Brain inspirations for power efficient Artificial intelligence

November 2021

Guillaume Bellec

Post-doc in the lab of Computational Neuroscience, EPFL

@BellecGuill

guillaume.bellec@epfl.ch

Part 1. Long-short term memory and back-prop through time in spiking neural networks (TU GRAZ)

[1] Long short-term memory and learning-to-learn in networks of spiking neurons (NeurIPS 2018)
Bellec*, Salaj*, Subramoney*, Legenstein, Maass


F. Scherr*


D. Salaj


E. Hajek


A. Subramoney

Part 2. Eligibility propagation: credit assignment in time with eligibility traces (TU GRAZ)

[2] Bellec*, Scherr*, Subramoney, Hajek, Salaj, Legenstein, & Maass (Nature comm. 2020)
A solution to the learning dilemma for recurrent networks of spiking neurons


R. Legenstein


W. Maass

Part 3. **Local plasticity** rules can learn deep representations using **self-supervised** contrastive predictions (EPFL)

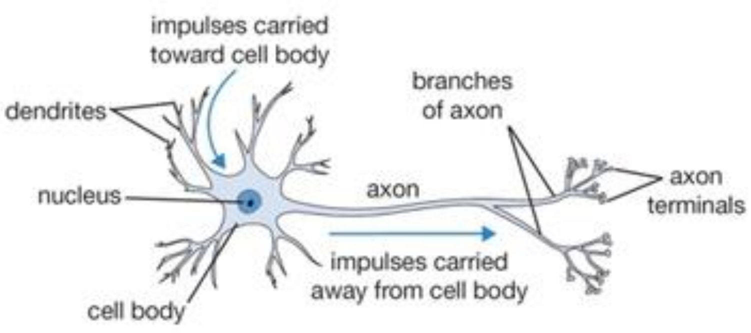
[3] Local plasticity rules can learn deep representations using self-supervised contrastive predictions (NeurIPS 2021)
Bernd Illing, Jean Ventura, Guillaume Bellec*, Wulfram Gerstner*


B. Illing


W. Gerstner

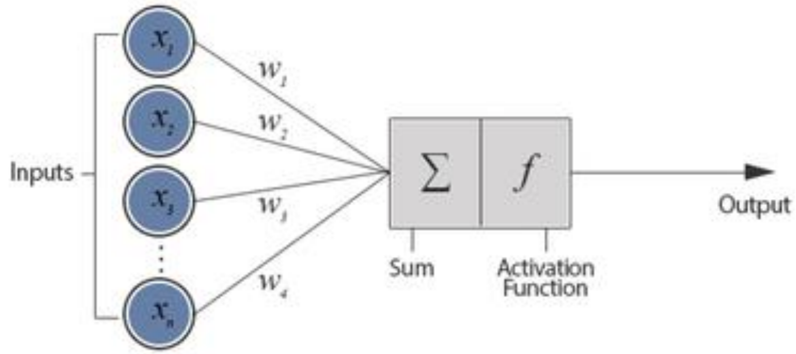
A simple spiking neuron model

Brain

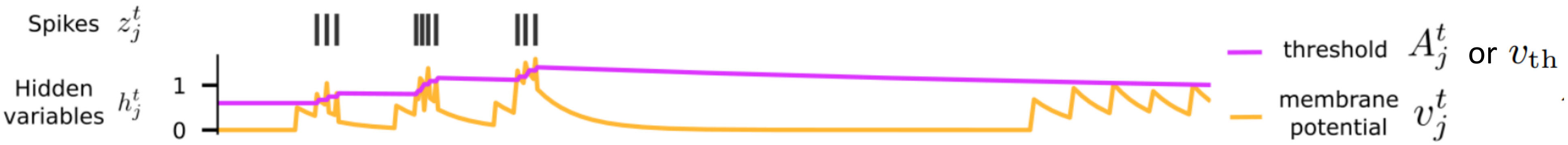


Picture from datacamp.com, Keras tutorial

Artificial models



The **Leaky Integrate and Fire** (LIF) is a simple biophysical model of a neuron that captures the spiking dynamics of neurons in the brain.



$$v_j^{t+1} = \alpha v_j^t + \sum_{i \neq j} W_{ji}^{\mathrm{rec}} z_i^t + \sum_i W_{ji}^{\mathrm{in}} x_i^{t+1} - z_j^t v_{\mathrm{th}}$$

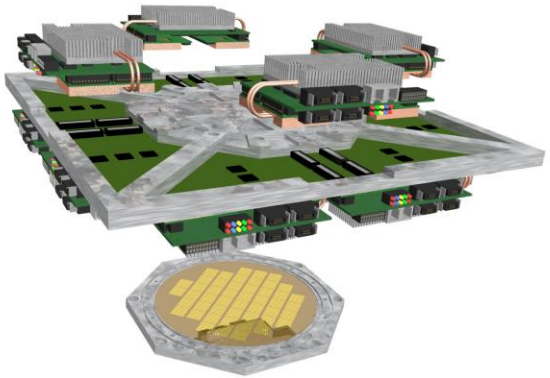
$$z_j^t = H\left(v_j^t - v_{\mathrm{th}}\right),$$

Neuromorphic hardware

Neuromorphic sensor

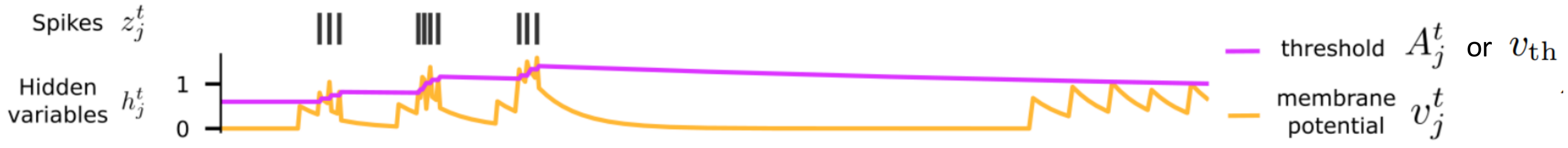
Digital Neuromorphic hardware

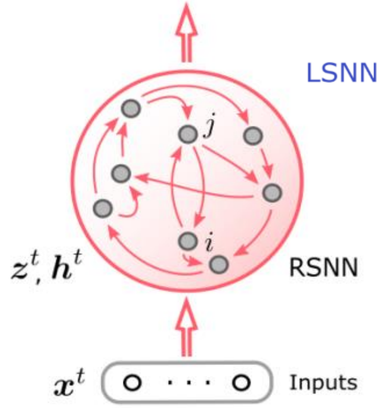
Analog neuromorphic hardware



Recurrent spiking neural networks (RSNN)

Adaptive Leaky integrate and fire (ALIF)



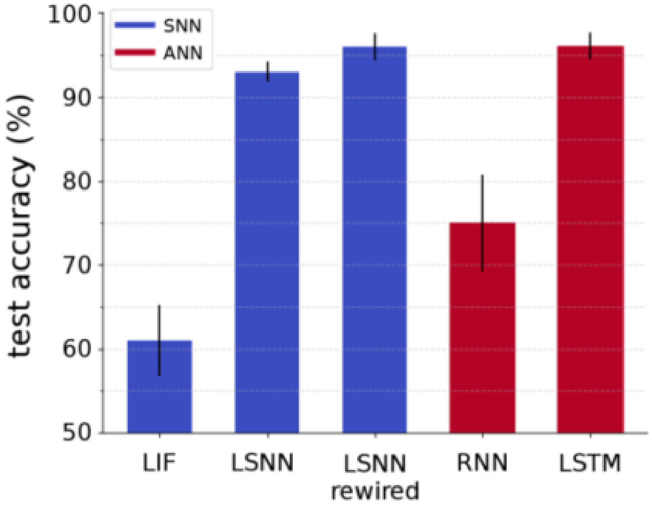


The learning performance is quantified for the loss function E

Temporal credit assignment is done with Back prop through time (BPTT) for RSNNs [1,2] which computes the gradient:

$$\Delta W_{ji} \propto -\frac{dE}{dW_{ji}}$$

LSNN: Long-short term memory **S**piking **N**eural **N**etwork, a recurrent network of adaptive LIF (A.LIF) neurons



[1] Long short-term memory and learning-to-learn in networks of spiking neurons (NeurIPS 2018)
Bellec*, Salaj*, Subramoney*, Legenstein, Maass

[2] Gradient Descent for Spiking Neural Networks (NeurIPS 2018)
Huh, Sejnowski

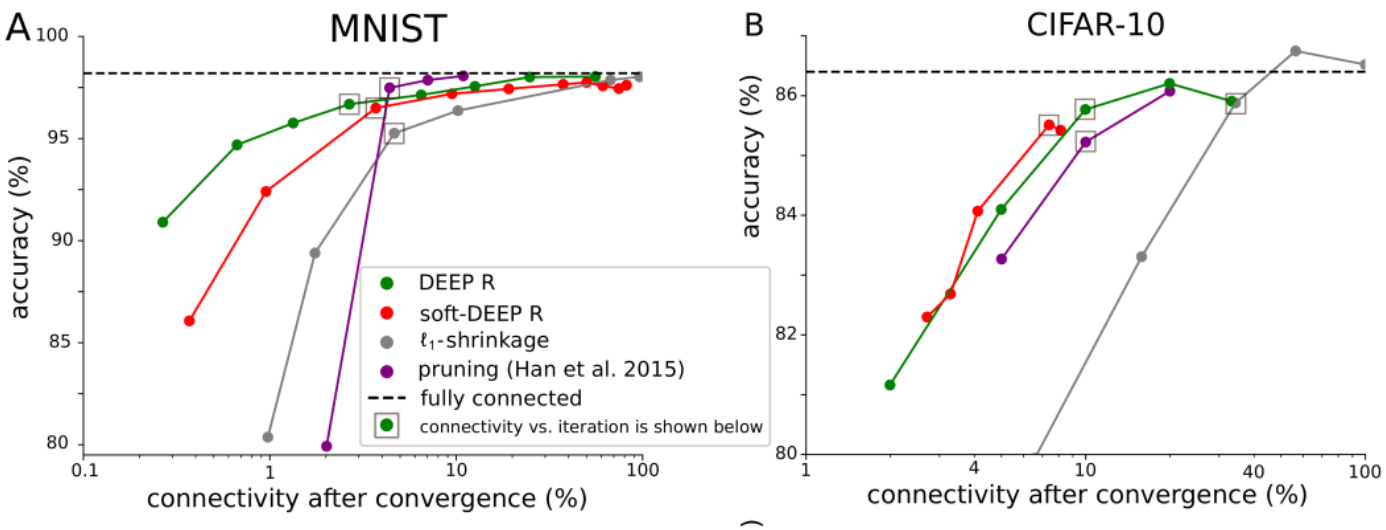
Deep rewiring

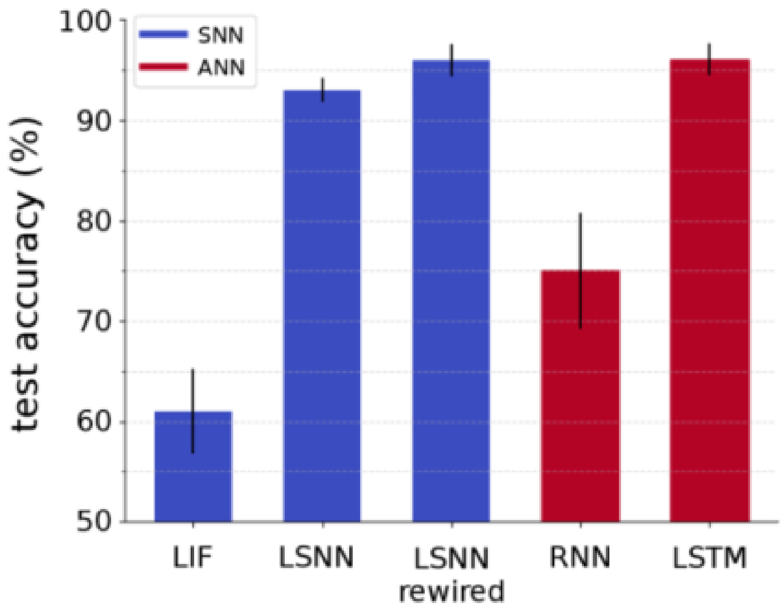
[1] Deep Rewiring: Training very sparse deep networks (ICLR 2018)
Bellec, Kappel, Maass, Legenstein

[2] Liu, C., Bellec, G., Vogginger, B., Kappel, D., Partzsch, J., Neumärker, F., ... & Mayr, C. G. (2018). Memory-efficient deep learning on a SpiNNaker 2 prototype. *Frontiers in neuroscience*, *12*, 840.

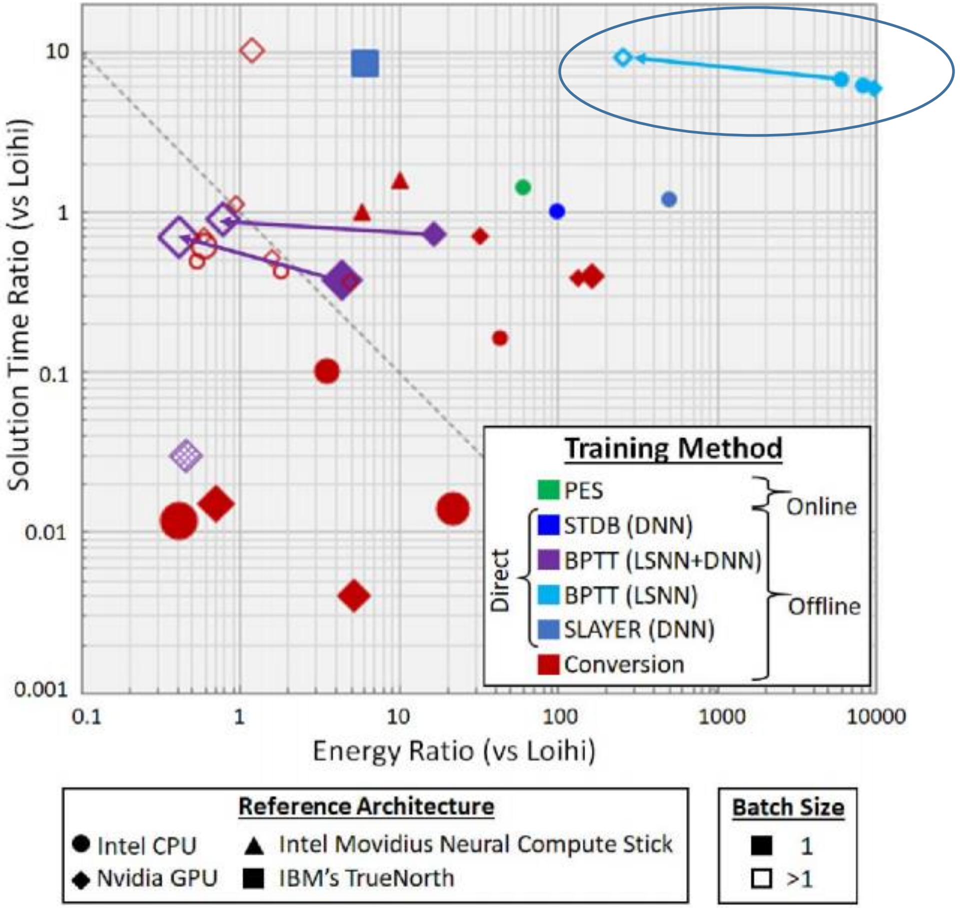
for i *in* $[1, N_{iterations}]$ **do**
  for *all active connections* k $(\theta_k \geq 0)$ **do**
    $\theta_k \leftarrow \theta_k - \eta \frac{\partial}{\partial \theta_k} E_{\mathbf{X},\mathbf{Y}^*}(\boldsymbol{\theta}) - \eta\alpha + \sqrt{2\eta T}\,\nu_k$;
    if $\theta_k < 0$ **then** set connection k dormant ;
  end
  while *number of active connections lower than* K **do**
    select a dormant connection k' with uniform probability and activate it;
    $\theta_{k'} \leftarrow 0$
  end
end

$$p^*(\boldsymbol{\theta}, \mathbf{c}) \;\propto\; p^*(\boldsymbol{\theta})\, \mathcal{C}(\boldsymbol{\theta}, \mathbf{c})\, p_{\mathcal{C}}(\mathbf{c}) \;,$$





Porting LSNN to neuromorphic hardware



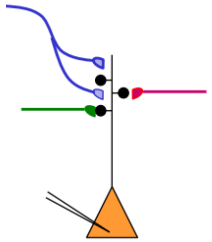
Intel Loihi (Digital)

Heidelberg Brain Scales (mixed digital and analog)

Spinnaker (Digital)

Davies, Mike, et al. "Advancing neuromorphic computing with Loihi: A survey of results and outlook." *Proceedings of the IEEE* 109.5 (2021): 911-934.

Part2. How does the brain learn?
Observed mechanism of synaptic plasticity

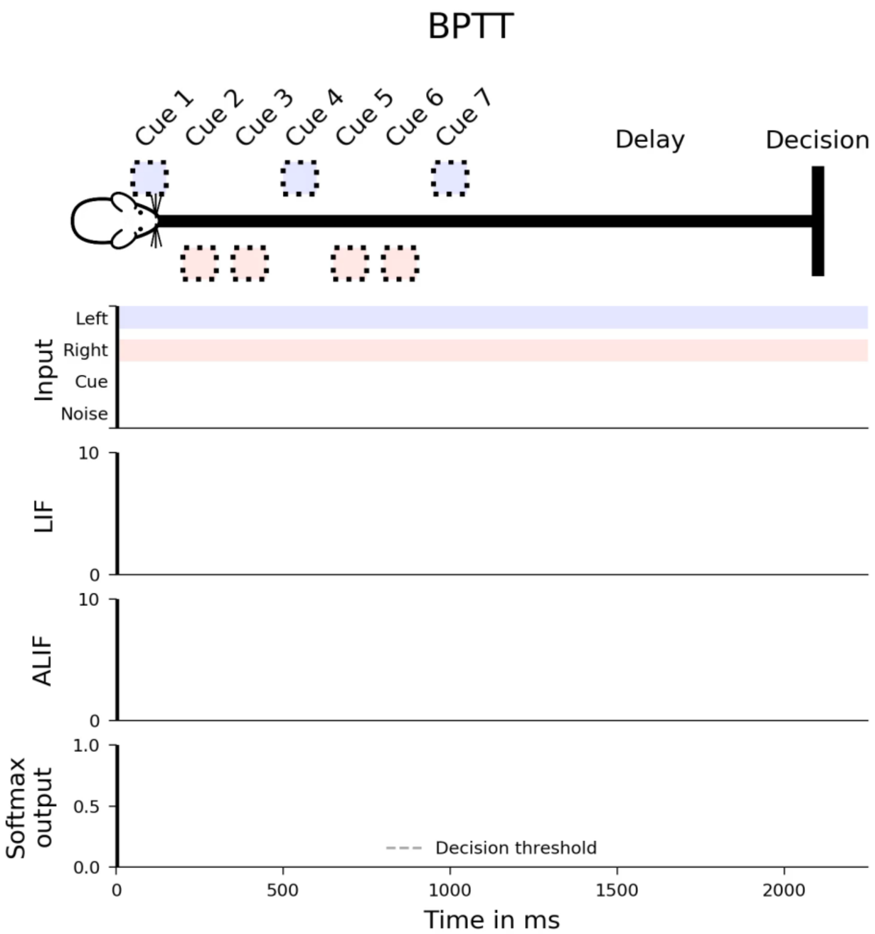


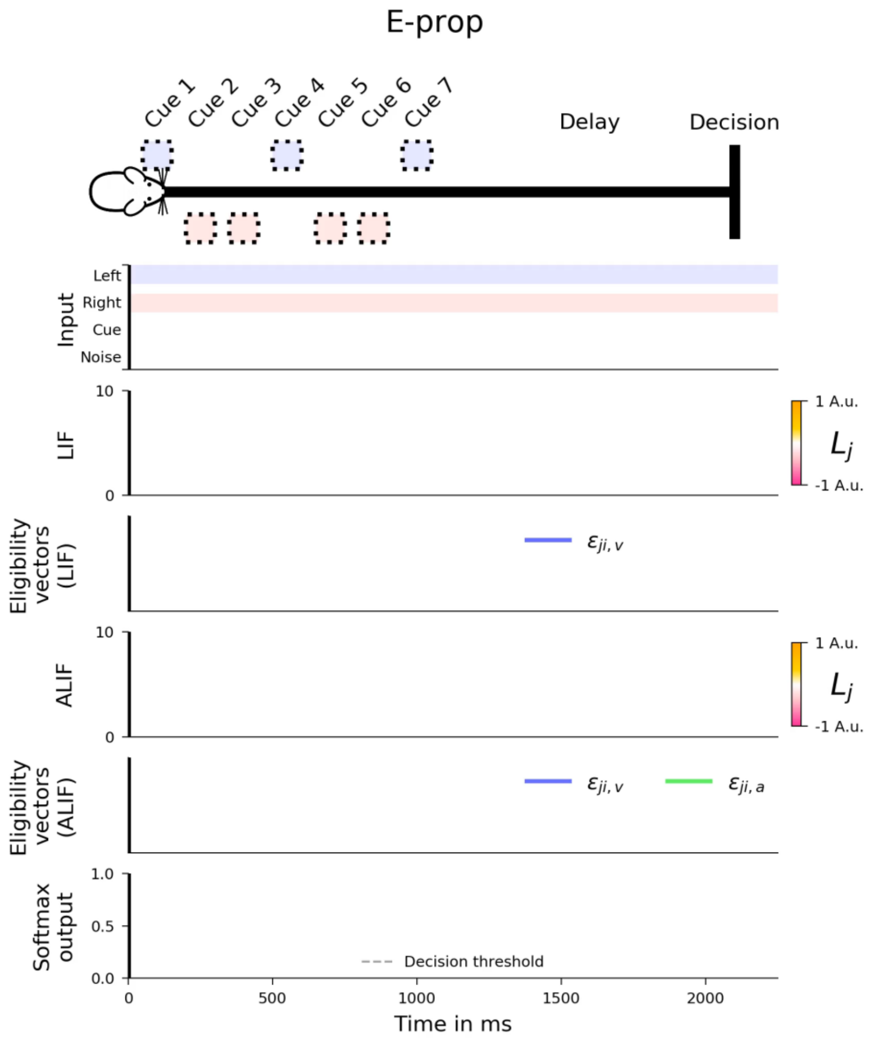
Review
W Gerstner, M Lehmann, V Liakoni, D Corneil, J Brea 2018

- We define as **eligibility trace**, a synaptic process that retain information about the history of local activity to make a potential change of synaptic efficacy

$$e_{ji}^t = f(z_i^{t-\Delta t} \cdots z_i^t, z_j^{t-\Delta t} \cdots z_j^t)$$

Credit assignment problem in **recurrent** dynamics



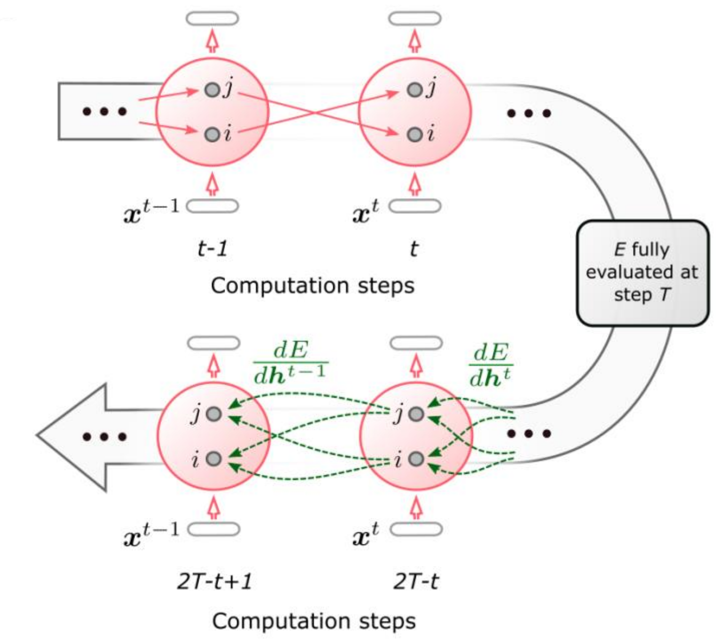


Eligibility propagation (e-prop)

How to compute gradients in recurrent neural networks: $\dfrac{dE}{dW_{ji}} = ?$

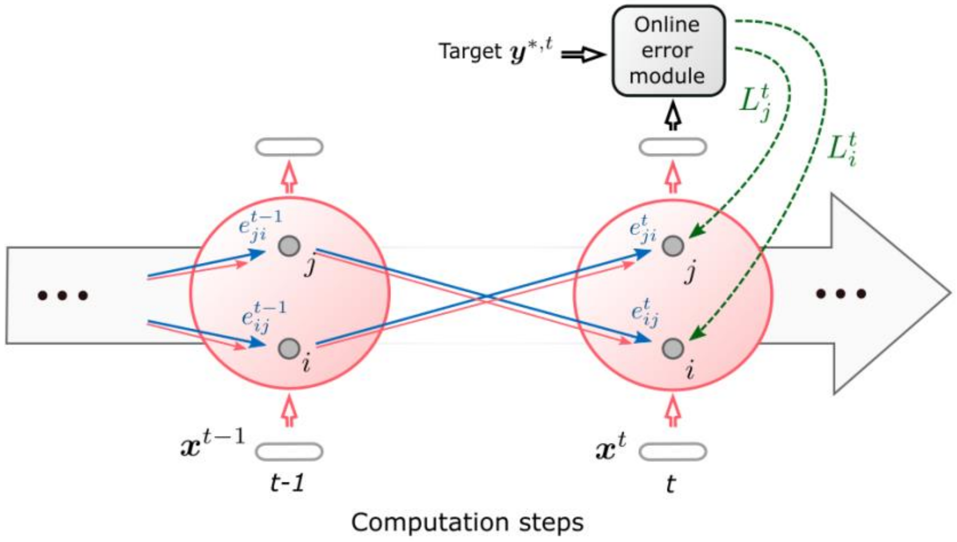
BPTT

Back-propragation through time



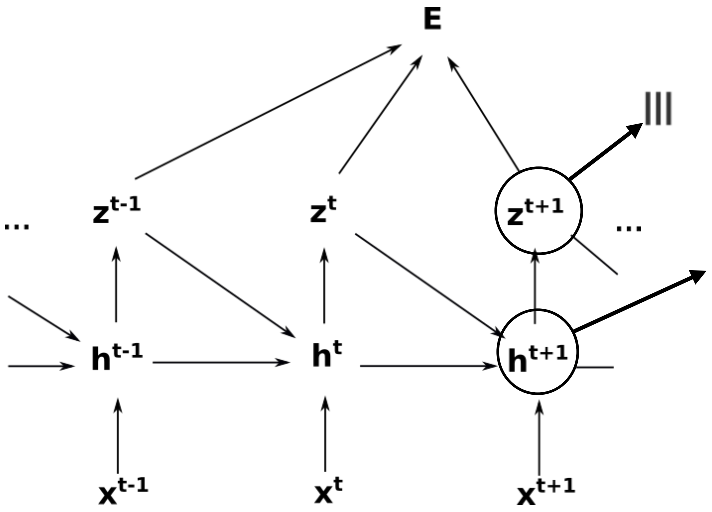
E-prop

Eligibility propagation



Three factorizations of the loss gradient: $\frac{dE}{dW_{ji}} = ?$

Many ways of applying the chain rule

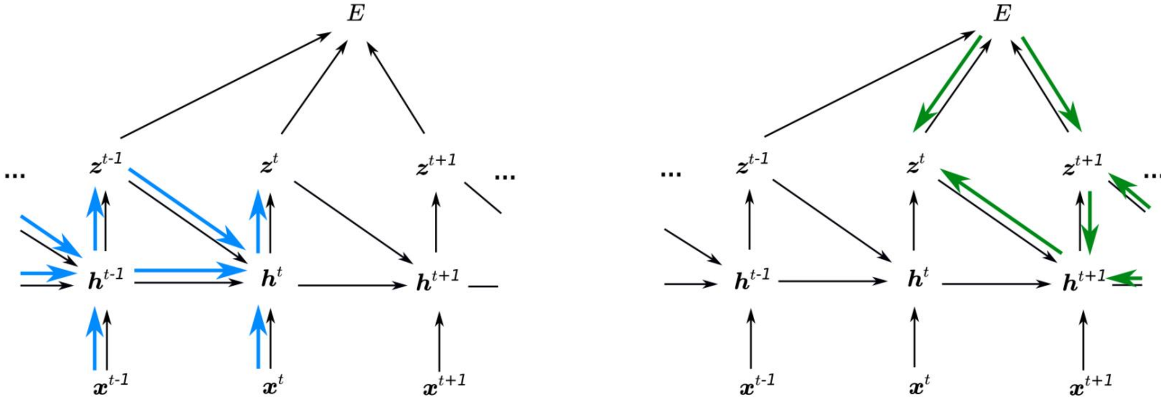


$$= \sum_t \frac{dE}{d\boldsymbol{h}_j^t} \frac{\partial \boldsymbol{h}_j^t}{\partial W_{ji}}$$

[1] Werbos (1990). Backpropagation through time: what it does and how to do it

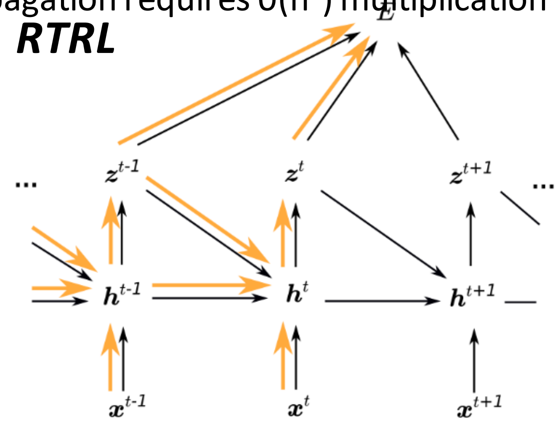
propagation requires O(n^2) multiplication and O($nT + n^2$) in memory

E-prop (a mixed forward-backward propagation)



[3] Bellec*, Scherr*, Subramoney, Hajek, Salaj, Legenstein, & Maass (2020)
A solution to the learning dilemma for recurrent networks of spiking neurons

RTRL



$$\frac{dE}{dW_{ji}} = \sum_t \sum_k \frac{\partial E}{\partial \boldsymbol{h}_k^t} \frac{d\boldsymbol{h}_k^t}{dW_{ji}}$$

[2] Williams & Zipser (1989). A learning algorithm for continually running fully RNNs

propagation requires O(n^4) multiplication and O(n^3) in memory

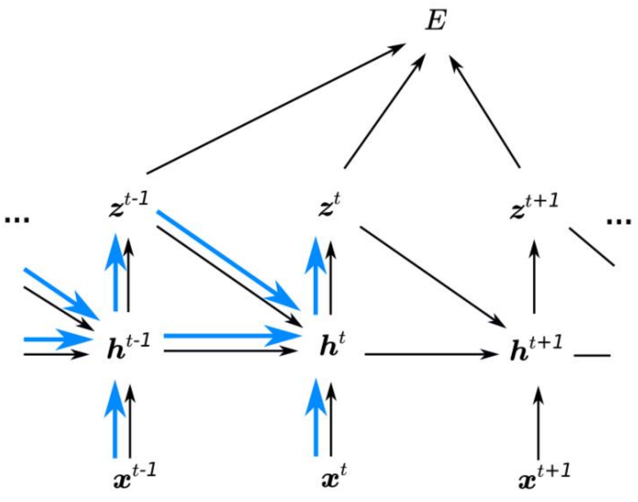
Eligibility traces require O(n^2) mult. O(n^2) in memory

The learning signal can be approximated O(nT + n2) -> O(n2)

$$\boxed{\frac{dE}{dW_{ji}} = \sum_t \frac{dE}{dz_j^t} \cdot \left[\frac{dz_j^t}{dW_{ji}}\right]_{\text{local}}}$$

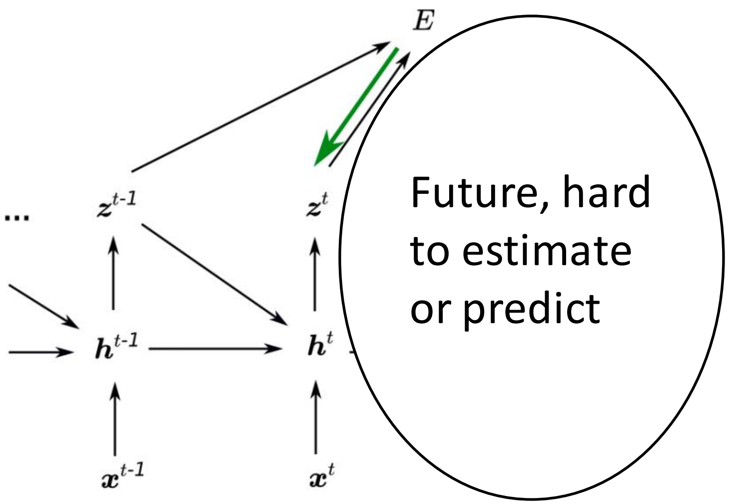
E-prop: a **neurocentric** factorization of RNN gradients (each term should be accessible locally)

Eligibility trace

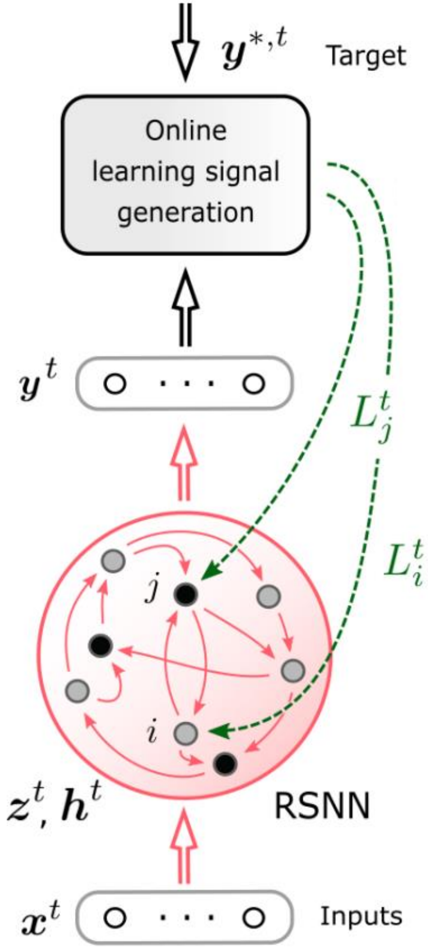


$$e_{ji}^t \overset{\text{def}}{=} \left[\frac{dz_j^t}{dW_{ji}}\right]_{\text{local}}$$

Learning signal



$$L_j^t \overset{\text{def}}{=} \frac{dE}{dz_j^t}$$



$$\boxed{\frac{dE}{dW_{ji}} = \sum_t \frac{dE}{dz_j^t} \cdot \left[\frac{dz_j^t}{dW_{ji}}\right]_{\text{local}}}$$

E-prop empirical success in simulations

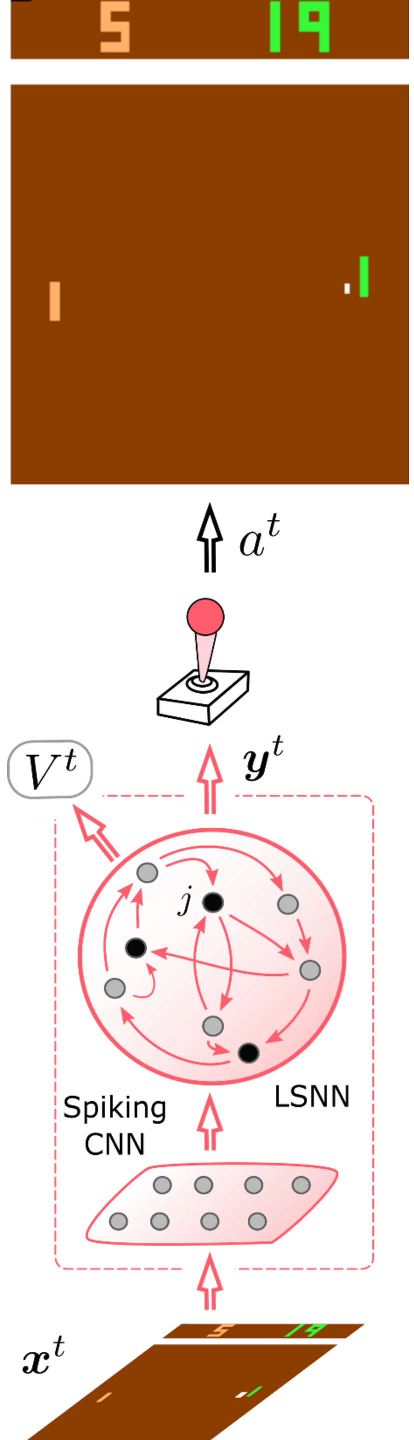
An approximation of the learning signal is required to do the computation fully locally.

E-prop (with this approximation) looses only a tiny bit of performance compared to BPTT on:

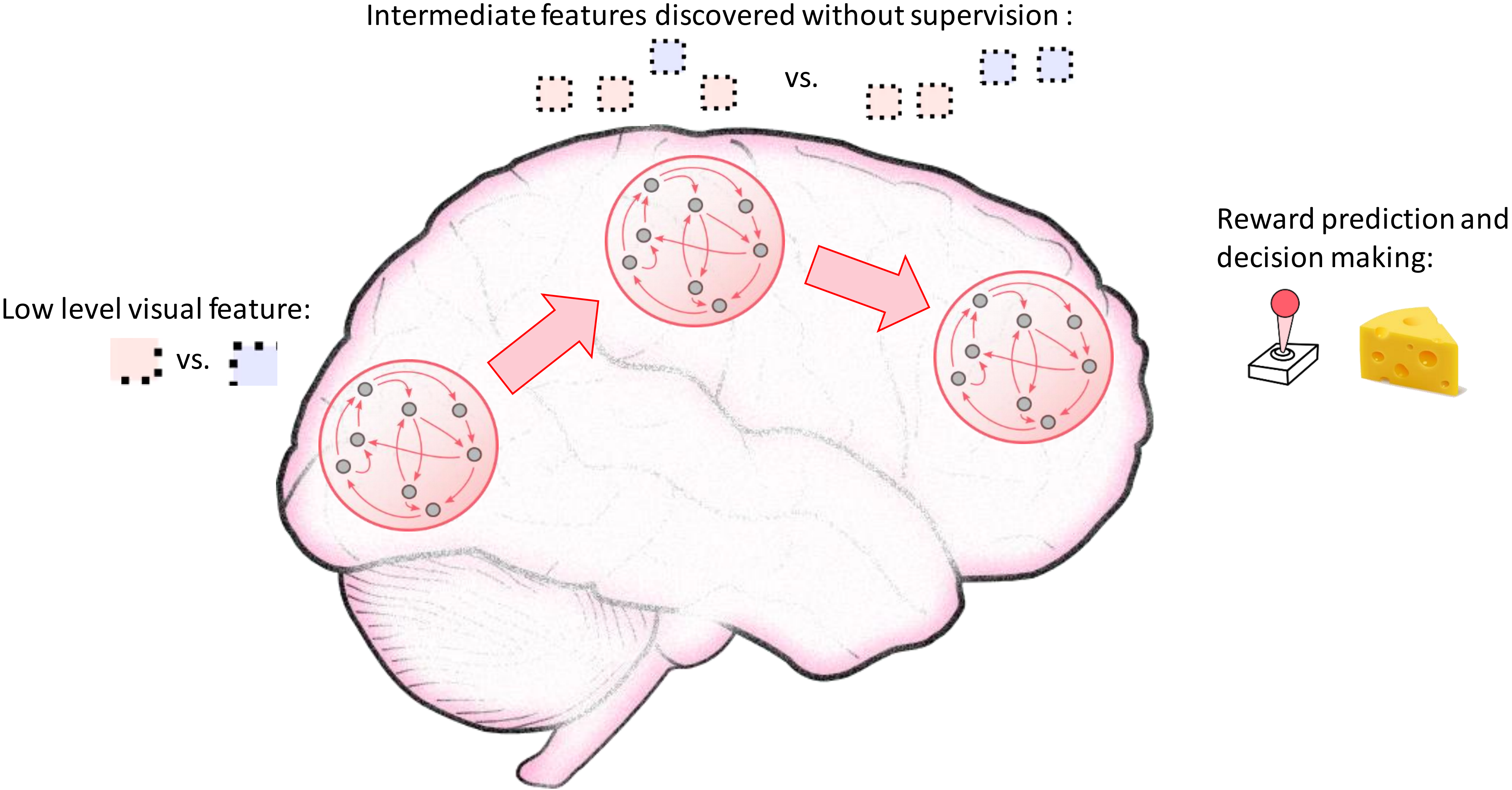
- speech processing [1] (see right),
- reinforcement learning [1] (ATARI games)
- natural language processing with Snap-1 [2].

Similar approximation for RNNs were discussed in [2,3,4], it 's hard to find a better **online approximation** of the loss gradient.

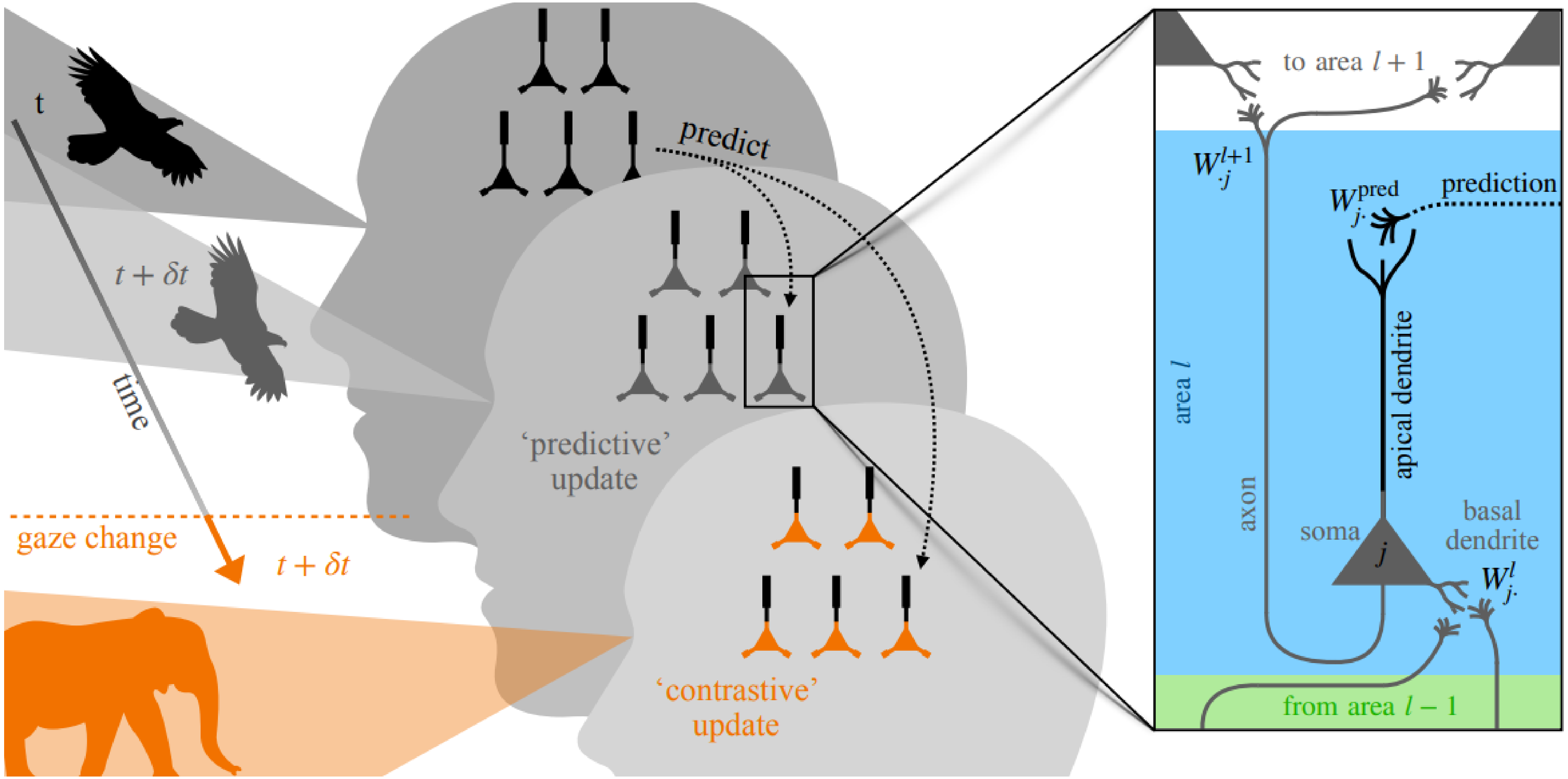
[1] Bellec*, Scherr*, Subramoney, Hajek, Salaj, Legenstein, & Maass (Nature comm. 2020)
A solution to the learning dilemma for recurrent networks of spiking neurons
[2] Menick, Elsen, Evci, Osindero, Simonyan, & Graves (ICLR 2021)
A Practical Sparse Approximation for Real Time Recurrent Learning
[3] James Murray (eLife 2019)
Local online learning in recurrent networks with random feedback
[4] Long short-term memory (1997)
S Hochreiter, J Schmidhuber



Part 3. What is the loss function E?
Is there a general principle for **learning representations**

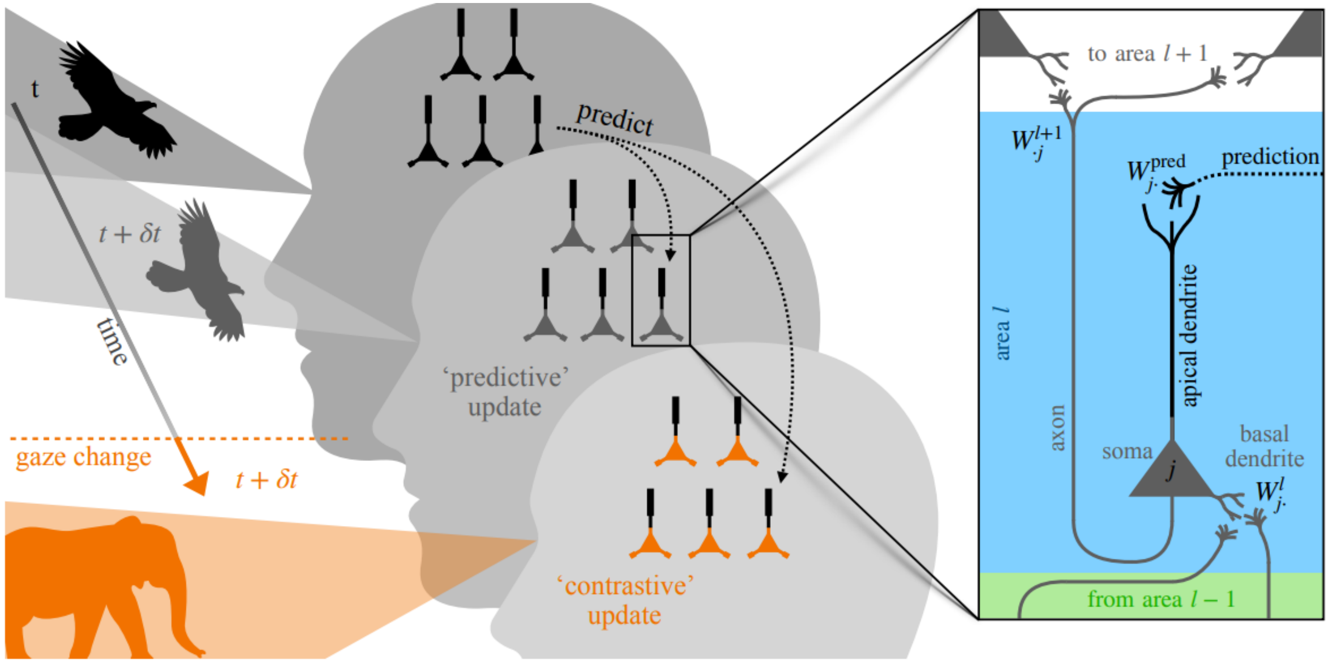


Predictions as a plausible principle for representation learning



Local plasticity rules can learn deep representations using self-supervised contrastive predictions
Illing, Ventura, Bellec*, Gerstner* (NeurIPS 2021)

Plausible plasticity rule with dendritic predictions

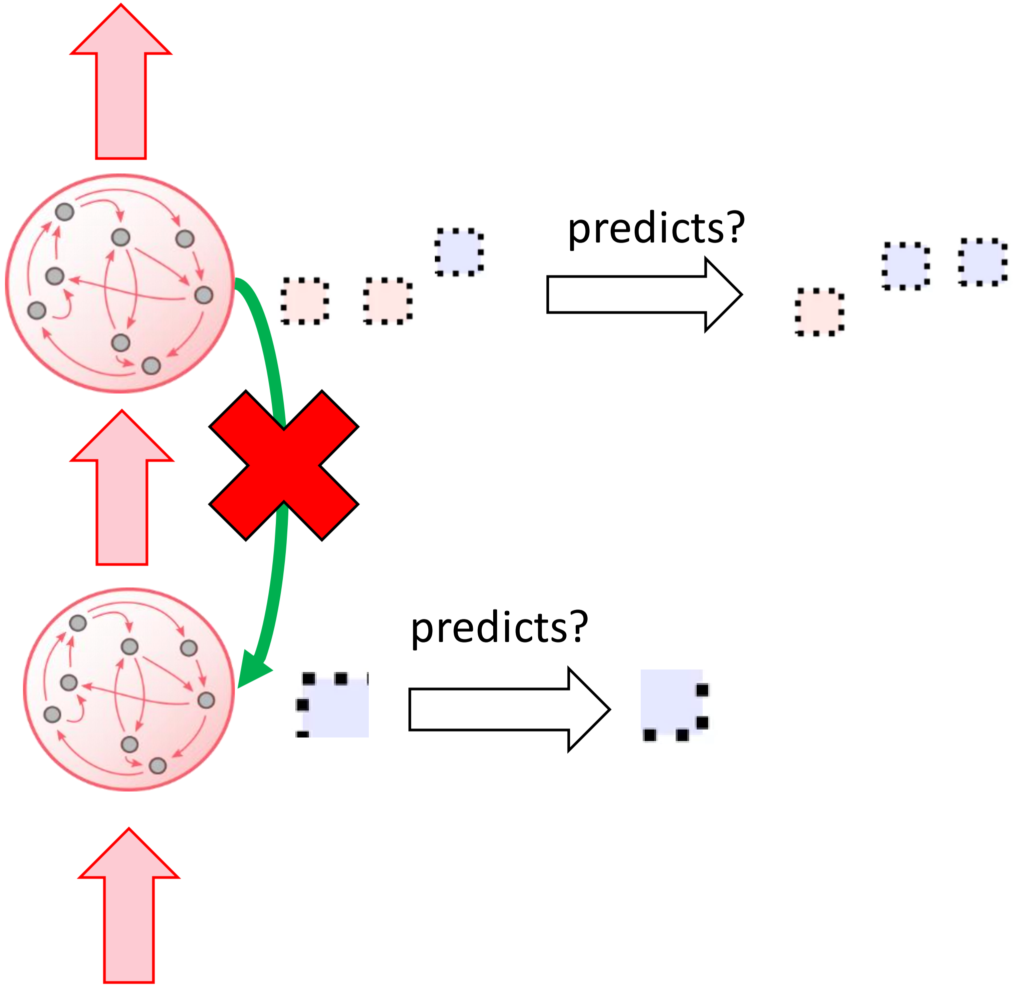


- **Predictive coding**: the cortex seems to constantly predict its own future activity
- Dentritic signals seem to have a **predictive** nature
- Plasticity pairing protocols involve **pre**, **post**-synpatic activity and a **3rd** factor

Local plasticity rules can learn deep representations using self-supervised contrastive predictions
Illing, Ventura, Bellec*, Gerstner* (very positive reviews, very likely acceptance at NeurIPS 2021)

$$\Delta W_{ji} \propto \underbrace{\text{modulators}}_{\text{broadcast factors}} \cdot \underbrace{(\boldsymbol{W}^{\text{pred}}\boldsymbol{c}^{t_1})_j}_{\text{dentritic prediction}} \cdot \underbrace{\text{post}_j^{t_2} \cdot \text{pre}_i^{t_2}}_{\text{local-activity}}$$

CLAPP: Contrastive, Local And Predictive Plasticity

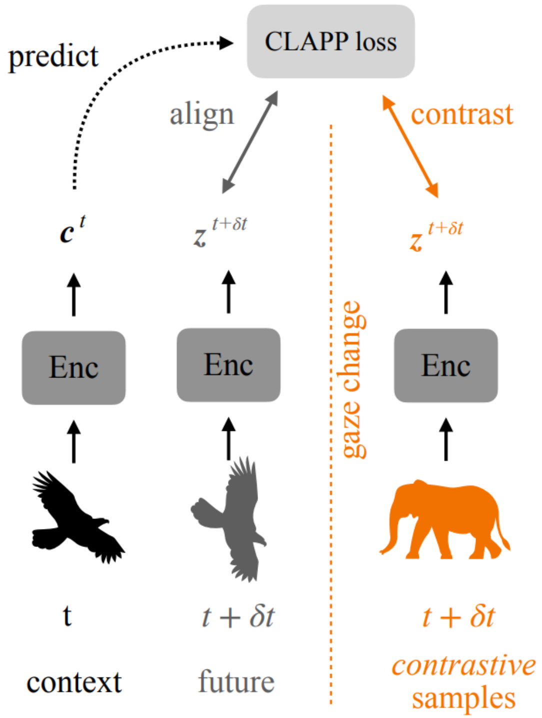


- Each layer minimizes a loss function to **predict its own future**
- **Only** requires the **activity of other neurons** is the (same) layer
- no other feedback or dense learning signal necessary

[1] Representation Learning with Contrastive Predictive Coding
Aaron van den Oord, Yazhe Li, Oriol Vinyals

[2] Putting An End to End-to-End: Gradient-Isolated Learning of Representations
Sindy Löwe, Peter O'Connor, Bastiaan S. Veeling

CLAPP: Constrastive, Local and Predictive Plasticity rule



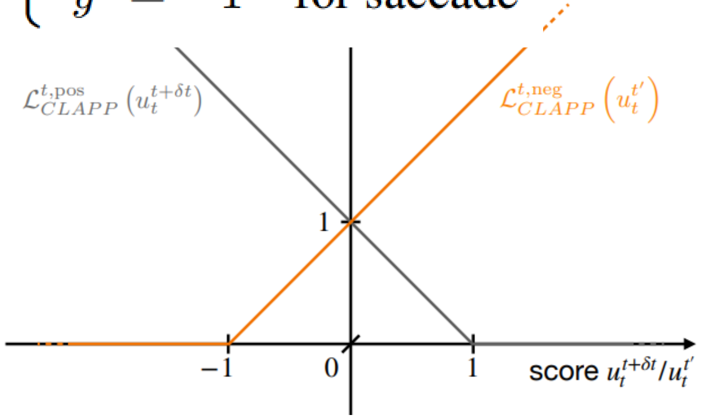
The loss function of CLAPP formalizes the **binary classification**:

Fixation vs. **Saccade** (gaze change towards the elephant)

If fixation, $u_t^{t+\delta t} = \boldsymbol{z}^{t+\delta t} \boldsymbol{W}^{\text{pred}} \boldsymbol{z}^t$ should be higher than 1

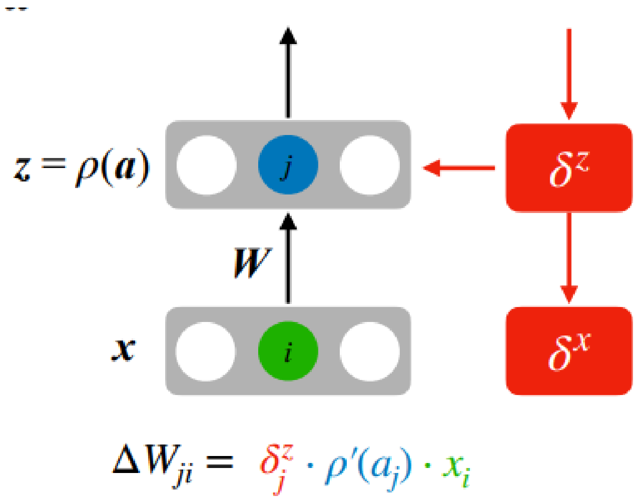
If saccade, $u_t^{t+\delta t}$ should be lower than -1

$$\mathcal{L}_{CLAPP}^t = \max\left(0, 1 - y^t \cdot u_t^{t+\delta t}\right) \quad \text{with} \quad \begin{cases} y^t = +1 & \text{for fixation} \\ y^t = -1 & \text{for saccade} \end{cases}$$



$$\frac{\partial \mathcal{L}_{CLAPP}^t}{\partial W_{ji}} = \pm(\boldsymbol{W}^{\text{pred}} \boldsymbol{z}^t)_j \, \rho'(a_j^{t+\delta t}) \, x_i^{t+\delta t}$$

CLAPP as a replacement for back-prop:



If we consider a neuron such that $\boldsymbol{z}^t = \rho(\boldsymbol{a}^t)$ with $\boldsymbol{a}^t = \boldsymbol{W}\boldsymbol{x}^t$

We find a weight update gated by γ^{τ} (it is 0 when the Hinge loss is saturated)

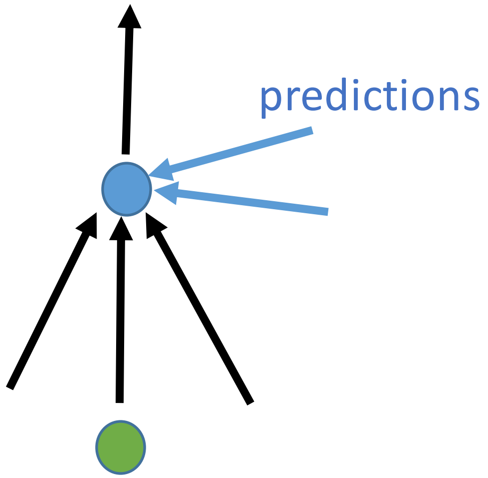
(for instance $\tau = t + \delta t$)

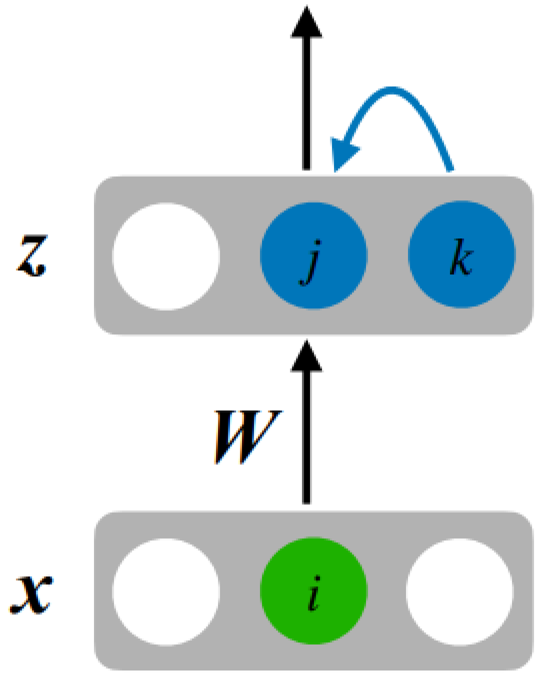
$$\Delta W_{ji}^{\tau} = \underbrace{\gamma^{\tau}}_{\text{global 3rd}} \underbrace{\left(\boldsymbol{W}^{\mathrm{pred}} \boldsymbol{z}^{t}\right)_j}_{\text{predictions}} \underbrace{\rho'(a_j^{\tau})}_{\text{post}} \underbrace{x_i^{\tau}}_{\text{pre}}$$

To align z_j^{τ} better with the prediction

$$\Delta W_{ji}^{\tau} = \gamma^{\tau} \left(\boldsymbol{W}^{\mathrm{pred},\top} \boldsymbol{z}^{\tau}\right)_j \rho'(a_j^{t})\, x_i^{t}$$

To improve the prediction made by z_j^t



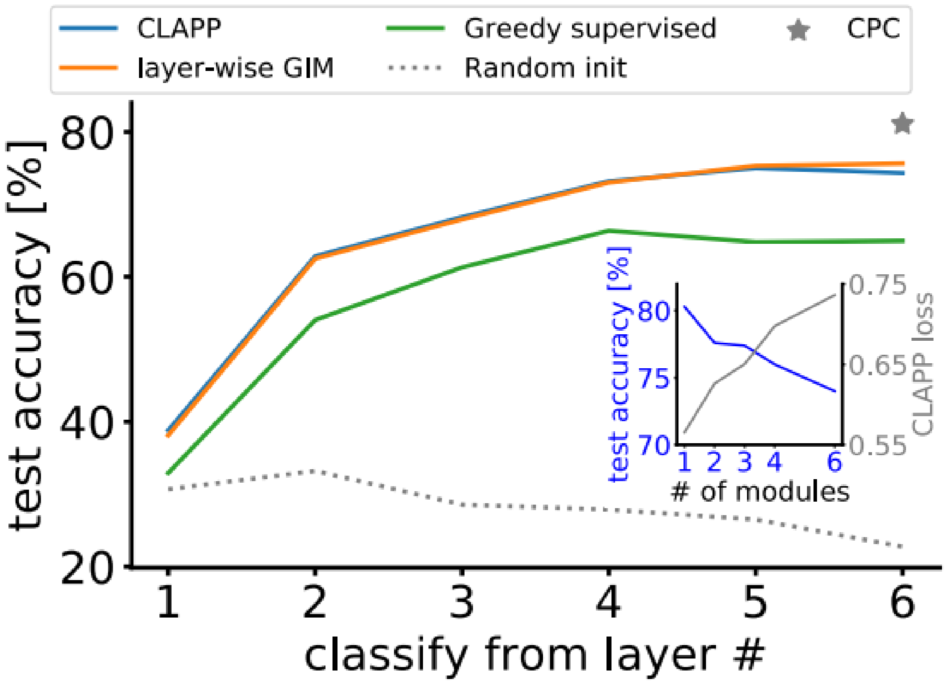


CLAPP trains deep CNNs efficiently even-though **no information is transmitted backward**

STL-10 unsupervised benchmark:

500 training images

100,000 unlabelled images



To evaluate the “richness” of the representation:

After learning with **unlabelled data** we train a linear classifier on a usual supervised object recognition task.

Performance **increases with depth** (not trivial).

Outperforms supervised learning rule by 9%.

No apparent decrease of performance compared to layer wise CPC

Porting E-prop and CLAPP to neuromorphic hardware

Stay tuned...

Intel Loihi (Digital)

Heidelberg Brain Scales (mixed digital and analog)

Spinnaker (Digital)

Part 1. and 2. TU GRAZ

F. Scherr*

D. Salaj

E. Hajek

A. Subramoney

R. Legenstein

W. Maass

[1] Bellec*, Scherr*, Subramoney, Hajek, Salaj, Legenstein, & Maass (Nature comm. 2020)
A solution to the learning dilemma for recurrent networks of spiking neurons

Part 3. EPFL

B. Illing

W. Gerstner

[2] Towards truly local gradients with CLAPP: Contrastive, Local And Predictive Plasticity (arxiv 2020)
Bernd Illing, Wulfram Gerstner, Guillaume Bellec