Neuromorphic computing

Frank Mizrahi, ESM 2024, York

laboratoire
Albert Fert

Neuromorphic Physics team

Julie
Grollier

Danijela
Marković

Dédalo Sanz
Hernandez

+ me

+ students and postdocs

laboratoire
Albert Fert

www.neurophysics.cnrs-thales.fr/

cnrs THALES Building a future we can all trust université PARIS-SACLAY

Neuromorphic Computing

Neuromorphic computing: why and what?

- Artificial neural networks
- The hardware problem: energy consumption
- Taking inspiration from the brain: the different approaches

Using emerging technologies: why and what?

- Key examples from spintronics and other technos
- Focus on RF spintronic neural networks

How to train neuromorphic systems?

Questions to have in mind for neuromorphic research

Artificial neural networks: algorithms and hardware

1) **Algorithms**
2) Hardware

laboratoire
Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

Artificial intelligence applications

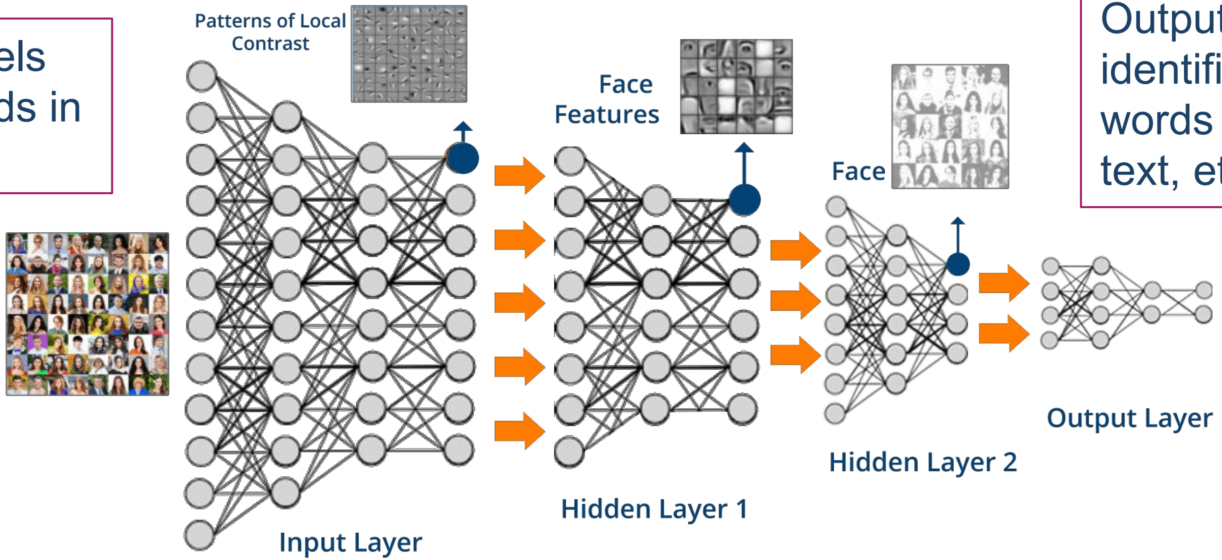
- Natural Language Processing (understanding and generating text)
- Image generation and recognition
- Time-series classification and prediction
- Finding patterns in data

A few examples:
Agriculture: automatic inspection of crops
Medicine: images, vital signs from sensors
Autonomous vehicles
Personal assistants
Industry: maintenance, tools

Artificial neural networks

- Flow of info in **non-linear** function, **tunable** parameters
- **Hierachical** structure inspired from cortex
- You can see it as a **giant fitting function**
- **Topology** depends on type of tasks: CNN with filters for vision, Transformers for text etc.

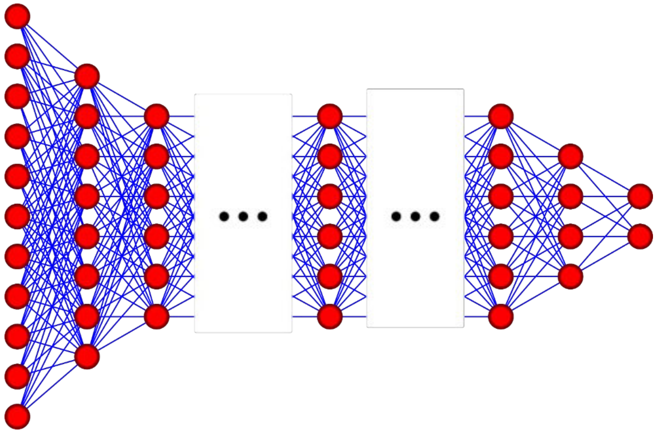
Inputs are pixels of image, words in a text, etc.



Outputs are identification of image, words in a generated text, etc.

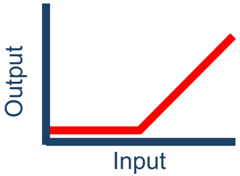
Basic blocks in a neural network

Inputs

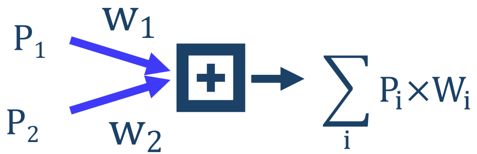


Outputs

Neurons
Non-linear activation function



Synapses: memory
Matrix multiplications with **tunable weights**



laboratoire Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

The weights are learned with data

- **Supervised**

You have “labeled data”: you know the correct output

Pro: gives the best performance

Con: requires labeled data

- **Reinforcement**

You only know if the output is “good” or “bad’

Relevant for some tasks such as gaming

- **Unsupervised**

You have no knowledge of the correct output. The network finds patterns and clusters data.

Pro: the most adapted to the real world

Con: performance is not as good. Typically combined with some supervised learning.

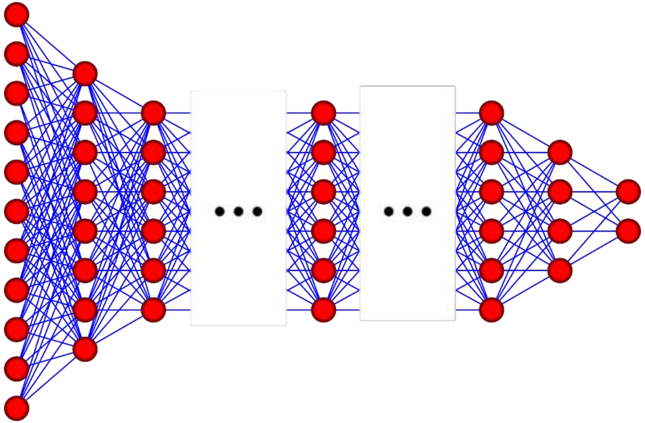
Supervised learning

We have a dataset with **inputs** and **targets** ("correct output")
We **split the dataset** into training and test sets

1) **Train** the network with training dataset
   - Show examples and compute the loss (i.e. error)
   - Update the weights to minimize the loss
   - Repeat many times!
2) **Test** the network with test dataset

How do you know how to update the weights???

Inputs



Outputs

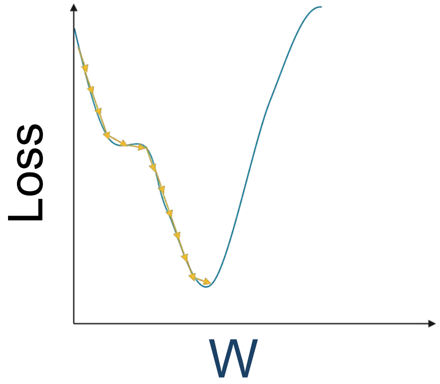
Loss = f(outputs, targets)

We use gradient descent and backpropagation to update the weights

- Gradient descent is a method to update the weights

$$W = W + \alpha \frac{\partial L}{\partial W}$$

α is the learning rate



- Backpropagation is a method to compute the loss gradient

$$\frac{\partial L}{\partial W} =?$$

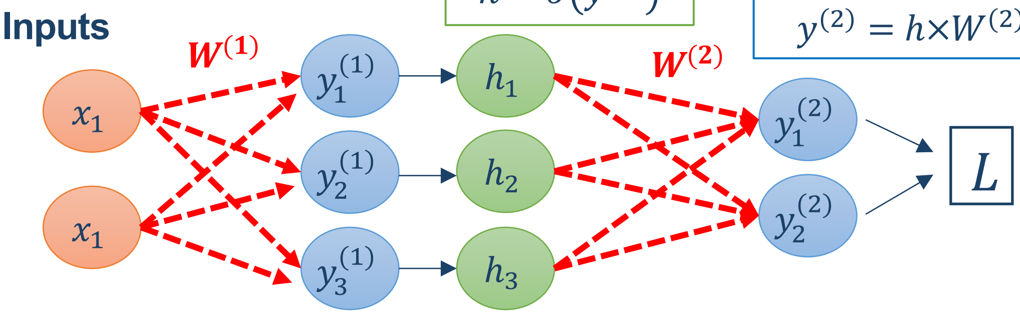
BackProp computes gradients using the chain rule

Forward

$$y^{(1)} = x_1 \times W^{(1)} + b^{(1)}$$

$$h = \sigma\left(y^{(1)}\right)$$

$$y^{(2)} = h \times W^{(2)} + b^{(2)}$$



$$\frac{\partial L}{\partial W_{i,j}^{(2)}} = \frac{\partial y_j^{(2)}}{\partial W_{i,j}^{(2)}} \times \frac{\partial L}{\partial y_j^{(2)}} = \frac{\partial L}{\partial y_j^{(2)}} \times h_i$$

$$\frac{\partial L}{\partial W_{i,j}^{(1)}} = \sum_k \frac{\partial y_j^{(1)}}{\partial W_{i,j}^{(1)}} \times \frac{\partial h_j}{\partial y_j^{(1)}} \times \frac{\partial y_k^{(2)}}{\partial h_j} \times \frac{\partial L}{\partial y_k^{(2)}} = \sum_k \frac{\partial L}{\partial y_k^{(2)}} \times W_{j,k}^{(2)} \times \sigma'\left(y_j^{(1)}\right) \times x_i$$

Backward

Recap of neural network algorithms

- NN are functions with a huge number of tunable parameters
- The basic blocks are non-linear activation functions (“neurons”) and tunable matrix multiplications (”synapses”)
- The parameters are learned from data
- Learning relies on computing an error and estimating how much each weight contributes to it

Questions?

Artificial neural networks: algorithms and hardware

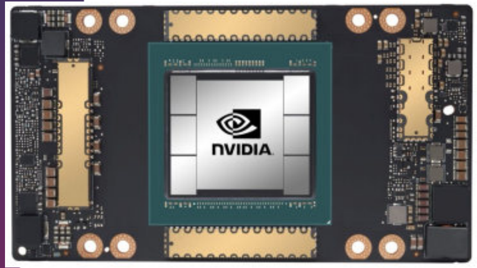
1) Algorithms
2) **Hardware**

laboratoire **Albert Fert**

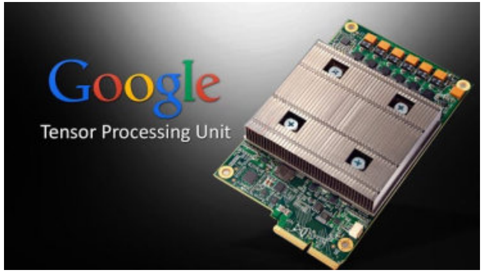
cnrs THALES Building a future we can all trust université PARIS-SACLAY

Modern hardware for neural networks

- CPUs are not optimal for neural networks (“fast car”: successive operations, very fast)
- GPUs/TPUs are better for matrix multiplications (‘big truck”: many identical operations in parallel)



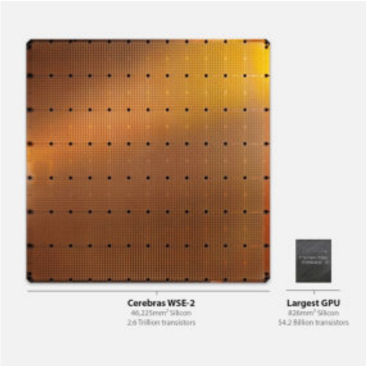
GPU
Conventional computing platform
High accuracy
10^2 GOPS/W



TPU

Digital CMOS ASICs
10^3-10^4 TOPS/W

Power
100-300 W



“World largest chip” (ASIC)
7 nm TSMC
Power > 15 kW
$ 2 M / chip

laboratoire Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

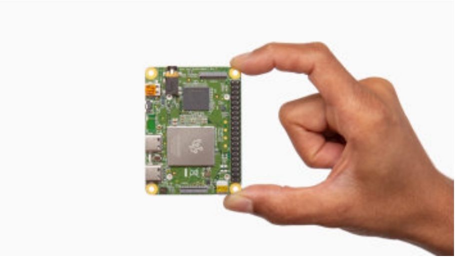
We have efficient AI edge accelerators…

- Google Edge TPU vs TPU

Infrastructure that works with you

By connecting edge tools with Google Cloud, you can deploy solutions that can bridge the functionality of the cloud with the availability of edge computing.

	Edge (Devices/nodes, Gateways, Servers)	Google Cloud
Tasks	ML inference	ML training and inference
Software, services	Linux, Windows	AI Platform, Kubernetes Engine, Compute Engine, Cloud IoT Core
ML frameworks	TensorFlow Lite, NN API	TensorFlow, scikit-learn, XGBoost, Keras
Hardware accelerators	Edge TPU, GPU, CPU	Cloud TPU, GPU, and CPU



Google Edge TPU

…but they are limited to inference

Why is this a problem?

- Huge consummation of datacenters
- Personal devices: learning on edge is required for privacy

Neuromorphic computing: Why and what?

1)Taking inspiration from the brain

2) Different approaches in neuromorphic computing

laboratoire **Albert Fert**

cnrs THALES Building a future we can all trust université PARIS-SACLAY

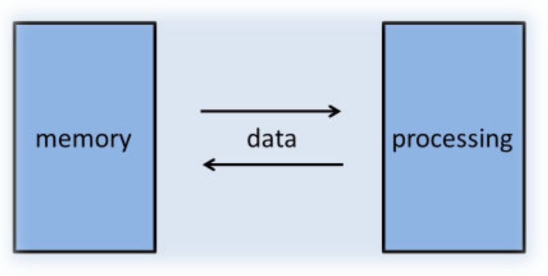
Taking inspiration from the brain

Warning….

- Not clear what brain does
- Not clear what brain does and is good for energy efficiency
- Some ideas seem necessary to reduce energy consumption, others are more up to debate
- People can have very different opinions on these topics…

Memory and computing should be close

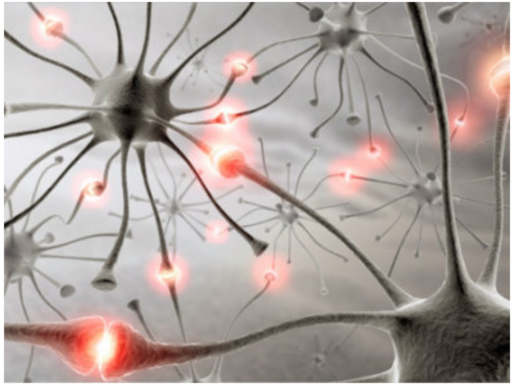
Computer:
Von Neumann architecture



Operation	Energy consumption
Addition of data	1x
Access data (onchip cache)	60x
Access data (offchip RAM)	3500x

Sze et al, IEEE Custom Integrated Circuits Conference (2017)

Brain:
Memory (synapses) and processing (neurons) are intertwined



laboratoire Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

We can work with reduced precision

Computer:

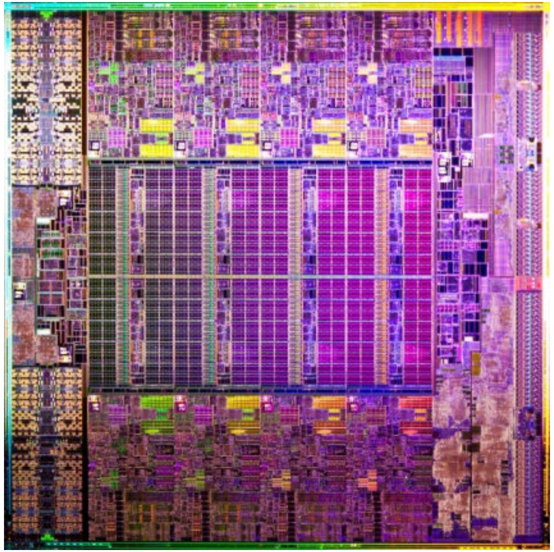
64 bits floating point accuracy

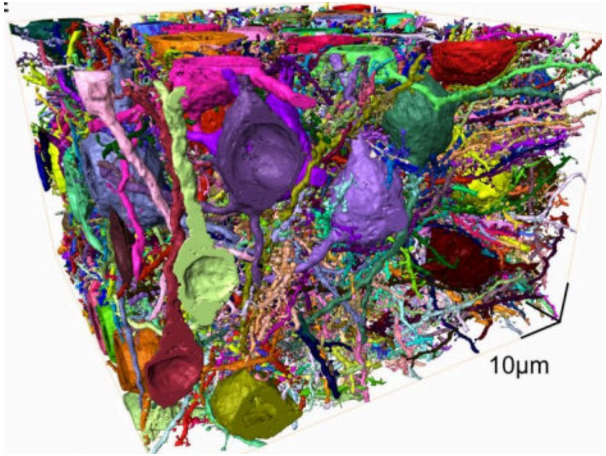
Brain:

Biology is messy!

=> The requirements to build a **neural network accelerator** (ex: understanding text, classifying data) are not the same as for a **general purpose computer** (ex: accounting, scientific simulations).

3D unlocks high density

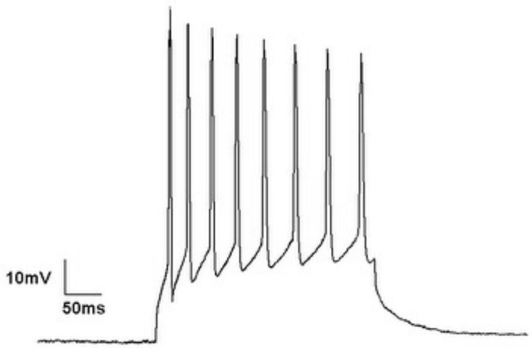


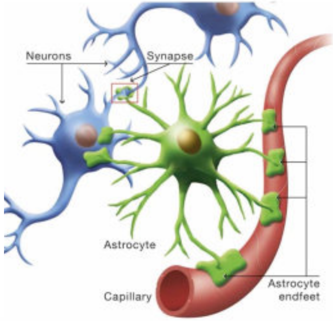


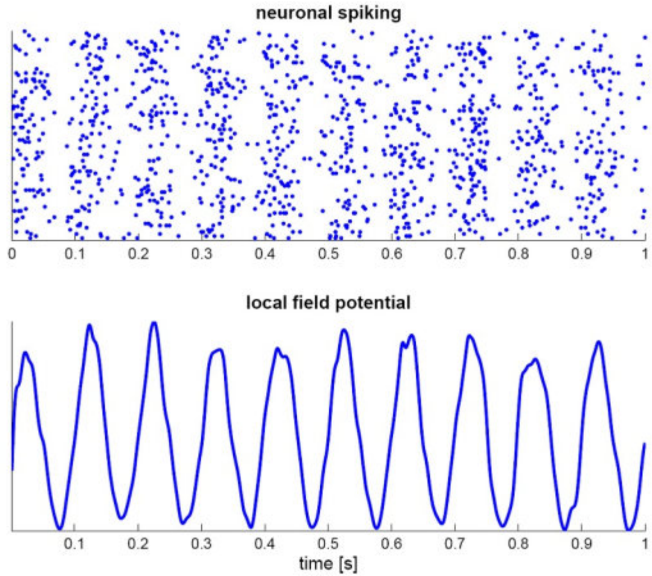
Motta et al, Science, 366, 6469 (2019)

3D has many promises but very challenging (access, heat, etc)

Spikes and dynamics?







spikes,

oscillations,

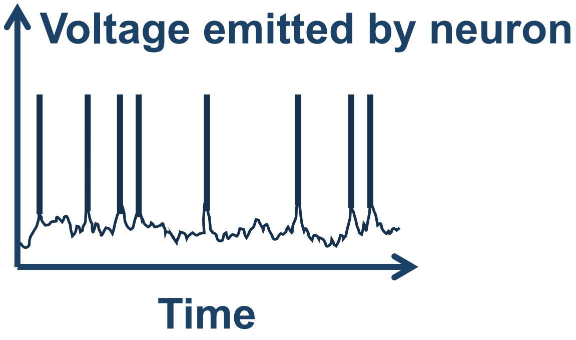
synchronization,

non-linear dynamics

They seem to play an important role in the brain..

…Are they useful for developing artificial systems?

Stochasticity and noise?



Spike trains of some neurons seem stochastic

Biology is noisy

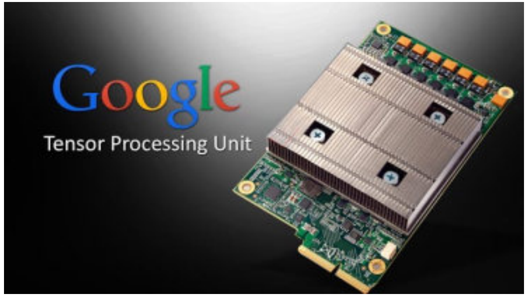
Individual synapses are unreliable

Attractive idea: we could work with noisy, imperfect, unreliable devices!

But… might be naïve view of brain, and not clear that it is the best for developing artificial systems?

Neuromorphic ideas are already in recent mainstream hardware

- Memory and computing closer
- Fixed-point precision (8 bit for edge TPU!)



TPU

Digital CMOS ASICs
10^3-10^4 TOPS/W
Memory and computing closer
Fixed-point precision

Neuromorphic computing: Why and what?

1)Taking inspiration from the brain

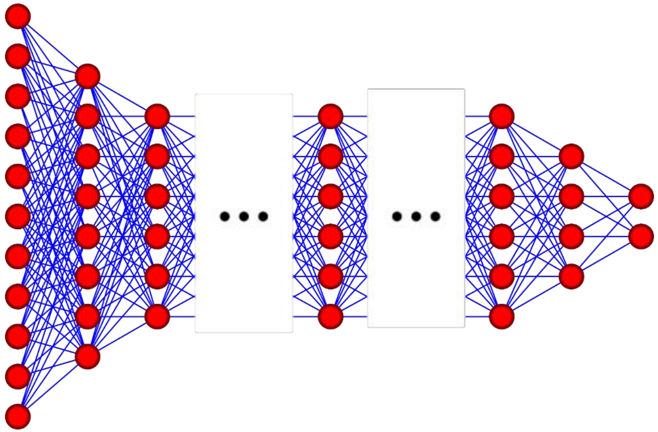
2) Different approaches in neuromorphic computing

laboratoire
Albert Fert

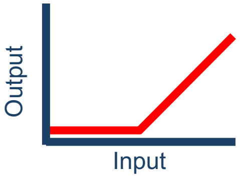
cnrs THALES Building a future we can all trust université PARIS-SACLAY

AI-inspired

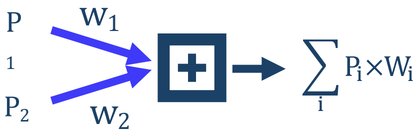
- What is used in applications
- Deep feed-forward networks
- Neurons are static
- Trained by backpropagation



<u>Neurons</u>
Non-linear activation function

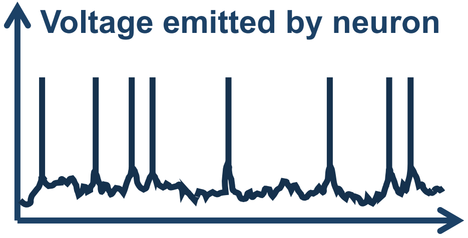


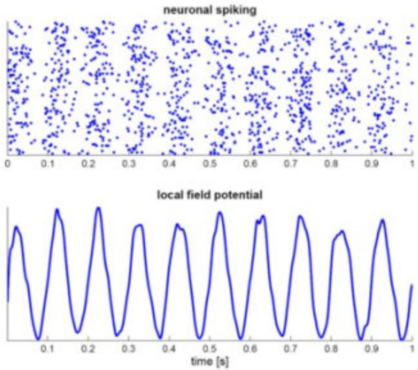
<u>Synapses: memory</u>
Matrix multiplications



Neuroscience-inspired

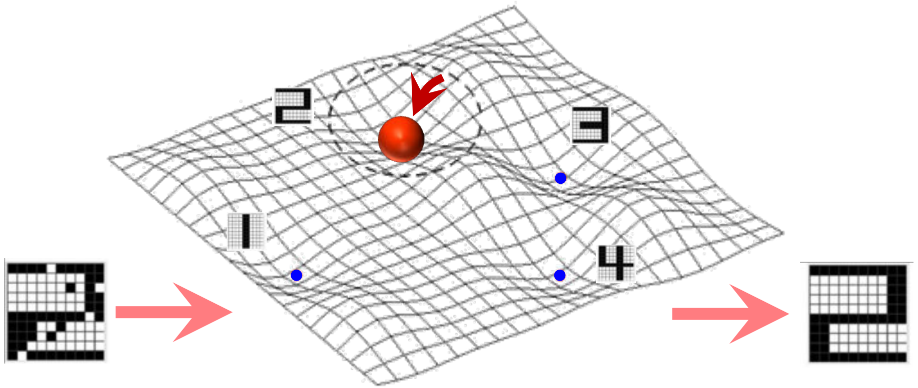
- Neurons are dynamic (spikes, oscillations etc.)
- Exotic learning (local learning, self-learning, unsupervised learning, etc.)





Physics-inspired

Idea: physical system will naturally go to energy minimum
=> let's make this minimum the result of our computation



Questions?

laboratoire
Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

Neuromorphic computing hardware

1) **Why emerging technologies?**
2) Examples of key ideas and realizations
3) Focus on RF spintronic networks

laboratoire
Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

CMOS based neuromorphic computing

- Developed since the 1980’s!
- Goals have evolved and were not mainly applicative

=> Emulate brain to understand it

=> Platform to test neuroscience algorithms

=> Brain-machine interface, prosthetics

=> Bio-inspired sensors (bio-inspired camera, cochlea etc.)

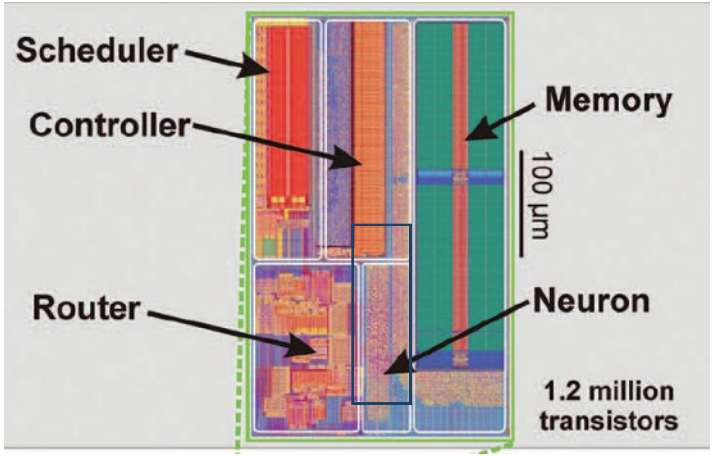
CMOS neurons and synapses are complex circuits

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

CMOS neuron

CMOS synapse

>10 µm

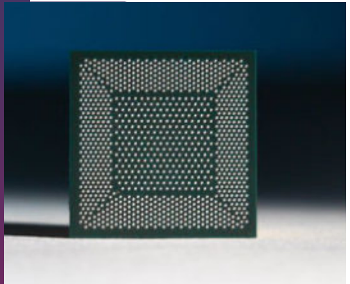


Merolla et al, *Science* **345**, 668 (2014)
Davies et al, *IEEE Micro*. **38**, 82–99 (2018)

BrainScales: 20 wafers, 4M neurons, 1B synapses

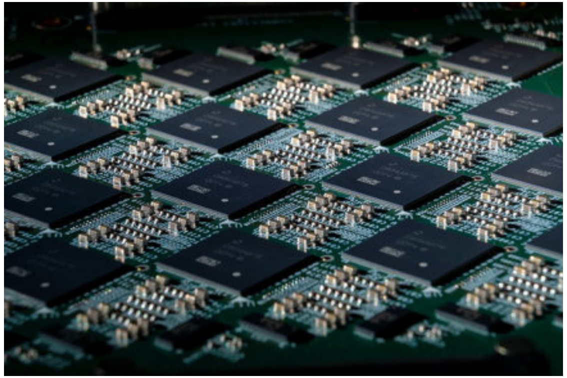
Neuromorphic CMOS chips are limited in the number of neurons and synapses they can include

Loihi Chip intel :
130k neurons



Spiking
Learning
MNIST
~ 1 Watt

Poihiki beach = 64 Loihi
8M neurons



E. Praxon Frady et al,
arXiv/2004.12691

Several Poihiki beach
100M neurons

Energy consumption

Strength and limits of CMOS

Strengths

- Versatile
- It actually works
- Super reliable, digital is self-correcting
- Established industry, tools etc.

Limits

- Memory is missing
- When going to analogue, advantages are not there

Neuromorphic computing hardware

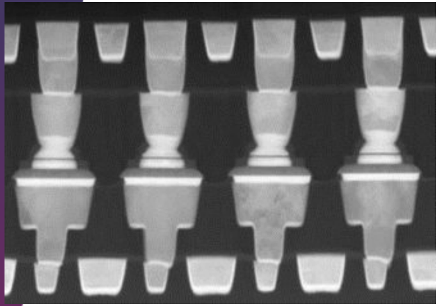
1) Why emerging technologies?
2) **Examples of key ideas and realizations**
3) Focus on RF spintronic networks

laboratoire
Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

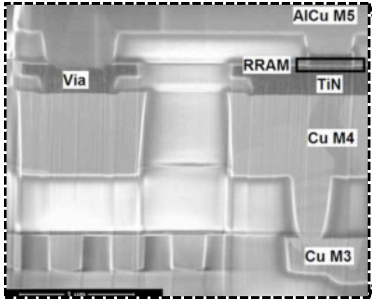
‘In-memory computing’ with nanodevices as non-volatile memory

Spintronics
MRAMs



Intel: MRAM integrated into 22nm FinFET CMOS, IEDM 2018

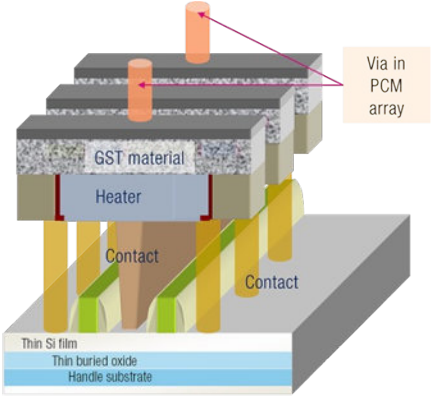
Resistive-Switching
ReRAMs



CEA LETI: 130nm CMOS + HfO_2 RRAM

Bocquet et al., IEEE IEDM, 2018

Phase Change



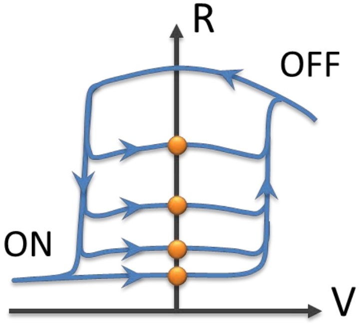
ST Microelectronics, IBM

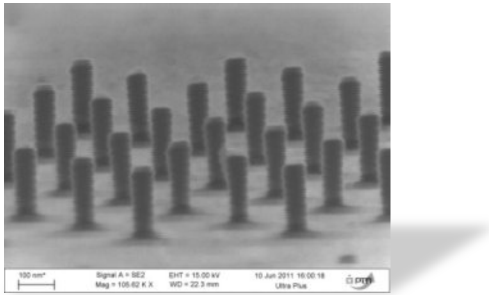
Gbit prototypes: billions of devices on a chip, monolithically intregrated with CMOS

Non-volatile memristors emulate synapses

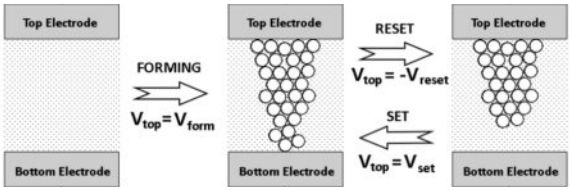
Variable resistor with memory

Chua, IEEE Trans. Circuit Theory (1971)



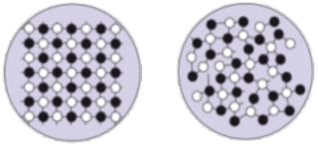


Filamentary switching



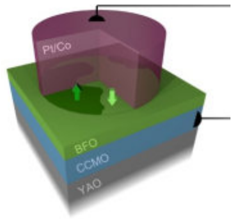
Yang et al., Nature Nano. (2013)

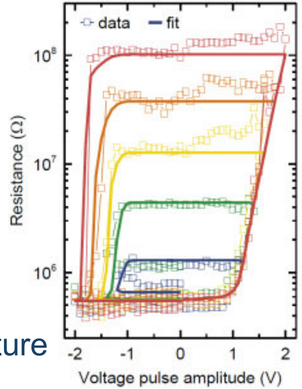
Phase change



Kuzum et al, Nanotechnology (2013)

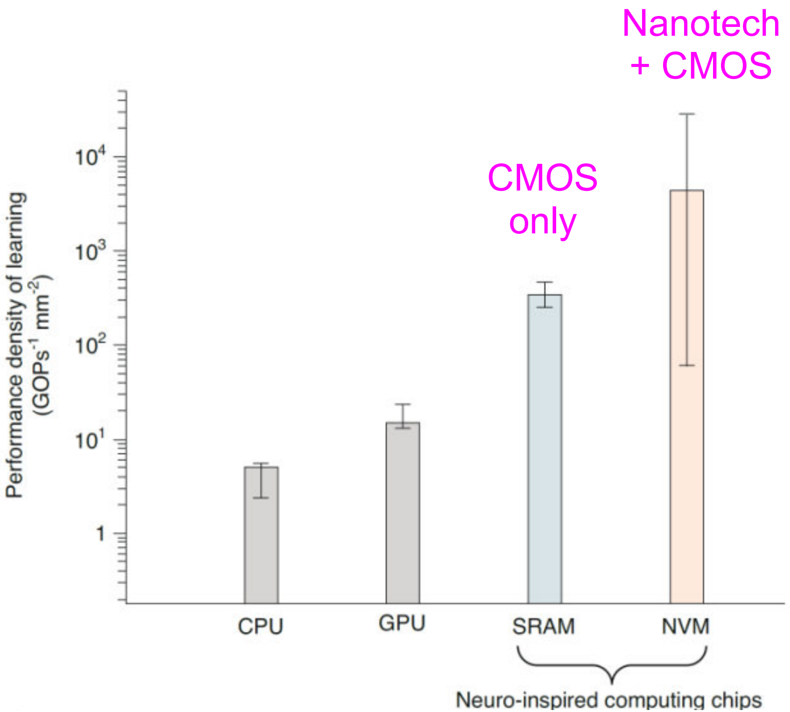
Ferroelectric

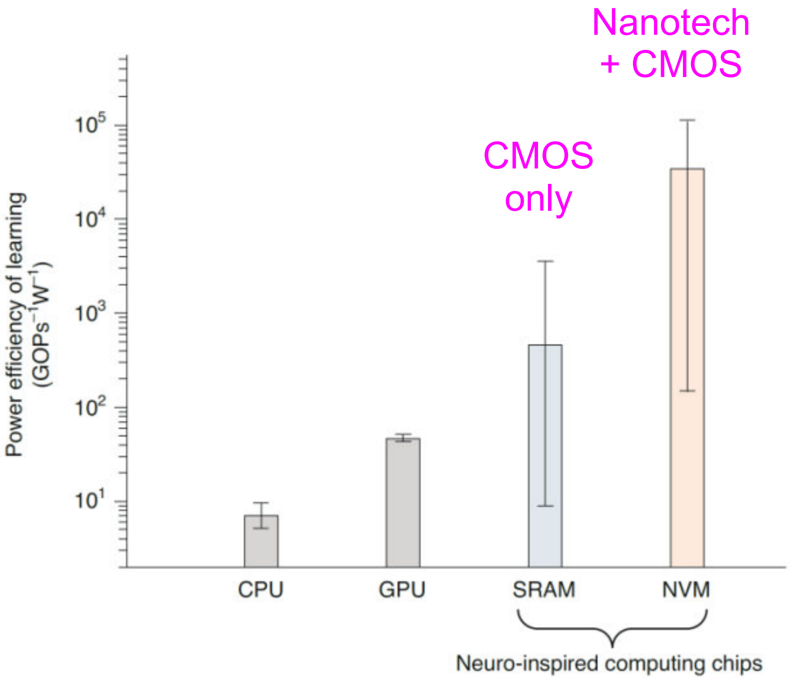




Chanthbouala et al, Nature Mat. (2012)

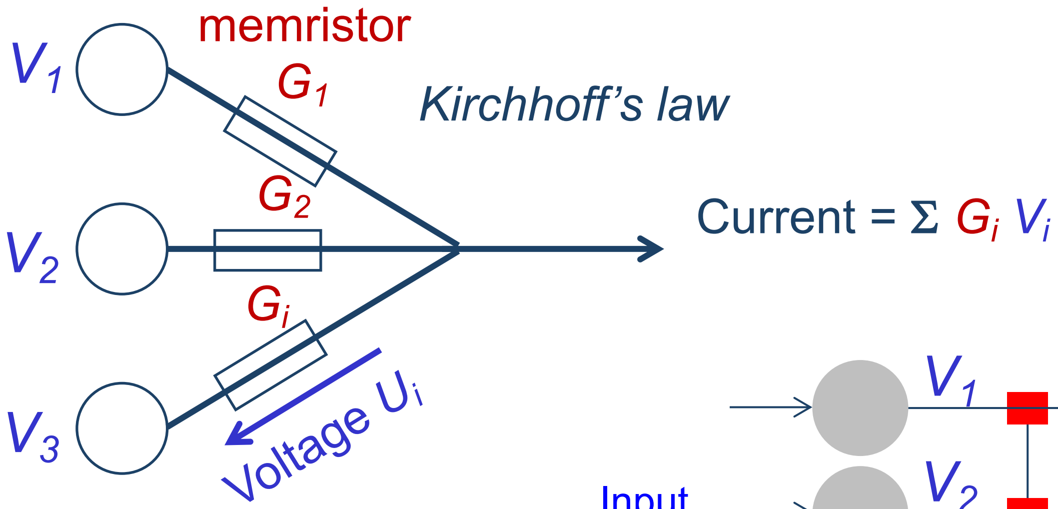
Non-volatile memory improves the efficiency

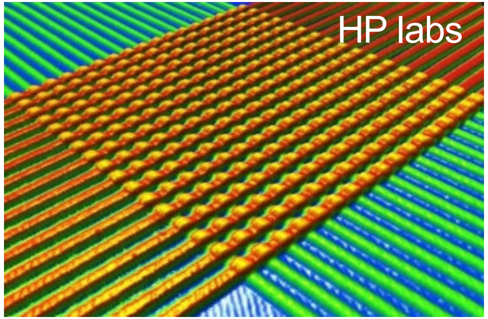


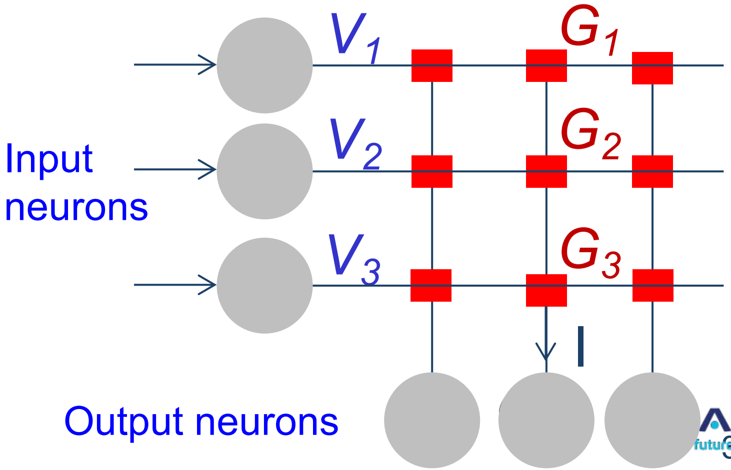


Zhang et al, Nature Electronics 3, 371 (2020)

Crossbar arrays of memristors physically implement matrix multiplication



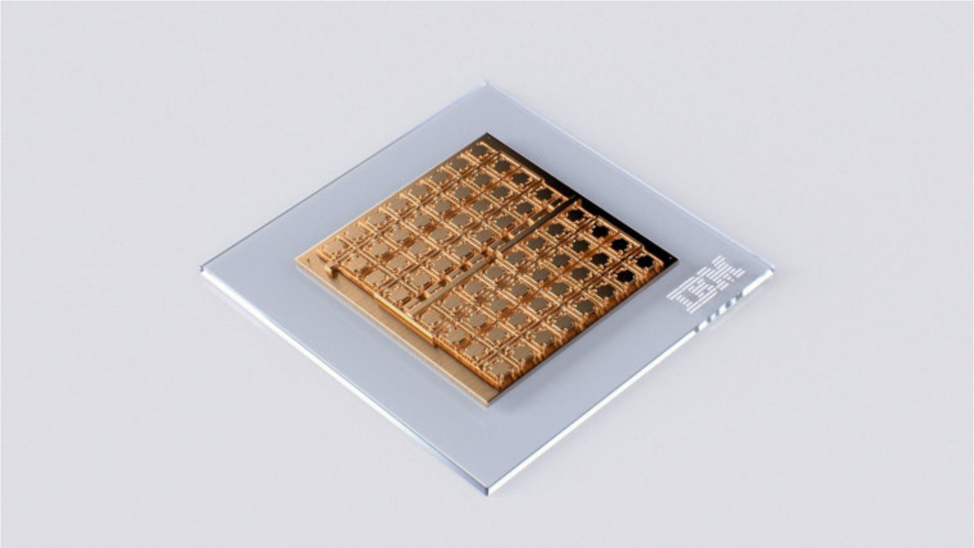




The physics is doing the computation!

State of the art

IBM analogue chip, *Nature*, 2023



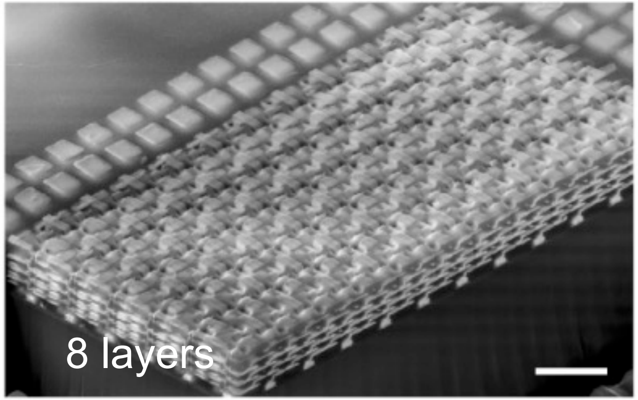
10+ millions synapses

Speech processing

10^4 GOPs/W

Limits

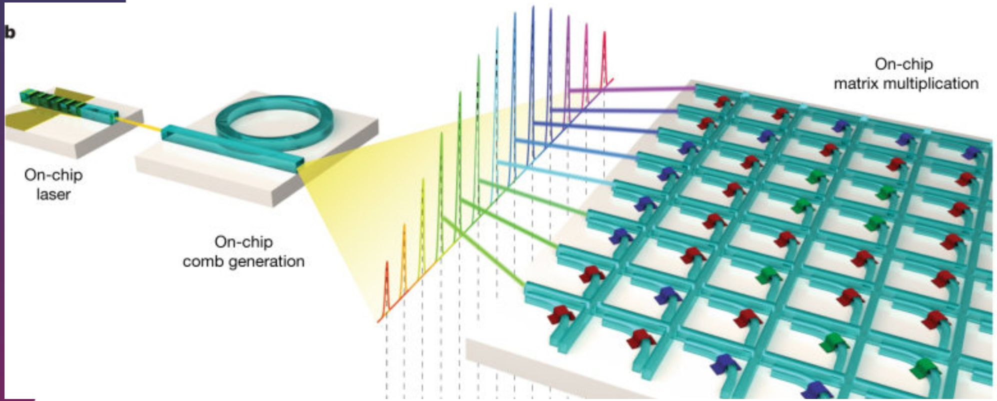
- Learning is hard to achieve: not easy to program the weights continuously (errors, non-linearities, lack of endurance)
- Architecture constrained to arrays with at least 1 transistor per nanodevice.
- 3D on the way but many challenges



Lin et al, Nature Electronics 3, 225 (2020)

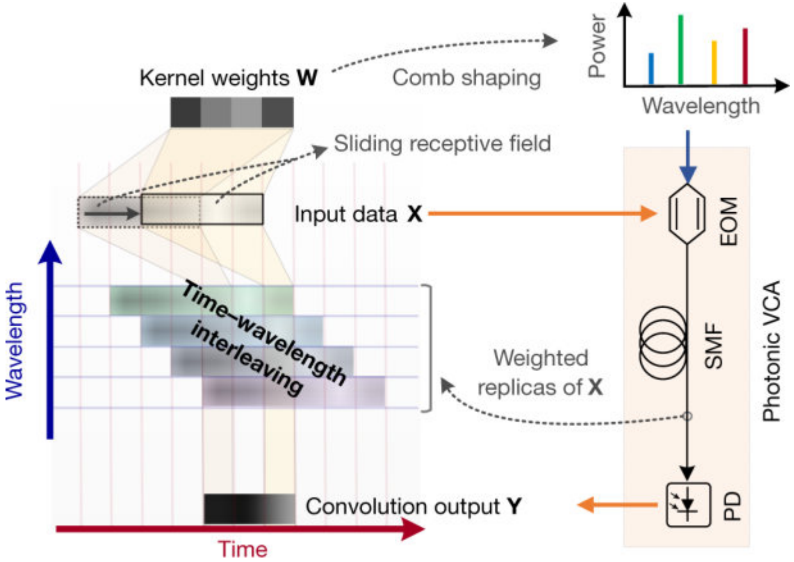
Photonics matrix multiplication

Ring resonators
Phase change devices as synapses



Feldmann, J., Youngblood, N., Karpov, M. *et al.* Parallel convolutional processing using an integrated photonic tensor core. *Nature* **589,** 52–58 (2021)

Mach-Zender interferometers



Xu, X., Tan, M., Corcoran, B. *et al.* 11 TOPS photonic convolutional accelerator for optical neural networks. *Nature* **589,** 44–51 (2021)

Strengths and limits of photonics

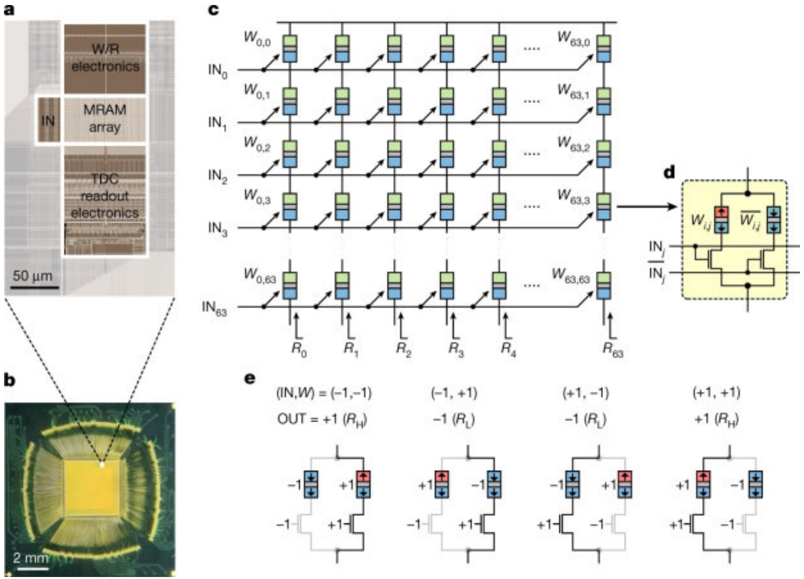
Strengths:

- Light is super fast!
- Waveguides can cross each other
- You can frequency multiplex over a huger frequency band and process many inputs in parallel
- Compatible with CMOS

Limits:

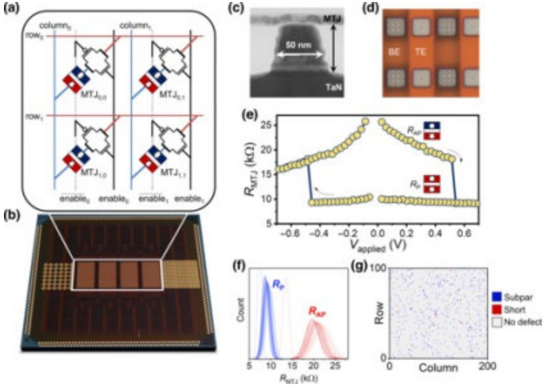
- Devices are much larger than nanodevices (micron size)
- Getting the non-linearity requires high power
- Conversion to electronics might remove the speed advantage

Spintronic synaptic arrays



64x64 synapses + CMOS

Jung, S., Lee, H., Myung, S. *et al.* A crossbar array of magnetoresistive memory devices for in-memory computing. *Nature* **601**, 211–216 (2022)



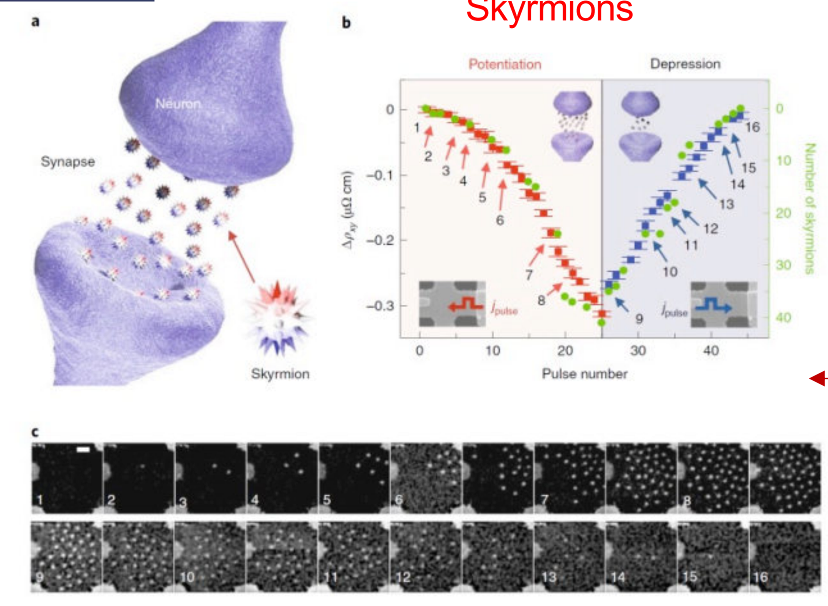
20,000 MTJs + CMOS

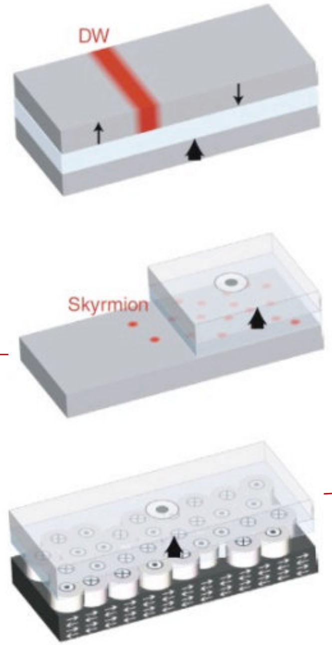
Borders et al. "Measurement-driven neural-network training for integrated magnetic tunnel junction arrays." *Physical Review Applied* 21.5 (2024)

CNRS THALES Building a future we can all trust université PARIS-SACLAY

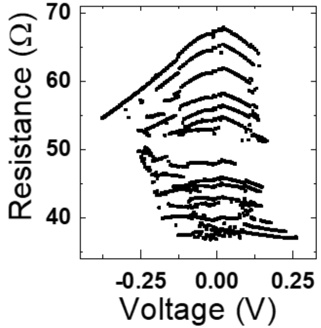
Multilevel synapses with magnetic textures

Skyrmions



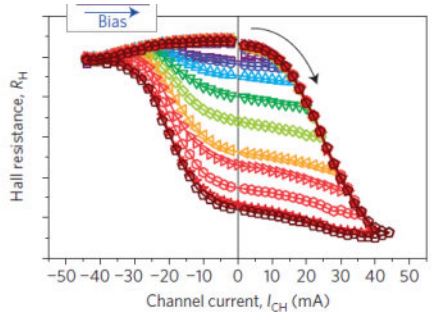


Domain wall



Lequeux, S *et al. Sci Rep* **6,** 31510 (2016)

Antiferro/ferro bilayer



K M Song et al, Nature Electronics 3, 148 (2020)
R Chen et al, Phys. Rev. Appl. 14, 014096 (2020)

laboratoire
Albert Fert

S. Fukami et al, Nat. Mater. 15, 535 (2016)

Strengths and limits of spintronics

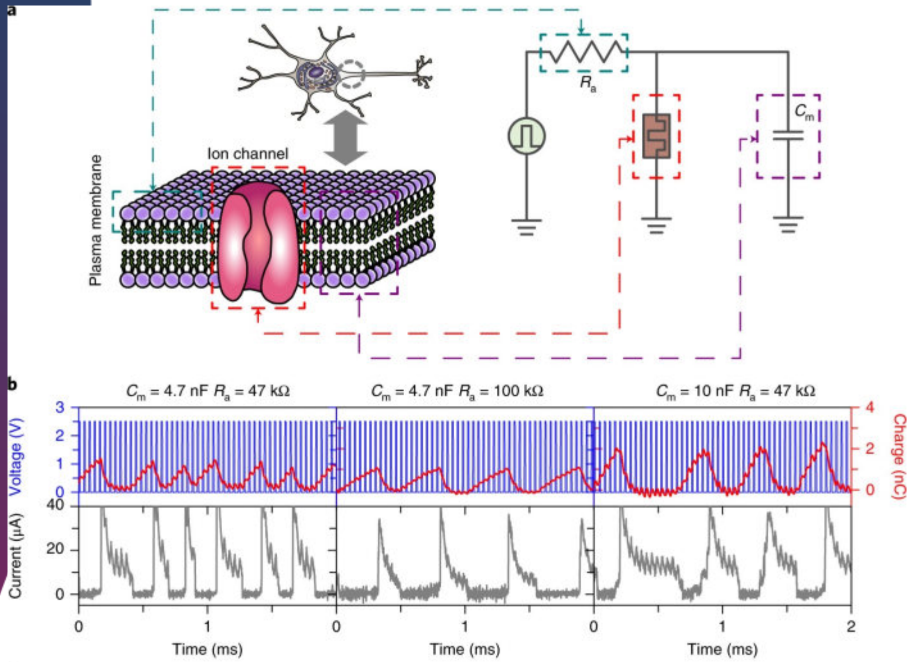
Strengths:

- MRAM is mature commercial techno
- Purely physical phenomena: endurance + predictive models
- Multifunctional
- High speed
- Nano

Limits (for now!):

- ON/OFF ratio is small (2-3 compared to > 10^4 in other technos)
- Multilevel synapses do not scale down well
- Smaller scale realizations

Spiking Neurons



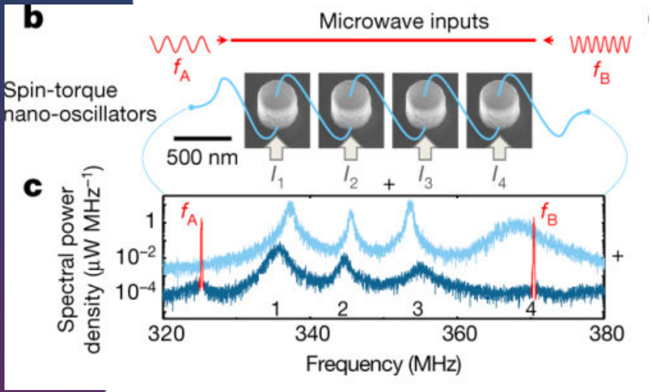
Wang et al, Nature Electronics, 1, 137 (2018)

- With volatile memristive devices
- With integrated photonics
- With spintronic devices

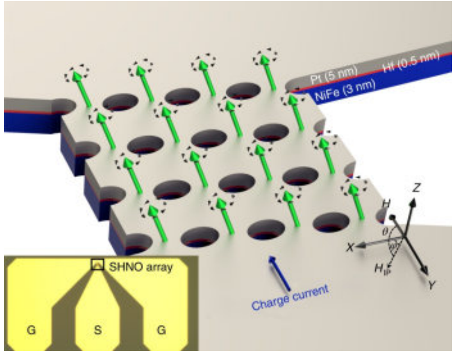
Pros:
- Complex bio-inspired behavior with compact device
- Spike enable bio-inspired learning rules (not BackProp)

Cons:
- Integration with synapses not easy
- Spiking network perfs behind BP

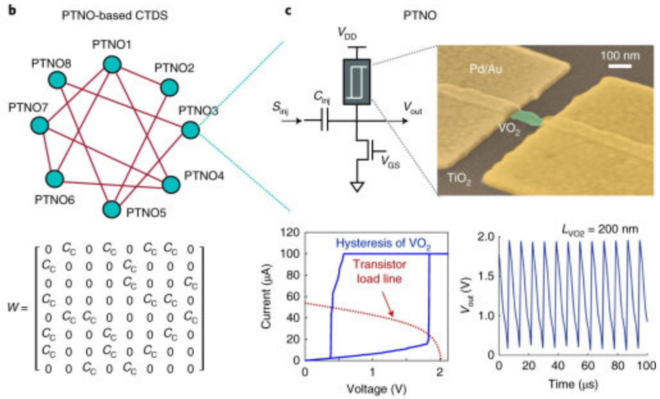
Leveraging oscillations



Romera et al. *Nature* (2018)



Zahedinejad, et al. *Nature nanotechnology* 15.1 (2020)



Dutta, *et al. Nat Electron* **4,** 502–512 (2021).

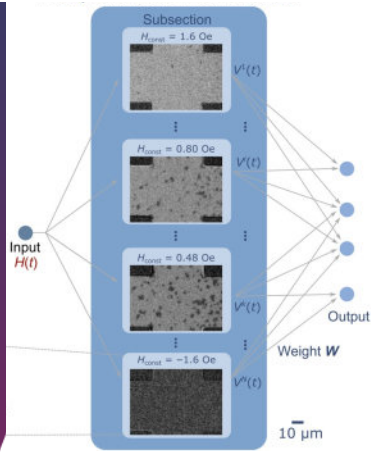
Network of coupled oscillators can store and retrieve patterns
Oscillations in the brain play role for learning
=> How to couple them efficiently?
=> Computing algorithm?

laboratoire Albert Fert

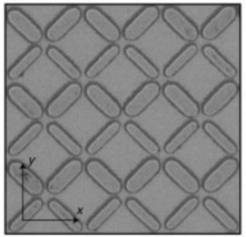
cnrs THALES Building a future we can all trust université PARIS-SACLAY

Leveraging non-linear dynamics

Interacting skyrmions



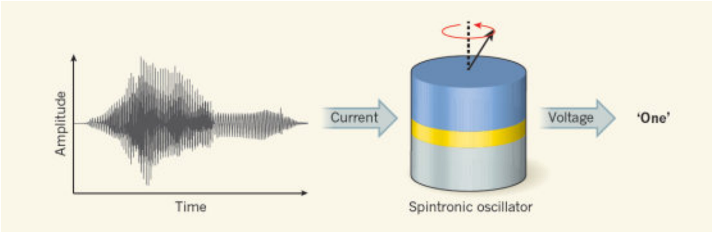
Yokouchi et al. *Science Advances* 8.39 (2022)



Spin-ices and spin waves

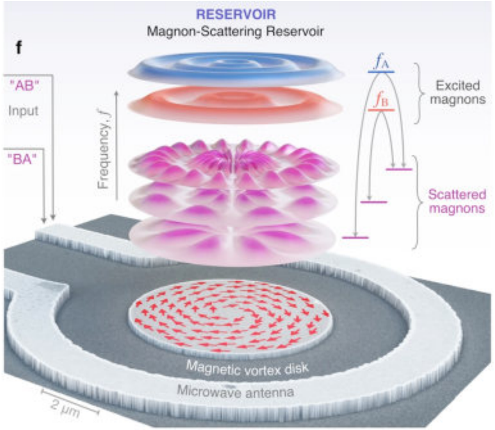
Gartside et al. *Nature Nanotechnology* 17.5 (2022)

Oscillator transient dynamics



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

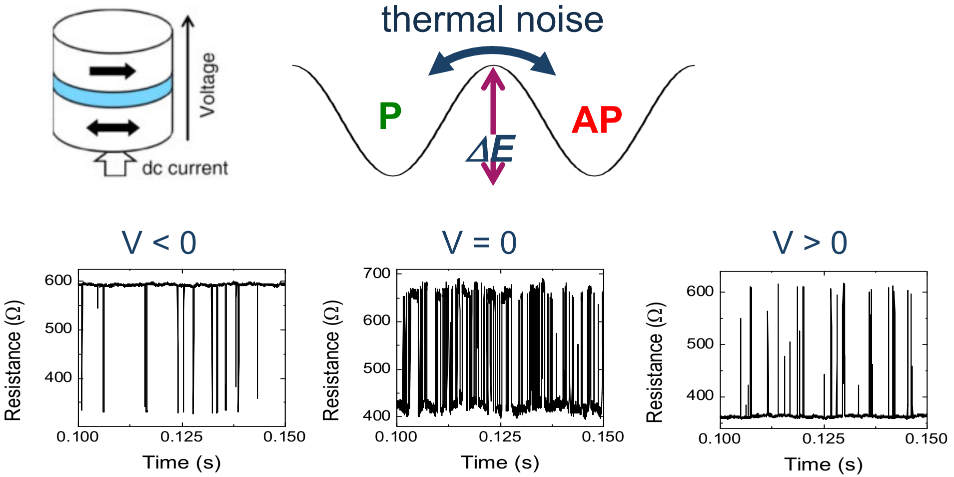
Magnon-scattering

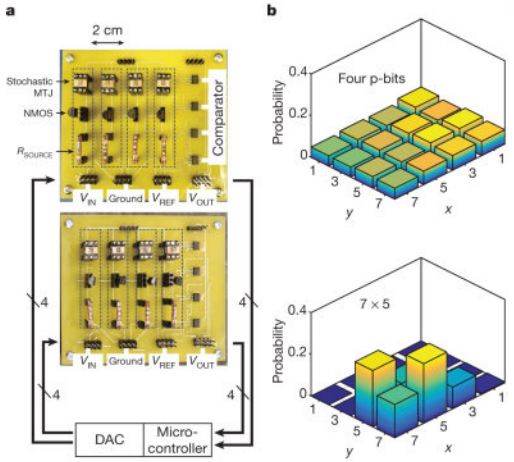


Körber et al. *Nature Communications* 14.1 (2023)

Take full advantage of physic of system
Next step is to go beyond reservoir to unlock complex tasks

Leveraging stochastic behavior





Neural networks

Mizrahi et al, Nature Communications 9 (1), 1 (2018)
Daniels et al, Phys. Rev. Applied, 13, 034016 (2020)

Ising machine

Borders et al. *Nature* 573.7774 (2019)

Thermal noise is leveraged for energy efficiency
Next step: high density coupling

laboratoire Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

Neuromorphic computing hardware

1) Why emerging technologies?
2) Examples of key ideas and realizations
3) **Focus on RF spintronic networks**

laboratoire
Albert Fert

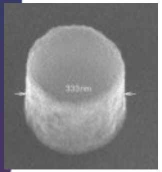
CNRS THALES Building a future we can all trust université PARIS-SACLAY

Problems we want to address

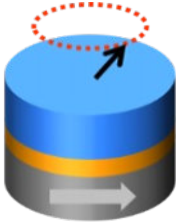
- Cascading layers to go towards deep networks
- Dense tunable connections to go toward complex tasks
- Think of full architecture and system integration

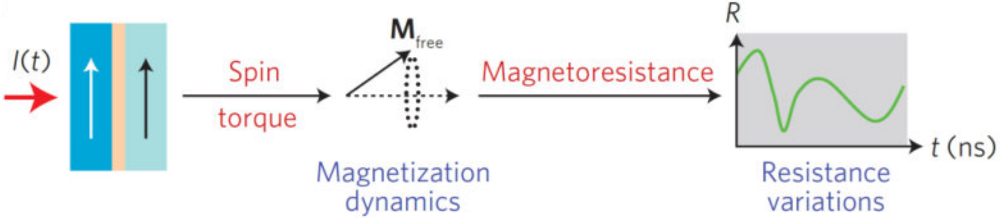
Magnetic Tunnel Junction as neuron and synapse

Electrical Read and Write
High speed dynamics



Ferromagnet
Tunnel barrier
Ferromagnet

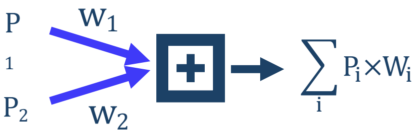




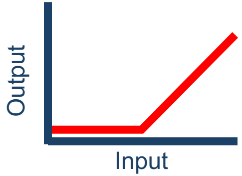
J. Grollier et al, PIEEE 104, 2024 (2016)

In this talk: AI-inspired approach

Tunable weighted sums



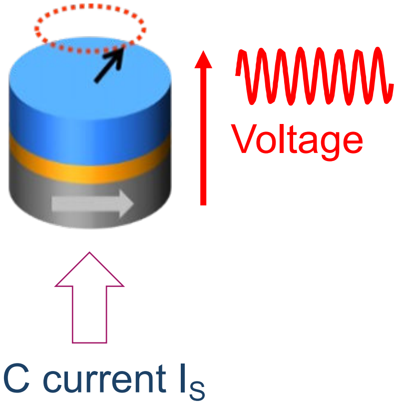
Non-linear activation function



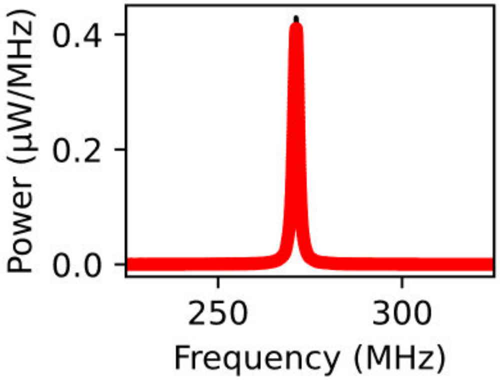
laboratoire Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

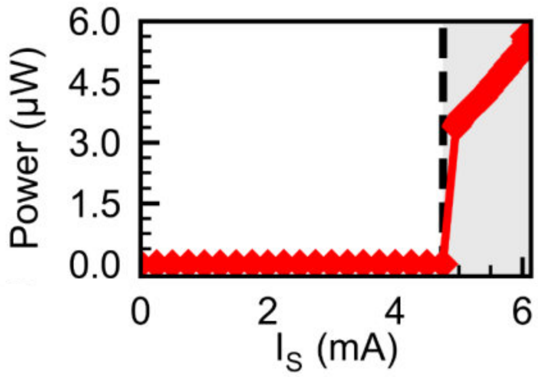
Magnetic tunnel junctions as radiofrequency neurons



RF voltage emitted



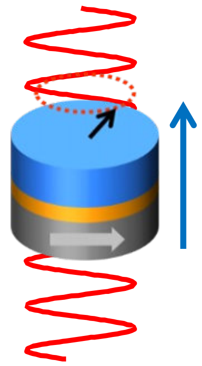
Non-linear activation function



Above a threshold current: auto-oscillations
Non-linear activation function => Neuron

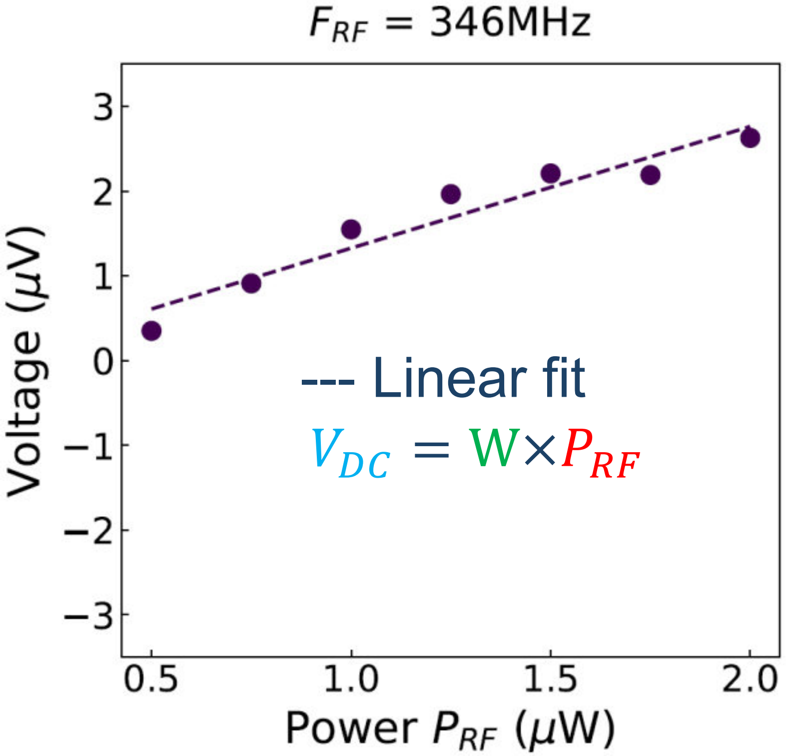
Torrejon et al. "Neuromorphic computing with nanoscale spintronic oscillators." *Nature* (2017)

MTJ as RF synapse



Resonance
V_{DC}("Spin diode effect")

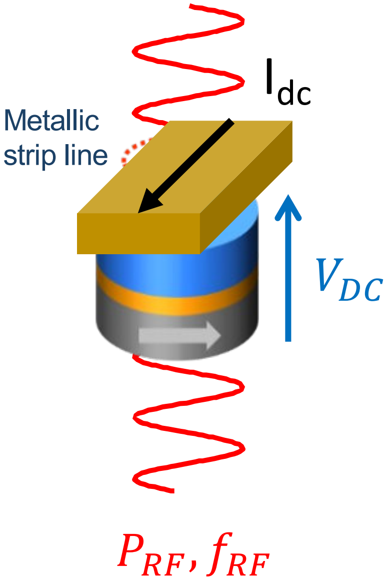
P_{RF}, f_{RF}



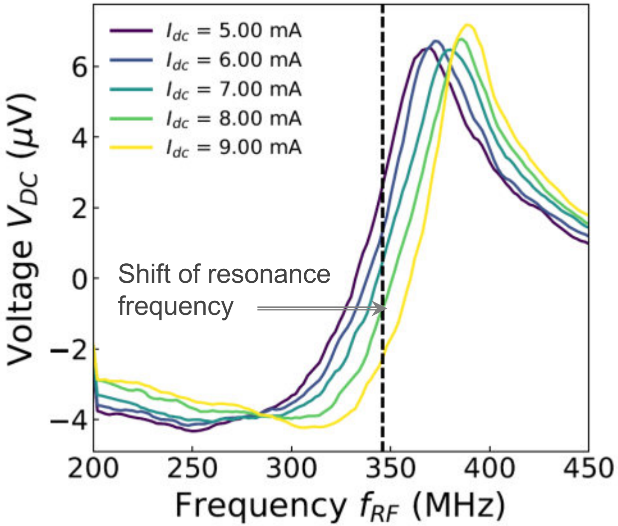
Synaptic multiplication :
Output = W x Input
$V_{DC} = W \times P_{RF}$

Theory: Leroux, et al, *Physical Review Applied* 15, 034067 (2021)
Experiments: Leroux et al. *Neuromorph. Comput. Eng.* 1, 011001 (2021)

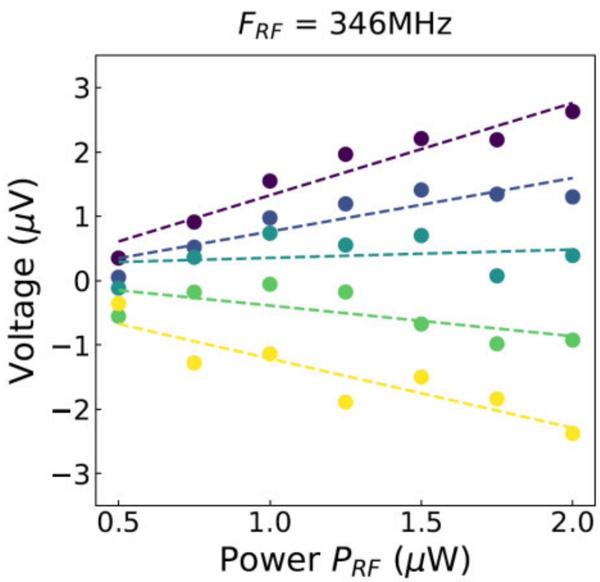
We tune the weight through the resonance frequency



Tuning the resonance frequency



Multiplication by tunable weight



$$V_{DC} = W(f_{res}) \times P_{RF}$$

laboratoire Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

How to get non-volatile synapses

Analogue non-volatile control of frequency through magnetic anisotropy by resistive switch material

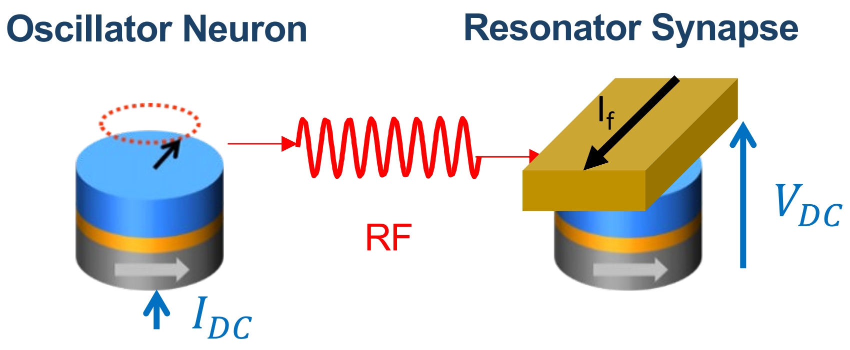
Ex: Choi *et al. Nat Commun* **13**, 3783 2022

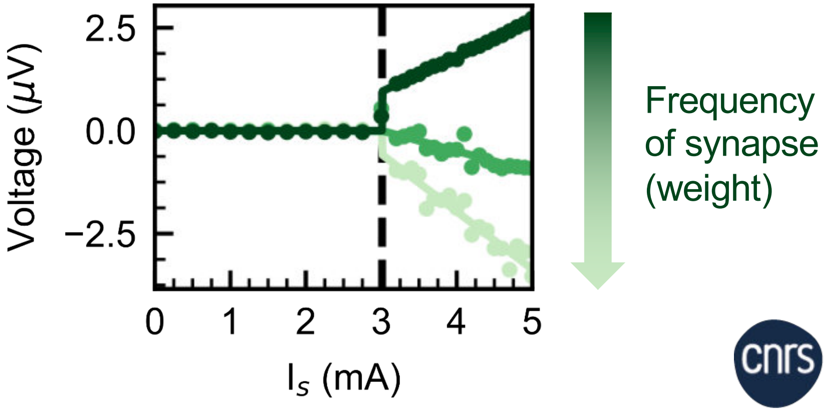
Binary non-volatile control of frequency through vortex polarity reversal

Ex: Pigeau *et al.* "Optimal control of vortex-core polarity by resonant microwave pulses." *Nature Physics 7.1 (2011)*

+ Others

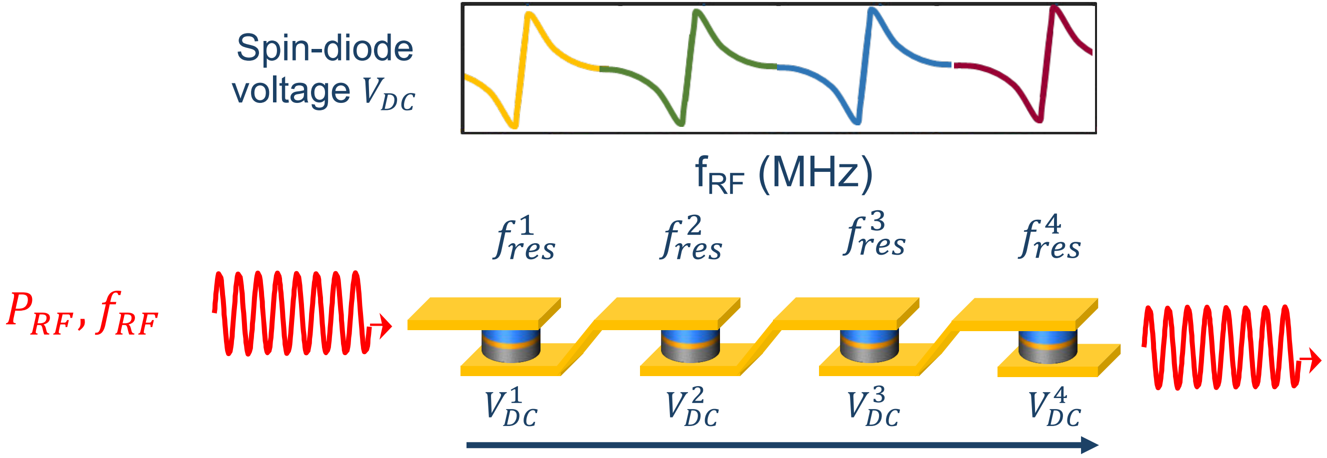
Connecting a nano-neuron to a nano-synapse





laboratoire Albert Fert

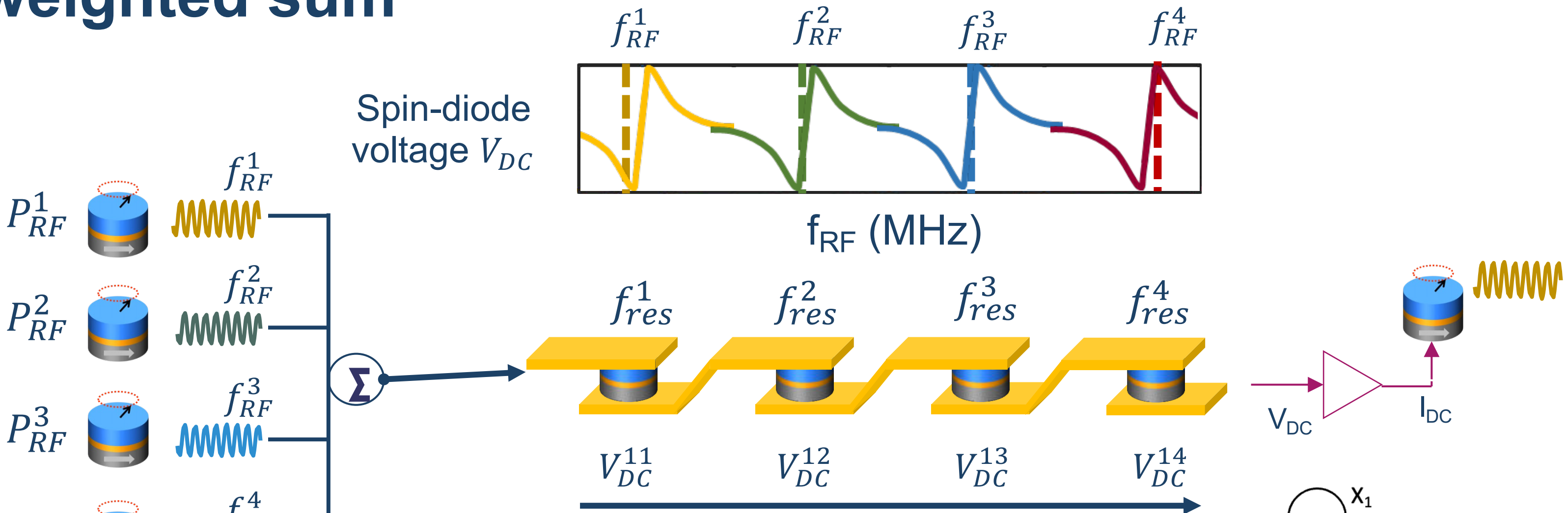
We connect several synapses of different resonance frequencies in series



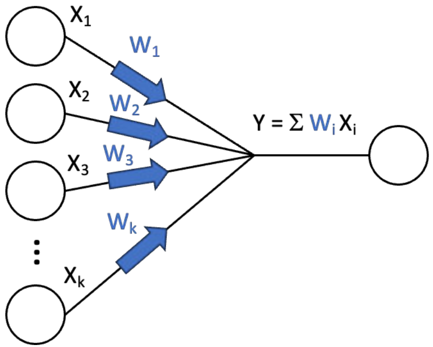
$$V_{DC} = \sum_i V^i_{DC}$$

Different frequencies achieved by diameter, shape, thickness etc.

We leverage frequency multiplexing to implement the weighted sum



$$V_{DC} = \sum_i V_{DC}^{1i} = \sum_i P_{RF}^i \, W(f_{RF}^i - f_{res}^i)$$

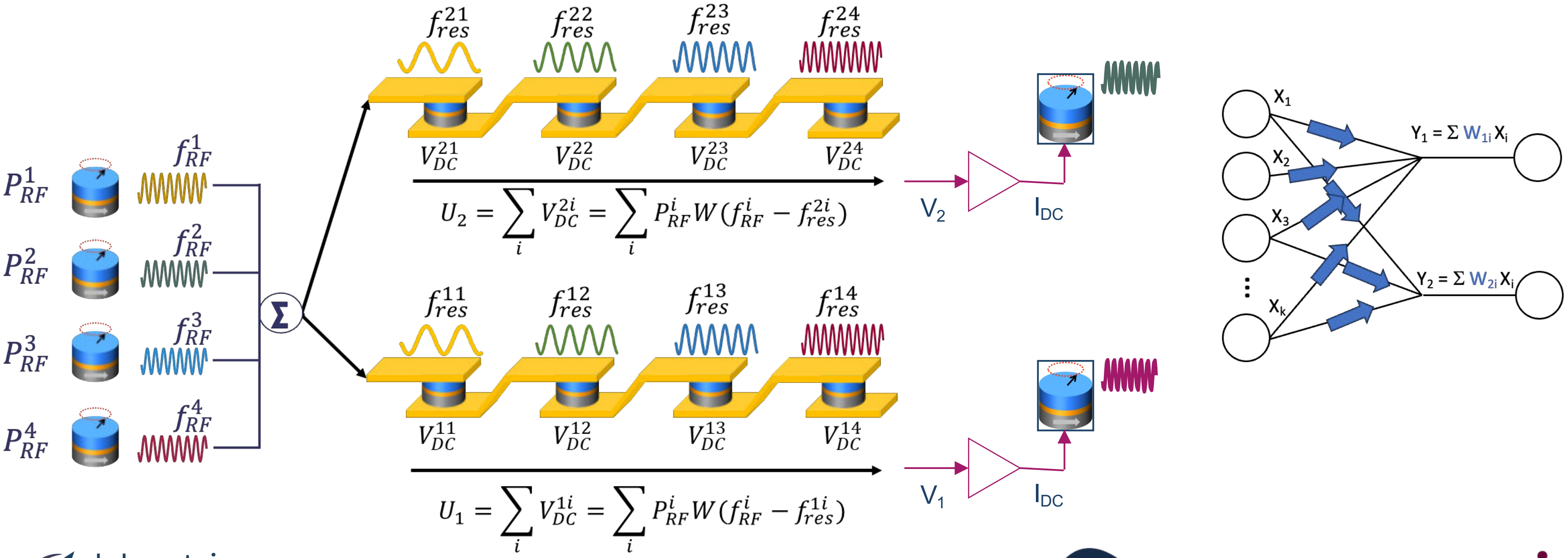


Rough frequency tuning => connectivity
Fine frequency tuning => weight tuning

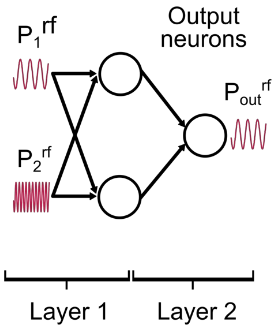
laboratoire Albert Fert

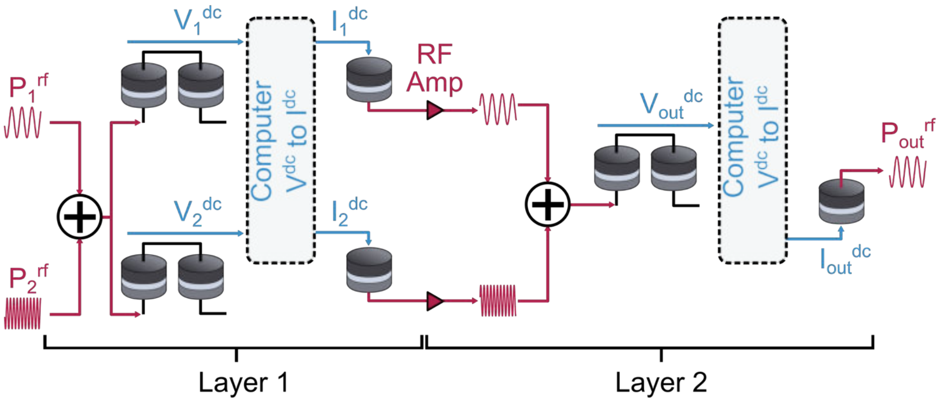
cnrs THALES Building a future we can all trust université PARIS-SACLAY

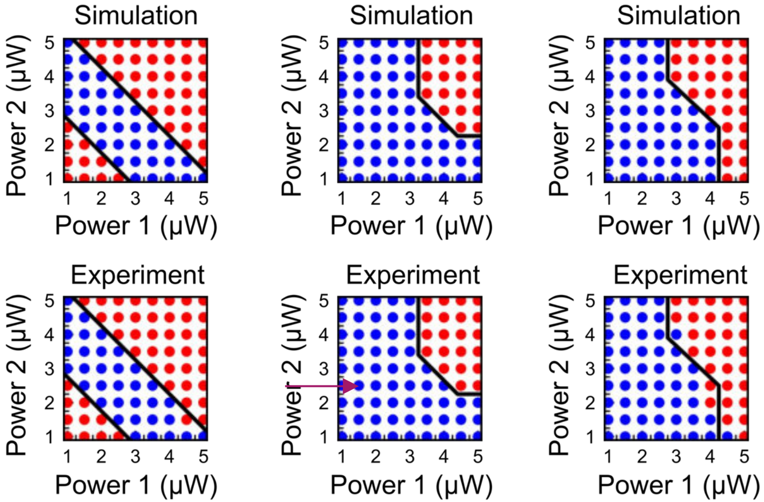
Frequency multiplexing simplifies the architecture => higher density is possible



Experimental non-linear classification







Each task: different set of weights

- $P_{out} = 0$
- $P_{out} \neq 0$

97.7% accuracy

Albert Fert

A. Ross, N. Leroux et al., *Nature Nano 2023*

cnrs
THALES Building a future we can all trust
université PARIS-SACLAY

Simulations of hardware neural networks

- **Big strength of spintronics: we have good models** (analytical, LLG, micromagnetics)

- Research in AI relies on libraries that perform training via BackPropogation.
- Computes the gradients of the loss versus the weights and updates the weights ("automatic differentiation") from your model

$$W = W + \alpha \frac{\partial L}{\partial W}$$

In my case we train the resonance frequencies:

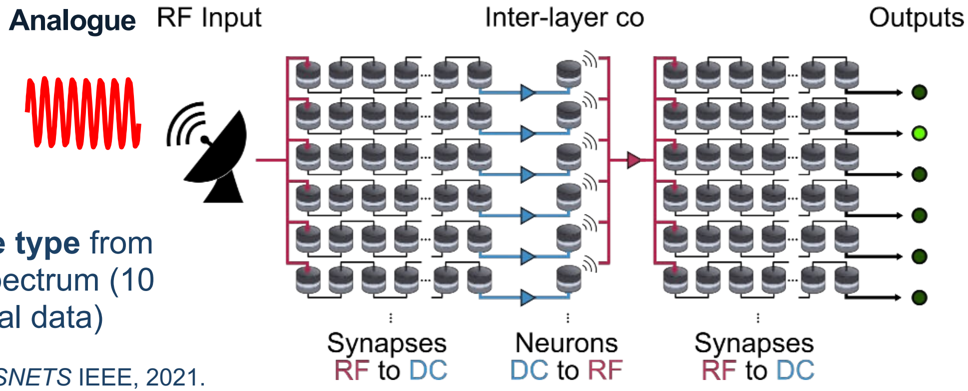
$$f = f + \alpha \frac{\partial L}{\partial f}$$

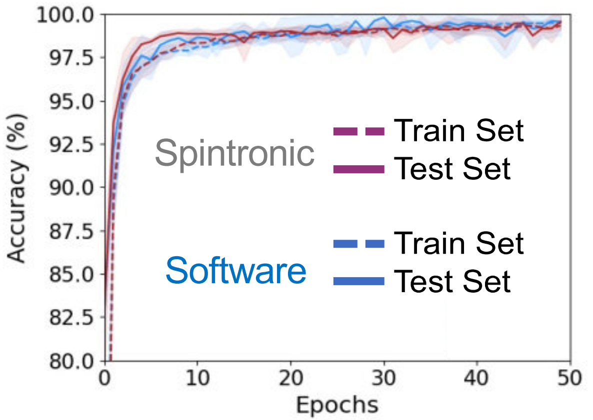
laboratoire Albert Fert

PyTorch

cnrs THALES Building a future we can all trust université PARIS-SACLAY

Performance of simulated spintronic network is as good as conventional software network

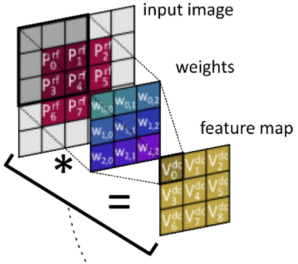




Identify drone type from its emission spectrum (10 classes, real data)

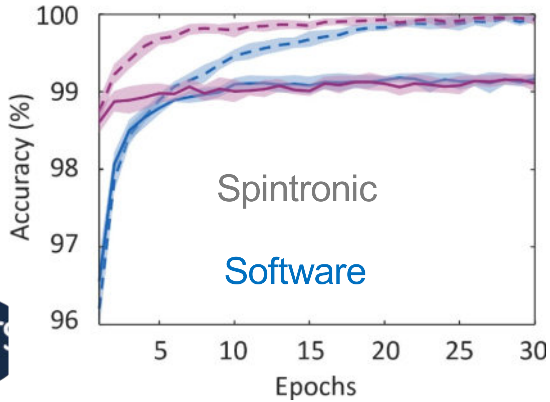
Basak, et al., *COMSNETS* IEEE, 2021.

A. Ross et al. Nature Nanotechnology (2023)



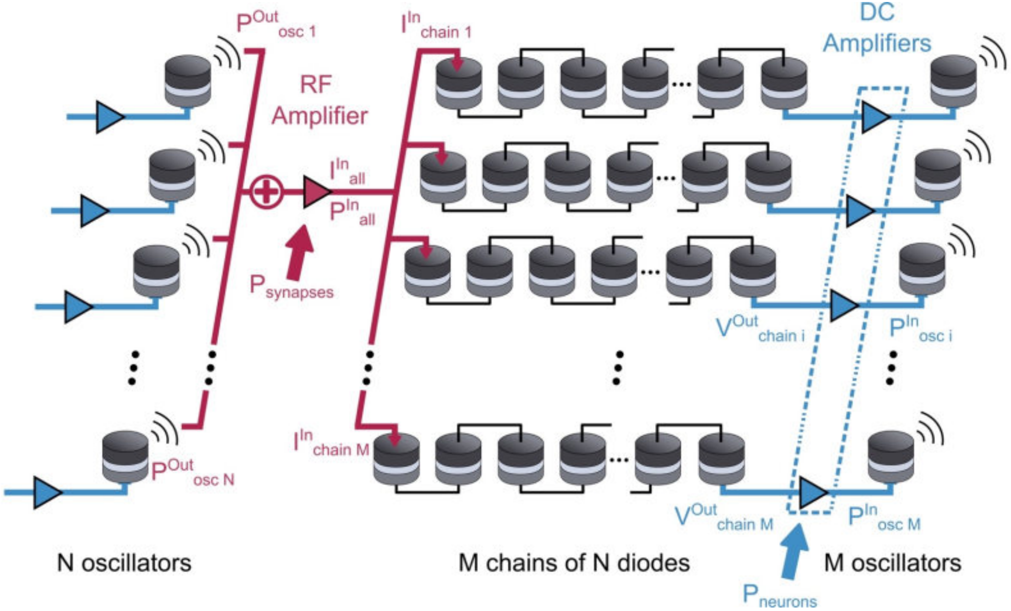
Convolutional neural network on MNIST

Leroux et al. "Convolutional Neural Networks with Radio-Frequency Spintronic Nano-Devices." *Neuromorph. Comput. Eng.* 2022



laboratoire Albert Fert

10 fJ/synapse and 100 fJ/neuron for MTJs with 20 nm diameter



Consumption of the whole architecture

Comparable to energy consumption estimations for memristive or optical devices

Much lower energy consumption than CMOS technologies

MNIST on RF spintronic network:

MLP with a 128-neuron hidden layer (1,084 neurons + 238,000 synapses) => 1 nJ.

CNN (8576 neurons + 6.7 million synapses) => 68 nJ.

Platform	Type	Technology	Coding	Input resolution	Network size/structure	Data augmentation/regularization	Energy per classification	Classifications per second[a]	Test accuracy (%)	Ref. (year)
Nvidia Tesla P100	Digital	14 nm	ANN	28 × 28	CNN[b]	Dropout	852 µJ	125,000	99.2	Supplementary Section SI.E.2
SpiNNaker	Digital	130 nm	Rate	28 × 28	784-600-500-10	Noisy input encoding	3.3 mJ	91	95.0	82 (2015)
True North	Digital	28 nm	Rate	28 × 28	CNN	Noisy input encoding	0.27 µJ	1,000	92.7	61 (2015)
True North	Digital	28 nm	Rate	28 × 28	CNN	Noisy input encoding	108 µJ	1,000	99.4	61 (2015)
Loihi	Digital	14 nm	Bin. rate	(20 × 20)[c]	400-400-10	Not available	2.5 µJ	5,917	96.2	83 (2021)
Unnamed (Intel)	Digital	10 nm	Temporal	(28 × 28)[d]	236-20	Stochastic spike loss	1.0 µJ	6,250	88.0	84 (2018)
BrainScaleS-2	Mixed	65 nm	Temporal	16 × 16	256-246-10	Input noise	8.4 µJ	20,800	96.9	This work; Supplementary Section SI.E.1

Göltz, *et al. Nat Mach Intell* **3**, 823–835 (2021)

RF drones task:

3.4 mW for MLP

USRP: 45 W
GHz ADC: several mW

Albert Fert

THALES Building a future we can all trust

université PARIS-SACLAY

Questions?

laboratoire
Albert Fert

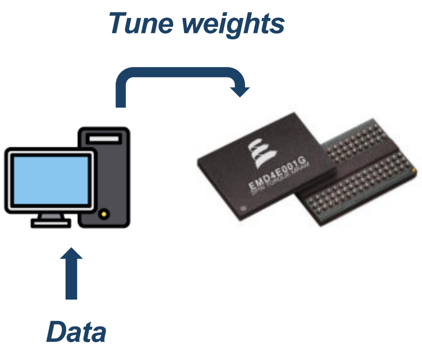
The problem of training

laboratoire
Albert Fert

How do you train your neural network?

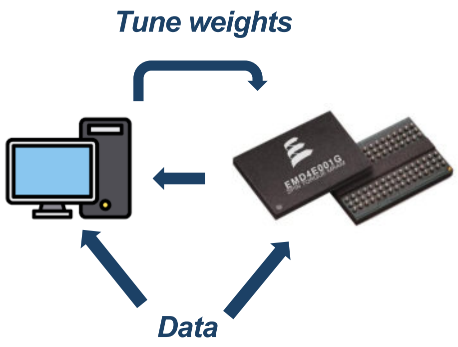
Off-chip training

1. Train on computer
2. Configure chip accordingly



Chip-in-the-loop training

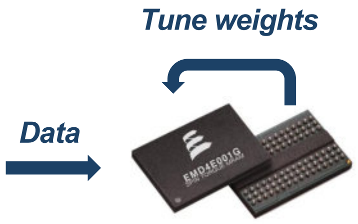
Computer is used
to train the chip



Chip can be only used for inference
Or has to communicate with server to learn new things

Self-learning

Chip trains itself
No computer needed



Chip can learn new things autonomously

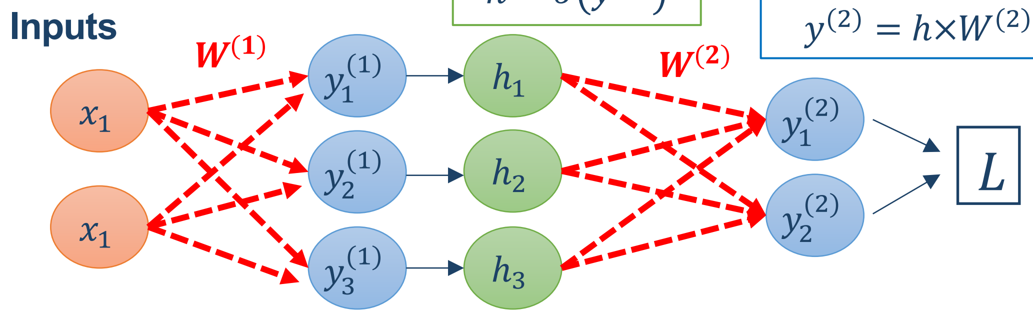
BackProp is non-local

Forward

$$y^{(1)} = x_1 \times W^{(1)} + b^{(1)}$$

$$h = \sigma\left(y^{(1)}\right)$$

$$y^{(2)} = h \times W^{(2)} + b^{(2)}$$



$$W = W + \alpha \frac{\partial L}{\partial W}$$

$$\frac{\partial L}{\partial W_{i,j}^{(2)}} = \frac{\partial y_j^{(2)}}{\partial W_{i,j}^{(2)}} \times \frac{\partial L}{\partial y_j^{(2)}} = \frac{\partial L}{\partial y_j^{(2)}} \times h_i$$

$$\frac{\partial L}{\partial W_{i,j}^{(1)}} = \sum_k \frac{\partial y_j^{(1)}}{\partial W_{i,j}^{(1)}} \times \frac{\partial h_j}{\partial y_j^{(1)}} \times \frac{\partial y_k^{(2)}}{\partial h_j} \times \frac{\partial L}{\partial y_k^{(2)}} = \sum_k \frac{\partial L}{\partial y_k^{(2)}} \times W_{j,k}^{(2)} \times \sigma'\left(y_j^{(1)}\right) \times x_i$$

Backward

Backpropagation vs. neuro/physics inspired rules

Backpropagation is state of the art performance on actual tasks

But… Not hardware friendly (non-local, very small weights updates etc.)

Neuro/physics rules are hardware friendly:
Local (synapse modified only by neurons around)
Self-learning by the physics of system

But…. Performance on hard tasks is low because they do not minimize the global error

Can we merge the two to get the advantages of both?

Does the brain perform some kind of backpropagation?

Hot topic in AI and computational neuroscience + critical for us

Lillicrap, T.P., Santoro, A., Marris, L. *et al.* Backpropagation and the brain. *Nat Rev Neurosci* **21,** 335–346 (2020)

Video of Hinton « Stanford Seminar - Can the brain do back-propagation? »
https://www.youtube.com/watch?v=VIRCybGgHts

Questions to have in mind when doing research in neuromorphic computing

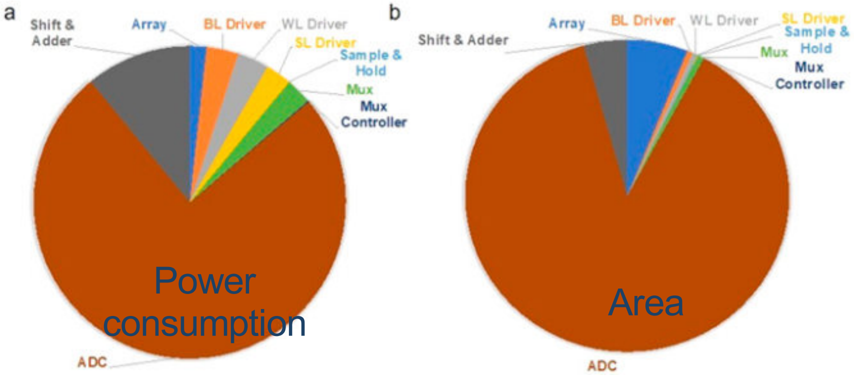
laboratoire
Albert Fert

cnrs THALES Building a future we can all trust université PARIS-SACLAY

What are the inputs and outputs?

How do you feed inputs to your system?
How do you read the outputs? (simple circuit or giant microscope?)

Can you build deep networks?
Are your inputs/outputs compatible with CMOS integration?



Christensen *et al* 2022 *Neuromorph. Comput. Eng.* **2** 022501

laboratoire Albert Fert

THALES Building a future we can all trust

université PARIS-SACLAY

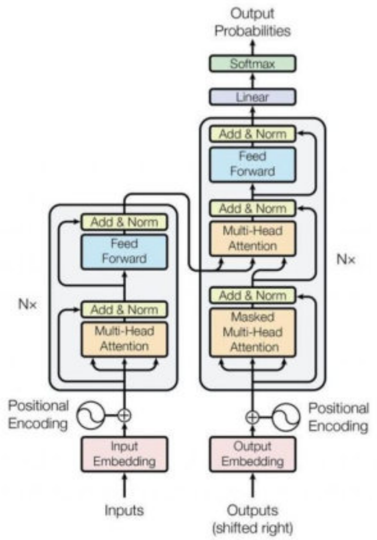
Is my system scalable to complex tasks?

Scalability

Can I stack layers into deep networks?
Can I achieve dense connectivity?
Can I have high fan in and fan out?
What will the total architecture look like? CMOS circuits etc…

Trainability

Are my connections tunable?
Can the weights be non-volatile?
How will I perform the learning? (Rule? Weights update?)



Transformer architecture

Is my system robust to reality?

How does my performance is affected by:

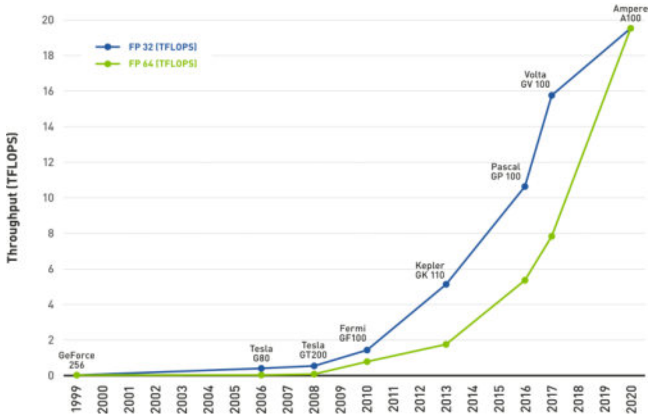
- Errors in programming
- Non-linearities
- Variability
- Noise
- Change of temperature

Etc.

Is my system competitive with other technologies?

<u>Important metrics:</u>

- Speed
- Energy consumption
- Power consumption
- Compactness



W. J. Dally, et al., "Evolution of the Graphics Processing Unit (GPU)," in *IEEE Micro*, 2021

Tops/W is much used metrics but can be computed in many ways. Use with caution…

Digital CMOS accelerators are really good now!

Inference-only neuromorphic systems may be not enough to compete except for niche application (ex: quantum hardware, specific sensor etc.) => learning

Conclusions

Neuromorphic computing is large and diverse field
=> need for interdisciplinary mindset

Challenges are how to build scalable deep dense networks, that can learn by themselves

Materials
Devices
Circuits
Architectures
Algorithms

Need to be co-designed

Many exciting research opportunities for you!

We are hiring postdocs and students

Reviews:
2022 roadmap on neuromorphic computing and engineering, DV Christensen *et al., Neuromorph. Comput. Eng.* **2** (2022)
Physics for neuromorphic computing, Marković, D., Mizrahi, A., Querlioz, D. *et al*. *Nat Rev Phys* **2,** 499–510 (2020)
Neuromorphic spintronics , Grollier, J., Querlioz, D., Camsari, K.Y. *et al*. *Nat Electron* **3,** 360–370 (2020)
Neurotech series of tutorials: https://neurotechai.eu/educational/