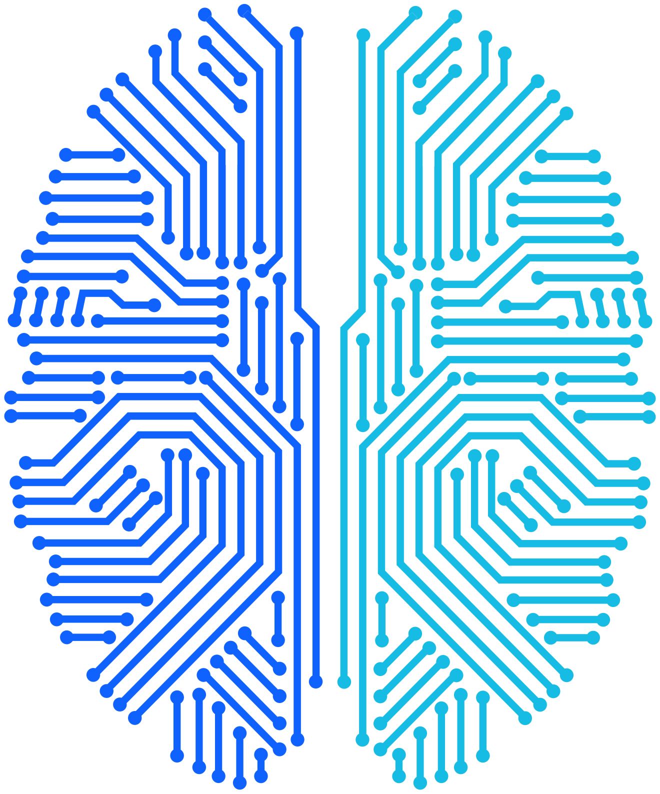
Neuromorphic computing

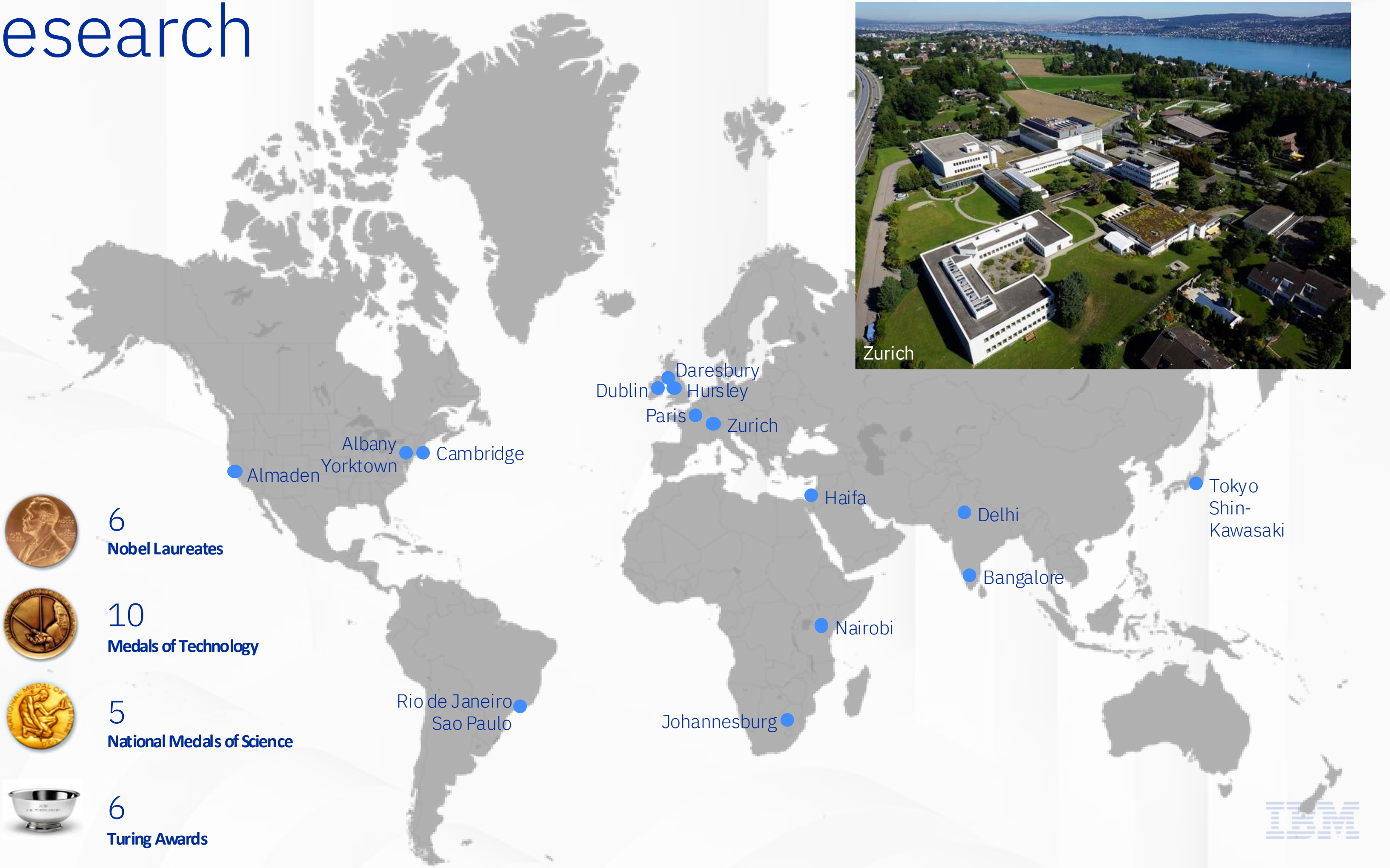
Dr. Stanisław Woźniak, Dr. Thomas Ortner
Emerging Computing & Circuits Team
IBM Research – Zurich

EdgeML School 2024, CERN, 25.09.2024

Contents



IBM Research



6
Nobel Laureates

10
Medals of Technology

5
National Medals of Science

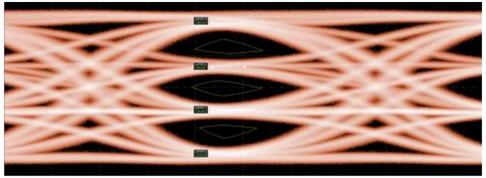
6
Turing Awards

Key focus areas of our team @ IBM Research – Zurich

Emerging Computing and Circuits
Dr. Angeliki Pantazi

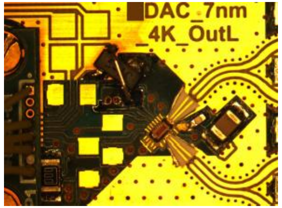
High-speed I/O Links

We are developing next-generation I/O Links for the IBM flagship Z and P processors and for future accelerators



Quantum Electronics

We are developing cryogenic CMOS electronics aiming to continue pushing the scalability and affordability of Quantum systems



Neuro-inspired Computing

We are exploring neuro-inspired models and learning algorithms towards energy- and data-efficient AI architectures

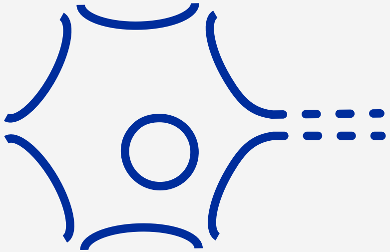
Motivations for neuromorphic computing

Improving AI systems

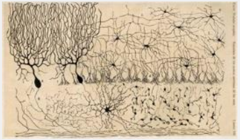
In order to make AI:	We need:
More efficient	Low power and low latency
Smarter	Advanced cognitive features
More flexible	Online and continual learning

Potential inspiration from the brain

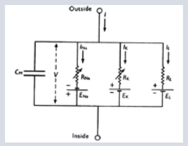
These tactics include:
Event-based communication
Efficient neuronal and synaptic dynamics
Local, supervised and unsupervised learning

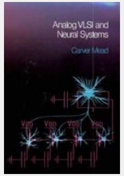


History of neuromorphic computing: Biology vs. Technology

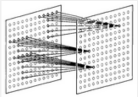


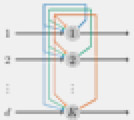
1906: Golgi, Cajal
interconnected
neurons





1952: Hodgkin & Huxley
1970s: Dodge, Cooley, Traub, Llinas
compartmental **Spiking Neuron**
1980s: **Neuromorphic Computing**





1979: Fukushima's **CNN**
1982: **Hopfield network**
1990: **Recurrent Networks**
Elman, Jordan

2005: Blue Brain Project
2013: HBP, Brain Initiative
2009: Neurogrid
2014: TrueNorth

Blue Brain Project

SNNs in deep learning:
2020: SNUs (LM, GAN)
BP alternatives:
2022: e-prop, OSTL

strong ties

2023: DeepMind's **LRU**

Back to O(n) inference scaling
+ large-scale models

Biology

What to model?

How to enrich dynamics?
(neuron)

How to enrich dynamics?
(network)

How to achieve efficiency?
(neuron)

How to achieve efficiency?
(network)

Inspiration

Questions

Neuro-inspired computing

Technology

How to model?

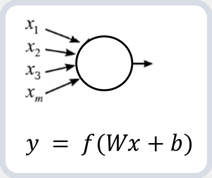
How to train?
(error calculation)

How to train?
(vanishing gradients)

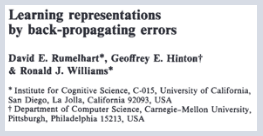
How to train large-scale models?
(scaling up)

Validation

1943: McCulloch and Pitts
Artificial Neuron



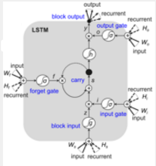
Control & optimization:
1960: ADALINE **GD with MSE**
1967: Amari's **SGD**
1982: Werbos's **MLP backprop**
1986: widespread popularity


Learning representations by back-propagating errors

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

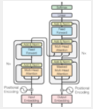
* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

1997: **LSTM**
2014: **GRU**



speech, writing,
translation, LMs

2012: **GPUs** for AlexNet record
2017: **Transformers**
2022: **LLMs:**
GPT, Gemini,
LLaMA, Granite

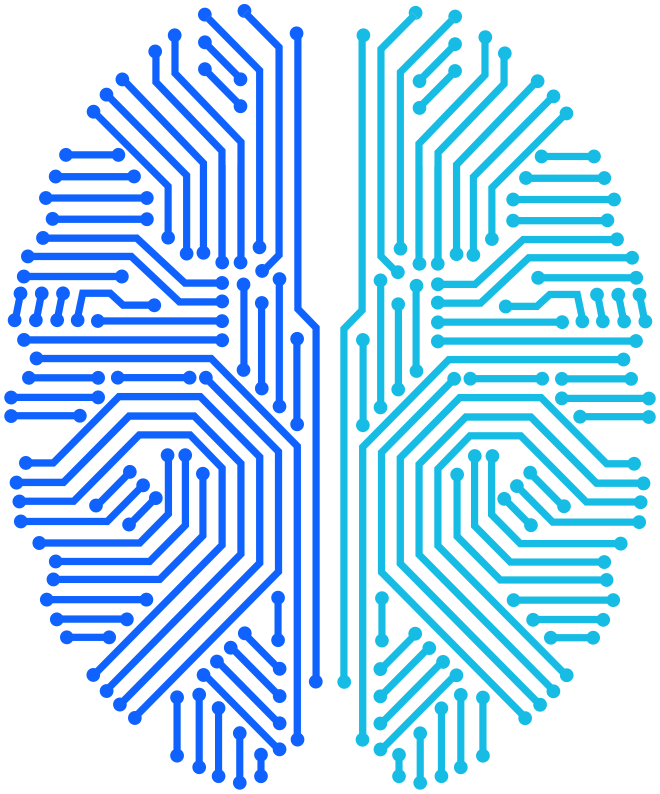


2021: recurrency in **SSM** S4

Away from biology:
- no recurrency
- simple activation functions
- non-biologically plausible models

O(n^2) inference scaling

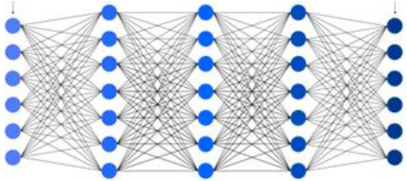
Contents



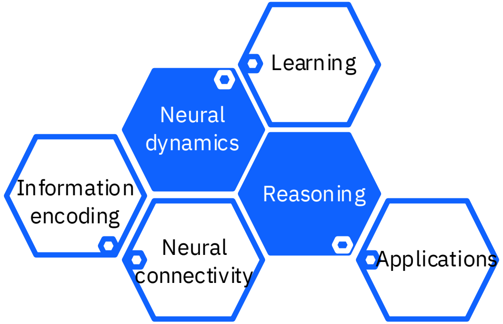
Neuromorphic research: Our approach

Taking inspiration from biology

Applying the rigor of machine learning



NeuroAI Toolkit



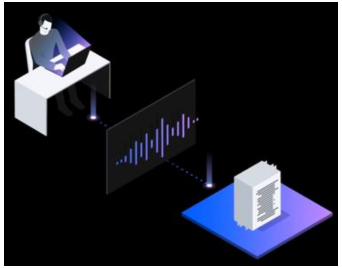
https://github.com/IBM/neuroaikit

https://research.ibm.com/projects/neuromorphic-computing

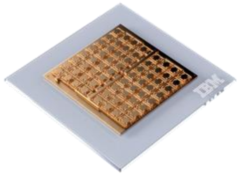
Research papers
1. S. Wozniak, et al. *Nature Machine Intelligence,* 2020
2. T. Ortner et al., *IEEE ICASSP,* 2022
3. T. Ortner et al., *IEEE Trans. Neural Networks Learn. Syst,* 2022
4. A. Stanojevic et al., *Neural Networks* 2023
5. G. Dellaferrera et al., *Nature Communications,* 2022
6. S. Wozniak, et al., *Nature Communications,* 2023
7. Y. Schnider, et al., *IEEE CVPRW,* 2023
8. A. Stanjoevic et al., *Nature Communications,* 2024

...

Provides efficient solutions for multiple AI applications



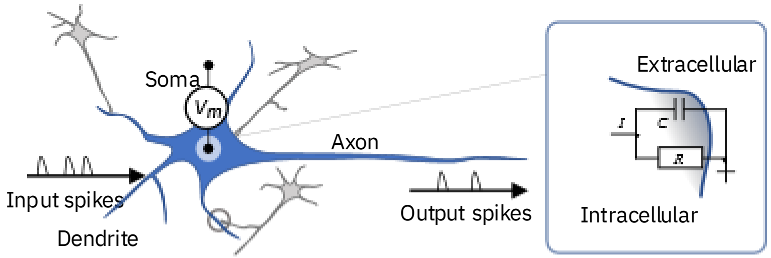
Exploits acceleration of the hardware infrastructure



Neural dynamics: Spiking Neural Unit (SNU)

SNUs operate either in spiking (binary signals) or non-spiking mode (real-valued signals)

Biology



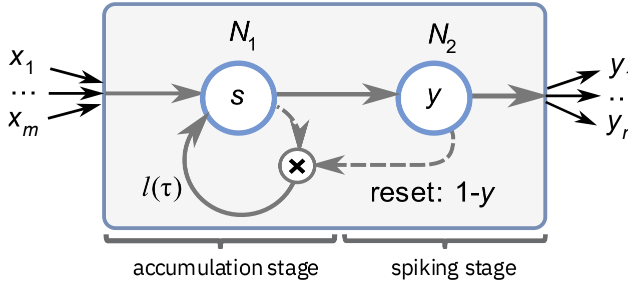
$$\tau \frac{\mathrm{d}V_m(t)}{\mathrm{d}t} = -V_m(t) + RI(t)$$

spike on

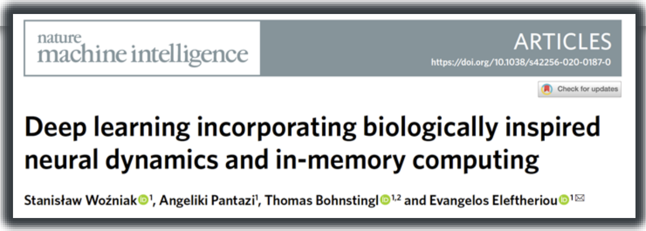
$V_m > V_{\mathrm{th}}$

Deep Learning

Spiking Neural Unit (SNU)



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



nature machine intelligence

ARTICLES

https://doi.org/10.1038/s42256-020-0187-0

Check for updates

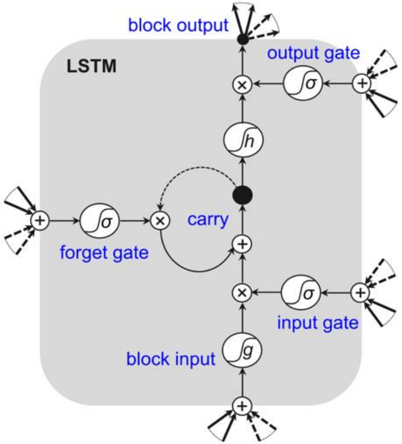
Deep learning incorporating biologically inspired neural dynamics and in-memory computing

Stanisław Woźniak[1], Angeliki Pantazi[1], Thomas Bohnstingl[1,2] and Evangelos Eleftheriou[1]

Easily build large models by replacing complex units with SNUs

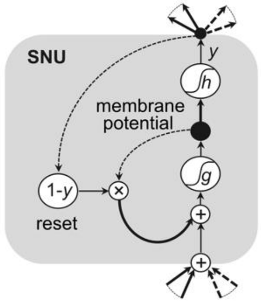
Traditional ANN (RNN Units)

- Require large number of trainable parameters
- Operate with complex internal dynamics and neuronal connectivity



Spiking Neural Unit (SNU)

- Requires fewer parameters
- Offers qualitatively different dynamics
- Easily extensible to incorporate additional features from neuroscience



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

Neural dynamics: Application examples

Solving visual analytic intelligence riddles

[+] improved accuracy

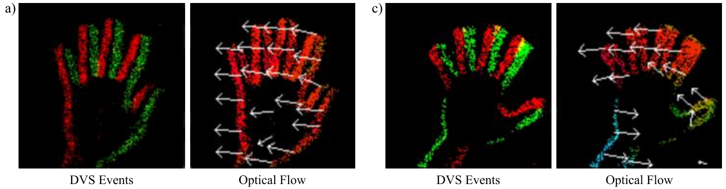
[+] smaller models vs. ANN

S. Woźniak, et al., “On the visual analytic intelligence of neural networks,” *Nat Commun*, vol. 14, no. 1, p. 5978, Sep. 2023.

Optical flow computation

[+] improved accuracy

[+] smaller models vs. ANN



Y. Schnider *et al.*, “Neuromorphic Optical Flow and Real-time Implementation with Event Cameras.” WEV CVPR, 2023.

Drone navigation

[+] improved accuracy

[+] higher sparsity vs. ANN

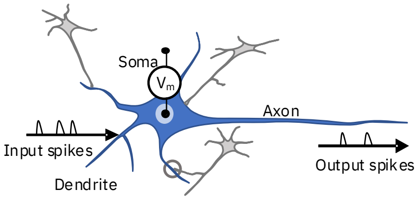
S. Govil, “Spiking Neural Networks for Drone Navigation”, MSc Thesis, RPG UZH & IBM Research – Zurich, Sept. 2023

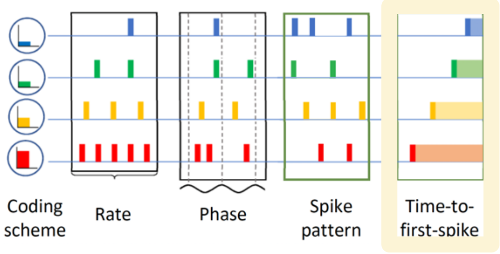
Common aspect: Temporal/sequential problems that leverage the unique neuronal dynamics

Information encoding

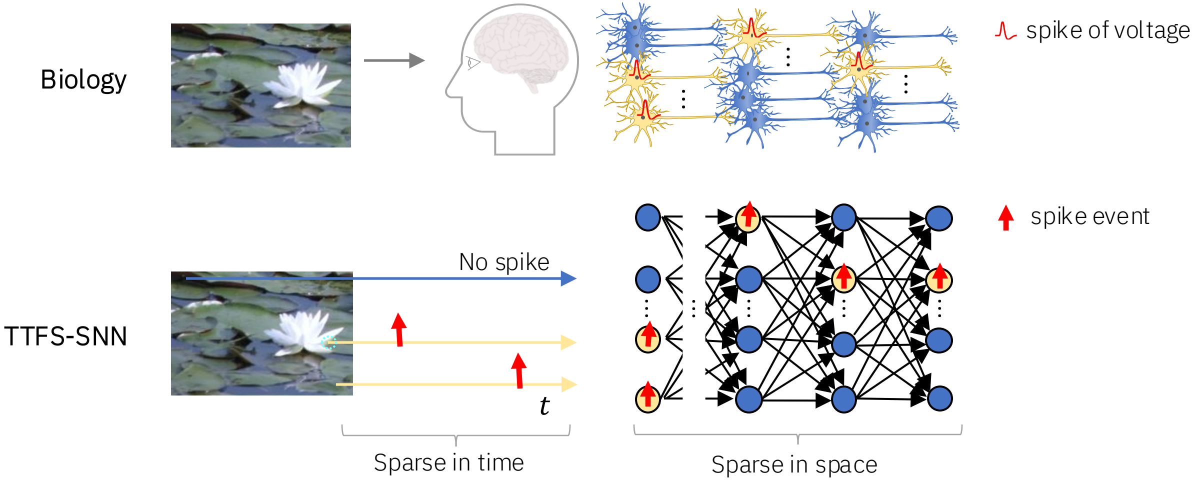
Information encoding: Time-To-First-Spike (TTFS) Networks

Different information coding schemes:

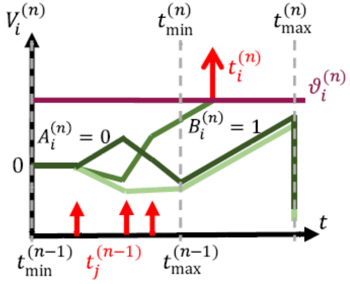




Leveraging temporal and spatial sparsity of TTFS:



Novel TTFS neuron [1]:



A. Stanojevic et al., "High-performance deep spiking neural networks with 0.3 spikes per neuron", *Nature Communications 2024.*

Information encoding: Time-To-First-Spike (TTFS) Networks

A network with proposed TTFS neurons [1]:

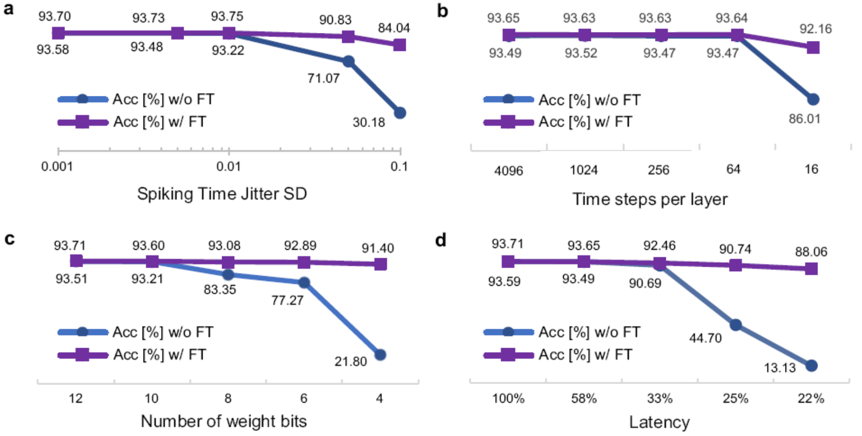
- achieves equivalent inference accuracy to the state-of-the-art ReLU networks
- enables lossless conversion from pre-trained ReLU networks
- follows the same training trajectories as ReLU networks, enabling high accuracy training
- enables fine-tuning for specifics of spiking neuromorphic hardware

Key aspects: Static problems. Neuronal dynamics is leveraged for TTFS-based communication, achieving ReLU equivalent computational logic with sparse spikes.

VGG16:

Dataset	Test accuracy [%] w/ FT		SNN
	ReLU	SNN	Sparsity
CIFAR10	**93.69 ± 0.02**	**93.69 ± 0.02**	**0.38**
CIFAR10[52]	–	91.90	0.24
CIFAR10[53]	–	92.68	0.62
CIFAR10 + L1	**93.28 ± 0.02**	**93.27 ± 0.02**	**0.20**
CIFAR100	**72.23 ± 0.06**	**72.24 ± 0.06**	**0.38**
CIFAR100[52]	–	65.98	0.28
CIFAR100 + L1	**72.20 ± 0.04**	**72.21 ± 0.04**	**0.24**
PLACES365	**53.86 ± 0.02**	**53.86 ± 0.02**	**0.54**
PLACES365 + L1	**48.88 ± 0.06**	**48.85 ± 0.06**	**0.27**

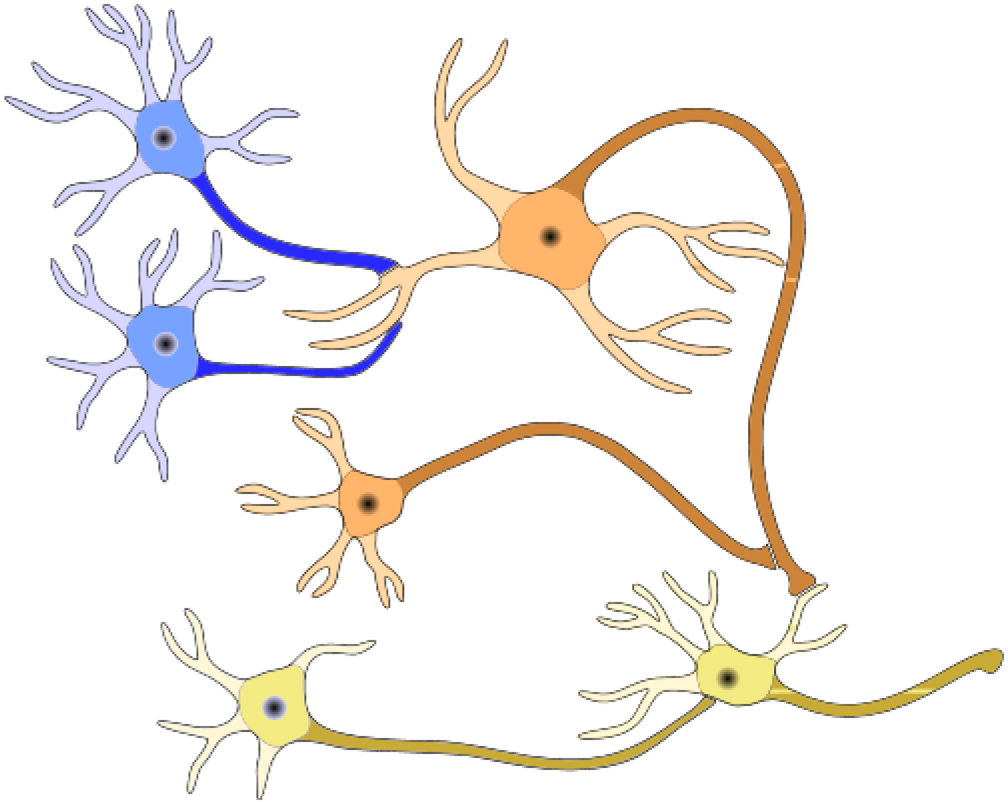
VGG16 CIFAR10 fine-tuning for hardware specifics



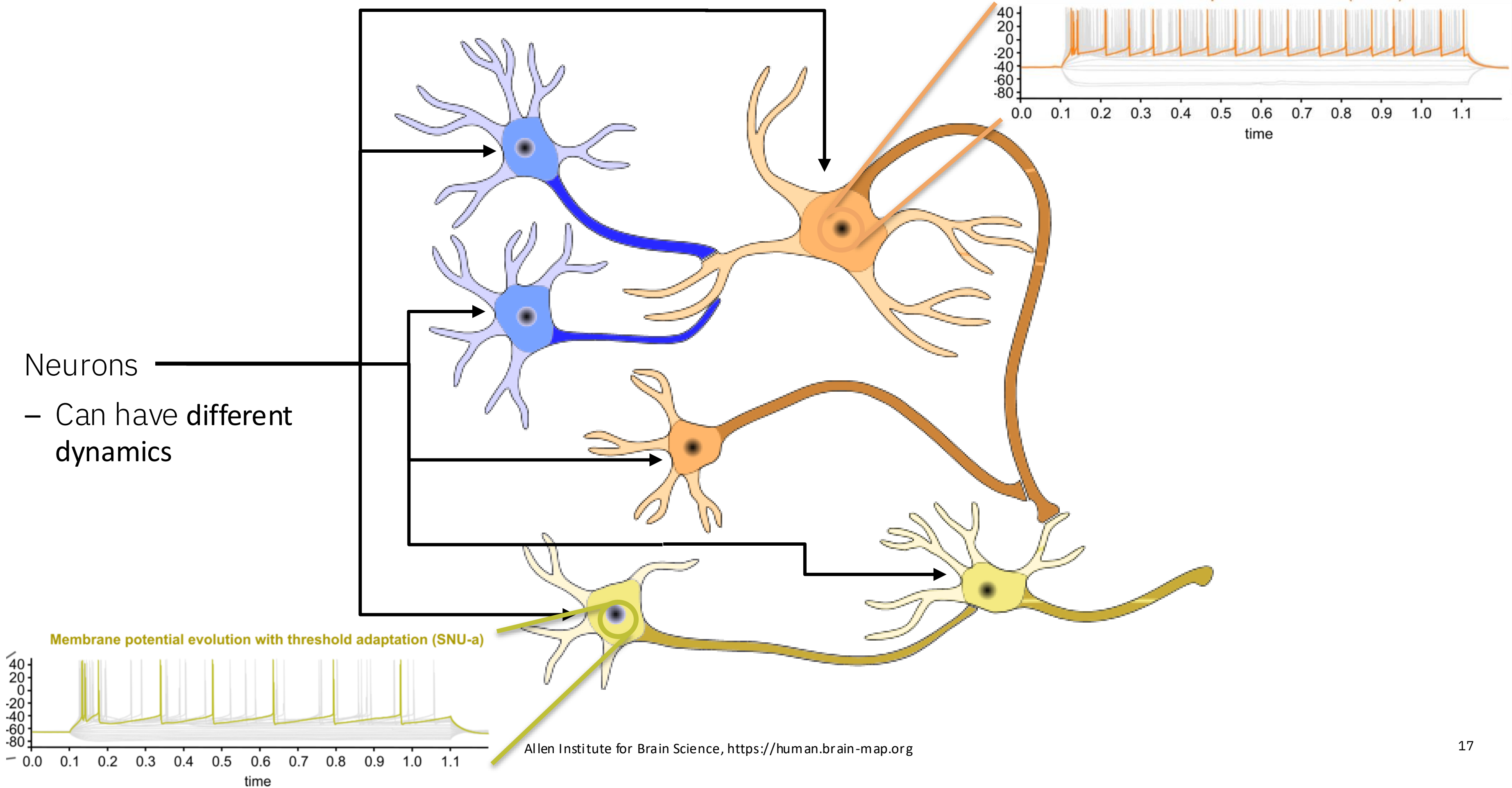
A. Stanojevic et al., "High-performance deep spiking neural networks with 0.3 spikes per neuron", *Nature Communications 2024.*

Neural connectivity: Modelling neural diversity

Biological neural networks are highly diverse



Biological neural networks are highly diverse



Neurons

– Can have **different dynamics**

Allen Institute for Brain Science, https://human.brain-map.org

Biological neural networks are highly diverse

Axo-Dendritic synapses

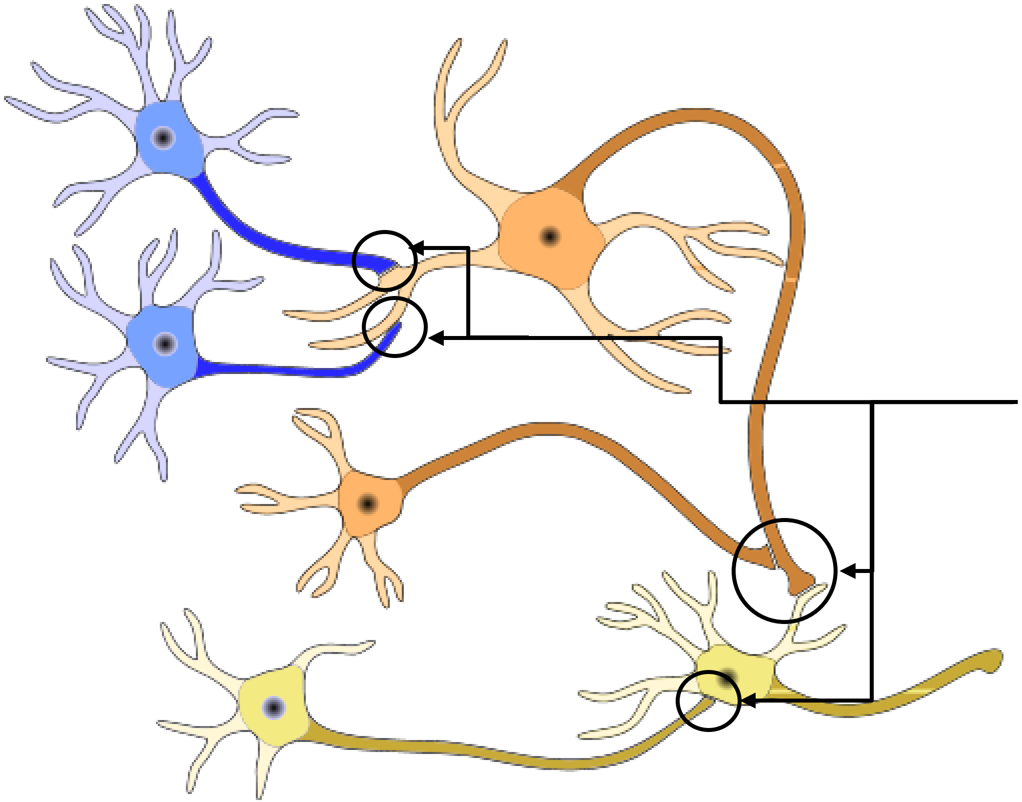
– Connecting the **axon** of the pre-synaptic neuron to the **dendrite** of the post-synaptic neuron

Axo-Somatic synapses

– Connecting the **axon** of the pre-synaptic neuron to the **soma** of the post-synaptic neuron

Axo-Axonic synapses

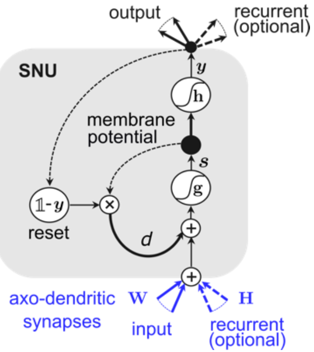
– Connecting the **axon** of the pre-synaptic neuron to the **axon** of the post-synaptic neuron



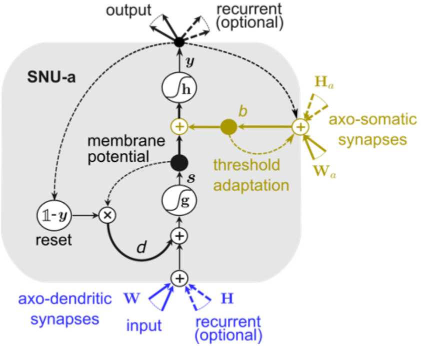
Synapses

– Can be of a **different kind**

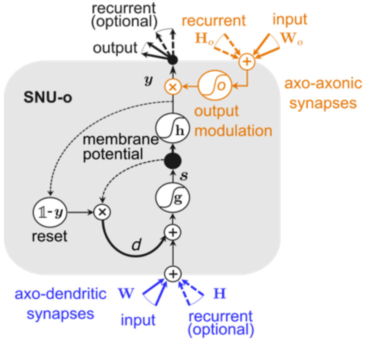
Various neuron and synapse types can be modelled with SNUs



$$s^t = g(Wx^t + Hy^{t-1} + l(\tau) \cdot s^{t-1} \odot (1 - y^{t-1}))$$
$$y^t = h(s^t + b)$$



$$s^t = g(Wx^t + Hy^{t-1} + l(\tau) \cdot s^{t-1} \odot (1 - y^{t-1}))$$
$$b^t = \rho \cdot b^{t-1} + (1 - \rho) \cdot (W_a x^t + H_a y^{t-1})$$
$$y^t = h(s^t + \beta b^t + b_0)$$



$$s^t = g(Wx^t + Hy^{t-1} + l(\tau) \cdot s^{t-1} \odot (1 - y^{t-1}))$$
$$y^t = h(s^t + b) \odot \sigma(W_o x^t + H_o y^{t-1} + b_o)$$

T. Bohnstingl et al., *ICASSP* 2022; T. Bohnstingl et al., *NeurIPS WS ENLSP* 2021;
https://doi.org/10.1109/AICAS57966.2023.10168623

Biologically-inspired learning

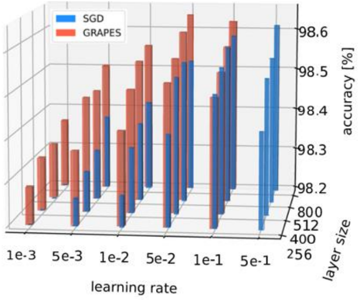
Biologically-inspired extension to Error Backpropagation (BP)

**Group Responsibility for Adjusting the Propagation of Error Signals*

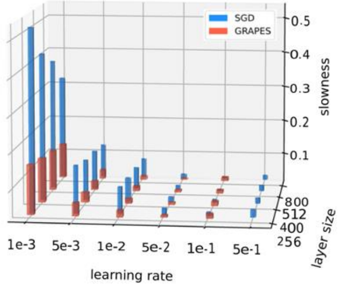
GRAPES* is an optimization strategy that relies on the notion of the **node importance** in propagating the error information during learning

GRAPES improves the accuracy and convergence rate of BP

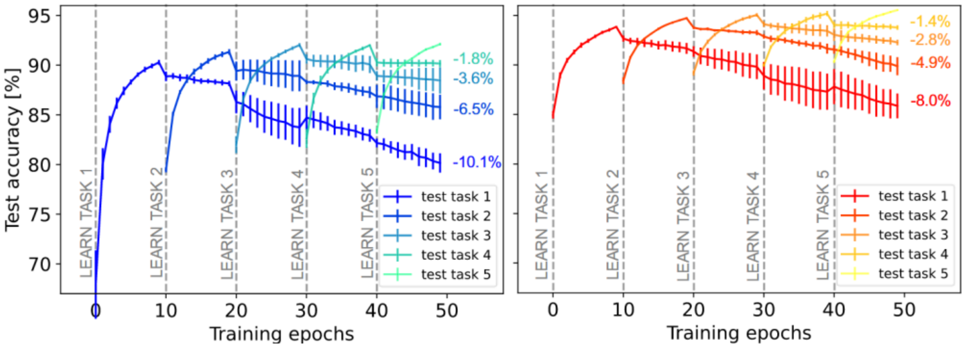
Accuracy



Convergence rate

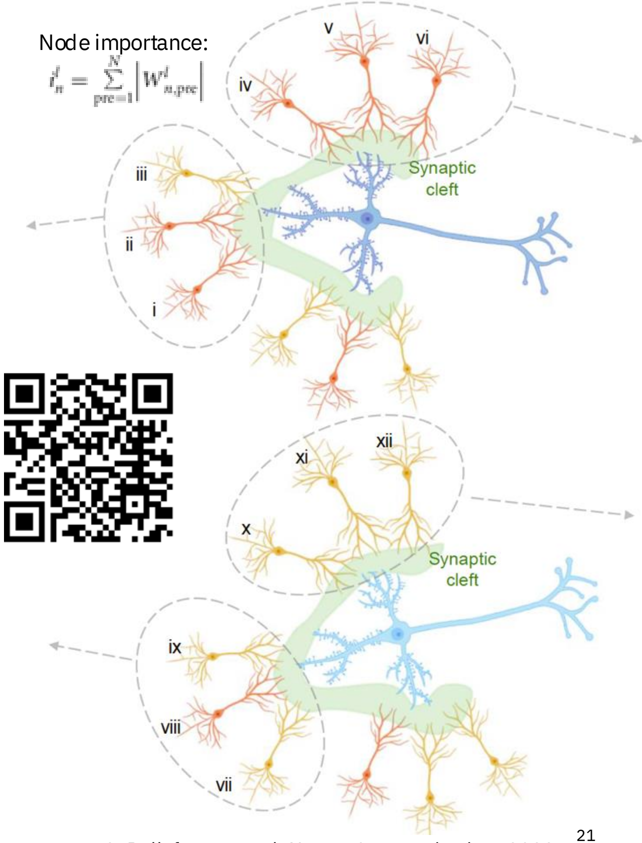


GRAPES reduces Catastrophic Forgetting



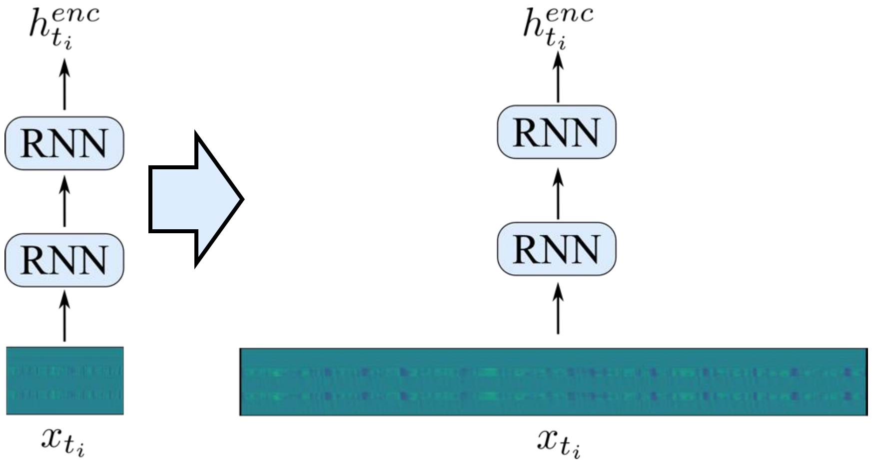
Node importance:

$$i_n^l = \sum_{pre=1}^{N} \left| w_{n,pre}^l \right|$$

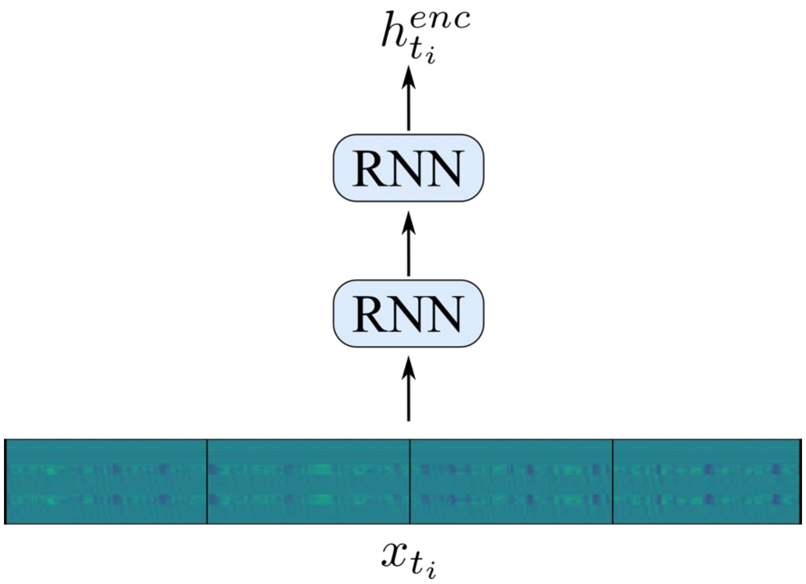


G. Dellaferrera et al. *Nature Communications* 2022
https://doi.org/10.1038/s41467-022-29491-2

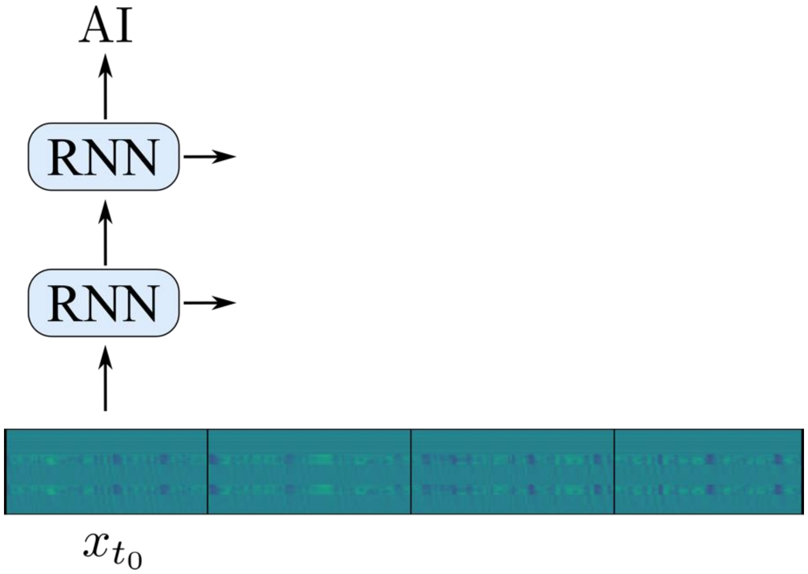
The inner workings of
Recurrent Neural Networks



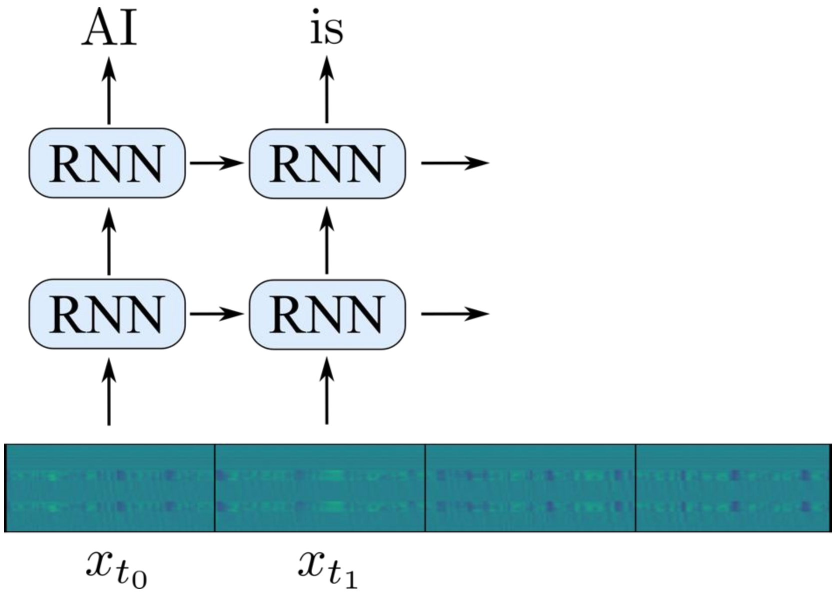
The forward path



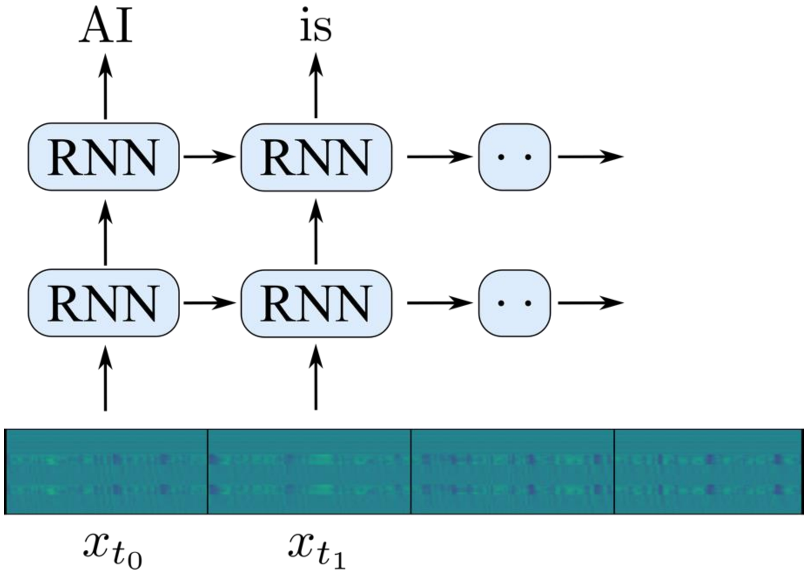
The forward path



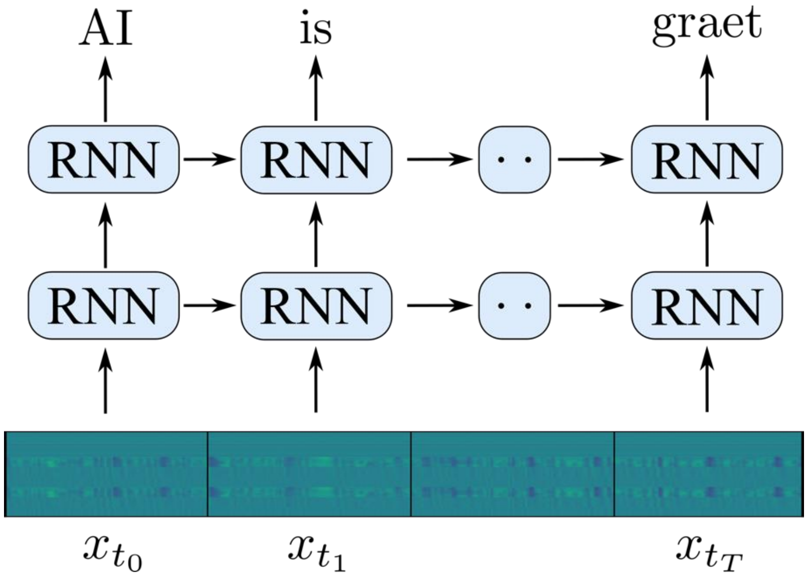
The forward path



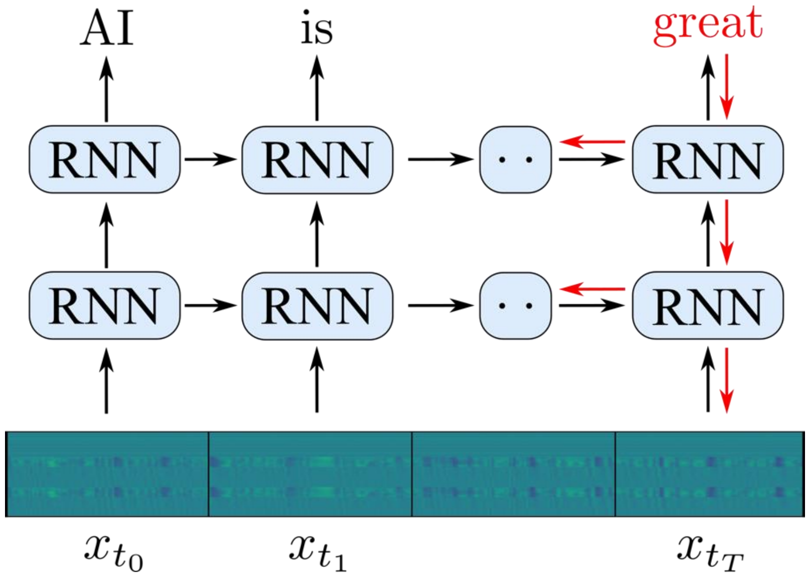
The forward path



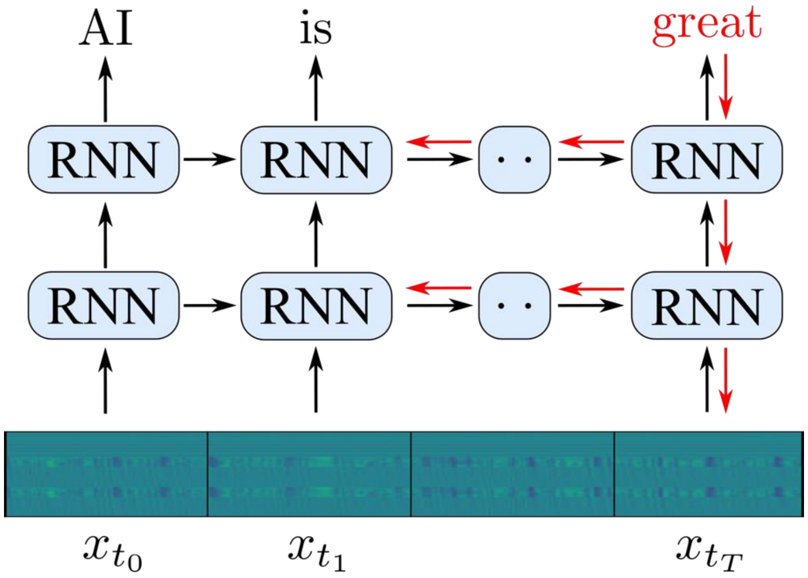
The forward path



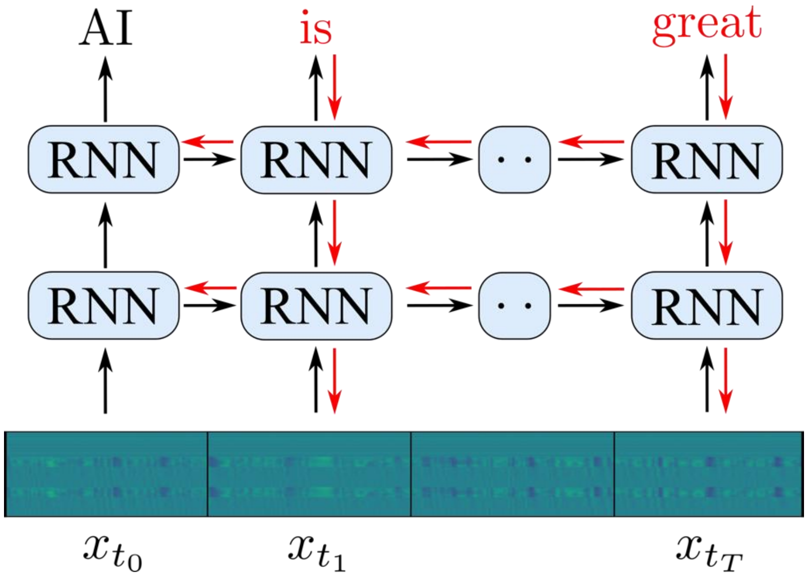
The backward path using Error Backpropagation through time (BPTT)



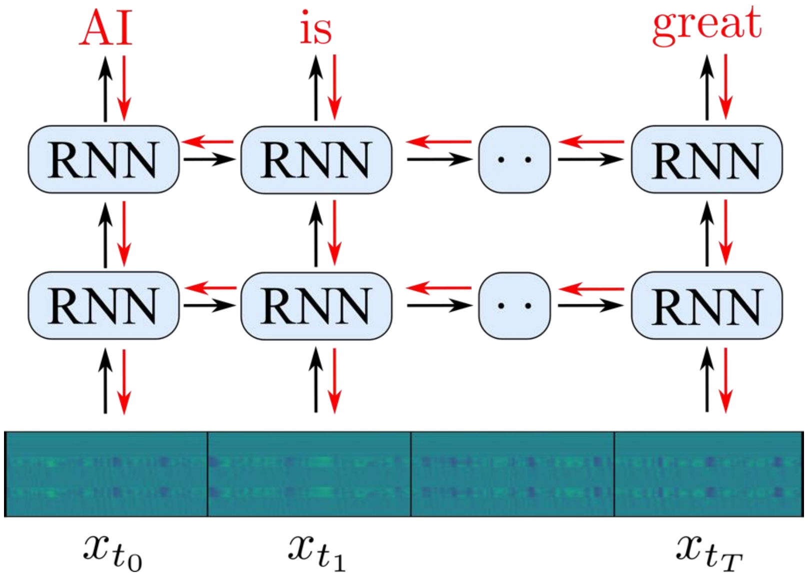
The backward path using Error Backpropagation through time (BPTT)



The backward path using Error Backpropagation through time (BPTT)

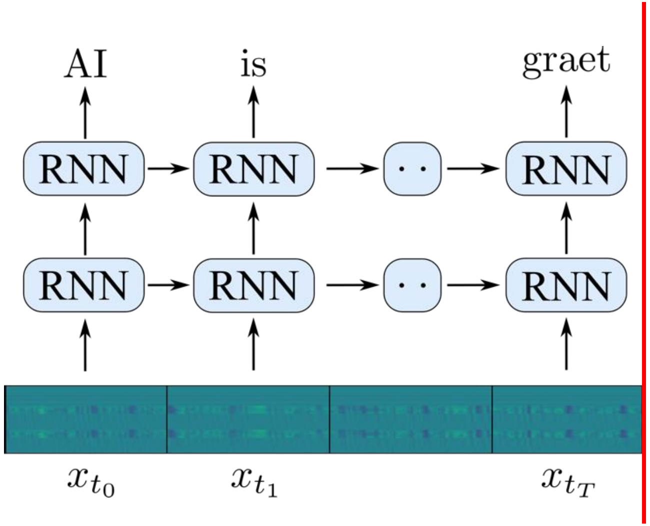


The backward path using Error Backpropagation through time (BPTT)



Backpropagation training suffers from at least three problems

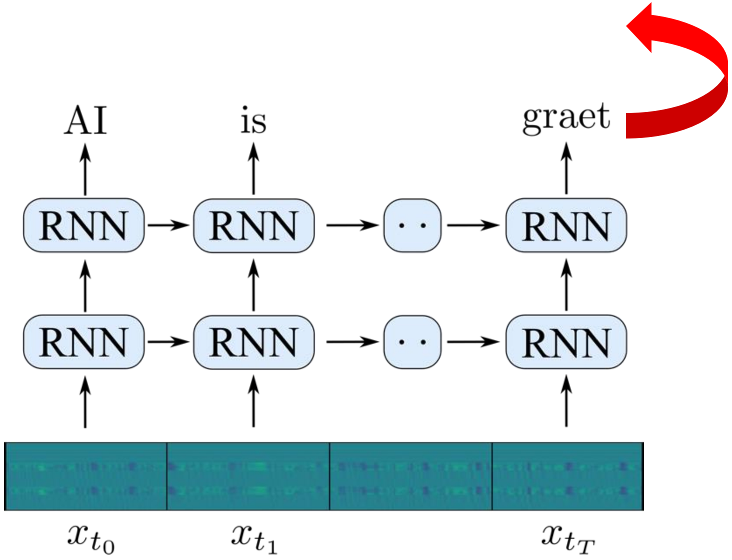
BPTT training suffers from at least three problems



Input sequence needs to be **truncated**

- Not suitable for applications where the end-of-sequence is not known apriori

BPTT training suffers from at least three problems



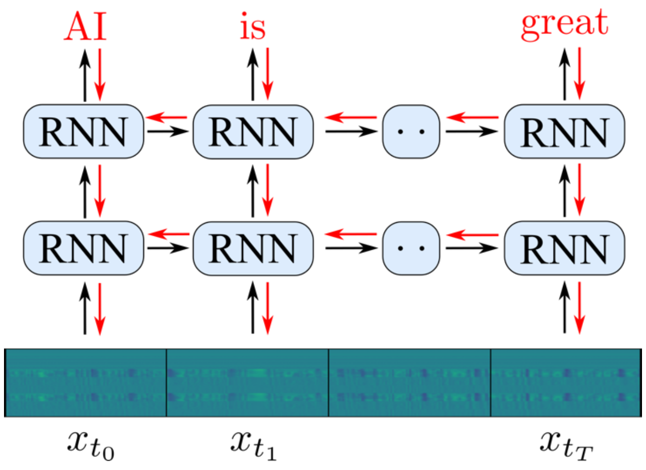
Input sequence needs to be **truncated**

– Not suitable for applications where the end-of-sequence is not known apriori

Forward network operation gets **interrupted**

– Not suitable for applications where continuous learning while receiving new inputs is critical

BPTT training suffers from at least three problems



Input sequence needs to be **truncated**

- Not suitable for applications where the end-of-sequence is not known apriori

Forward network operation gets **interrupted**

- Not suitable for applications where continuous learning while receiving new inputs is critical

Memory requirements **grow with time**

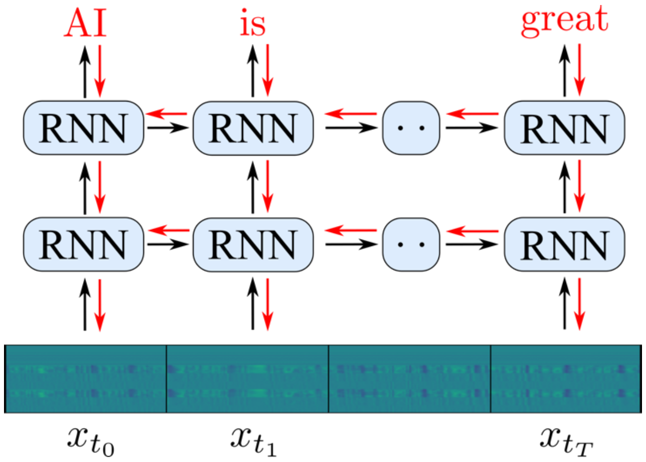
- The memory required to update the parameters of the network grow linearly with the sequence length

Online Spatio-Temporal Learning (OSTL)
as alternative to BPTT

BPTT is a **gradient-based** training algorithm

- Parameters $\boldsymbol{\theta}_l$ of neural network are modified based on the gradients computed by $\frac{\mathbf{d}E}{\mathbf{d}\boldsymbol{\theta}_l}$

$$\frac{\mathrm{d}E}{\mathrm{d}\theta_l} = \sum_{1 \le i \le T} \frac{\partial E^t}{\partial h^{enc}_{t_i,L}} \left(\frac{\partial h^{enc}_{t_i,L}}{\partial s^t_L} \frac{\mathrm{d}s^t_L}{\mathrm{d}\theta_l} + \frac{\partial h^{enc}_{t_i,L}}{\partial \theta_l} \right)$$

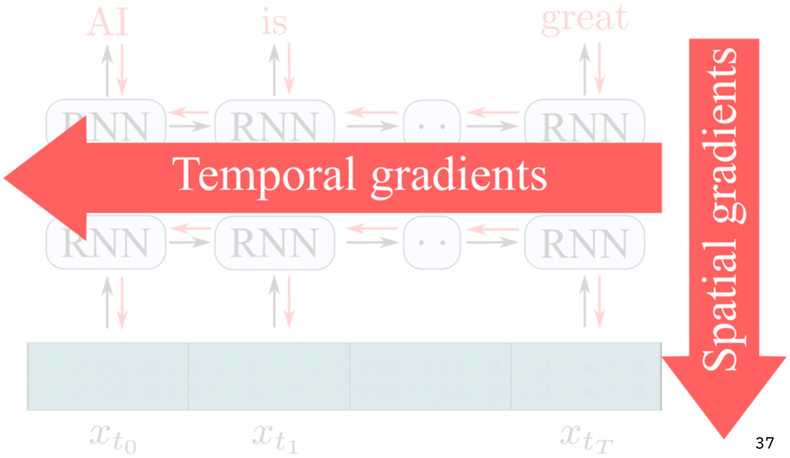


Online Spatio-Temporal Learning (OSTL)
as alternative to BPTT

BPTT is a **gradient-based** training algorithm

– Parameters $\boldsymbol{\theta_l}$ of neural network are modified based on the gradients computed by $\frac{\mathbf{d}E}{\mathbf{d}\boldsymbol{\theta}_l}$

$$\frac{\mathrm{d}E}{\mathrm{d}\theta_l} = \sum_{1 \le i \le T} \frac{\partial E^t}{\partial h^{enc}_{t_i,L}} \left(\frac{\partial h^{enc}_{t_i,L}}{\partial s^t_L} \frac{\mathrm{d}s^t_L}{\mathrm{d}\theta_l} + \frac{\partial h^{enc}_{t_i,L}}{\partial \theta_l} \right)$$



Online Spatio-Temporal Learning (OSTL) as alternative to BPTT

BPTT is a **gradient-based** training algorithm

- Parameters $\boldsymbol{\theta_l}$ of neural network are modified based on the gradients computed by $\frac{\mathbf{d}\boldsymbol{E}}{\mathbf{d}\boldsymbol{\theta_l}}$
- Gradient computations can be rearranged without loss of generality into a combination of Learning signals $\boldsymbol{L_l^t}$ and eligibility traces $\boldsymbol{e_l^{t,\theta_l}}$

$$\frac{\mathrm{d}E}{\mathrm{d}\theta_l} = \sum_{1\leq i\leq T} \frac{\partial E^t}{\partial h^{enc}_{t_i,L}} \left(\frac{\partial h^{enc}_{t_i,L}}{\partial s^t_L} \frac{\mathrm{d}s^t_L}{\mathrm{d}\theta_l} + \frac{\partial h^{enc}_{t_i,L}}{\partial \theta_l} \right)$$

$$\frac{\mathrm{d}E}{\mathrm{d}\theta_l} = \sum_{1\leq i\leq T} \boxed{L^t_l e^{t,\theta_l}_l} + R$$

Online Spatio-Temporal Learning (OSTL) as alternative to BPTT

BPTT is a **gradient-based** training algorithm

– Parameters θ_l of neural network are modified based on the gradients computed by $\frac{\mathrm{d}E}{\mathrm{d}\theta_l}$

– Gradient computations can be rearranged without loss of generality into a combination of Learning signals L_l^t and eligibility traces e_l^{t,θ_l}

Eligibility traces represent **temporal gradients**

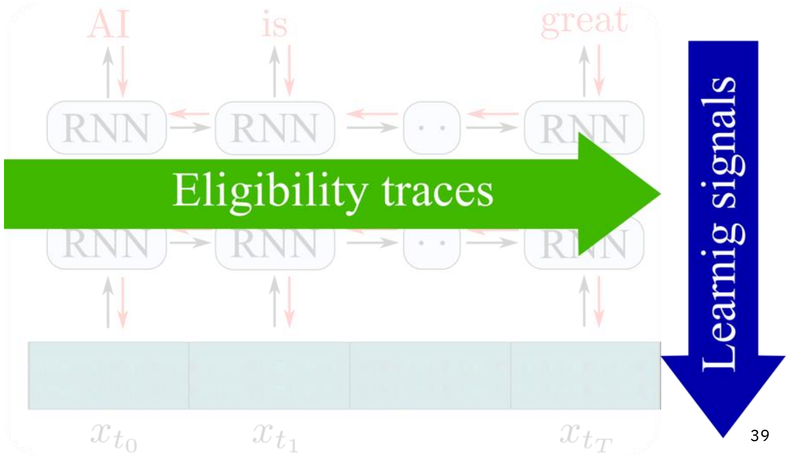
– Can be seen as activity information that every synapse maintains over time

Learning signals represent **spatial gradients**

– Can be seen as teaching signals from the environment targeting neurons

$$\frac{\mathrm{d}E}{\mathrm{d}\theta_l} = \sum_{1\le i\le T} \frac{\partial E^t}{\partial h_{t_i,L}^{enc}} \left(\frac{\partial h_{t_i,L}^{enc}}{\partial s_L^t} \frac{\mathrm{d}s_L^t}{\mathrm{d}\theta_l} + \frac{\partial h_{t_i,L}^{enc}}{\partial \theta_l} \right)$$

$$\frac{\mathrm{d}E}{\mathrm{d}\theta_l} = \sum_{1\le i\le T} L_l^t e_l^{t,\theta_l} + R$$



Online Spatio-Temporal Learning (OSTL) as alternative to BPTT

BPTT is a **gradient-based** training algorithm

- Parameters θ_l of neural network are modified based on the gradients computed by $\frac{\mathrm{d}E}{\mathrm{d}\theta_l}$
- Gradient computations can be rearranged without loss of generality into a combination of Learning signals L_l^t and eligibility traces e_l^{t,θ_l}

Eligibility traces represent **temporal gradients**

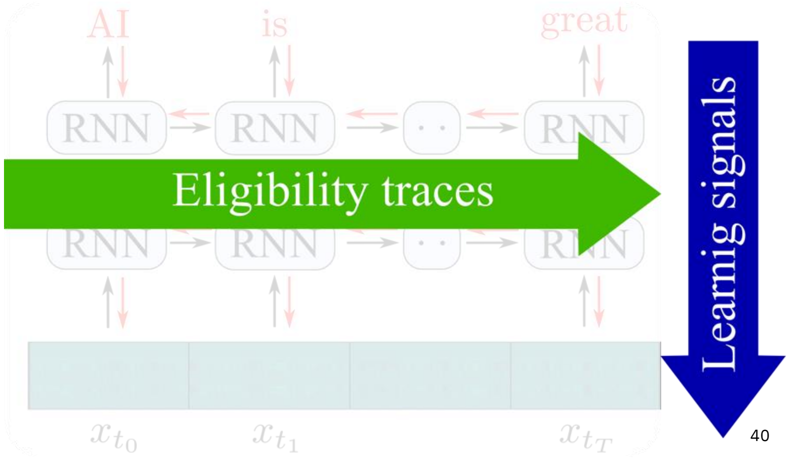
- Can be seen as activity information that every synapse maintains over time

Learning signals represent **spatial gradients**

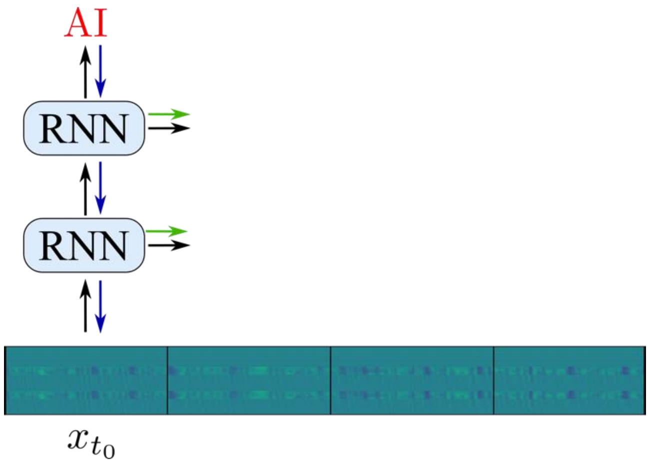
- Can be seen as teaching signals from the environment targeting neurons

$$\frac{\mathrm{d}E}{\mathrm{d}\theta_l} = \sum_{1 \leq i \leq T} \frac{\partial E^t}{\partial h^{enc}_{t_i,L}} \left(\frac{\partial h^{enc}_{t_i,L}}{\partial s^t_L} \frac{\mathrm{d}s^t_L}{\mathrm{d}\theta_l} + \frac{\partial h^{enc}_{t_i,L}}{\partial \theta_l} \right)$$

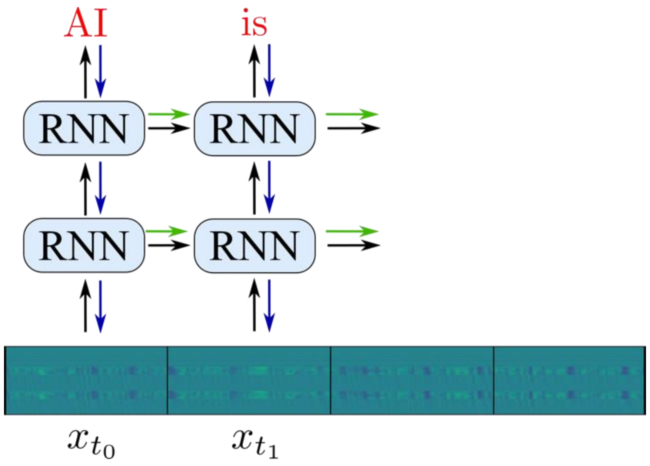
$$\frac{\mathrm{d}E}{\mathrm{d}\theta_l} = \sum_{1 \leq i \leq T} L_l^t e_l^{t,\theta_l} + \boxed{\cancel{R}}$$



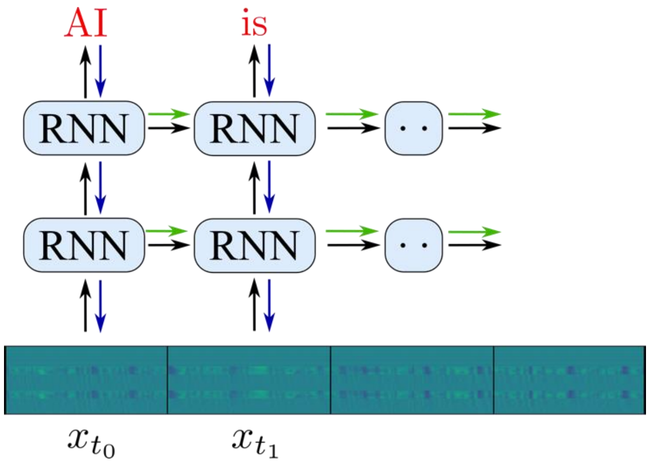
Training procedure with OSTL



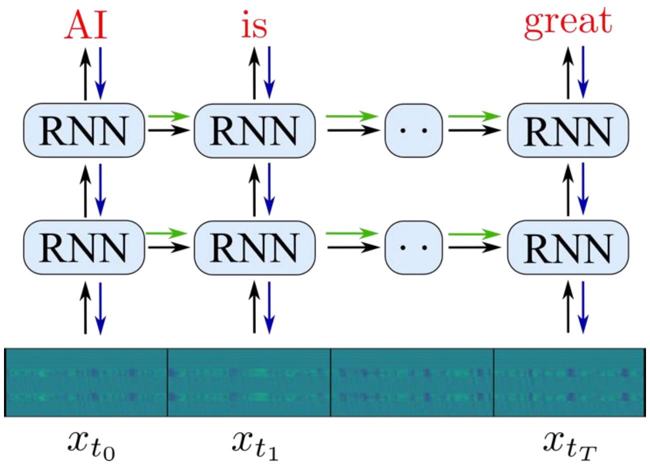
Training procedure with OSTL



Training procedure with OSTL



Training procedure with OSTL



Input sequence **does not need** to be truncated

- Suitable for applications where the end-of-sequence is not known apriori

Forward network operation does not get interrupted

- Suitable for applications where continuous learning while receiving new inputs is critical

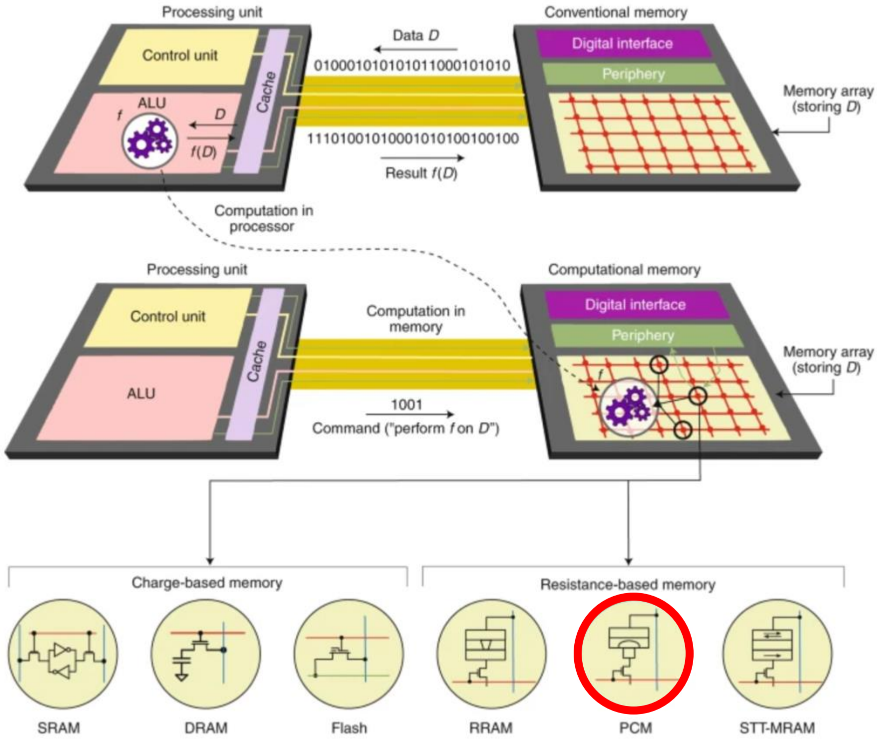
Constant memory requirements*

Compatible with any RNN → We will show a demo

T. Bohnstingl et al. *TNNLS* 2020
https://doi.org/10.1109/TNNLS.2022.3153985

In-memory computing using neuromorphic hardware

Compute in-memory



A. Sebastian et al., *Nature Nanotechnology* 2020
https://doi.org/10.1038/s41565-020-0756-8

Phase-Change Memory (PCM)

A nanometric volume of phase change material placed between two electrodes

– Different geometries possible, so-called mushroom cells are commonly used

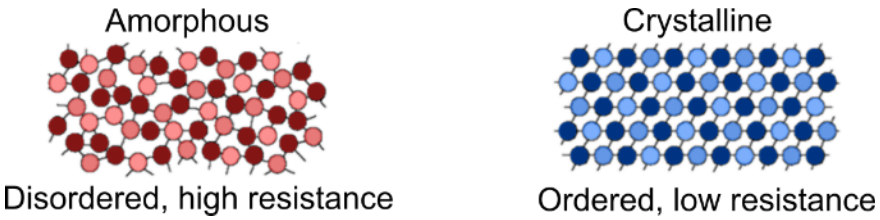
Information is stored in terms of the atonic arrangements (phase configuration)

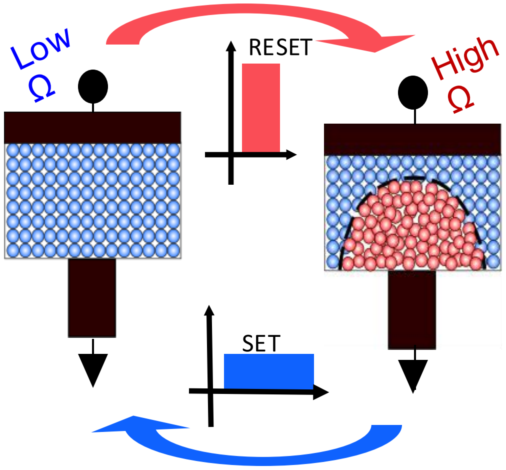
– **Amorphous phase**: highly disordered and high resistive

– **Polycrystalline phase**: highly ordered and low resistive

PCM is essentially an analog storage device

– Non-idealities limit the amount of resistance levels





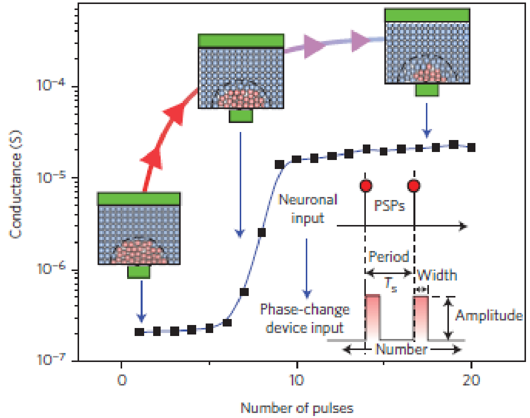
Spiking Neurons can be realized with PCM devices

The neuronal **membrane potential** of an artificial neuron is stored using PCM devices

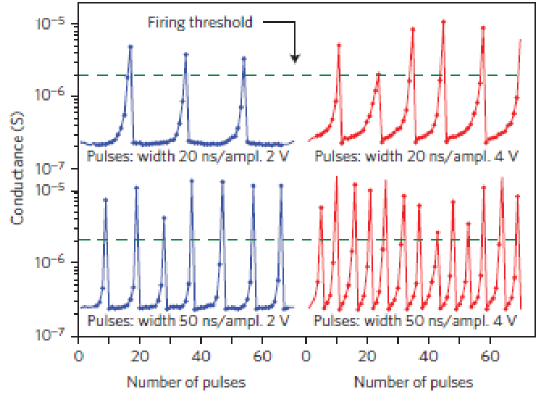
I&F dynamics emulated by the physical properties of the device

Stochasticity enables computation using populations of phase-change neurons

“Integrate… ” by successive application of crystallizing pulses



“… and fire” after reaching a conductance threshold. Then the device is reset.



T. Tuma et al., *Nature Nanotech* 2016
https://doi.org/10.1038/nnano.2016.70

SNUs and in-memory computing

Easy integration of SNUs into emerging in-memory computing architectures

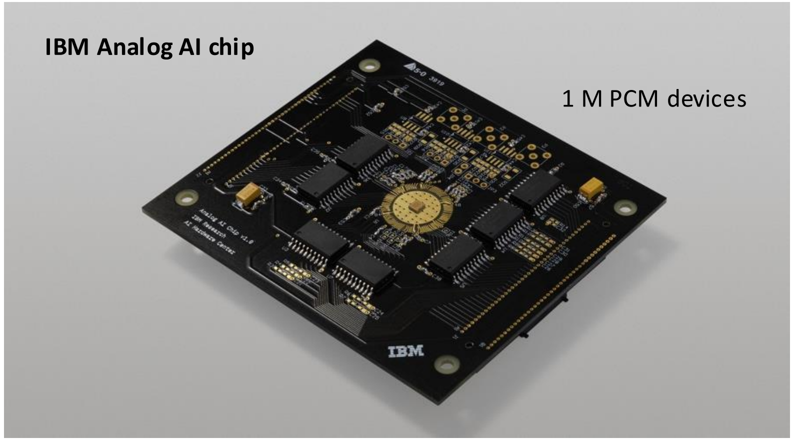
– Weights of SNU network represented with PCM devices

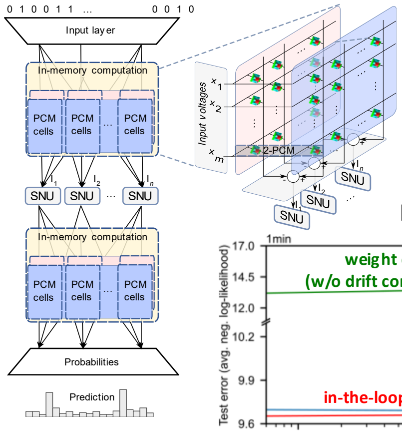
Training with **hardware-in-the-loop** compensates for imperfections

– Noise and drift effects can largely be alleviated

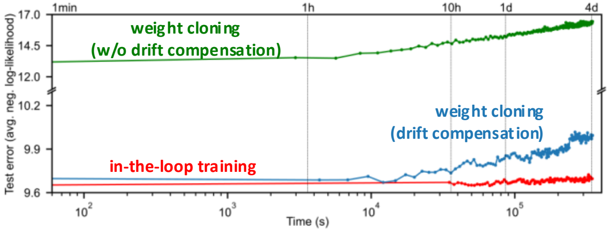
Unified HW design approach supporting both ANNs and SNNs

– Neuromorphic hardware hosts digital processing unit





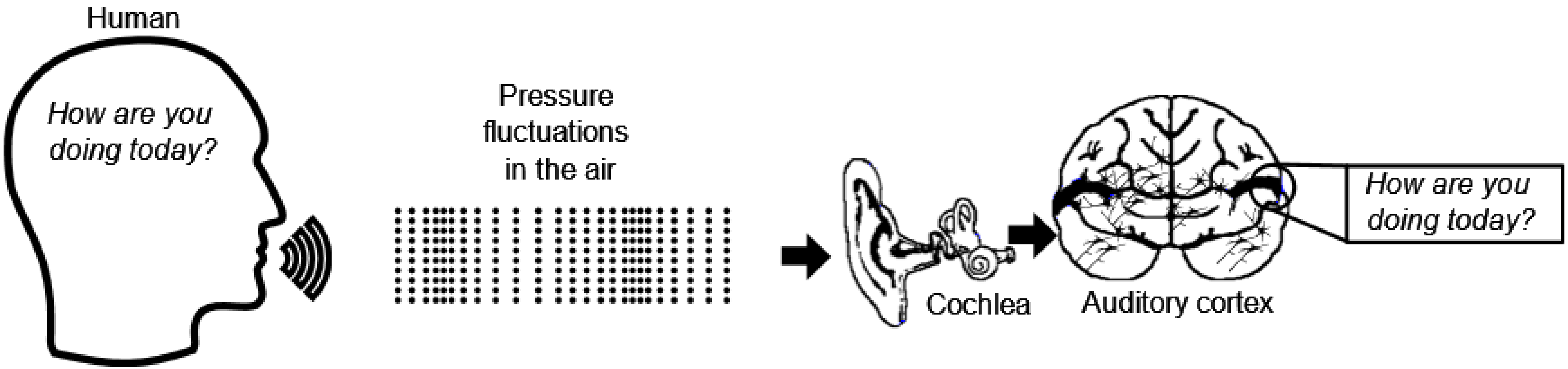
Music prediction inference over 4 days



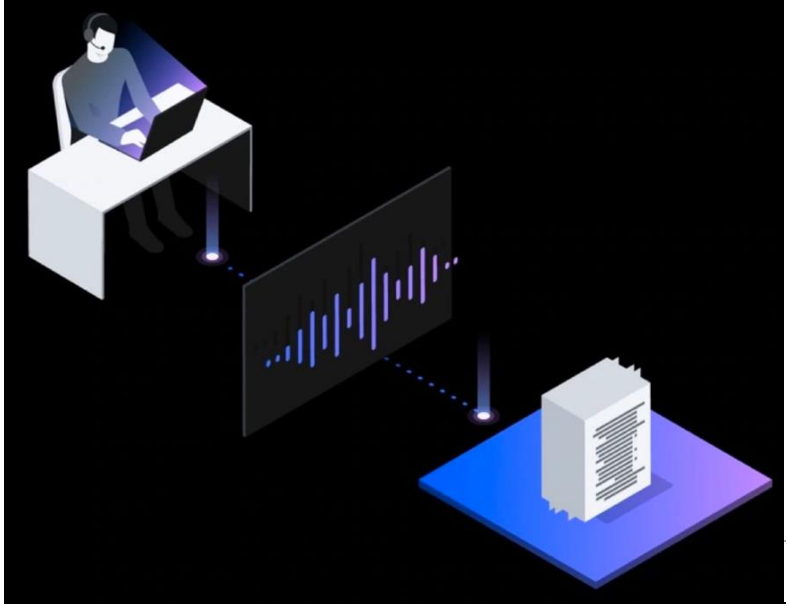
S. Woźniak et al., *Nature Machine Intelligence* 2020
https://doi.org/10.1038/s42256-020-0187-0
M. Le Gallo et al. *Nature Electronics* 2023
https://doi.org/10.1038/s41928-023-01010-1

Application example
Speech recognition

Speech presents the most natural way for humans to communicate



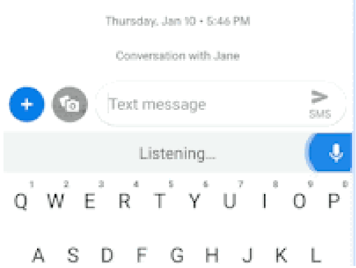
Vast number of usecases – Big challenge for machines



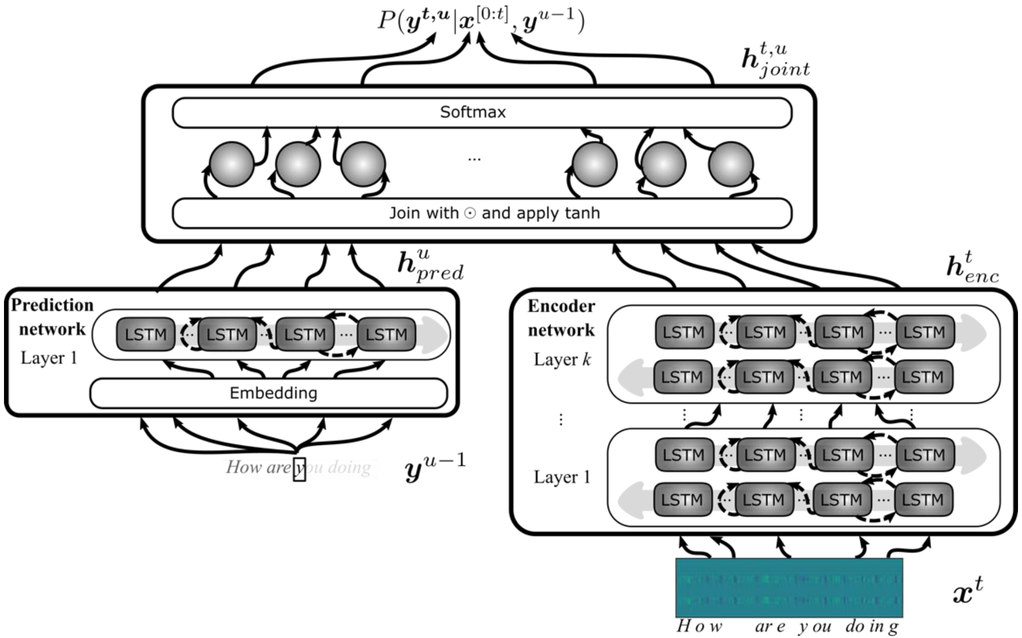
Machine learning approaches face severe challenges

- Large datasets are required
- High computational cost for training
- Need to deal with harsh environments

On device voice



Recurrent Neural Network Transducer (RNN-T) – A state-of-the-art machine learning network



Common architecture in machine learning

- Sequence-to-sequence transduction
- Suited for low-latency applications
- Deployed in cloud services and on hand-held devices

Encoder network acts as feature encoder

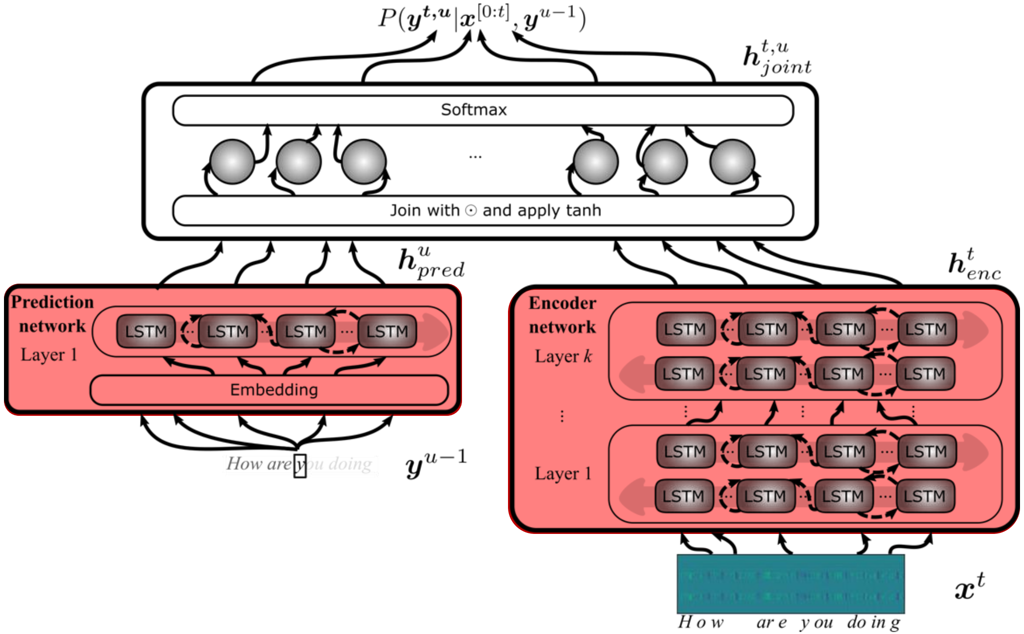
- 6 layers of bidirectional LSTMs

Prediction network acts as language model

- 1 layer of unidirectional LSTMs

A. Graves, *arXiv 2012*
https://doi.org/10.48550/arXiv.1211.3711

Recurrent Neural Network Transducer (RNN-T) – A state-of-the-art machine learning network



Common architecture in machine learning

- Sequence-to-sequence transduction
- Suited for low-latency applications
- Deployed in cloud services and on hand-held devices

Encoder network acts as feature encoder

- 6 layers of bidirectional LSTMs

Prediction network acts as language model

- 1 layer of unidirectional LSTMs

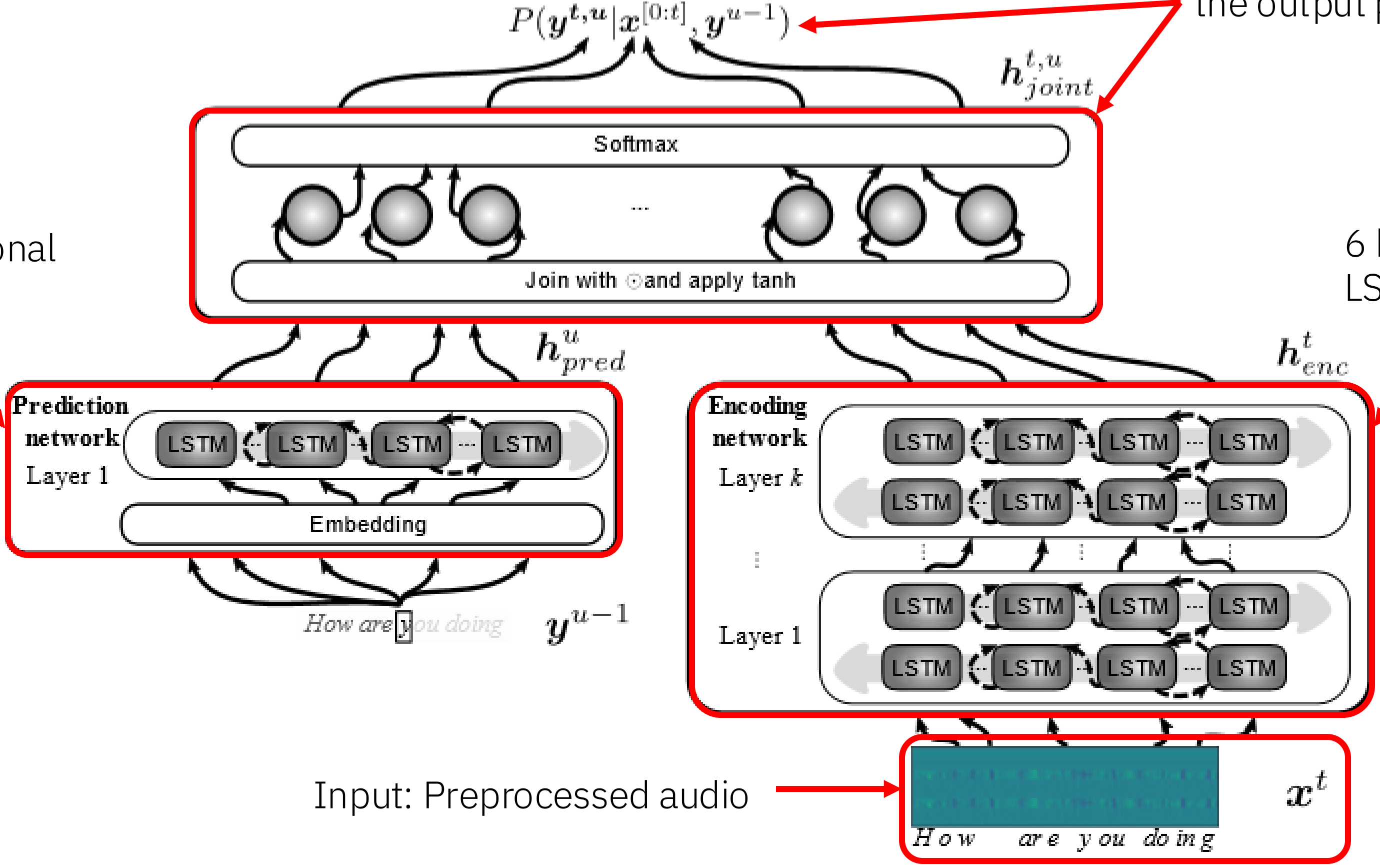
RNN-T architecture

Output: Probability of a speech transcript

Joint network producing the output probabilities

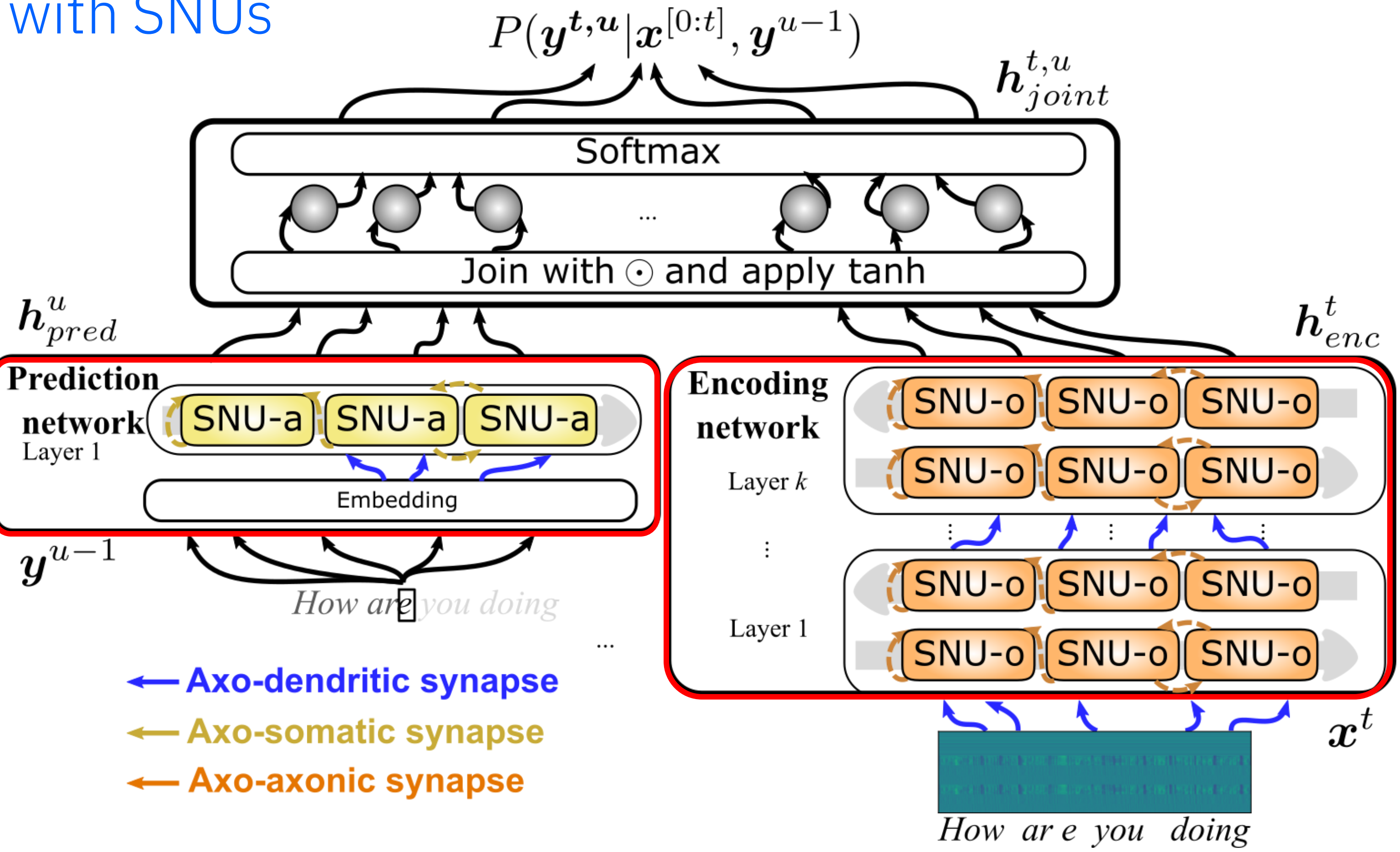
1 uni-directional LSTM layer

6 bi-directional LSTM layers

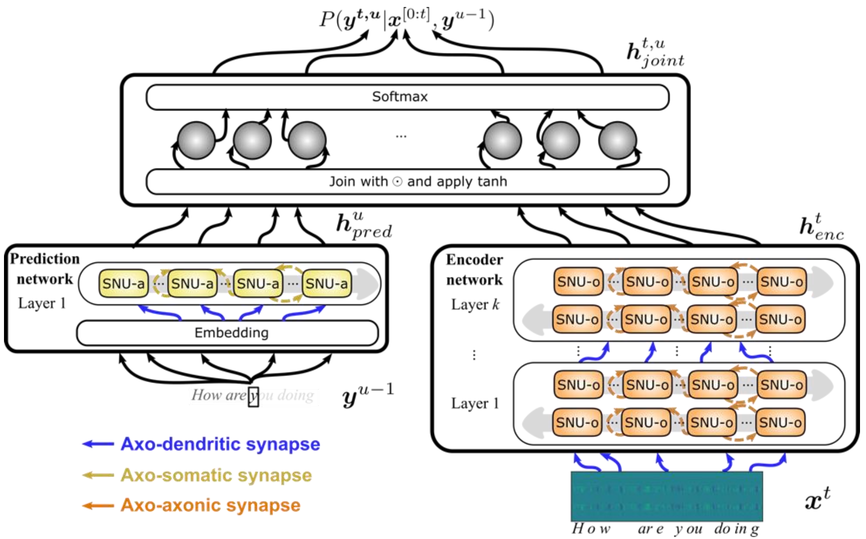


Input: Preprocessed audio

RNN-T architecture
solely with SNUs



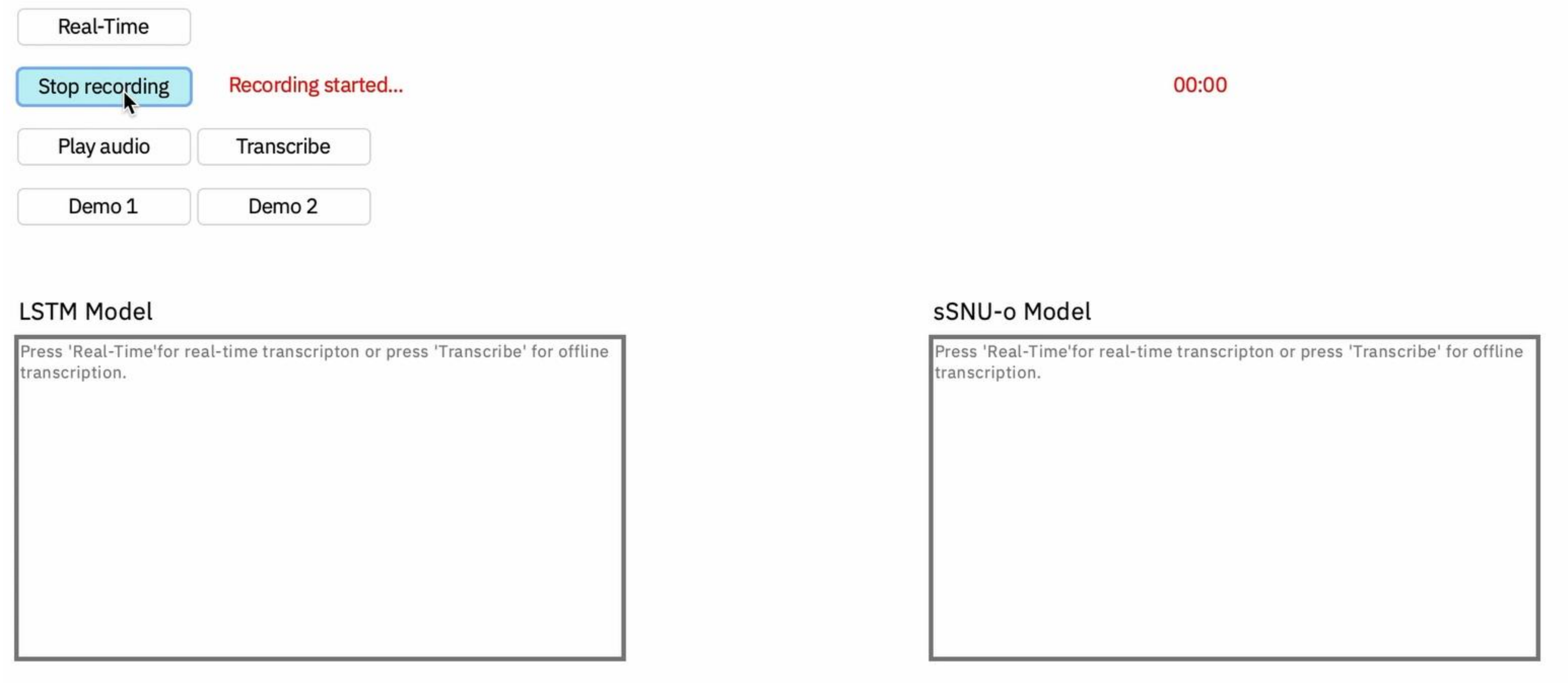
Neural diversity reduces computational cost of RNN-T



Prediction	Encoder	WER (%)	# Multiplications	t_{inf} (s)
LSTM	LSTM	12.7	56M	2.78
SNU-a	LSTM	12.0 (-0.7%)	55M	2.78
LSTM	SNU-o	14.7 (+2.0%)	29M (-48%)	1.76 (-37%)
SNU-o	SNU-o	14.9 (+2.2%)	28.3M (-50%)	1.66 (-40%)

T. Bohnstingl et al. *ICASSP* 2022
https://doi.org/10.1109/ICASSP43922.2022.9747499

Speech-to-Text Demo – SNUs outperform LSTMs



Conclusion

The SNU allows to incorporate dynamics from biology into deep learning

Biology leverages a wide variety of mechanisms for compute

– Diverse types of neuron and synapses provide richer network dynamics

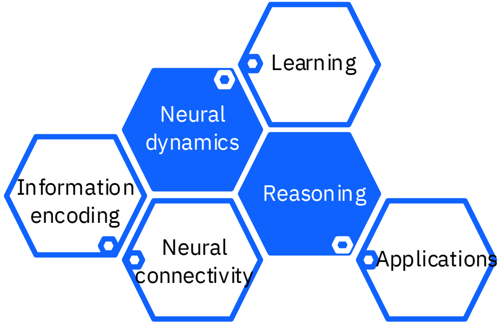
Biology employs more efficient information encoding schemes

Neuroscience can enhance state-of-the-art learning algorithms

Biologically-inspired neural networks can work with large-scale machine learning models

– Diverse types of neuron and synapses provide richer network dynamics

NeuroAI Toolkit



https://research.ibm.com/projects/neuromorphic-computing

All modelled in the NeuroAI Toolkit

Acknowledgements

Emerging Computing and Circuits team of Dr. Angeliki Pantazi

IBM Research colleagues

Collaborators: EPFL Lausanne, TU Graz, ETH Zurich, University of Zurich, fortiss

Funding

SMALL project chist-era

FNSNF
SWISS NATIONAL SCIENCE FOUNDATION

IBM