Novel Technologies for Artificial Intelligence: prospects and challenges

Stefano Ambrogio
Pritish Narayanan
Robert M. Shelby
Hsinyu Tsai
Geoffrey W. Burr
IBM Research – Almaden

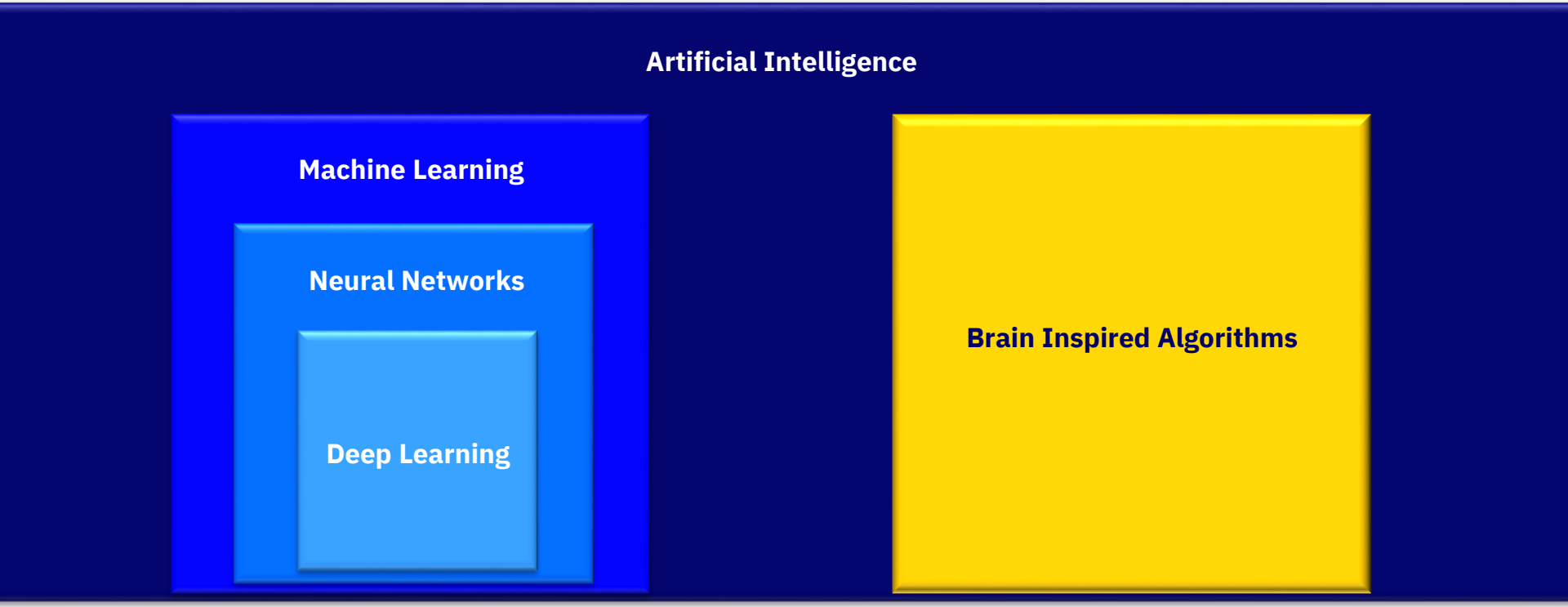
Outline

- Introduction
- A brain-inspired algorithm: **Spike Timing Dependent Plasticity**
- A machine learning algorithm: **Back-Propagation**
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Outline

- Introduction
- A brain-inspired algorithm: **Spike Timing Dependent Plasticity**
- A machine learning algorithm: **Back-Propagation**
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

What is AI?


Artificial Intelligence
Machine Learning
Neural Networks
Deep Learning
Brain Inspired Algorithms

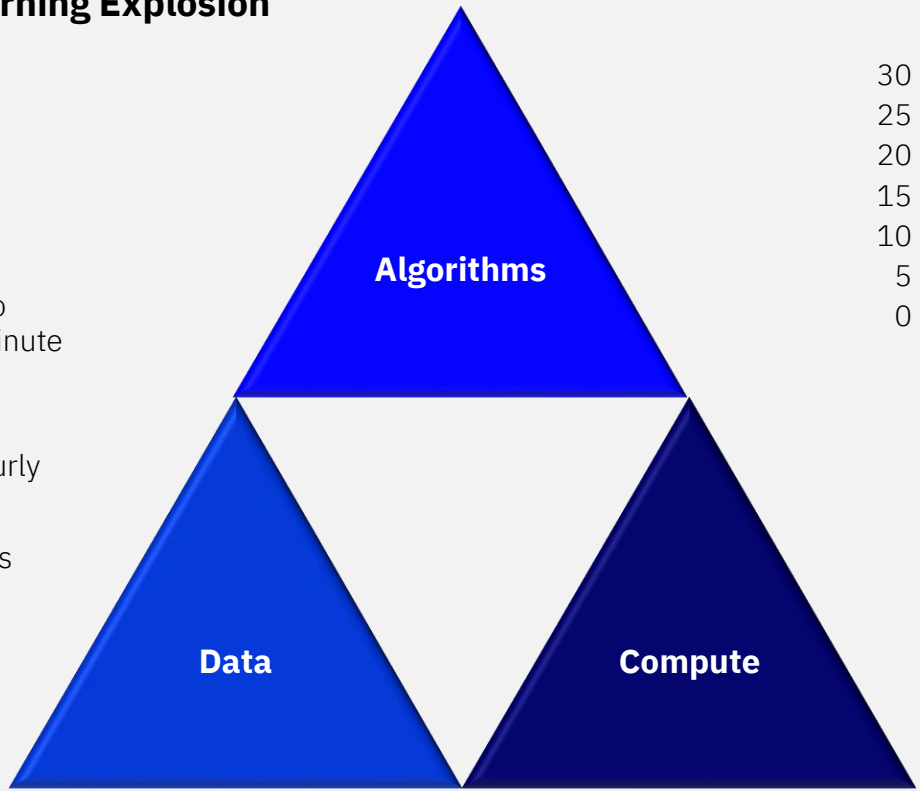
2012: AI foundations

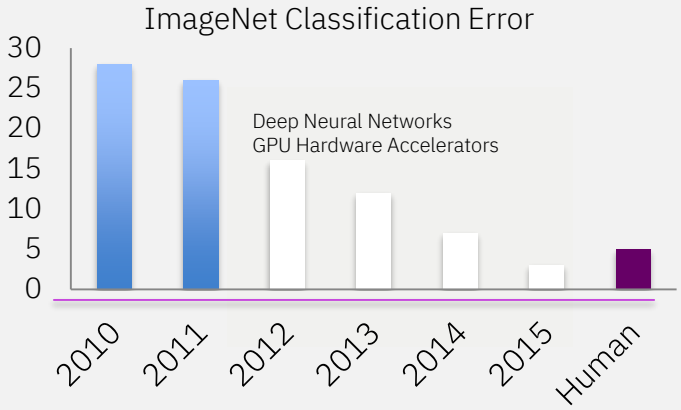
The Deep Learning Explosion

YouTube
400 hours of video uploaded every minute

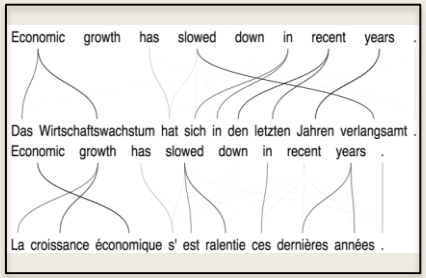
Walmart
2.5 petabytes of customer data hourly

Facebook
350 million images uploaded daily

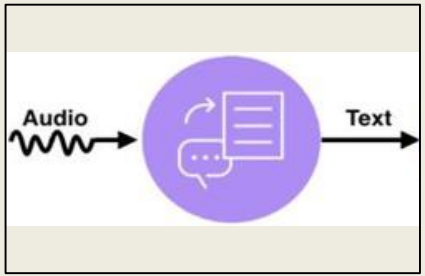




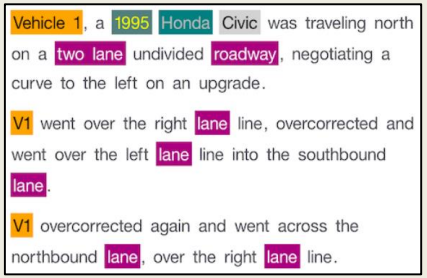
Tasks performed by specialized AI



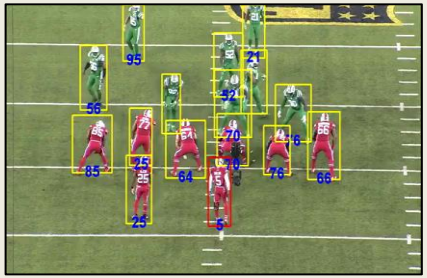
Language Translation



Speech Transcription



Language Processing

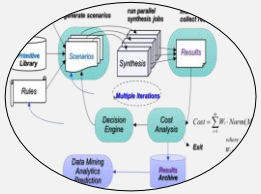


Object Detection

Face Recognition

Example AI challenges now tackled

Design Automation

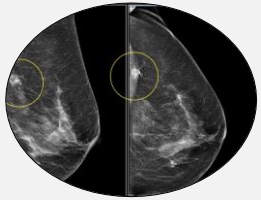


AI based design

Industrial

Guide me through fixing malfunctioning components

Healthcare



Improve the accuracy of breast cancer screening

Visual Inspection

Find rust on electric towers, using drones

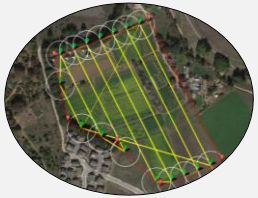
Customer Care

Bot that can guide a user through buying the right insurance policy

Marketing / Business

Summarize the strategic intent of a company based on recent news articles

IoT



Predict yield of field based on images and sensor data

Compliance

Is my organization compliant with latest regulatory documents

The evolution of AI

Narrow AI
Emerging

Broad AI
Disruptive and Pervasive

General AI
Revolutionary

▼ We are here

2050 and beyond

The evolution of AI

Narrow AI

Single task, single domain

Superhuman accuracy and speed for certain tasks

Broad AI

Multi-task, multi-domain

Multi-modal

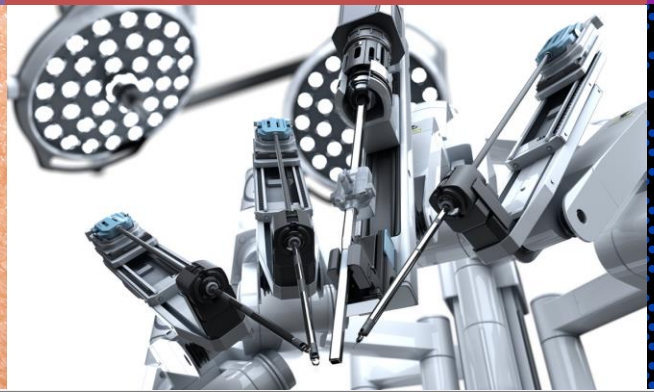
Distributed AI

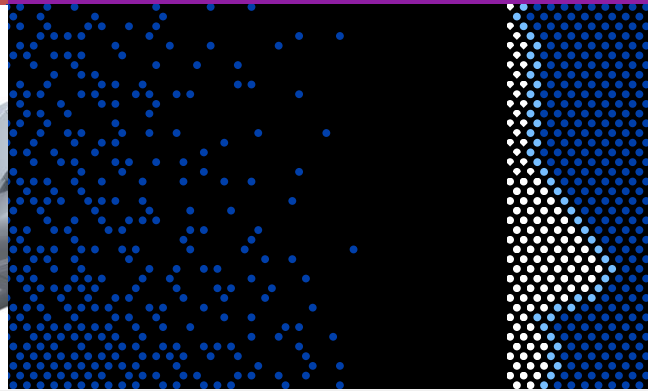
Explainable

General AI

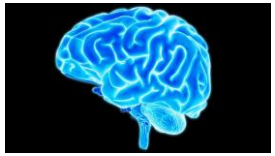
Cross-domain learning and reasoning

Broad autonomy





Neuromorphic hardware

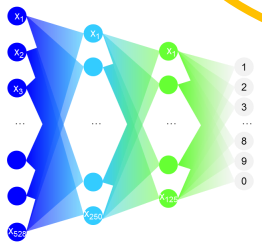


Brain-Inspired hardware

Algorithms: Hebbian Learning, STDP,… (Spike-Timing-Dependent Plasticity)

Naturally implemented with Spikes due to strong dependence on relative timings

Neuromorphic Hardware



Deep Neural Networks Hardware

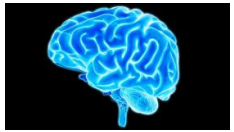
Algorithms: Back-Propagation

Non-Spike Implementation (classic approach)

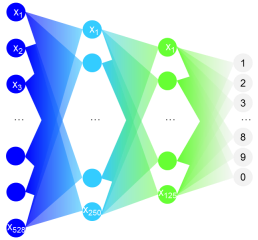
Spike Implementation (*TrueNorth*)

The choice of which hardware and algorithm to use strongly depends on the desired application

Which Algorithm?



Different algorithms provide different solutions



Brain Inspired Algorithms

Provide unsupervised online learning
→ Learning of novel unseen classes

Not very good at classification, accuracy on datasets generally lower than Back-propagation

The brain-inspired conception is not useful as a feature, more interest in the result we can achieve

Back-Propagation

Provides highest achievable accuracy
Algorithm very well known

Classification of novel unseen classes not available (Catastrophic forgetting)
Need for medium/large training datasets

The choice of the network type (Fully Connected, Convolutional, …) depends on the available power, area and complexity constraints

Spike or continuous implementation?

It is generally believed that Spike implementations (whether bio-inspired or not) provide less power consumption due to chip activation only when needed

It depends on several factors…

In case the information is NATURALLY spike based

→ typically very power-efficient

In case the information is NON-Spike based

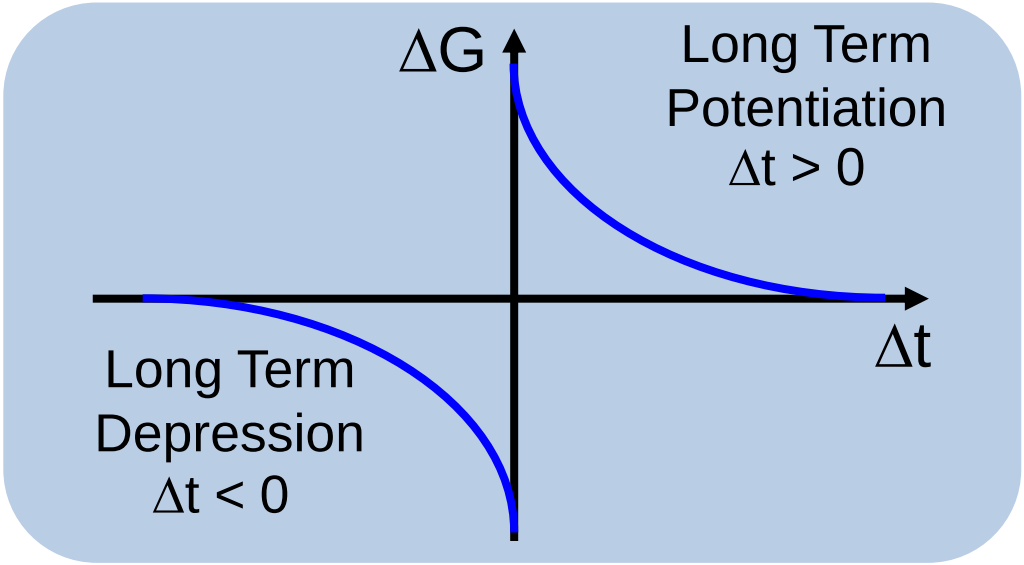
→ the circuitry to convert information into spike consumes power, which sometimes is not taken into consideration

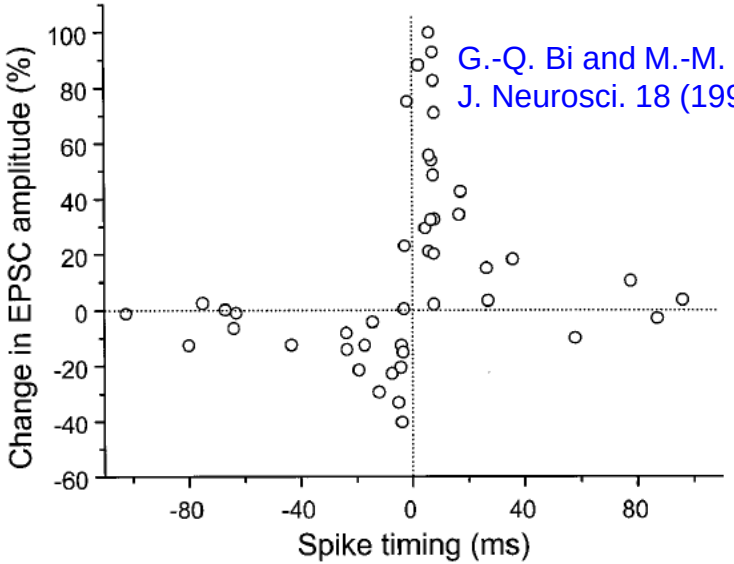
Spike implementation is generally slower to react, since it needs to capture many spikes to provide right answers

Outline

- Introduction
- A brain-inspired algorithm: **Spike Timing Dependent Plasticity**
- A machine learning algorithm: **Back-Propagation**
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Brain Inspired: Spike Timing Dependent Plasticity

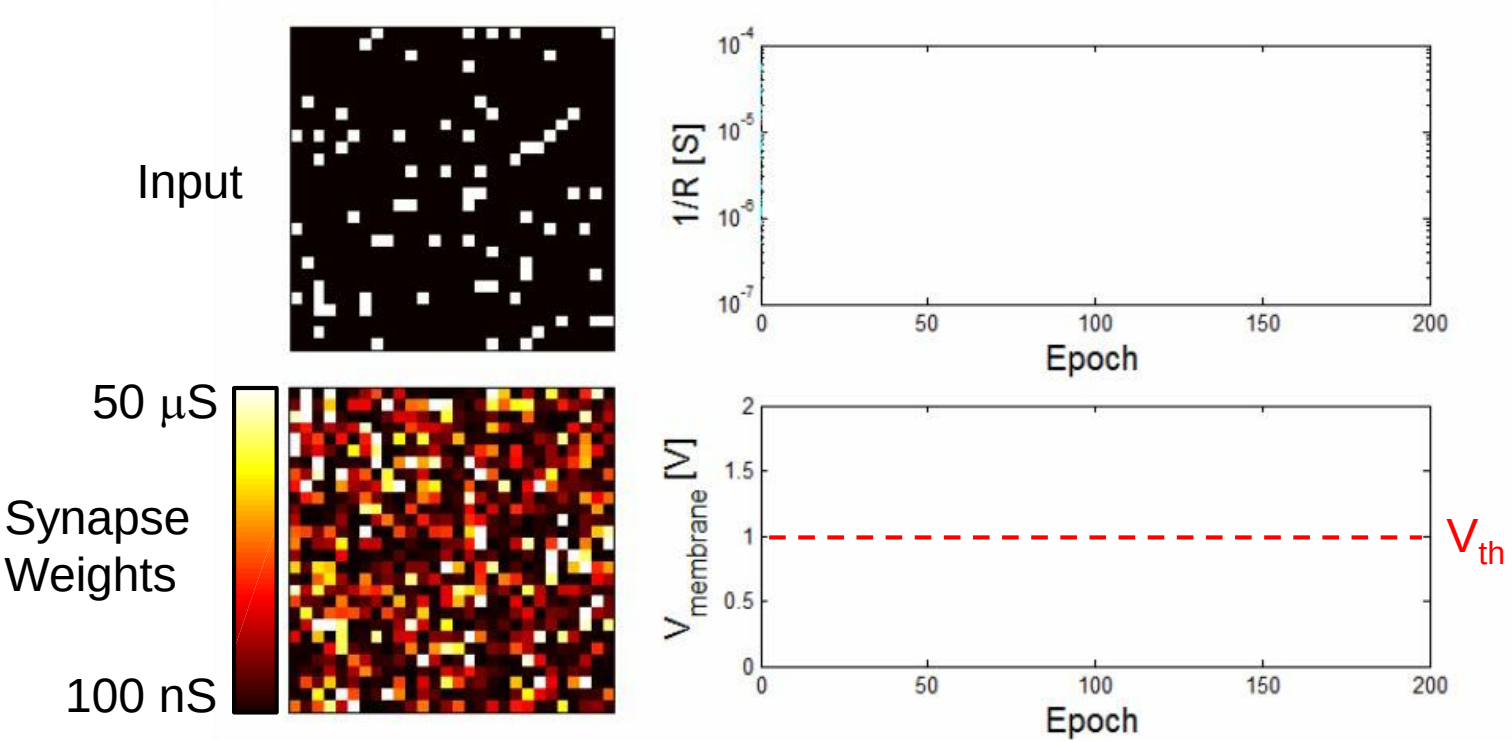




G.-Q. Bi and M.-M. Poo, J. Neurosci. 18 (1998)

- Biological protocol for synapse weight update

Unsupervised learning of features



S. Ambrogio, et al., Symp. VLSI, 1-2 (2016)

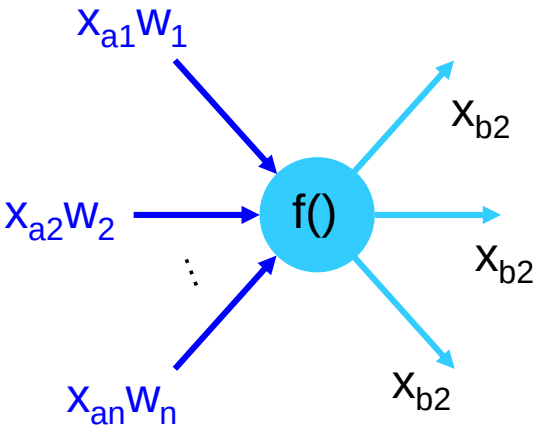
- The network enables robust implementation of pattern learning and recognition

Outline

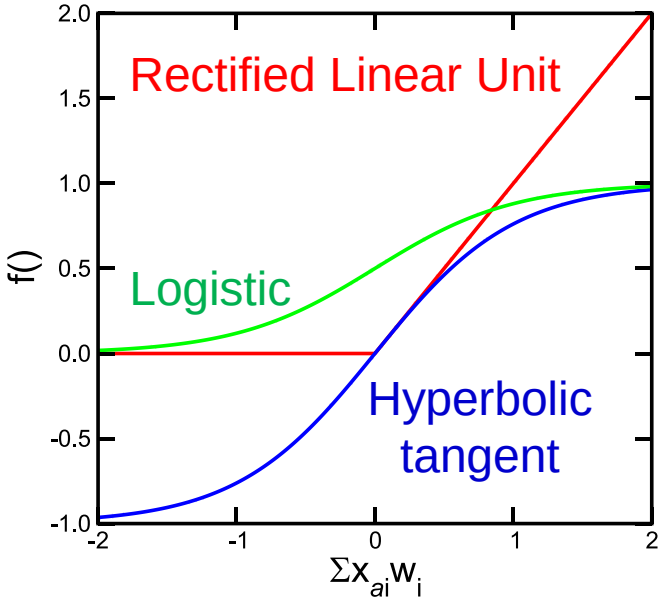
- Introduction
- A brain-inspired algorithm: **Spike Timing Dependent Plasticity**
- A machine learning algorithm: **Back-Propagation**
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

What is backpropagation?

- Global learning rule for training Fully Connected and Convolutional Neural Networks

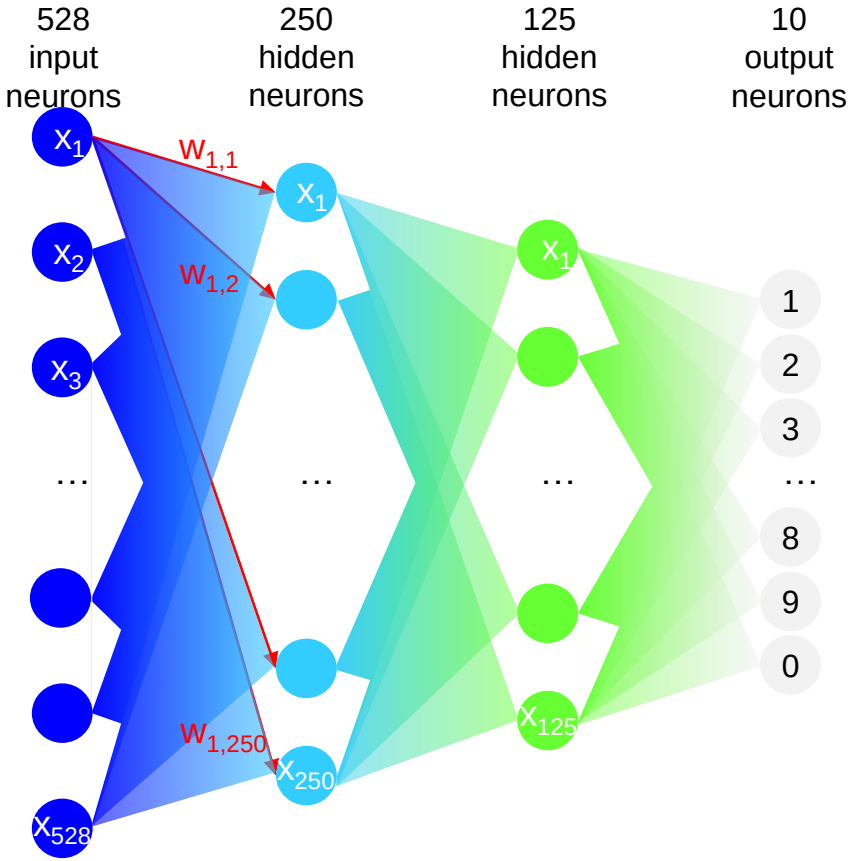


$$x_{bi} = f(\Sigma\, x_{ai} w_i)$$

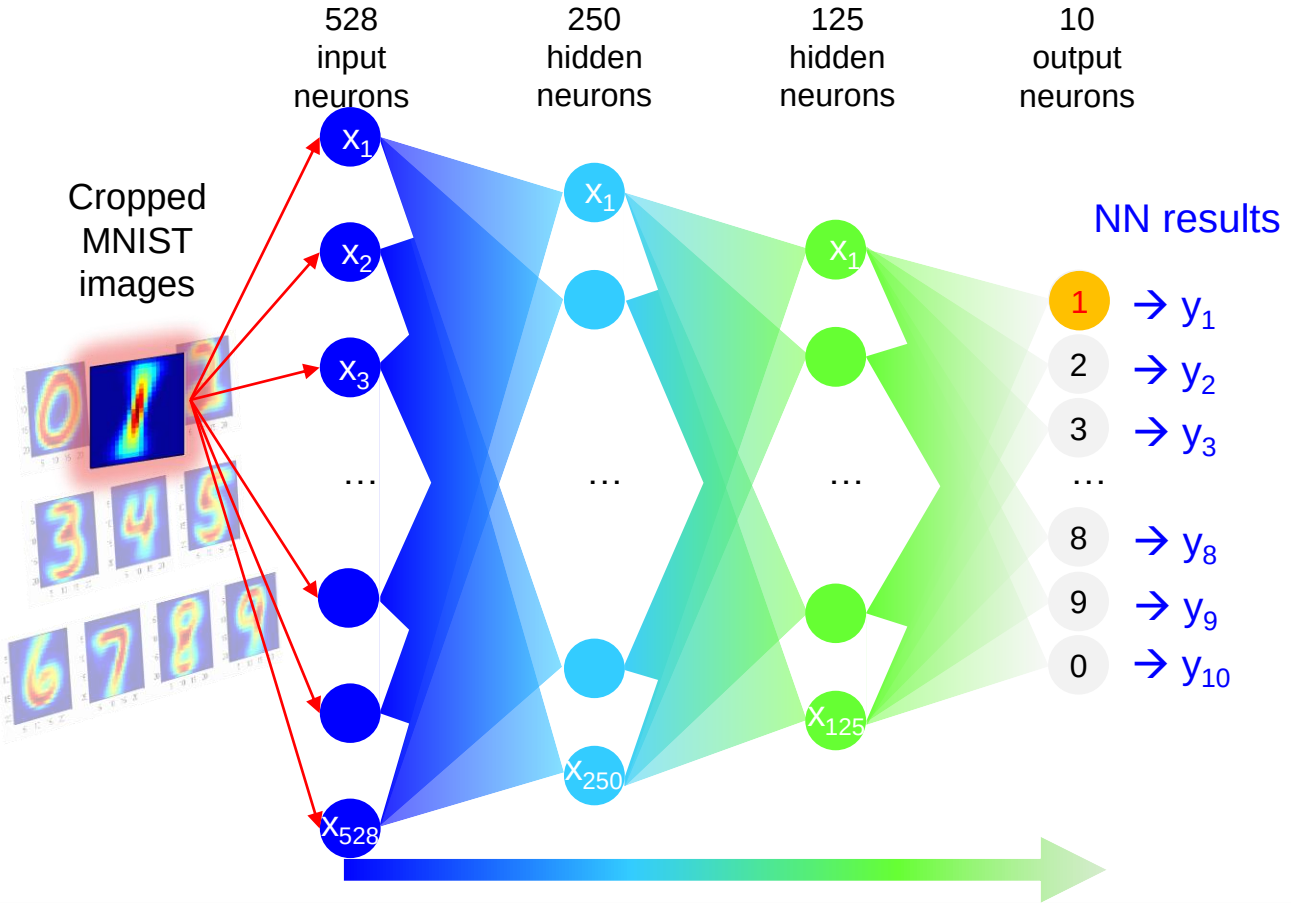


- The neuron output is the weighted sum of the inputs, passed through a nonlinearity

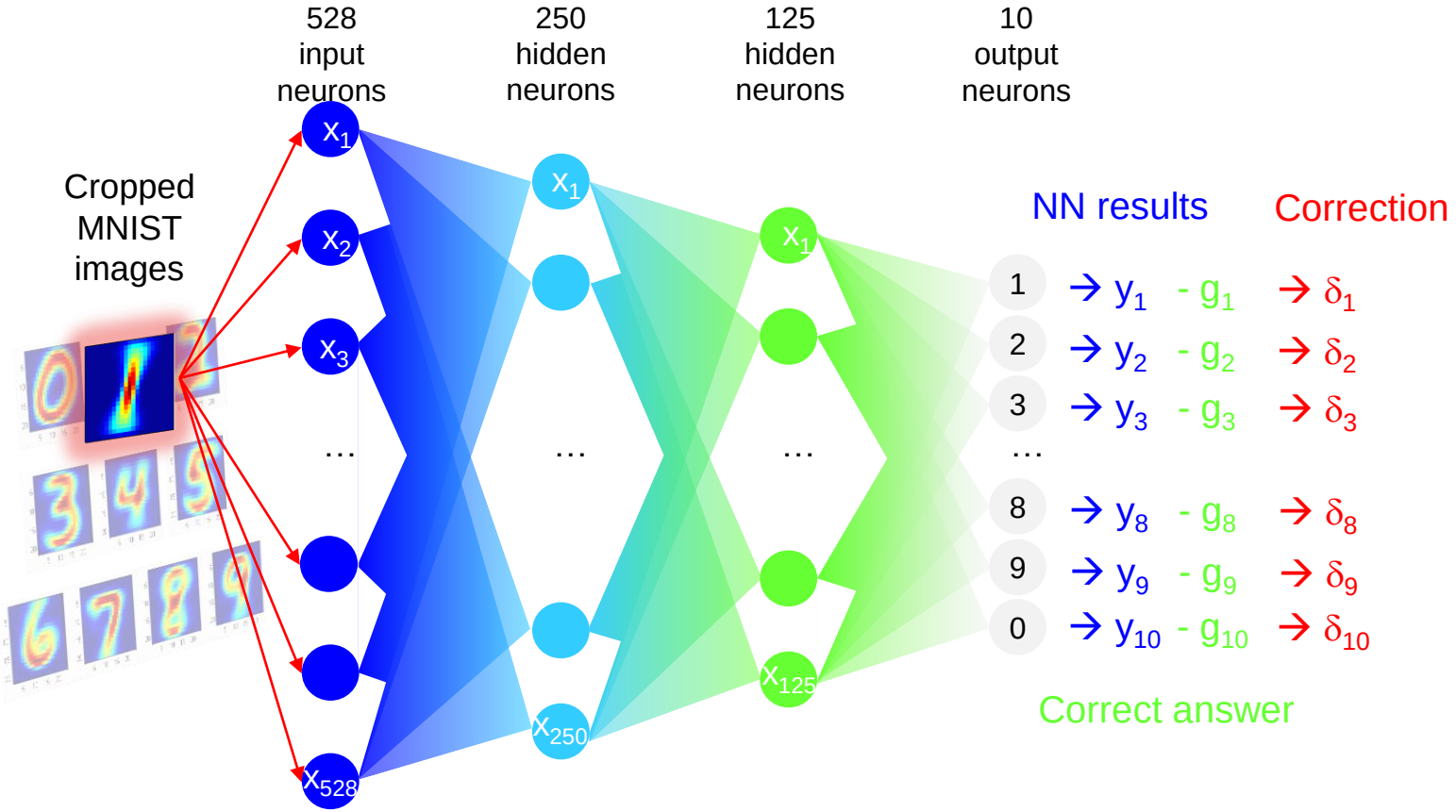
Network topology



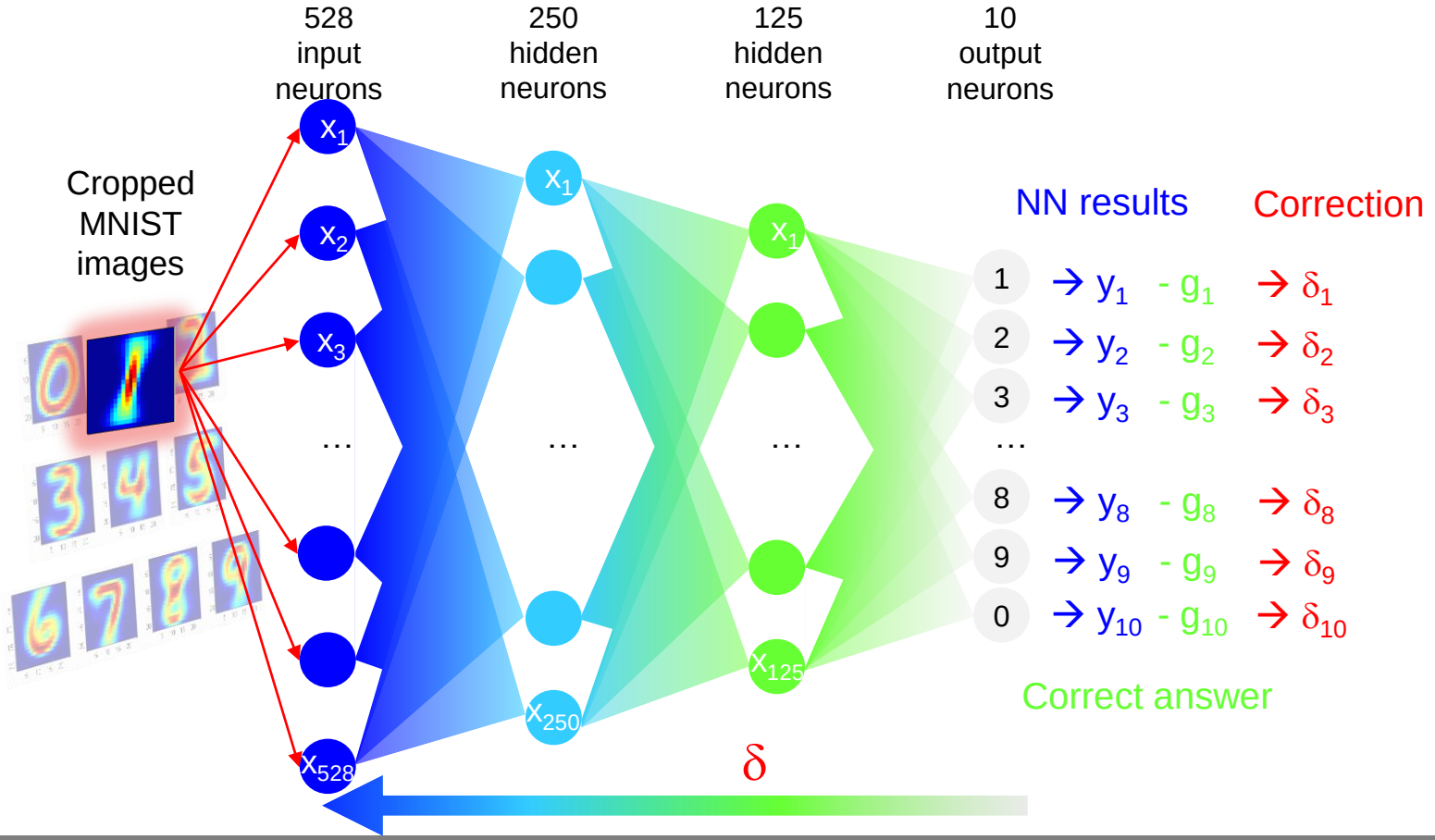
Forward propagation



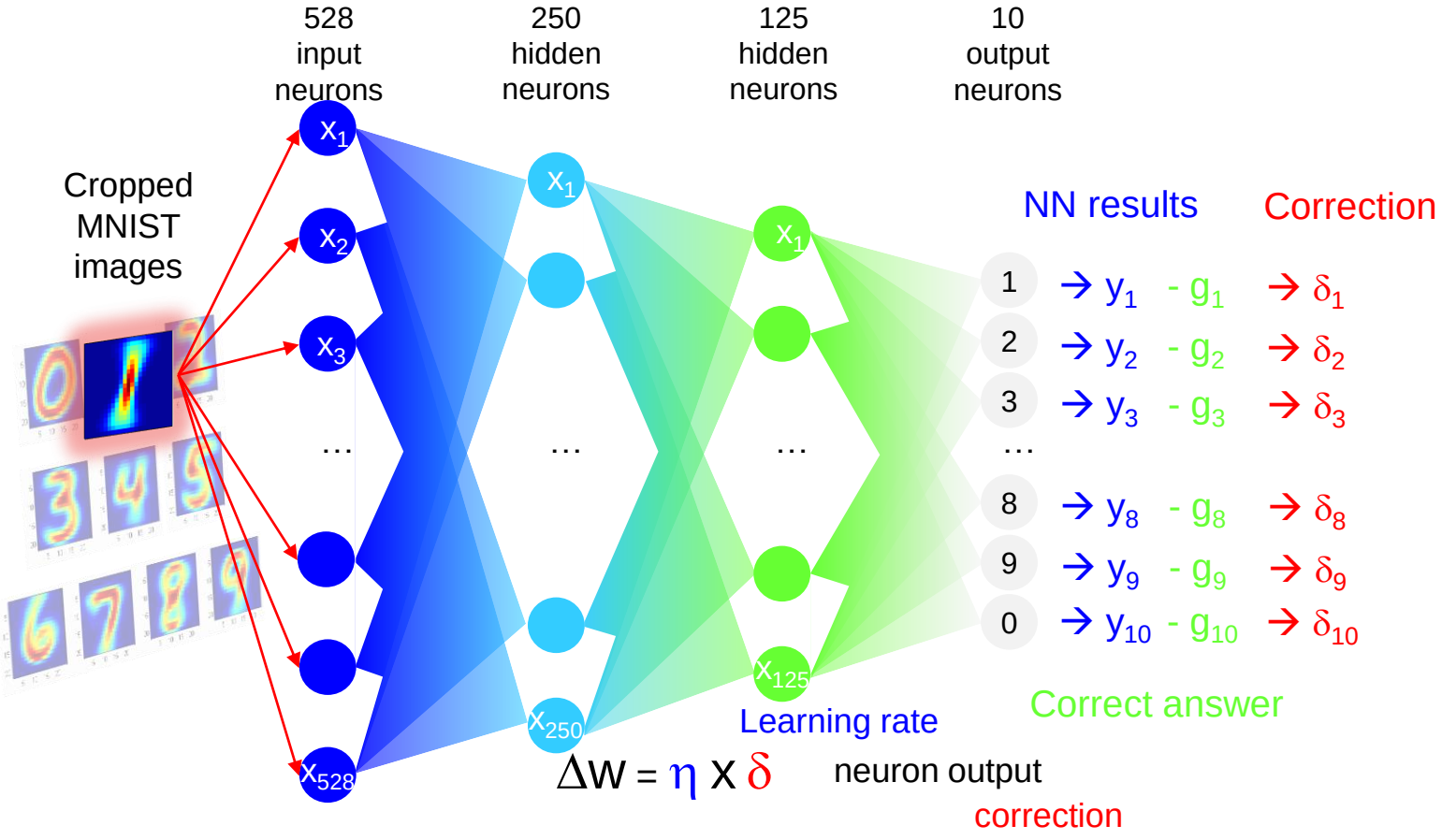
Comparison with the correct answer



Backpropagation

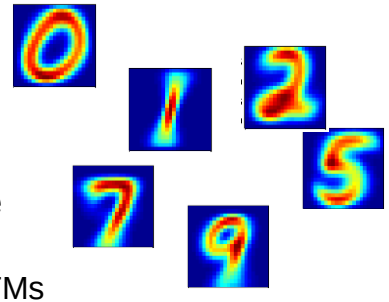


Weight update

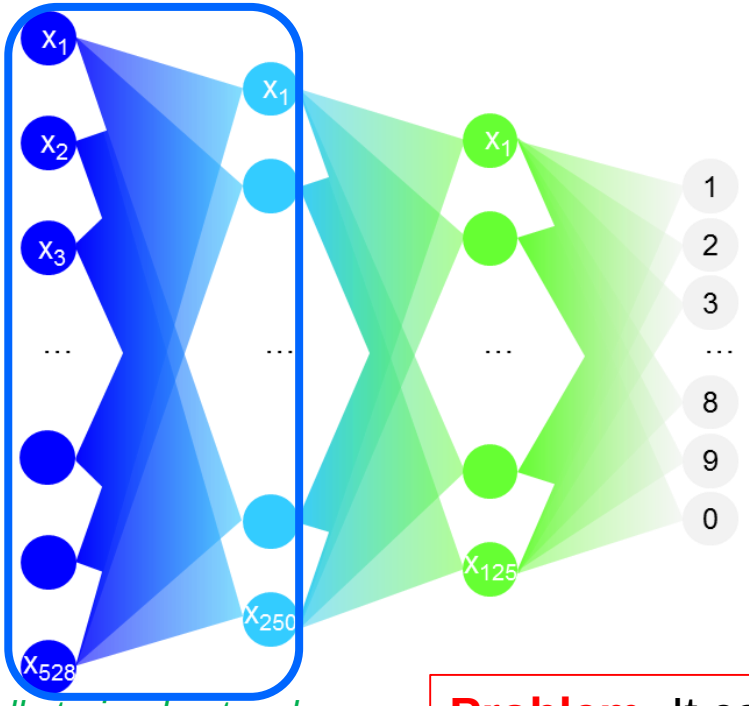


Hardware opportunities

Input data (images, raw speech data, etc.) input to neural network



"MNIST" database
~1998
→ check-reading ATMs



Forward inference:

Fully trained network

Hardware opportunity: Efficient, **low-power** deplo

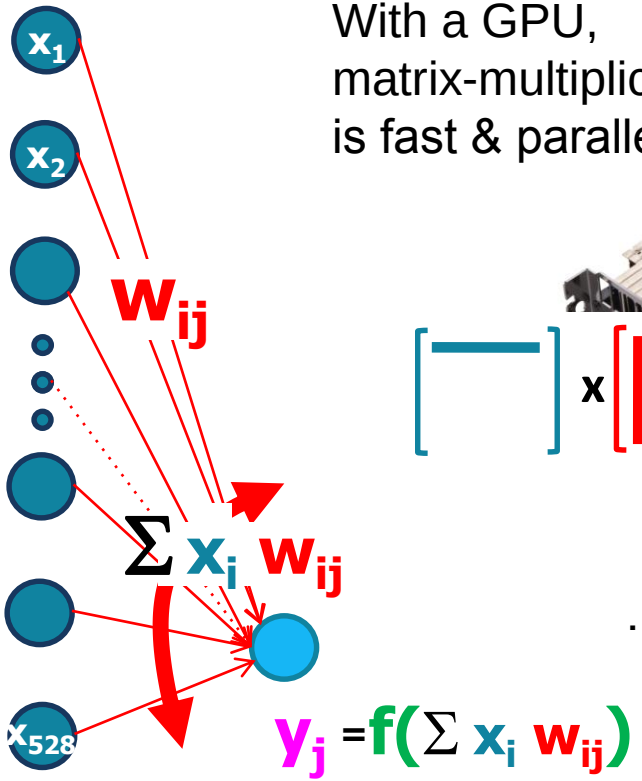
Training:

UN-trained network

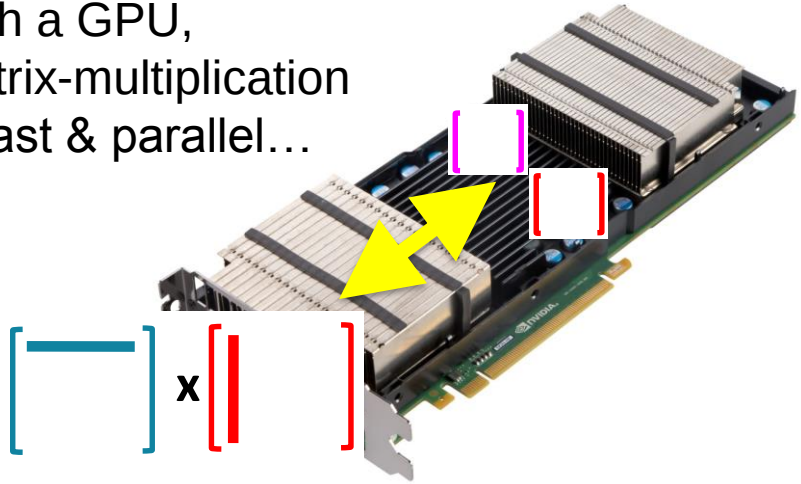
Problem: It can take WEEKS to train these networks, even with many GPUs.

Hardware opportunity: Train big networks FASTER and at LOWER POWER.

Computation needed: “Multiply-accumulate”

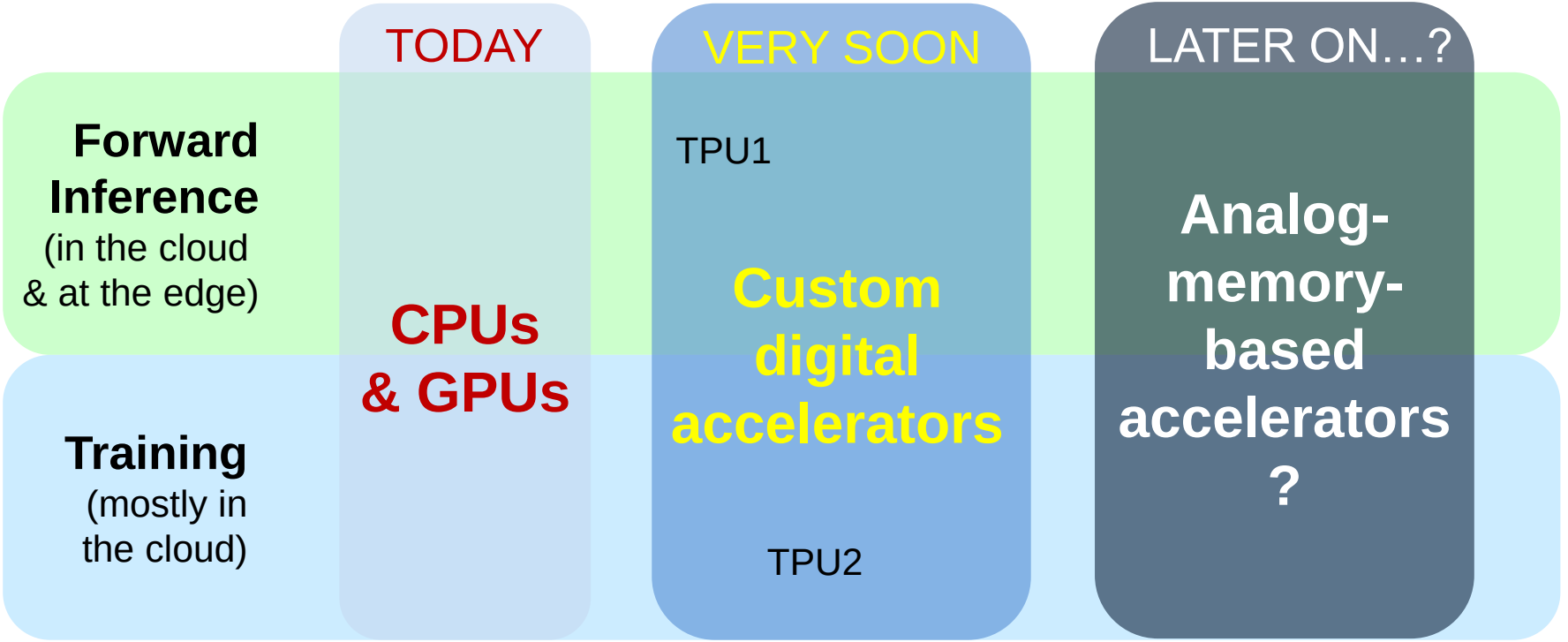


With a GPU,
matrix-multiplication
is fast & parallel…



… but x and w values must arrive from DRAM,
and new y values sent back to DRAM

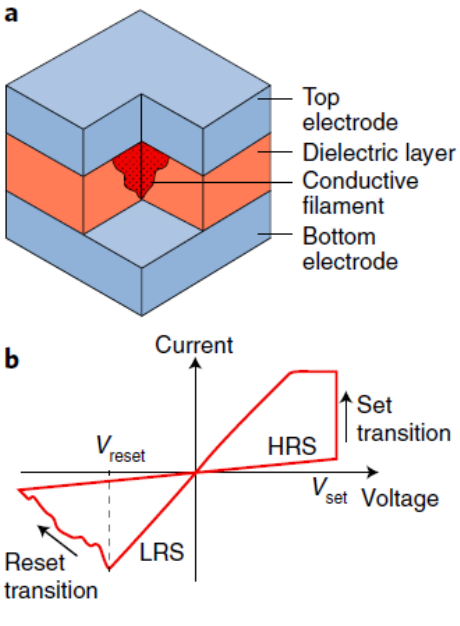
AI hardware, present & near-future: high-level view



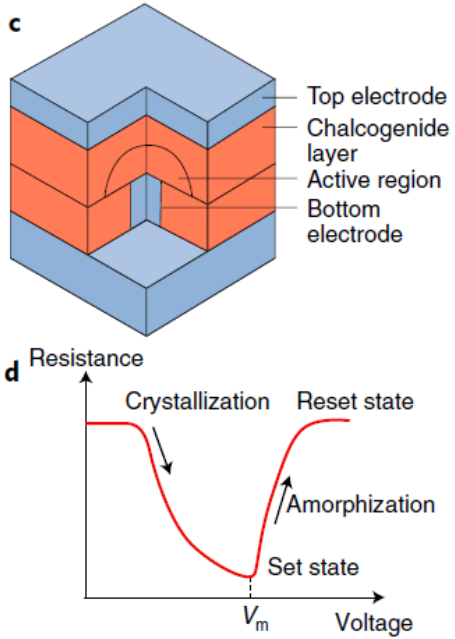
Outline

- Introduction
- A brain-inspired algorithm: **Spike Timing Dependent Plasticity**
- A machine learning algorithm: **Back-Propagation**
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

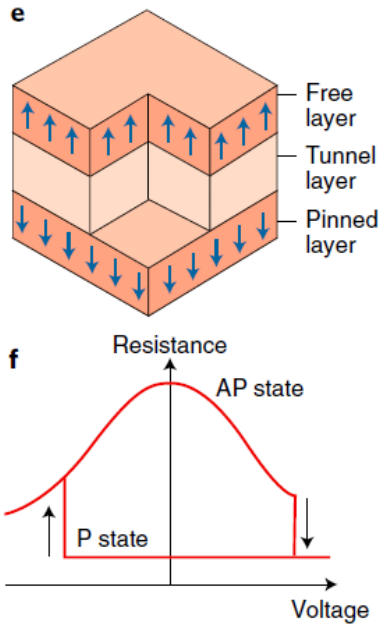
Emerging devices for memory and computing



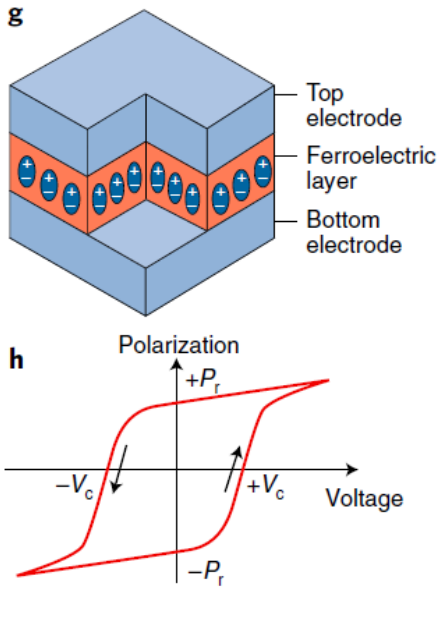
Resistive Memory (RRAM)



Phase-Change Memory (PCM)



Magnetic Memory (MRAM)



Ferro-Electric Memory (FeRAM)

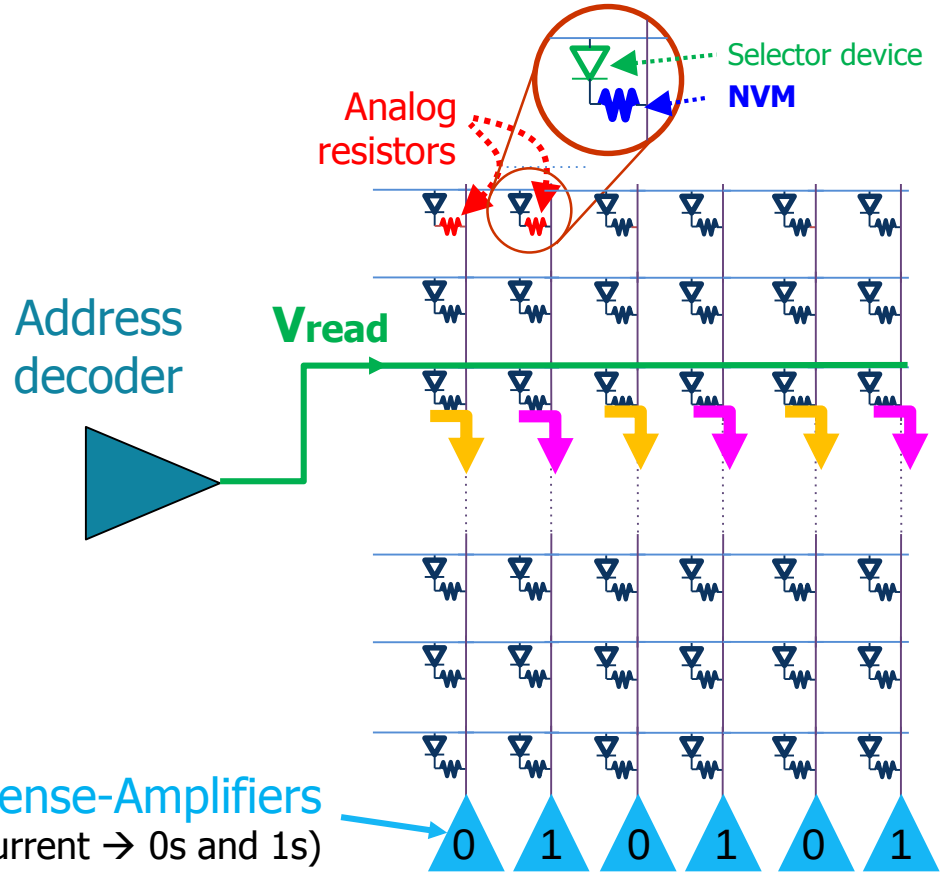
- Information encoded in the device conductance

D. Ielmini, H.-S. P. Wong, Nature Electronics (2018)

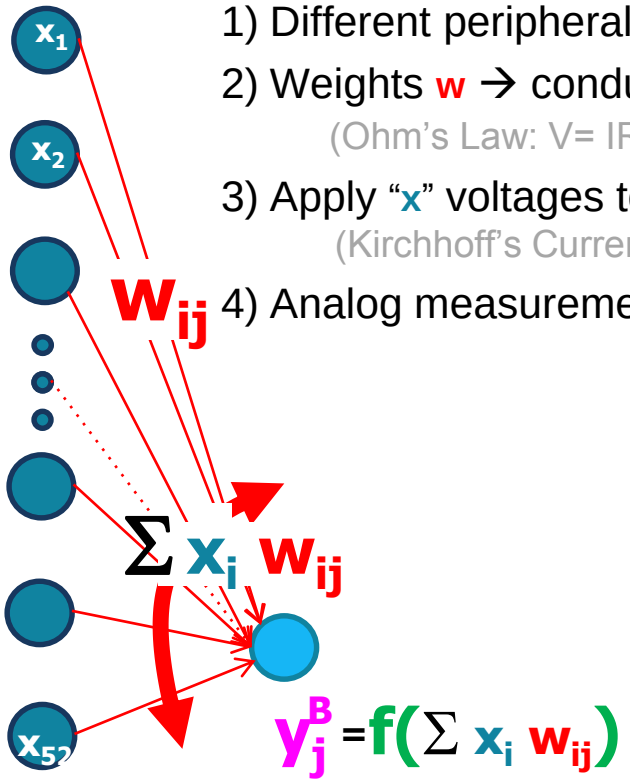
NVM (Non-Volatile Memory): usually for storing digital data (0s and 1s)

NVM technologies include:
MRAM (Magnetic RAM)
PCM (Phase-Change Memory)
RRAM (Resistance RAM)

Like conventional memory (SRAM/DRAM/Flash), an NVM is addressed one row at a time, to retrieve previously-stored digital data.



Multiply-accumulate with NVM: computed at the data, by physics...



1) Different peripheral circuitry

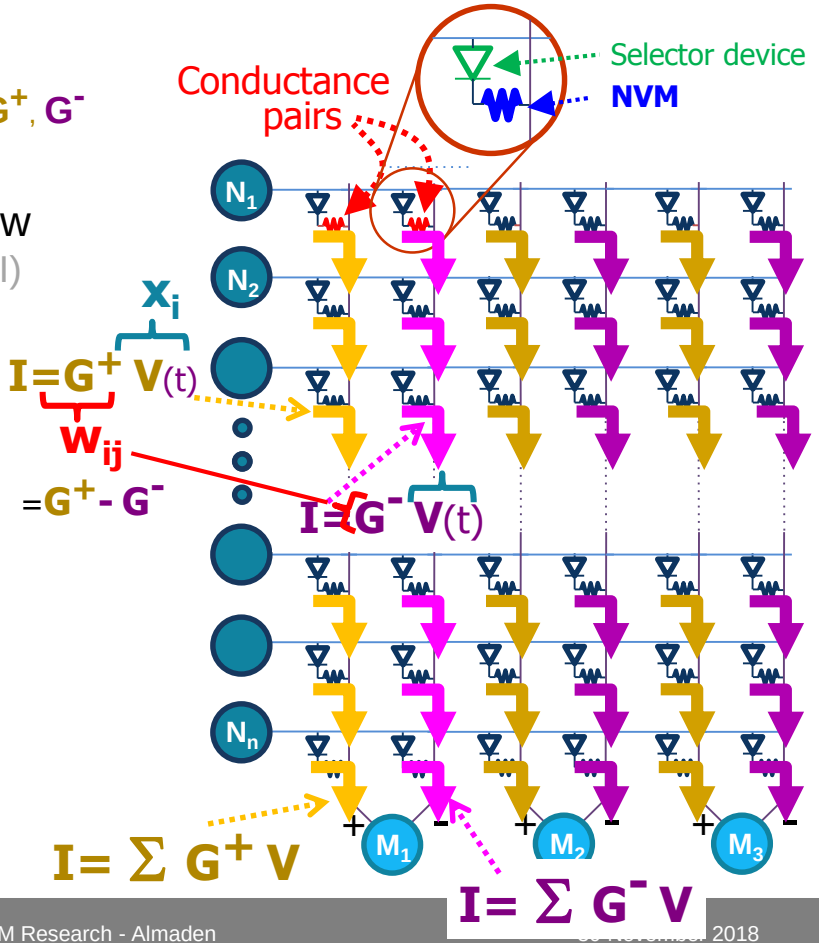
2) Weights **w** → conductances G^+, G^-

(Ohm's Law: V= IR → I = GV)

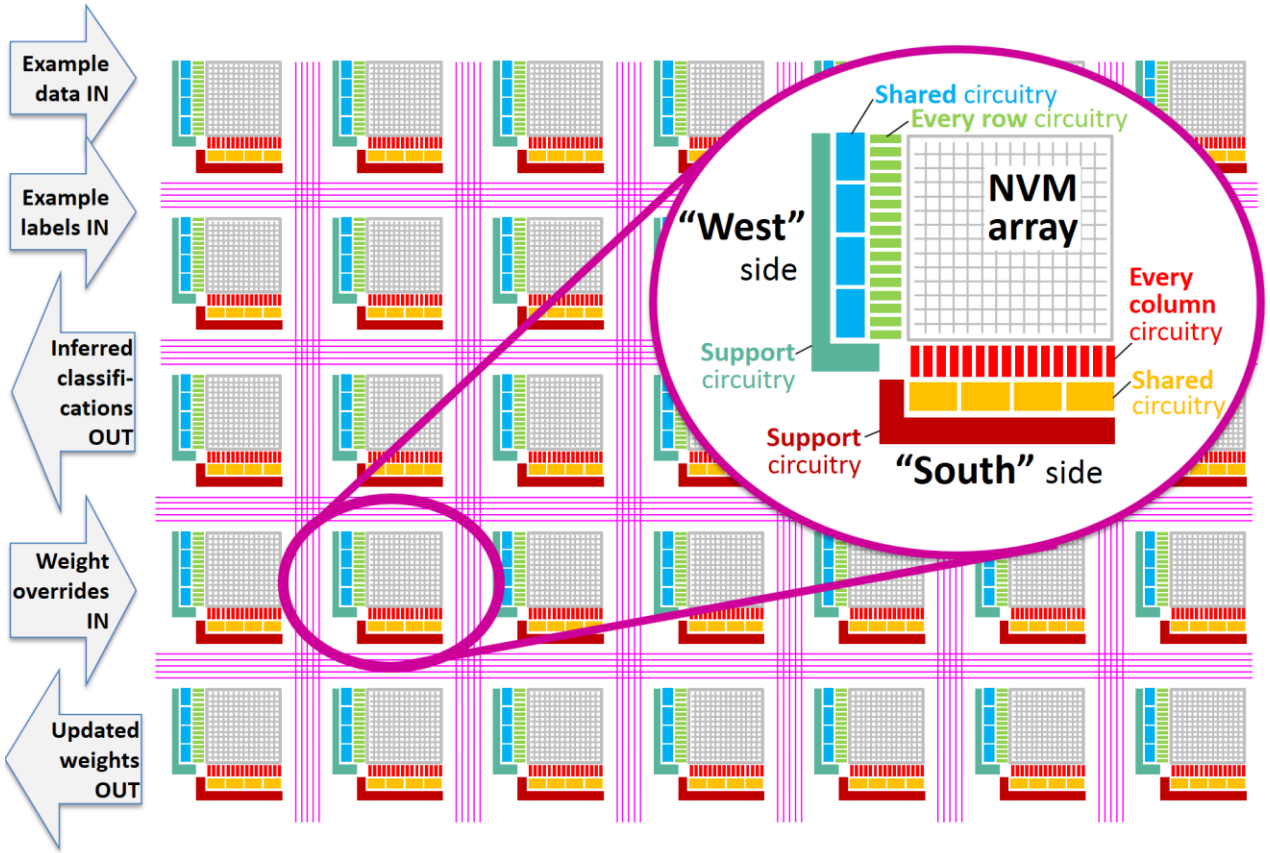
3) Apply "x" voltages to **every** row

(Kirchhoff's Current Law → Σ I)

4) Analog measurement



Vision: NVM-based Deep Learning Chip



- Support multiple deep learning algorithms
- Reconfigurable routing: Map different neural net topologies to the same chip
- Weight override mechanism for distributed learning

Maximizing the future business case (vs. a GPU)

Low Power

(inherent in the physics, but possible to lose in the engineering…)

Of zero interest

Still of interest for power-constrained situations: learning-in-cars, etc.

Accuracy

(essential that final Deep-NN performance be indistinguishable from GPUs – hardest technical challenge)

Of zero interest

Sweet spot: rather than buy GPUs, people buy this chip instead for training of Deep-NN's

Still of interest for some situations: learning-in-server-room

Of zero interest

Of zero interest

(circuitry must be massively parallel)

Faster

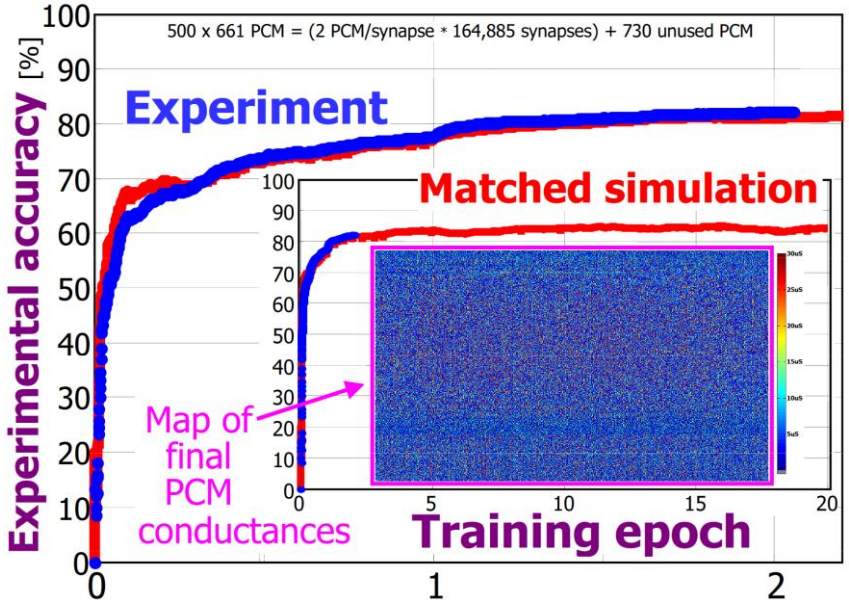
Outline

- Introduction
- A brain-inspired algorithm: **Spike Timing Dependent Plasticity**
- A machine learning algorithm: **Back-Propagation**
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Our journey towards high DNN accuracy

Where we were in **June 2014**

- Experiments on MNIST Dataset
- 82% accuracy w/ 5,000 examples,
- Too slow for 60,000 examples



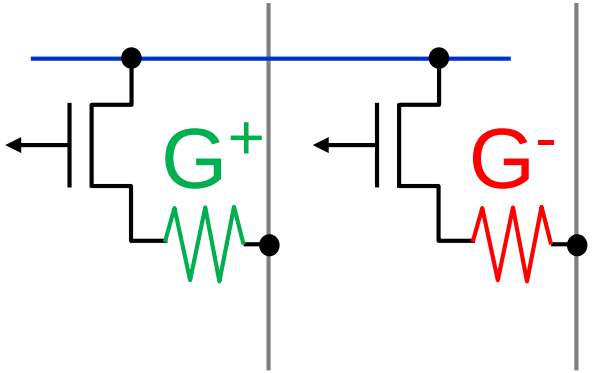
G. W. Burr, R. M. Shelby, et al., *IEDM Technical Digest*, 29.5, (2014).

"What a GPU would get" with this network…

97-98% TEST accuracy w/ 60,000 examples

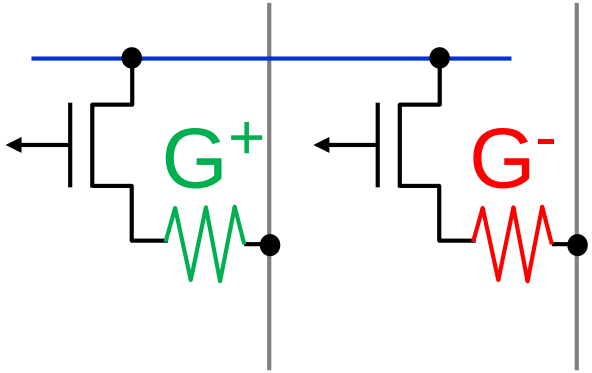
94% TEST accuracy w/ 5,000 examples

} Non-idealities in Real PCM Devices

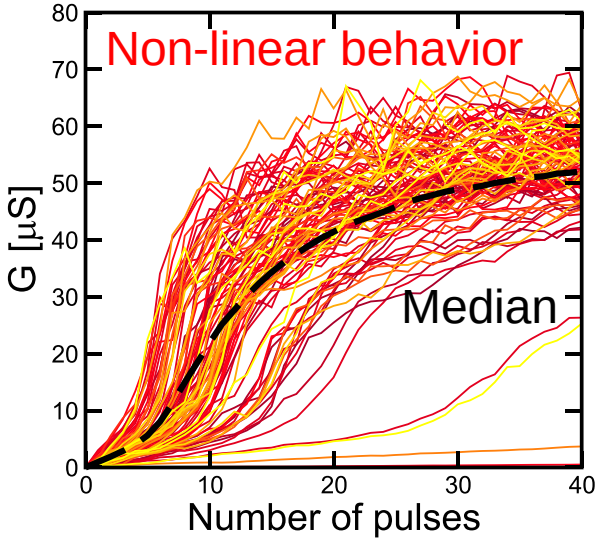


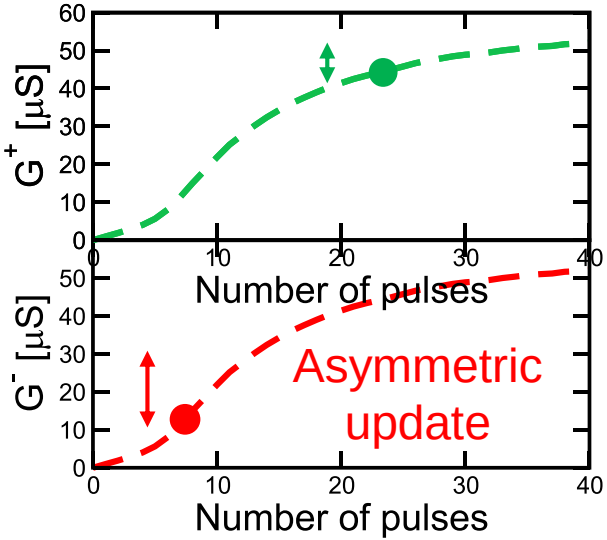
$W = G^+ - G^-$

Study: 2-PCM: Asymmetric Conductance Response



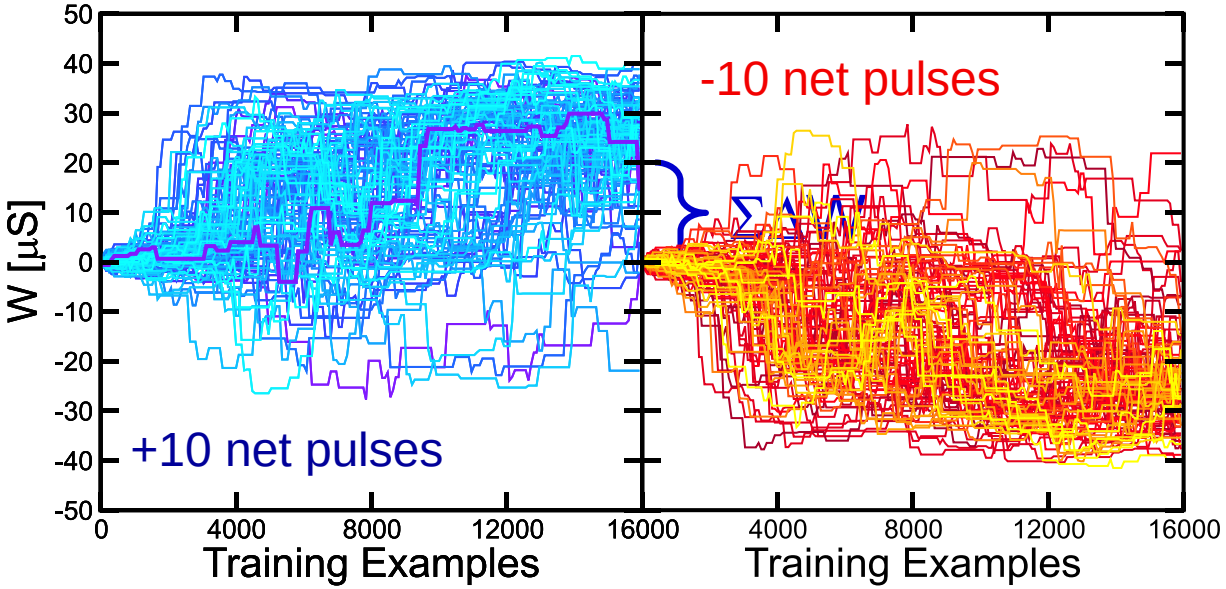
$W = G^{+} - G^{-}$

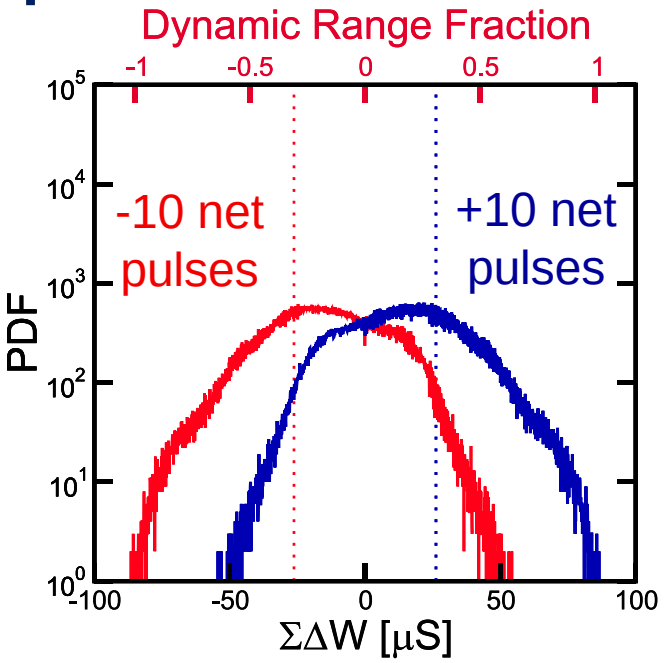




- 2-PCM unit cell is non-linear and asymmetric
- Symmetry is crucial to balance UP and DOWN steps and accurately implement open-loop weight update
- Strong impact on Neural Network training accuracy

2-PCM scheme: dependence on applied pulses





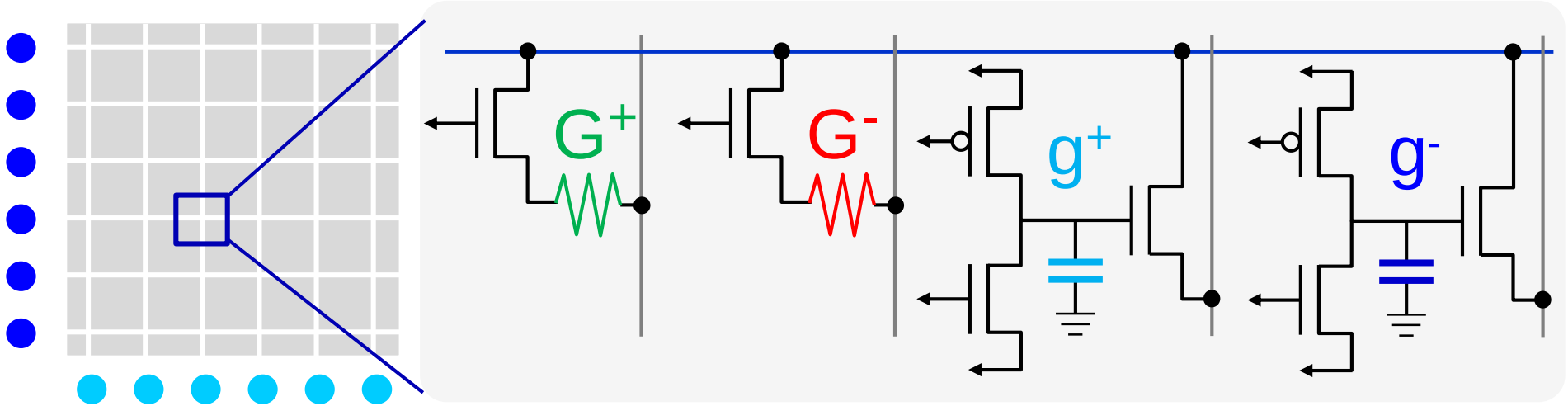
- ΣΔW distributions are overlapped, preventing a clear distinction of increase and decrease weight requests
- MNIST accuracy is lower than accuracy achieved with TensorFlow on a same size network

MNIST Accuracy
TensorFlow: 97.94%
2-PCM: 93.77%

Novel 2T2R + 3T1C unit cell

Most Significant Pair (MSP) | Least Significant Pair (LSP)

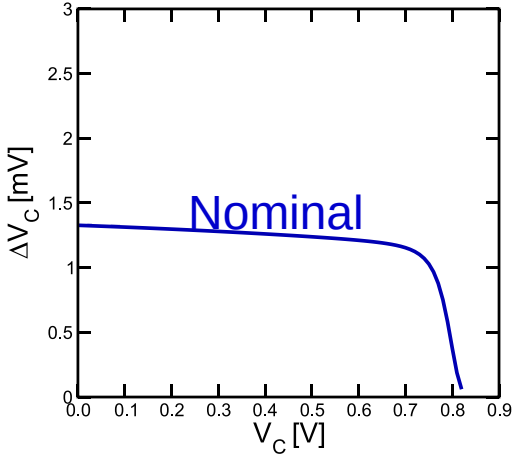
$$W = F \times (G^{+} - G^{-}) + g^{+} - g^{-}$$

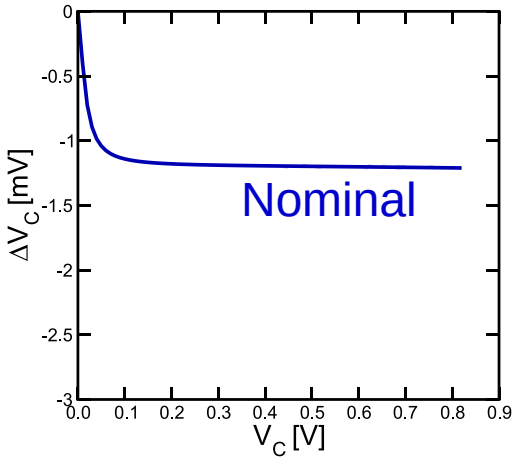


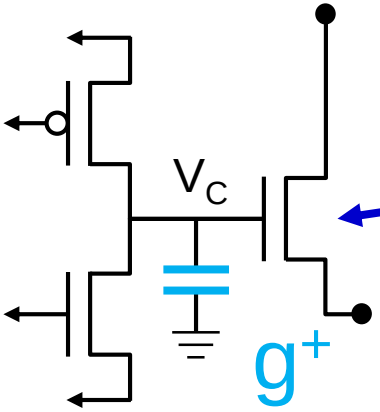
- Symmetry → Weight update performed on g+ only
  - g^{-} shared among many columns (e.g. 128 columns)
- Dynamic Range → Gain factor F (e.g. F = 3)
- Non-Volatility → Weight transferred to PCMs infrequently (every 1000s of images)

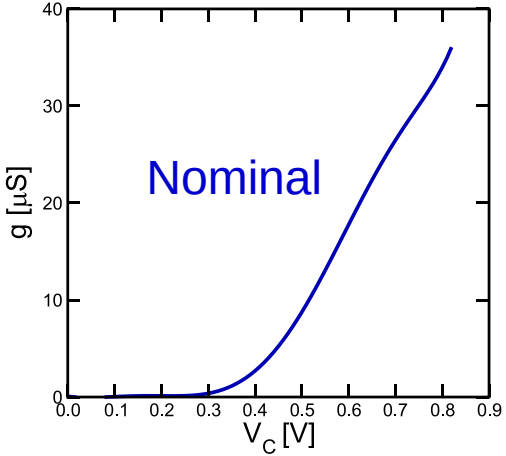
S. Ambrogio et al, *Nature*, 558, 60 (2018)

Novel unit cell: 2T2R + 3T1C, nominal behavior





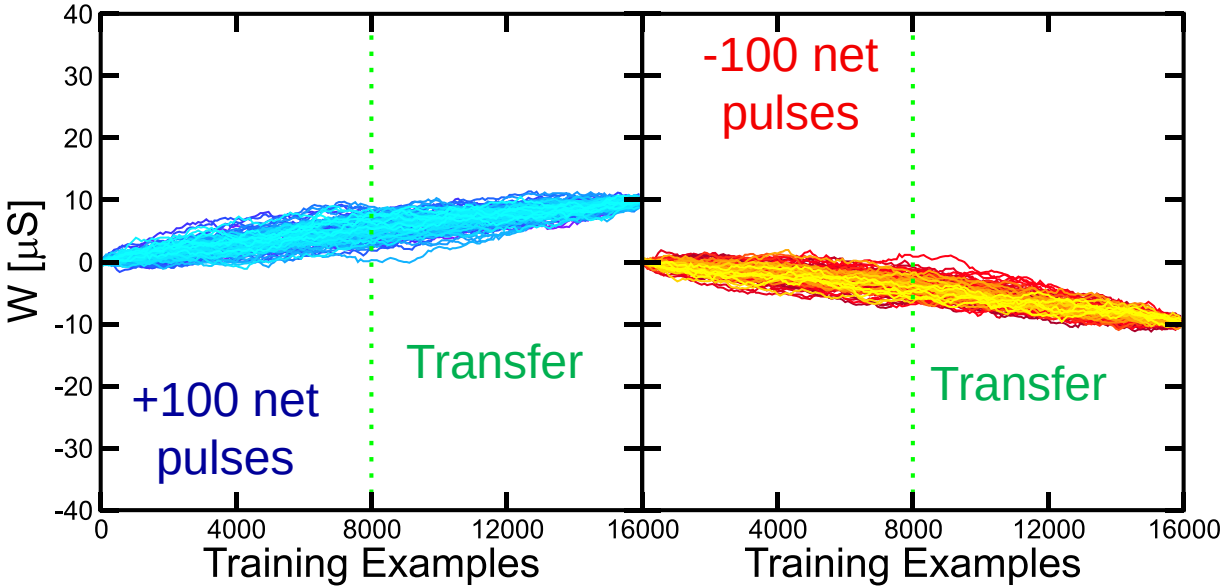


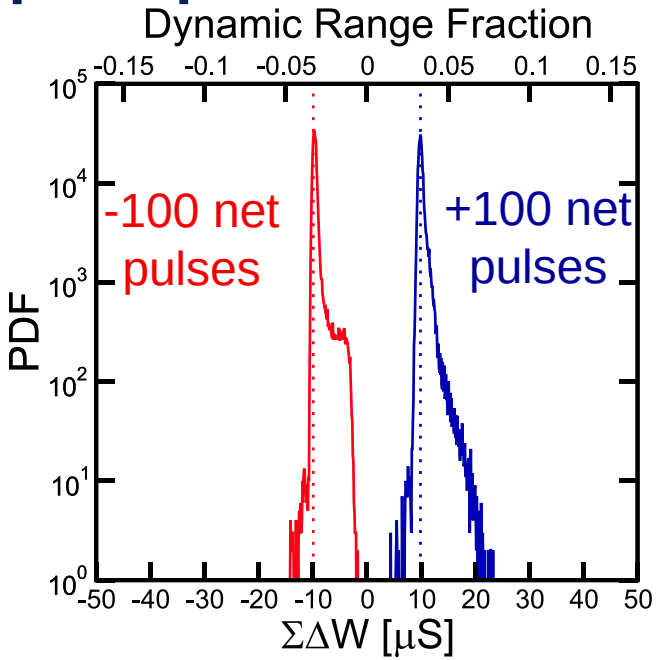


S. Ambrogio et al, *Nature*, 558, 60 (2018)

- PMOS charges the capacitor, increasing g+ and W
- NMOS discharges the capacitor, decreasing g+ and W
- Read MOS shows a linear dependence of g on V_C
- PMOS and NMOS provide the same current, balancing UP and DOWN weight updates

2T2R+3T1C scheme: dependence on applied pulses

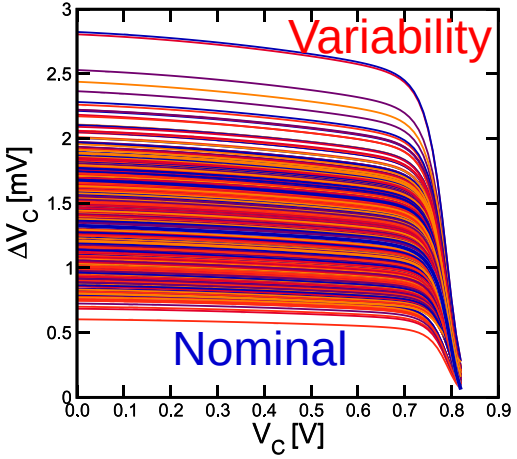


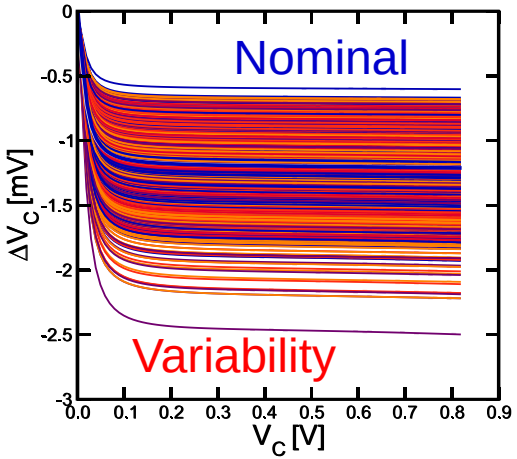


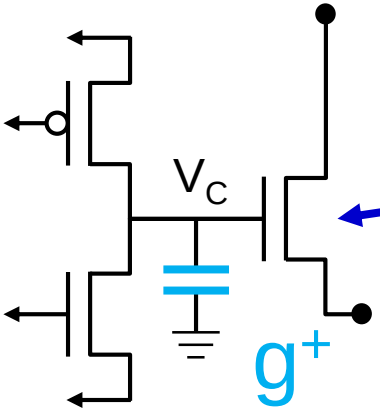
- Higher number of requested pulses due to very small g^+ update
- MNIST accuracy is equivalent to accuracy achieved with TensorFlow on a same size network

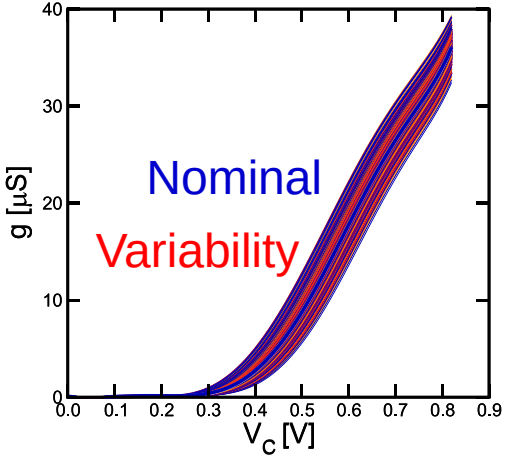
MNIST Accuracy
TensorFlow: 97.94%
2T2R+3T1C: 98.10%

Novel unit cell: 2T2R + 3T1C, CMOS variability





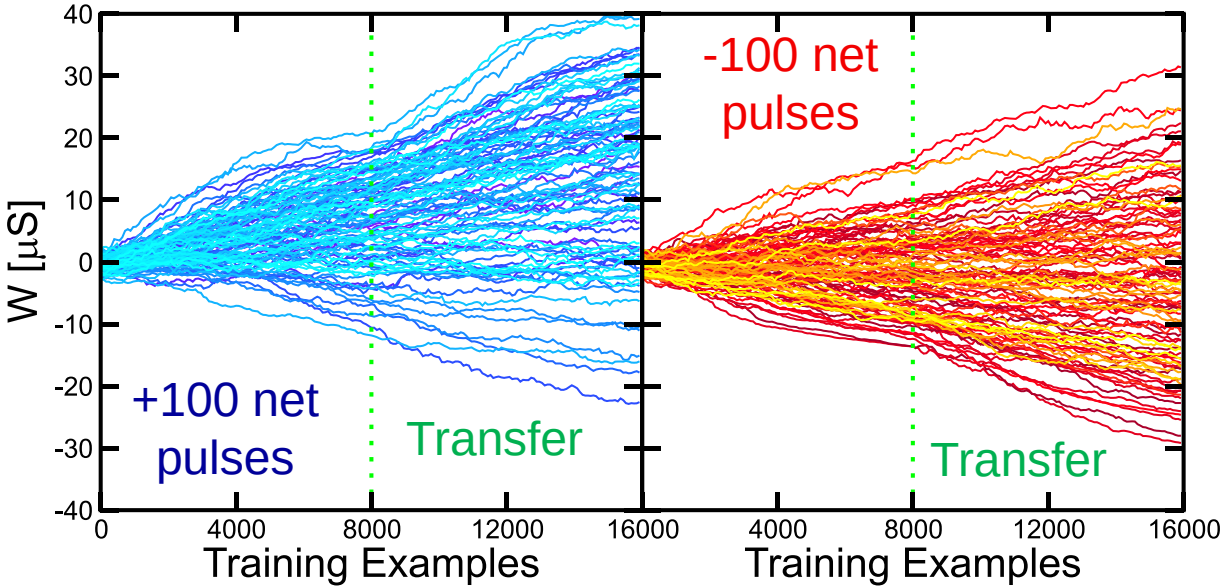


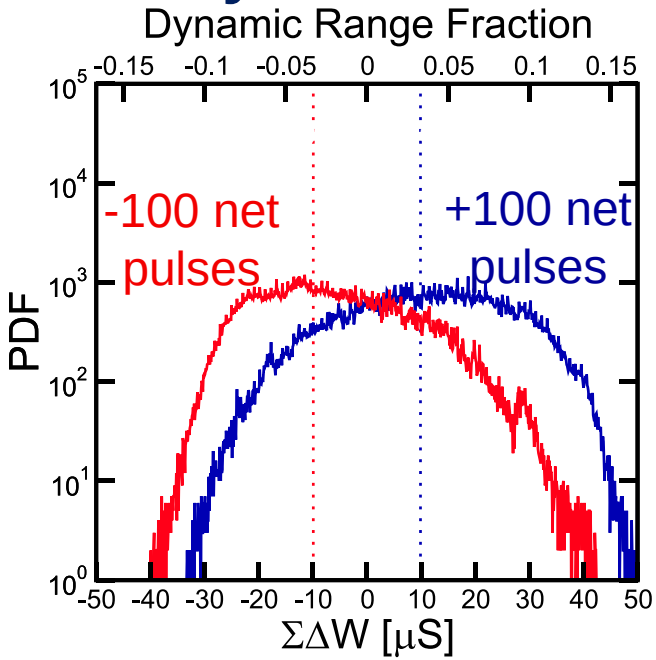


S. Ambrogio et al, *Nature*, 558, 60 (2018)

- PMOS charges the capacitor, increasing g+ and W
- NMOS discharges the capacitor, decreasing g+ and W
- Read MOS shows a linear dependence of g on V_C
- PMOS and NMOS never provide the same current, causing UP and DOWN weight updates asymmetry

2T2R+3T1C scheme: impact of CMOS variability

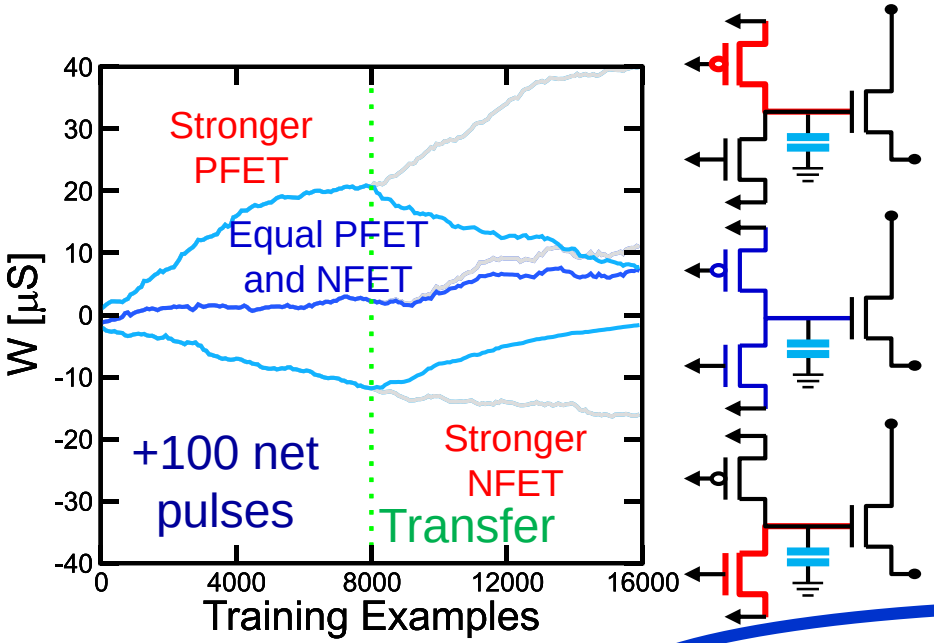


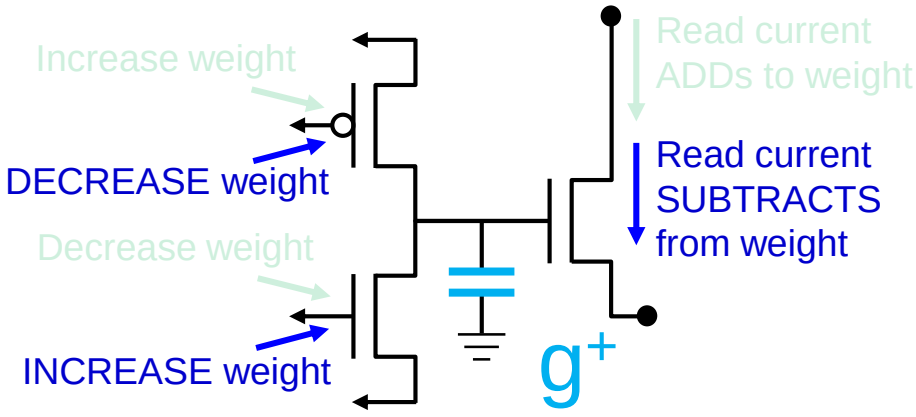


- Asymmetry in PMOS and NMOS strongly broadens ΣΔW distributions
- MNIST accuracy is highly degraded with respect to accuracy achieved with TensorFlow

MNIST Accuracy
TensorFlow: 97.94%
2T2R+3T1C: 98.10%
+Variability: 92.42%

2T2R+3T1C scheme: polarity inversion



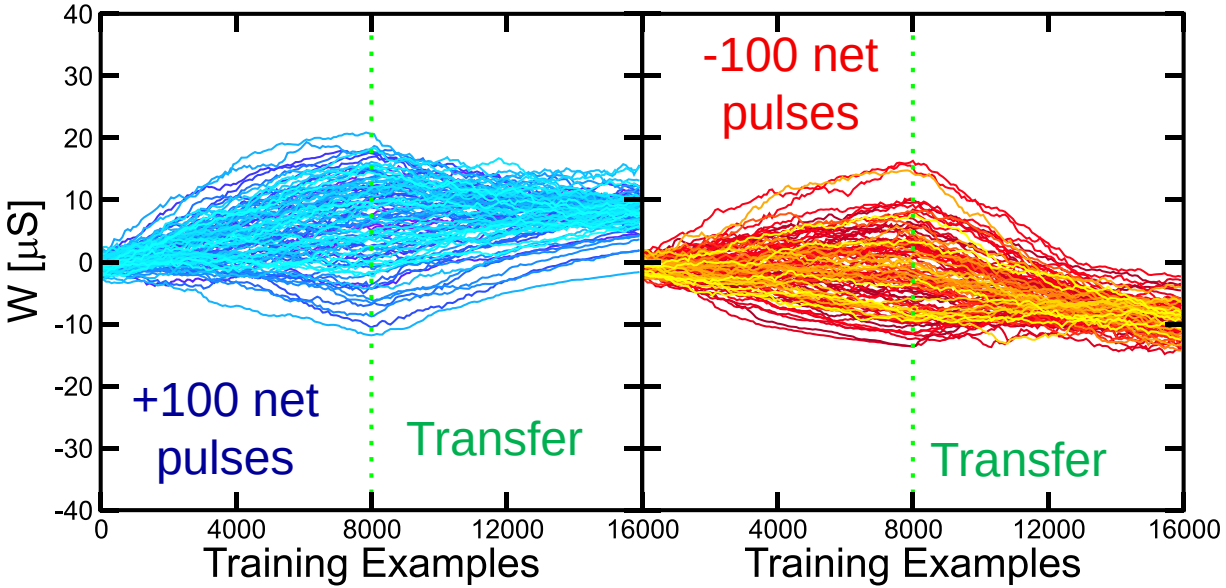


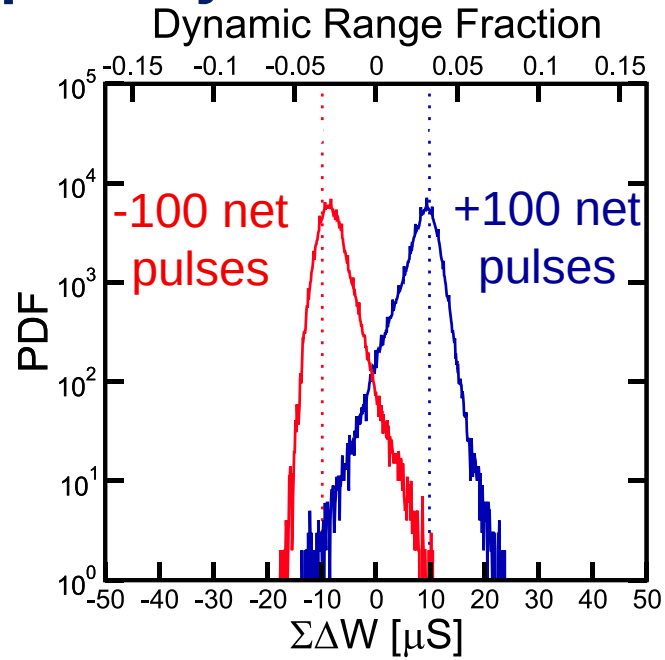
$W = F \times (G^+ - G^-) + g^+ - g^-$ | Transfer | $W = F \times (G^+ - G^-) - (g^+ - g^-)$

Polarity inversion: Invert the sign of the lower significance conductance between transfers to higher significance pair

S. Ambrogio et al, *Nature*, 558, 60 (2018)

2T2R+3T1C scheme: CMOS variability, polarity inversion

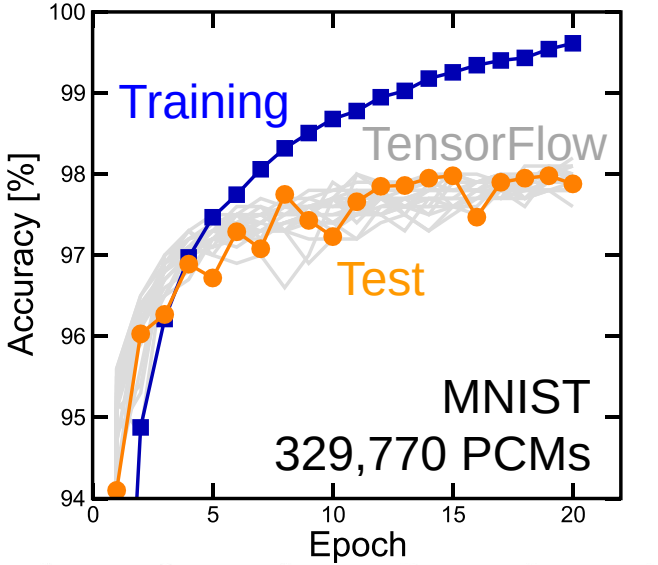


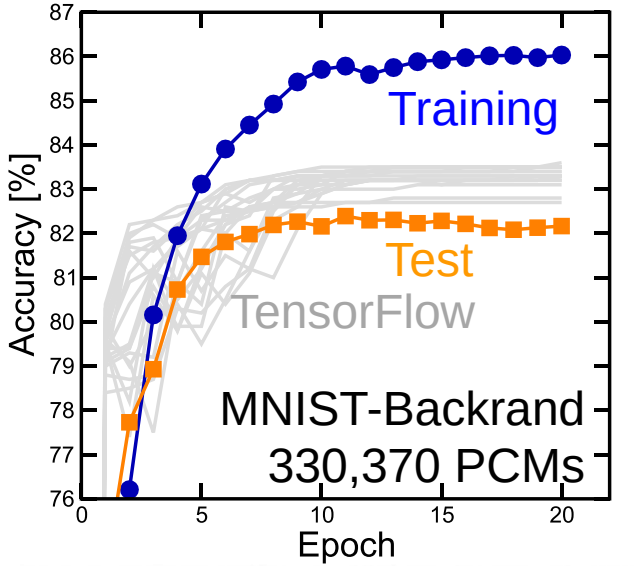


- Asymmetry in PMOS and NMOS is averaged by polarity inversion
- MNIST accuracy is equivalent to accuracy achieved with TensorFlow

MNIST Accuracy
Tensorflow: 97.94%
Polarity Inv: 97.95%

Accuracy on MNIST and MNIST backrand



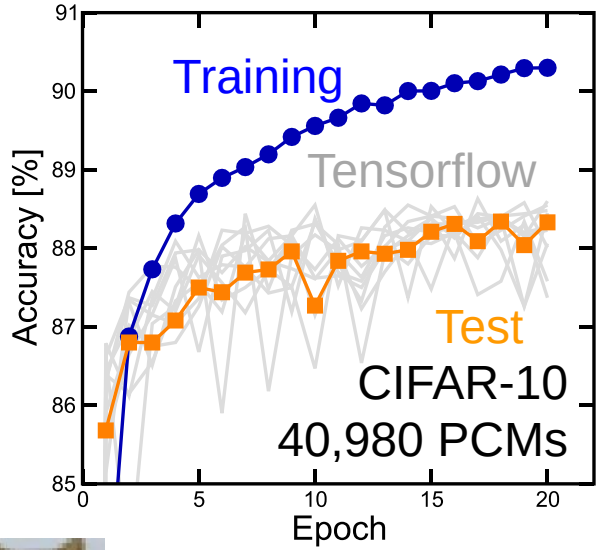


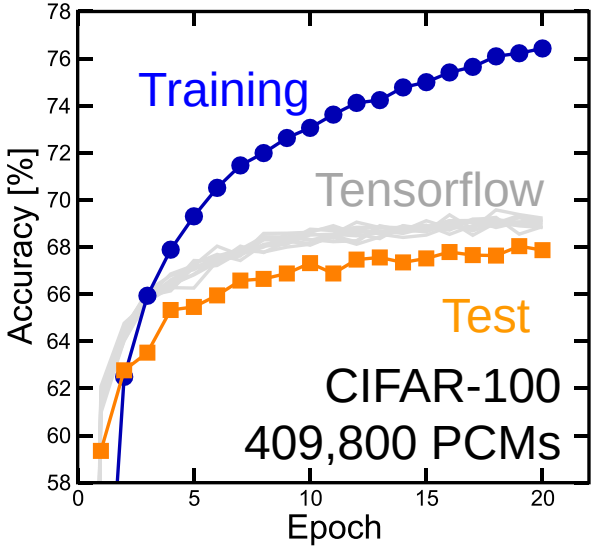
S. Ambrogio et al, *Nature*, 558, 60 (2018)

Mixed hardware-software experiment: every synaptic weight → 2 real PCM devices

Transfer learning from ImageNet to CIFAR-10/100

Mixed hardware-software experiment

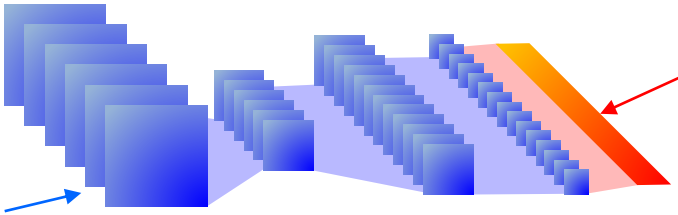




ImageNET

CIFAR-10/100

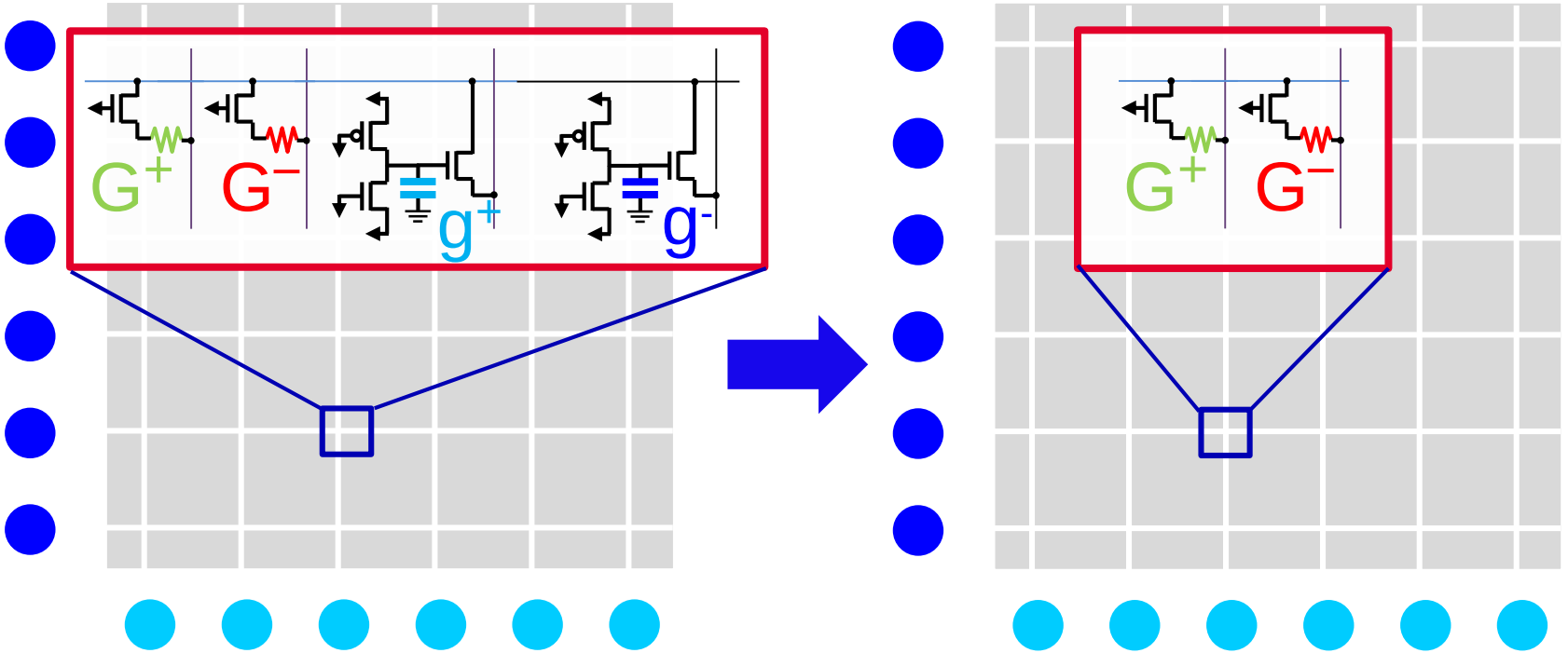
Transfer Learning: Use pre-trained, scaled weights from ImageNET for convolution layers



S. Ambrogio et al, *Nature*, 558, 60 (2018)

Full 2-Analog Memory structure

$W = F \times (G^{+} - G^{-}) + g^{+} - g^{-}$



- Single pair of devices performing the entire training

Single device requirements

- Several specifications are requested to single resistive device in order to obtain software-equivalent accuracies

- A minimum of 1000 different conductance steps are required → extremely hard to obtain

- A maximum 5% of asymmetry between up and down conductance updates

  → need for very linear and symmetric devices

Our solution → Multiple conductances of varying significance, diversification of requirements

TABLE 2 | Summary of RPU device specifications.

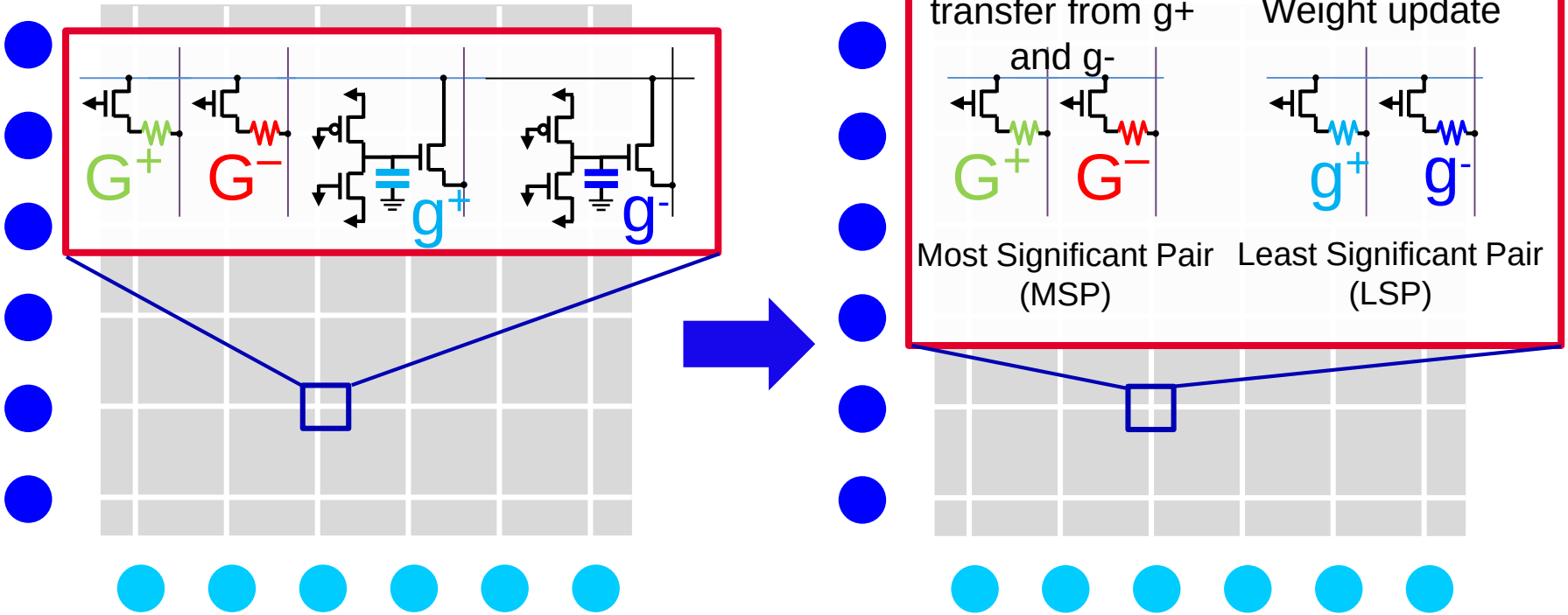
Specs	Parameter	Value	Tolerance
Pulse duration		1 *ns*	
Operating voltage	$\pm V_S$	1 *V*	
Maximum device area		0.04 μm^2	
Average device resistance	R_{device}	24 *MΩ*	7 *MΩ*
Maximum device resistance	$\max\left(g_{ij}\right)$	112 *MΩ*	7 *MΩ*
Minimum device resistance	$\min\left(g_{ij}\right)$	14 *MΩ*	7 *MΩ*
Resistance on/off ratio	$\max\left(g_{ij}\right)/\min\left(g_{ij}\right)$	8	
Resistance change at $\pm V_S$	$\Delta g^{\pm}_{min}$	100 *KΩ*	30 *KΩ*
Resistance change at $\pm V_S/2$		10 *KΩ*	
Storage capacity	$\left(\max\left(g_{ij}\right) - \min\left(g_{ij}\right)\right)/\Delta g_{min}$	1000 levels	
Device up/down asymmetry*	$\Delta g^{+}_{min}/\Delta g^{-}_{min}$	1.05	2%

*Note that these numbers are derived from the radar diagram in **Figure 4A** and correspond to the shaded area. *Global asymmetry in up/down responses can be to a large extend compensated by proper adjustment of pulse widths and/or pulse amplitude.*

T. Gokmen, Y. Vlasov, *Frontiers in neuroscience* 10, 333 (2016)

Full 4-Analog Memory structure

$W = F \times (G^{+} - G^{-}) + g^{+} - g^{-}$



- Most Significant Pair: Infrequent, **Closed Loop Programming** Operation
- Least Significant Pair: Frequent, **Open Loop Programming** Operation

Suggestions for new analog memory devices

- **Larger unit cell with two components**

1. More-significant pair of non-volatile conductances (e.g., PCM) stores “most” of the weight info

    - Non-linear conductance update → OK
    - DOES need to be able to tune these conductances rapidly in a CLOSED-LOOP manner

2. We perform all the OPEN-LOOP programming using a “less-significant” pair of conductances

    - Poor retention → OK
    - Significant device-to-device fixed variabilities → OK
    - DOES need to offer highly linear conductance update

→ **Reduces the difficulty of device requirements**

S. Ambrogio et al, *Nature*, 558, 60 (2018)
G. Cristiano et al, J. Appl. Phys. 124 (15), 151901 (2018)

Comparison of device specifications for MSP and LSP

Specifications	Parameter	MSP	LSP
Initial Step-size	ΔG_0 (ΔG_0^*)	< 21 μS (42%)	< 1.4 μS (2.8%)
Intra-device Variability	σ_{intra}	< 1.5 μS	< 0.8 μS
Inter-device Variability	σ_{Gmax}	< 10 μS	< 12 μS
	$\sigma^*_{\Delta G0}$	< 200%	< 95%
Faulty devices	Dead C.R.	< 7%	< 7%
	Stuck On C.R.	< 35%	< 10%
Dynamic range	Number of levels	> 13	> 110
Retention	Time before data loss	Higher	Lower
Endurance	Number of Set/Reset	Lower	Higher

Perspective on Training Fully Connected Networks with Resistive Memories: Device Requirements for Multiple Conductances of Varying Significance

Giorgio Cristiano,[1,2] Massimo Giordano,[1,2] Stefano Ambrogio,[1] Louis P. Romero,[1] Christina Cheng,[1] Pritish Narayanan,[1] Hsinyu Tsai,[1] Robert M. Shelby,[1] and Geoffrey W. Burr[1,a)]
[1)] *IBM Research AI, IBM Research–Almaden, 650 Harry Road, San Jose, CA USA 95120*
[2)] *EPFL, Route Cantonale, 1015 Lausanne, Switzerland*

G. Cristiano et al, J. Appl. Phys. 124 (15), 151901 (2018)

Outline

- Introduction
- A brain-inspired algorithm: **Spike Timing Dependent Plasticity**
- A machine learning algorithm: **Back-Propagation**
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Long-term: maximizing the future business case (vs. a GPU)

Low Power

(inherent in the physics, but possible to lose in the engineering…)

Still of interest for power-constrained situations: learning-in-cars, etc.

Accuracy

(essential that final Deep-NN performance be indistinguishable from GPUs –hardest technical challenge)

Sweet spot: rather than buy GPUs, people buy this chip instead for training of Deep-NN's

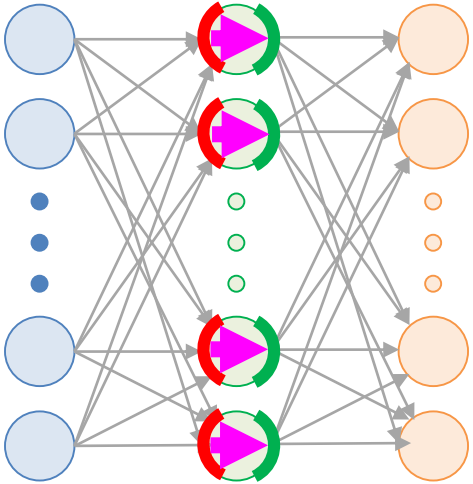
Still of interest for some situations: learning-in-server-room

(circuitry must be massively parallel)

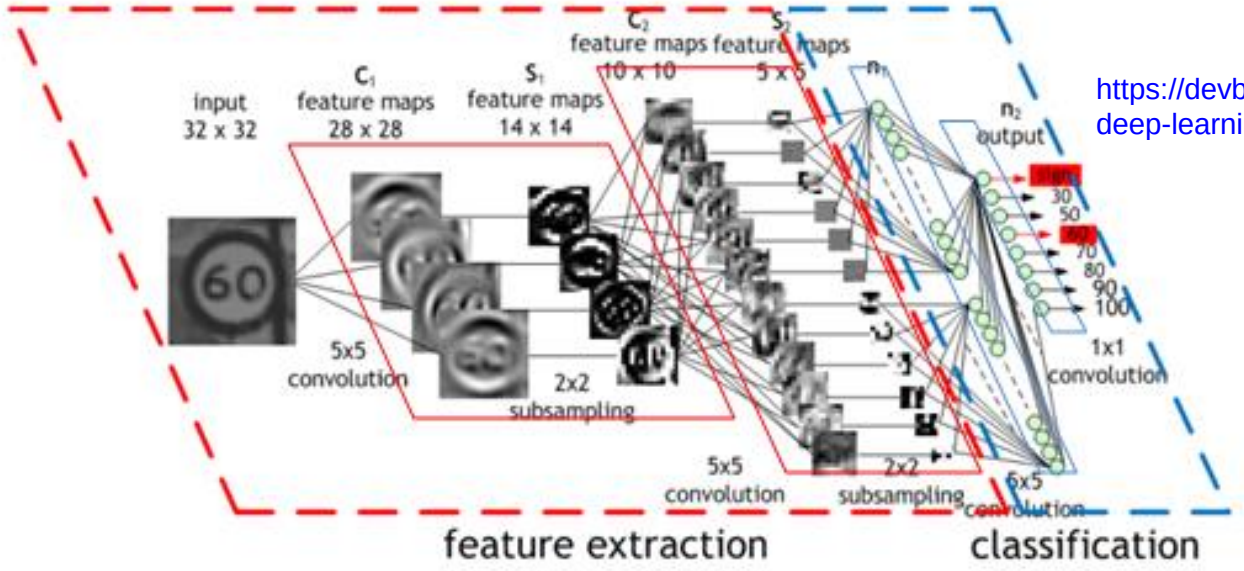
Faster

Suggestions from circuit design work

1) Parallelism is key
2) Avoiding ADC (Analog-to-Digital Conversion) saves time, power and area
3) Do the necessary computations (squashing functions) but be as “approximate” as you can (get away with)
4) Need to get vectors of data from the bottom of one array to the edge of the next one
5) Digital accelerators are at their best w/ **convolutional** layers; Analog-memory accelerators are at their best w/ **fully-connected** layers.



Impact on Convolutional Neural Networks



https://devblogs.nvidia.com/parallelforall/deep-learning-nutshell-core-concepts/

- Only the last layers in a Convolutional Neural Network are Fully Connected due to memory constraints
- Hardware accelerators could easily implement FC layers, what could be the impact on CNN topology and performance?

Outline

- Introduction
- A brain-inspired algorithm: **Spike Timing Dependent Plasticity**
- A machine learning algorithm: **Back-Propagation**
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion

Conclusion

- **AI is introducing novel tools to develop solutions to everyday challenges**
  - Brain Inspired approach
  - Deep Learning approach
- **NVM-based crossbar arrays can accelerate the training of Deep Machine Learning compared to GPU-based training**
  - Multiply-accumulate performed at the data
  - Possible 500x speedup and orders-of-magnitude lower power
- **Experimental results on a 2T2R+3T1C unit cell demonstrate software-equivalent training accuracy**
  - MNIST, MNIST-backrand, CIFAR-10 and CIFAR-100 tested
- **Need area-efficient peripheral circuitry**
  - Tradeoffs balancing simplicity and area-efficiency against impact on ANN performance

stefano.ambrogio@ibm.com