Learning algorithms for spiking and physical neural networks
Today: Overview recent work on learning algorithms

- Surrogate gradients for spiking neural networks
  - Getting spiking neural networks to do something interesting (recap)
    Neftci, Mostafa, and Zenke (2019) *IEEE SPM*
  - Sidestepping device mismatch on analog neuromorphic substrates

- Resurrecting local learning rules (beyond backprop)
  - Training noisy substrates with holomorphic equilibrium propagation
    Laborieux and Zenke (2022) *Neurips*
  - Online self-supervised learning with local learning rules
    Halvagal and Zenke (2023) *Nature Neuroscience*
How do we get spiking neural networks to do something interesting?

Through end-to-end training!
Recap: Training spiking networks end-to-end

- Spiking neurons & networks are RNNs
- They have implicit and explicit recurrence
- Known training procedures for networks with hidden units
  - Backpropagation-through time (BPTT)
  - Real-time recurrent learning (RTRL)

\[
S_i^{(1)}[n] = \Theta \left( U_i^{(1)}[n] - \vartheta \right)
\]

\[
U_i^{(1)}[n + 1] = \beta U_i^{(1)}[n] + I_i^{(1)}[n] - S_i[n]
\]

\[
I_i^{(1)}[n + 1] = \underbrace{\alpha I_i^{(1)}[n]}_{\text{exp. current decay}} + \sum_j W_{ij} S_j^{(0)}[n]
\]

Forward Euler integration

Neftci, Mostafa, & Zenke (2019)
Problem: The derivative of a spike train is zero almost everywhere
Solution: Surrogate gradient

Bohte (2011), Esser et al. (2015), Bellec et al. (2018), Shrestha & Orchard (2018), Zenke & Ganguli (2018), ...

In ML: “Straight-through estimators” Bengio et al. (2013)

\[ \Theta(U(t) - \theta) \approx \sigma(U(t)) \]

\[ \frac{\partial L}{\partial w_k} \neq 0 \]

Neftci, Mostafa, & Zenke (2019)
Surrogate gradients
No assumption about rate or time coding required.

Cramer, Stradmann, Schemmel, and Zenke (2020)
Loss landscape of a spiking net (2D projection)

Integrated surrogate gradient

**Problem:** Surrogate gradients are a heuristic and lack theory.
Surrogate gradients are related to well-defined gradients in expectation in single stochastic neurons


But does not work in multi-layer networks because it breaks the chain rule:

$$
E \left[ \frac{\partial y}{\partial p_y} \frac{\partial p_y}{\partial h_2} \frac{\partial h_2}{\partial p_2} \cdots \frac{\partial p_1}{\partial w_1} \right] \neq E \left[ \frac{\partial y}{\partial p_y} \right] E \left[ \frac{\partial h_2}{\partial p_2} \right] E \left[ \frac{\partial p_2}{\partial w_1} \right] \cdots E \left[ \frac{\partial p_1}{\partial w_1} \right]
$$
Stochastic Automatic Differentiation provides missing theoretical foundation for surrogate gradients

- Finite differences? Does not scale, high variance → not an option

- “Stochastic automatic differentiation”

\[
\frac{\partial}{\partial w_1} \mathbb{E}[y] \approx \frac{\partial}{\partial h_2} \mathbb{E}[y|h_2^*] \frac{\partial}{\partial h_1} \mathbb{E}[h_2|h_1^*] \frac{\partial}{\partial w_1} \mathbb{E}[h_1|x]
\]

- Surrogate gradients fall out of this framework
How do we perform efficient inference with (spiking) neural networks?

With ultra-low power neuromorphic hardware!
(use the device physics)
Problem: Device mismatch

Software implementation

Hardware implementation

→ costly calibration
To study this question we used the BrainScaleS-2 analog neuromorphic hardware system.
In-the-loop surrogate gradient training
Forward-pass on chip and backward pass in software

1) Forward pass on chip

2) Measure on-chip analog voltage traces

3) Inject voltage into computational graph

4) Compute surrogate gradients → update weights

Functional spiking neural networks trained on BrainScaleS-2 analog neuromorphic hardware


$>80k$ images/sec @ 200mW
Surrogate gradient learning self-calibrates the analog neuromorphic substrate

Speech classification and keyword spotting (SHD)

**Summary:** Voltage aware surrogate gradients can self-calibrate analog neuromorphic substrates

Still, training was done offline and used backprop.
Backprop is difficult to implement on neuromorphic systems

1. Nonlinear computation
   \[ s_i = \sigma \left( \sum_j w_{ij} s_j + b_i \right) \]

2. Error Backpropagation (BP)
   \[ \Delta w_{ij} \propto - \frac{dL}{dw_{ij}} = - \delta_i \sigma(s_j) \]

loss function quantifying good or bad

Rumelhart et al. Nature 1986

non local
Question
How to train noisy physical networks without backprop?

Laborieux and Zenke (2022) Neurips
Holomorphic Equilibrium Propagation Computes Exact Gradients Through Finite Size Oscillations
Equilibrium Propagation (EP) is an alternative

\[ E(s, W) = \frac{1}{2} \sum_i s_i^2 - \sum_{i<j} w_{ij} \sigma(s_i) \sigma(s_j) \]
Local learning rule

$$\Delta w_{ij} \propto \sigma(s_{\beta,i}^*)\sigma(s_{\beta,j}^*) - \sigma(s_{0,i}^*)\sigma(s_{0,j}^*)$$

$$\frac{-d\mathcal{L}}{dw_{ij}}$$

when $\beta \to 0$
Classic Equilibrium Propagation is noise sensitive

Laborieux and Zenke (2022) Neurips
Complex analysis: Derivatives can be expressed as integrals

- Complex differentiability:
  \[ f'(a) := \lim_{z \to a} \frac{f(z) - f(a)}{z - a} \]
  ‘holomorphic’

- Cauchy integral:
  \[ f'(a) = \frac{1}{2i\pi} \oint_{\gamma} \frac{f(z)}{(z - a)^2} \, dz \]

Laborieux and Zenke (2022) Neurips
Integration over one oscillation yields local learning rule

Replace derivative by a Cauchy integral

\[
- \frac{d\mathcal{L}}{dW} \bigg|_{w_0} = \left. \frac{d}{d\beta} \right|_{\beta=0} \left( \sigma(s^*_\beta)\sigma(s^*_\beta)^T \right) = \frac{1}{2i\pi} \oint_{\gamma} \frac{\sigma(s^*_\beta)\sigma(s^*_\beta)^T}{\beta^2} d\beta
\]

\( \beta \in \mathbb{C} \)

We choose the path: \( t \in [0, T_{osc}] \mapsto \beta(t) = |\beta| e^{2i\pi t/T_{osc}} \)

\[
- \frac{d\mathcal{L}}{dW} \bigg|_{w_0} = \frac{1}{T_{osc} |\beta|} \int_0^{T_{osc}} \sigma(s^*_{\beta(t)})\sigma(s^*_{\beta(t)})^T e^{-2i\pi t/T_{osc}} dt
\]

Gradient = first Fourier coefficient of nonlinear neural oscillations
Holomorphic Equilibrium Propagation is robust

Laborieux and Zenke (2022) Neurips
Holomorphic EP scales to ImageNet (thanks to larger teaching amplitudes)

Laborieux and Zenke (2022) Neurips
Summary: Holomorphic equilibrium propagation allows computing exact gradients on noisy physical systems (without backprop)

Laborieux and Zenke (2022) Neurips
Holomorphic Equilibrium Propagation Computes Exact Gradients Through Finite Size Oscillations

Follow-up paper dealing with weight asymmetry:
Laborieux and Zenke (2024) accepted at ICLR
Improving equilibrium propagation without weight symmetry through Jacobian homeostasis

Ongoing work: Make it real
Latent Predictive Learning
Online self-supervised learning with local rules

Halvagal and Zenke (2023) Nat Neurosci
Disentangled representations

"Cat"

"Dog"
**Problem:** Hebbian plasticity, a bio-inspired local learning rule, does not learn *good* representations in deep nets.
Idea: Optimize for latent space prediction

predictive learning

Oja (1982):
Hebbian plasticity maximizes variance
Combining latent space prediction and Hebbian plasticity yields local learning rule

\[ \mathcal{L} = \mathcal{L}_{\text{Pred.}} + \mathcal{L}_{\text{Hebb}} \]

Resulting learning rule is local!

“Latent Predictive Learning LPL”
LPL disentangles objects from video data

https://github.com/deepmind/3d-shapes
LPL learns invariant representations from augmented images

- VGG-11 model
- Neurons in all layers learn with LPL
- No backprop
LPL learns invariant representations from augmented images

- VGG-11 model
- Neurons in all layers learn with **LPL**
- No backprop

After “watching” millions of image sequences ...

![Graph showing linear readout accuracy over layers for LPL, LPL shuffled, Hebb, Hebb off, and Decorr. off models.](https://www.zenkelab.org)
LPL can be formulated as a local spiking learning rule

Based on SuperSpike: Zenke & Ganguli (2018)

\[ \frac{dw_{ij}}{dt} = \eta \alpha \left( \varepsilon \left. \frac{S_j(t)}{S_j(t)} \right| f'(U_i(t)) \right) \left[ \alpha \left( \frac{- \left( S_i(t) - S_i(t - \Delta t) \right)}{\text{predictive}} \right) + \frac{\lambda}{\sigma_i^2 + \xi} \left( S_i(t) - \bar{S}_i(t) \right) \right] + \eta \delta S_j(t) \]

transmitter-triggered

\[ \mathcal{L} = \mathcal{L}_{\text{pred}} + \mathcal{L}_{\text{Hebb}} + \text{inhibitory neurons & plasticity} \]
LPL learns interesting features in streaming data

Inhibitory Plasticity
Latent Predictive Learning: Enables online learning with a local rule without supervision (also works in spiking nets)

Halvagal and Zenke (2023) Nat Neurosci
Summary

- Surrogate gradients allow training spiking neural networks end to end. Neftci, Mostafa, and Zenke (2019) IEEE SPM


- Latent predictive learning enables online learning without supervision. Halvagal and Zenke (2023) Nature Neuroscience
Way forward for online learning for Edge AI

• Need lightweight online learning rules:
  - Must be robust to noise and heterogeneity
  - No Backprop please!
  - Algorithms like holomorphic EP are promising, but must be practical (real-numbered)
  - Close the gap between self-supervised and supervised learning.

• Need joint efforts in algorithms and hardware development
Thanks!

The team [https://zenkelab.org/team/](https://zenkelab.org/team/)

Axel Laborieux  Mattias Nilsson  Manu Srinath Halvagal  Julian Rossbroich  Julia Gygax  Peter Buttaroni  Jeremias Seitz  Atena Mohammadi  Tengjun Liu  Aaron Spieler

Collaborators

Benjamin  Sebastian  Johannes Schemmel  Uni Heidelberg  Emre Neftci  FZ Julich

Funding

Swiss National Science Foundation

Andreas Luthi (FMI), Georg Keller (FMI), Rainer Friedrich (FMI), Emre Neftci (FZ Julich), Johannes Schemmel (Uni Heidelberg), Richard Naud (uOttawa), Blake Richards (Mila)