Neural dynamics and learning in Spiking Neural Networks

Angeliki Pantazi

Manager, Neuromorphic Computing & I/O Links

IBM Research - Zurich

IBM Research – Zurich

Established in 1956

45+ different nationalities

Open Collaboration:

- Horizon2020: 50+ funded projects and 500+ partners

Two Nobel Prizes:

- 1986: Nobel Prize in Physics for the invention of the scanning t unneling microscope by Heinrich Rohrer and Gerd K. Binnig
- 1987: Nobel Prize in Physics for the discovery of high-temperat ure superconductivity by K. Alex Müller and J. Georg Bednorz

European Physical Society Historic Site

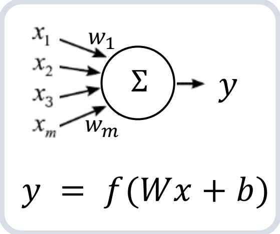
Binnig and Rohrer Nanotechnology Centre (Public Private Partnership with ETH Zürich and EMPA)

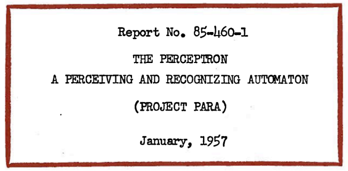
The Evolution of Neural Networks

Simulation of the structure

Artificial Neural Networks (ANNs)

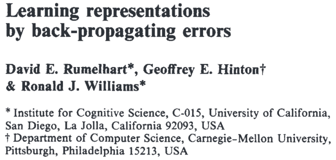
1943
McCulloch and Pitts present an Artificial Neuron




Report No. 85-460-1
THE PERCEPTRON
A PERCEIVING AND RECOGNIZING AUTOMATON
(PROJECT PARA)
January, 1957

1957
Rosenblatt’s perceptron (1L ANN)

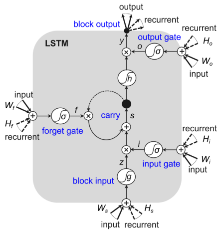
1986
Hinton – learning representations with BP (multi-layer ANN)


Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

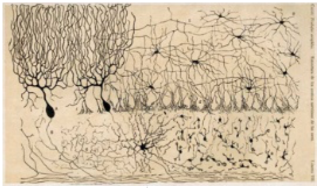
* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

1997
Schmidhuber LSTM

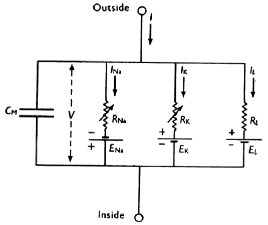


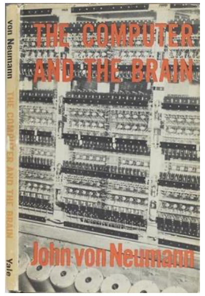
2012 - today
GPUs for ANNs, large-scale processing systems deep learning “revolution”

1906
Ramon y Cajal - the structure of the nervous system – Nobel Prize



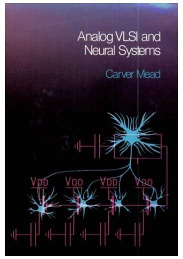
1952
Hodgkin-Huxley model of spiking neural dynamics – Nobel Prize

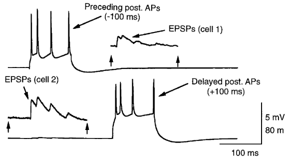




1956
von Neumann postulates SNN-based architectures

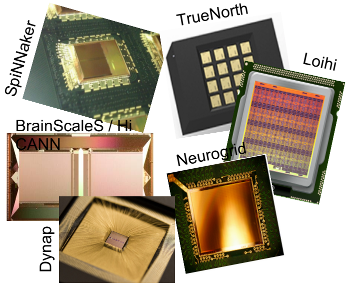
1980s
C.Mead @ Caltech – Neuromorphic Engineering





1997
Markram STDP learning

2014 - today
Neuromorphic accelerators

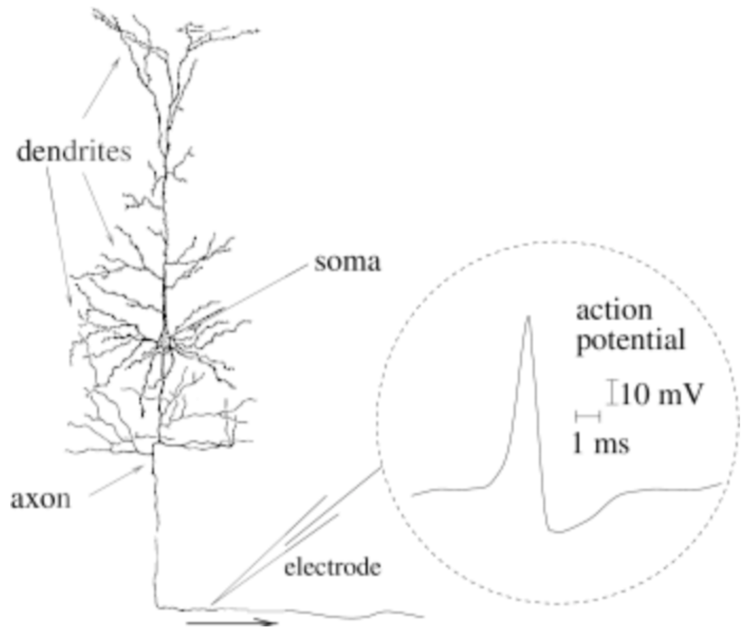


Bio-realistic emulation of the dynamics

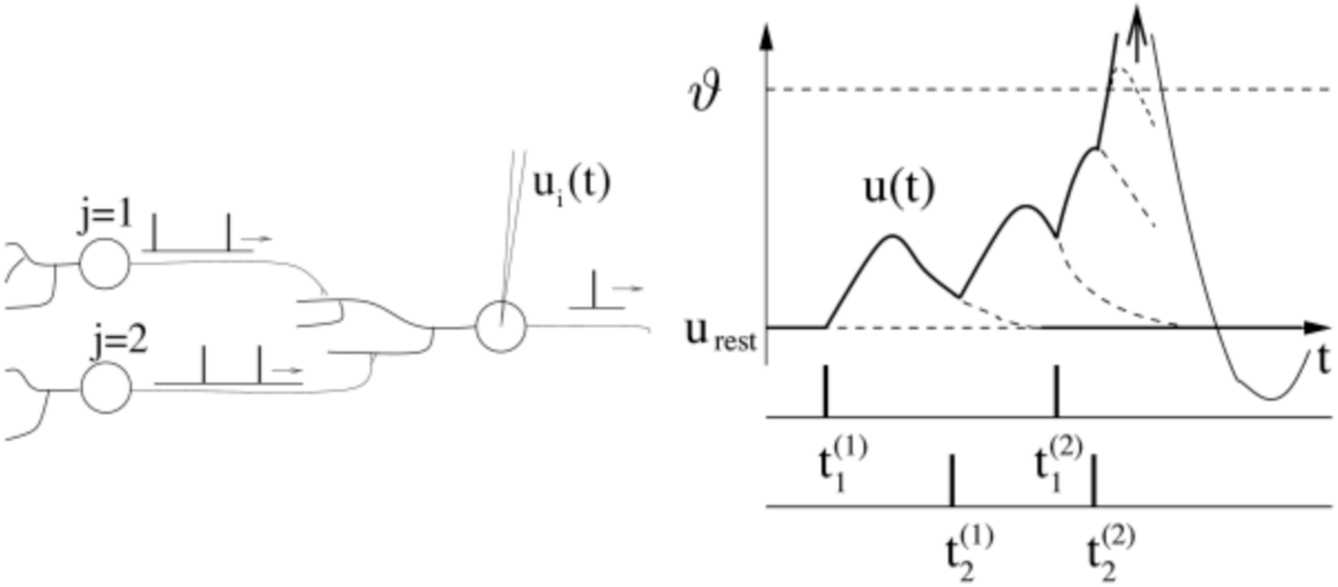
Spiking Neural Networks (SNNs)

Spiking Neurons: Neuroscience Concepts

Spiking Neuron

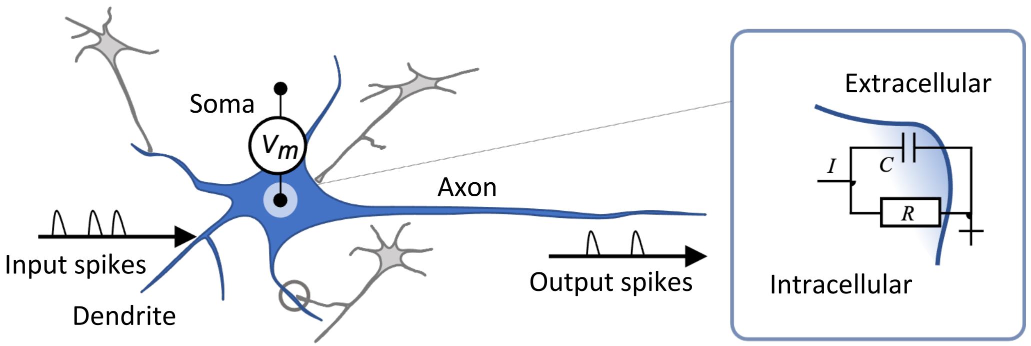


Leaky Integrate-and-Fire (LIF) dynamics



W. Gerstner, W. M. Kistler, R. Naud and L. Paninski, Neuronal Dynamics, *Cambridge University Press*, 2014

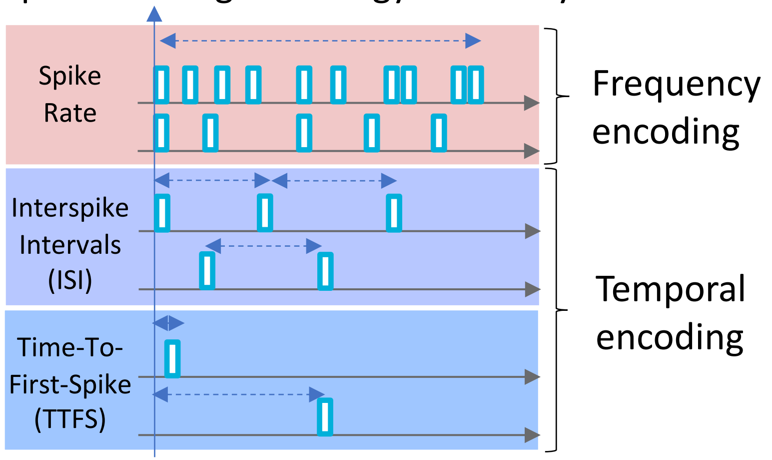
Spiking Neural Networks



$$\tau \frac{\mathrm{d}V_m(t)}{\mathrm{d}t} = -V_m(t) + RI(t)$$

spike on $V_m > V_{\mathrm{th}}$

Input encoding for energy efficiency



Key features

– Neurons communicate with spikes, encoding timing information

– Independent firing of neurons and asynchronous communication

– Stateful and adaptive neuronal dynamics

– Short and long-term synaptic plasticity

– Local and event-based learning

Low latency

High energy efficiency

Novel AI features

Memory & energy efficiency

Challenges

– Lack of a general training approach limits the accuracy, scalability and applicability

– Demonstrate how SNN unique features can substantially impact AI applications

– Design energy efficient neuromorphic accelerators

Approaches for Training Deep SNNs

- **Biologically inspired local learning rules**
  - Spike-timing dependent learning
- **Conversion from ANNs**
  - Porting the weights trained in ANNs to SNNs
- **Training constrained ANN networks**
  - Before conversion, constraints are used to model the properties of spiking neurons
- **Supervised learning directly on SNNs**
  - Training using variations of error backpropagation

M. Pfeiffer and T. Pfeil, *Front. Neurosci. 12:774, 2018. doi: 10.3389/fnins.2018.00774*

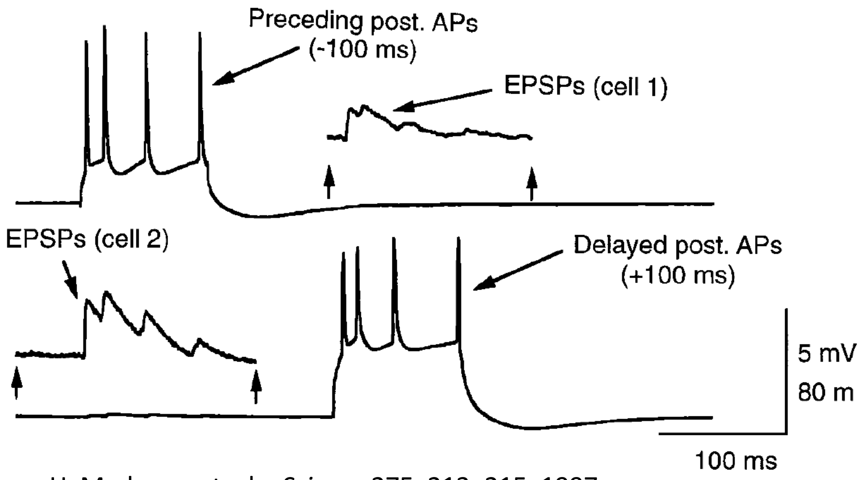
Recent developments bridge the ANN and SNN worlds!

Training of Deep SNNs: Local learning rules

- **Methodology**
  - Biologically inspired local learning such as Spike Timing Dependent Plasticity (STDP)
  - The synaptic weight adjustments depend on the timing between input and output spikes

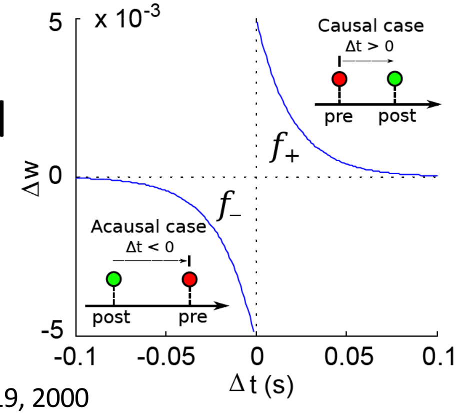
- **Advantages**
  - Energy efficient learning, suitable for hardware implementations

- **Limitations**
  - Reduced accuracy in complex problems and limited scaling to deep SNNs



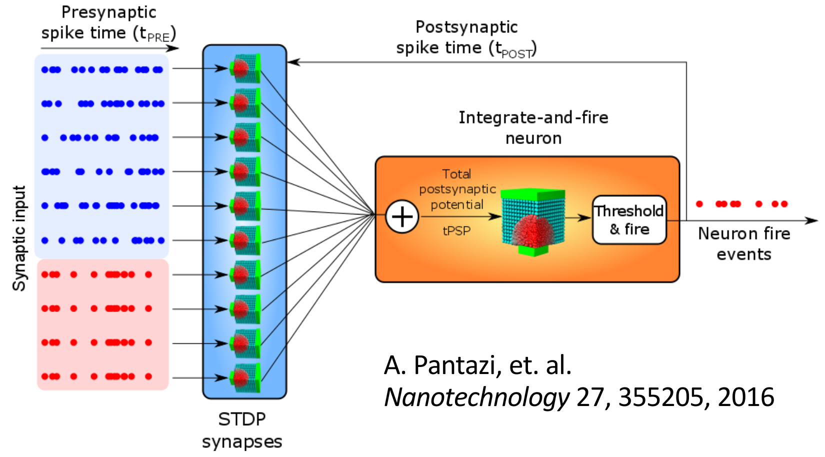
H. Markram, et. al., *Science* 275, 213–215, 1997

Typical STDP



S. Song, et.al. *Nat. Neurosci*. 3:919, 2000

Correlation detection



A. Pantazi, et. al.
Nanotechnology 27, 355205, 2016

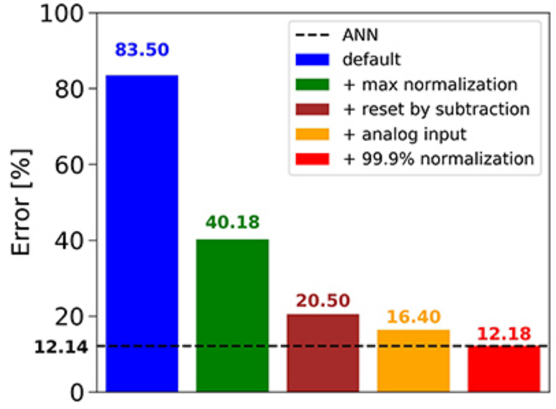
Training of Deep SNNs: Conversion from ANNs

- **Methodology**
  - A pre-trained ANN is converted into an SNN by adapting the synaptic weights
  - Activations of ANN neurons are translated into firing rates of spiking neurons
  - Spiking equivalents of ANN operators were introduced for CNN architectures

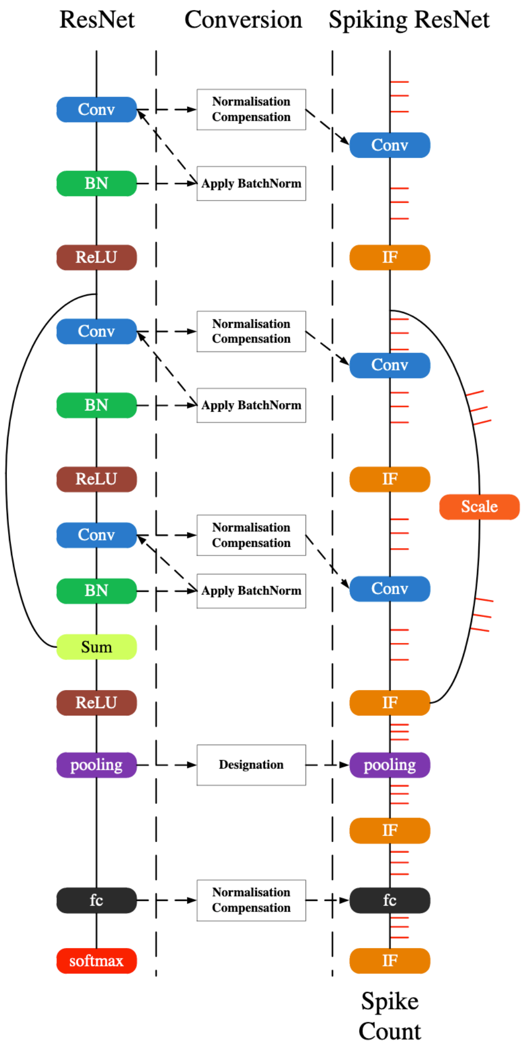
- **Advantages**
  - Deep learning framework can be exploited to train SNNs

- **Limitations**
  - Several ANN functionalities are difficult to realize in the spiking domain, such as batch normalization

SNN error rate for CIFAR-10



B. Rueckauer, et. al. *Front. Neurosci.* 11:682,00682, 2017



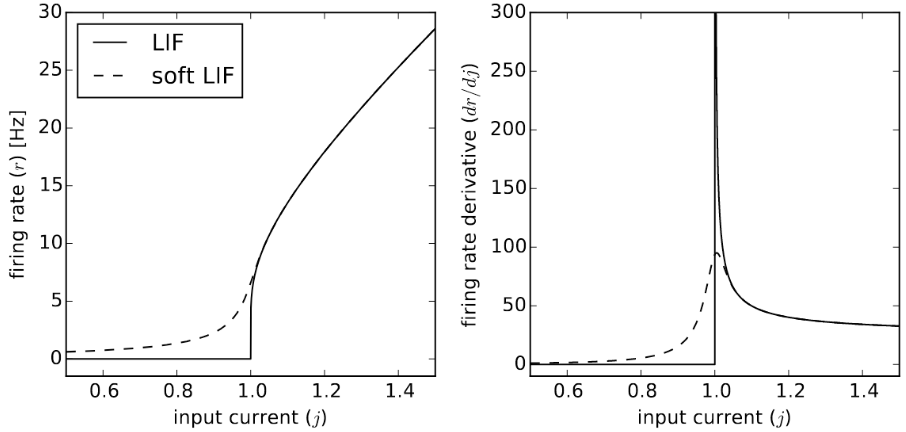
Y. Hu, et. al. *arXiv*:1805.01352, 2018

Training of Deep SNNs: Training constrained ANN networks

- **Methodology**
  - Constraints are included in the ANN training to capture the properties of the spiking neurons
  - After training, the parameters of the constrained ANN are used as parameters of the SNN

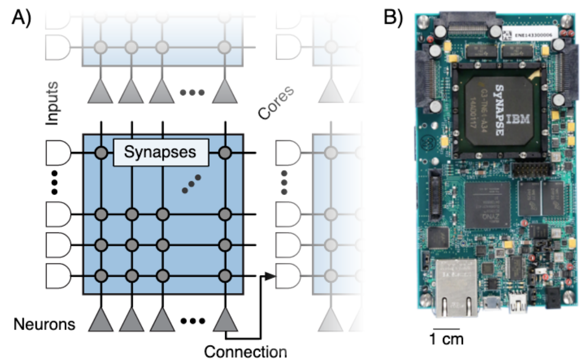
- **Advantages**
  - ANN training includes the characteristics of the SNN → results in higher accuracies

- **Limitations**
  - Training requires the transformation of the non-differentiable spiking neuron models



E. Hunsberger, and C. Eliasmith, *arXiv*:1611.05141, 2016

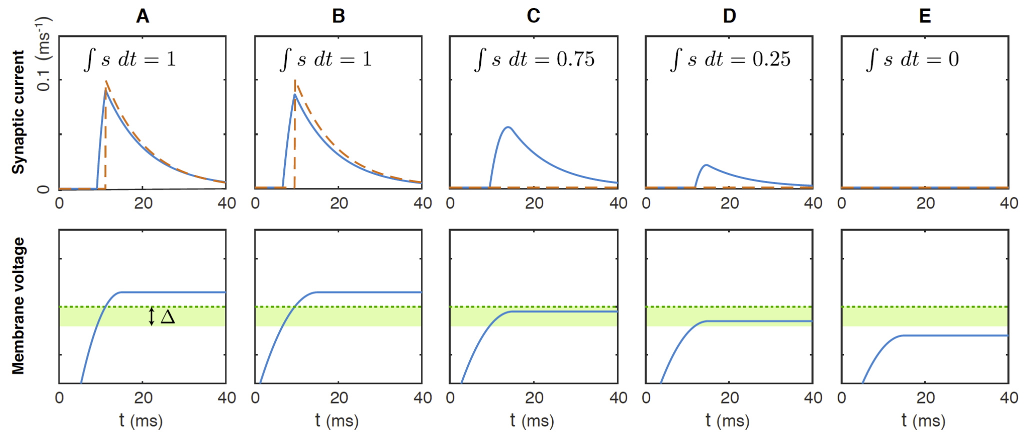
Hardware demonstration with TrueNorth



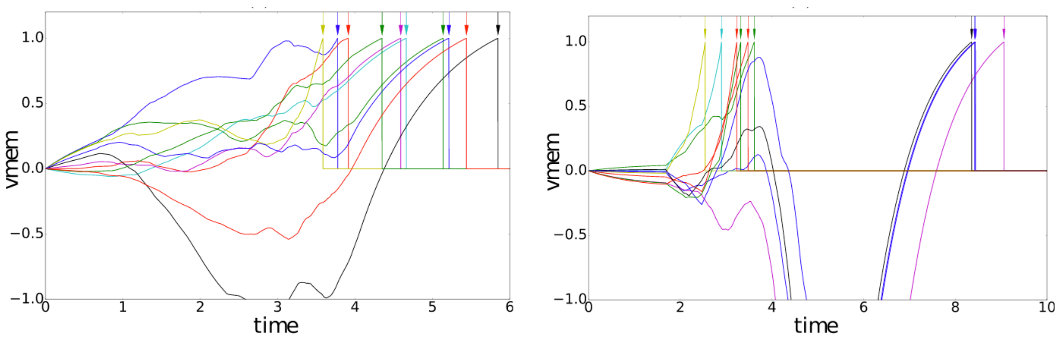
S. K. Esser et. al. *PNAS* 113 (41) 11441-11446, 2016

Training of Deep SNNs: Supervised learning directly on SNNs

- **Methodology**
  - Introduce supervised training directly on spiking neurons
  - Typically, utilizing variants of backpropagation to train the SNNs
- **Advantages**
  - Training directly on the spiking neurons and on temporal spiking patterns
- **Limitations**
  - Find a differentiable alternative construct to enable backpropagation training



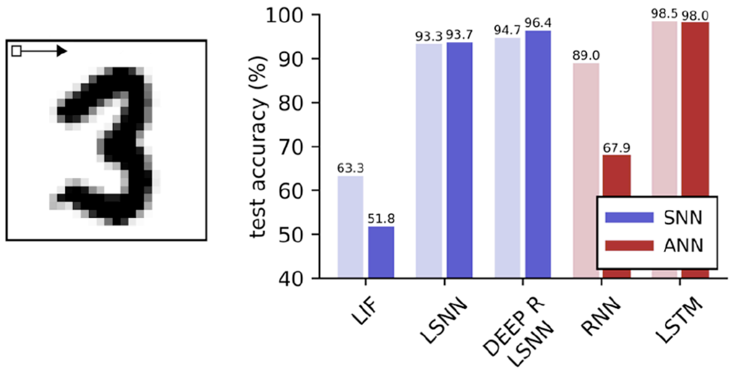
D. Huh and T. J. Sejnowski, *NeurIPS* 2018



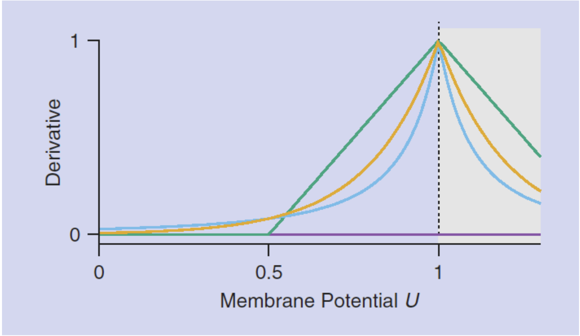
H. Mostafa, *IEEE Trans. Neural Netw. Learn. Syst.* 29, 3227–3235, 2018

Training of Deep SNNs: Bridging ANN and SNN worlds

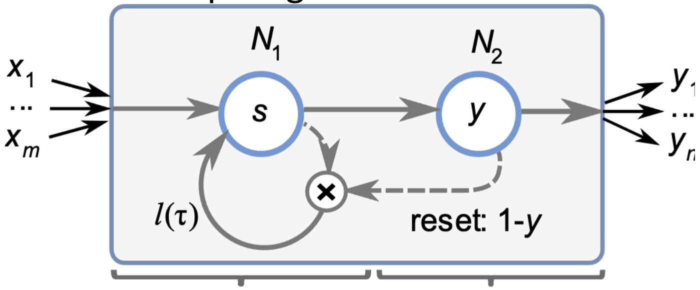
- **Methodology**
  - Transfer the SNN dynamics to RNNs and train with BPTT
  - For the non-differentiable elements, a pseudo-derivative is used in the backward pass
- **Advantages**
  - BPTT training enables high accuracy in deep SNNs
  - Scalable deep and recurrent SNN architectures
- **Limitations**
  - Possible dependence of the performance on the selection of pseudo-derivative



G. Bellec, et. al., *NeurIPS*, 2018



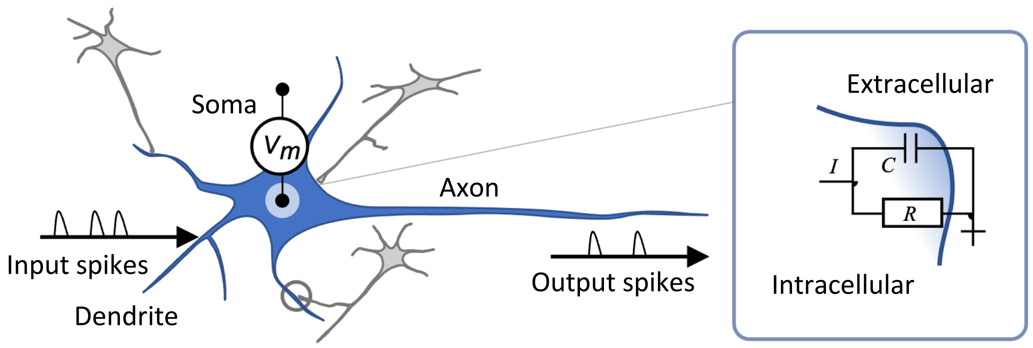
E. O. Neftci, et.al.
IEEE Signal Process. Mag., 2019



S. Wozniak, et. al., *arXiv*, 2018.
S. Wozniak, et. al., *Nat. Mach. Intell.*, 2020

Training of Deep SNNs: Neuro-inspired units - SNUs

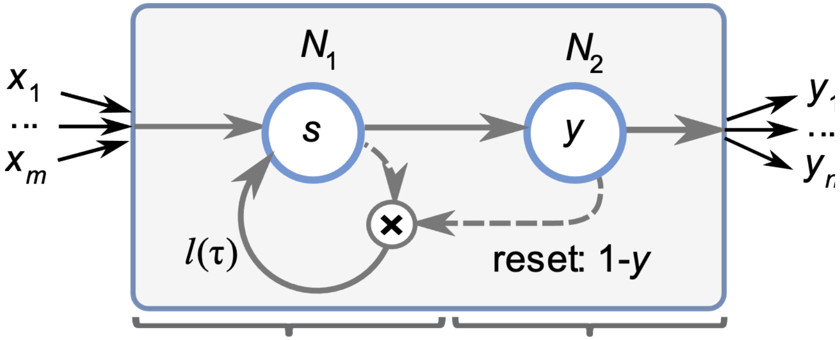
Biology



$$\tau \frac{\mathrm{d}V_m(t)}{\mathrm{d}t} = -V_m(t) + RI(t)$$

spike on
$V_m > V_{\mathrm{th}}$

Deep Learning



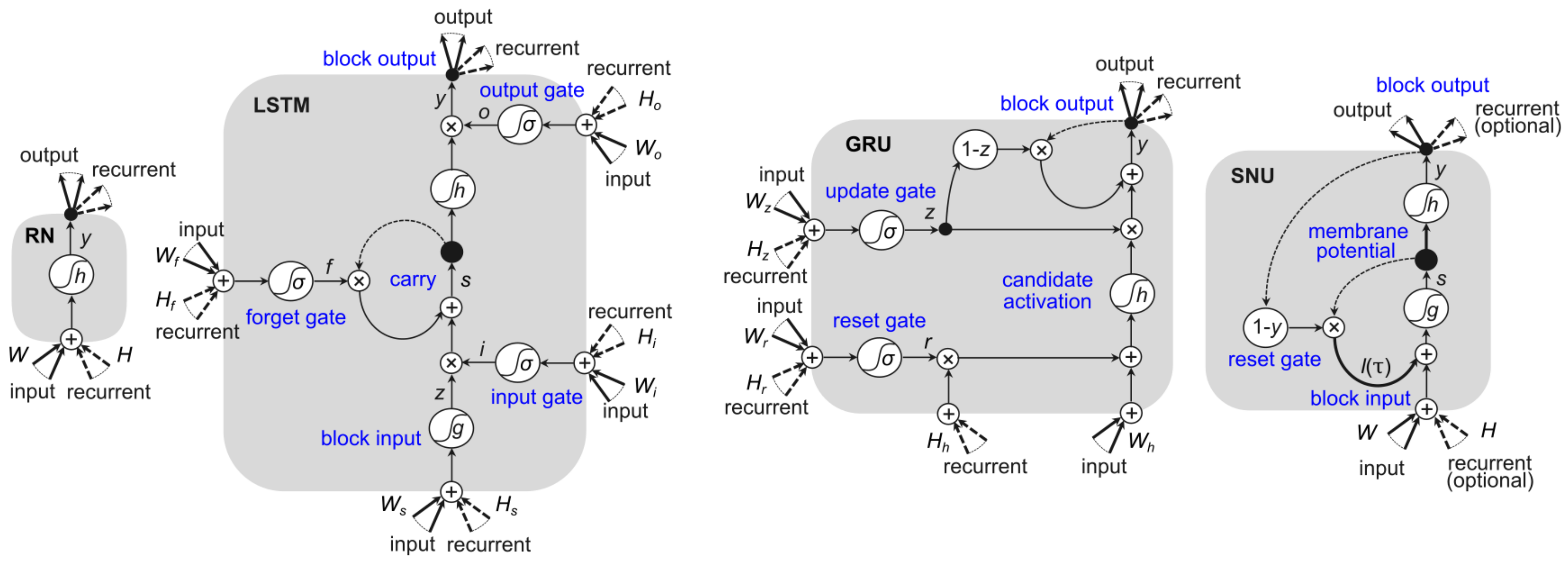
$$s_t = g\left(Wx_t + l(\tau)\odot s_{t-1}\odot(1 - y_{t-1})\right)$$
$$y_t = h(s_t + b)$$

S. Woźniak et al., *Nature Machine Intelligence* 325–336, 2020

- **SNUs enable modelling of SNNs in deep learning frameworks and training with BPTT**
- **Soft SNU(sSNU) variant introduces novel temporal dynamics into ANNs**

Significant step towards exploiting the computational efficiency of SNNs

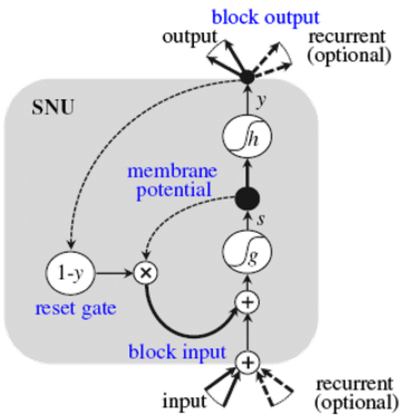
SNUs compared with RNNs, LSTMs and GRUs



- **Structural similarities, i.e. with LSTM's carry and GRU's internal connection to the output**
- **Qualitatively different dynamics than these of existing RNN units**
- **SNUs are similarly robust to vanishing gradients as LSTMs**

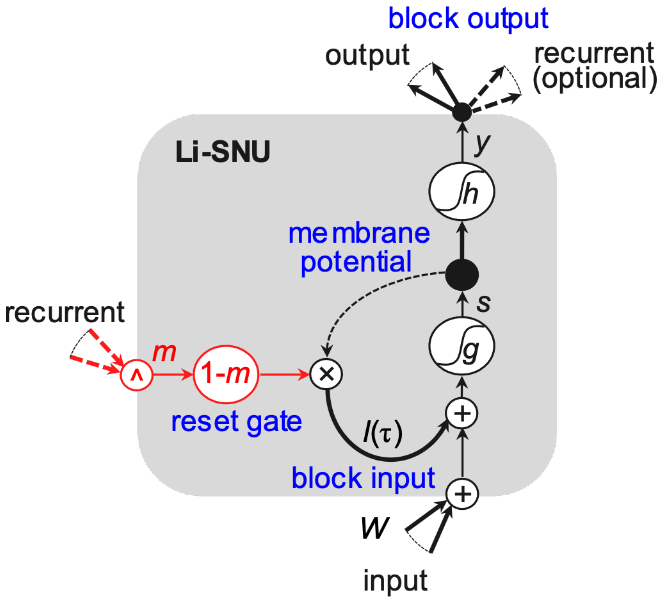
Biologically-inspired SNU extensions

SNU



$$s_t = g(Wx_t + l(\tau) \odot s_{t-1} \odot (1 - y_{t-1}))$$
$$y_t = h(s_t + b)$$

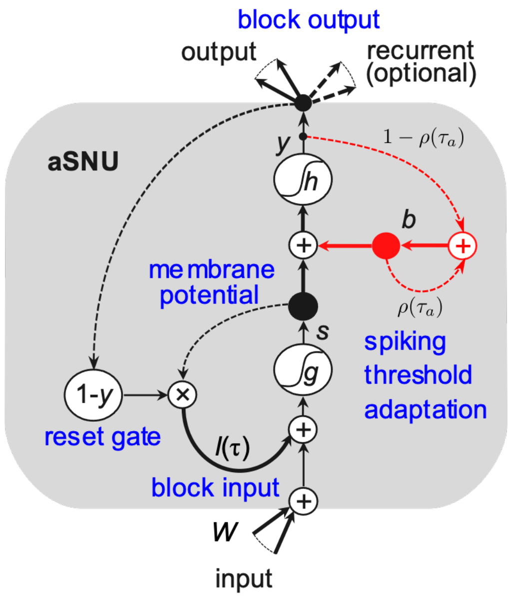
Lateral Inhibition SNU (LI-SNU)



$$s_t = g(Wx_t + l(\tau) \odot s_{t-1} \odot (1 - \max(y_{t-1})))$$

$$y_t = h(s_t + b)$$

Adaptive SNU (a-SNU)



$$s_t = g(Wx_t + l(\tau) \odot s_{t-1} \odot (1 - y_{t-1}))$$

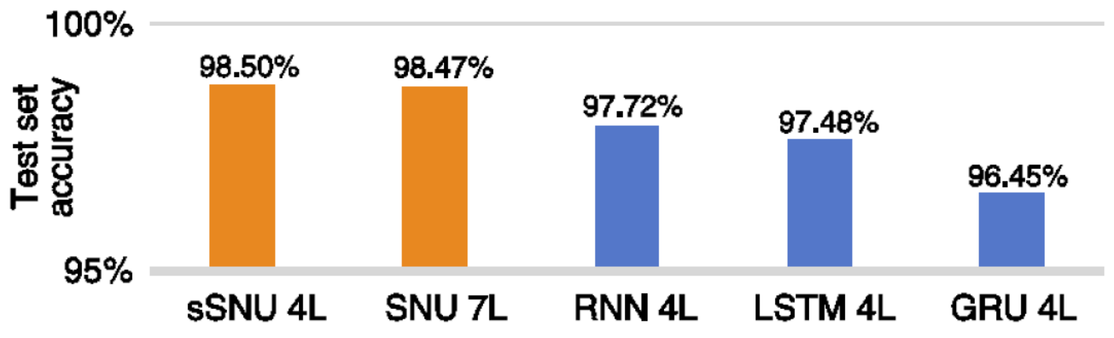
$$b_t = \rho(\tau_a) \odot b_{t-1} + (1 - \rho(\tau_a)) \odot y_{t-1}$$

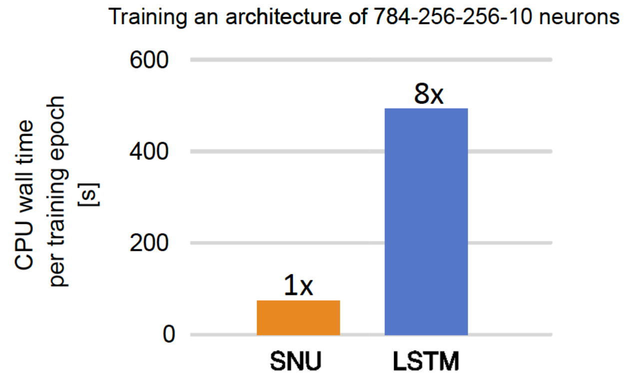
$$y_t = h(s_t + \beta \odot b_t + b_0)$$

SNU offers a new framework for modelling and understanding the neural dynamics

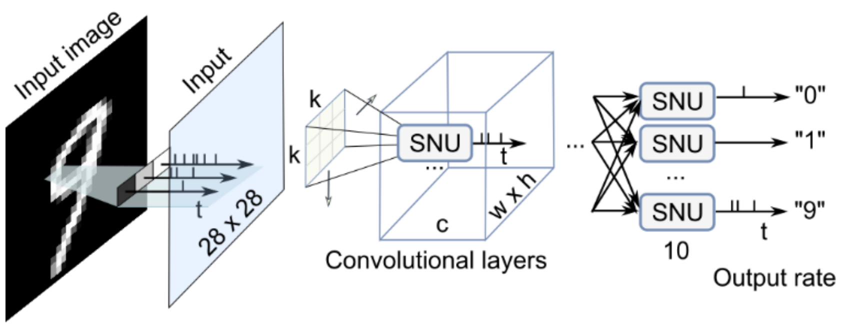
Digit recognition using rate-coded MNIST dataset

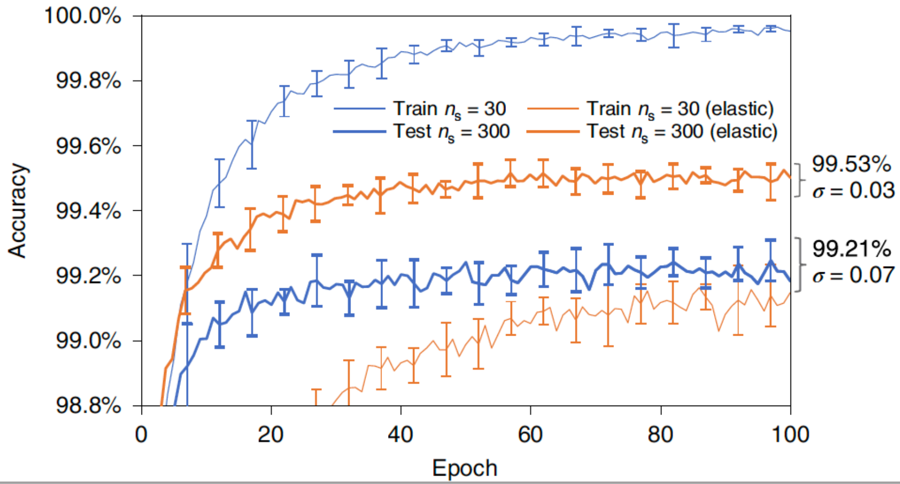
Fully-connected architecture





Convolutional architecture

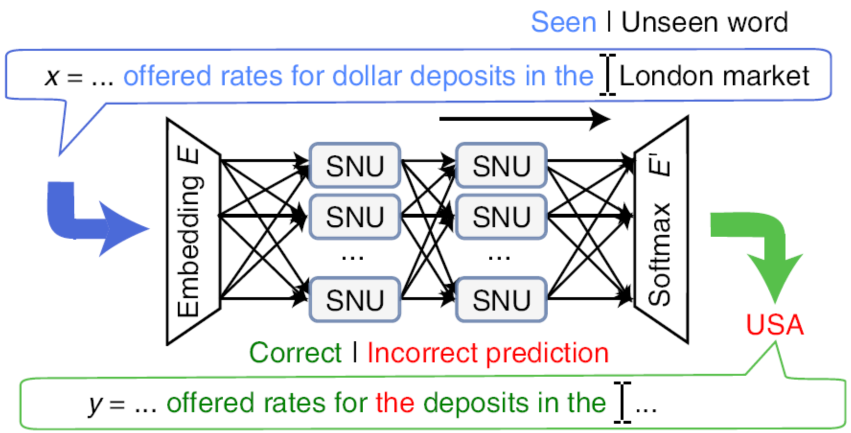


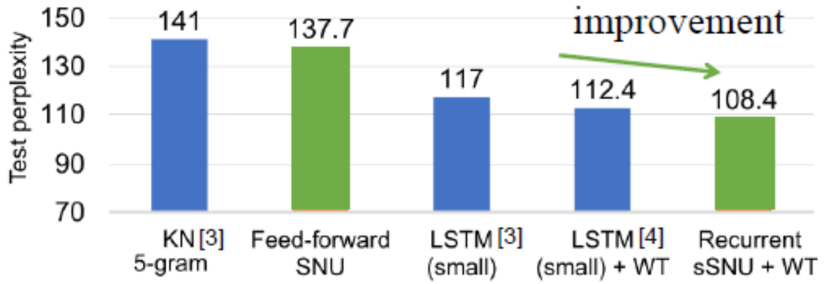


Record SNN accuracy: 99.53% !

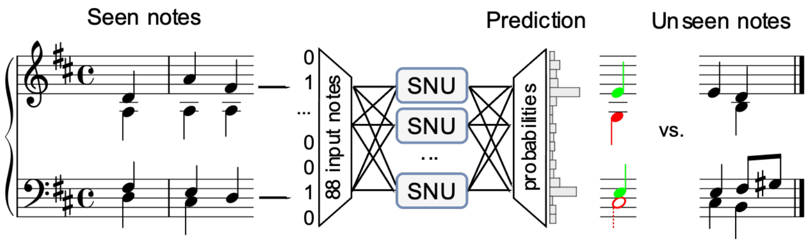
Applications using ANN datasets

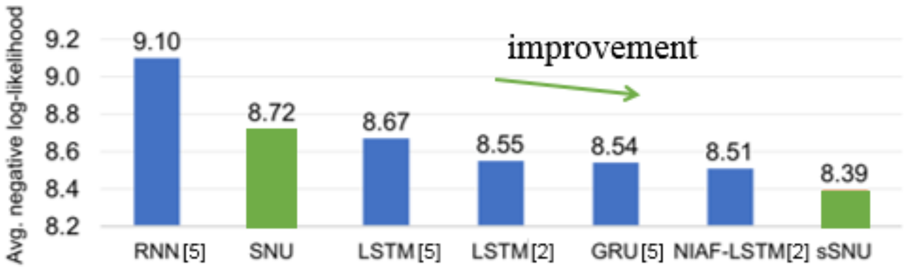
Language modelling (PTB)



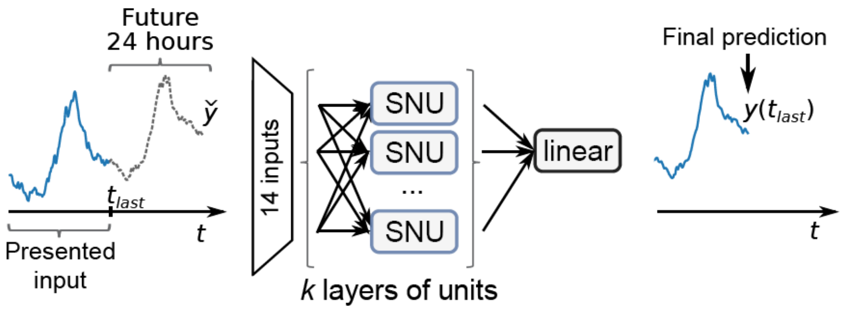


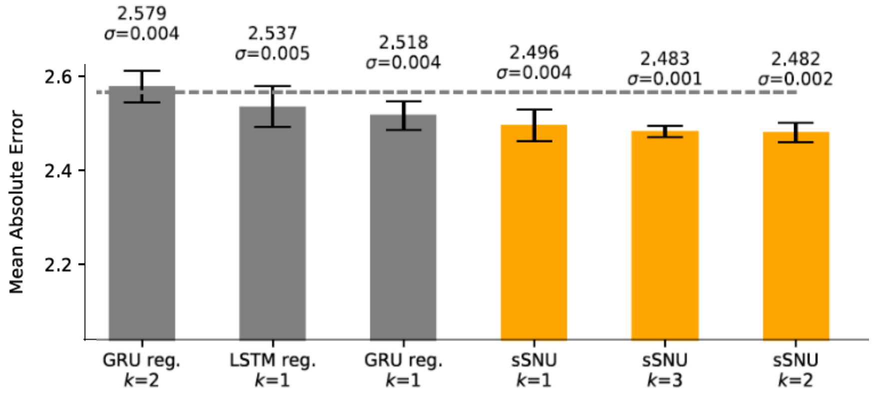
Music prediction (JSB)





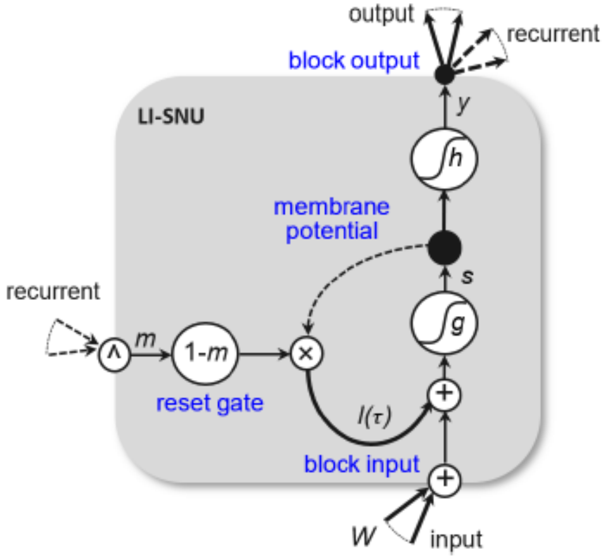
Weather prediction (Jena)



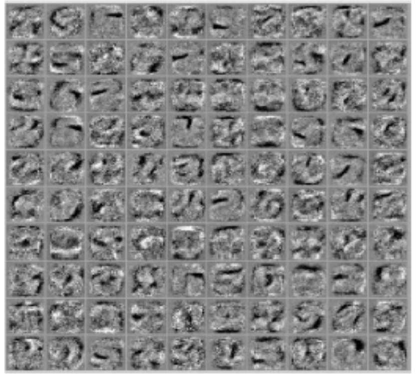


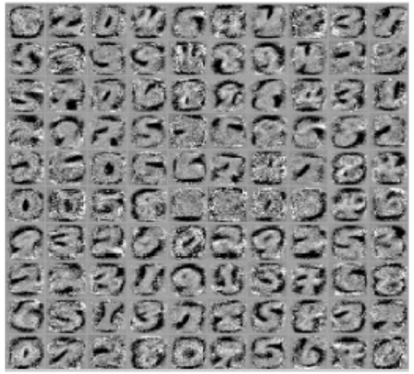
SNU variants: Lateral Inhibition

Lateral Inhibition SNU (LI-SNU)

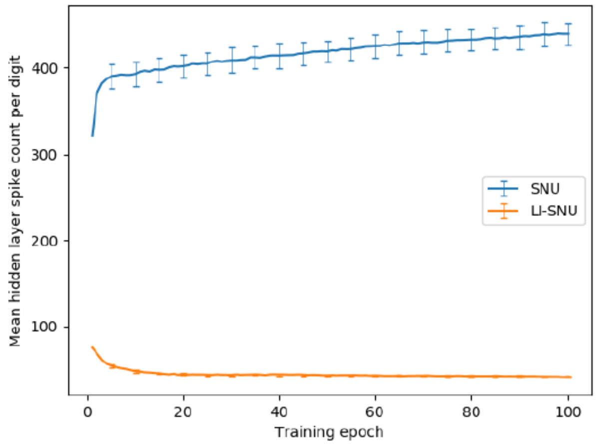


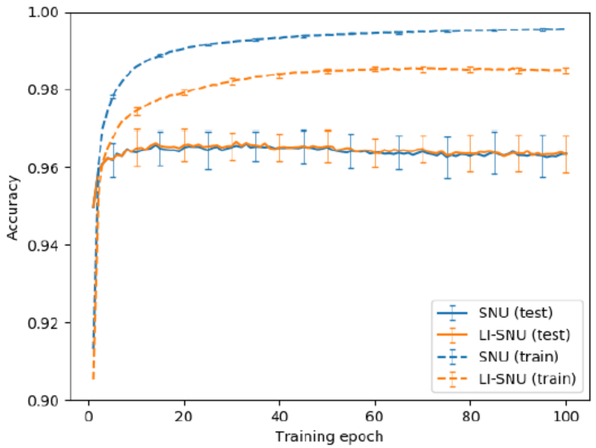
Internal representation: Different kind of features (larger motifs)



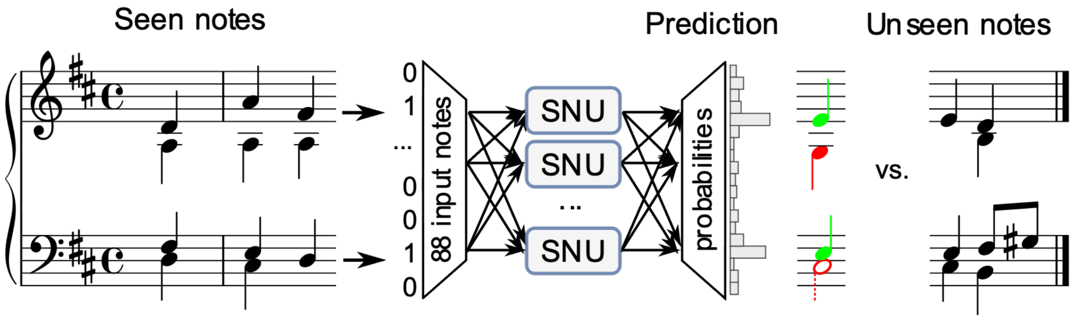


Significant neural activity reduction, and thus lower energy footprint

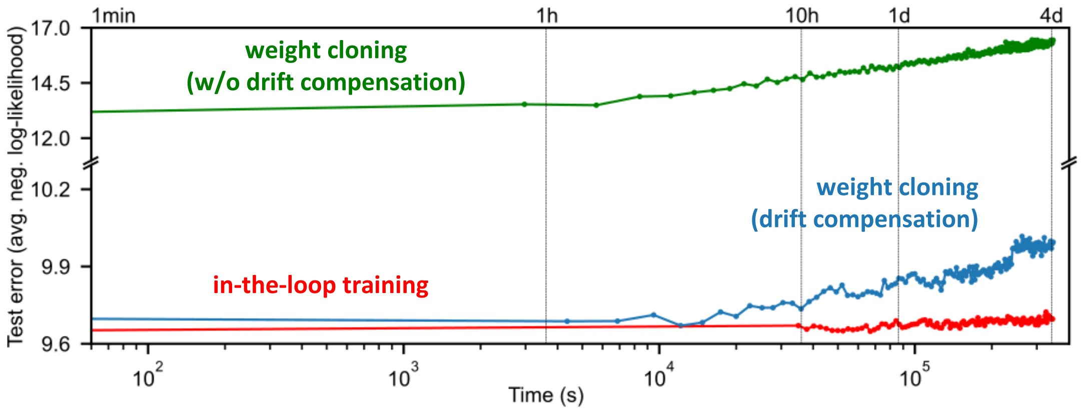




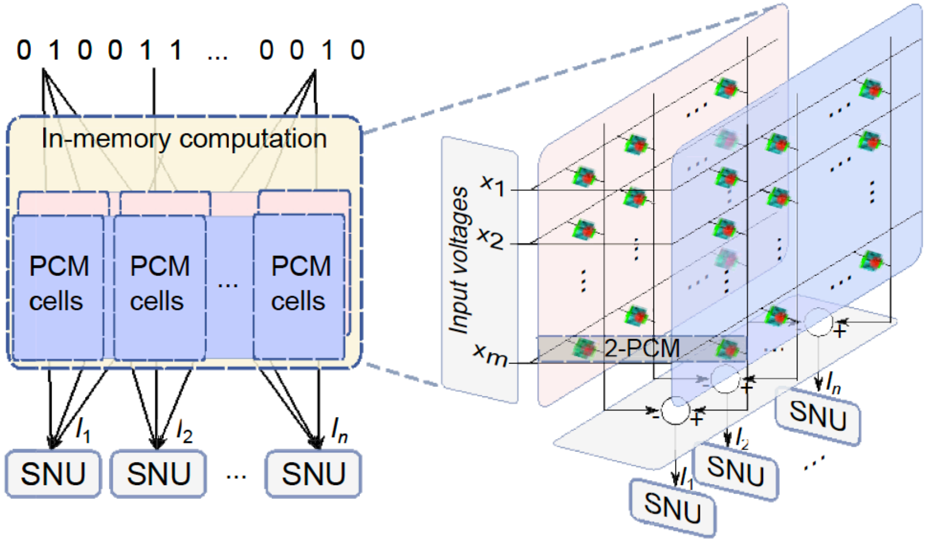
SNUs and in-memory computing



Music prediction: The task is to predict a set of notes (chord) that will be played based on the past notes



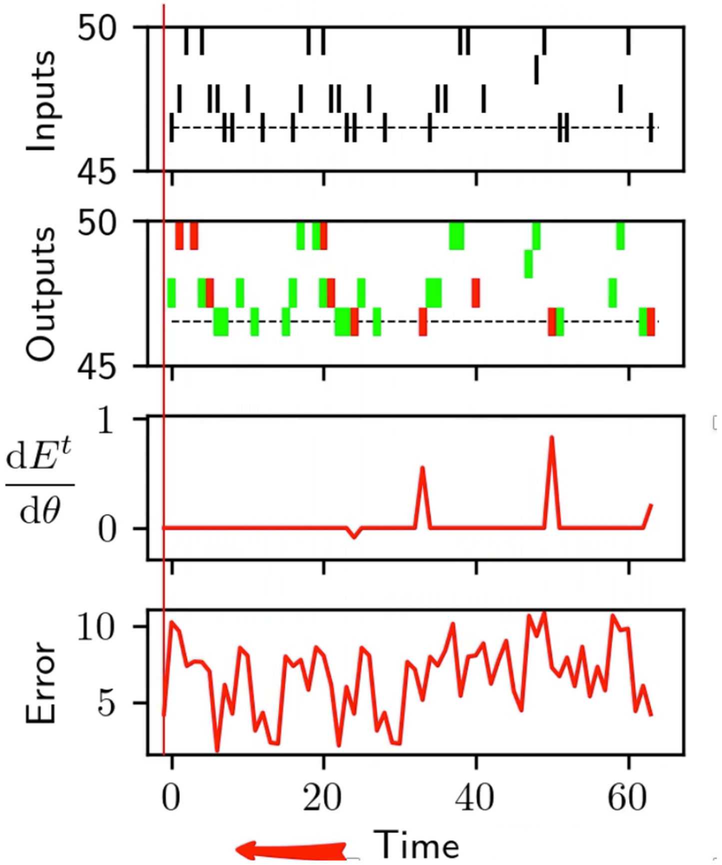
Music prediction inference over 4 days



- **Easy integration of SNNs into emerging in-memory computing**
- **Unified HW design approach supporting both ANNs and SNNs**
- **Training with hardware-in-the-loop compensates for PCM imperfections**

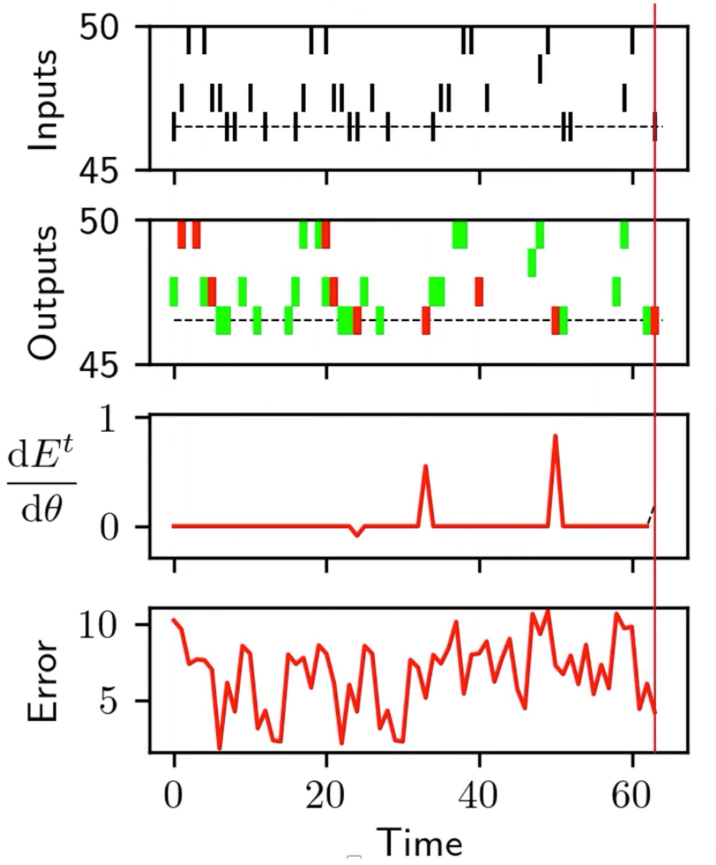
Online learning

Training with BPTT



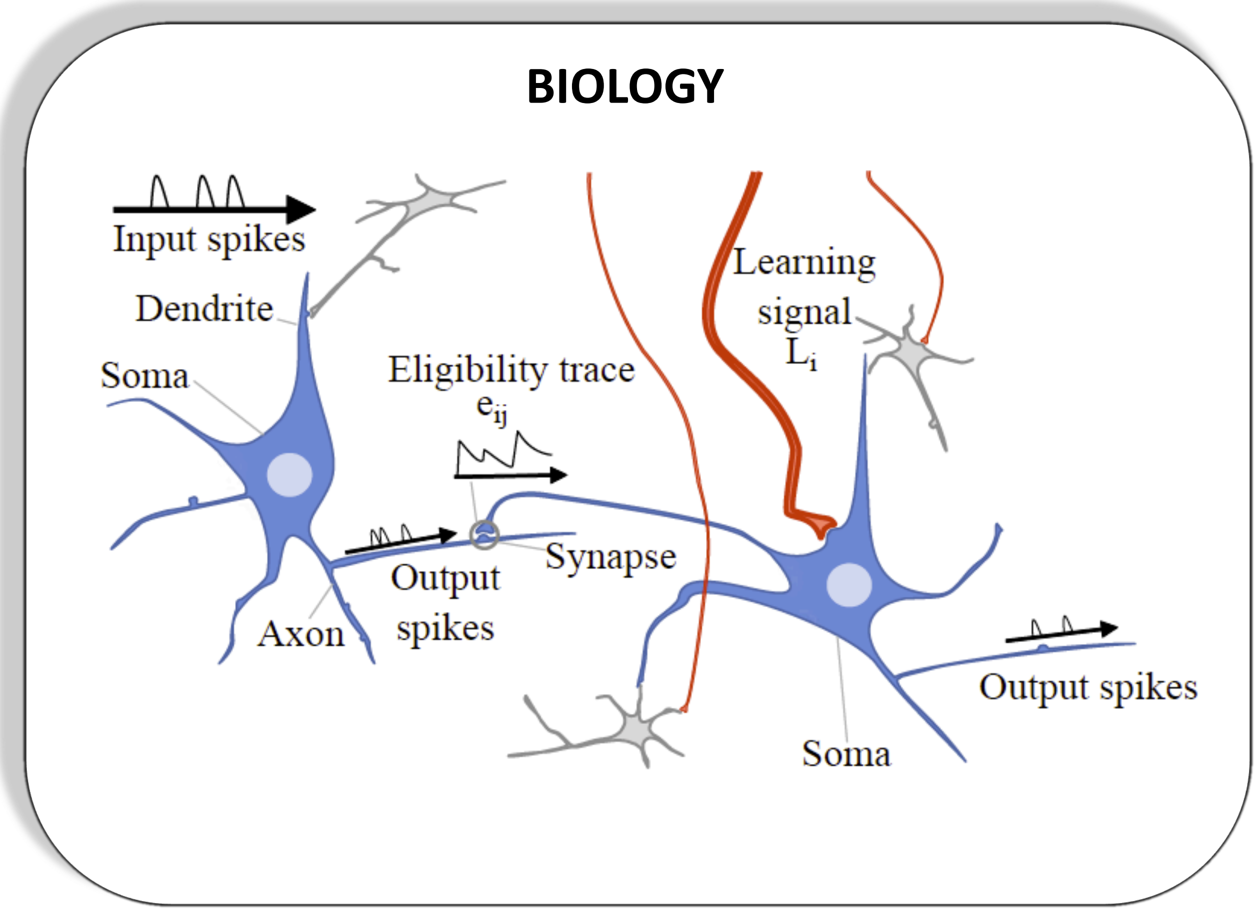
- Input time sequence needs to be truncated
- Normal network operation gets interrupted
- Memory requirement grows with unroll-length

Online learning



- Low-latency learning algorithm
- Continuous network operation
- Constant memory requirements

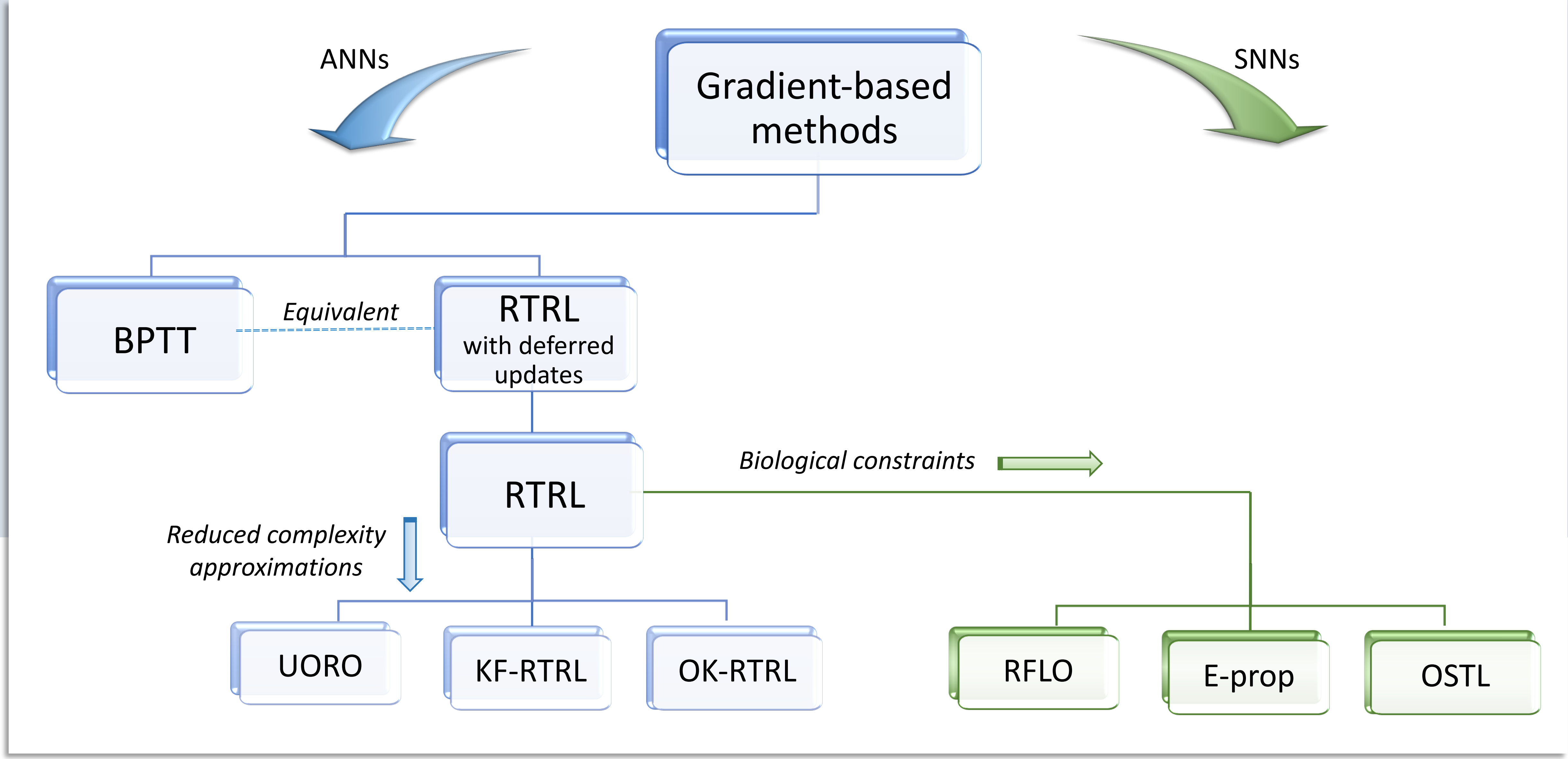
Eligibility traces and Learning signals



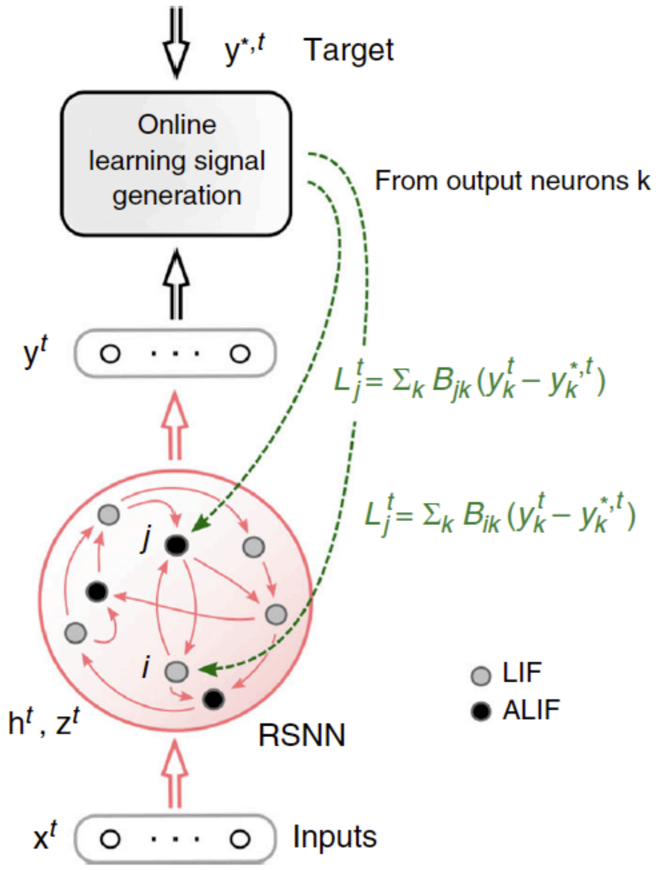
Eligibility traces (e_{ij}) maintain a temporal trace of past neuronal events

Learning signals (L_i) are propagated spatially from different brain regions

Online learning alternatives to BPTT



Online learning alternatives to BPTT: E-prop



<u>E-prop is based on a derivation using a local gradient:</u>

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

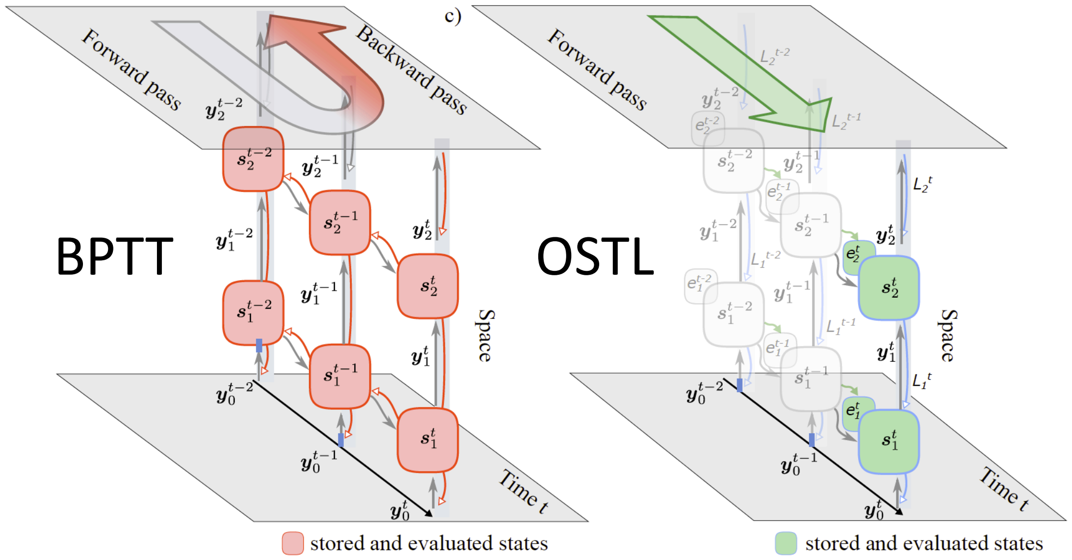
$$\frac{dE}{dW_{ji}} = \sum_t L_j^t \, e_{ji}^t$$

- Eligibility trace for synapse between neuron i to neuron j at time t: e_{ji}^t
- Learning signal for neuron j at time t: L_j^t

G. Bellec, et. al., *Nat. Commun.*, vol. 11, no. 3625, pp. 1–15, Jul 2020

Online learning alternatives to BPTT: OSTL

Online Spatio Temporal Learning (OSTL)
separates spatial and temporal gradients



<u>OSTL exploits a recursion in BPTT:</u>

$$\Delta\theta_l = -\alpha \frac{\mathrm{d}E}{\mathrm{d}\theta_l} = -\alpha \sum_t \frac{\mathrm{d}E^t}{\mathrm{d}\theta_l} = -\alpha \sum_t \frac{\partial E^t}{\partial y_k^t}\left[\frac{\partial y_k^t}{\partial s_k^t}\boxed{\frac{\mathrm{d}s_k^t}{\mathrm{d}\theta_l}} + \frac{\partial y_k^t}{\partial \theta_l}\right]$$

$$= -\alpha \sum_t \boxed{\frac{\partial E^t}{\partial y_k^t}}\boxed{\left[\frac{\partial y_k^t}{\partial s_k^t}\boxed{\epsilon_l^{t,\theta}} + \frac{\partial y_k^t}{\partial \theta_l}\right]}$$

$$= -\alpha \sum_t \boxed{L_l^t}\boxed{e_l^{t,\theta}}$$

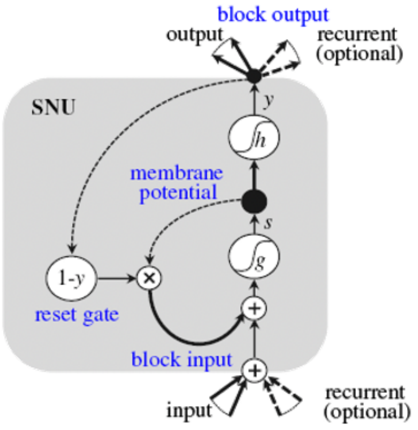
- Learning signal L_l^t — represents spatial gradients
- Eligibility trace e_l^t — represents temporal gradients

T. Bohnstingl, et al. *arXiv*, 2020, arXiv:2007.12723v2

OSTL is gradient-equivalent to BPTT for shallow networks

OSTL for spiking neurons

SNU



$$s_t = g\big(W x_t + l(\tau) \odot s_{t-1} \odot (1 - y_{t-1})\big)$$
$$y_t = h(s_t + b)$$

Eligibility traces and Learning signal for an SNU-based network

$$\boldsymbol{\epsilon}_l^{t,\boldsymbol{\theta}} := \frac{\mathrm{d}\boldsymbol{s}_l^t}{\mathrm{d}\boldsymbol{\theta}_l} = \left(\frac{\mathrm{d}\boldsymbol{s}_l^t}{\mathrm{d}\boldsymbol{s}_l^{t-1}} \boldsymbol{\epsilon}_l^{t-1,\boldsymbol{\theta}} + \left(\frac{\partial \boldsymbol{s}_l^t}{\partial \boldsymbol{\theta}_l} + \frac{\partial \boldsymbol{s}_l^t}{\partial \boldsymbol{y}_l^{t-1}} \frac{\partial \boldsymbol{y}_l^{t-1}}{\partial \boldsymbol{\theta}_l} \right) \right)$$

$$\mathbf{e}_l^{t,\boldsymbol{\theta}} = \frac{\partial \boldsymbol{y}_l^t}{\partial \boldsymbol{s}_l^t} \boldsymbol{\epsilon}_l^{t,\boldsymbol{\theta}} + \frac{\partial \boldsymbol{y}_l^t}{\partial \boldsymbol{\theta}_l}$$

$$\mathbf{L}_l^t = \frac{\partial E^t}{\partial \boldsymbol{y}_k^t} \left(\prod_{(k-l+1)>m\geq 1} \frac{\partial \boldsymbol{y}_{k-m+1}^t}{\partial \boldsymbol{s}_{k-m+1}^t} \frac{\partial \boldsymbol{s}_{k-m+1}^t}{\partial \boldsymbol{y}_{k-m}^t} \right)$$

OSTL has been derived for deep recurrent networks comprising
spiking neurons, LSTMs, GRUs, biological models

T. Bohnstingl, et al. *arXiv*, 2020, arXiv:2007.12723v2

Comparison of Online Learning Algorithms

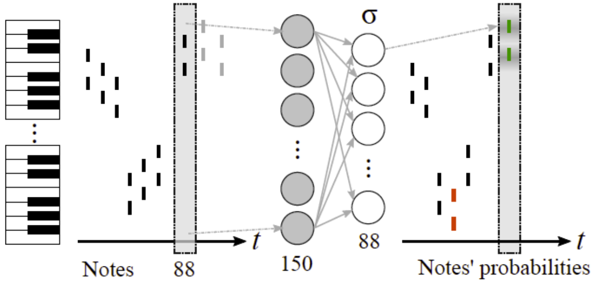
Algorithm	Memory complexity	Time complexity	Exact gradients (vs. BPTT)	Derived for
BPTT (unrolled for T time steps)	Tn	Tn^2	✓	RNNs
RTRL with deferred updates	n^3	n^4	✓	RNNs
RTRL	n^3	n^4	×	RNNs
UORO	n^2	n^2	×	RNNs
KF-RTRL	n^2	n^3	×	RNNs
OK-RTRL (for r summation terms)	rn^2	rn^3	×	RNNs
RFLO	n^2	n^2	×	SNNs & RNNs
SuperSpike (for integration period t)	n^2	tn^2	×	SNNs
E-prop	n^2	n^2	×	SNNs & RNNs
OSTL: feed-forward SNNs (k layers)	kn^2	kn^2	×	SNNs
OSTL: recurrent SNNs (k layers, w/o H)	kn^2	kn^2	×	SNNs
OSTL: feed-forward SNNs	n^2	n^2	✓	SNNs
OSTL: recurrent SNNs (w/o H)	n^2	n^2	×	SNNs
OSTL: generic RNNs (k layers)	kn^3	kn^4	×	SNNs & RNNs
OSTL: generic RNNs	n^3	n^4	✓	SNNs & RNNs

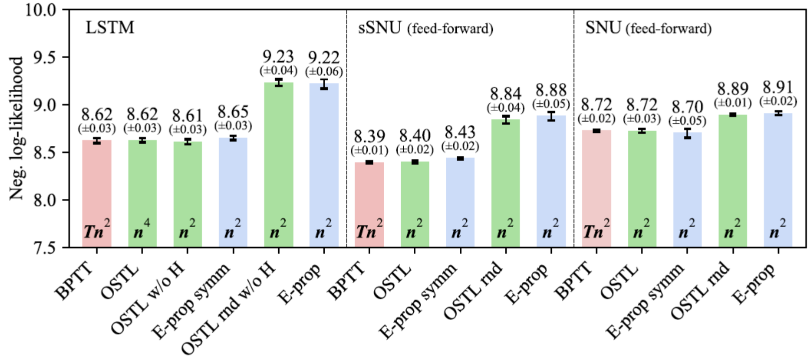
T. Bohnstingl, et al. *arXiv*, 2020, arXiv:2007.12723v2

OSTL results

Music prediction

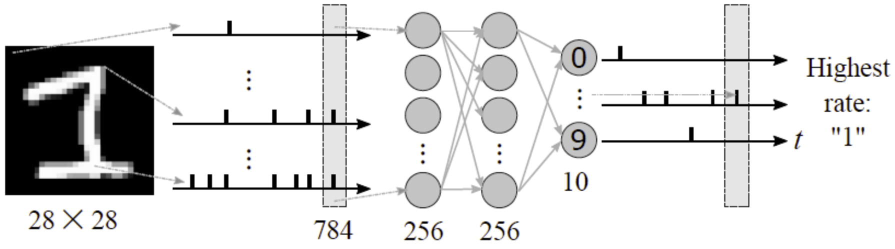
- Johann Sebastian Bach Chorales dataset
- Prediction of next chords
- Gradient-equivalence to BPTT

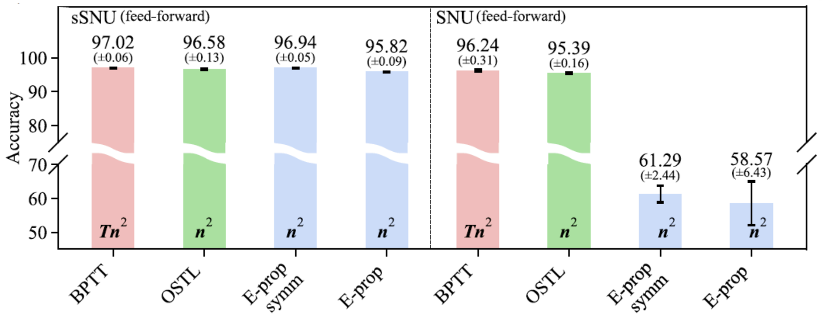




Handwritten digit classification

- MNIST Dataset
- Digit classification



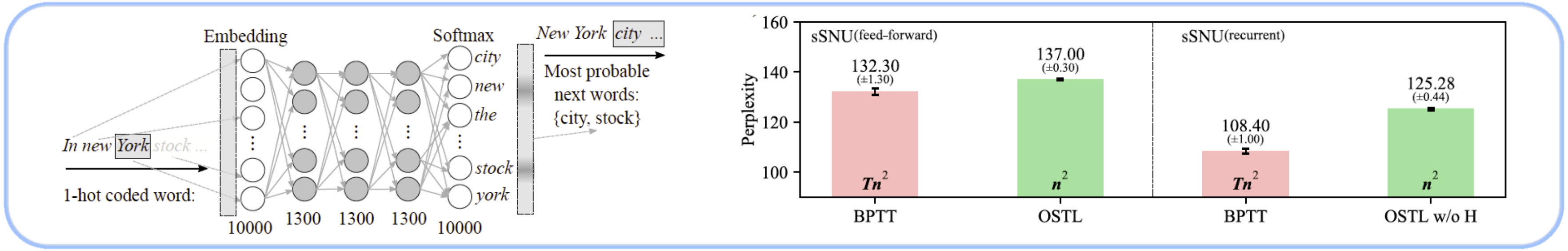


T. Bohnstingl, et al. *arXiv*, 2020, arXiv:2007.12723v2

OSTL results

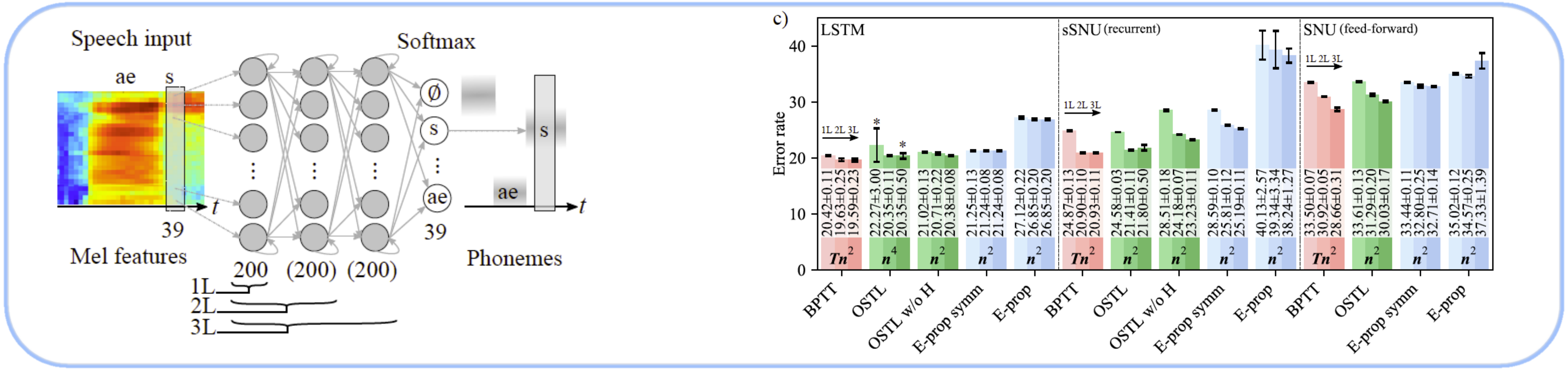
Word-level language modelling

- Penn Tree Bank Dataset
- Next-word prediction (10k words)



Speech recognition

- TIMIT Dataset
- Framewise phoneme classification

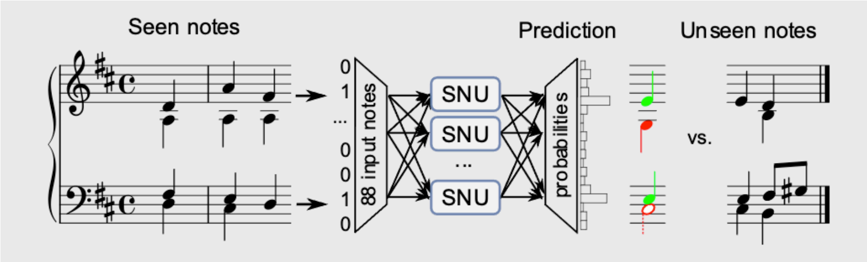


T. Bohnstingl, et al. arXiv, 2020, arXiv:2007.12723v2

AI Applications

New features in tasks for classification and prediction

Explore the effect of adaptation dynamics on the learning capability of neural networks



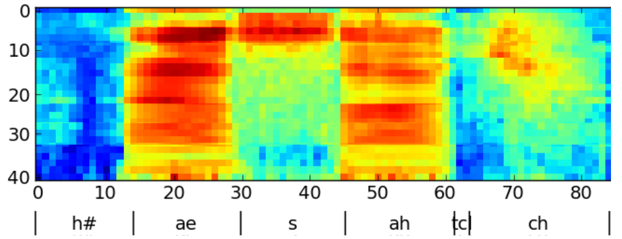
Adaptive low-power neuromorphic AI machinery

based on SNNs with memristive synapses using L2L

Proof of concept: real-world robotics environment

Large-scale application to speech recognition

Go beyond the standard research benchmarks

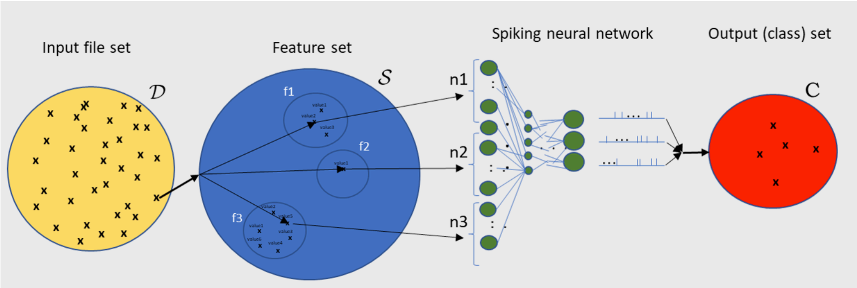


Abstract reasoning

Investigate neural network systems that exhibit analytic intelligence

Real-time classification, storage and recall

Design neural networks for real-time classification of data in resource-constrained environments



Acknowledgements

Stanisław Woźniak

Thomas Bohnstingl

Giorgia Dellaferrera

Ana Stanojevic

Ayush Garg

Giovanni Cherubini

Evangelos Eleftheriou

and current and former colleagues

from the Neuromorphic Computing & IO Links team

FNSNF
SWISS NATIONAL SCIENCE FOUNDATION

chist-era

IBM