EBRAINS

Co-funded by the European Union

jaxsnn

Event-driven Gradient Estimation for Analog Neuromorphic Hardware

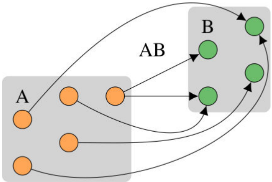
Eric Müller

mueller@kip.uni-heidelberg.de

BrainScaleS/EINC, Kirchhoff Institute for Physics
Ruprecht-Karls-Universität Heidelberg

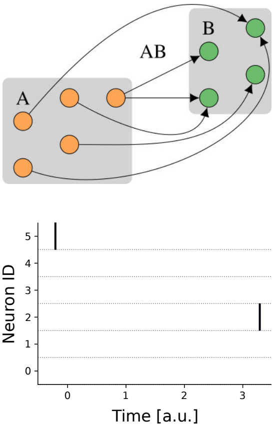
NICE 2024
La Jolla, CA, USA

Motivation I



- Spiking neural networks → timed events, sparse event-driven communication
- Neuromorphic hardware often provides high time resolution
- Data flow in SNNs → signal graphs

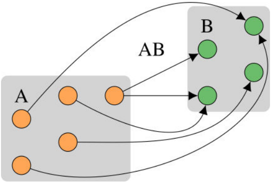
Motivation I

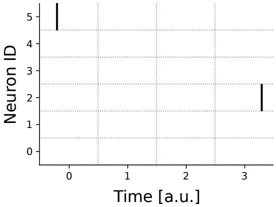


- Spiking neural networks $\rightarrow$ timed events, sparse event-driven communication
- Neuromorphic hardware often provides high time resolution
- Data flow in SNNs $\rightarrow$ signal graphs

Co-funded by the European Union
jaxsnn (/ˈdʒæksən/)
EBRAINS

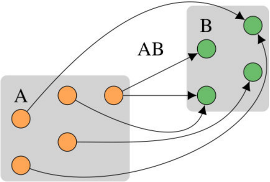
Motivation II

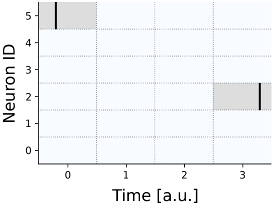




- But: Many gradient-based SNN training libraries sit on top of ANN ML toolkits,
  - using timestep-wise formulation of dynamics,
  - representing time as extra tensor dimension $\rightarrow$ fixed-resolution binning.

Motivation III





- ML toolkit function transformation (e.g., Autograd) capabilities often based on “tensor” data structures

⇒ Expression of system dynamics in an event-driven fashion inhibits use of function transformation capabilities (esp. Autograd)

Goals I

- Explicit handling of time & flexible data structures
- Event-driven numerical simulation and support for gradient estimation
- Composable function transformations, e.g.,
  - create jitted function (XLA)
  - create vectorized function by mapping function over argument axes
  - create function evaluating gradient (or vjp) of a function

Goals II

- Flexible swapping in/out of numerically simulated layers and other backends, e.g.,
  - blackbox simulations
  - neuromorphic hardware (BrainScaleS-2)
  - hardware simulation
- Offloading/code generation (e.g., for BSS-2 embedded plasticity processors)
- Support everything that the BSS-2 hardware does, e.g., AdEx, multi-compartment neurons, programmable plasticity, . . .

  ⇒ the one top-level API to replace all other (BSS-2) modeling APIs (`PyNN.brainscales`, `hxtorch`)

→ Be useful for “No backprop, please!” use cases

jaxsnn

- `jaxsnn` library
  `https://github.com/electronicvisions/jaxsnn`
- Numerical simulation of SNNs and gradient estimation
- Built on top of `JAX`:
  - Sufficient flexibility to express event-driven system dynamics
  - with function transformation capabilities,
  - incl. autograd, parallelization, vectorization and offloading to numerical hardware accelerators.

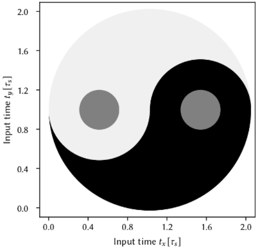
Co-funded by the European Union

jaxsnn

```
from jax import custom_vjp, grad

@jax.custom_vjp
def apply_fn(topology, params, input_spikes):
    return hxtorch.snn.run(params, input_spikes)

def apply_forward(topology, params, input_spikes):
    observables = apply(topology, params, input_spikes)
    return observables, (input_spikes, params, topology, \
                                                 observables)

def apply_backward(res, g):
    input_spikes, params, topology, observables = res
    vjp = event_prop_grads(res, g)
    return (vjp, None)

apply_fn.defvjp(apply_forward, apply_backward)

observables = apply_fn(topology, params, input_spikes)
grads = jax.grad(loss_fn)(observables, target)
```


EBRAINS

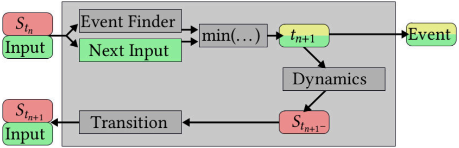
jaxsnn — current state



- Neuromorphic hardware platform: BrainScaleS-2
- Event-based SNN training: EventProp
- Dataset: YinYang
- Validated against a time-grid-based implementation and analytical solutions
- Initial code base was made available for last year's (2023) CapoCaccia workshop
- Integrated into the latest EBRAINS Software Distribution (`esd@24.04` RC)

ESD

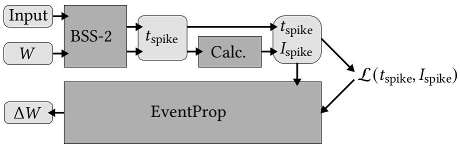
Event-based dynamics (of spiking neurons) in `jaxsnn` I



Simulating hybrid (continuous dynamics with transition events) systems in `jaxsnn`:

- Initial state and input data
  $\rightarrow$ next event time, evolve dynamics
- Apply transition/discontinuity
  $\rightarrow$ new state, output event

Event-based dynamics (of spiking neurons) in `jaxsnn` II



Example: EventProp algorithm for BSS-2 in-the-loop training

- Delegate “Event Finder” to BrainScaleS-2 $\rightarrow$ observe spikes
- Derive synaptic current at spike time numerically
- Based on a loss function compute parameter updates using EventProp

Co-funded by the European Union

jaxsnn (/'dʒæksən/)

EBRAINS

Result accuracies on YY test set (in sim. & BSS-2-in-the-loop)

Gradient Estimator	Substrate	Size	Loss	Acc. [%]
Fast & Deep [1]	sim	120	TTFS	95.9 ± 0.7
Fast & Deep [1]	BSS-2	120	TTFS	95.0 ± 0.9
EventProp [2]	sim	200	TTFS	98.1 ± 0.2
EventProp [3]	sim	120	MOT	97.9 ± 0.6
EventProp [3]	BSS-2	120	MOT	96.1 ± 0.9
Norse in JAX	sim	120	MOT	96.4 ± 0.2
jaxsnn F&D	sim	120	TTFS	98.1 ± 0.3
jaxsnn EventProp	sim	120	TTFS	98.2 ± 0.2
jaxsnn EventProp	mock	100	TTFS	98.0 ± 0.3
jaxsnn EventProp	BSS-2	100	TTFS	94.8 ± 0.2

Co-funded by the European Union

jaxsnn (/'dʒæksən/)

EBRAINS

Conclusion

- Initial implementation of a modular and composable event-driven numerical SNN simulation tool with gradient estimation
- Validated against existing baseline experiments
- Support for offloading “event finding” in the forward pass
- Proof-of-concept implementation based on `LIF_curr_exp` dynamics

Outlook

- Support all BSS-2 hardware features (and beyond):
  - Implementation of full AdEx parameterization and complex structured neurons (to exploit the full BSS-2 hardware neuron feature set)
  - Online plasticity
  - (Delays (axonal and synaptic))
- Implementation performance roughly comparable to `hxtorch` library, an in-depth analysis is pending (scaling behavior (network size, depth, sparsity, used observables) vs. overhead in runtime, memory consumption)

Co-funded by the European Union

EBRAINS

Co-funded by the European Union

jaxsnn (/ˈdʒæksən/)

EBRAINS

Thank you!

Eric Müller

Moritz Althaus

Elias Arnold

Philipp Spilger

Christian Pehle

Johannes Schemmel

ESD

References

[1] J. Göltz et al. "Fast and energy-efficient neuromorphic deep learning with first-spike times". In: Nature Machine Intelligence 3.9 (2021), pp. 823–835. DOI: 10.1038/s42256-021-00388-x.

[2] T. C. Wunderlich and C. Pehle. "Event-based backpropagation can compute exact gradients for spiking neural networks". In: Scientific Reports 11.1 (2021), pp. 1–17. DOI: 10.1038/s41598-021-91786-z.

[3] C. Pehle, L. Blessing, E. Arnold, E. Müller, and J. Schemmel. Event-based Backpropagation for Analog Neuromorphic Hardware. 2023. arXiv: 2302.07141 [q-bio.NC].