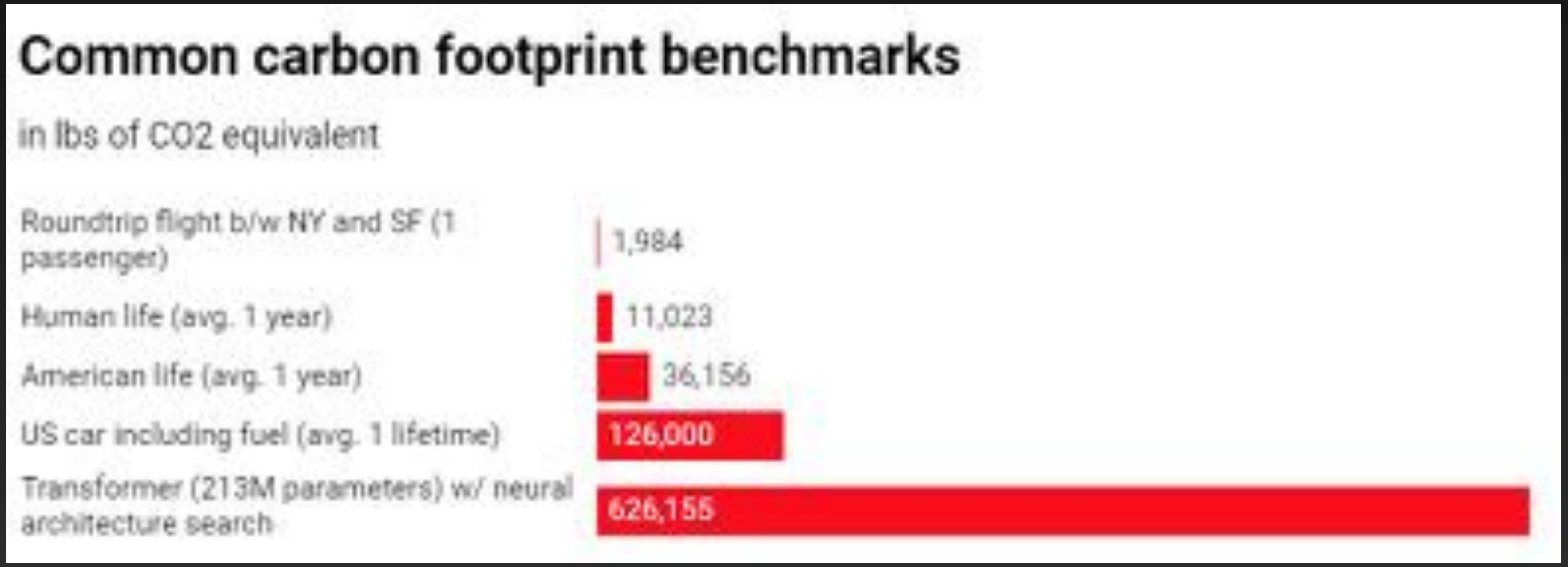
Bio-inspired deep learning

Introduction

To use the principles from nature to lead to better deep learning



Not only better deep learning, but insights into how our own brain works

Let's back up a little

How does nature optimise?

- Life has had a really long time to optimise

Natural selection does not grant organisms what they "need".

- Cost function of life?
- No gradient descent
- It is a random exploration where the fittest survives. However, this is not necessarily the "best"

Bio-inspired technology

Nature is pretty good at optimising

- Biomimicry
- Velcro
- Spider silk-based materials
- Sharkskin
- and on and on....

Bio-inspired algorithms

Yo - they be optimising algorithms too

- Ant colony optimisation

- Particle Swarm optimisation

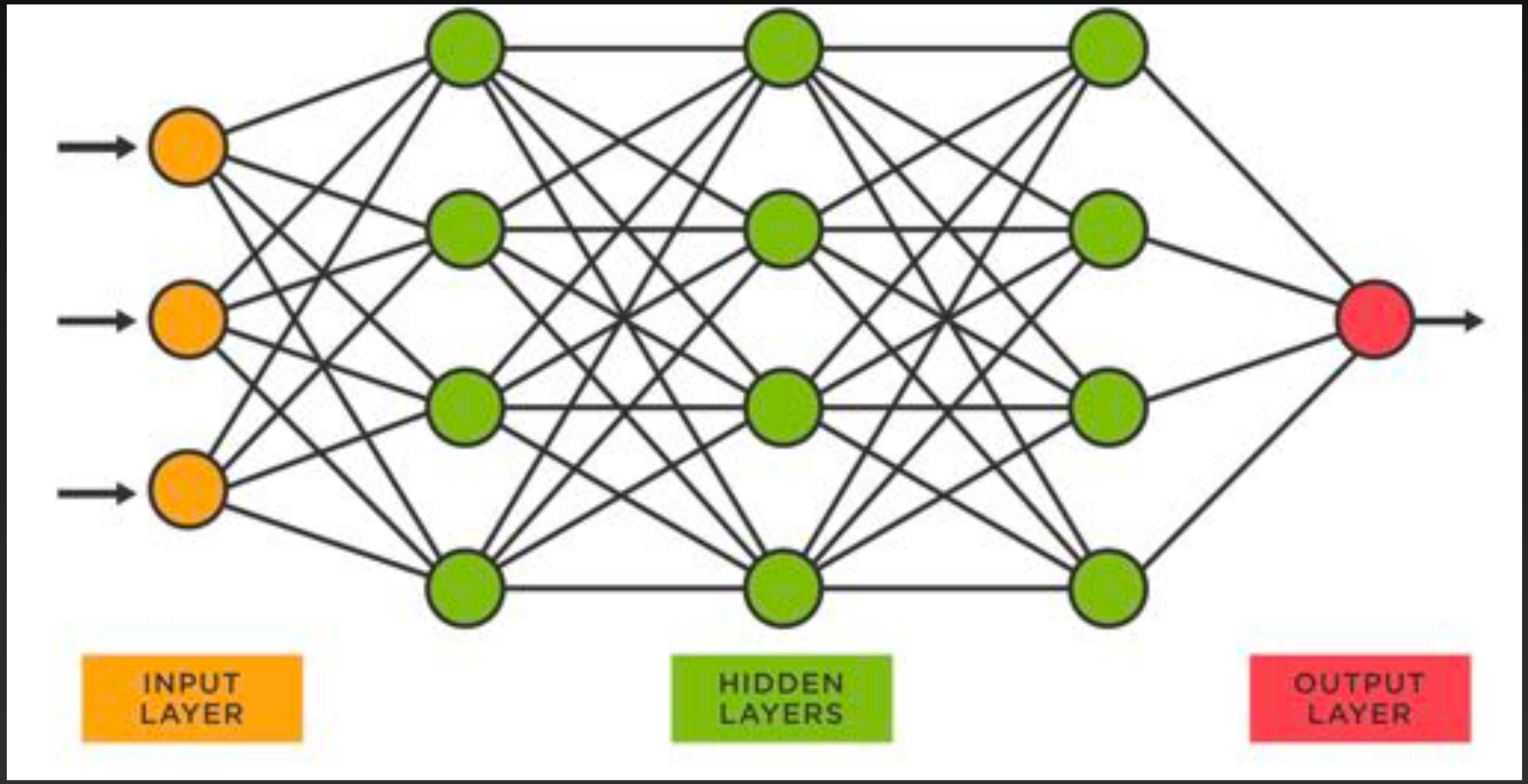
Bio-inspired algorithms

Yo - they be optimising algorithms too

Swarm intelligence based algorithms			Bio-inspired (not SI-based) algorithms		
Algorithm	Author	Reference	Algorithm	Author	Reference
Accelerated PSO	Yang et al.	[69], [71]	Atmosphere clouds model	Yan and Hao	[67]
Ant colony optimization	Dorigo	[15]	Biogeography-based optimization	Simon	[56]
Artificial bee colony	Karaboga and Basturk	[31]	Brain Storm Optimization	Shi	[55]
Bacterial foraging	Passino	[46]	Differential evolution	Storn and Price	[57]
Bacterial-GA Foraging	Chen et al.	[6]	Dolphin echolocation	Kaveh and Farhoudi	[33]
Bat algorithm	Yang	[78]	Japanese tree frogs calling	Hernández and Blum	[28]
Bee colony optimization	Teodorović and Dell'Orco	[62]	Eco-inspired evolutionary algorithm	Parpinelli and Lopes	[45]
Bee system	Lucic and Teodorovic	[40]	Egyptian Vulture	Sur et al.	[59]
BeeHive	Wedde et al.	[65]	Fish-school Search	Lima et al.	[14], [3]
Wolf search	Tang et al.	[61]	Flower pollination algorithm	Yang	[72], [76]
Bees algorithms	Pham et al.	[47]	Gene expression	Ferreira	[19]
Bees swarm optimization	Drias et al.	[16]	Great salmon run	Mozaffari	[43]
Bumblebees	Comellas and Martinez	[12]	Group search optimizer	He et al.	[26]
Cat swarm	Chu et al.	[7]	Human-Inspired Algorithm	Zhang et al.	[80]
Consultant-guided search	Iordache	[29]	Invasive weed optimization	Mehrabian and Lucas	[42]
Cuckoo search	Yang and Deb	[74]	Marriage in honey bees	Abbass	[1]
Eagle strategy	Yang and Deb	[75]	OptBees	Maia et al.	[41]
Fast bacterial swarming algorithm	Chu et al.	[8]	Paddy Field Algorithm	Premaratne et al.	[48]
Firefly algorithm	Yang	[70]	Roach infestation algorithm	Havens	[25]
Fish swarm/school	Li et al.	[39]	Queen-bee evolution	Jung	[30]
Good lattice swarm optimization	Su et al.	[58]	Shuffled frog leaping algorithm	Eusuff and Lansey	[18]
Glowworm swarm optimization	Krishnanand and Ghose	[37], [38]	Termite colony optimization	Hedayatzadeh et al.	[27]
Hierarchical swarm model	Chen et al.	[5]	Physics and Chemistry based algorithms		
Krill Herd	Gandomi and Alavi	[22]	Big bang-big Crunch	Zandi et al.	[79]
Monkey search	Mucherino and Seref	[44]	Black hole	Hatamlou	[24]
Particle swarm algorithm	Kennedy and Eberhart	[35]	Central force optimization	Formato	[21]
Virtual ant algorithm	Yang	[77]	Charged system search	Kaveh and Talatahari	[34]
Virtual bees	Yang	[68]	Electro-magnetism optimization	Cuevas et al.	[13]
Weightless Swarm Algorithm	Ting et al.	[63]	Galaxy-based search algorithm	Shah-Hosseini	[53]
Other algorithms			Gravitational search	Rashedi et al.	[50]
Anarchic society optimization	Shayeghi and Dadashpour	[54]	Harmony search	Geem et al.	[23]
Artificial cooperative search	Civicioglu	[9]	Intelligent water drop	Shah-Hosseini	[52]
Backtracking optimization search	Civicioglu	[11]	River formation dynamics	Rabanal et al.	[49]
Differential search algorithm	Civicioglu	[10]	Self-propelled particles	Vicsek	[64]
Grammatical evolution	Ryan et al.	[51]	Simulated annealing	Kirkpatrick et al.	[36]
Imperialist competitive algorithm	Atashpaz-Gargari and Lucas	[2]	Stochastic difusion search	Bishop	[4]
League championship algorithm	Kashan	[32]	Spiral optimization	Tamura and Yasuda	[60]
Social emotional optimization	Xu et al.	[66]	Water cycle algorithm	Eskandar et al.	[17]

Table 1. A list of algorithms

Neural networks are also bio-inspired!



Convolutional NN

- Convolutional neural networks: subregions of a given space

- Cat visual cortices contain neurons: small regions of the visual field.

RNN

- In biological systems, neurons are connected in a network and can send signals to each other in a loop, creating a feedback loop.
- Implemented in ANNs this allows sequences of data to be remembered as well as previous inputs

Joe's dad has 4 kids. Three of these are called Mary, James and Anna. What is the name of the 4th child?

Reinforcement learning

- Many behavioural studies are performed in this way

- A way for the brain to learn

- Effective way to train ANNs (there are still some issues)

The brain vs the machine

- The human brain is estimated to consume around 20 watts of energy at rest
- Learns and perform multiple tasks
- Based on chemical signalling: slow and imprecise
- Error correcting and self-regenerating

- Large CNN for image recognition can consume as much energy as driving a car for 1,000 miles
- Catastrophic forgetting
- Based on electrical signals: fast and precise
- Cannot correct errors and also not self-regenerating

The brain vs the machine

- Our brains can get tired and experience “fatigue”
- Rewiring occurs regularly
- Brains are constantly learning and adjusting their weights

- Artificial neural networks can keep going as long as they get juice
- Model is fixed
- Artificial neural networks don’t learn by recalling information — they only learn during training

Conclusion: ANNs are biologically inspired - but they are very different from our Brains!

How can we improve the machine?

Structure

- How can we change the structure of the network to better approximate biological principles

Learning (credit assignment)

- How can we update weights during (and maybe after training) to be closer to biological principles

Neurons

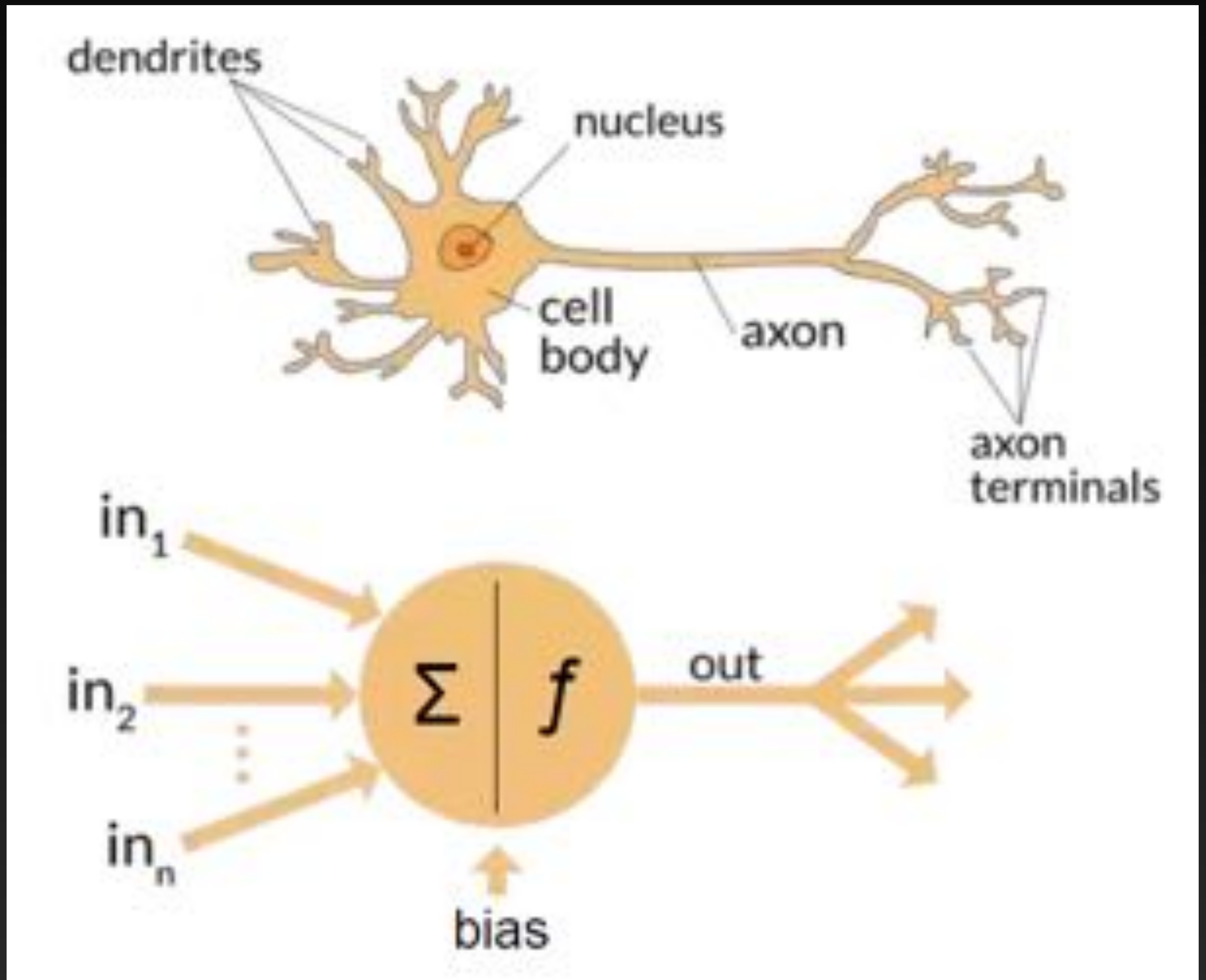
- How can we implement biological neurons in ANNs to reach biological principles?

Dendritic computation

Neurons

Dendritic Computing: Branching Deeper into Machine Learning

Jyotibdha Acharya [a], Arindam Basu [b], Robert Legenstein [c], Thomas Limbacher [c], Panayiota Poirazi [d], Xundong Wu [e 1 2]

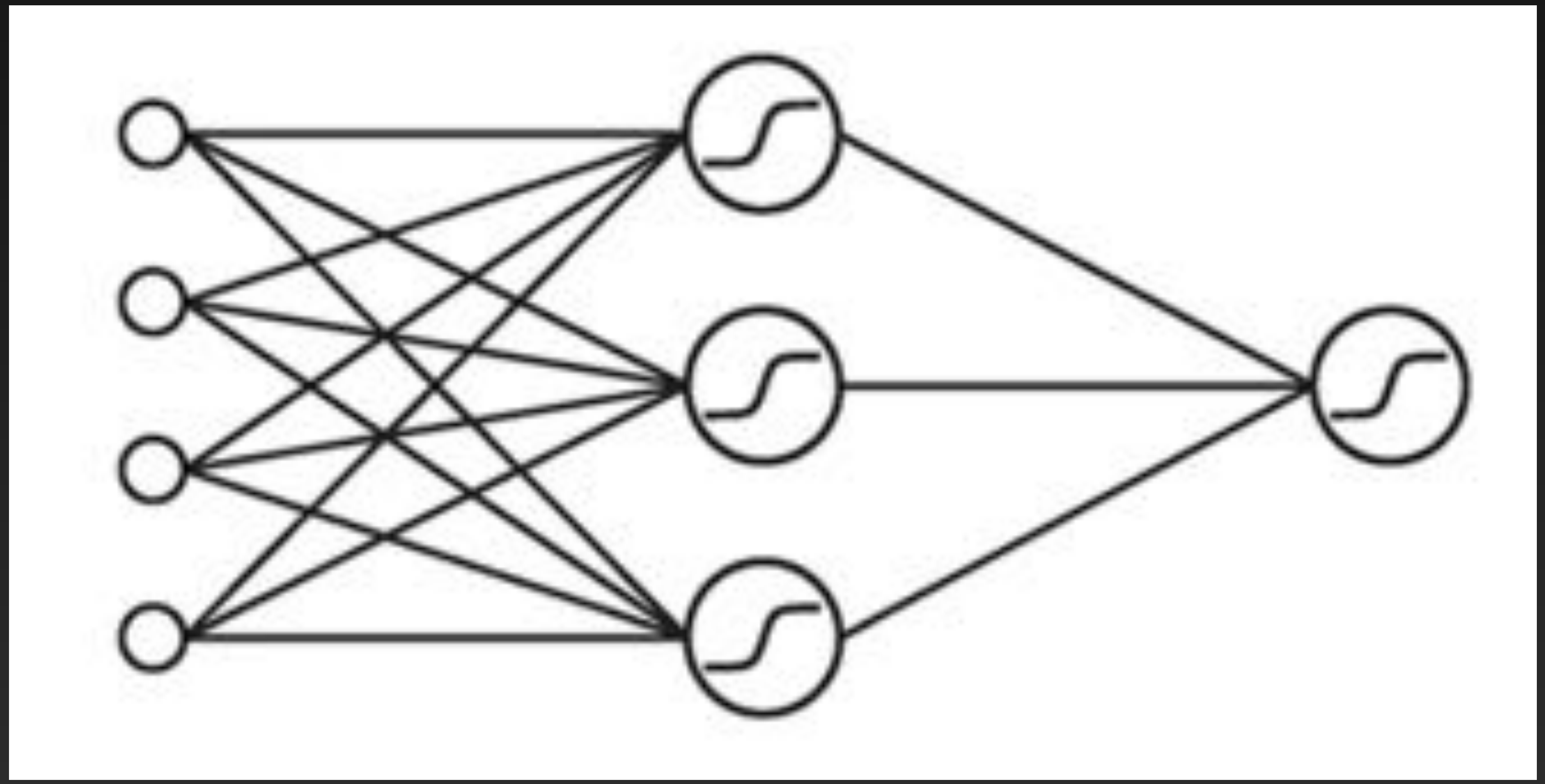


- The current models "…totally ignore[s] any possible computational role played by dendrites, the thin processes that extend from the cell bodies of neurons and serve as their primary receiving end"

Dendritic computation

Neurons

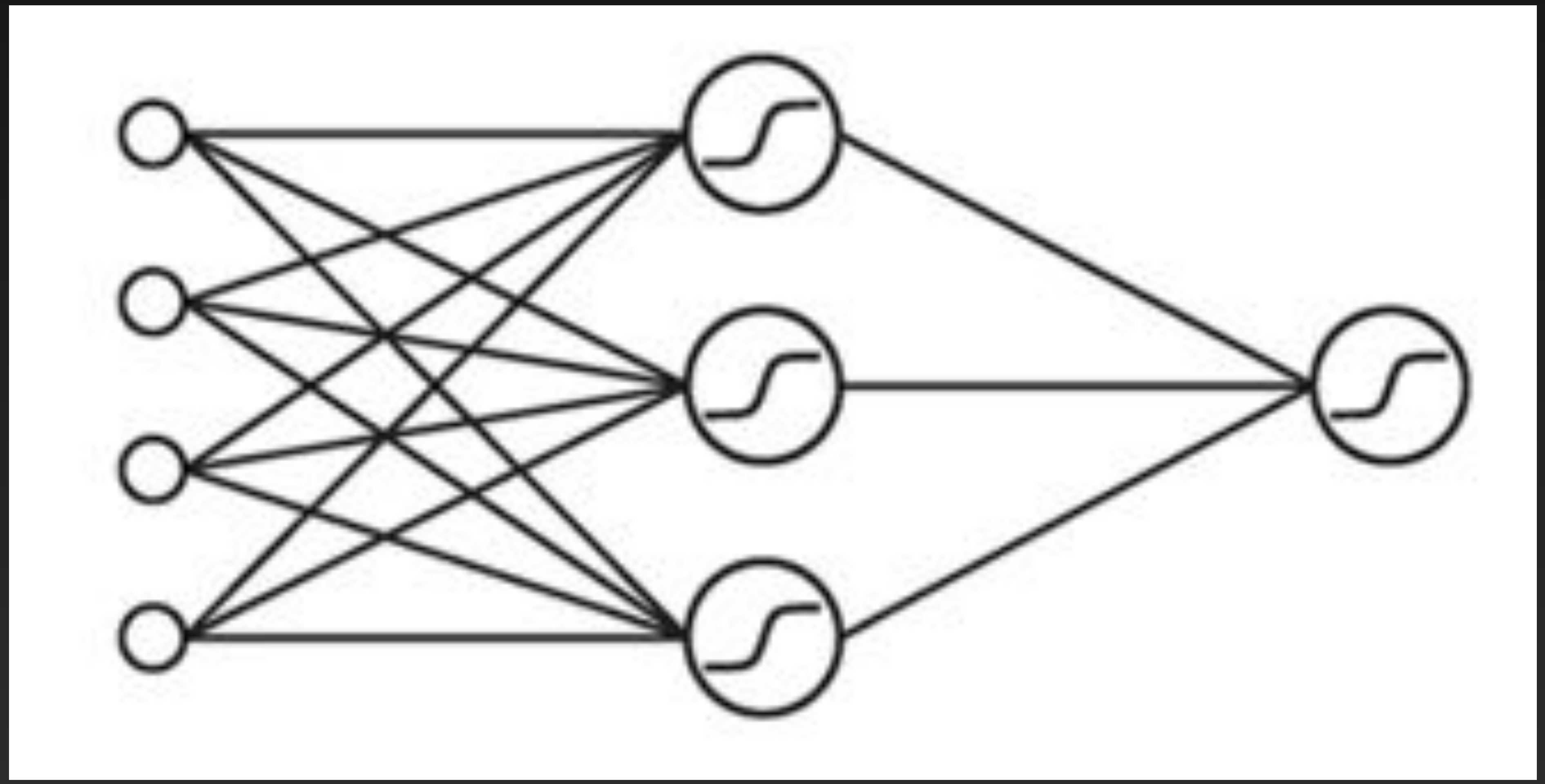
- Improved expressivity of single neurons
- Improved use of neuronal resources and generalization capabilities
- Utilization of internal learning signals
- Enabling continual learning.



Dendritic computation

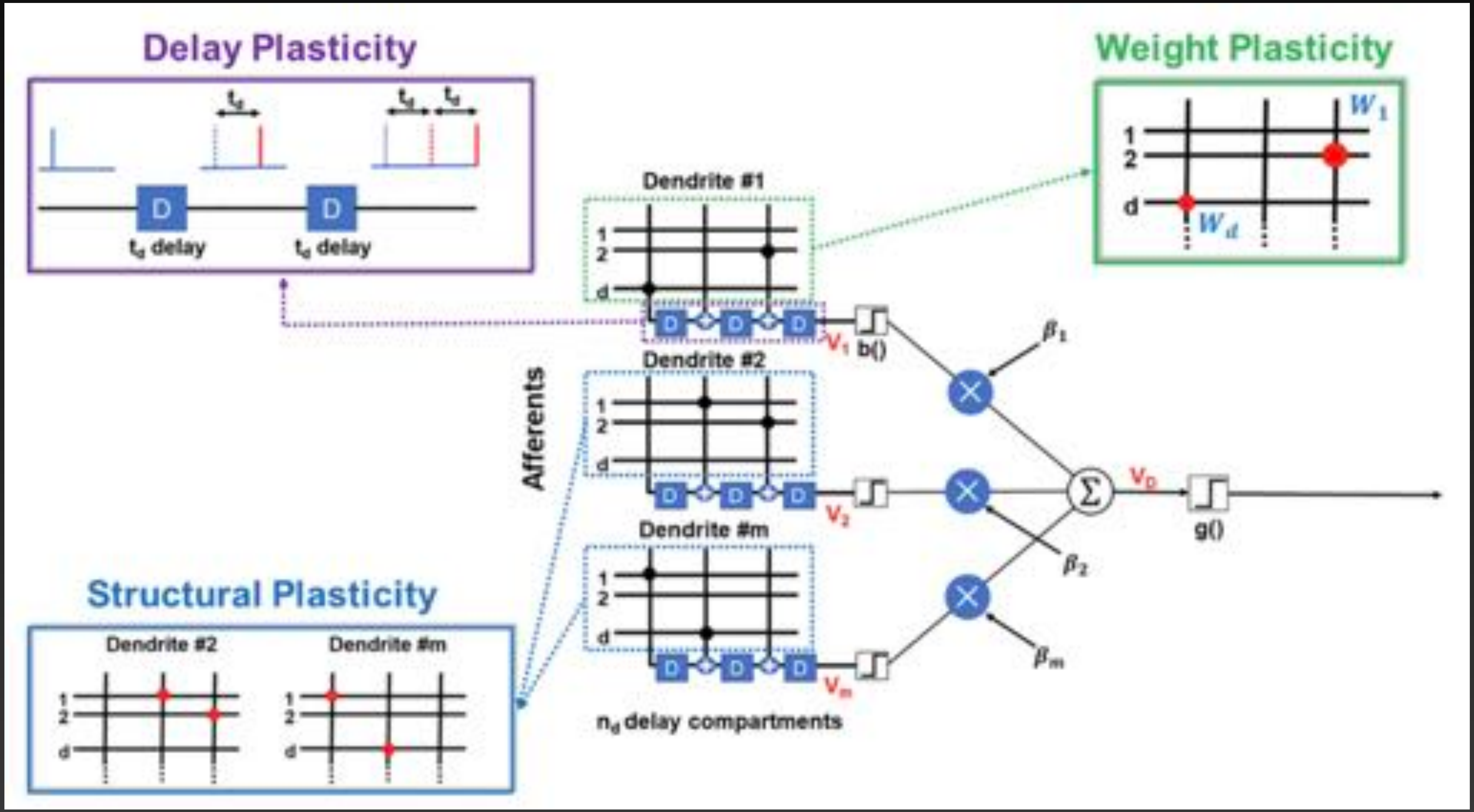
Neurons

- Local voltages: reinforce plasticity of co-active inputs within a dendrite
- The backdrop action: communicate that the neuron is already active
- This can be utilized by plasticity rules to ensure that different branches -> different associations



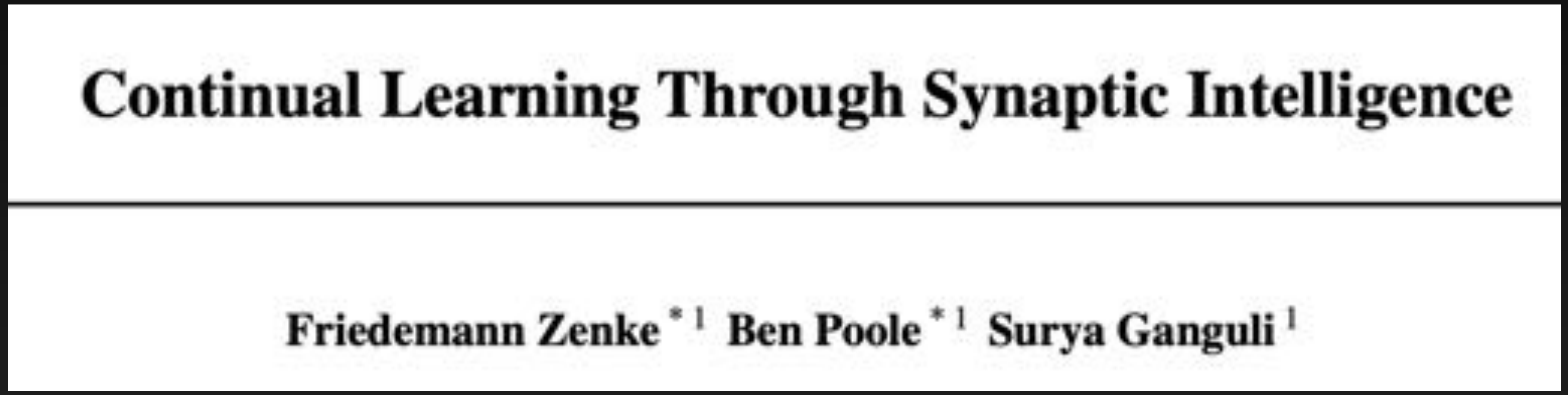
Dendritic computation

Neurons

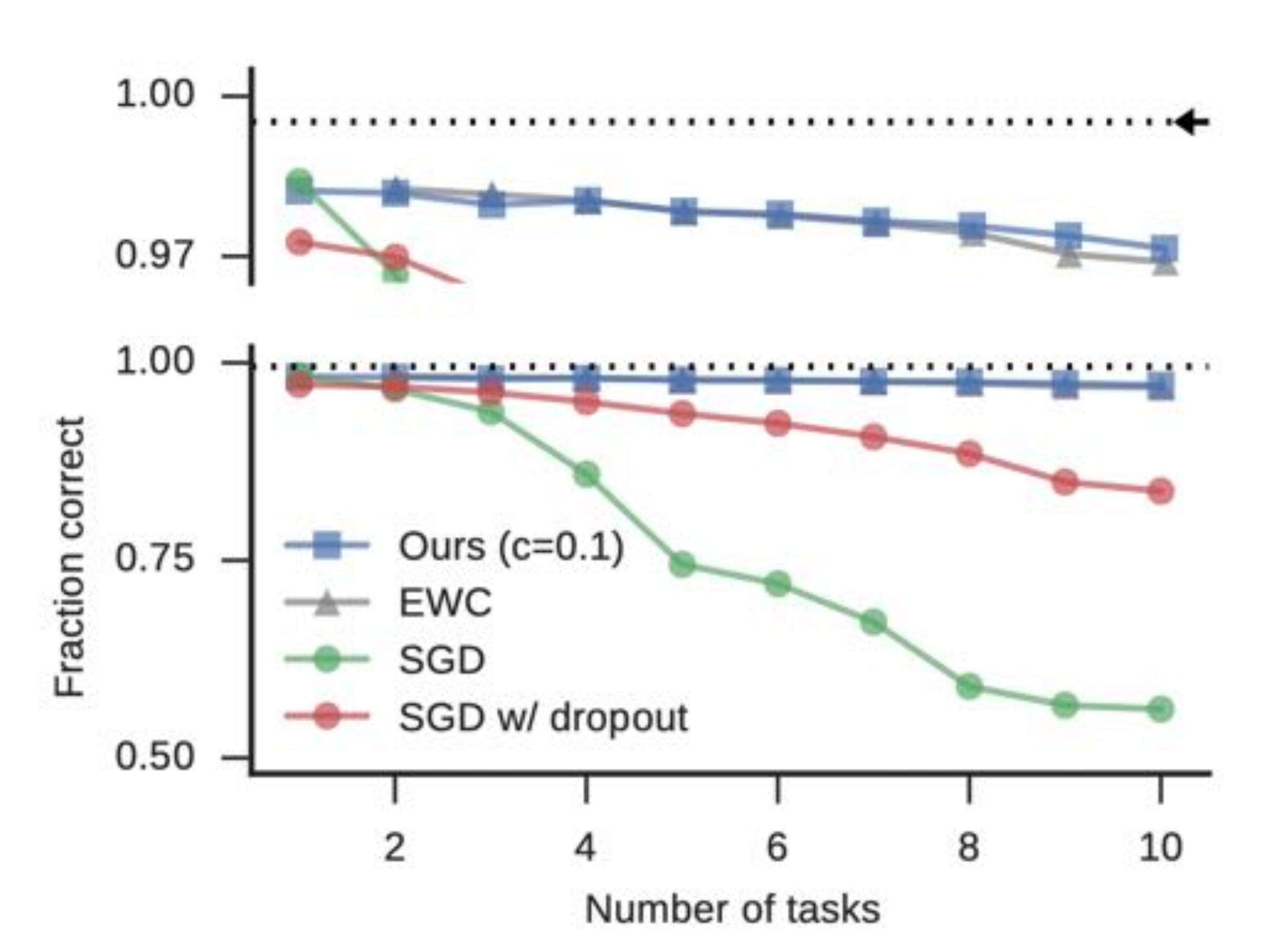


“Intelligent” synapses

- Introduce a function that represents that “importance” of a given synapses for a given task.
- For the next task, penalise changes to those synapses that have high importance on previous tasks

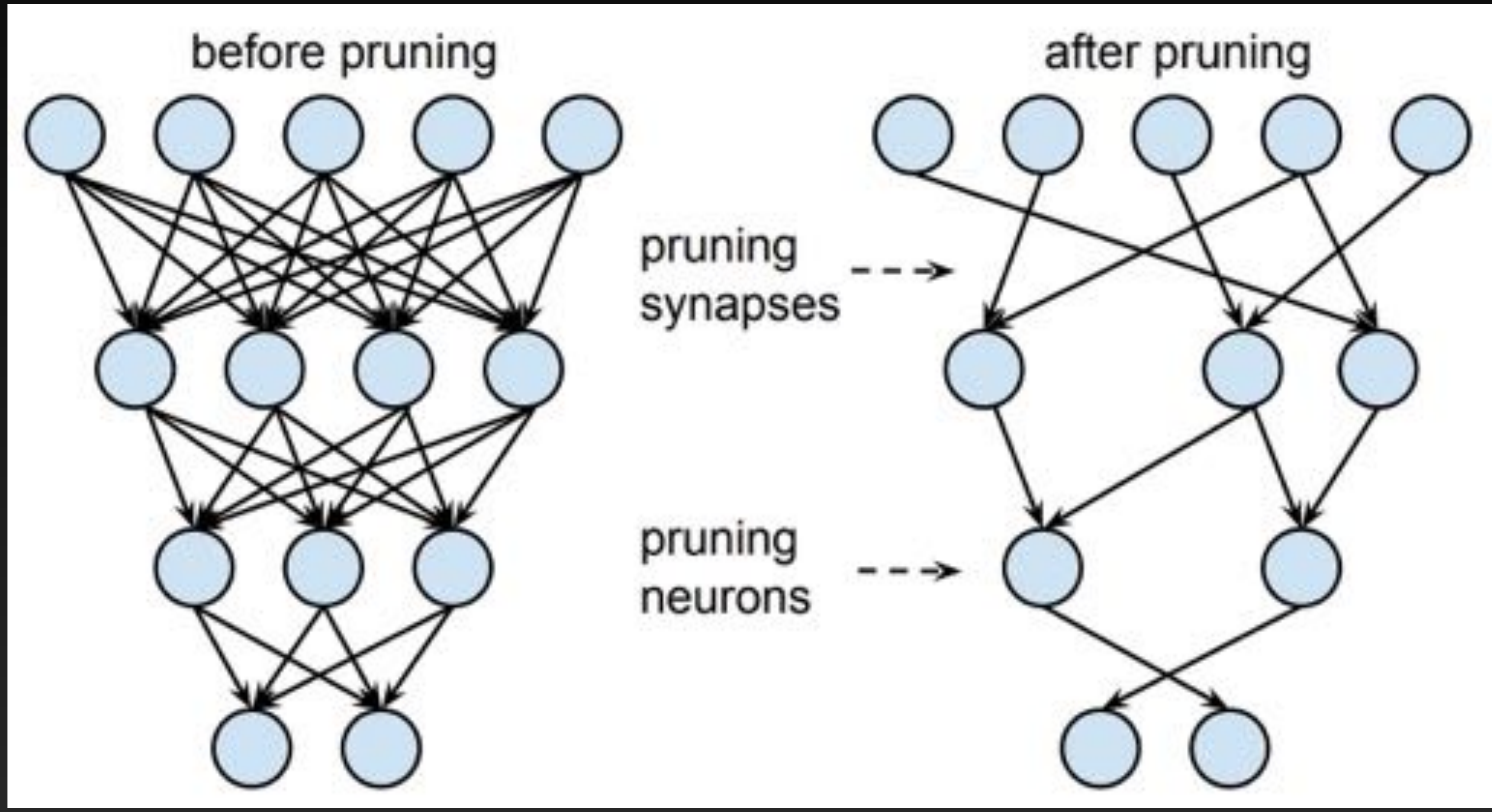

Continual Learning Through Synaptic Intelligence

Friedemann Zenke *[1] Ben Poole *[1] Surya Ganguli [1]



Sparse neural networks

- Only a small fraction of weights are used during inference while the rest remain zero
- Reduces computational cost and memory requirements
- Making it easier to train and interpret, and reducing overfitting.
- More related to biological networks - not everything is connected!



THE LOTTERY TICKET HYPOTHESIS:
FINDING SPARSE, TRAINABLE NEURAL NETWORKS

Jonathan Frankle
MIT CSAIL
jfrankle@csail.mit.edu

Michael Carbin
MIT CSAIL
mcarbin@csail.mit.edu

Sparsity and rewiring
Structure

DEEP REWIRING: TRAINING VERY SPARSE DEEP NETWORKS

Guillaume Bellec, David Kappel, Wolfgang Maass & Robert Legenstein
Institute for Theoretical Computer Science
Graz University of Technology
Austria
{bellec,kappel,maass,legenstein}@igi.tugraz.at

- A sparse network, where new connections can go dormant or be activated
- Weight update is combined with a random walk
- In this setup, like in neurobiology, the sign of a weight does not change during learning

Sparsity and rewiring
Structure

DEEP REWIRING: TRAINING VERY SPARSE DEEP NETWORKS

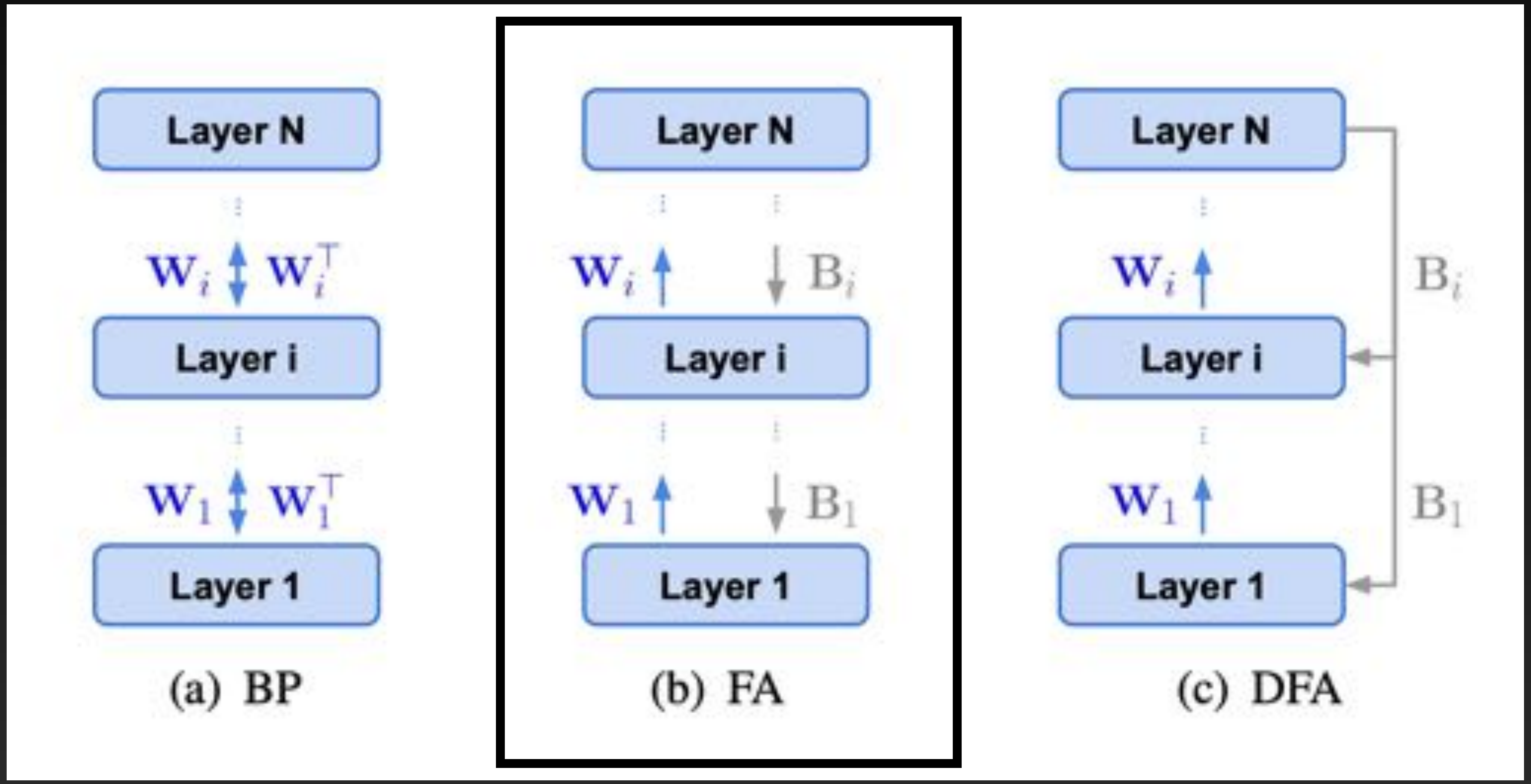
Guillaume Bellec, David Kappel, Wolfgang Maass & Robert Legenstein
Institute for Theoretical Computer Science
Graz University of Technology
Austria
{bellec,kappel,maass,legenstein}@igi.tugraz.at

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

Algorithm 1: Pseudo code of the DEEP R algorithm. ν_k is sampled from a zero-mean Gaussian of unit variance independently for each active and each update step. Note that the gradient of the error $E_{\mathbf{X},\mathbf{Y}^*}(\boldsymbol{\theta})$ is computed by backpropagation over a mini-batch in practice.

Feedback alignment

- Random fixed matrices are used to propagate error signals from the output layer to the hidden layers
- Teaching signal should on average lie within 90° of backprop
- We push the network in roughly the same direction as backprop would.
- Problems with large networks



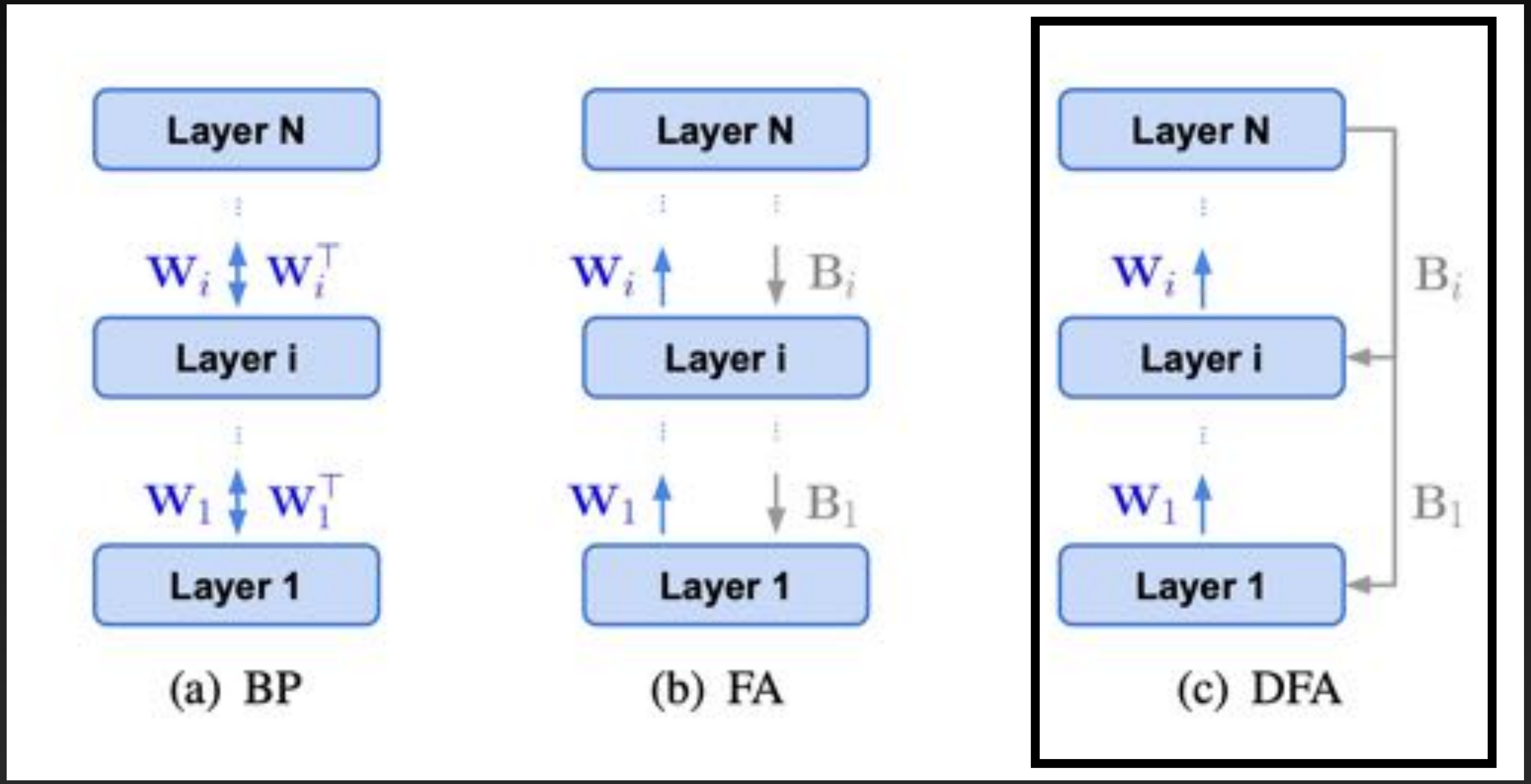
Random synaptic feedback weights support error backpropagation for deep learning

Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed & Colin J. Akerman

Feedback alignment

- Error signals are computed using a fixed random matrix and the output layer activations, rather than the true target values

- Better performance on large networks



Direct Feedback Alignment Provides Learning in Deep Neural Networks

Arild Nøkland
Trondheim, Norway
arild.nokland@gmail.com

HyperNetworks
(as an aside)

- Use one network (hypernetwork) to shape the weights of another network
- Biologically plausible, due to hierarchical structure of the brain
- Smaller hypernetwork is easier to train
- The goal of training the hypernetwork is to find a set of weights -> better convergence of main network

HYPERNETWORKS

David Ha,* Andrew Dai, Quoc V. Le
Google Brain
{hadavid,adai,qvl}@google.com

BAYESIAN HYPERNETWORKS

David Krueger[†‡*] Chin-Wei Huang[†*] Riashat Islam[§] Ryan Turner[†]
Alexandre Lacoste[‡] Aaron Courville[†||]
* Equal contributors
[†]Montreal Institute for Learning Algorithms (MILA)
[‡]ElementAI [§]McGill University [||]CIFAR Fellow
david.krueger@umontreal.ca chin-wei.huang@umontreal.ca
riashat.islam@mail.mcgill.ca turnerry@iro.umontreal.ca
allac@elementai.com courvila@iro.umontreal.ca

CONTINUAL LEARNING WITH HYPERNETWORKS

Johannes von Oswald*, Christian Henning*, Benjamin F. Grewe, João Sacramento
*Equal contribution

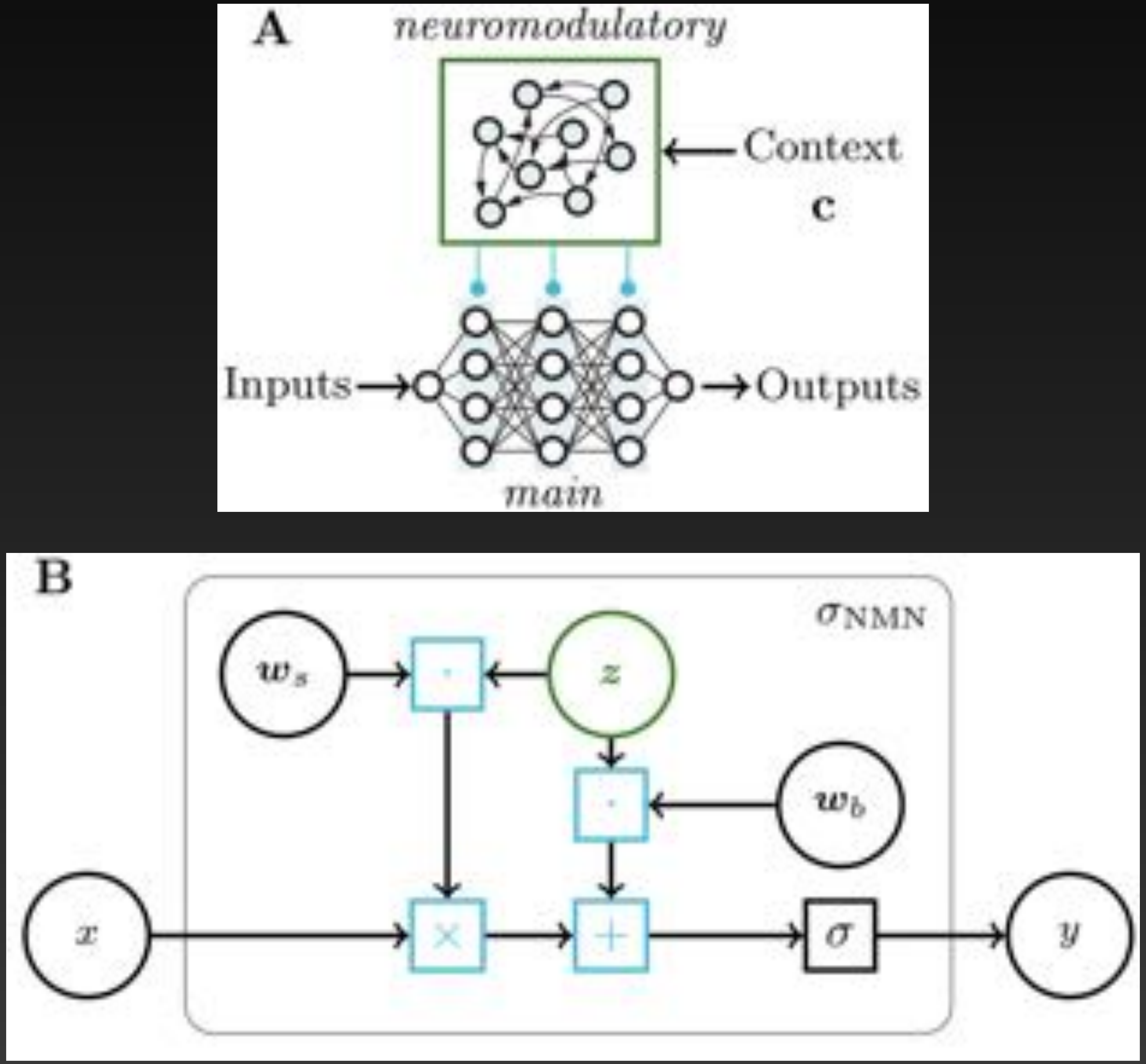
Institute of Neuroinformatics
University of Zürich and ETH Zürich
Zürich, Switzerland
{voswaldj,henningc,bgrewe,rjoao}@ethz.ch

Neuromodulation

Introducing neuromodulation in deep neural networks to learn adaptive behaviours

Nicolas Vecoven, Damien Ernst, Antoine Wehenkel, Guillaume Drion

- Neuromodulation is the use of chemical signals to modulate the activity of neurons in the brain.
- The task of the NM is to tune the activation functions of the main network
- Important in many learning and memory processes in the brain



Difference target propagation

Local error-driven learning is a learning rule that updates the weights of a neural network based on local error signals that are computed at each neuron.

- Associate with each feedforward unit's activation value a *target value* rather than a *loss gradient*.
- Learning proceeds by updating the forward weights to minimize these local layer-wise activity differences
- It can be shown that under certain conditions the updates approximate those that would have been prescribed by backprop.

Difference Target Propagation

Dong-Hyun Lee[1], Saizheng Zhang[1], Asja Fischer[1], and Yoshua Bengio[1,2]

[1] Université de Montréal, Quebec, Canada
[2] CIFAR Senior Fellow

$$f_i(g_i(\mathbf{h}_i)) \approx \mathbf{h}_i \quad \text{or} \quad g_i(f_i(\mathbf{h}_{i-1})) \approx \mathbf{h}_{i-1}$$

Local error learning

Local error-driven learning is a learning rule that updates the weights of a neural network based on local error signals that are computed at each neuron.

- Boltzmann machines: The network is trained using an unsupervised learning algorithm
  - maximize the likelihood of the training data.
- GeneRec's: simple and local
  - each synaptic weight change should be proportional to the difference between the product of the presynaptic and postsynaptic activities from the positive and negative phases.

A Learning Algorithm for Boltzmann Machines*

DAVID H. ACKLEY
GEOFFREY E. HINTON
Computer Science Department
Carnegie-Mellon University
TERRENCE J. SEJNOWSKI
Biophysics Department
The Johns Hopkins University

Biologically Plausible Error-driven Learning using Local Activation Differences:
The Generalized Recirculation Algorithm

Randall C. O'Reilly

Forward-forward
Learning

- Greedy multi-layer learning procedure inspired by Boltzmann machines
- The aim of the learning is to make the goodness be well above some threshold value for real data and well below that value for negative data.
- Normalisation between layers to force next layers to use information in the relative activities of the previous layers
- Does not generalize as well as backpropogation, and not suited for large networks

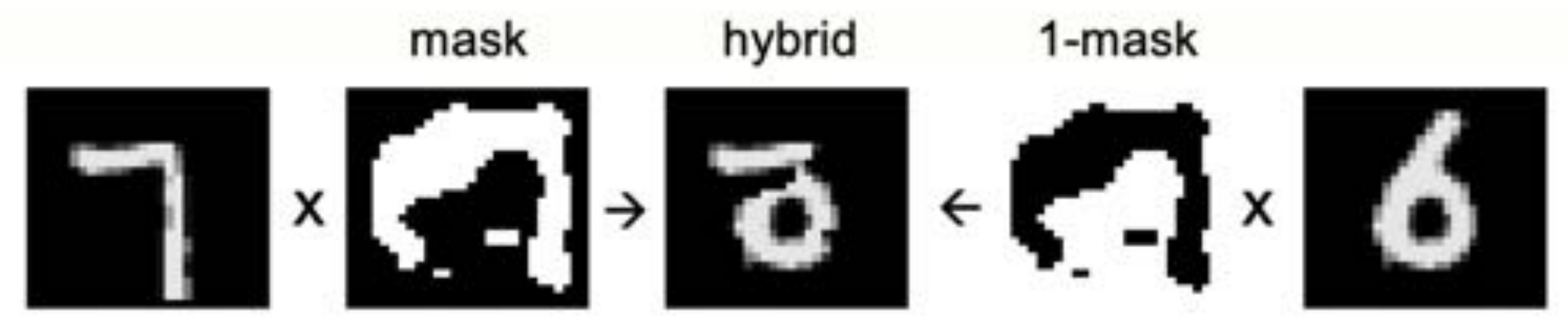


Figure 1: A hybrid image used as negative data

Forward-forward

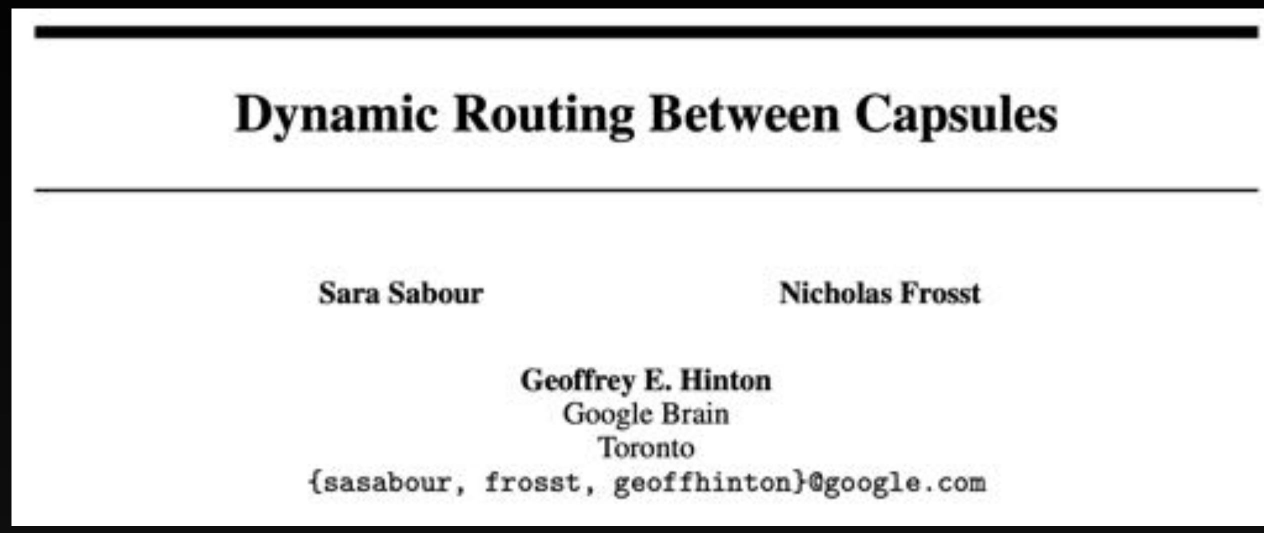
Biological interpretation

- Analog to brain:
  - positive data processed when awake
  - negative data processed during sleep (and created when awake)

"I reported that it was possible to do multiple updates on positive data followed by multiple updates on negative data, generated by the net itself, with very little loss in performance. I have been unable to replicate this result and I now suspect it was due to a bug."

- A different goodness function would allow the positive and negative phases

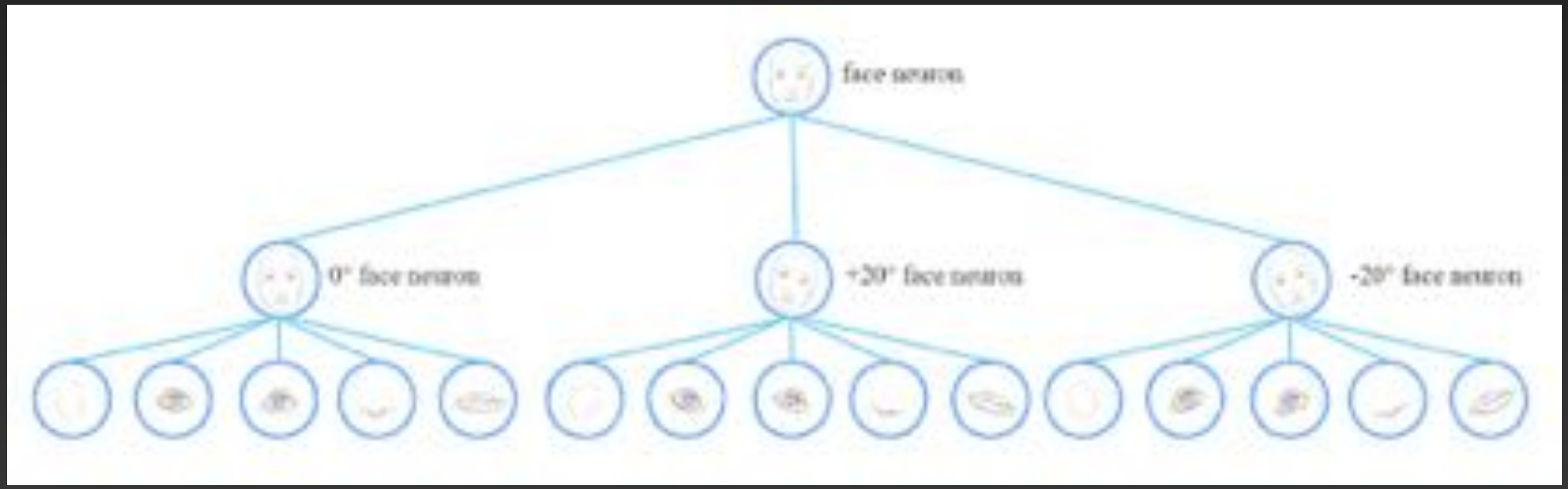
Capsule Neural Networks

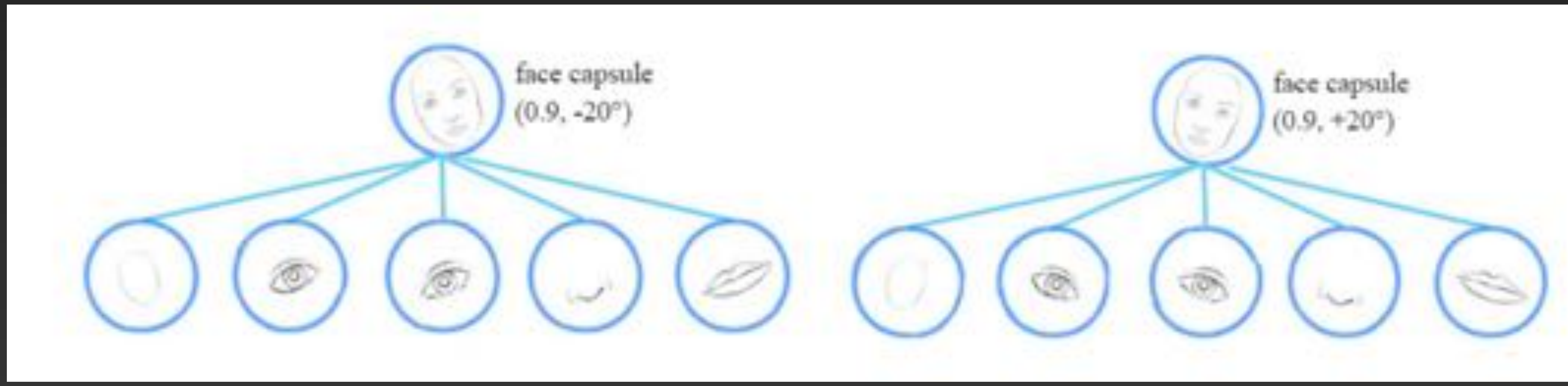

Dynamic Routing Between Capsules

Sara Sabour   Nicholas Frosst

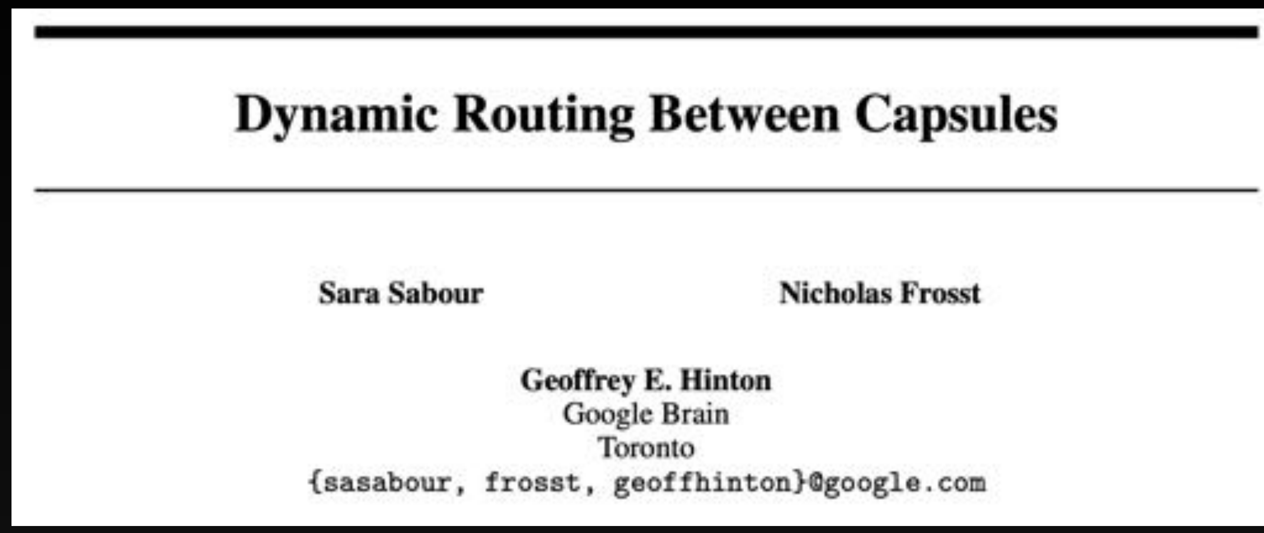
Geoffrey E. Hinton
Google Brain
Toronto
{sasabour, frosst, geoffhinton}@google.com

- CapsNets aims to mimic biological neural organizations (hierarchy)
- A hierarchy of capsules, where each capsule represents a different feature or entity in the input data.





Capsule Neural Networks


Dynamic Routing Between Capsules

Sara Sabour Nicholas Frosst

Geoffrey E. Hinton
Google Brain
Toronto
{sasabour, frosst, geoffhinton}@google.com

- Dynamic routing: allows higher-level capsules to select and route information from lower-level capsules
- Capsules allow for multiple instantiations of the same feature —> having its own set of pose, scale, orientation, and deformation parameters.
- CapsNets can address the “Picasso problem” in image recognition

ANNs vs SNNs

- Continuous-valued inputs and outputs.
- Weighted connections between neurons.
- Neurons are typically updated using backpropagation
- ANNs are well-suited for a wide range of tasks, including pattern recognition, classification, and regression.

- Electrical pulses or "spikes" when the neuron reaches a certain threshold of activation.
- Timing and frequency of spikes.
- Backpropagation-like training methods and plasticity-based learning methods.
- SNNs are well-suited for tasks that involve temporal dynamics

ANNs vs SNNs

- Continuous-valued inputs and outputs.
- Weighted connections between

- ANNs are well-suited for a wide range of tasks, including pattern recognition, classification, and regression.

- Electrical pulses or "spikes" when the neuron reaches a certain threshold of activation.

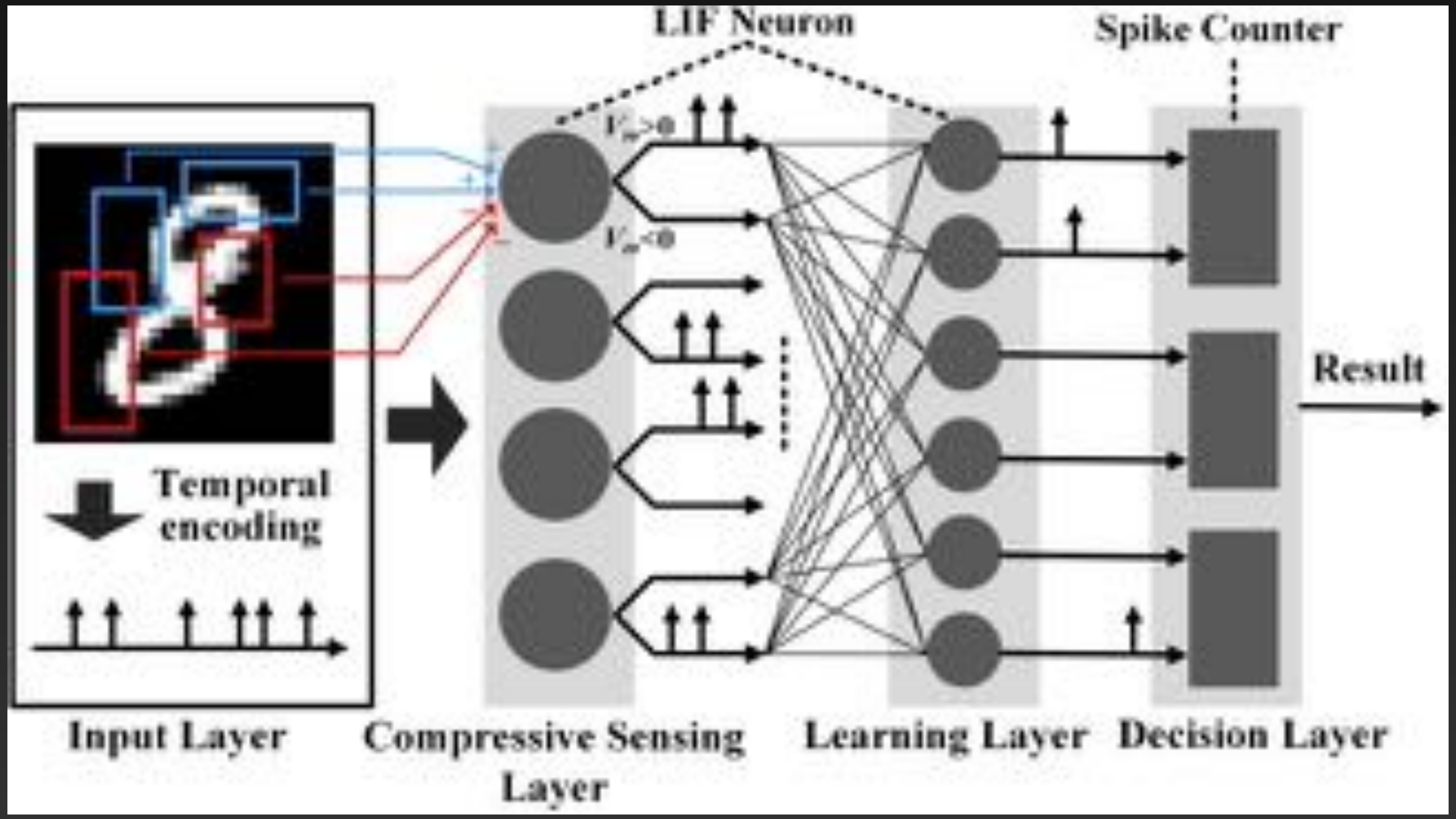
of the spikes.

- SNNs are well-suited for tasks that involve temporal dynamics

Overall, SNNs provide a more biologically plausible model of neural behavior, while ANNs are more flexible and widely used in a range of applications.

Spiking neural networks

- There are different ways to simulate SNNs in computer systems
- Low power consumption, fast inference, event-driven processing, online learning, and massive parallelism
- Spiking neurons’ transfer function is usually non-differentiable, which prevents using backpropagation.



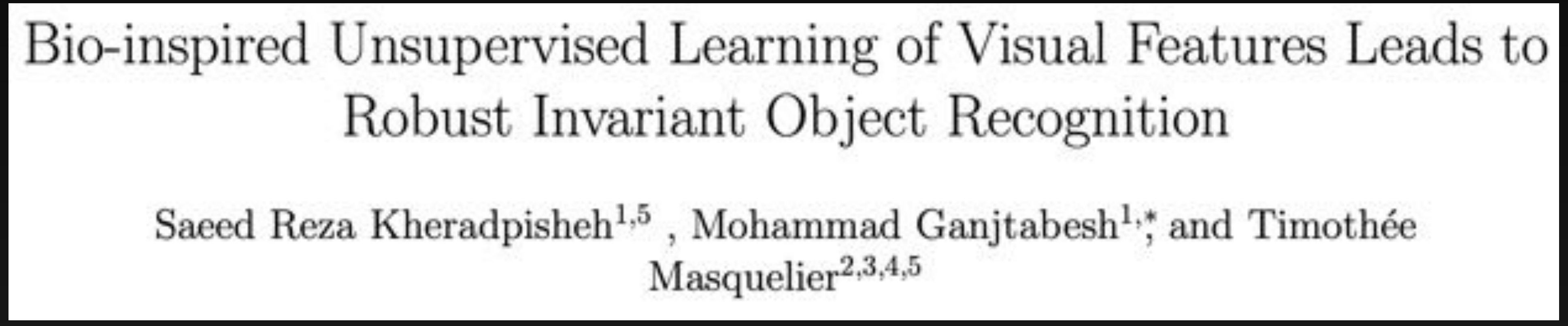
Deep Learning in Spiking Neural Networks

Amirhossein Tavanaei*, Masoud Ghodrati†, Saeed Reza Kheradpisheh‡,
Timothée Masquelier§ and Anthony Maida*

Model	Architecture	Learning method	Dataset	Acc
Feedforward, fully connected, multi-layer SNNs				
O'Connor (2016) [137]	Deep SNN	Stochastic gradient descent	MNIST	96.40
O'Connor (2016) [137]	Deep SNN	Fractional stochastic gradient descent	MNIST	97.93
Lee (2016) [57]	Deep SNN	Backpropagation	MNIST	98.88
Lee (2016) [57]	Deep SNN	Backpropagation	N-MNIST	98.74
Neftci (2017) [138]	Deep SNN	Event-driven random backpropagation	MNIST	97.98
Liu (2017) [108]	SNN	Temporal backpropagation (3-layer)	MNIST	99.10
Eliasmith (2012) [129]	SNN	Spaun brain model	MNIST	94.00
Diehl (2015) [130]	SNN	STDP (2-layer)	MNIST	95.00
Tavanaei (2017) [118]	SNN	STDP-based backpropagation (3-layer)	MNIST	97.20
Mostafa (2017) [109]	SNN	Temporal backpropagation (3-layer)	MNIST	97.14
Querlioz (2013) [139]	SNN	STDP, Hardware implementation	MNIST	93.50
Brader (2007)[128]	SNN	Spike-driven synaptic plasticity	MNIST	96.50
Diehl (2015) [140]	Deep SNN	Offline learning, Conversion	MNIST	98.60
Neil (2016) [144]	Deep SNN	Offline learning, Conversion	MNIST	98.00
Hunsberger (2015) [177], [178]	Deep SNN	Offline learning, Conversion	MNIST	98.37
Esser (2015) [141]	Deep SNN	Offline learning, Conversion	MNIST	99.42
Spiking CNNs				
Lee (2016) [57]	Spiking CNN	Backpropagation	MNIST	99.31
Lee (2016) [57]	Spiking CNN	Backpropagation	N-MNIST	98.30
Panda (2016) [173]	Spiking CNN	Convolutional autoencoder	MNIST	99.05
Panda (2016) [173]	Spiking CNN	Convolutional autoencoder	CIFAR-10	75.42
Tavanaei (2017) [171], [172]	Spiking CNN	Layer wise sparse coding and STDP	MNIST	98.36
Tavanaei (2018) [174]	Spiking CNN	Layer-wise and end-to-end STDP rules	MNIST	98.60
Kheradpisheh (2016) [170]	Spiking CNN	Layer wise STDP	MNIST	98.40
Zhao (2015) [169]	Spiking CNN	Tempotron	MNIST	91.29
Cao (2015) [183]	Spiking CNN	Offline learning, Conversion	CIFAR-10	77.43
Neil (2016) [179]	Spiking CNN	Offline learning, Conversion	N-MNIST	95.72
Diehl (2015) [140]	Spiking CNN	Offline learning, Conversion	MNIST	99.10
Rueckauer (2017) [142]	Spiking CNN	Offline learning, Conversion	MNIST	99.44
Rueckauer (2017) [142]	Spiking CNN	Offline learning, Conversion	CIFAR-10	90.85
Hunsberger (2015) [177]	Spiking CNN	Offline learning, Conversion	CIFAR-10	82.95
Garbin (2014) [181]	Spiking CNN	Offline learning, Hardware	MNIST	94.00
Esser (2016) [182]	Spiking CNN	Offline learning, Hardware	CIFAR-10	87.50
Esser (2016) [182]	Spiking CNN	Offline learning, Hardware	CIFAR-100	63.05
Spiking RBMs and DBNs				
Neftci (2014) [203]	Spiking RBM	Contrastive divergence in LIF neurons	MNIST	91.90
O'Connor (2013) [204]	Spiking DBN	Offline learning, Conversion	MNIST	94.09
Stromatias (2015) [205]	Spiking DBN	Offline learning, Conversion	MNIST	94.94
Stromatias (2015) [206]	Spiking DBN	Offline learning, Hardware	MNIST	95.00
Merolla (2011) [207]	Spiking RBM	Offline learning, Hardware	MNIST	94.00
Neil (2014) [208]	Spiking DBN	Offline learning, Hardware	MNIST	92.00

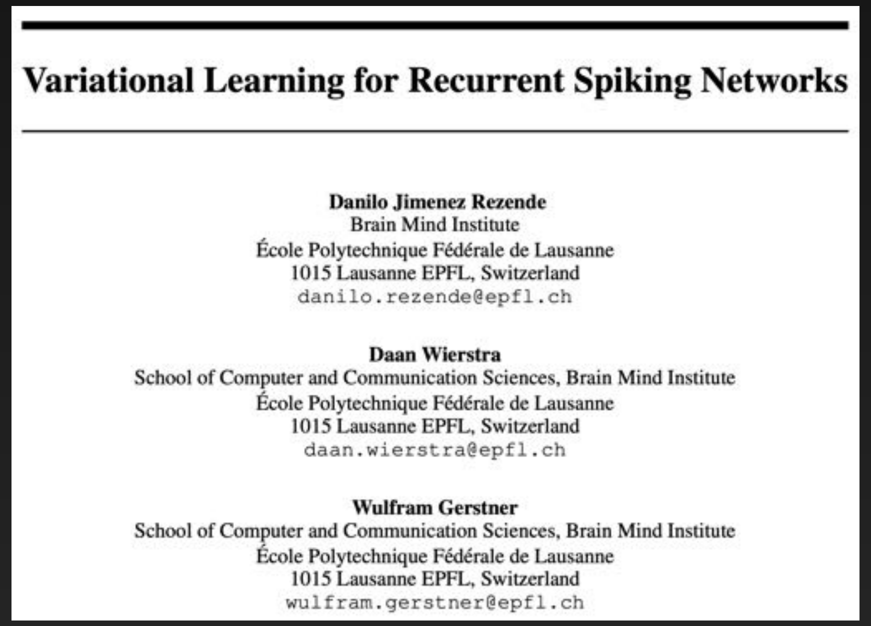
SNNs

- Learning is realized by adjusting scalar-valued synaptic weights
- Three common ways of training SNN:
  - Unsupervised Learning via STDP
  - Probabilistic Characterization of Unsupervised STDP
  - Supervised Learning



Bio-inspired Unsupervised Learning of Visual Features Leads to Robust Invariant Object Recognition

Saeed Reza Kheradpisheh[1,5], Mohammad Ganjtabesh[1,*] and Timothée Masquelier[2,3,4,5]

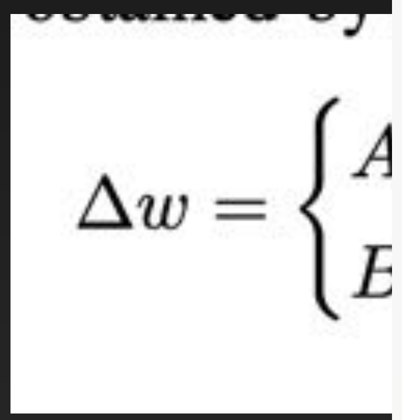


Variational Learning for Recurrent Spiking Networks

Danilo Jimenez Rezende
Brain Mind Institute
École Polytechnique Fédérale de Lausanne
1015 Lausanne EPFL, Switzerland
danilo.rezende@epfl.ch

Daan Wierstra
School of Computer and Communication Sciences, Brain Mind Institute
École Polytechnique Fédérale de Lausanne
1015 Lausanne EPFL, Switzerland
daan.wierstra@epfl.ch

Wulfram Gerstner
School of Computer and Communication Sciences, Brain Mind Institute
École Polytechnique Fédérale de Lausanne
1015 Lausanne EPFL, Switzerland
wulfram.gerstner@epfl.ch



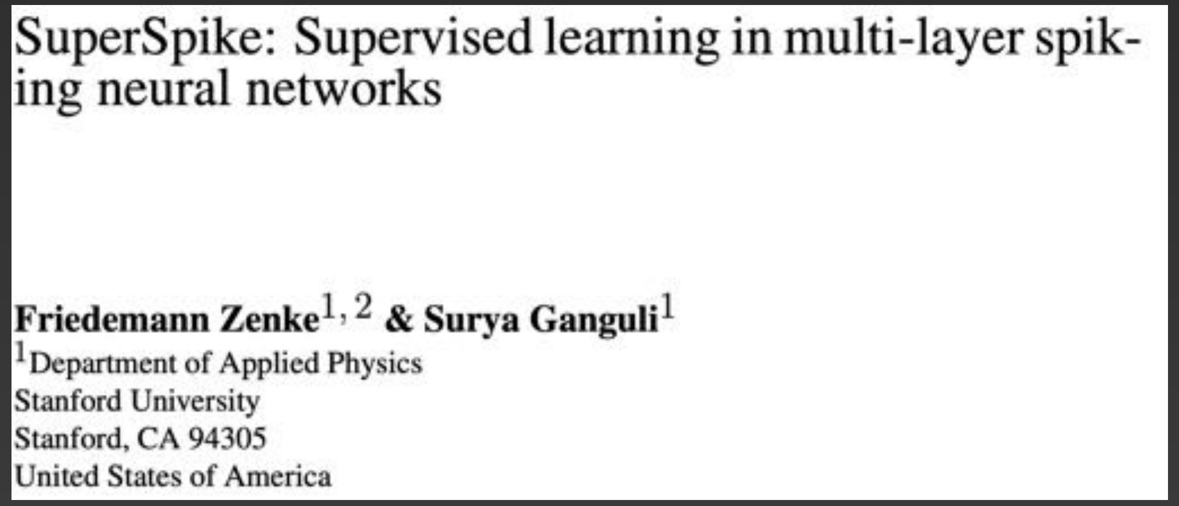
Spatio-Temporal Backpropagation for Training High-Performance Spiking Neural Networks

Yujie Wu[1†], Lei Deng[1,2†], Guoqi Li[1†], Jun Zhu[3*] and Luping Shi[1*]

Towards Biologically Plausible Deep Learning

Yoshua Bengio[1], Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard and Zhouhan Lin

- Supervised learning:
  - Gradients - no bueno: addressed by using substitute or approximate derivatives (bio-plausible?)
  - Weight-transport problem (no bueno)



SuperSpike: Supervised learning in multi-layer spiking neural networks

Friedemann Zenke[1, 2] & Surya Ganguli[1]
[1]Department of Applied Physics
Stanford University
Stanford, CA 94305
United States of America

Spiking neural unit

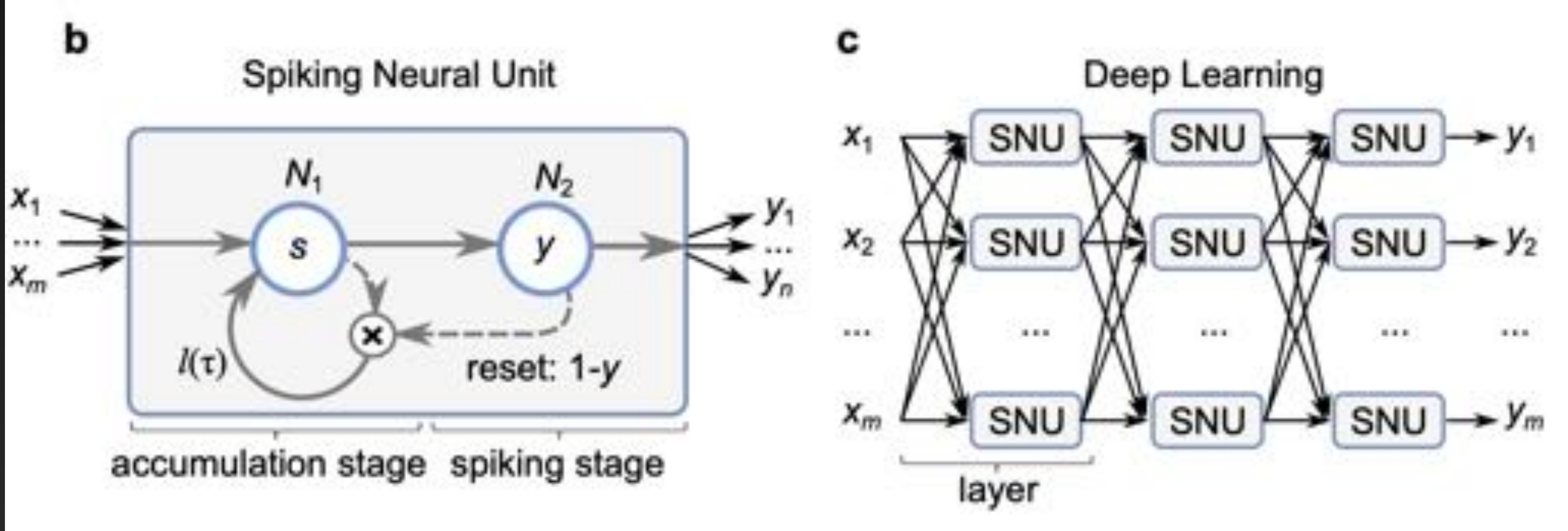
Neurons

- Provided a constructive proof: spiking neuron of LIF type can be transformed into a recurrent ANN unit called Spiking Neural Unit
- Provided a proof mapping ReLU networks to spiking neural networks.
- Benefits: Can train ANN easily and then map to SNN.

Article | Published: 15 June 2020

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

Stanisław Woźniak, Angeliki Pantazi, Thomas Bohnstingl & Evangelos Eleftheriou



AN EXACT MAPPING FROM RELU NETWORKS TO SPIKING NEURAL NETWORKS

Ana Stanojevic[1,2] Stanisław Woźniak[1] Guillaume Bellec[2]
Giovanni Cherubini[1] Angeliki Pantazi[1] Wulfram Gerstner[2]
[1] IBM Research Europe – Zurich, Rüschlikon, Switzerland
[2] École polytechnique fédérale de Lausanne, Lausanne EPFL, Switzerland

Meta RL

- Dopamine implementing model-free RL, PFC performing model-based RL

Prefrontal cortex as a meta-reinforcement learning system

Jane X. Wang, Zeb Kurth-Nelson, Dharshan Kumaran, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Demis Hassabis & Matthew Botvinick

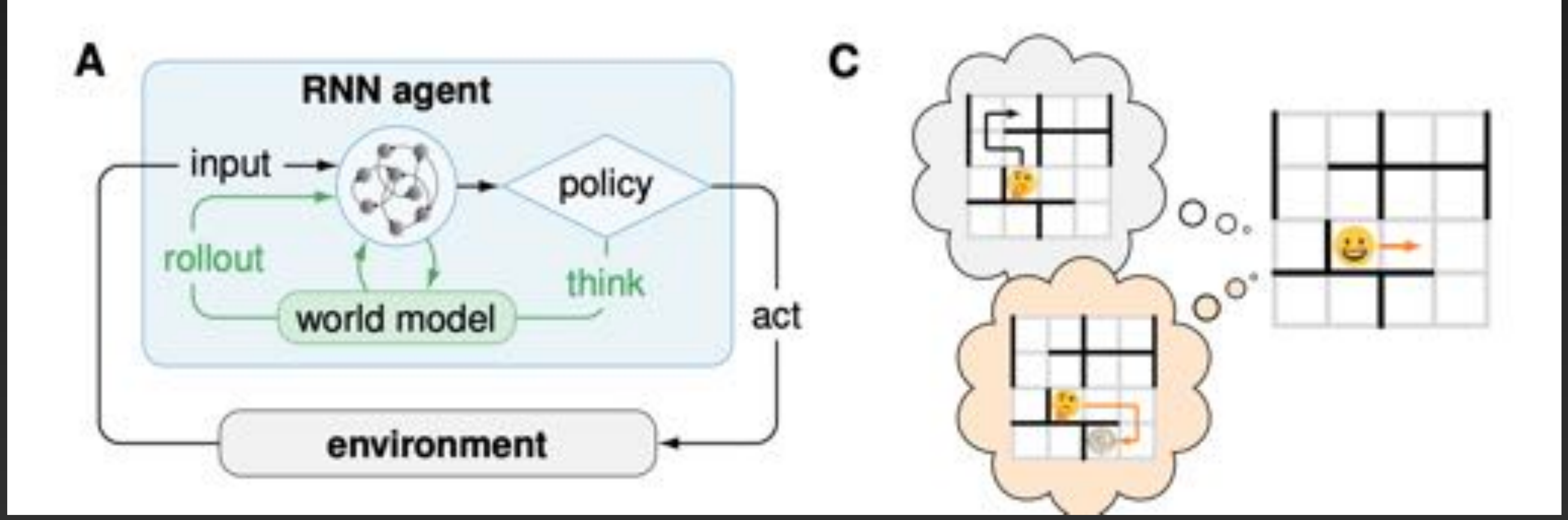
- DA-based RL created a second RL algorithm

Meta RL

- Combines slow synaptic learning with fast adaptation through recurrent dynamics in the prefrontal net work.
- Given policy, the agent can also choose to “think”.
- Rollouts improve the policy of the RL agent

A recurrent network model of planning explains hippocampal replay and human behavior

Kristopher T. Jensen[@1], Guillaume Hennequin[*1], and Marcelo G. Mattar[*2,3]



There is so so much more

A deep learning framework for neuroscience

...hua Bengio, Rafal Bogacz, Amelia ... Surya Ganguli, Colleen J. Gillon, ...am, Grace W. Lindsay, Kenneth D. ... Miller, ...ter Roelfsema, João Sacramento, ... Andrew ... Greg Wayne, Daniel Yamins, ... Friedem... ...Kording — Show fewer authors

A Review of Bio-Inspired Computational Intelligence Algorithms in Electricity Load Forecasting

By Siva Sankari Subbiah, Jayakumar Chinnappan

Studies in Computational Intelligence 903

Akash Kumar Bhoi
Pradeep Kumar Mallick
Chuan-Ming Liu
Valentina E. Balas *Editors*

Bio-inspired Neurocomputing

Bio inspired computing – A review of algorithms and scope of applications

Arpan Kumar Kar

Memristors—From In-Memory Computing, Deep Learning Acceleration, and Spiking Neural Networks to the Future of Neuromorphic and Bio-Inspired Computing

Adnan Mehonic, Abu Sebastian, Bipin Rajendran, Osvaldo Simeone, Eleni Vasilaki, Anthony J. Kenyon

A comparative study on bio-inspired algorithms for sentiment analysis

Ashima Yadav & Dinesh Kumar Vishwakarma

Nature and Biologically Inspired Image Segmentation Techniques

Simrandeep Singh, Nitin Mittal, Diksha Thakur, Harbinder Singh, Diego Oliva & Anton Demin