https://github.com/LubovCh/bats.git
https://github.com/LubovCh/slayerPytorch.git

Effects of time discretization on unconstrained Spiking Neural Networks

AUTHORS

Lubov Chalakova
chalakova.lubov@gmail.com

SUPERVISORS

Nergis Tomen
Aurora Michel

How does time-discretization affect training convergence rate in spike-based backpropagation as opposed to backpropagation through time?

1. Background

- __Spiking Neural Networks (SNNs)__: **energy-efficient**, **biologically plausible** alternatives to costly Deep Neural Networks. [1]
- These models can be simulated on **digital chips**, which do not work on continuous time. Therefore, the models would need to be time discretized. [3]
- **backpropagation through time**: tailor weights at every timestep [2] → SLAYER; Fig 1
- **spike-based backpropagation**: tailor weights at each output spike [2] → BATS; Fig.2
- SLAYER and BATS accompanied by explicit analysis and description of the respective backpropagation. Both use CuBa-LIF and achieve state-of-the art performance.
- **convergence rate:** how fast does the model learn; relevant for implementing online learning; insights into learning process

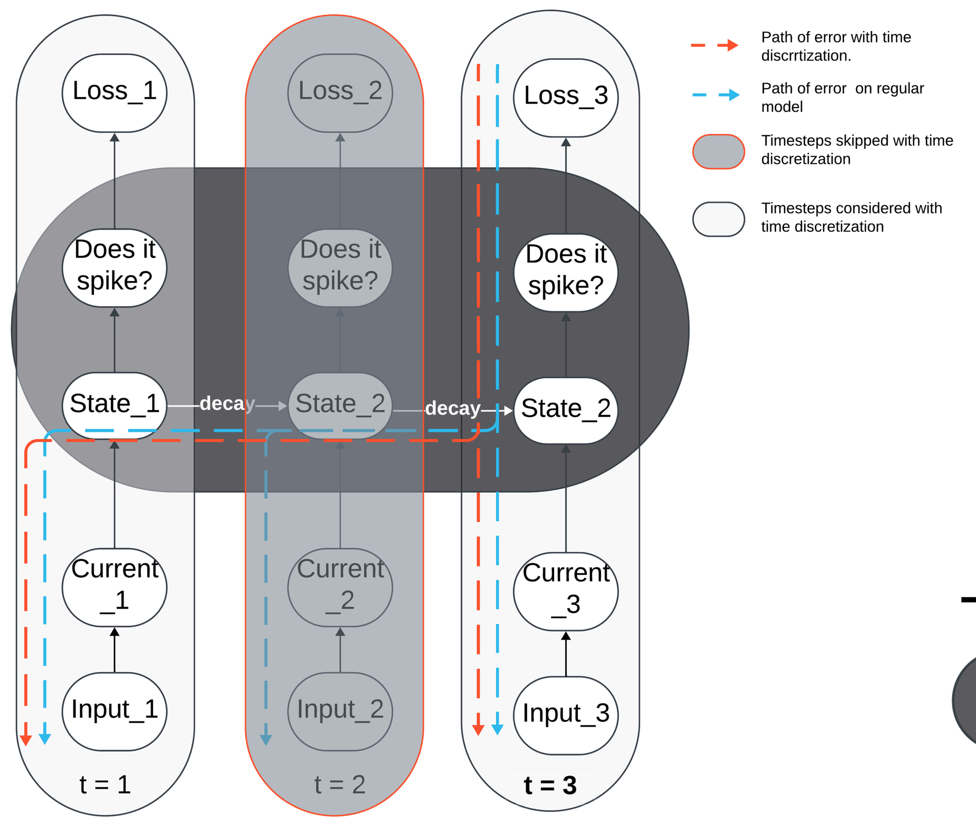


Fig 1 The figure illustrates the **dynamics of BPTT for an arbitrary neuron with and without time discretization.** The error progression for timestep 3 is shown. The regular model will propagate error on every timestep, but the time-discretized model will skip some.

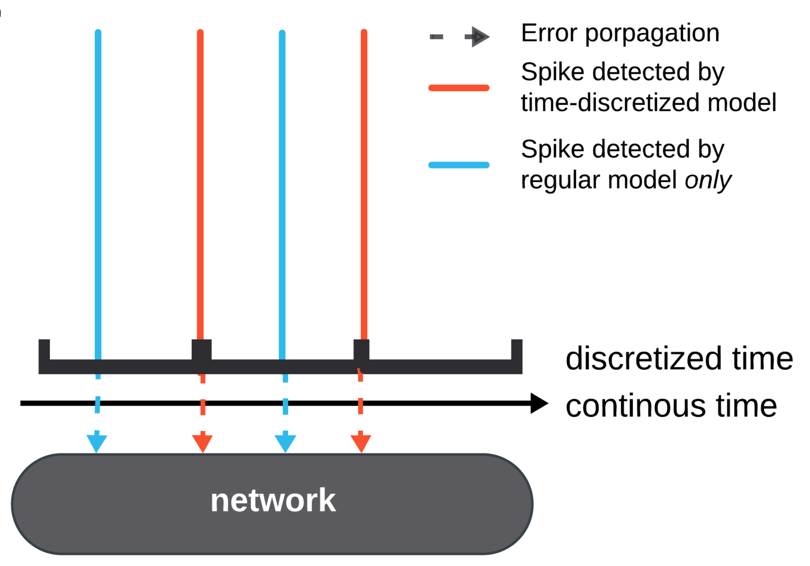


Fig 2. This figure illustrates **the effect of time discretization on a method with spike-based backpropagation.** The regular model will update the model at every output spike, whereas the time-discretized model will do that only on specific intervals equal to the timestep.

2. Objective

Compare the effects of **time-discretization** on the **convergence rate** of __BATS__ [5] (**spike-based** backpropagation) as opposed to __SLAYER__ [6] (backpropagation **through time**) for different timestep sizes

__Hypothesis:__ The **convergence rate** will **decrease** in both models as the as the timestep size grows, but it will be **more noticeable in spike-based** models as they depend on exact spiking times.

3. Methodology

CuBa-LIF

- The CuBa-LIF neuron describes the neuron state in terms of **membrane potential** which determines the spiking behavior of the neuron, and a **current** which represents the pre-synaptic potential.
- The neuron has been discretized based on the Loihi model.

SLAYER Time-Discretization

- The model is already working based on timesteps.
- Both forward propagation and backward propagation are time-discretized by skipping a number of timesteps equal to the timestep size.

BATS

- The model is adapted to use the time-discretized version of CuBa-LIF.
- The input is processed to delay spike times to a product of the time-step.
- Straight-through estimator is used for backpropagation.

Time Discretization

- There is a varying representation of time in time-based models (working in time-steps) and event-based models (working in seconds)
- Using total number of **timebins** and simulation time to derive timestep sizes

$$\text{timestep size} = \frac{\text{simulation time}}{\#\text{time bins}}$$

Datasets

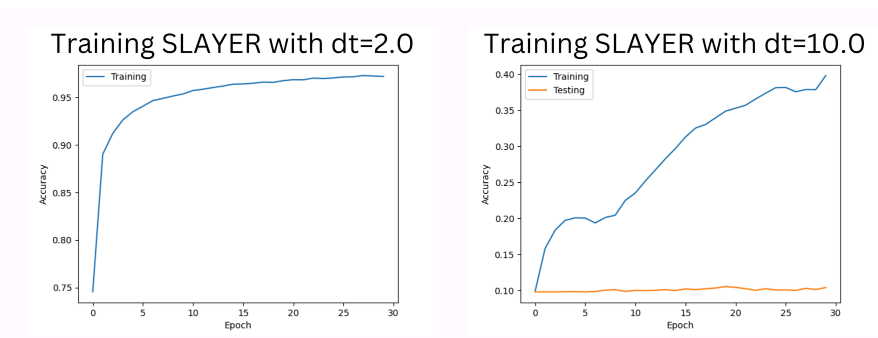
- **MNIST:** for BATS: 28x28 pixel images of handwritten digits
- **NMNIST**: For SLAYER; a neuromorphic version of MNIST; 34x34x2
- Two datasets are used due to need of out of scope model restructuring

Change in Convergence Rate

- becnhmark convergence rate: the convergence rate achieved by the non-discretized version of each model
- The change illustrates how the timestep size affects the initial model and allows for a comparison between the two models

$$\text{change in CR} = \frac{(\text{data point CR}) - (\text{benchmark CR})}{(\text{benchmark CR})}$$

4. Results and Discussion

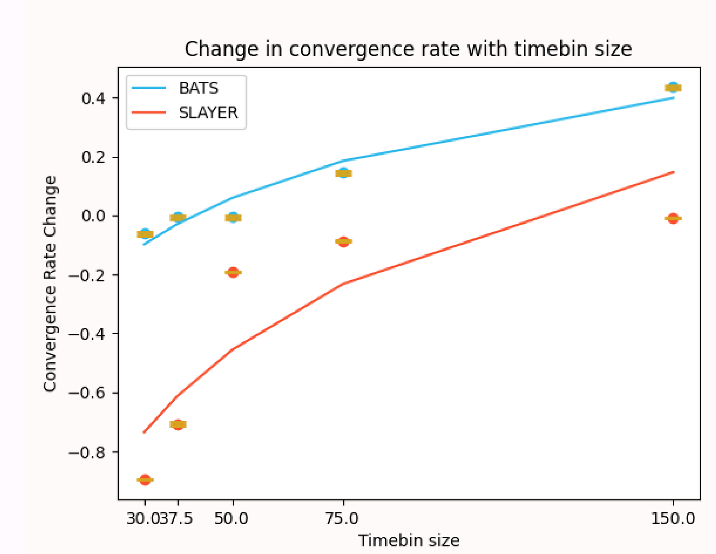


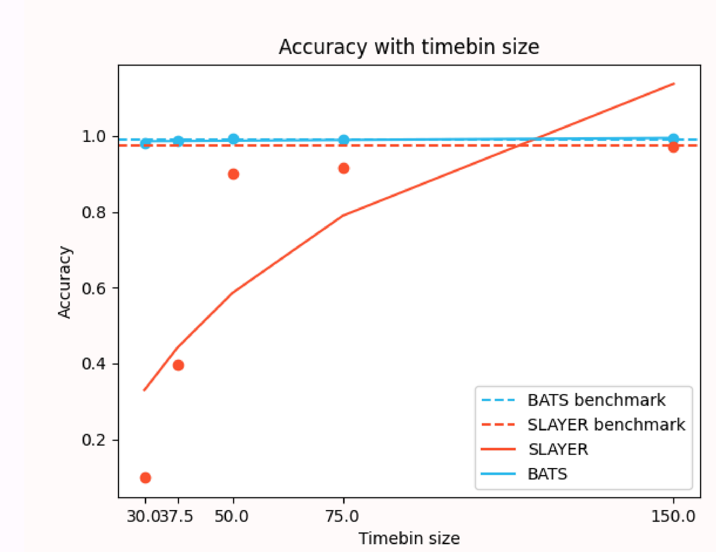
BATS

- *The accuracy and convergence rate decrease with timebins. ->decreasing accuracy of spike time approximation.*
- *The model performs better than baseline for timebins 75 and 150, which is surprising. -> sign of noise being filtered for the particular dataset.*
- *Seems to be almost unaffected by time discretization. -> independence from past timesteps; timestep sizes not big enough; the dataset is not neuromorphic.*

SLAYER

- *The accuracy and convergence rate decrease with timebins. -> large amount of timesteps missed.*
- *Model performs close to baseline for large number of bins, but drastically deteriorates. -> time dependence of the model; the neuron data decays at the same rate it is learned from.*





5. Conclusion

Time-discretization minimally affects the convergence rate of the spike-based BATS model for any timebin size. The BPTT SLAYER model is unaffected for small timesteps, but appears to self-destruct for large ones.

Limitations & Future Work

- Two different dataset formats were used. They need to be standardized for more conclusive results.
- The backpropagation mechanisms were tested on two different models. An attempt should be made to embed one into the other.
- A larger sample of timesteps should be tested to investigate model behavior further, especially spike-based models.

References

[1] M. Dampfhoffer, T. Mesquida, A. Valentian, and L. Anghel, “Are snns really more energy- efficient than anns? an in-depth hardware-aware study,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 7, no. 3, pp. 731–741, 2023.
[2] M. Dampfhoffer, T. Mesquida, A. Valentian, and L. Anghel, “Backpropagation-based learning techniques for deep spiking neural networks: a survey,” IEEE Transactions on Neural Networks and Learning Systems, pp. 1–16, 2023. Hal-04064177.
[3] J. K. Eshraghian, M. Ward, E. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, “Training spiking neural networks using lessons from deep learning,” CoRR, vol. abs/2109.12894, 2021.
[4] S. Valadez-Godínez, H. Sossa and R. Santiago-Montero, "The step size impact on the computational cost of spiking neuron simulation," 2017 Computing Conference, pp. 722-728, 2017
[5] F. Bacho and D. Chu, “Exploring Trade-Offs in Spiking Neural Networks,” Neural Computation, vol. 35, pp. 1627–1656, 09 2023
[6] S. B. Shrestha and G. Orchard, “Slayer: Spike layer error reassignment in time,” 2018