BACKPROPAGATING IN TIME-DISCRETIZED SPIKING NEURAL NETWORKS

Max Guichardm.guichard@student.tudelft.nl
Supervised by: Nergis Tömen, Aurora Micheli, Olaf Booij

01. Introduction

- **Spiking neural networks** are used in neuroscience research to understand the interactions of neurons. They also represent a new machine learning paradigm.
- In a network we focus on the **spike times** of neurons. Solving for this in complex neuron models is not analytically possible.
- **Numerical integrations** are used to approximate the neurons. Commonly, the simple forward-Euler method is used in training [1].
- We explore the use of the **Parker-Sochacki** [2] integration method and compare it to an analytical baseline from the **BATS** paper [3]. This has a current-based LIF model and event-based system, which we discretize.
- Many more integration methods exists, such as implicit integrations, or the Runge-Kutta family of integrators. We chose the **Parker-Sochacki method** as it is a generalization of forward-Euler and its **order** is easily adjustable.

- Previous work finds increased simulation accuracy with higher order integrations [4]. This trend also holds for a simple function regression task [5].
- We contribute a method for effectively discretizing event-based systems using **spike-time interpolation** and how to properly apply backpropagation. Finally, we explore the effects of different integration parameters.

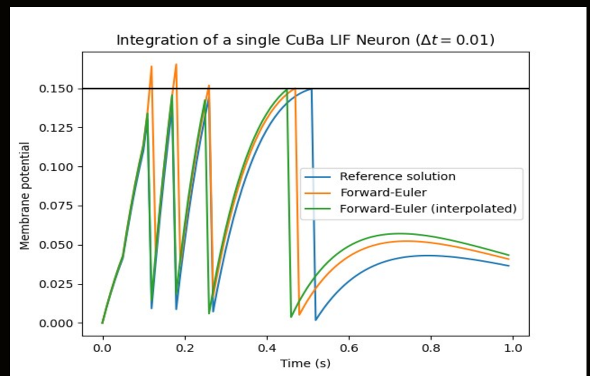

Figure 1: Membrane potential of CuBa LIF neuron. Blue is an analytical reference solution. The forward-Euler method is applied with and without spike time interpolation. With interpolation, the potential never goes above the threshold.

How are the training accuracy and training speed (in epochs and time) of a spiking neural network affected when numerically integrating with the forward-Euler and Parker-Sochacki methods?

02. Methodology

- Two experiments were conducted: assess effects of spike time interpolation, train numerically integrated networks directly.
- For experiment 1, train the analytical network and then transfer weights to numerically integrated networks.
- Two set of backpropagation methods are derived: piggybacking (naive), and adjusted for integration procedure.

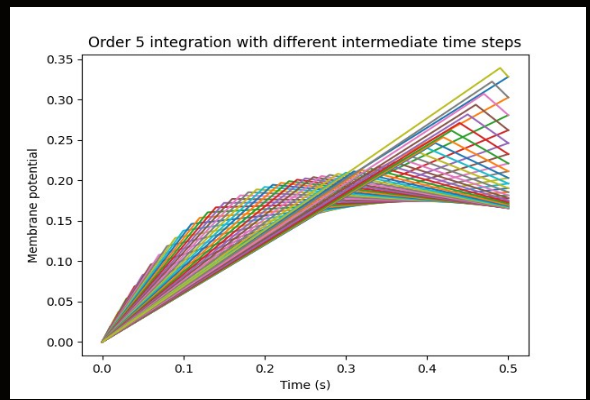

Figure 2: Order 5 integration with Parker-Sochacki for a single neuron. No inputs or outputs are produced so the integration should have the same result not matter the way it is done. This done not hold when we change the intermediate integration point.

03. Results

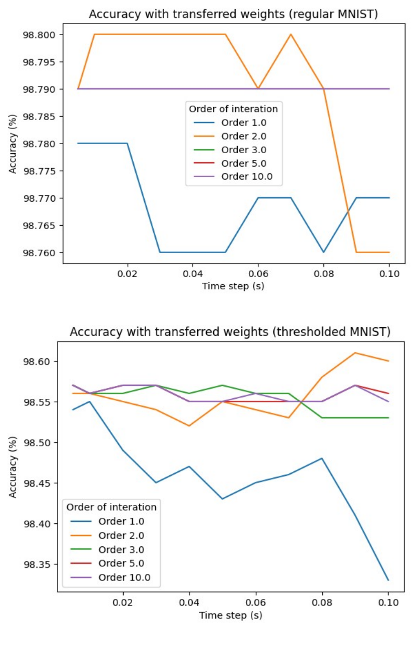

Figure 3: Output accuracy of numerically integrated networks at different orders and time deltas. The top graph is for egular MNIST. The bottom graph has a thresholded MNIST supposed to suppress the effects of spike time interpolation.

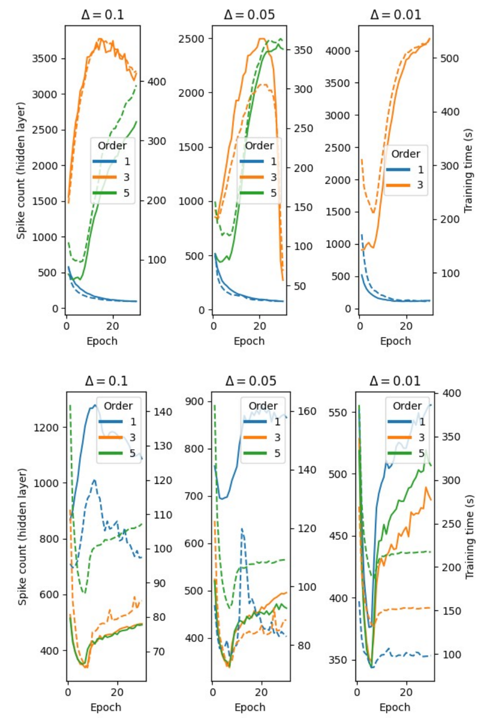

Figure 4: Hidden layer spike count (solid line) and real training time (dashed line) of different orders and time deltas. Top graph uses naive backpropagation. Bottom one applies the adjusted method.

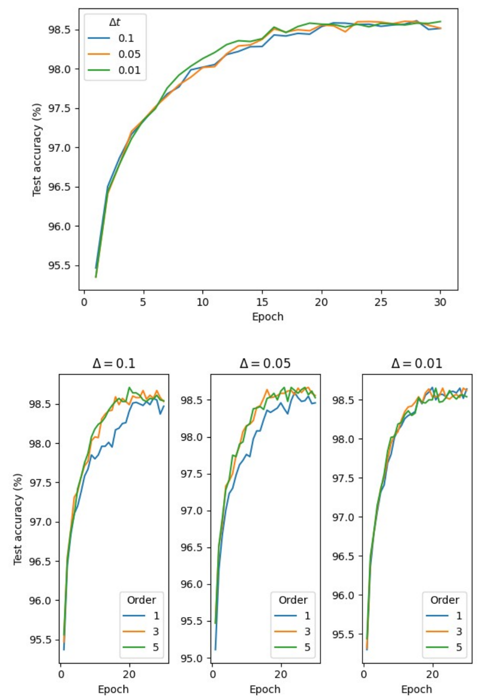

Figure 5: Test accuracies of models trained with the adjusted backpropagation. The top graph shows the average accuracy per time step. The bottom shows accuracies per order per timestep.

04. Discussion

- Spike time interpolation helps **stabilize** networks with high activity.
- **High order** integrations work robustly over **large time delta** choices. They find **sparser** solutions, **converge faster** and have **higher accuracies** in general.
- First order (forward-Euler) integrations benefit from **lower training times**.
- **Piggybacking** backpropagation has **un-favourable** training times, despite similar accuracy results. This is due to noise introduced by the integration process.

Limitations

- Low experiment numbers used to derive conclusions could generalize poorly.
- MNIST is a trivial dataset and may not give a full picture for more complex tasks/ networks.
- We combine spike time interpolation and numerical integration, an ablation study could be beneficial.

05. Conclusion

- We applied integration methods in discretizing a spiking neural networks and highlight pitfalls in training them through backpropagation.

We found:

- Numerically integrated networks perform similarly to their analytical counterpart.
- The choice of integration parameters affects the convergence rate, accuracy and training time of the network. High orders tend to perform more favourably.

Future work

- Explore more neuron models and deeper training tasks.
- Apply other integrators: adjoint sensitivity analysis can provide a more general approach.

References

[1] Jason K Eshraghian, Max Ward, Emre O Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. Proceedings of the IEEE, 2023.
[2] Robert D Stewart and Wyeth Bair. Spiking neural network simulation: numerical integration with the parker-sochacki method. Journal of Computational Neuroscience, 27:115–133, 2009.
[3] Florian Bacho and Dominique Chu. Exploring Trade-Offs in Spiking Neural Networks. Neural Computation, 35(10):1627–1656, 09 2023.
[4] Giuseppe de Alteriis and Calogero Maria Oddo. Trade-off between accuracy and computational cost of euler and runge kutta ode solvers for the izhikevich spiking neuron model. In 2021 10th International IEEE/EMBS Conference on Neural Engineering (NER), pages 730–733, 2021.
[5]Mario De Florio, Adar Kahana, and George Em Karniadakis. Analysis of biologically plausible neuron models for regression with spiking neural networks. arXiv preprint arXiv:2401.00369, 2023.