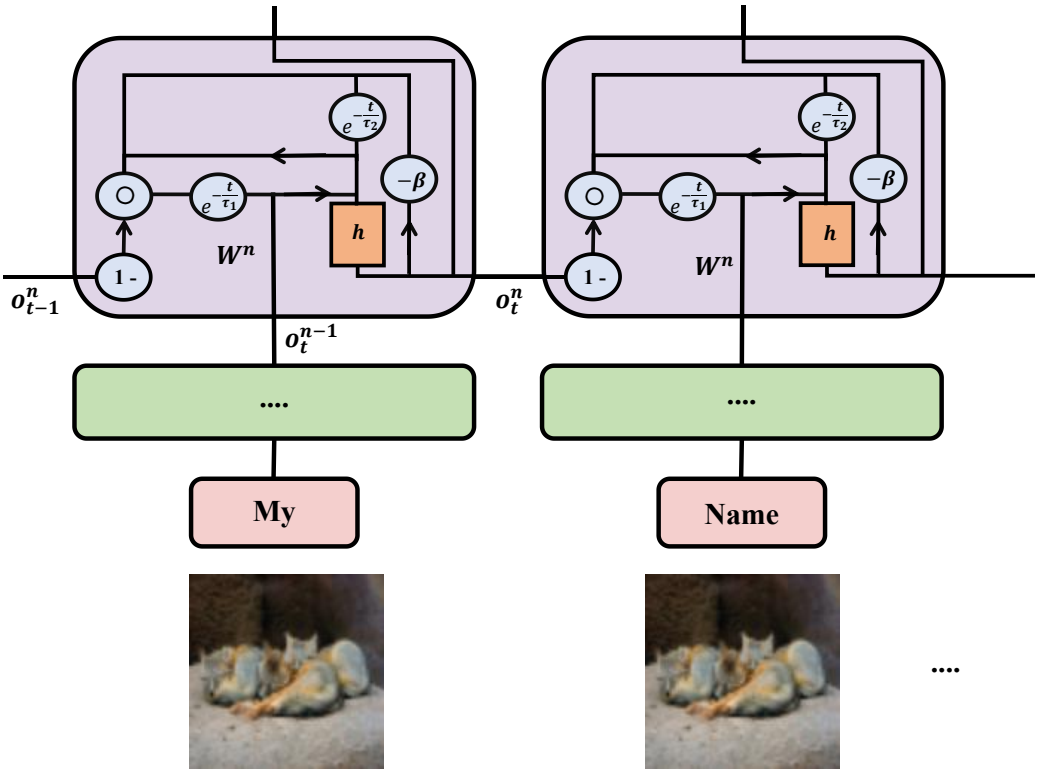

Figure 2: The demonstration of RWKV-like parallel training and serial read out during inference

based on straight-through estimator with learnable threshold and can be formulated as :

$$\frac{\partial o}{\partial u} \approx h(u) = \frac{1}{a} sign(|u - V_{th}| < \frac{a}{2})$$

4 EXPERIMENTS

We conduct experiments both for image classification tasks and natural language processing tasks.

4.1 EVALUATION SETUP

4.2 IMAGE CLASSIFICATION TASKS

For Image classification tasks, we consider both the static datasets and neuromorphic datasets. For static datasets, we evaluate the acurracy result of CIFAR-10, CIFAR100 and ImageNet-64. Static input data is repeated identical to the number of training TimeSteps and the final output is averaged to get the final result. For neuromorphic dataset, we consider the DVS-CIFAR10Li et al. (2017) dataset. DVS-CIFAR10, is another neuromorphic dataset and is very easy to overfit, and is a more challenging dataset. The ImageNet-64, For these tasks, we use the model architecture similar to the base-Mixer model with hidden dimension of 768 and pretrain on ImageNet-1k dataset. The TimeSteps of these tasks is set to 4 and for dvs-cifar10 the TimeStep is set to 10.

4.3 TEXT CLASSIFICATION TASKS

For text tasks, we use classification tasks in standard GLUEWang et al. (2018) and three more challenging tasks, the imdbMaas et al. (2011) dataset, the sst-5 datasetSocher et al. (2013) for fine-grained sentimental classification with 5 different classes and MTOP dataset.The MTOP datasetLi

Model	Param(M)	Cifar-10	IMN-64	DVS-Cifar-10	TimeStep
Mixer	10.2	Yellow	11.2	Black	4
Mixer(128)	0.75	**74.2**	**87.3**	**96.5**	4
Mixer(512)	4.18	**74.2**	**52.24**	**96.5**	**4**
Mixer(768)	8.44	**74.2**	**52.01**	**96.5**	**4**
Spiking Mixer(128)	0.75	**79.85**	**42.33**	**96.5**	**4**
Spiking Mixer(512)	**4.18**	**86.59**	**64.54**	**96.5**	**4**
Spiking Mixer(768)	8.44	**87.66**	**67.00**	**96.5**	**4**

Table 1: Classification accuracy results on six different datasets.

Model	SST-2	SST-5	MTOP	IMDB	Subj	MR	Avg
SpikeGPT(45M)[1]	80.39	37.69	-	-	69.23	88.45	68.94
SpikeGPT(216M)[1]	82.45	38.91	-	-	68.11	89.10	69.64
S-TextCNN-direct[2]	75.73	23.08	-	-	51.55	53.30	50.91
S-TextCNN-Finetune+convert[2]	80.91	41.63	-	-	90.6	75.45	72.14
pNLP-Mixer[3]	72.88	39.31	79.0	88.1	97.3	93.4	89.58
Spiking Mixer	75.0	39.4	82.6	90.9	97.4	93.9	90.85
Spiking Mixer(+TimeShift)	76.45	39.63	83.4	92.2	97.7	93.9	91.1
Spiking Mixer(+MultiComp)	77.29	39.63	83.6	91.9	72.03	88.86	91.45
Spiking Mixer(+rep)	**75.11**	**39.36**	**80.3**	**90.1**	**97.6**	**92.5**	89.6

[1] from Zhu et al. (2023)

[2] from Lv et al. (2023)

[3] from Fusco et al. (2023)

Table 2: Classification accuracy results on text classification datasets.

et al. (2020) contains six languages,English, Spanish, French, German, Hindi, and Thai, all translated from English. The MTOP is a token classification task and the accuracy is the number of instance with all tokens being classified to the correct label in all instances. All tasks except the MTOP is trained using Mixer with hidden dimension of 256 and for MTOP we use the model with hidden dimension of 512. The Timestep of these models is identical to the input sequence length.

4.4 Implementation Details

For the image classification tasks. We train from scratch using ImageNet-64Chrabaszcz et al. (2017) and finetune for small scale datasets. The ImageNet-64 is a down sampled version(64×64) of ImageNet-1k with the same image numbers as ImageNet-1k, it's more challenging and more noisy. It contains 1.28 million training data and 50,000 testing data. For the training of ImageNet-64, we follow the training recipe of DeiT and use a batch size of 96. The learning rate is set to 5e-3 with weight decay of 5e-2, momentum of 0.9. We train for 300 epochs with learning rate warm-up for 5 epochs, after warm-up we use cosine learning rate scheduler.Standard data augmentation like random crop, random erase, horizontal flipping, mixup, label-smoothing is used.

During the finetuning of small sized datasets, for the static datasets, standard preprocessing like random crop and flip, normalize is used and we resize the dataset to $64\times$ 64.For neuromorphic datasets, the event stream is integrate to frames with 10ms resolution and downsampled to $64\times$ 64.

For the text classification, we modified the model structure by using a bert tokenizer, a min hash based mapping block as well as a bloomcounter to extract word representation like inFusco et al. (2023), the hash length of min-hash is set to 64, the depth of Mixer is set to 2, the sentence is padded to the length of 1024.

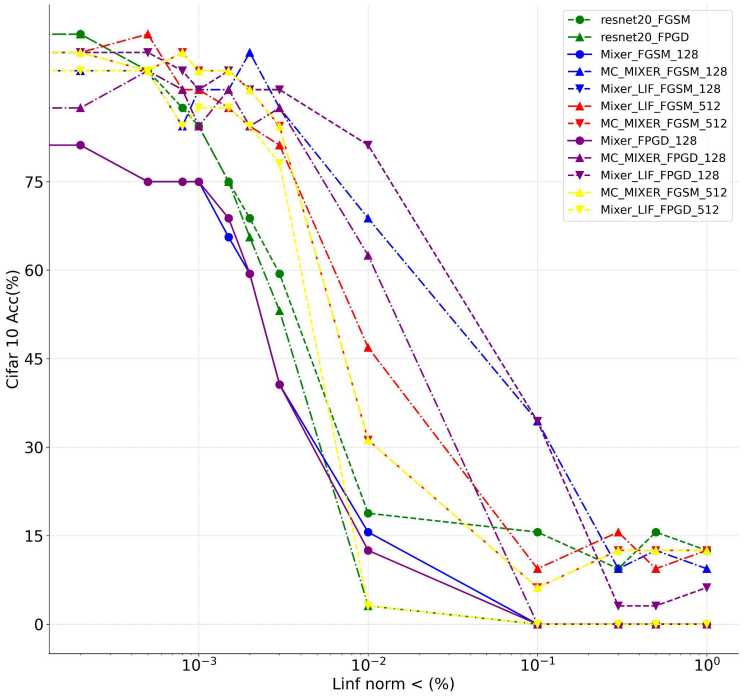

Figure 3: The adversial robustness of Spiking Mixer

4.5 Experiment Result

From table 1, we can find that the Mixer based model performs consistently better than CNN based methods. And counter intuitively, the spiking version of MLP-Mixers will performs better than their ANN counterparts, as pNLP-Mixer and Spiking Mixer. That's partly because Spiking Mixer inherently capture much contextual information with membrane potential as a special gate. By applying TimeShift as in RWKV model or applying multi-compartment neurons, the spiking mixer model will further improve its performance. Another very important finding is when changeing the input representation from the original representation of simply repeating the input to view the sentence to temporal sequence, the performance of the model will not degrade very much but significantly improve the inference cost and enable token-level serial readout to further boost the performance of inference.

4.6 Ablation Study

We compare the effect of using different forms of neurons compared to using the generalized LIF and directly train on small scale Image classification tasks. As can e seen from the chart below, the multi compartment setting greatly improve the expressiveness compared to the simple LIF based

models.We study the reason for such improved expressiveness by visualizing the feature map in some of the layers in the total Mixer, as shown in the graph.

4.7 ANALYSIS

5 CONCLUSION

We demonstrate for the first time of using Spiking Neural Network based Mixer model for both vision and NLP tasks. Compare to existing works in Clancey (2021) for NLP tasks, our model is directly trained and is much lighter with comparable results. Our model is also pure SNN implementation compared to the Clancey (2021), and our model is much lighter with similar performance. As for Clancey (2021), our model has similar performance with their model but our model can be further extend to NLP tasks. For the vision model, using generalized LIF SNN, our model can successfully scale to large model size, and for NLP tasks, our model can be directly trained end-to-end without severe accuracy degradation with the help of RWKV like temporal-sentense encoding, time-shift and feedback. We also demonstrate the efficiency of sentence generation of our Spiking Mixer, which is particular important for inference. We hope this work will spark future research on SNN based NLP or the research of other domains.

computation cost between traditional LIF and proposed Lif

and computation cost between original representation and chnaged representation

The adverserial robustness of spiking neural networks

the spike based multimodal attack methodology

the regularization based $\mathrm{l}_i nfrobusttrainingmethod$

the actual computation benefits of snn

REFERENCES

Wuque Cai, Hongze Sun, Rui Liu, Yan Cui, Jun Wang, Yang Xia, Dezhong Yao, and Daqing Guo. A spatial-channel-temporal-fused attention for spiking neural networks, 2023.

Kaiwei Che, Zhaokun Zhou, Zhengyu Ma, Wei Fang, Yanqi Chen, Shuaijie Shen, Li Yuan, and Yonghong Tian. Auto-spikformer: Spikformer architecture search, 2023.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecasting, 2023.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the CIFAR datasets. *CoRR*, abs/1707.08819, 2017. URL `http://arxiv.org/abs/1707.08819`.

William J. Clancey. The Engineering of Qualitative Models. Forthcoming, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Lang Feng, Qianhui Liu, Huajin Tang, De Ma, and Gang Pan. Multi-level firing with spiking ds-resnet: Enabling better and deeper directly-trained spiking neural networks, 2023.

Francesco Fusco, Damian Pascual, Peter Staar, and Diego Antognini. pnlp-mixer: an efficient all-mlp architecture for language, 2023.

Boyan Li, Luziwei Leng, Ran Cheng, Shuaijie Shen, Kaixuan Zhang, Jianguo Zhang, and Jianxing Liao. Efficient deep spiking multi-layer perceptrons with multiplication-free inference, 2023a.

Haoran Li, Abhinav Arora, Shuohui Chen, Anchit Gupta, Sonal Gupta, and Yashar Mehdad. MTOP: A comprehensive multilingual task-oriented semantic parsing benchmark. *CoRR*, abs/2008.09335, 2020. URL https://arxiv.org/abs/2008.09335.

Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: An event-stream dataset for object classification. *Frontiers in Neuroscience*, 11, 2017. ISSN 1662-453X. doi: 10.3389/fnins.2017.00309. URL https://www.frontiersin.org/articles/10.3389/fnins.2017.00309.

Wenshuo Li, Hanting Chen, Jianyuan Guo, Ziyang Zhang, and Yunhe Wang. Brain-inspired multilayer perceptron with spiking neurons, 2022a.

Yudong Li, Yunlin Lei, and Xu Yang. Spikeformer: A novel architecture for training high-performance low-latency spiking neural network, 2022b.

Zhe Li, Zhongwen Rao, Lujia Pan, and Zenglin Xu. Mts-mixers: Multivariate time series forecasting via factorized temporal and channel mixing, 2023b.

Hanxiao Liu, Zihang Dai, David R. So, and Quoc V. Le. Pay attention to mlps, 2021.

Xiaoyan Liu, Huanling Tang, Jie Zhao, Quansheng Dou, and Mingyu Lu. Tcamixer: A lightweight mixer based on a novel triple concepts attention mechanism for nlp. *Engineering Applications of Artificial Intelligence*, 123:106471, 2023. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2023.106471. URL https://www.sciencedirect.com/science/article/pii/S0952197623006553.

Changze Lv, Jianhan Xu, and Xiaoqing Zheng. Spiking convolutional neural networks for text classification. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=pgU3k7QXuz0.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1015.

Wolfgang Maass. Networks of spiking neurons: The third generation of neural network models. *Neural Networks*, 10(9):1659–1671, 1997. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(97)00011-7. URL https://www.sciencedirect.com/science/article/pii/S0893608097000117.

Florian Mai, Arnaud Pannatier, Fabio Fehr, Haolin Chen, Francois Marelli, Francois Fleuret, and James Henderson. Hypermixer: An mlp-based low cost alternative to transformers, 2023.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Xiangru Tang, Bolun Wang, Johan S. Wind, Stansilaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Jian Zhu, and Rui-Jie Zhu. Rwkv: Reinventing rnns for the transformer era, 2023.

A. Rajagopal and V. Nirmala. Convolutional gated mlp: Combining convolutions gmlp, 2021.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision, 2021.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training, 2021.

Asher Trockman and J. Zico Kolter. Patches are all you need?, 2022.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461, 2018. URL `http://arxiv.org/abs/1804.07461`.

Man Yao, Huanhuan Gao, Guangshe Zhao, Dingheng Wang, Yihan Lin, Zhaoxu Yang, and Guoqi Li. Temporal-wise attention spiking neural networks for event streams classification, 2021.

Shimin Zhang, Qu Yang, Chenxiang Ma, Jibin Wu, Haizhou Li, and Kay Chen Tan. Long short-term memory with two-compartment spiking neuron, 2023.

Chenlin Zhou, Han Zhang, Zhaokun Zhou, Liutao Yu, Zhengyu Ma, Huihui Zhou, Xiaopeng Fan, and Yonghong Tian. Enhancing the performance of transformer-based spiking neural networks by snn-optimized downsampling with precise gradient backpropagation, 2023.

Zhaokun Zhou, Yuesheng Zhu, Chao He, Yaowei Wang, Shuicheng Yan, Yonghong Tian, and Li Yuan. Spikformer: When spiking neural network meets transformer, 2022.

Rui-Jie Zhu, Qihang Zhao, Guoqi Li, and Jason K. Eshraghian. Spikegpt: Generative pre-trained language model with spiking neural networks, 2023.

A Appendix

derivation of θ-related norm for adversarial robustness of SNN:

$$m^l(t) = \lambda(m^l(t^- - s^l(t)r^l(t)$$

when $s^l(t) = 1r^l(t) = m^l(t); elser^l(t) = 0$

$$r^l(t)s^l(t) \geq \theta s^l(t)$$

the equal holds when doing the soft reset

$$m^l(t) - \lambda m^l(t-1) = \lambda(W^l s^{l-1}(t) - s^l(t)r^l(t)$$

$$\lambda m^l(t) - \lambda^2 m^l(t-2) = \lambda^2(W^l s^{l-1}(t) - s^l(t-1)r^l(t-1))$$

$$m^l(T) = W^l \sum_{i=1}^{T} \lambda^{T+1-i} s^{l-1}(i) - \sum_{i=1}^{T} \lambda^{T+1-i} s^l(i)r^l(i)$$

$$h(t) := \sum_{i=1}^{T} \lambda^{t+1-i} s(i)$$

$$s^l(t) = \frac{h^l(t) - h^l(t-1)}{\lambda}$$

$$\eta^l = \sup_{s\neq 0, s\in\chi^{N-1}} \|W^l s\|_\infty \geq 0$$

$$\mu^l = -\sup_{s\neq 0, s\in\chi^{N-1}} \| - W^l s\|_\infty \leq 0$$

$$\mu^l t \leq W^l h^{l-1}(t) - \sum_{i=1}^{t} \lambda^{t+1-i} s^l(i)r^l(i) \leq W^l h^{l-1}(t) -^l (t) \leq \eta^l(t)$$

$$\frac{W^l h^{l-1}(t) - \eta^l t}{\theta} \leq h^l(t) \leq \frac{W^l h^{l-1}(t) - \mu^l t}{\theta}$$

$$s^l(t) - \widetilde{s}^l(t) = \frac{1}{\lambda}[h^l(t) - h^l(t-1) - \widetilde{x}^l(t) + \widetilde{x}^l(t-1)] \leq 2t\frac{\eta^l - \mu^l}{\lambda\theta} + \frac{1}{\theta}W^l(s^{l-1}(t) - \widetilde{s}^{l-1}(t)$$

$$s^l(t) - \widetilde{s}^l(t) \leq 1$$

$$\frac{1}{\theta} W^l(s^{l-1}(t) - \widetilde{s}^{l-1}(t)) \leq \frac{\eta^l - \mu^l}{\theta}$$

$$s^l(t) - \widetilde{s}^l(t) + \frac{1}{\theta} W^l(s^{l-1}(t) - \widetilde{s}^{l-1}(t)) \leq 1 + \frac{\eta^l - \mu^l}{\theta}$$

$$|s^l(t) - \widetilde{s}^l(t)|^2 - \frac{1}{\theta^2} |s^{l-1}(t) - \widetilde{s}^{l-1}(t)|^2 \leq \frac{2t}{\lambda} [\frac{\eta^l - \mu^l}{\theta} + (\frac{\eta^l - \mu^l}{\theta})^2]$$