

On-Device Machine Learning with Memristors in the Neuromorphic Era

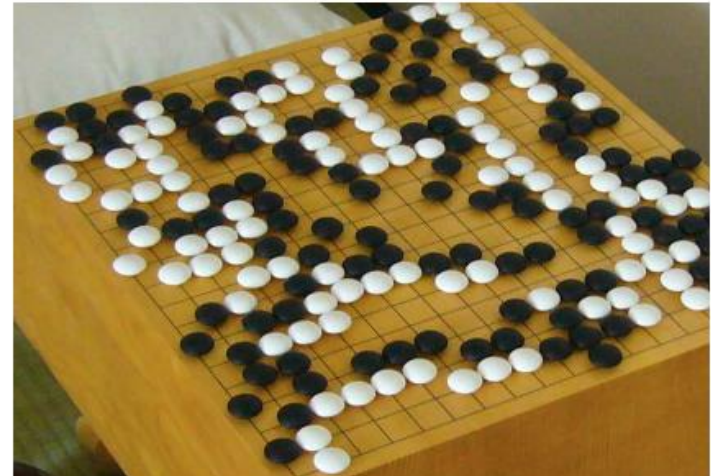
Shahar Kvatinsky

Viterbi Faculty of Electrical and Computer Engineering
Technion – Israel Institute of Technology

June 2024



AlphaGo Example



AlphaGo Example

CNN BUSINESS

Markets Tech Media Success Video

Innovate

Computer scores big win against humans in ancient game of Go

by Jethro Mullen @CNNTech

🕒 January 28, 2016: 1:09 AM ET



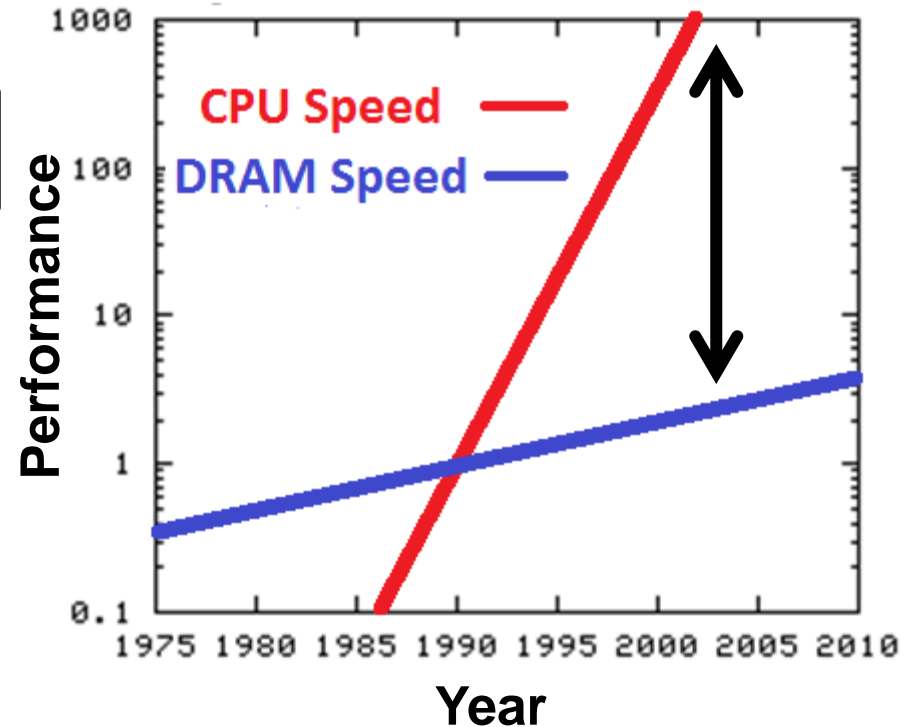
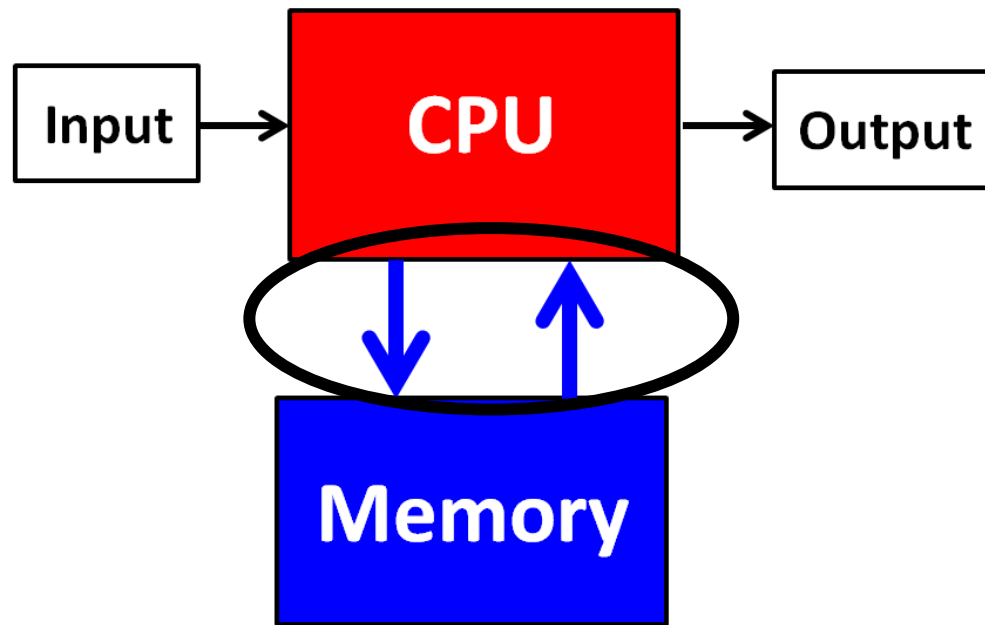
AlphaGo Example

- 1920 CPUs and 280 GPUs
- \$3000 electric bill per game



The External Memory Wall Problem

von Neumann (Architecture) Bottleneck



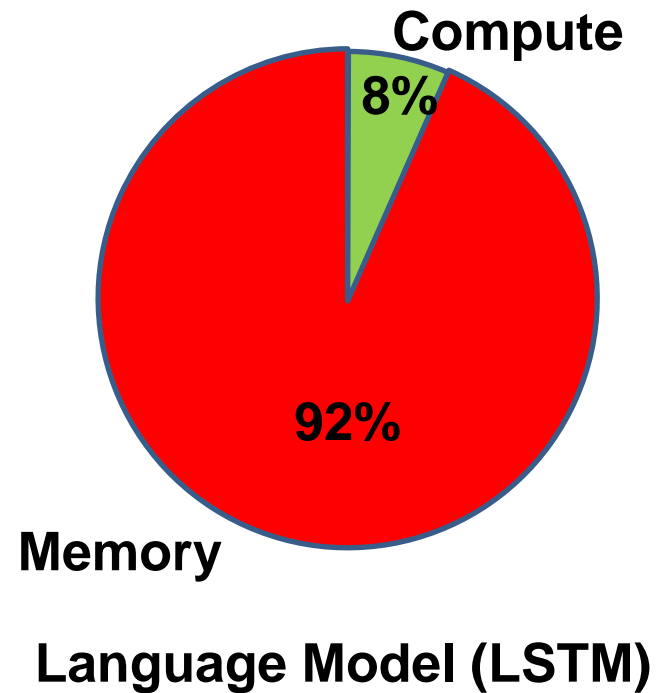
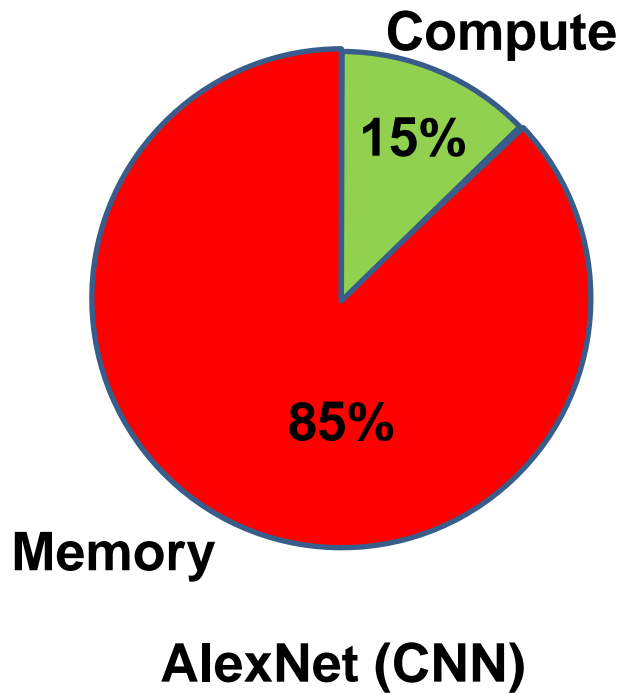
A bottleneck of both speed and power!

And a Huge Energy Bottleneck

Operation (16-bit operand)	Energy/Op (45 nm)	Cost (vs. Add)
Add operation	0.18 pJ	1X
Load from on-chip SRAM	11 pJ	61X
Send to off-chip DRAM	640 pJ	3,556X

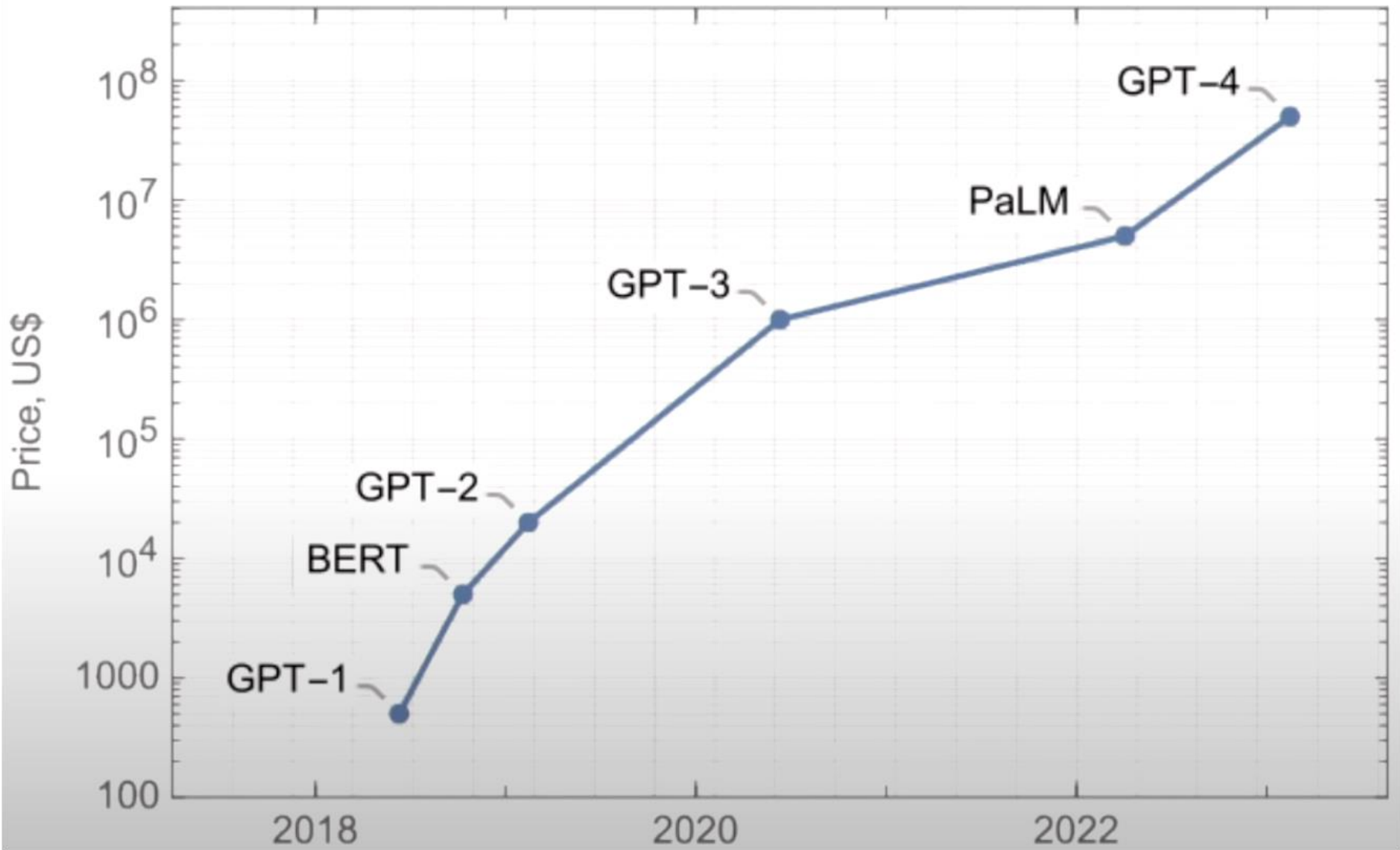
>1000X more energy to go to memory

Energy Consumption Breakdown



And Expensive

LLM training prices (at the time of their creation)



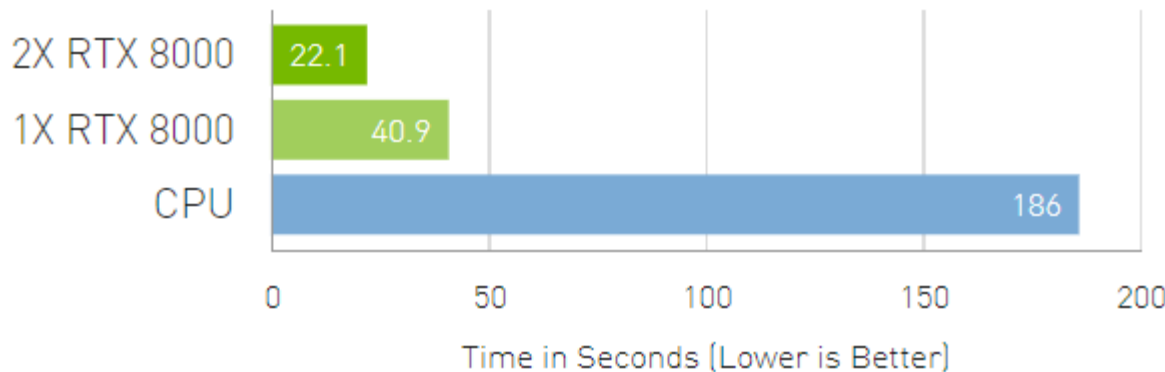
Immediate Solutions

Use Commodity Hardware

- GPU (Graphic Processing Unit) for training
- Software platforms
- ASIC/FPGA for inference



Training



*
(TLAB)
)



Google
(C++, Python)



Facebook / NYU
(C, C++, Lua)

U. Montreal
(Python)

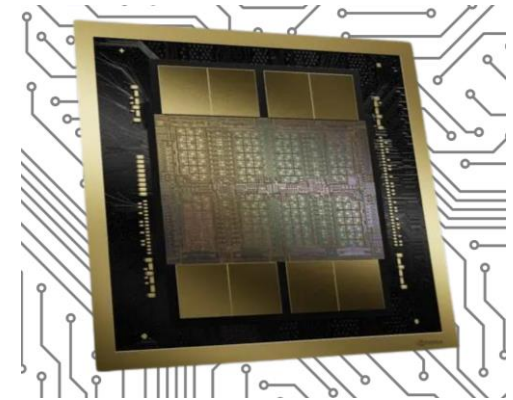
Approach 1: Improve GPUs

- Increasing GPUs, improve their software, add AI units
- Build large GPU-based systems

$$D = \begin{pmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} \end{pmatrix} \begin{pmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{pmatrix} + \begin{pmatrix} C_{0,0} & C_{0,1} & C_{0,2} & C_{0,3} \\ C_{1,0} & C_{1,1} & C_{1,2} & C_{1,3} \\ C_{2,0} & C_{2,1} & C_{2,2} & C_{2,3} \\ C_{3,0} & C_{3,1} & C_{3,2} & C_{3,3} \end{pmatrix}$$

FP16 or FP32 FP16 FP16 or FP32

Tensor core
(now also FP4, 6, 8)



NVIDIA Blackwell

HBM3e

80B Transistors

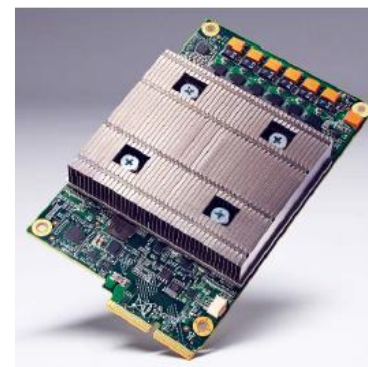
High power (300W)

Tensor core



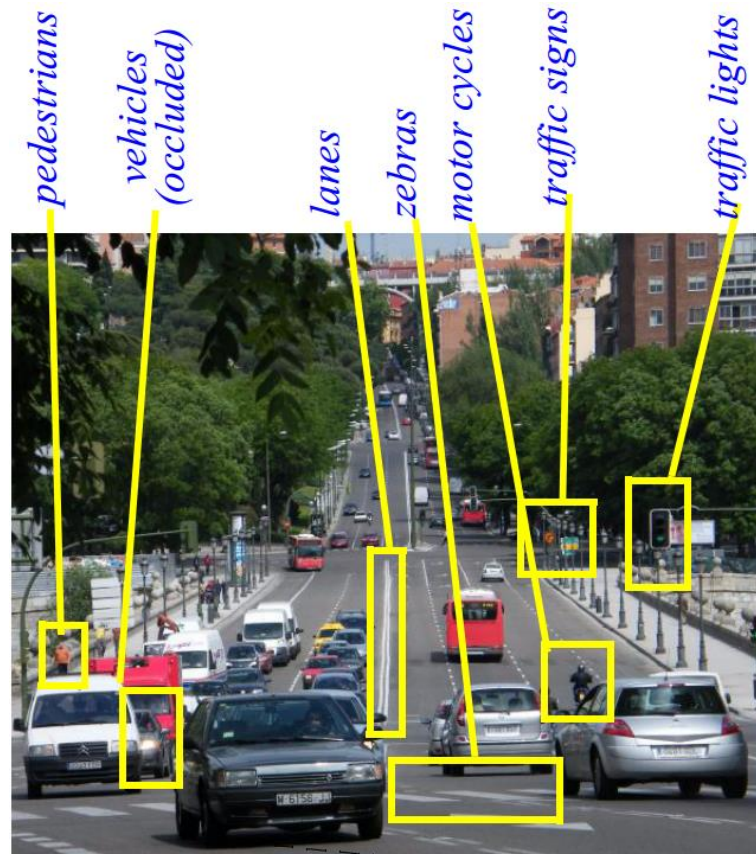
Approach 2: AI Dedicated Hardware

- Google TPU (Tensor Processing Unit)
 - Systolic arrays
 - Large on-chip memory
 - Massive parallel processing units
 - 8-bit for inference (32-bit training in advanced versions), bfloat16
 - Optical communication
- ASIC edge inference devices



Motivation

The Brain is Good at Perception Tasks



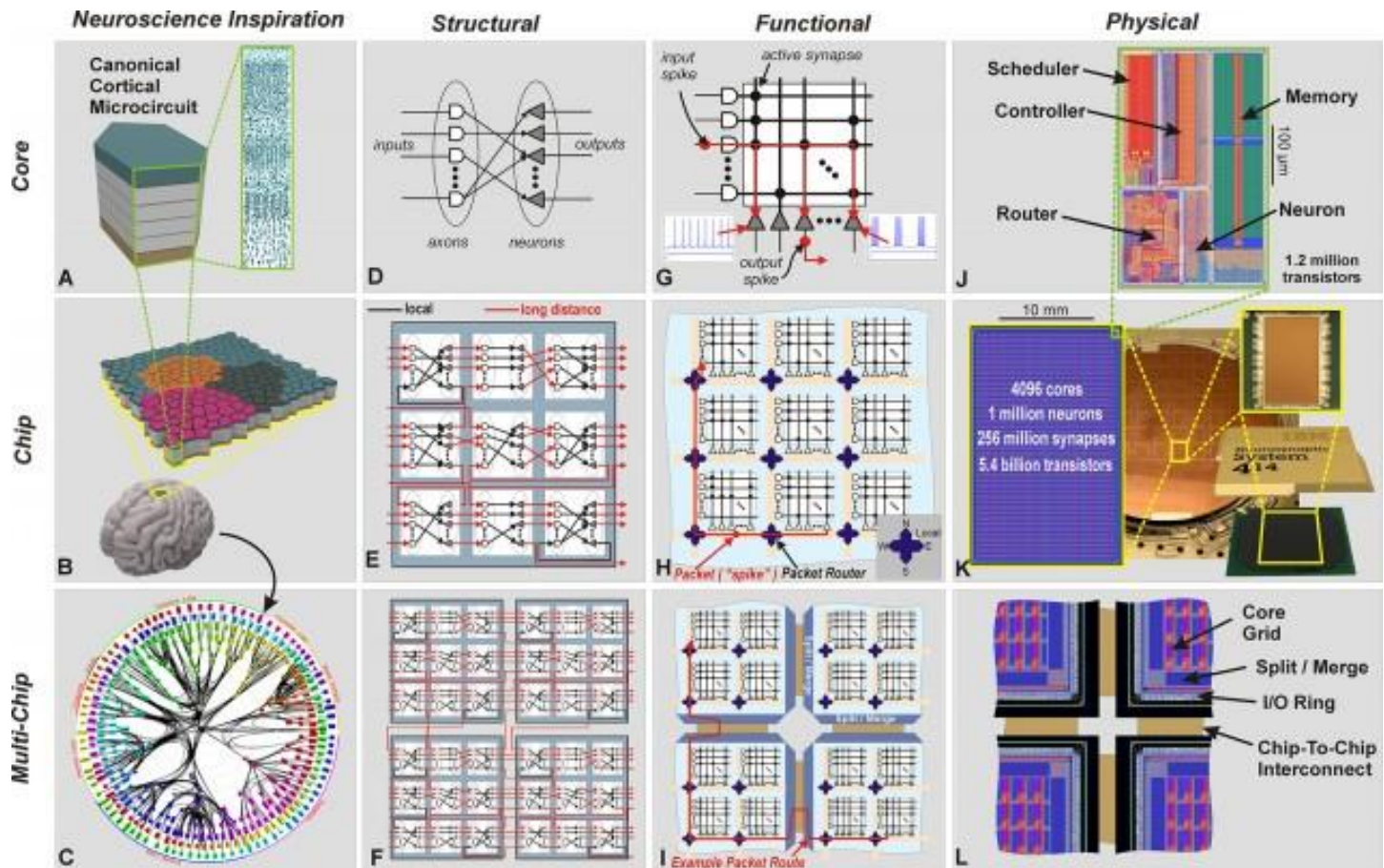
- Brain power consumption = 20W
- GPU power consumption = 300-500W
- Power to train GPT4 = 7.5 MegaW for 100 days!

Agenda

- The limitations of modern AI hardware
- **Neuromorphic computing with memristors**
- Training memristive neuromorphic systems
- Low-power memristive neuromorphic systems
- Security and summary

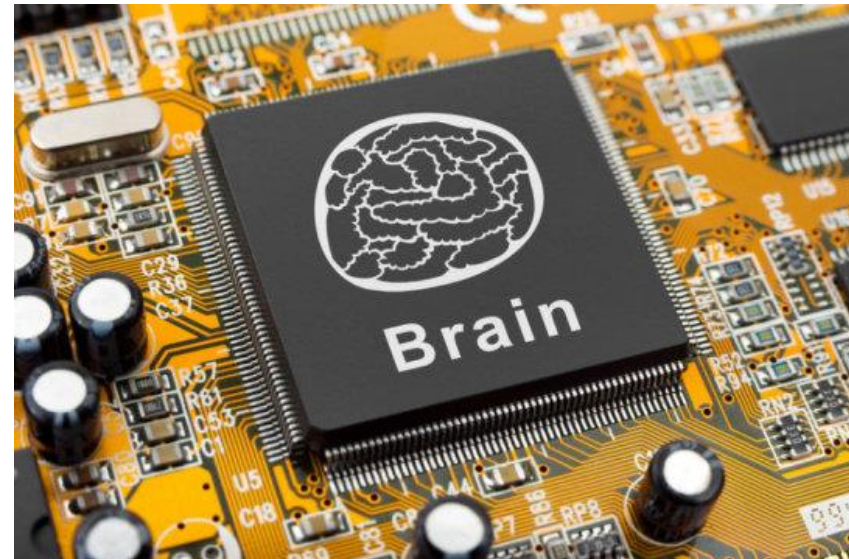
The Potential of Neuromorphic Computing

- Originally proposed by Carver Mead, late 1980s
- VLSI systems (usually analog) to mimic the nervous systems (neural networks)

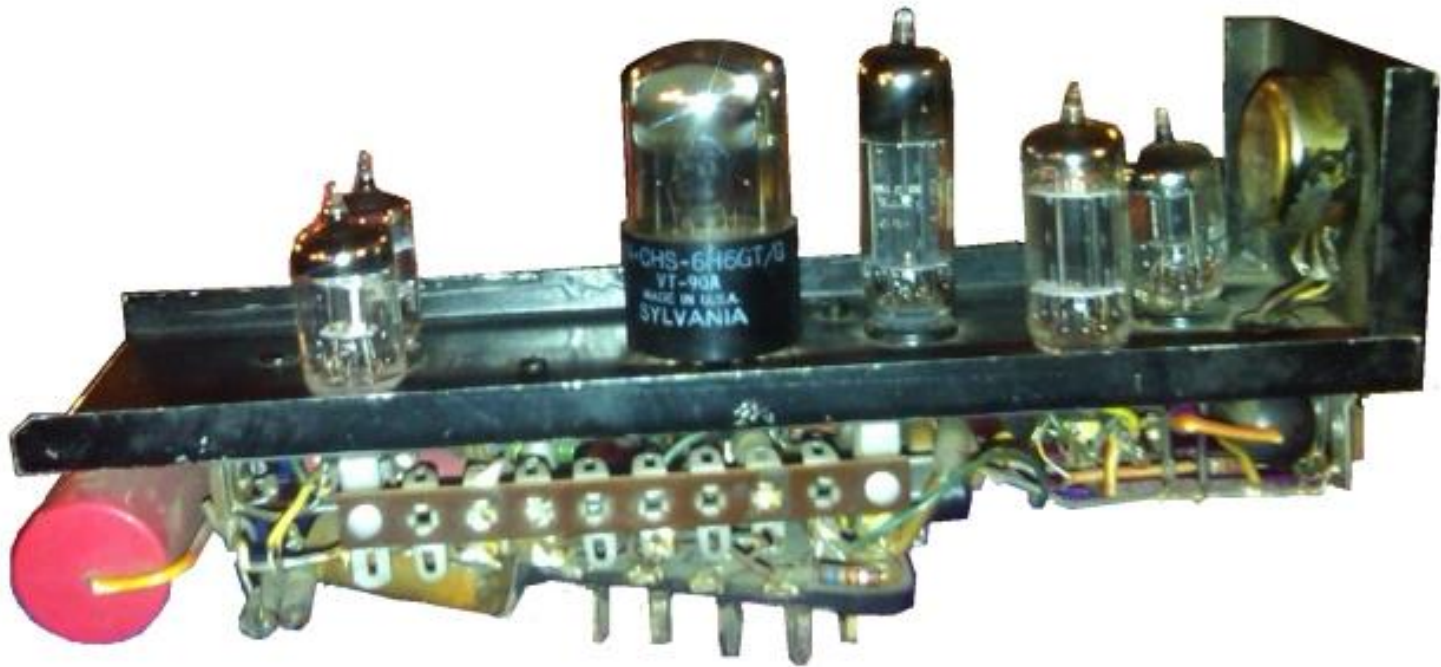


What is Neuromorphic Computing?

- Software?
- Digital hardware? (what about FPGA or GPU?)
- Analog hardware?
- System?



SNARC



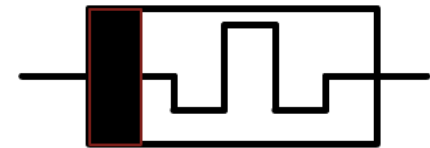
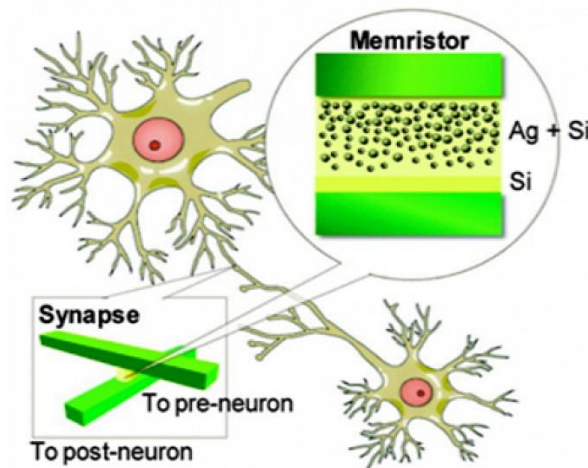
Stochastic Neural Analog Reinforcement Calculator

Marvin Minsky, 1951

40 neurons

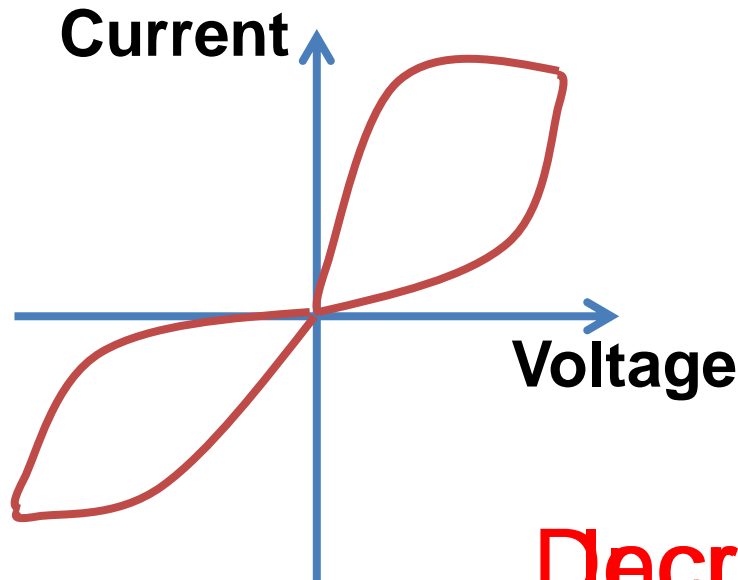
Neuromorphic with Emerging Technologies

- Use new technologies (not transistors) to build more efficient neuromorphic systems
- Emerging nonvolatile technologies (memristors, RRAM, PCM, even Flash) behave like synapses
- Moving to analog computing? No more 0s and 1s?



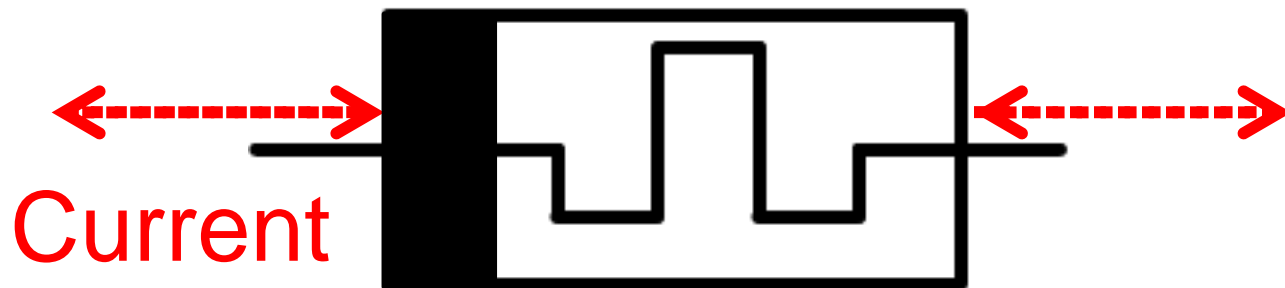
Memristor – Memory Resistor

Resistor with Varying Resistance



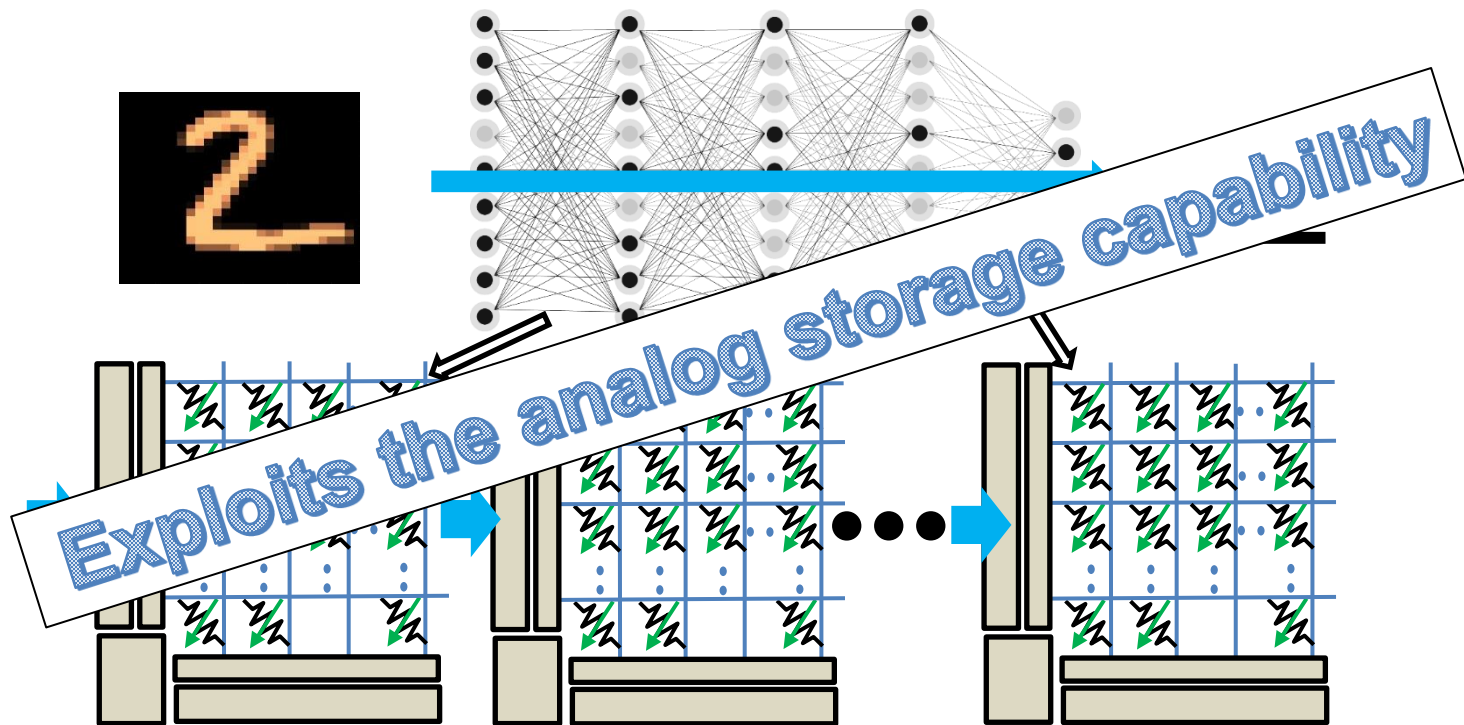
High resistive state
(R_{OFF} , LRS)

Decrease resistance



Memristor

Deep Learning: Inference



- The trained synaptic weights are mapped to memristive arrays
- Memristive memory cores perform matrix-vector multiply operations

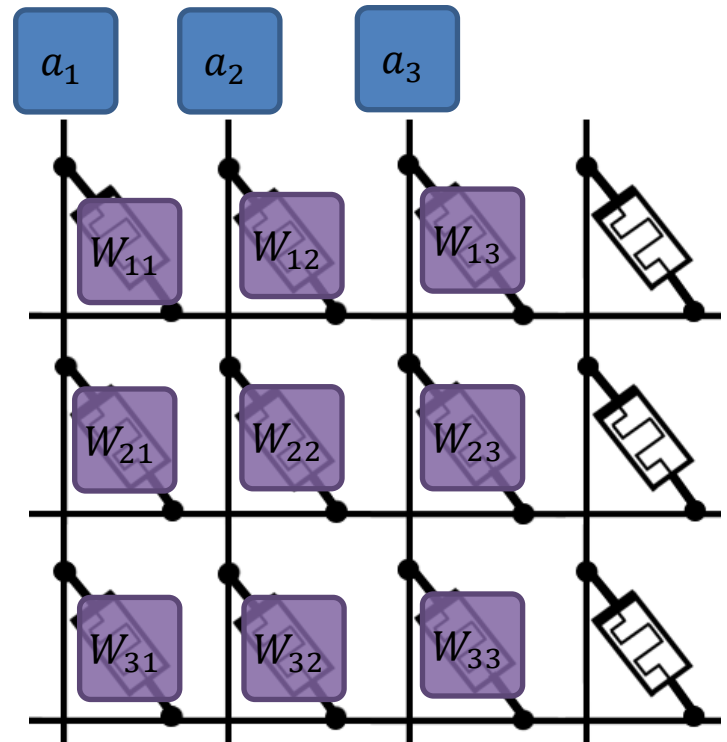
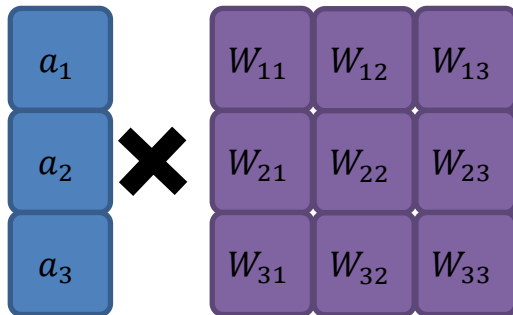
Mapping the Weights

- The crossbar is used to store the weights data

The conductance = the weight value

$$G_{ij} = W_{ij}$$

- The activation is the voltage drop over the columns (rows)



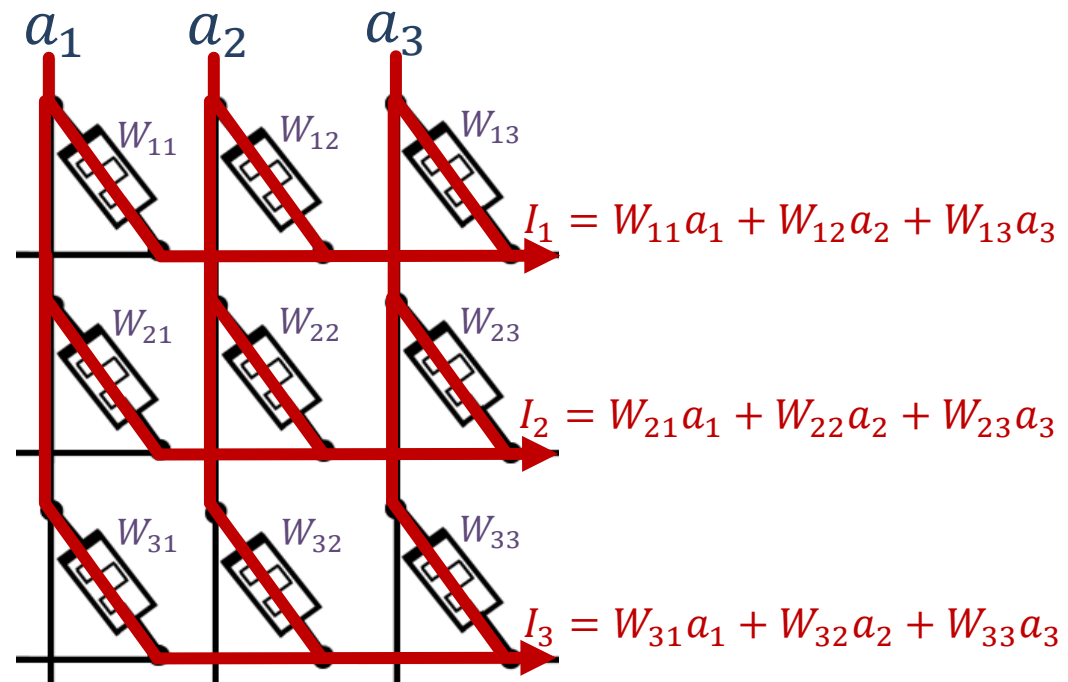
Multiply and Accumulate (Feed Forward)

- Using KCL

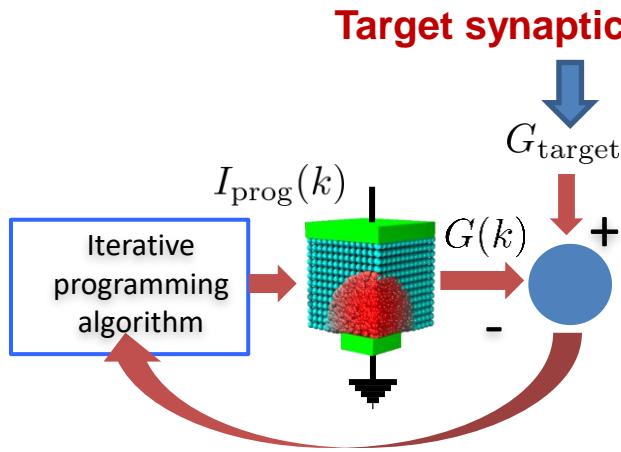
The output current is the weighted sum (inner-product)

$$I_j = \sum_{i=1}^N W_{ij} a_i$$

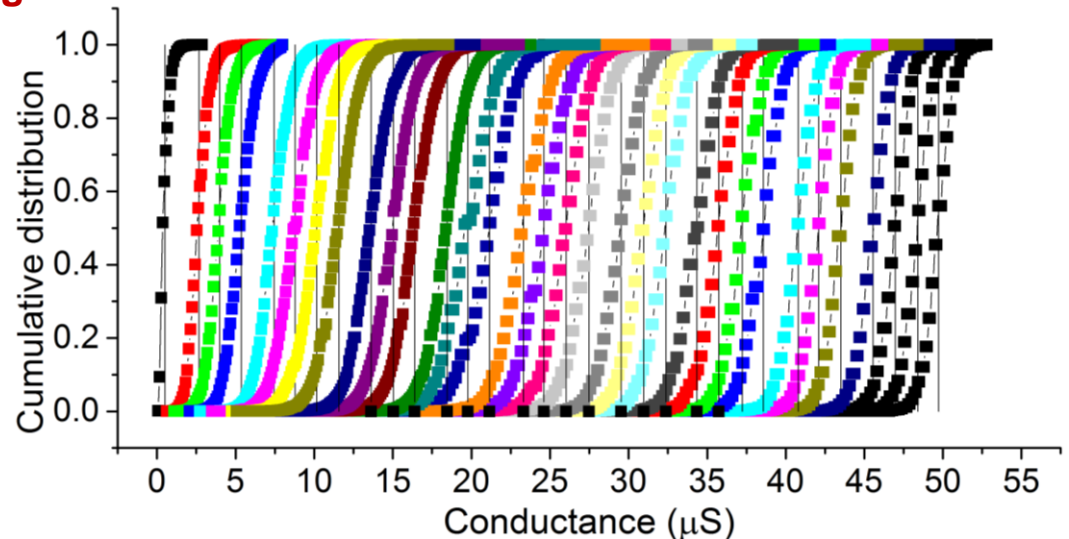
- Parallel operation to all rows is the crossbar



Mapping Synaptic Weights to PCM Devices



Measurements based on >10k devices
32 representative states

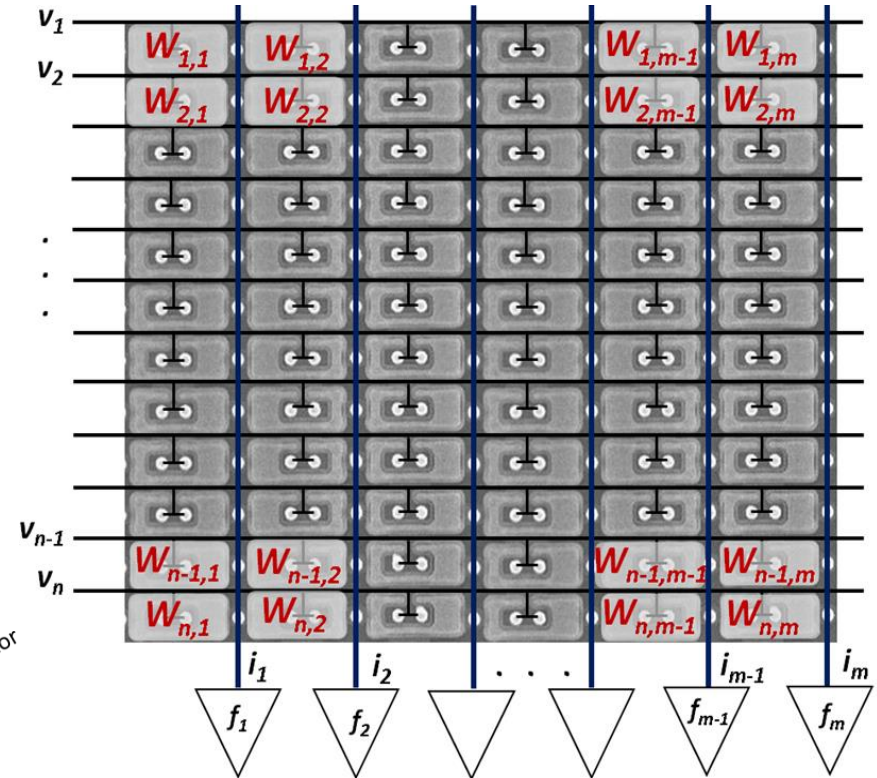
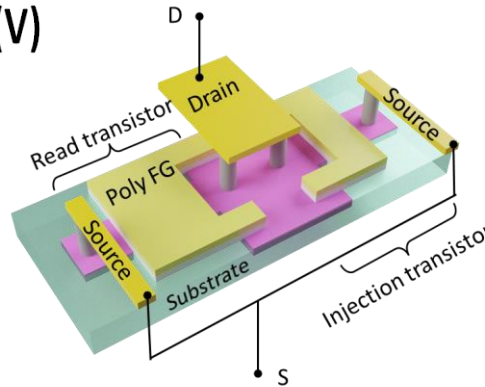
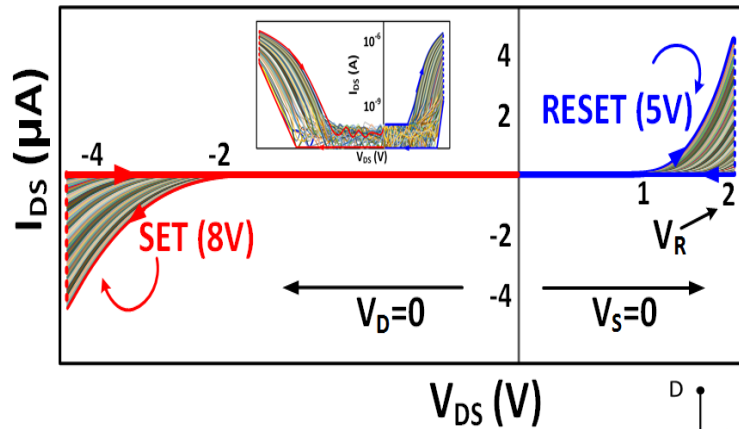


- **Iterative programming algorithms** used to achieve a target conductance value
- Non-ideal analog storage \rightarrow Distribution of conductance values

Papandreou et al., ISCAS (2011)
Sebastian et al., E/PCOS (2016)

Y-Flash Memristors for Neuromorphic

(Danial et al., Nature Electronics 2019, DATE 2020,
Wang et al. APL 2021)

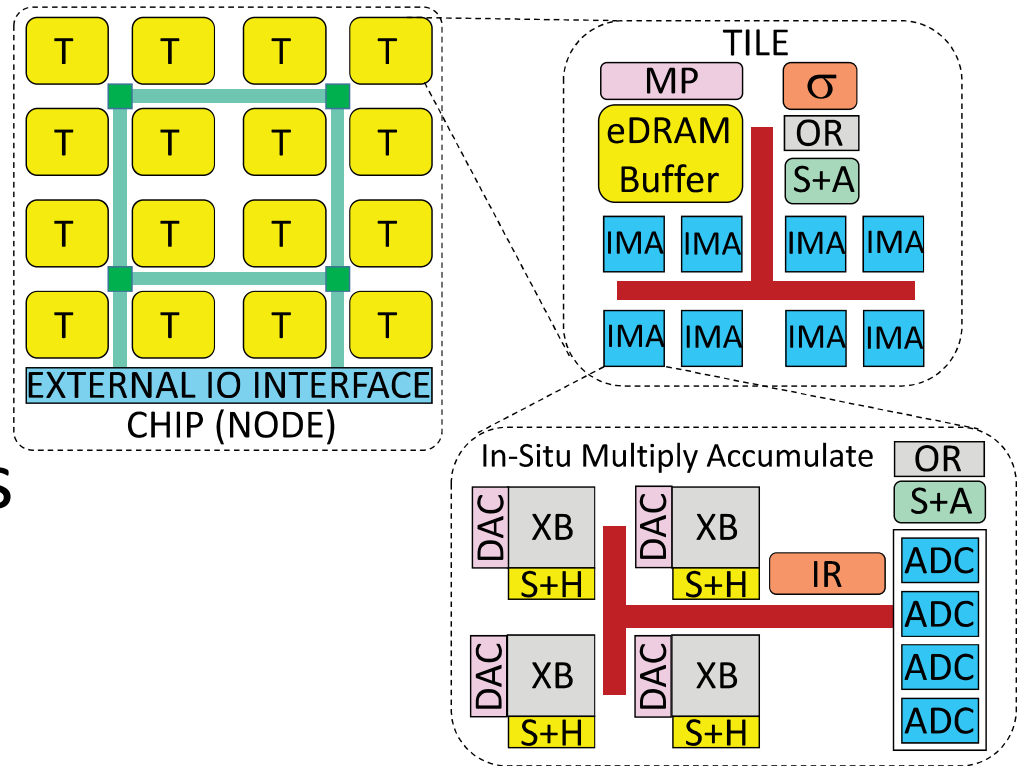


- Two-terminal floating-gate memristor
- Number of distinct resistive levels (weights) > 65
- Standard single-poly CMOS 180nm
- Mature technology for VLSI neuromorphic systems

Tower
Semiconductor

ISAAC – Inference Accelerator

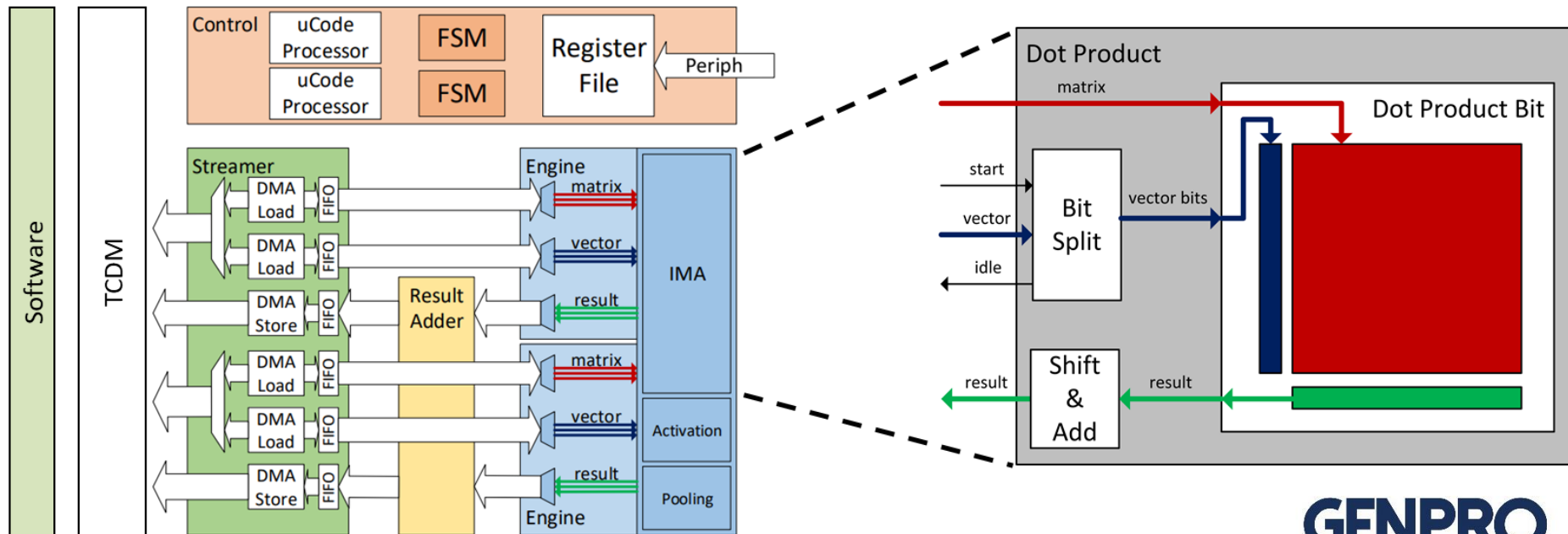
- Tile-based architecture
- In-situ multiply-accumulate (IMA) unit
 - 128×128 crossbars
 - Shared 8-bit ADC
 - Private DAC + S&H
 - Shift & Add unit
- Only activation moves



MultiPULPly (Eliahu et al., JETC 2021)

Ultra-Low-Power Neuromorphic Accelerator

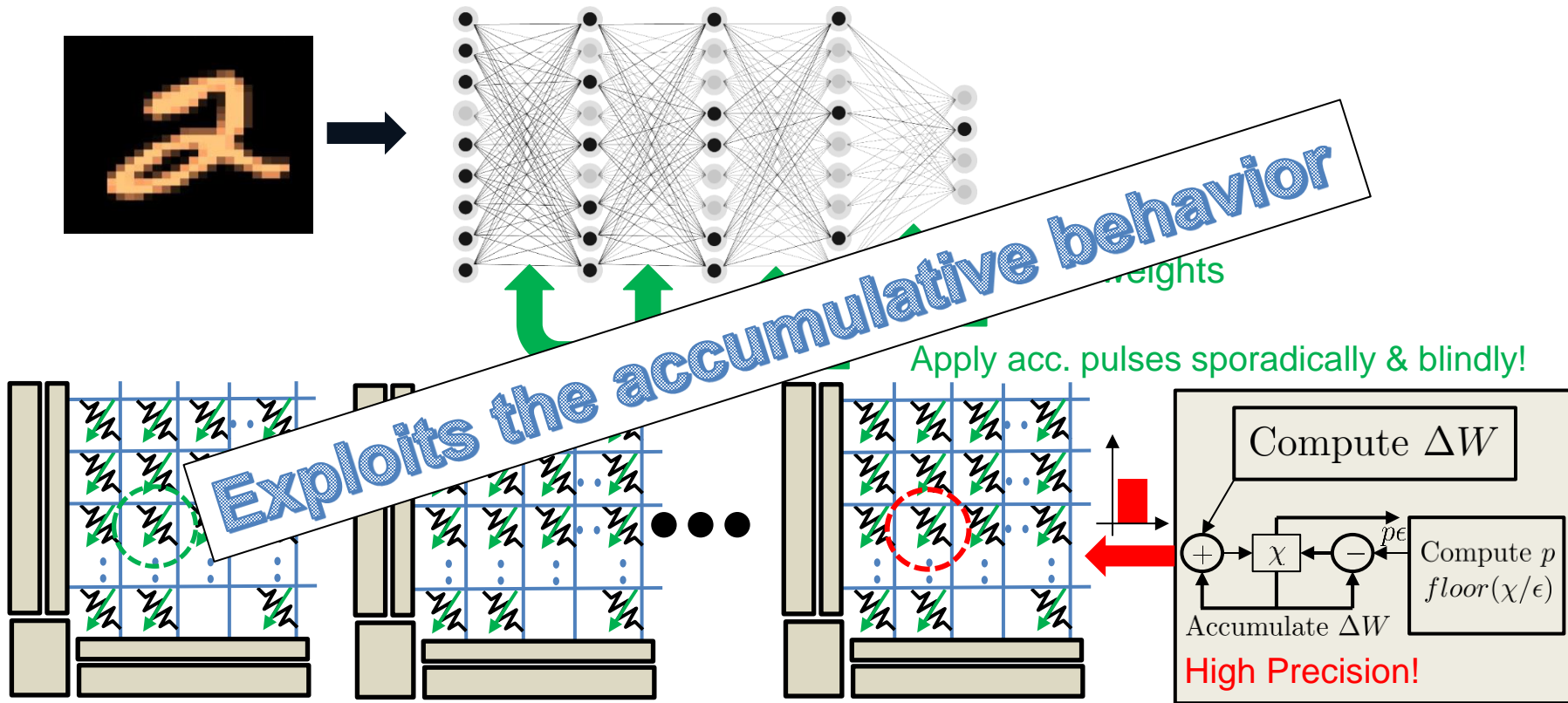
- PULPISSIMO extension (RISC-V based platform)
- 1.5-8 mW, 30-82 GOp/s, 10-20 TOPs/W (@22nm)
- 15-67 ms to run MobileNet



Agenda

- The limitations of modern AI hardware
- Neuromorphic computing with memristors
- **Training memristive neuromorphic systems**
- Low-power memristive neuromorphic systems
- Security and summary

Deep Learning Training



Nandakumar et al., ArXiv, 2017

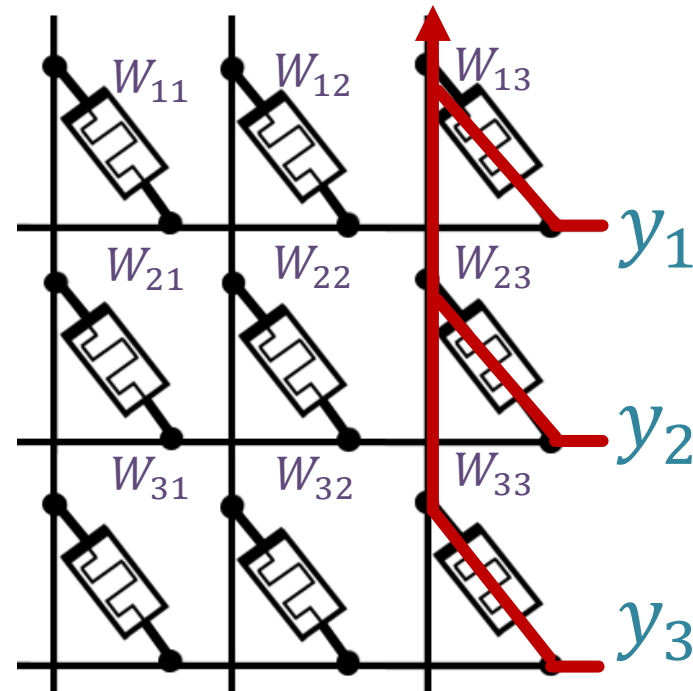
Sebastian et al., VLSI, 2019

Backpropagation

The error δ_l is propagate according to:

$$y_l \triangleq \underbrace{(W_{l+1}^T y_{l+1})}_{\delta_l} \times \sigma'(z_l)$$

$$\delta_3 = W_{13} y_1 + W_{23} y_2 + W_{33} y_3$$



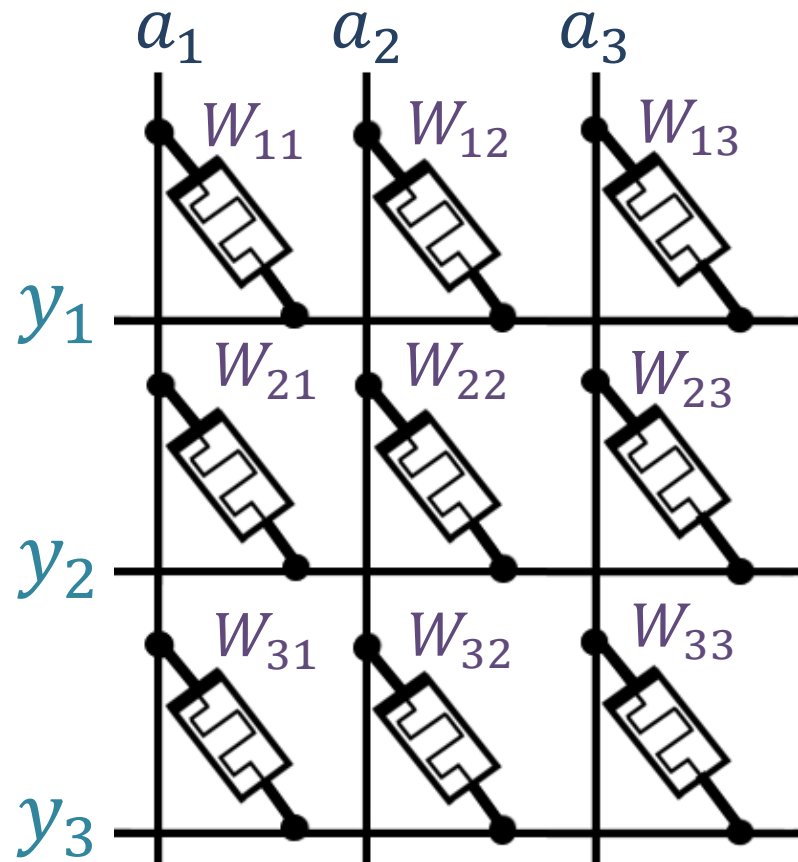
Stochastic Gradient Descent Training

The gradient value ΔW for fully connected layers:

$$\frac{\partial C}{\partial W} = ya^T$$

For Stochastic gradient descent (SGD):

$$W = W - \eta \frac{\partial C}{\partial W} = W - \eta ya^T$$



SGD Memristive Weight Update

Conductivity:

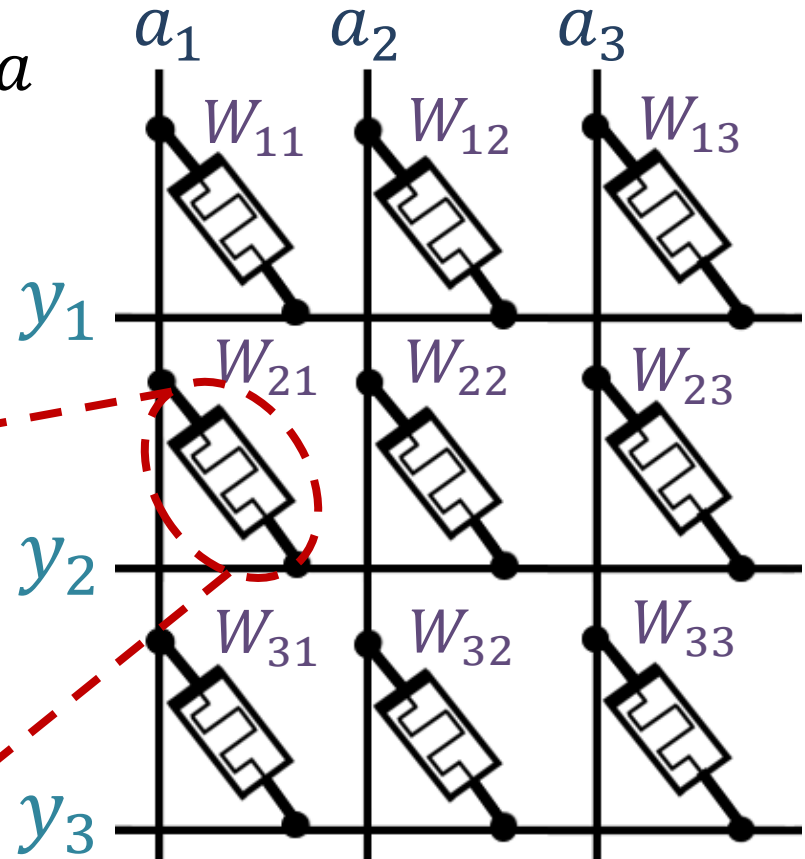
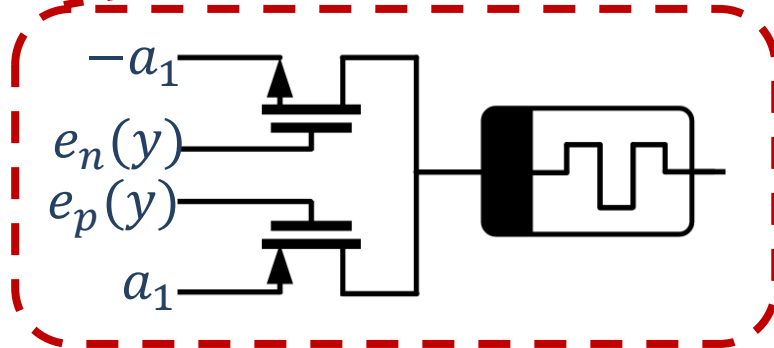
$$\Delta G = f_1(\Delta T_{wr} V) = f_1(\Delta W)$$

$$\Delta T_{wr} = \text{abs}(y), V = \text{sign}(y)a$$

Moving from **voltage**
to **time** and **voltage**

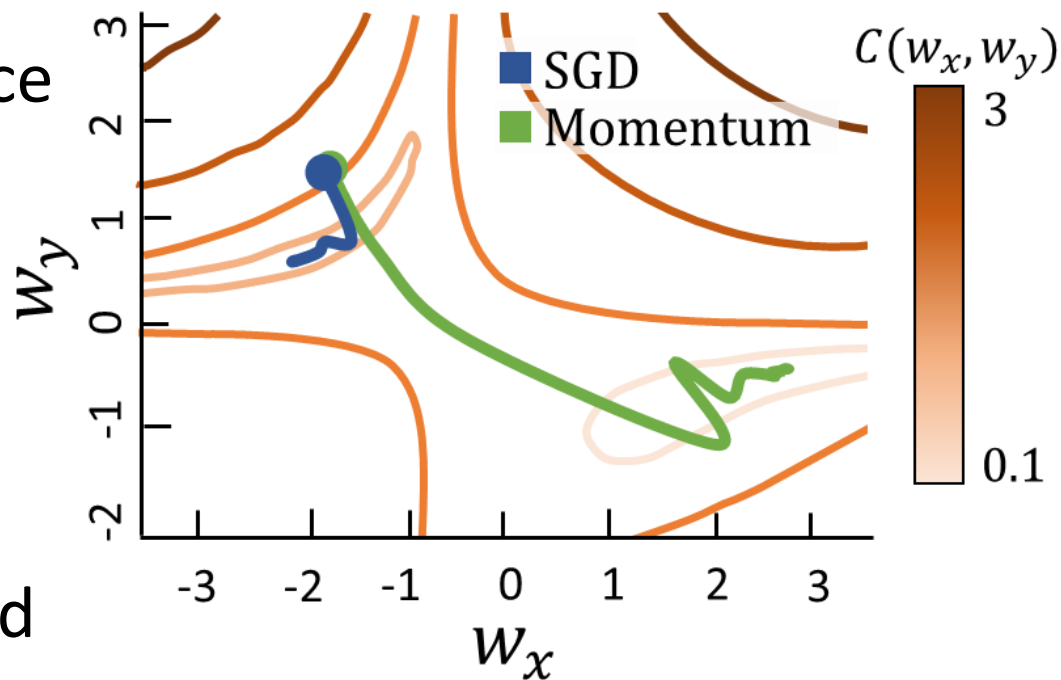
$a \rightarrow V$ (voltage)

$y \rightarrow e$ (voltage and duration)



Going Beyond SGD (Optimization Algorithms)

- Helps SGD performance
 - Increase accuracy
 - Accelerate convergence
- Momentum
 - Keeps the update history
 - Memory overhead
 - Computation overhead



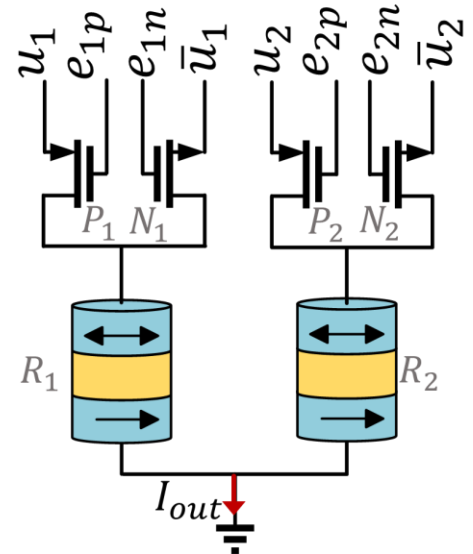
Agenda

- The limitations of modern AI hardware
- Neuromorphic computing with memristors
- Training memristive neuromorphic systems
- **Low-power memristive neuromorphic systems**
- Security and summary

Training at the Edge with Low Precision

- Low precision \rightarrow low energy, high speed
- Going as low as 1-2 bits (BNN/TNN) \rightarrow stochastic training
- Using Magnetic Tunnel Junction (MTJ)
- 3 TOPs/W for training

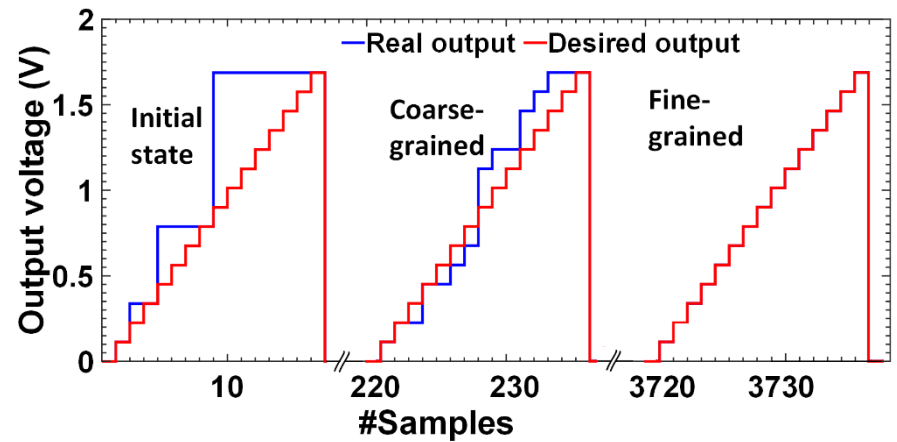
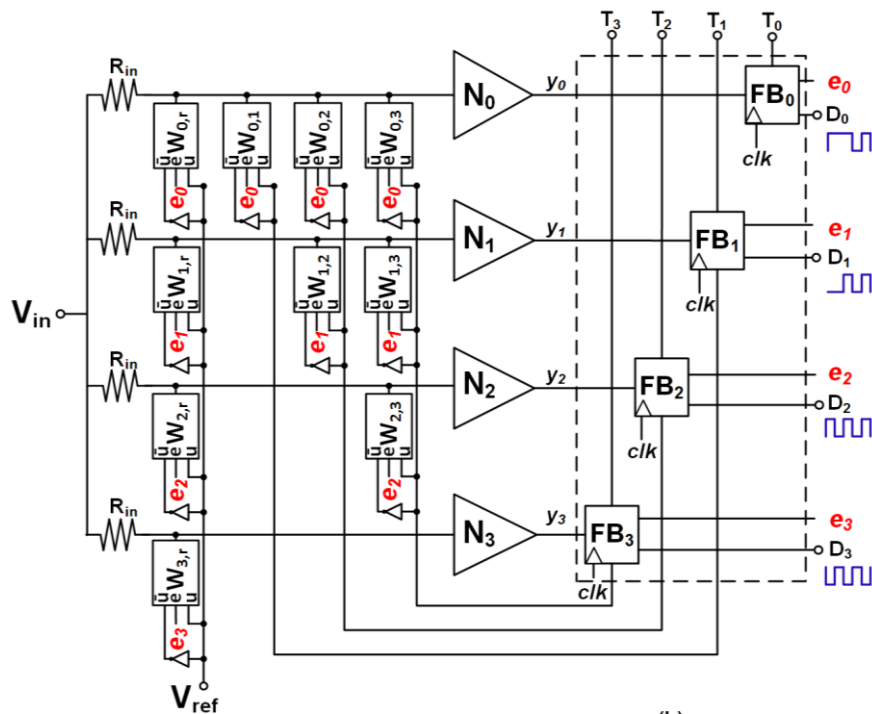
$$P_{sw} = P(\Delta t > \tau) = 1 - \operatorname{erf}\left(\frac{\pi}{2\sqrt{2}\theta_0 \exp\left(\frac{\Delta t V_{up}}{CR}\right)}\right)$$



Smart Data Converters

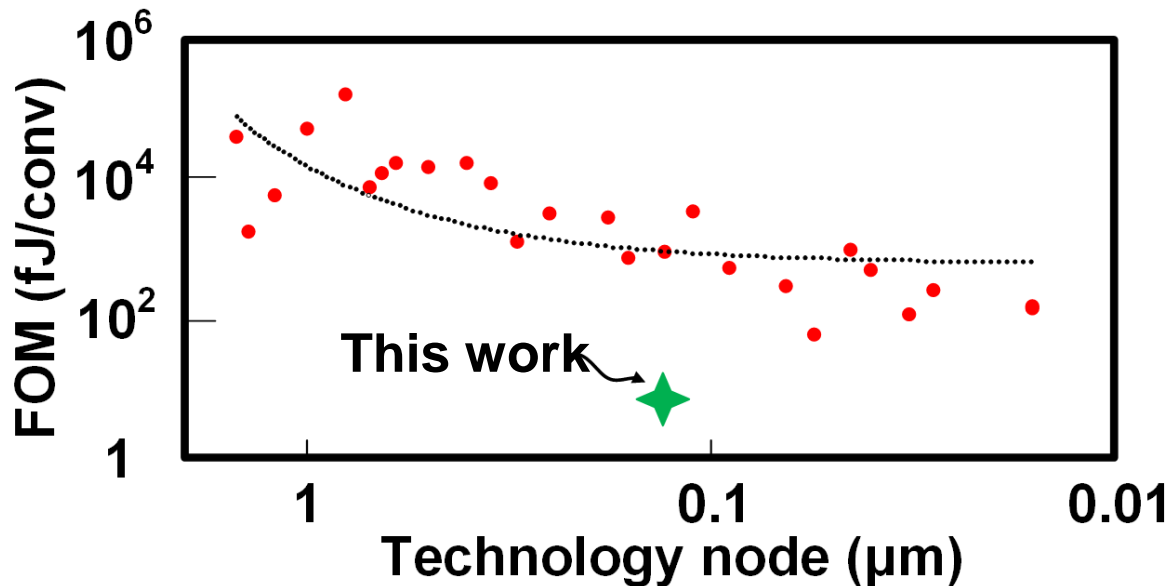
(Danial et al., Nanoarch 2018, TETCI 2018, JETCAS 2018, BioCAS 2019, ISCAS 2020, DATE 2020)

- Trainable A2D and D2A converters



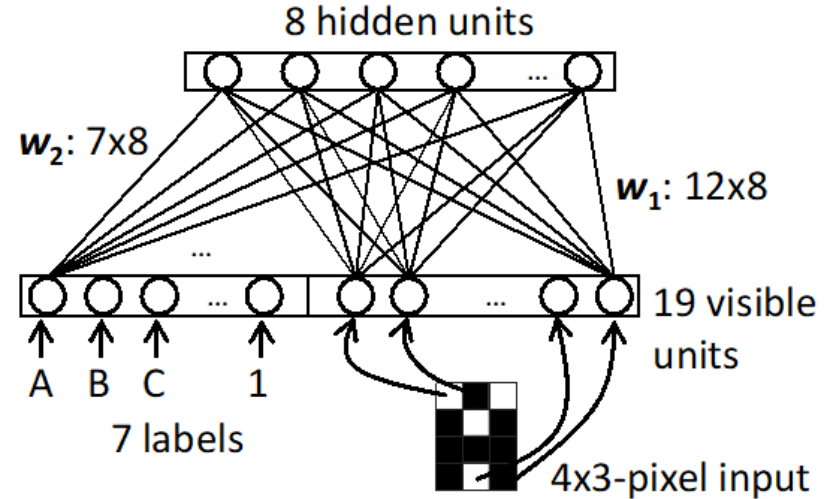
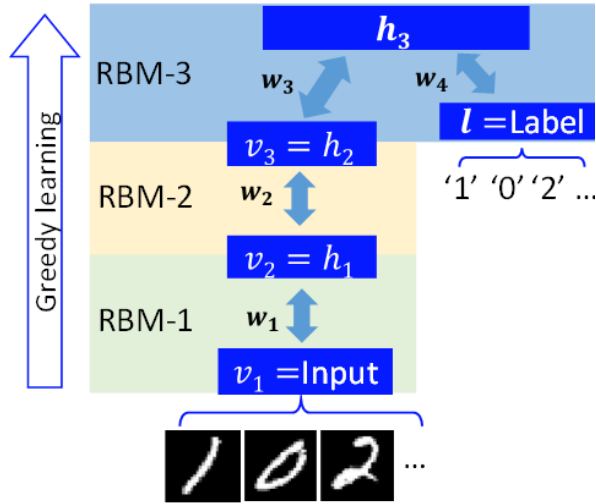
Advantages from Trainable Converters

- Generic and flexible (logarithmic for example)
- Excellent figure-of-merit (optimization)
- Self calibration (PVT, application)
- Sensor fusion and analog input, digital output



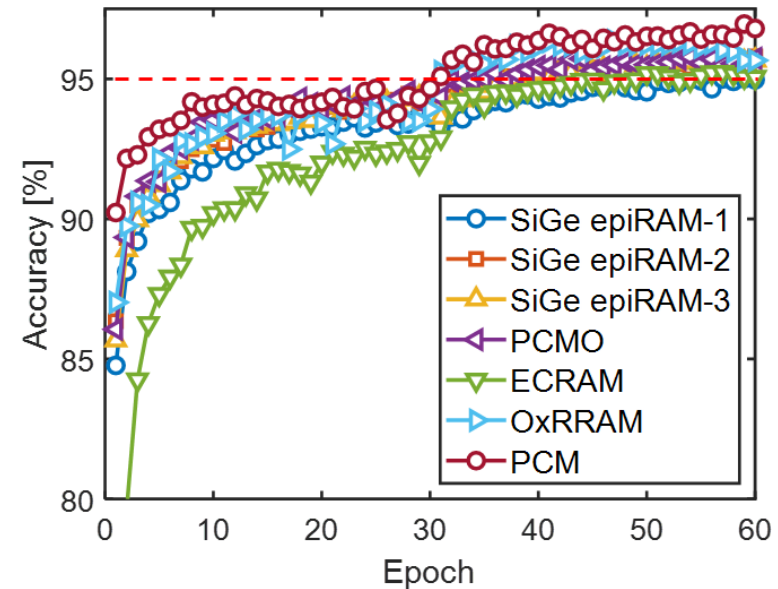
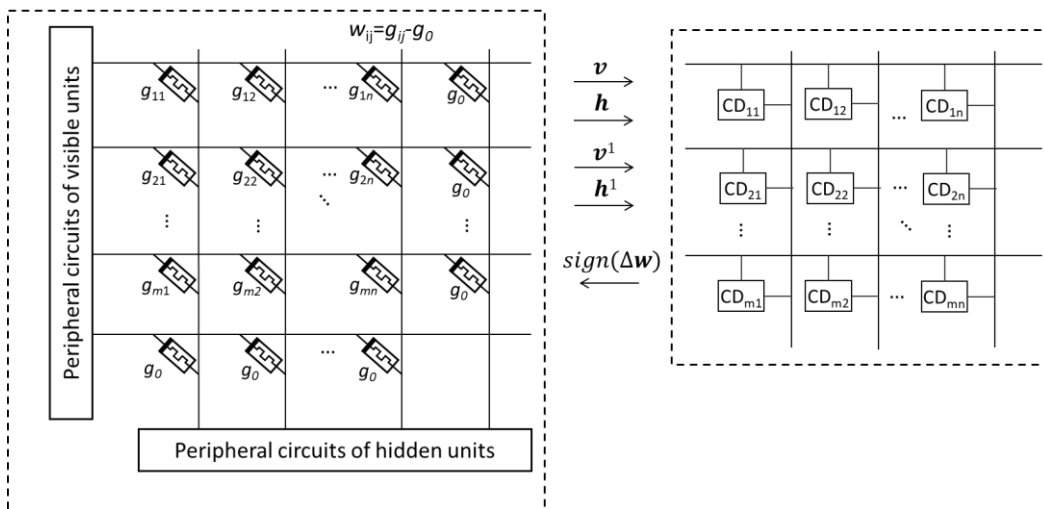
Simpler Neural Network Models

- Data conversion is the new bottleneck
- Moving to models that require no data conversion such as deep belief networks

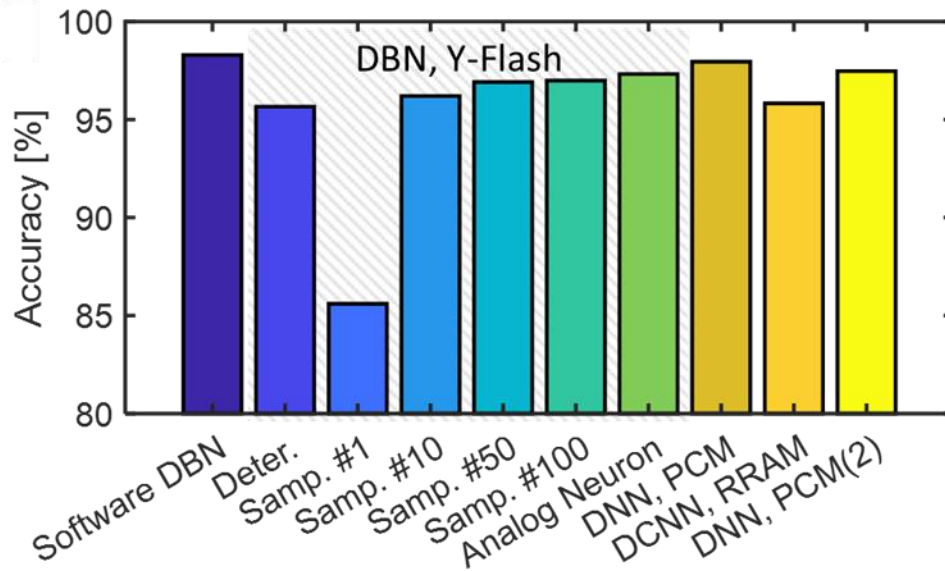
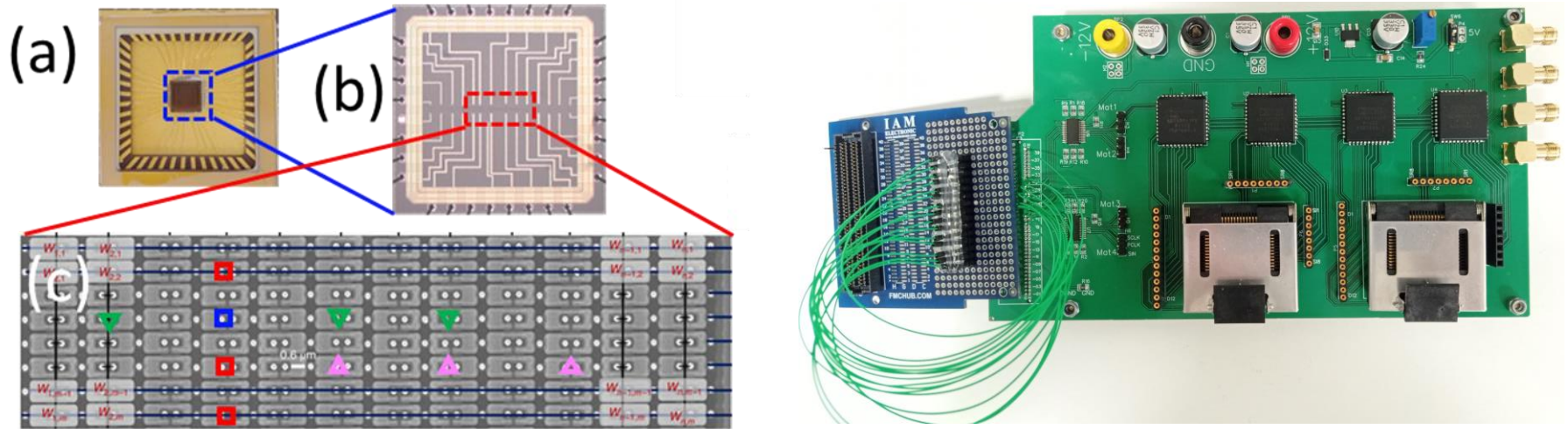


Memristive Deep Belief Networks

- 1R arrays + digital counter
- >97% accuracy for MNIST
- No analog, no nonlinear functions



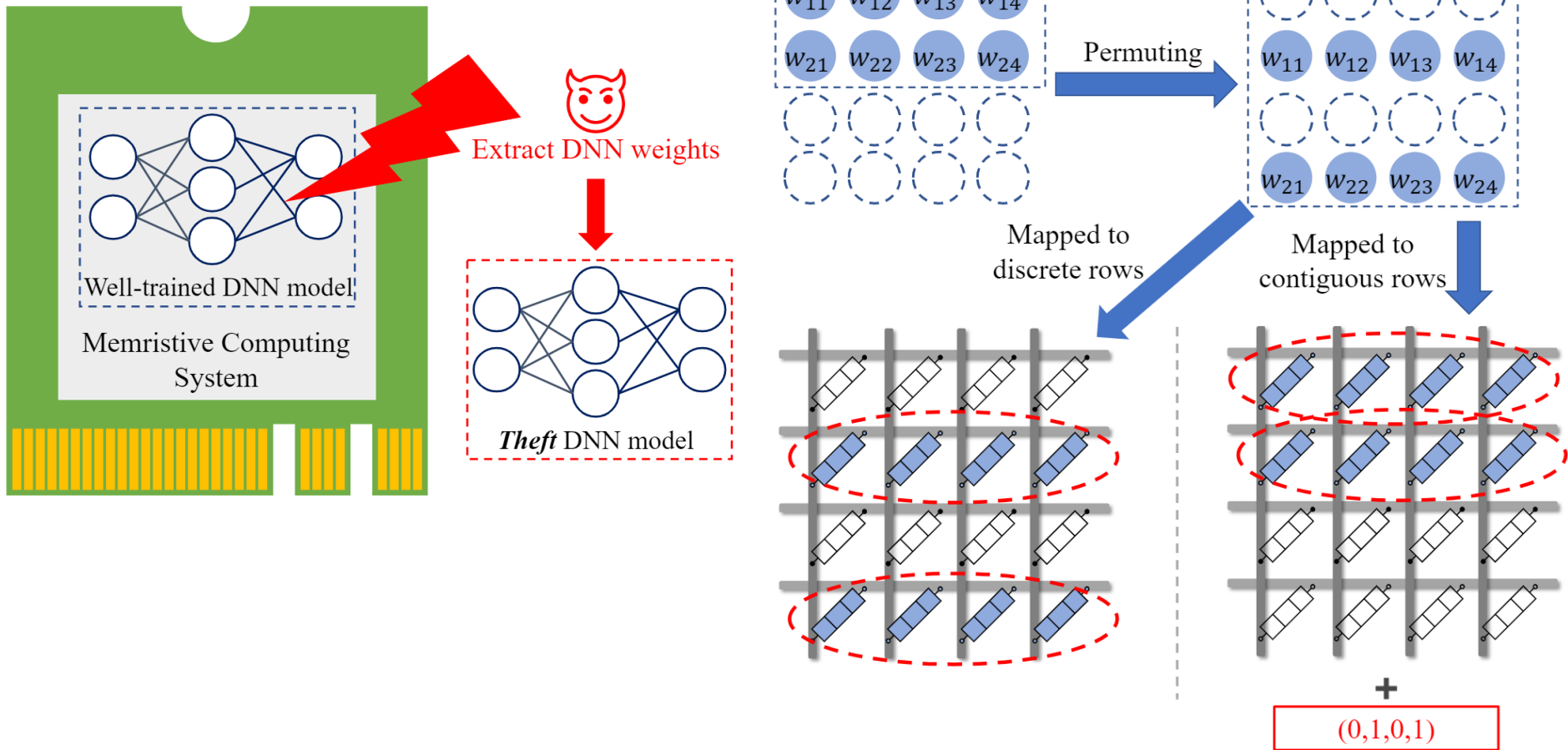
Y-Flash Memristor Deep Belief Networks



Agenda

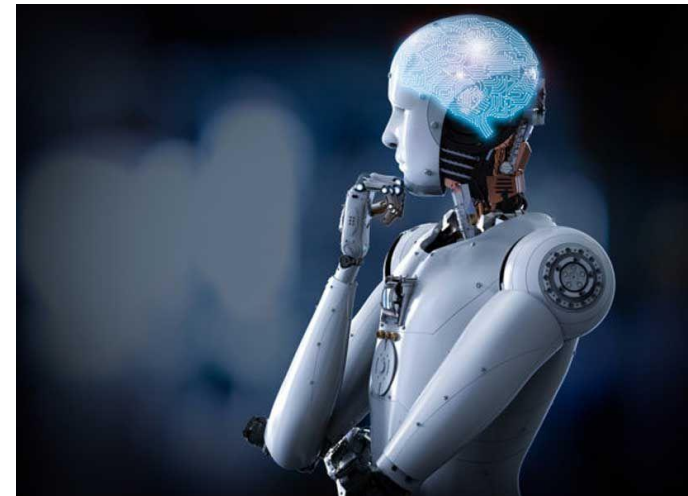
- The limitations of modern AI hardware
- Neuromorphic computing with memristors
- Training memristive neuromorphic systems
- Low-power memristive neuromorphic systems
- **Security and summary**

Security Issues



On-Device Memristive Neuromorphic ML

- Inspiration from biology
- Different technologies – Y-Flash, RRAM, PCM, volatile memristor
- On-device training → security, simpler networks



Thanks!



**BRING THEM
HOME NOW!**



**Horizon 2020
European Union Funding
for Research & Innovation**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the grant agreement 964877 - NEUCHIP.