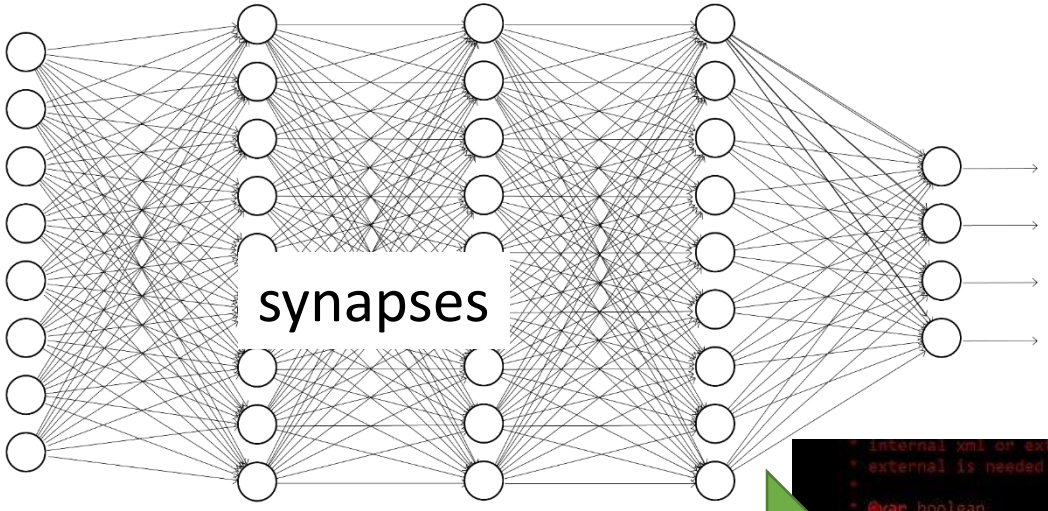
Neuromorphic Computing: AI needs new hardware

Julie Grollier

Unité Mixte de Physique CNRS/Thales, Palaiseau

bio*SPIN*spired

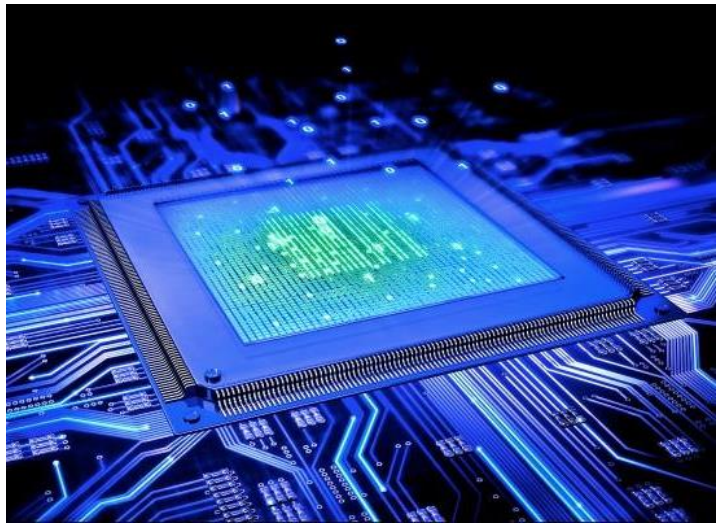
Deep Neural networks run on unoptimized hardware




```
 * internal xml or external
 * external is needed when running in static mode
 *
 * @var boolean
 */
define('PSI_INTERNAL_XML', false);

if (version_compare("5.2", PHP_VERSION, ">")) {
    die("PHP 5.2 or greater is required!!!");
}
if (!extension_loaded("pcre")) {
    die("phpSysInfo requires the pcre extension to php in order to work
        properly.");
}

require_once APP_ROOT.'/includes/autoloader.inc.php';

// Load configuration
require_once APP_ROOT.'/config.php';

if (!defined('PSI_CONFIG_FILE') || !defined('PSI_DEBUG')) {
    $tpl = new Template("/templates/html/error_config.html");
    echo $tpl->fetch();
    die();
```

0011011

GPUs, TPUs, FPGAs



Energy consumption of AI

Consumption	**CO_2e (lbs)**
Air travel, 1 passenger, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experimentation	78,468
Transformer (big)	192
w/ neural architecture search	626,155

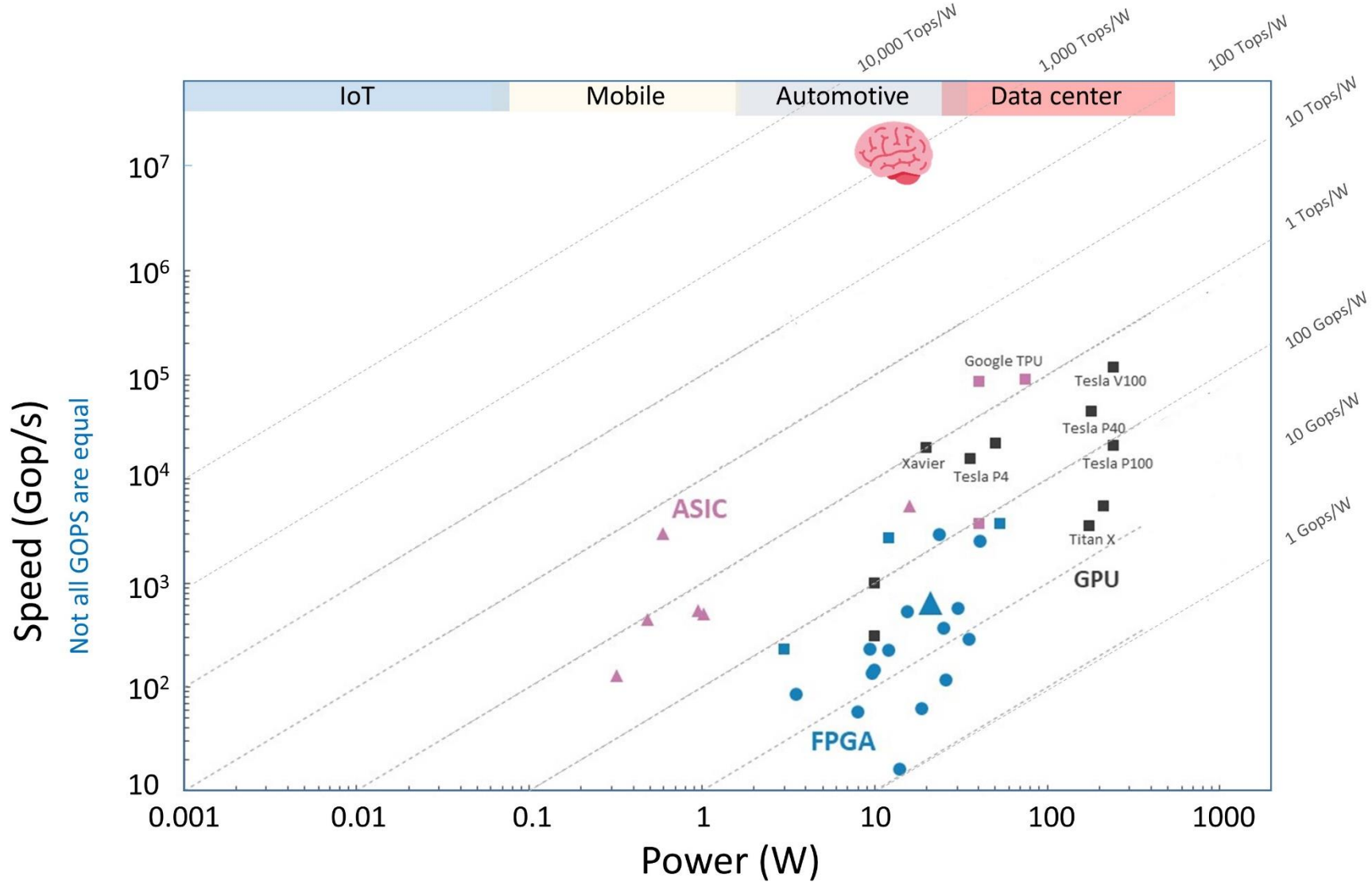
Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

Model	Hardware	Power (W)	Hours	kWh·PUE	CO_2e	Cloud compute cost
Transformer$_{base}$	P100x8	1415.78	12	27	26	$41–$140
Transformer$_{big}$	P100x8	1515.43	84	201	192	$289–$981
ELMo	P100x3	517.66	336	275	262	$433–$1472
BERT$_{base}$	V100x64	12,041.51	79	1507	1438	$3751–$12,571
BERT$_{base}$	TPUv2x16	—	96	—	—	$2074–$6912
NAS	P100x8	1515.43	274,120	656,347	626,155	$942,973–$3,201,722
NAS	TPUv2x1	—	32,623	—	—	$44,055–$146,848
GPT-2	TPUv3x32	—	168	—	—	$12,902–$43,008

Table 3: Estimated cost of training a model in terms of CO_2 emissions (lbs) and cloud compute cost (USD).[7] Power and carbon footprint are omitted for TPUs due to lack of public information on power draw for this hardware.

E. Strubell et al, https://arxiv.org/abs/1906.02243v1 (2019)

Current CMOS processors cannot run future AI

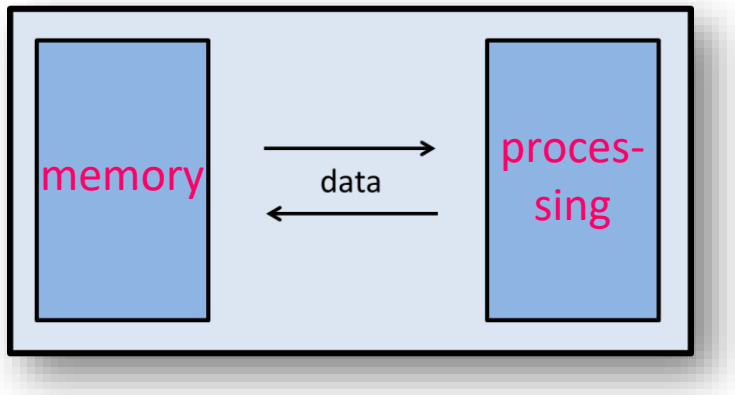


[Based on https://nicsefc.ee.tsinghua.edu.cn/projects/neural-network-accelerator/]

Training neural networks on todays computers is extremely power inefficient

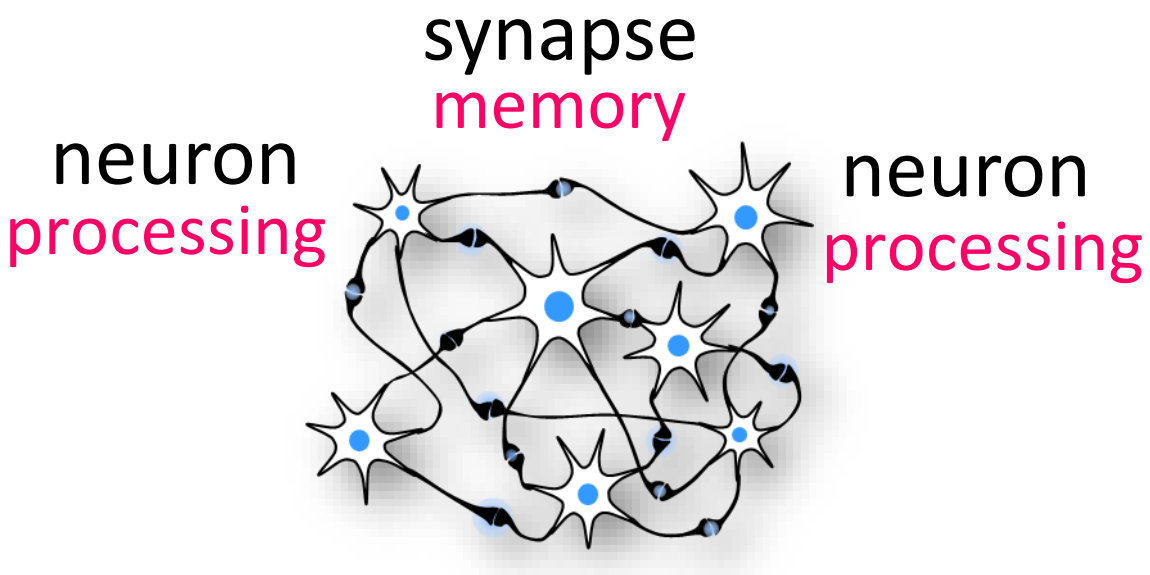
Digital computer:

CPUs, GPUs, TPUs, FPGAs



1000 kW.h to train a
Natural Language Processor

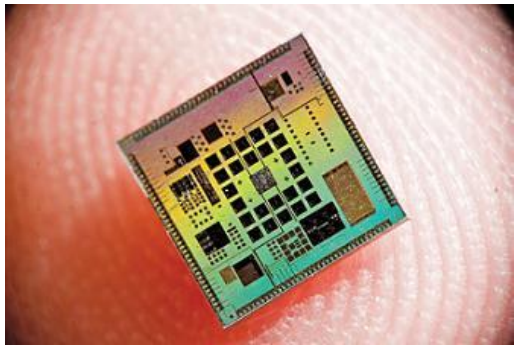
Brain : 20 W

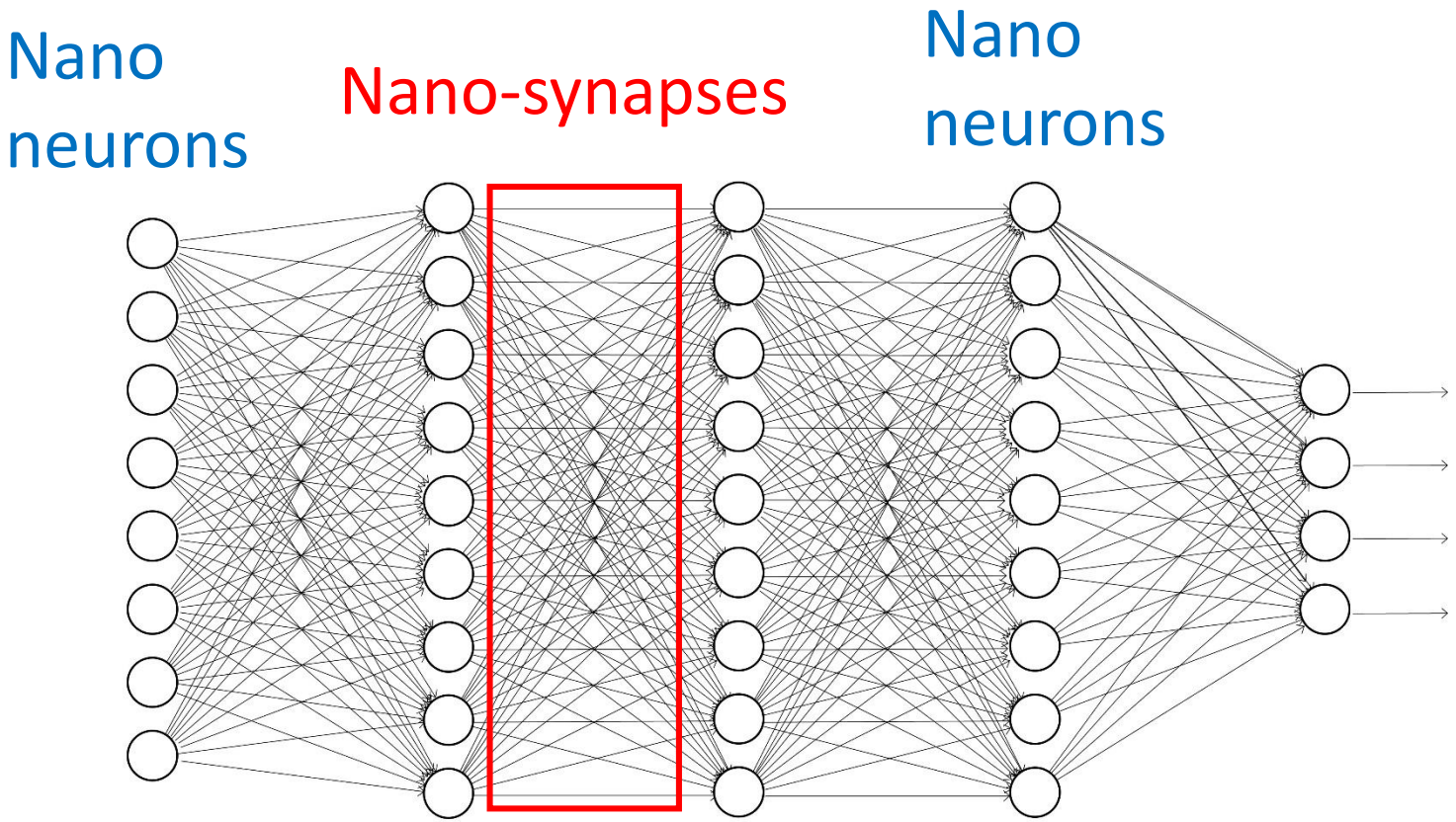


6 years of brain operation

D. Marković et al, "Physics for neuromorphic computing", Nature Review Physics 2020

Orders of magnitude in energy can be saved by assembling physical synapses and neurons in neuromorphic chips





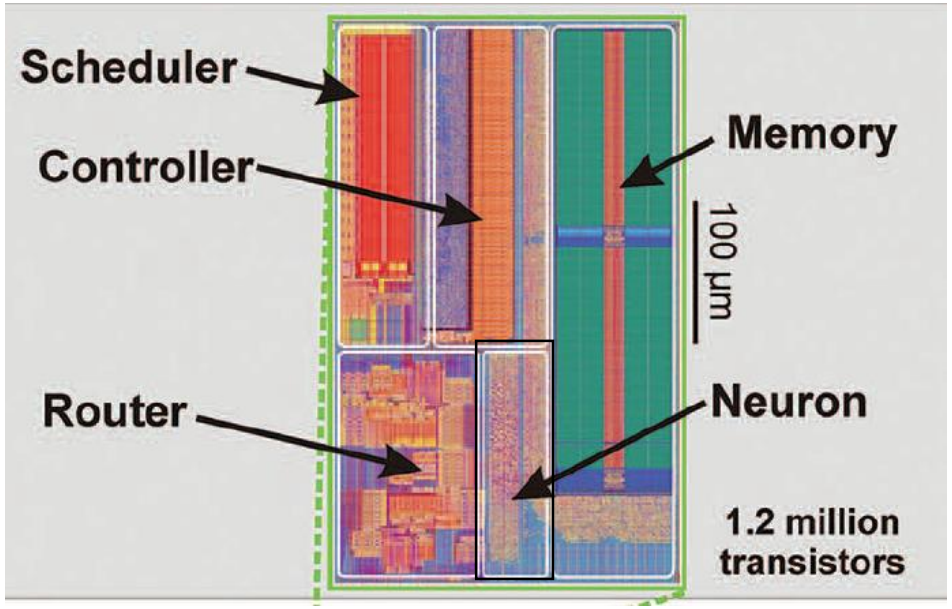
Hundred millions of neurons and synapses in a 1 cm^2 chip

→ Each device smaller than 1 μm^2

CMOS neurons and synapses are complex circuits

- A transistor is nanoscale but it is just a switch
- CMOS does not provide memory (volatile)

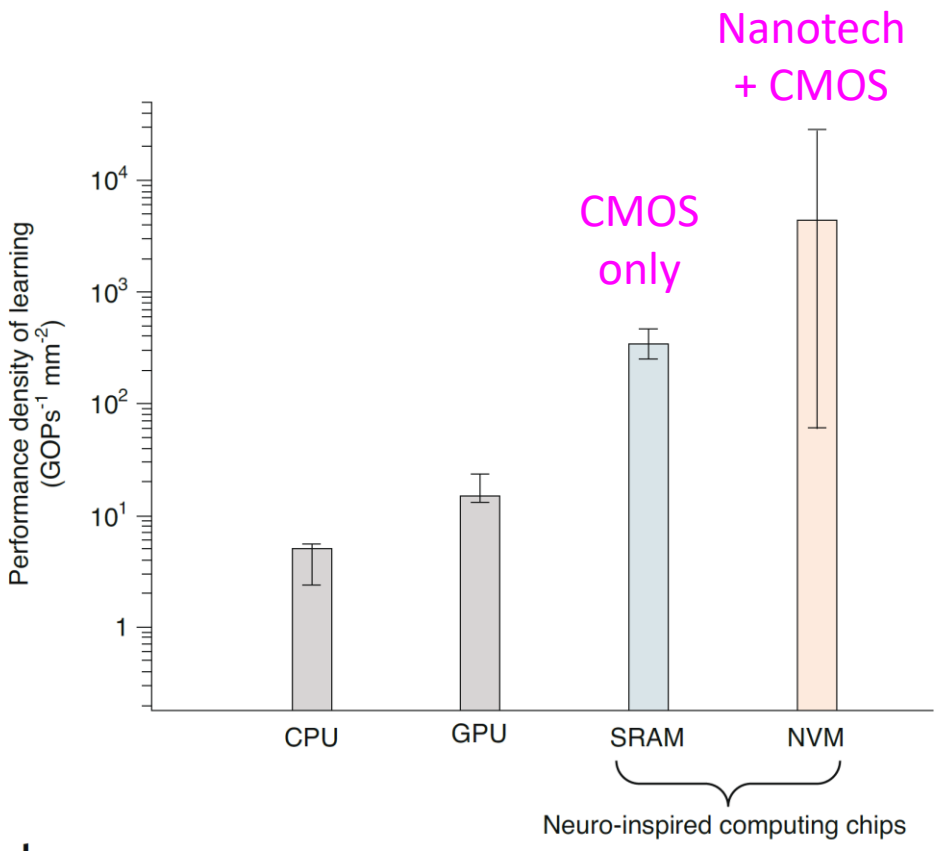
CMOS neuron **10-100 µm**
CMOS synapse **10 µm**

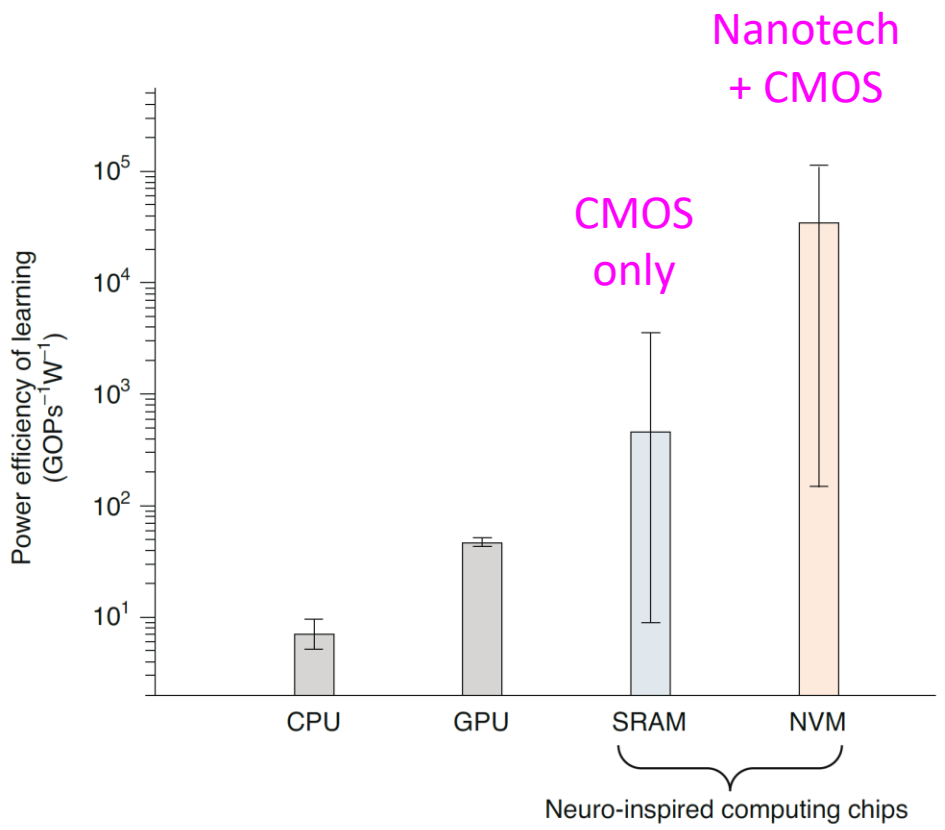


Merolla et al, *Science* **345**, 668 (2014)

Brainscales 20 wafer machine. 4M neurons, 1B synapses

Transistors alone won't do the job: they should be complemented by emerging nanotechnologies





Zhang et al, Nature Electronics 3, 371 (2020)

The power of novel nanotechnologies for AI

They are multifunctional: they can emulate many features of neurons and synapses

Filamentary switching

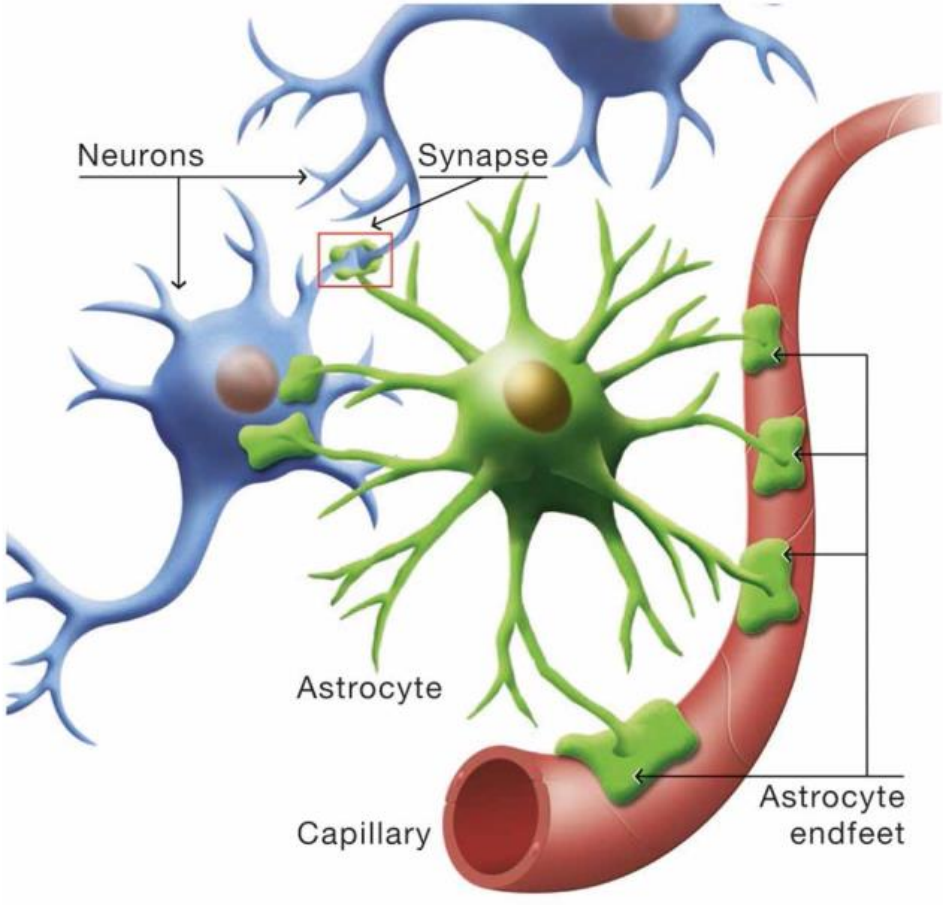
Phase-change

Optics

Ferroelectrics

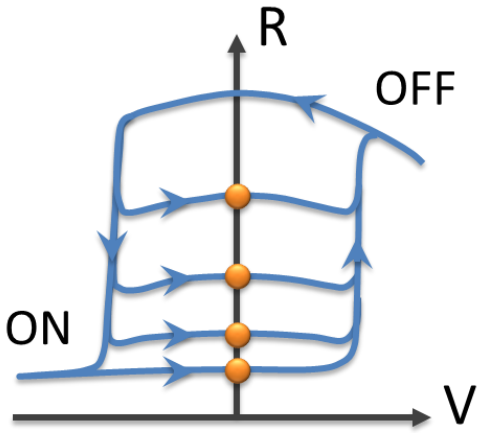
Organics

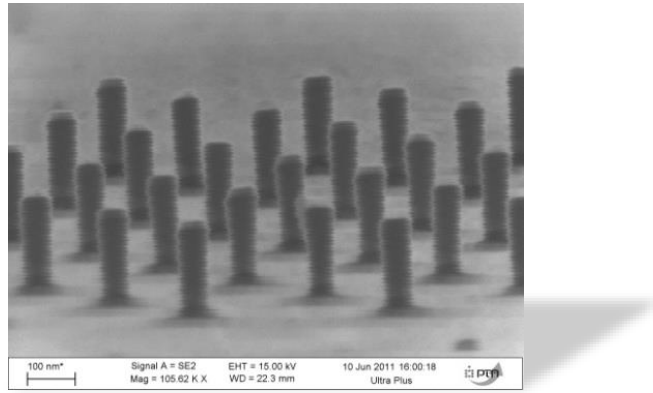
Spintronics



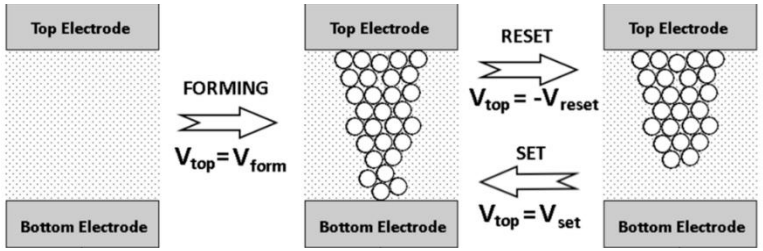
Non-volatile memristors emulate synapses

Chua, IEEE Trans.
Circuit Theory (1971)



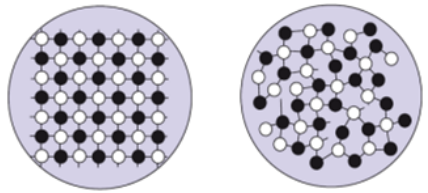


Filamentary switching



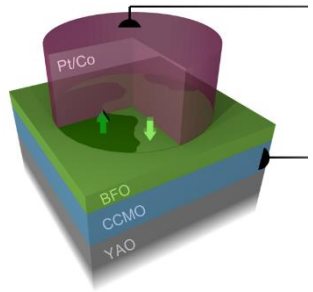
Yang et al.,
Nature Nano. (2013)

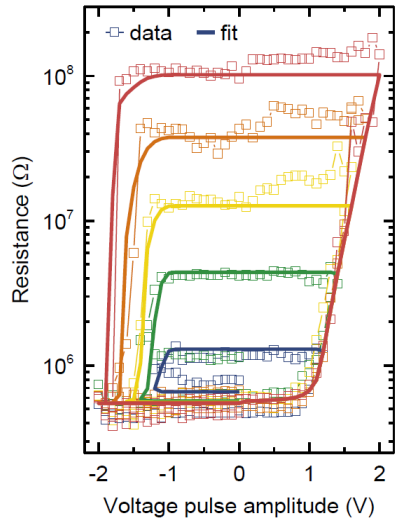
Phase change



Kuzum et al,
Nanotechnology (2013)

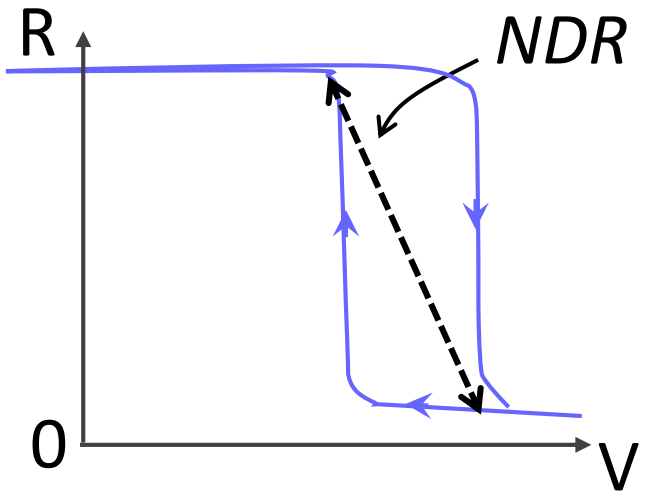
Ferroelectric

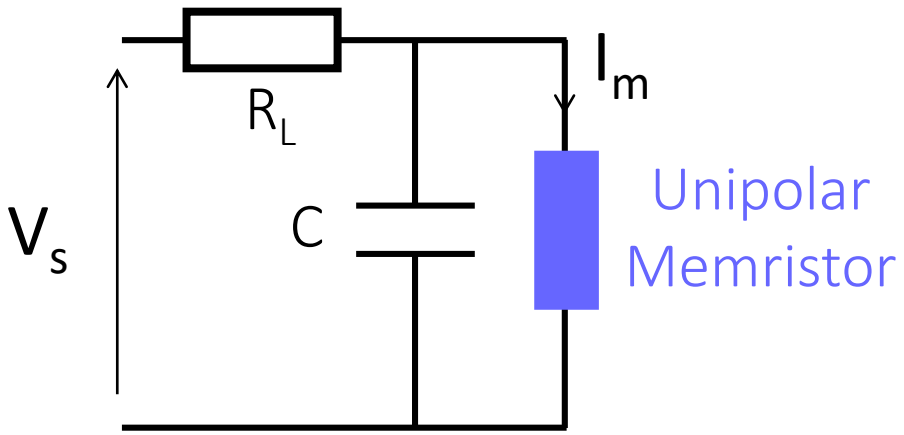


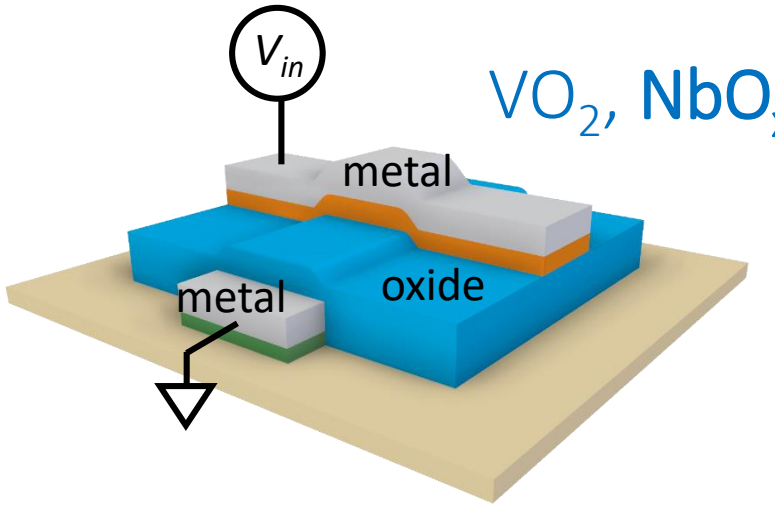


Chanthbouala et al,
Nature Mat. (2012)

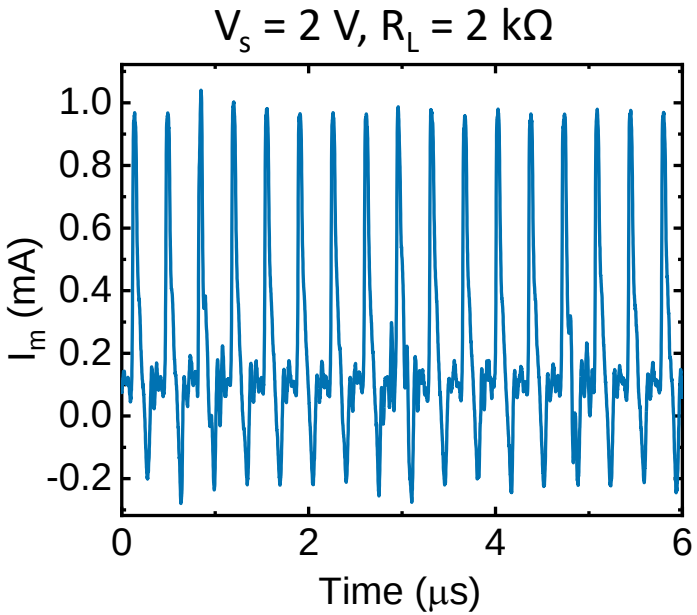
Volatile memristors can be used as neurons







VO_2, NbO_x …

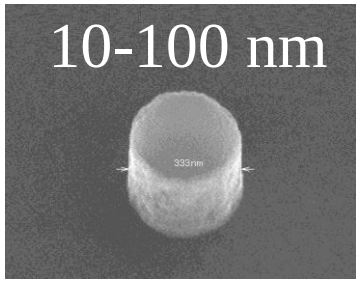


Pickett et al, *Nature Mat.* 12, 114 (2013)

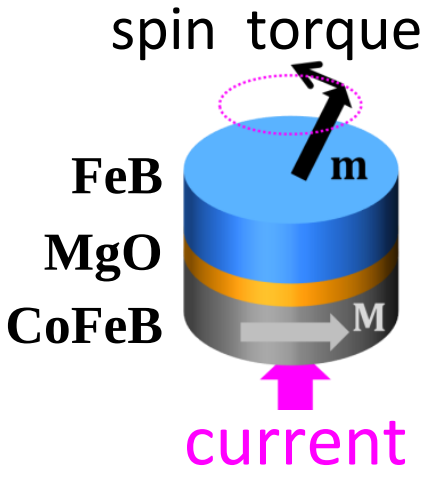
S. Li et al, APL 106, 212902 (2015)

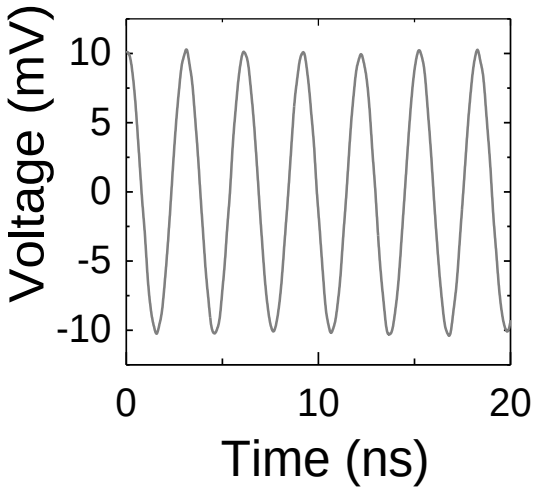
Spintronic oscillators can be used as radio-frequency neurons

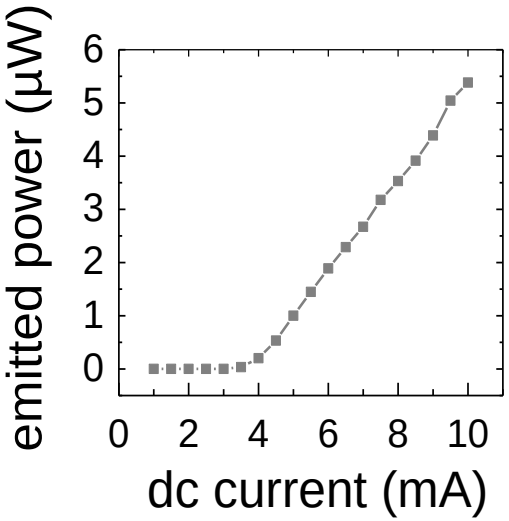
magnetic tunnel junction

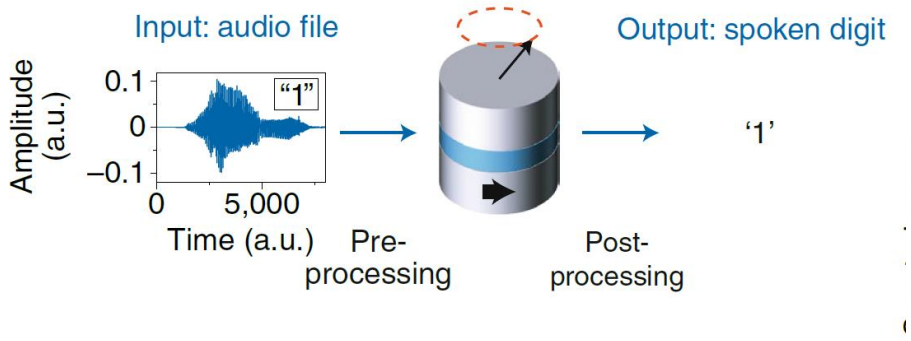


compatible with CMOS

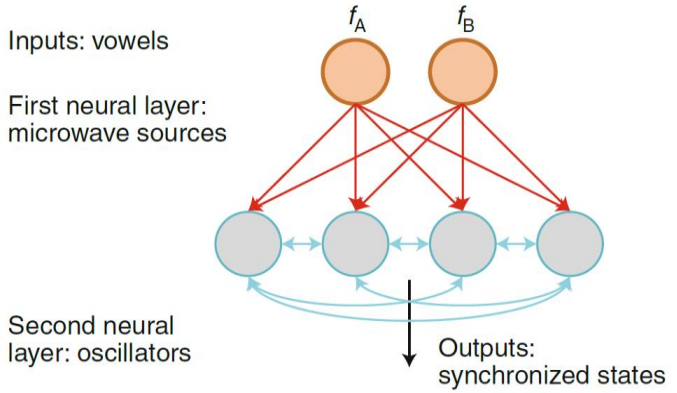






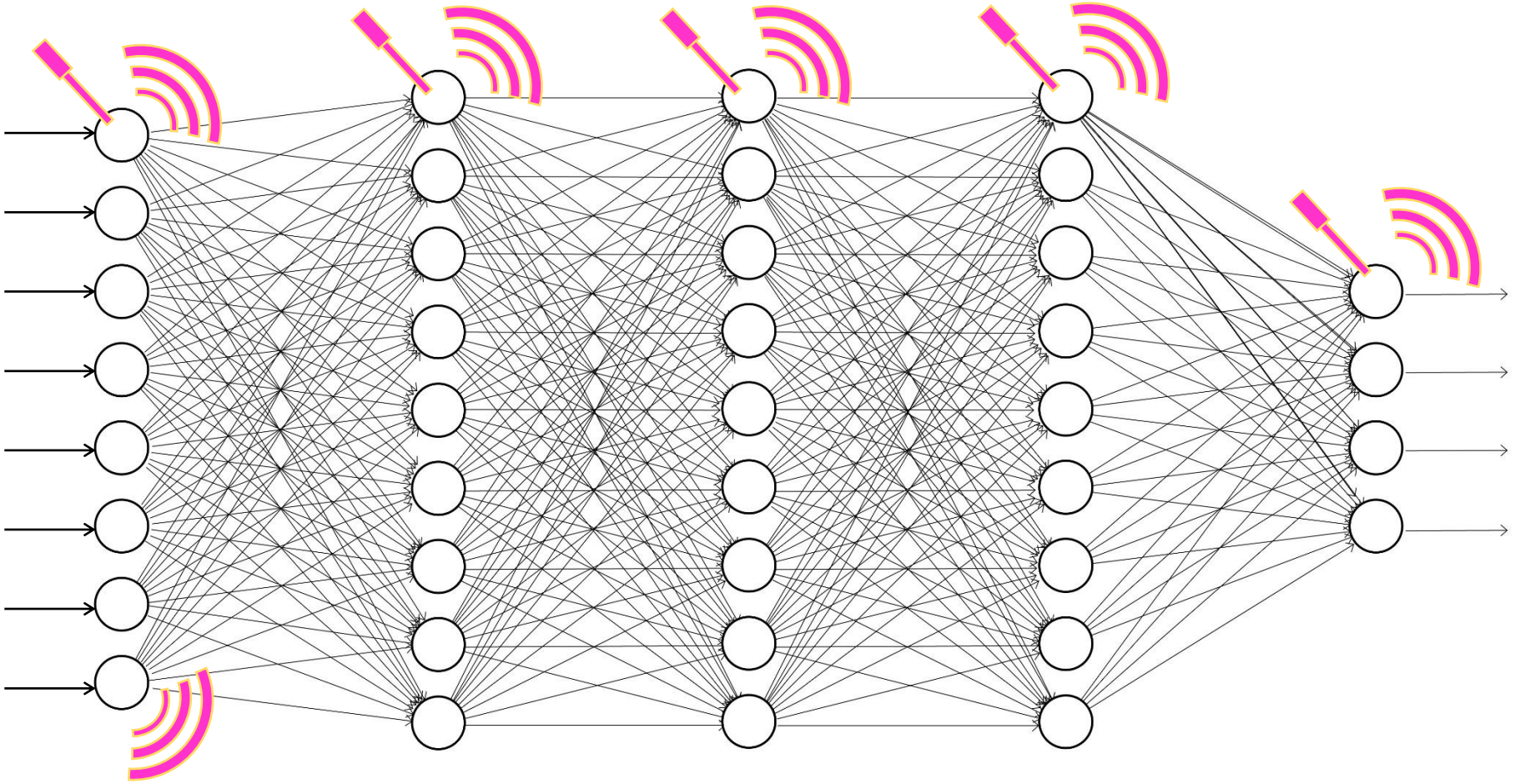


J. Torrejon et al, Nature 547, 428 (2017)



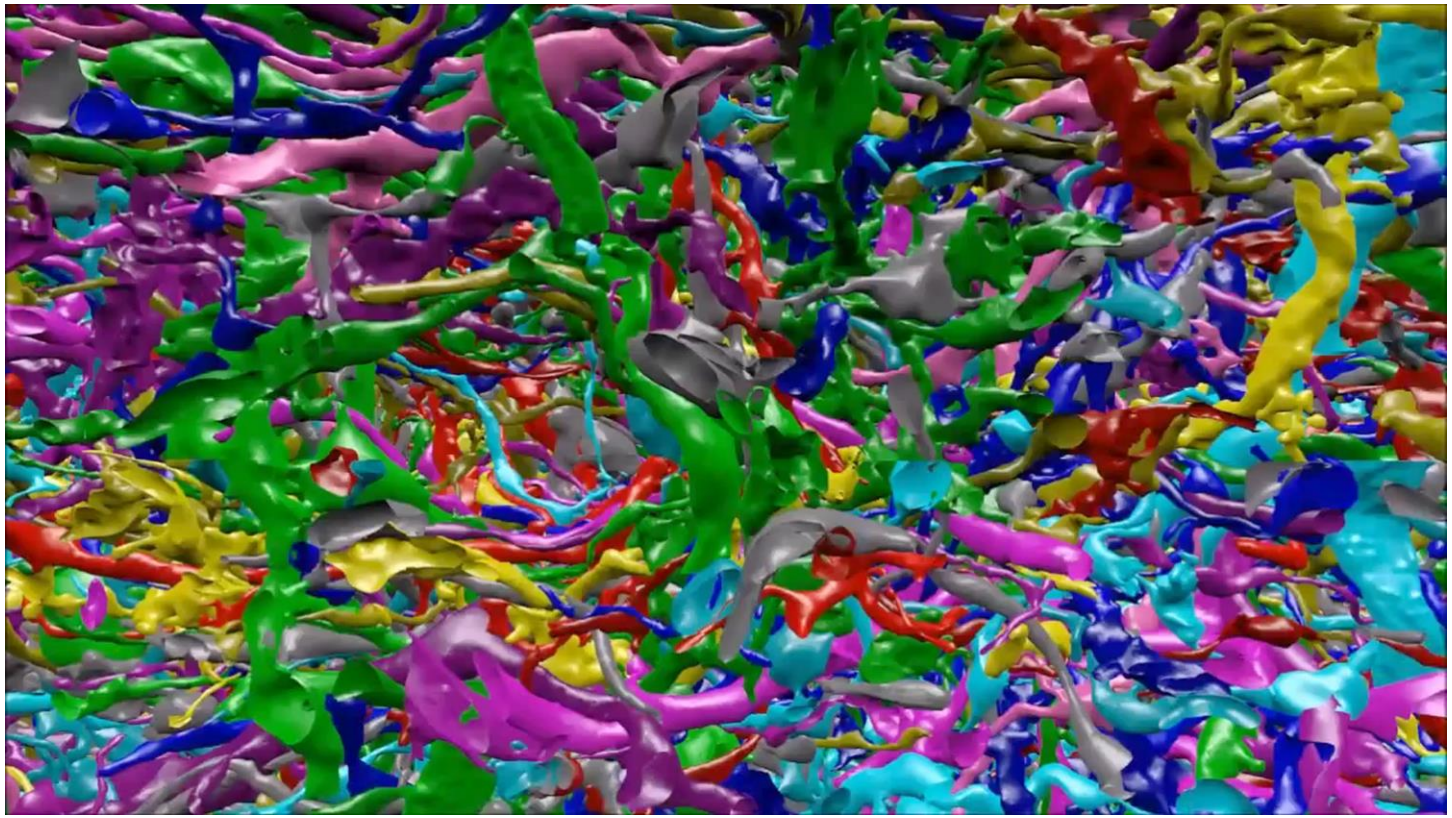
M. Romera et al, Nature 563, 230 (2018)

Deep learning through RF communications?



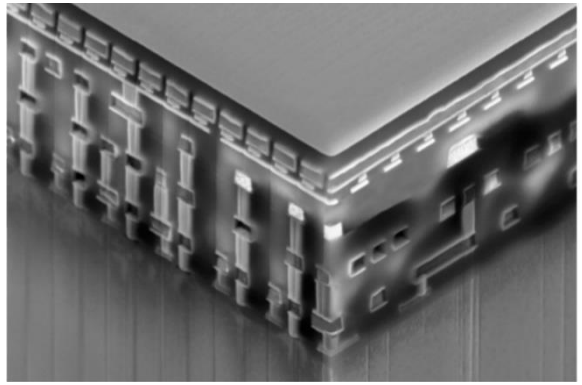
Novel nanotechnologies open new ways to interconnect synapses and neurons

Cortex: 10^4 synapses / neurones = 10^4 wires/neurons

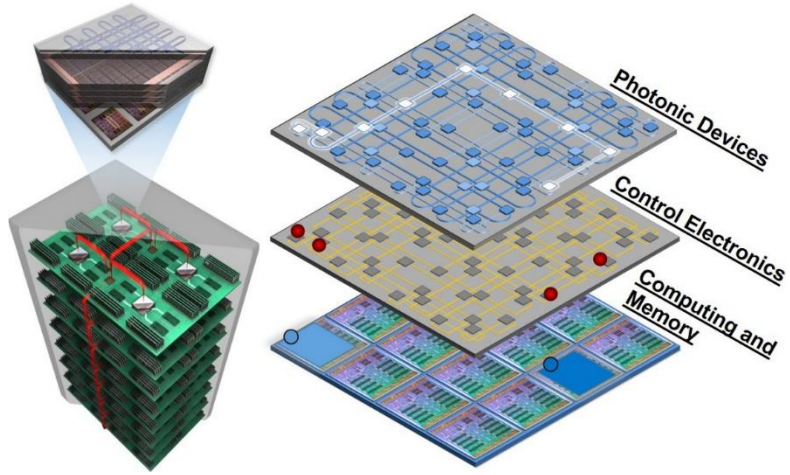


3D interconnections

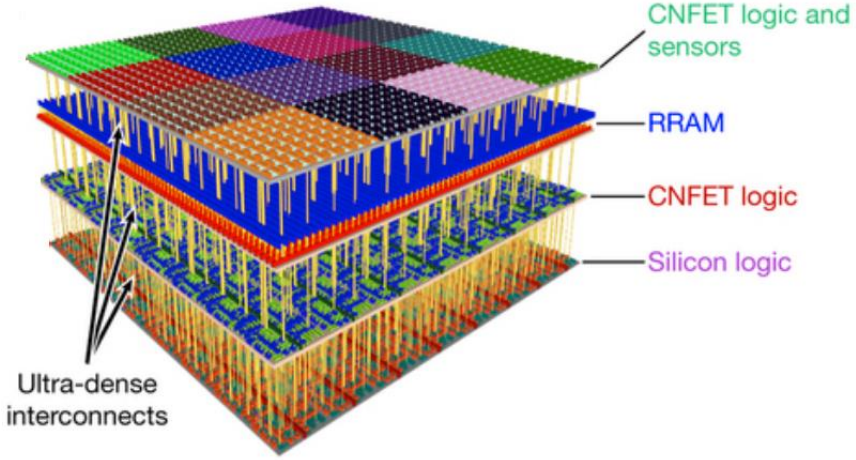
CMOS



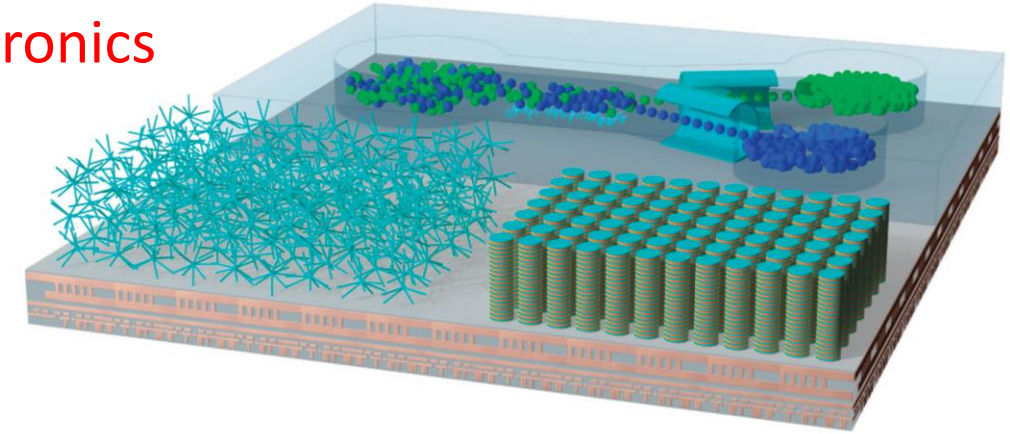
Photonics



Heterogeneous integration



Spintronics



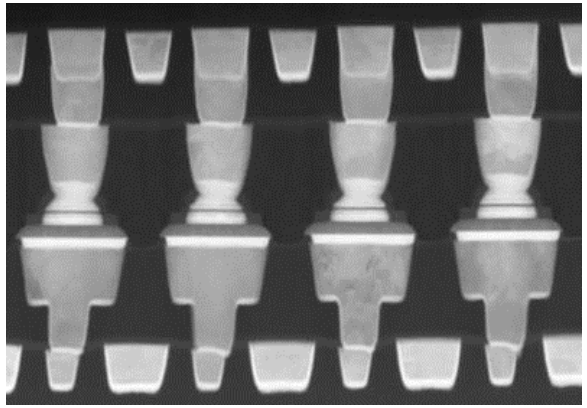
M.M. Shulaker et al , Nature 547, 74 (2017)

A. Fernandez- Pacheco et al, Nature Com. 8:15756 (2017)

Novel nanotechnologies are integrated in major foundry process : they will become commercially available soon

Spintronics

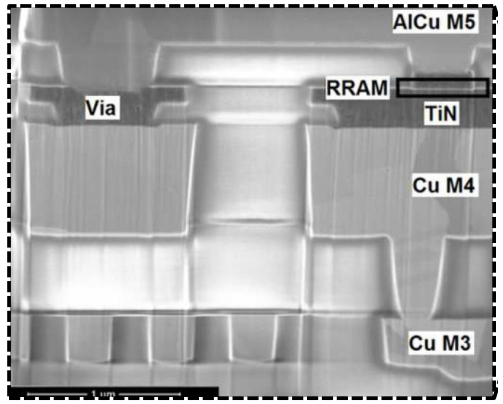
magnetic tunnel junctions



Intel: MRAM integrated into 22nm FinFET CMOS

Memristors

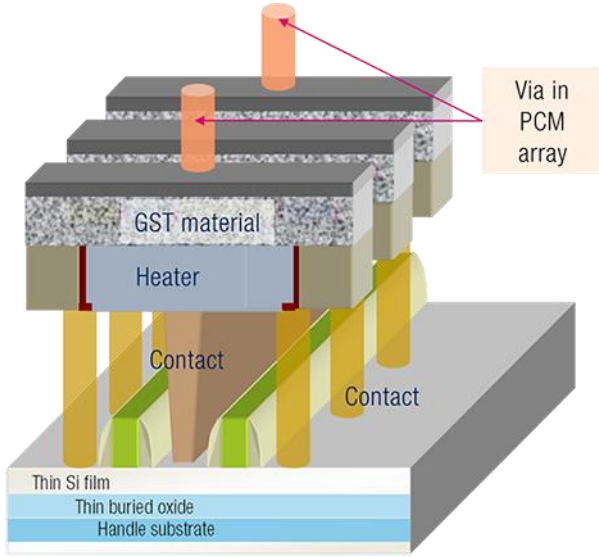
ReRAMs



CEA LETI: 130nm CMOS + HfO_2 RRAM

Bocquet, …, Vianello, Portal, Querlioz,
IEEE IEDM, 2018

Phase Change

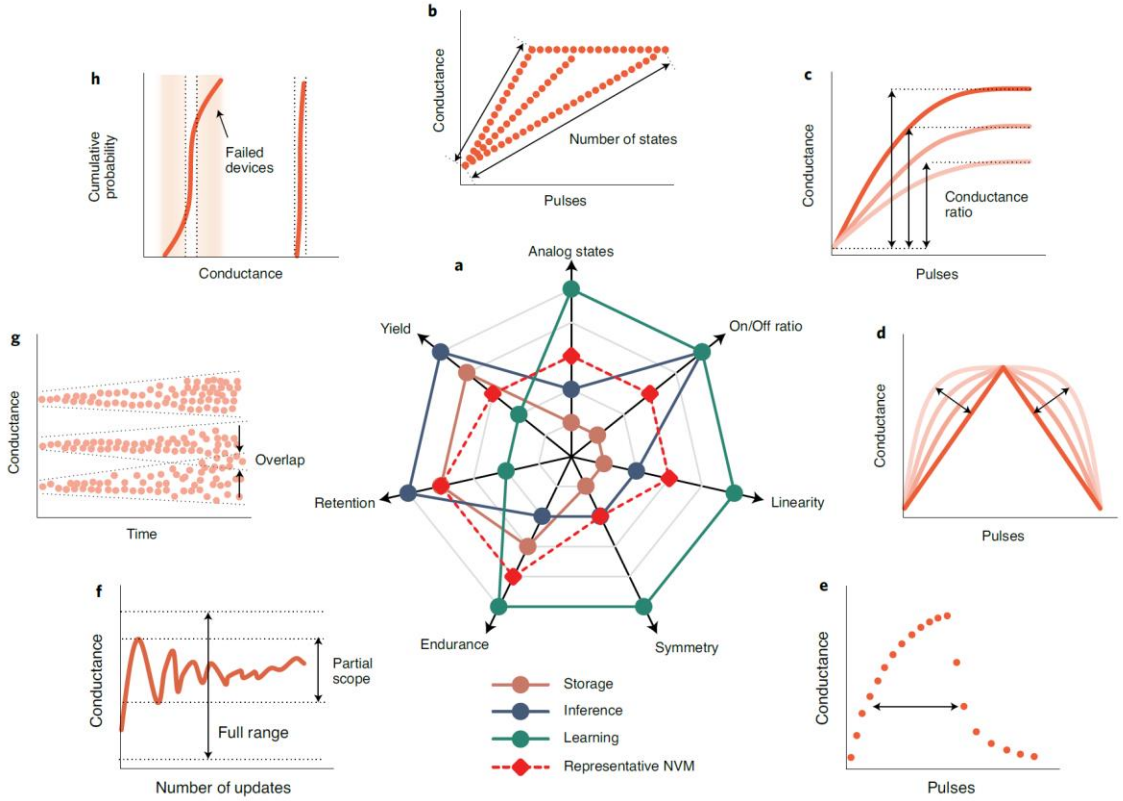


ST microelectronics

The downside of novel nanotechnologies for AI

Nanodevices are by essence noisy, imperfect and highly variable from device to device

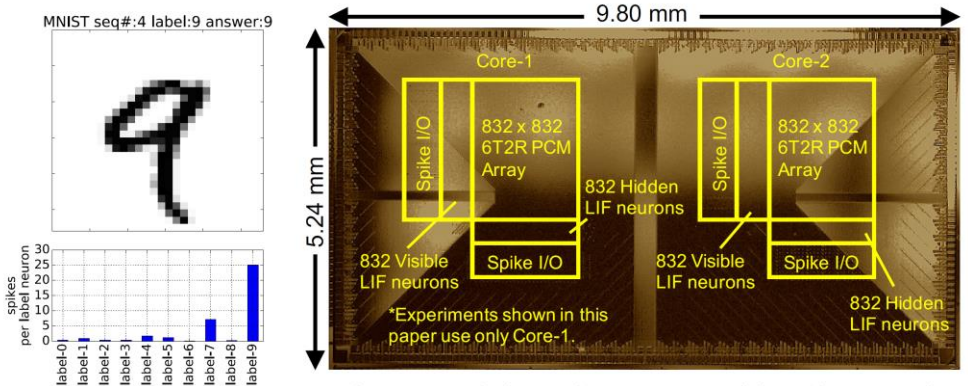
Panorama of memristor synapse faults



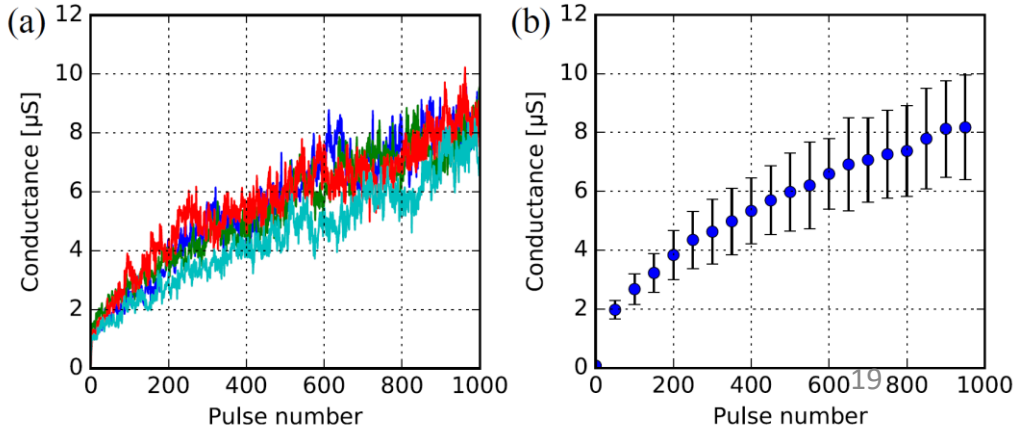
Zhang et al, Nature Electronics 3, 371 (2020)

On-Chip Trainable 1.4M 6T2R PCM Synaptic Array with 1.6K Stochastic LIF Neurons for Spiking RBM

M. Ishii[1*], S. Kim[2*], S. Lewis[3], A. Okazaki[1], J. Okazawa[1], M. Ito[1], M. Rasch[3], W. Kim[3], A. Nomura[1], U. Shin[2], K. Hosokawa[1], M. BrightSky[3], and W. Haensch[3]
[1]IBM Research – Tokyo, Japan, [2]Seoul National University, South Korea, [3]IBM Research, T.J. Watson Research Center, USA
*These authors contributed equally to this work, email: ishiim@jp.ibm.com, sangbum.kim@snu.ac.kr

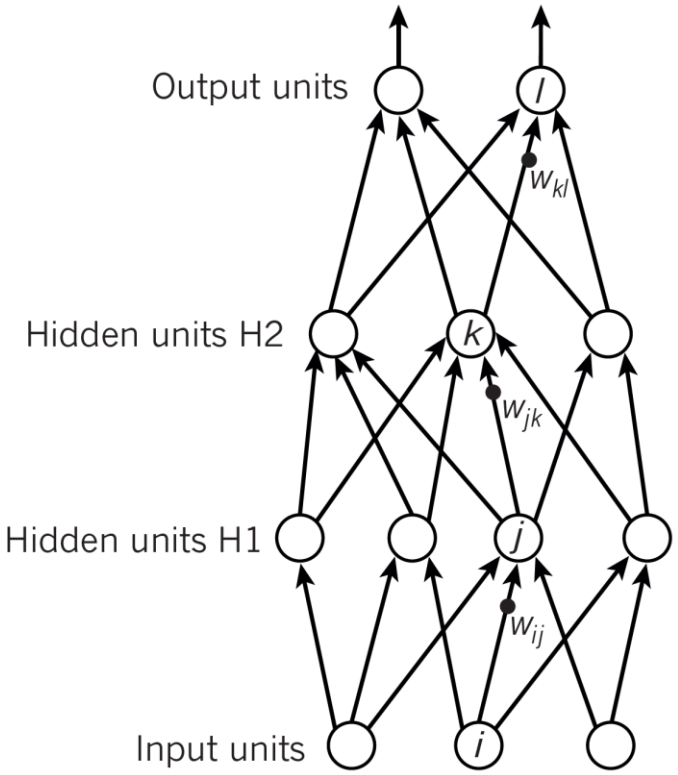


First fully integrated memristor/CMOS chip: only 92% on MNIST due to device variability



They are hardly compatible with the flagship training algorithm of deep neural networks: backpropagation of errors

Forward pass: inference



$$y_l = f(z_l)$$
$$z_l = \sum_{k\ \varepsilon\ \text{H2}} w_{kl}\, y_k$$

$$y_k = f(z_k)$$
$$z_k = \sum_{j\ \varepsilon\ \text{H1}} w_{jk}\, y_j$$

$$y_j = f(z_j)$$
$$z_j = \sum_{i\ \varepsilon\ \text{Input}} w_{ij}\, x_i$$

$$\Delta w = -\alpha \frac{\partial E}{\partial w}$$

$$\frac{\Delta w}{w} < 10^{-5}$$

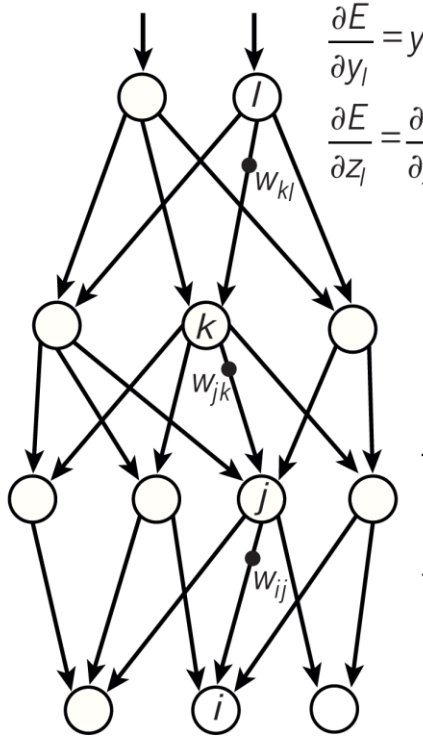
Backward pass

Compare outputs with correct answer to get error derivatives

$$\frac{\partial E}{\partial y_l} = y_l - t_l$$
$$\frac{\partial E}{\partial z_l} = \frac{\partial E}{\partial y_l}\frac{\partial y_l}{\partial z_l}$$

$$\frac{\partial E}{\partial y_k} = \sum_{l\ \varepsilon\ \text{out}} w_{kl}\frac{\partial E}{\partial z_l}$$

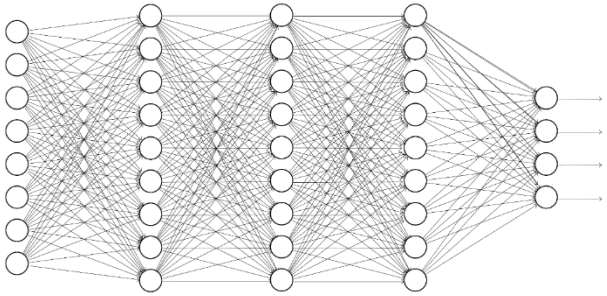
$$\frac{\partial E}{\partial z_k} = \frac{\partial E}{\partial y_k}\frac{\partial y_k}{\partial z_k}$$

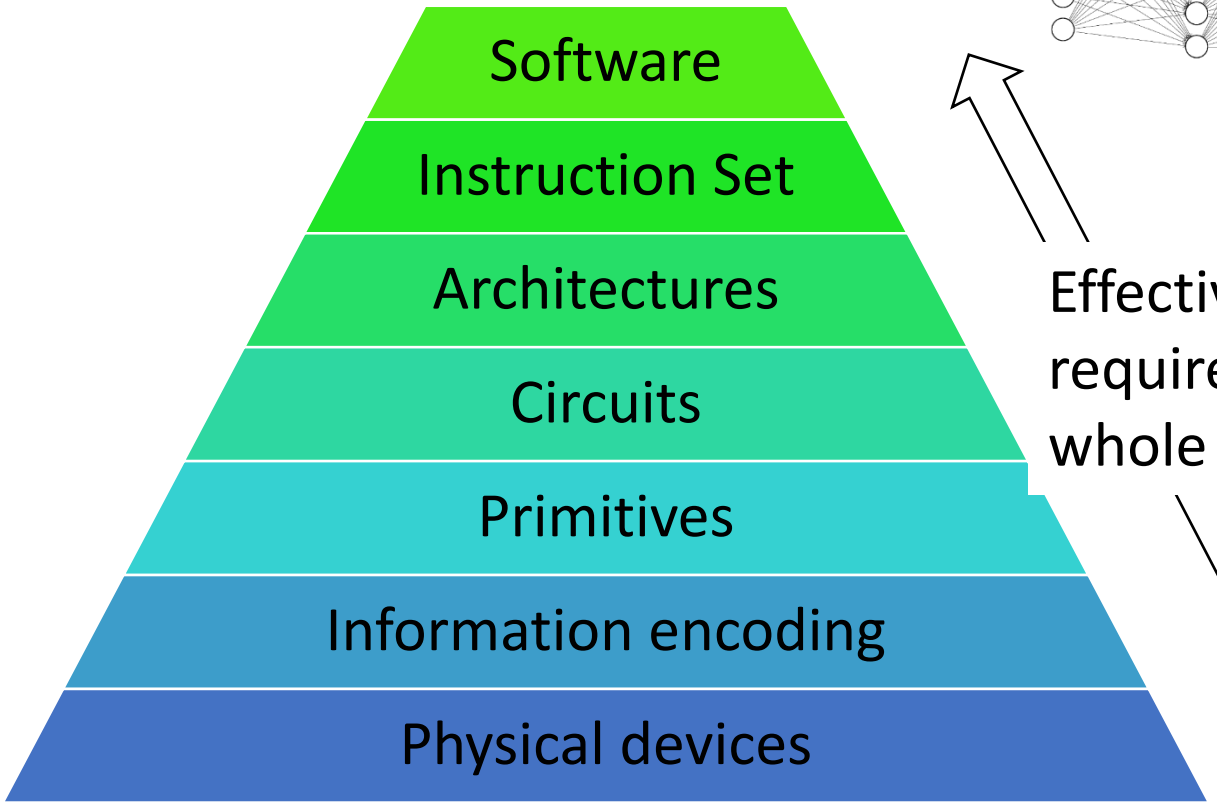


$$\frac{\partial E}{\partial y_j} = \sum_{k\ \varepsilon\ \text{H2}} w_{jk}\frac{\partial E}{\partial z_k}$$

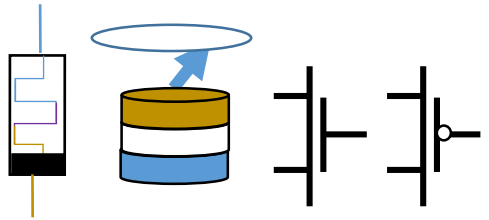
$$\frac{\partial E}{\partial z_j} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial z_j}$$

Yann Lecun, Yoshua Bengio and Geoffrey Hinton, *Nature* 521, 436 (2015)





Effective use of new devices requires working across the whole computational stack



Three main approaches

1- implement backpropagation

AI inspired

2- make backpropagation more hardware-compatible (top-down)

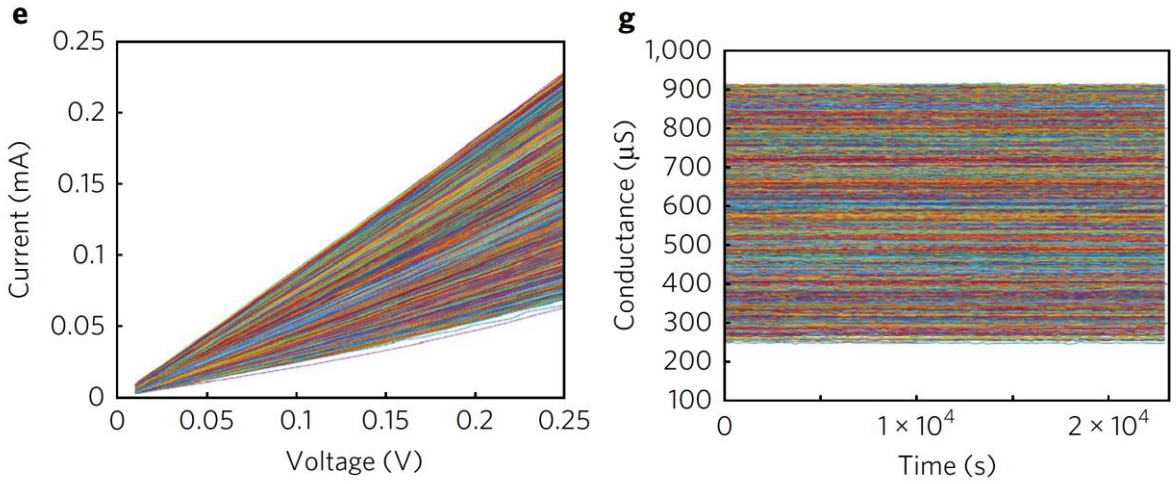
3 - find new ways to perform hardware-compatible learning (bottom-up)

Neuroscience & AI inspired

1- Implement backpropagation

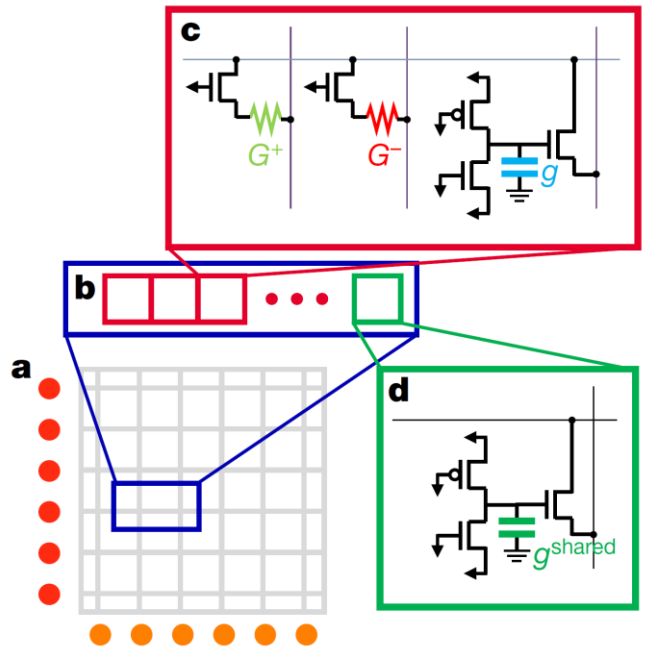
Efforts on hardware side to improve nanodevice properties

- Optimizing memristor properties



Li et al, Nature electronics **1**, 52–59 (2018)

- Complementing memristors with high accuracy weight



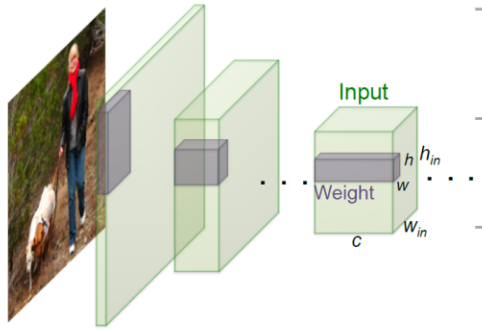
Accuracy on MNIST 98%

Ambrogio et al, Nature 558, 60 (2018)

Efforts on algorithmic side to decrease required precision though pruning and quantization

Quantized Guided Pruning for Efficient Hardware Implementations of Deep Neural Networks
GB Hacene, V Gripon, M Arzel, N Farrugia, Y Bengio
2020 18th IEEE International New Circuits and Systems Conference (NEWCAS …

XNOR nets, Rastegari et al, arXiv:1603.05279 → binary synapses at inference



	Network Variations	Operations used in Convolution	Memory Saving (Inference)	Computation Saving (Inference)	Accuracy on ImageNet (AlexNet)
Standard Convolution	Real-Value Inputs 0.11 -0.21 … -0.34 / -0.25 0.61 … 0.52; Real-Value Weights 0.12 -1.2 … 0.41 / -0.2 0.5 … 0.68	+ , − , ×	1x	1x	%56.7
Binary Weight	Real-Value Inputs 0.11 -0.21 … -0.34 / -0.25 0.61 … 0.52; Binary Weights 1 -1 … 1 / -1 1 … 1	+ , −	~32x	~2x	%56.8
BinaryWeight Binary Input (**XNOR-Net**)	Binary Inputs 1 -1 … -1 / -1 1 … 1; Binary Weights 1 -1 … 1 / -1 1 … 1	XNOR , bitcount	~32x	~58x	%44.2

However, high precision weights and large networks still required for training

Hybrid Analog-Digital Learning with Differential RRAM Synapses
T Hirtzlin, M Bocquet, M Ernoult, JO Klein, E Nowak, E Vianello, JM Portal, D Querlioz
2019 IEEE International Electron Devices Meeting (IEDM), 22.6. 1-22.6. 4

Three main approaches

1- implement backpropagation

AI inspired

2- make backpropagation more hardware-compatible (top-down)

3 - find new ways to perform hardware-compatible learning (bottom-up)

Neuroscience & AI inspired

Geoffrey Hinton
AI pioneer
Turing Prize

Stanford Seminar - Can the brain do back-propagation?

Can the brain do backpropagation?

2- make backpropagation more hardware-compatible (top-down)

Use random weights for propagating error backwards

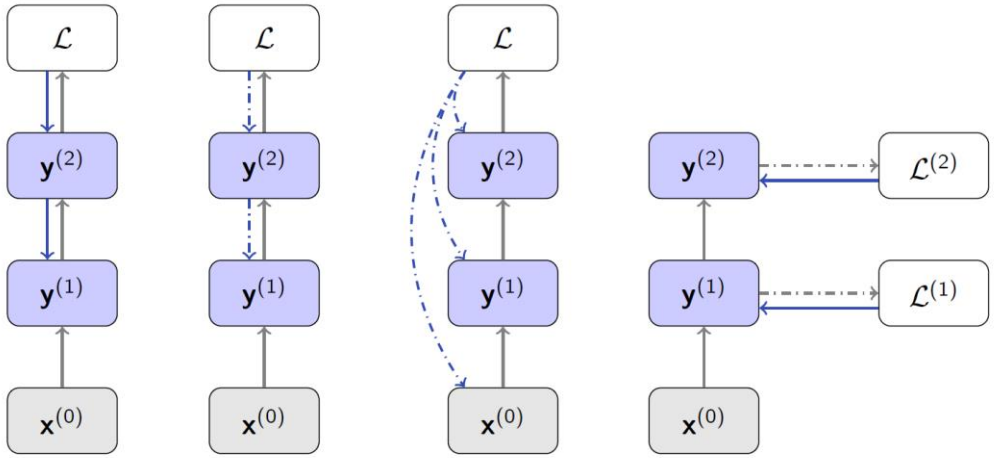
ARTICLE

Received 7 Jan 2016 | Accepted 16 Sep 2016 | Published 8 Nov 2016

DOI: 10.1038/ncomms13276 OPEN

Random synaptic feedback weights support error backpropagation for deep learning

Timothy P. Lillicrap[1,2], Daniel Cownden[3], Douglas B. Tweed[4,5] & Colin J. Akerman[1]



(a) BP (b) FA (c) DFA (d) Local Errors

Direct Feedback Alignment Provides Learning in Deep Neural Networks

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

Direct Feedback Alignment Scales to Modern Deep Learning Tasks and Architectures

Julien Launay[1,2] **Iacopo Poli**[1] **François Boniface**[1] **Florent Krzakala**[1,2]
[1]LightOn [2]École Normale Supérieure
{julien, iacopo, francois, florent}@lighton.ai

https://arxiv.org/pdf/2006.12878.pdf

Make backpropagation compatible with spiking neural networks

ARTICLE

https://doi.org/10.1038/s41467-020-17236-y OPEN

Check for updates

A solution to the learning dilemma for recurrent networks of spiking neurons

Guillaume Bellec[1,2], Franz Scherr[1,2], Anand Subramoney[1], Elias Hajek[1], Darjan Salaj[1], Robert Legenstein[1] & Wolfgang Maass[1✉]

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

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

Learning signal Eligibility trace

S4NN: temporal backpropagation for spiking neural networks with one spike per neuron

Saeed Reza Kheradpisheh[1,*] and Timothée Masquelier[2]

[1] *Department of Computer and Data Sciences, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran*

[2] *CERCO UMR 5549, CNRS - Universite Toulouse 3, Toulouse, France*

Journals & Magazines > IEEE Signal Processing Magazine > Volume: 36 Issue: 6

Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks

Publisher: IEEE

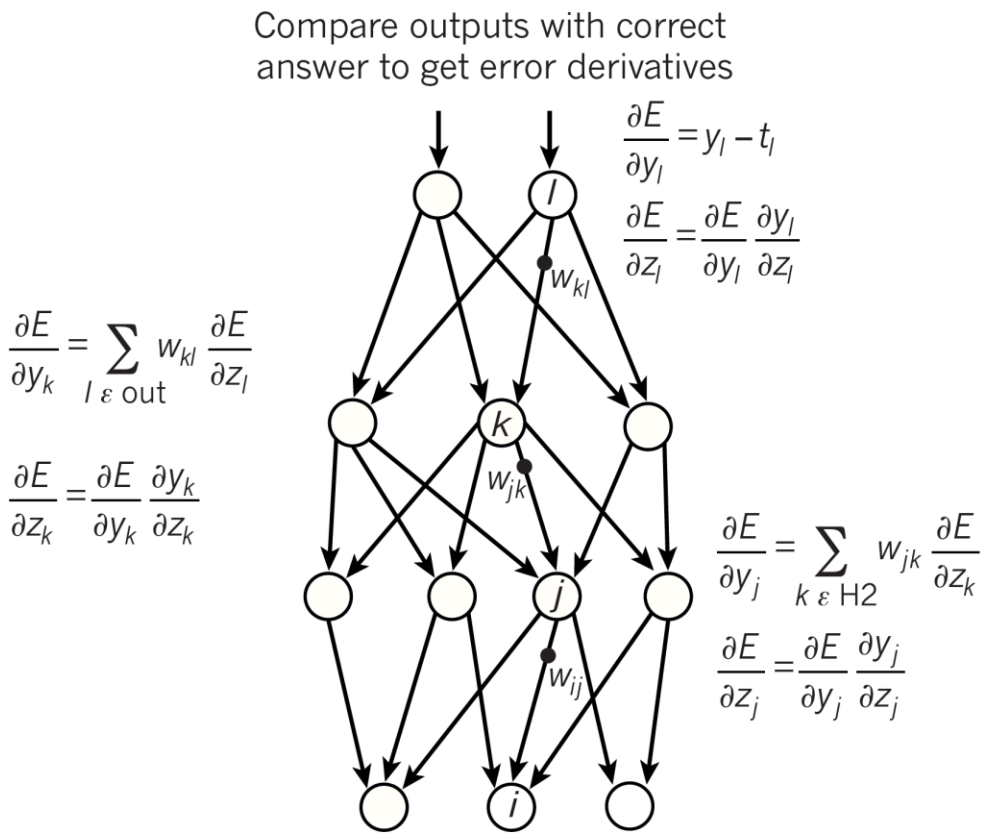
Cite This PDF

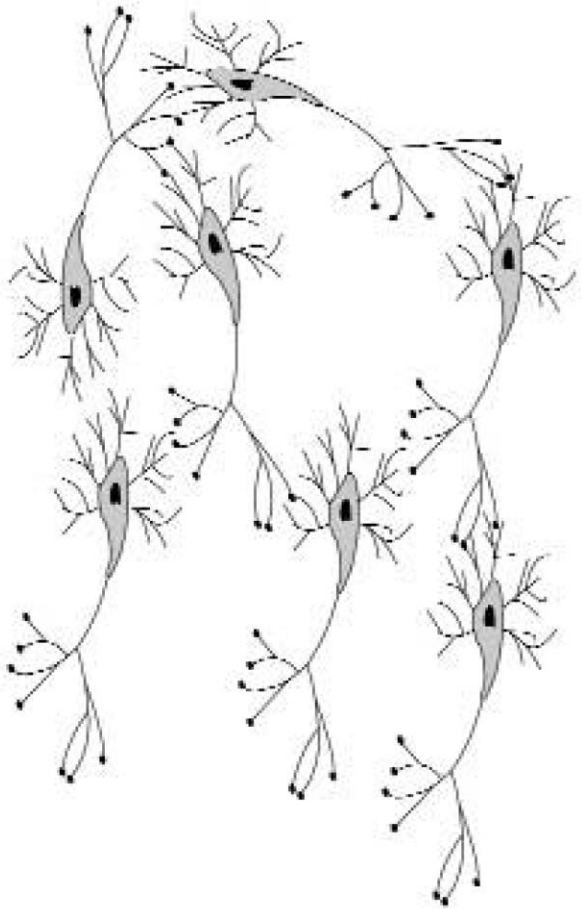
Emre O. Neftci ; Hesham Mostafa ; Friedemann Zenke All Authors

3 - find new ways to perform hardware-compatible learning with high accuracy (bottom-up)

Backpropagation requires cumbersome external circuits and additional memories to store activations and gradients

Backward pass





There are no external circuits, no additional memories in the brain: how are gradients computed, stored and applied to synapses ?

Lillicrap et al, "Backpropagation and the brain", Nature Reviews Neuroscience (2020)

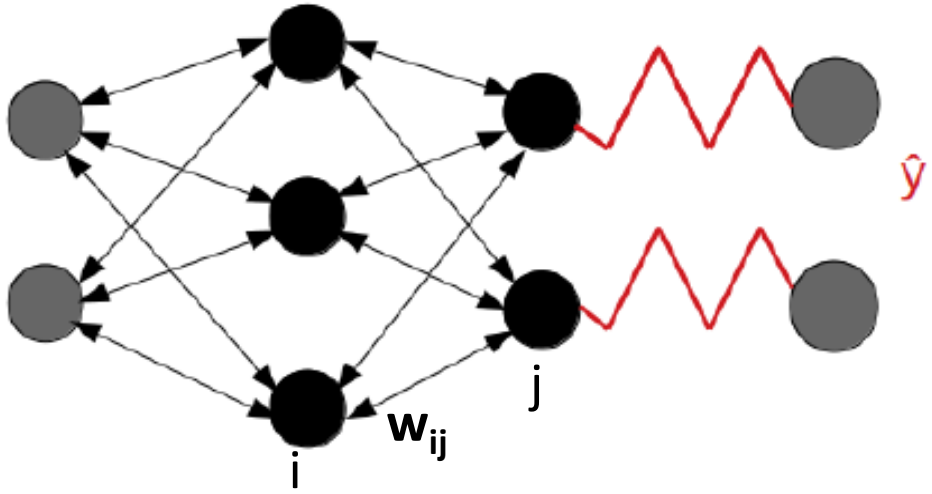
Learning through physics: networks that minimize their error at the same time as they minimize their energy

$$\frac{ds}{dt} = -\frac{\partial F}{\partial s}$$

$$F = E(s) + \beta C(y, \hat{y})$$

Cost function

B. Scellier &
Y. Bengio,
Front. Comput.
Neuroscience
04 May 2017



s → neuron state

ρ → neuron rate = neuron output

Learning rule: $$\frac{dw_{ij}}{dt} = \dot{\rho}(s_i)\rho(s_j) + \dot{\rho}(s_j)\rho(s_i)$$

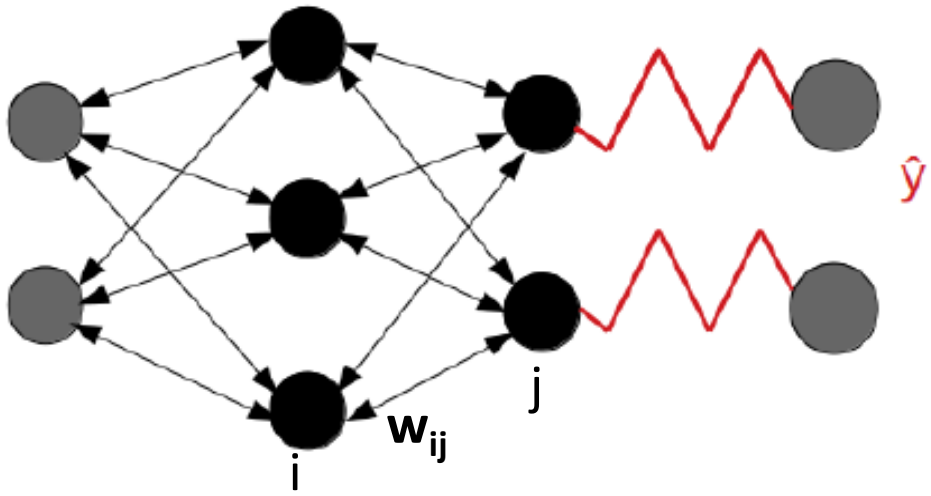
Learning through physics: networks that minimize their error at the same time as they minimize their energy

$$\frac{ds}{dt} = -\frac{\partial F}{\partial s}$$

$$F = E(s) + \beta C(y, \hat{y})$$

Cost function

B. Scellier &
Y. Bengio,
Front. Comput.
Neuroscience
04 May 2017



s → neuron state

ρ → neuron rate = neuron output

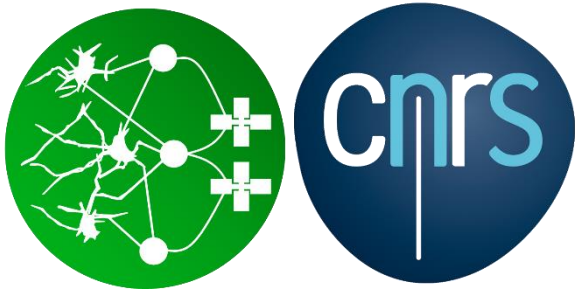
Learning rule: $$\frac{dw_{ij}}{dt} = \dot{\rho}(s_i)\rho(s_j) + \dot{\rho}(s_j)\rho(s_i)$$

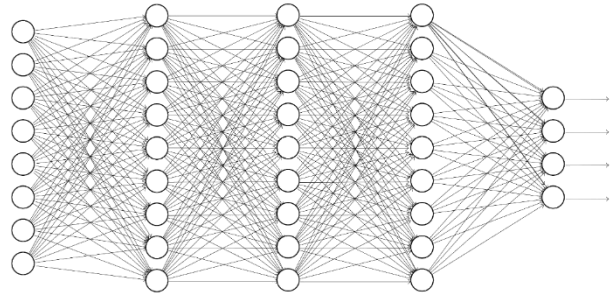
The EP learning rule is equivalent to Backpropagation through time M Ernoult, J Grollier, D Querlioz, Y Bengio, B Scellier, NeurIPS 2019

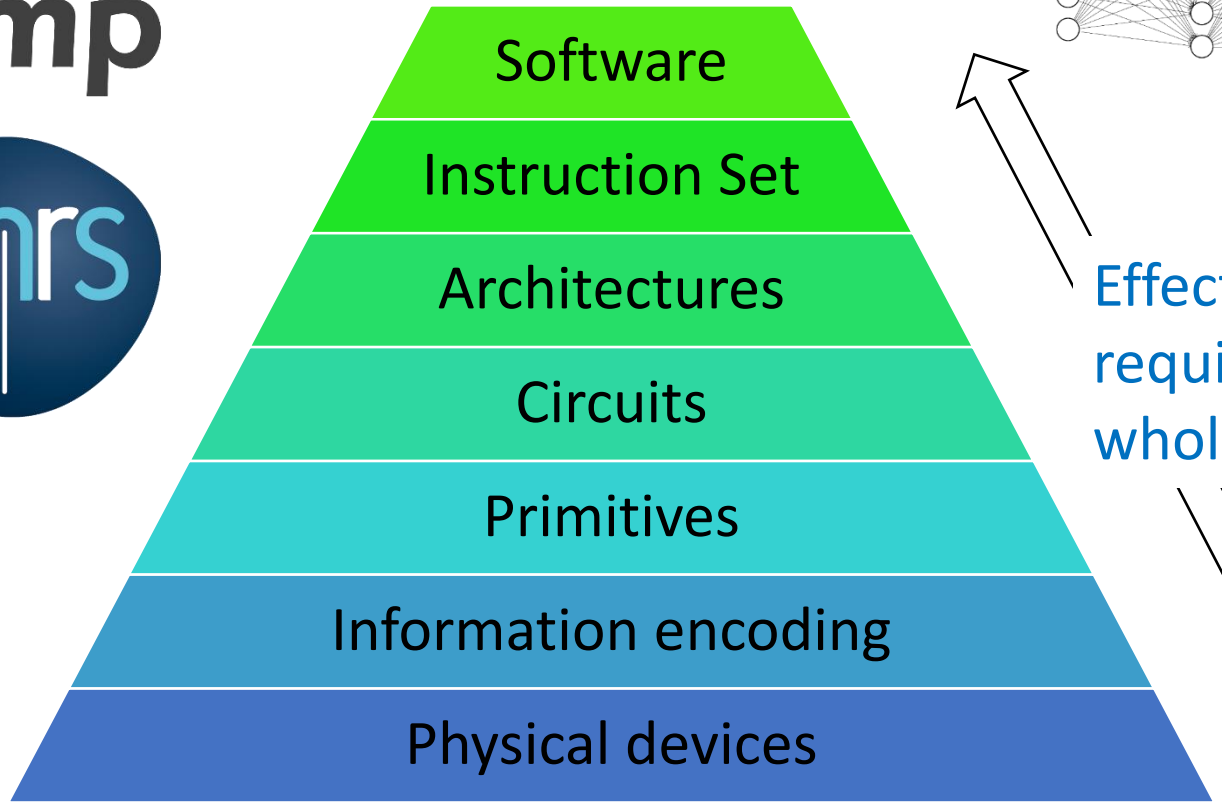
Conclusion

Future high performance, low power AI requires emerging nanotechnologies and physics

GDR BioComp







Effective use of new devices requires working across the whole computational stack