A Study of Optimization Methods for TTFS based Spiking Neural Networks

Supervisor:
Prof. Dr. Gitta Kutyniok

Advisor:
Manjot Singh

Contact:
singh@math.lmu.de

Type:
BA/MA

Research area:
Optimization in ML

Programming language:
Python

Required skills:
Programming, Theory

Topic Background

In recent years, optimization methods for artificial neural networks (ANNs) have seen substantial progress, with adaptive and preconditioned optimizers such as Adam, AdamW, Shampoo, and Muon significantly improving training stability and performance. At the same time, spiking neural networks (SNNs), and in particular, time-to-first-spike (TTFS) models, have emerged as promising architectures for temporal and event-based data. TTFS models introduce additional parameters such as trainable delays and thresholds which makes the training difficult. However, most TTFS based SNNs are still trained using optimizers originally designed for ANNs, without accounting for additional complexities coming from the SNN intrinsic structure.

This raises the question of whether TTFS model require specialized optimization strategies and and how ideas from modern ANN optimizers can be adapted to this setting.

Description of the Project

The project focuses on analyzing existing optimizers, identifying their limitations in the TTFS setting, and implementing modified or new optimization strategies that better capture the structure of TTFS models. The final expected outcomes are:

1. a systematic comparison of standard ANN optimizers (e.g., SGD, Adam, AdamW, Muon, Shampoo, etc.) when applied to TTFS based SNN model

2. insights into adapting step sizes using gradient information from both weights and temporal delay parameters

3. an empirical analysis of inductive biases introduced by different optimization strategies

Tasks

The specific steps of the project are as follows:

1. conduct a literature review on optimization methods for ANNs and SNNs, with emphasis on adaptive and preconditioned optimizers

2. implement baseline TTFS SNN model and train them using standard optimizers such as SGD, Adam, and AdamW, etc.
3. identify limitations of existing optimizers in the TTFS setting, such as slow delay updates and/or unstable convergence
4. design and implement simple TTFS-specific optimization modifications, such as separate learning rates or adaptive scaling for delay parameters
5. evaluate and compare the proposed optimization strategies on real and synthetic temporal tasks and event-based datasets
6. depending on time and project progress, additional tasks may include theoretical investigations such as the role of initialization strategies and/or convergence of the optimization algorithm
7. depending on the student's background and interests, the project can be oriented more towards theory, with less emphasis on the experimental component

Additional note The project combines theoretical understanding and empirical analysis providing the student with hands-on experience in modern optimization methods and spiking neural network models. A working implementation is already available and will serve as the starting point.

References

1. Singh, M., Fono, A., and Kutyniok, G. (2024). *Expressivity of Spiking Neural Networks*. arXiv:2308.08218.
2. Neuman, A. M., Dold, D., and Petersen, P. C. (2025). *Stable Learning Using Spiking Neural Networks Equipped With Affine Encoders and Decoders*. arXiv:2404.04549.
3. Dold, D. and Petersen, P. C. (2025). *Causal pieces: analysing and improving spiking neural networks piece by piece*. arXiv:2504.14015.