Training physical models of deep spiking neural networks

BrainComp 2022 (19. - 23. September) | Sebastian Billaudelle | Kirchhoff-Institute for Physics, Heidelberg University

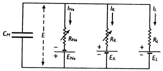
Neuromorphic computing

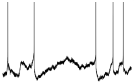
νεῦρον + μορφή

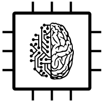
Neuromorphic systems replicate (aspects of) the nervous system to perform computation.

$$
\begin{aligned}
I &= C_M \frac{\mathrm{d}V}{\mathrm{d}t} + \bar{g}_{\mathrm{K}} n^4 (V - V_{\mathrm{K}}) + \ldots \\
\mathrm{d}n/\mathrm{d}t &= \alpha_n(1-n) - \beta_n n, \\
\mathrm{d}m/\mathrm{d}t &= \alpha_m(1-m) - \beta_m m, \\
\mathrm{d}h/\mathrm{d}t &= \alpha_h(1-h) - \beta_h h,
\end{aligned}
$$

**

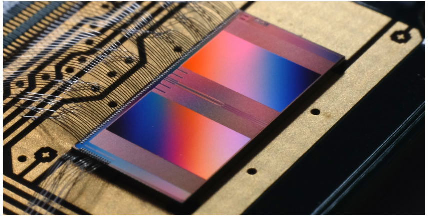






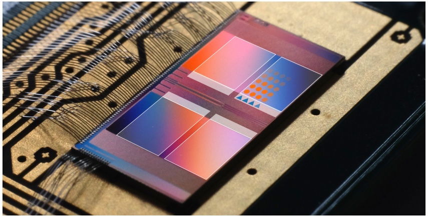
* Surace and Pfister, 2015 ** Hodgkin and Huxley, 1952

BrainScaleS-2



- mixed-signal neuromorphic system, 65 nm
- physical emulation of neuronal and synaptic dynamics

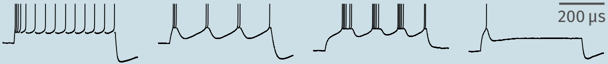
BrainScaleS-2



- mixed-signal neuromorphic system, 65 nm
- physical emulation of neuronal and synaptic dynamics

- 512 (multicompartment-capable) neurons
- 512 × 256 synapses

- LIF + AdEx dynamics
- multicompartment-capable
- flexibly parameterizable
- 1000× accelerated



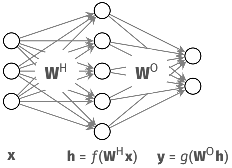
Billaudelle, Weis, et al., 2022

- in-memory-compute arrays
- local weight storage (6 bit + sign)
- programmable plasticity

Friedmann, Schemmel, et al., 2017

Training multi-layer spiking neural networks

Spatial credit assignment

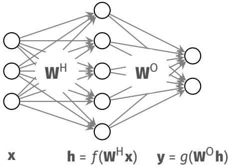


→ calculate gradients of a loss function L w.r.t. network parameters

→ propagate errors through network layers by computing gradients

Training multi-layer spiking neural networks

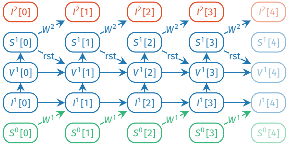
Spatial credit assignment



→ calculate gradients of a loss function L w.r.t. network parameters

→ propagate errors through network layers by computing gradients

Temporal credit assignment



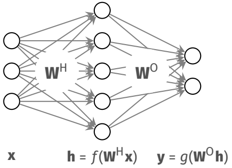
- capture temporal propagation of information

→ construct multidimensional compute graph

→ backpropagate through time (BPTT, Werbos, 1990)

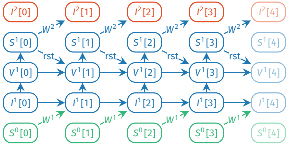
Training multi-layer spiking neural networks

Spatial credit assignment



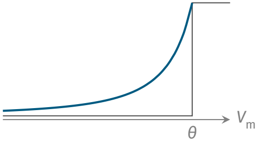
- → calculate gradients of a loss function L w.r.t. network parameters
- → propagate errors through network layers by computing gradients

Temporal credit assignment



- ○ capture temporal propagation of information
- → construct multidimensional compute graph
- → backpropagate through time (BPTT, Werbos, 1990)

Gradients of spikes?



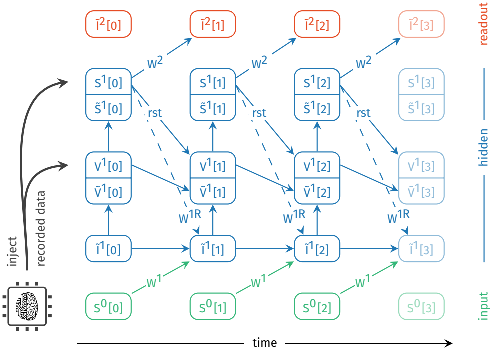
- ○ gradients can be estimated only in presence of spike
- ○ spikes introduce discontinuities in parameter space
- → adopt surrogate gradients (Neftci, Mostafa, et al., 2019)

Training event-based analog systems

Cramer, Billaudelle, et al., 2022

∂ 

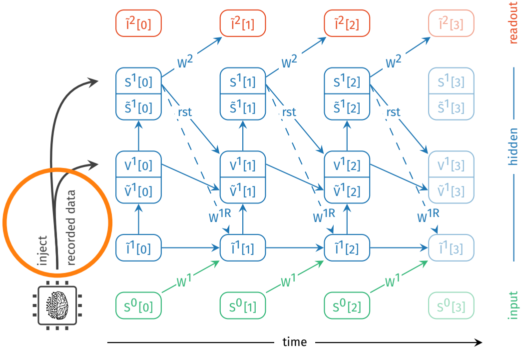
$$\frac{\partial \,\cdot}{\partial\theta} = ?$$



Training event-based analog systems

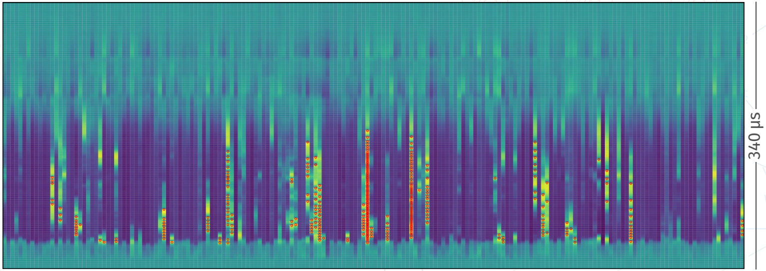
Cramer, Billaudelle, et al., 2022

$$\frac{\partial \,\square}{\partial \theta} = ?$$



Training event-based analog systems

Cramer, Billaudelle, et al., 2022



parallel recording of membrane traces and spikes (1.2 Gbit s^{-1})

Highly streamlined programming interface

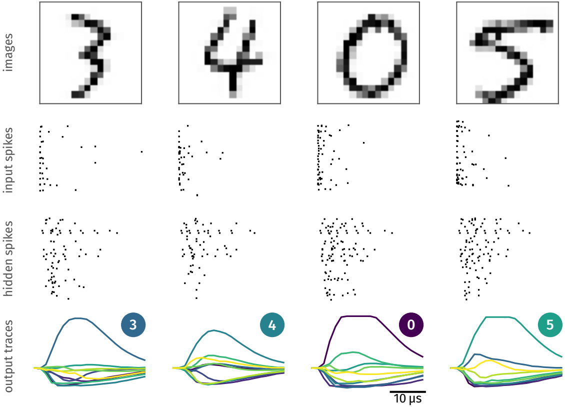
```
network = strobe.nn.Network([
        strobe.nn.Linear(n_input, n_hidden),
        strobe.nn.LIFLayer(n_hidden),
        strobe.nn.Linear(n_hidden, n_output),
        strobe.nn.LILayer(n_output)
    ])
network.connect(...)

y = network(x)

loss = my_loss(y, y_prime)
loss.backward()
```

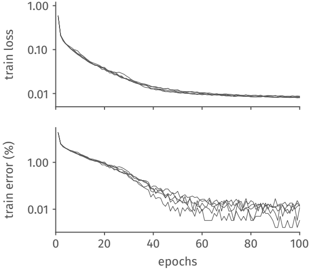
built on top of

PyTorch

Classification of handwritten digits

Cramer, Billaudelle, et al., 2022

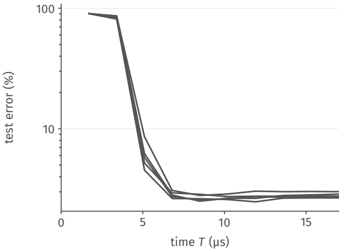




97.6(1) % test accuracy

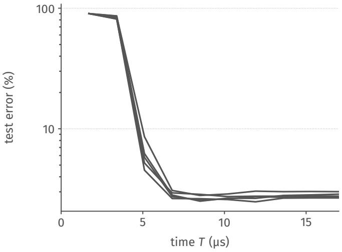
Ultrafast and efficient inference

Cramer, Billaudelle, et al., 2022



Ultrafast and efficient inference

Cramer, Billaudelle, et al., 2022

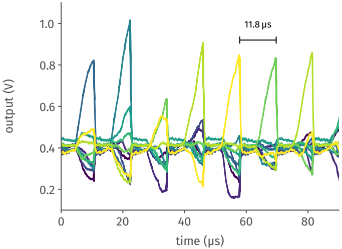


< 8 μs

latency

Ultrafast and efficient inference

Cramer, Billaudelle, et al., 2022

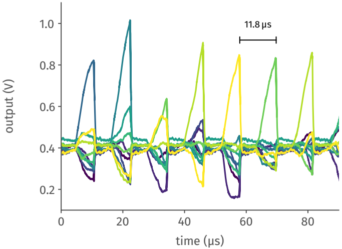


< 8 µs

latency

Ultrafast and efficient inference

Cramer, Billaudelle, et al., 2022

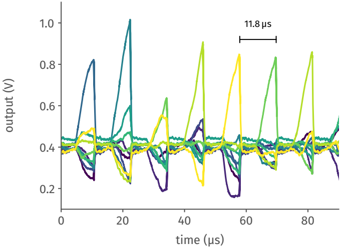


< 8 µs
latency

84 k/s
throughput

Ultrafast and efficient inference

Cramer, Billaudelle, et al., 2022



< 8 µs
latency

84 k/s
throughput

200 mW
power

2.4 µJ
energy

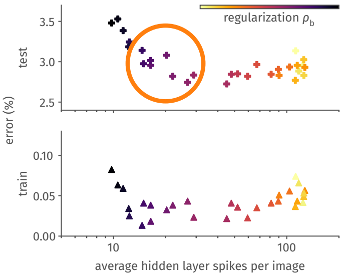
Ultra-efficient messaging within the network

Cramer, Billaudelle, et al., 2022

- during training: optimize for reduced network activity

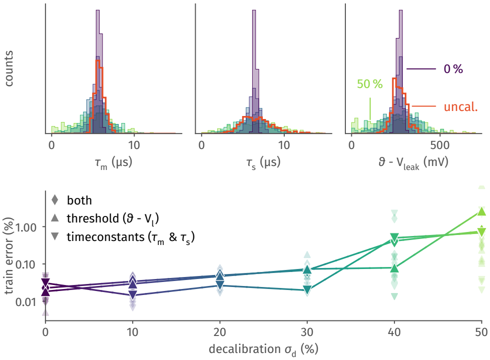
→ extreme temporal sparsity

only 0.08 spikes per neuron!



Robustness to variability in analog circuits

Cramer, Billaudelle, et al., 2022

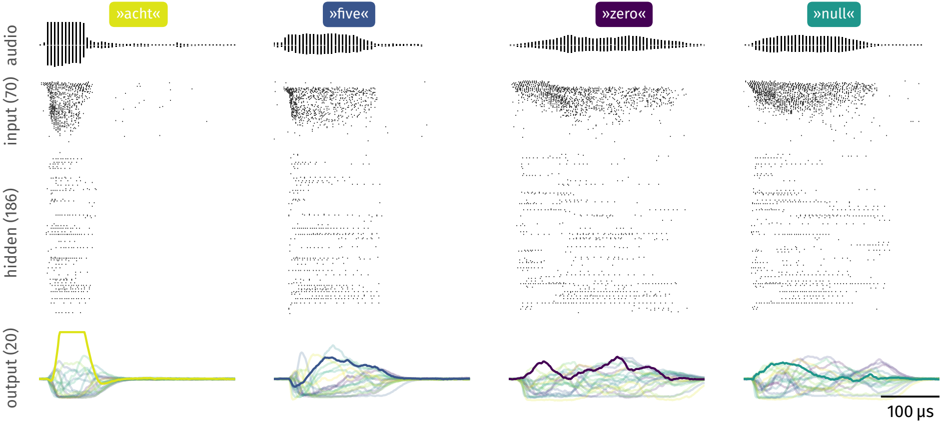


- analog circuits suffer from production-induced parameter variability
- train on deliberately detuned neuron dynamics

→ training scheme inherently resilient to these effects

Recurrent spiking neural networks for speach recognition

Cramer, Billaudelle, et al., 2022



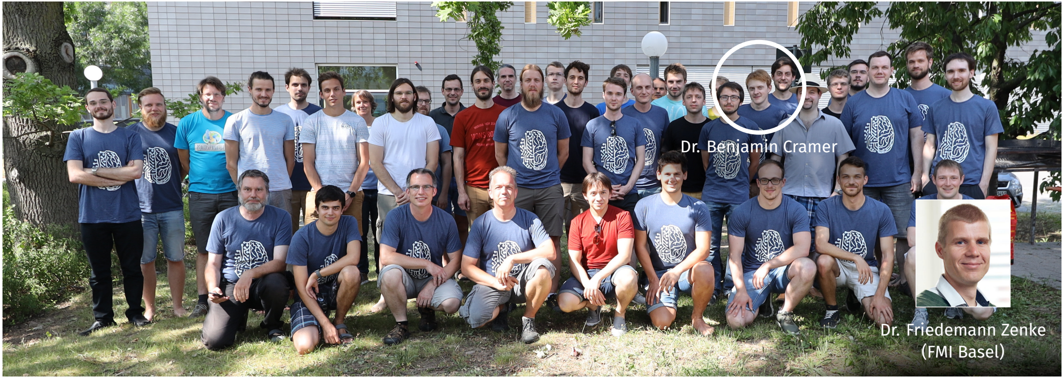
76.2 % (96.6 %) on BrainScaleS-2

Summary

- high-level user interface, integrated into PyTorch
- robust, self-correcting training

- efficiency through analog computation and event-based communication
- low classification latencies
- high throughput

Thanks!



References

Billaudelle, Sebastian et al. (2022). “An accurate and flexible analog emulation of AdEx neuron dynamics in silicon”. In: arXiv preprint arXiv:2209.09280.

Cramer, Benjamin et al. (2022). “Surrogate gradients for analog neuromorphic computing”. In: Proceedings of the National Academy of Sciences 119.4, e2109194119.

Friedmann, S. et al. (Feb. 2017). “Demonstrating Hybrid Learning in a Flexible Neuromorphic Hardware System”. In: IEEE Transactions on Biomedical Circuits and Systems 11.1, pp. 128–142. ISSN: 1932-4545. DOI: 10.1109/TBCAS.2016.2579164.

Hodgkin, Alan L and Andrew F Huxley (1952). “A quantitative description of membrane current and its application to conduction and excitation in nerve”. In: The Journal of physiology 117.4, p. 500.

Neftci, Emre O et al. (2019). “Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks”. In: IEEE Signal Processing Magazine 36.6, pp. 51–63.

Surace, Simone Carlo and Jean-Pascal Pfister (2015). “A statistical model for in vivo neuronal dynamics”. In: Plos one 10.11, e0142435.

Werbos, Paul J (1990). “Backpropagation through time: what it does and how to do it”. In: Proceedings of the IEEE 78.10, pp. 1550–1560.