July 16 2016

Using conversion algorithm to compensate errors in analog computing via nano-crossbar

Miao Hu, John Paul Strachan, Stanley R. Williams
Miao.hu@hpe.com
Hewlett Packard Enterprise Labs

Outline

–Motivation and concept

–Design challenges

–Devices

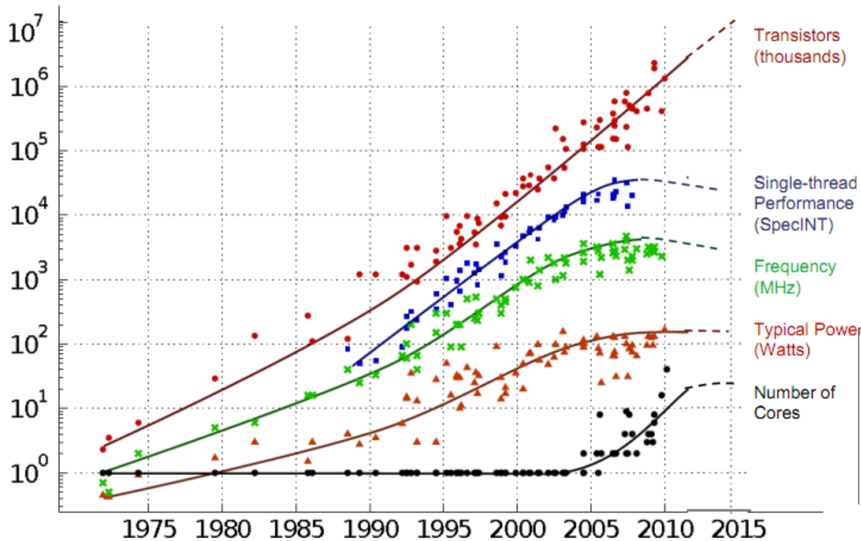
–Solution with the conversion algorithm

–Result

–Conclusion

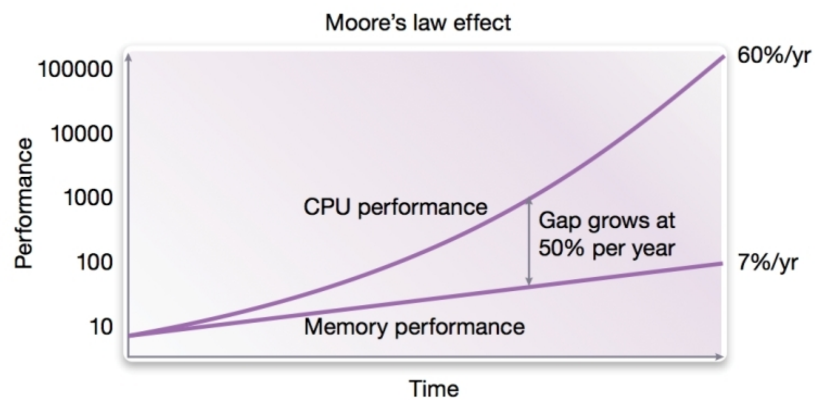
Von-Neumann machines reach its bottleneck

The end of Dennard scaling for general-purpose CMOS



Original data collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond and C. Batten
Dotted line extrapolations by C. Moore

Von-Neumann Bottleneck

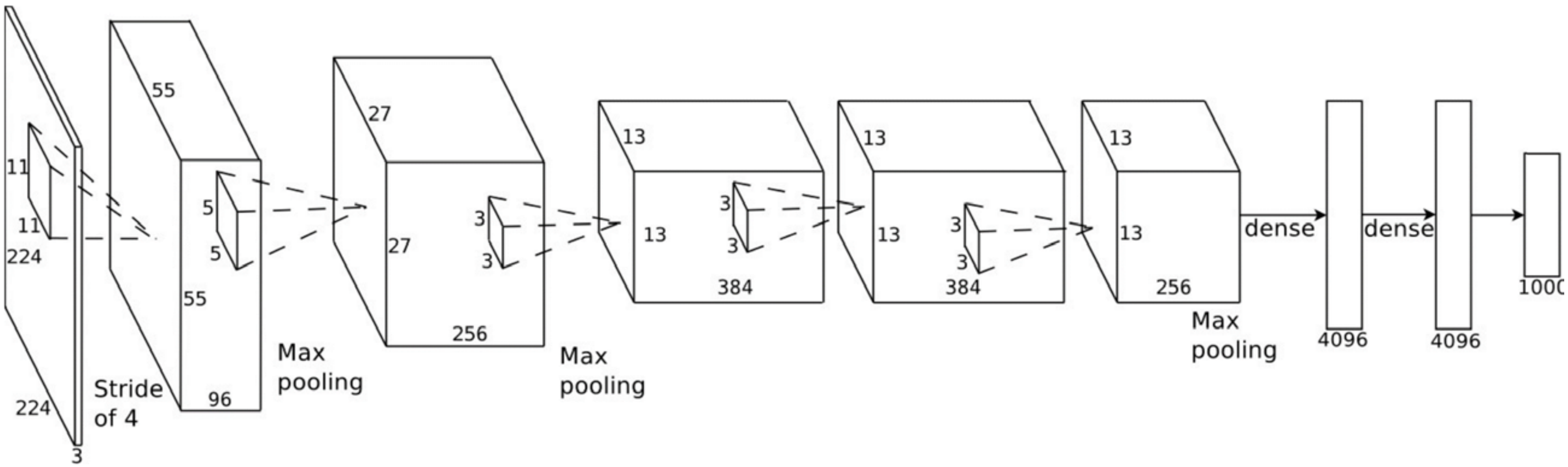


Michael Byrne, "Memory Is Holding Up the Moore's Law Progression of Processing Power", 2014.

Compute with a system of efficient SoCs and accelerators having computing in/near memory features

Important applications with high computing complexity but low computing accuracy

–Image classification in Deep Learning neural network



Opportunities for DPE

1) 70-90% of computation time consumed in the Convolution layers [1]
2) Recent work shows that only 10-12 bit representations required to maintain state-of-the-art classification accuracy [2]

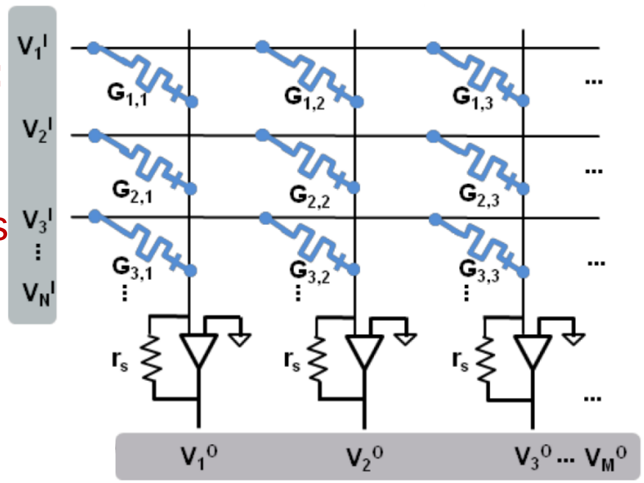
[1] *F.* Abuzaid, et al., “Caffe con Troll: Shallow Ideas to Speed Up Deep Learning” arXiv:1504.04343 [cs.LG]
[2] *M. Courbariaux, J.P. David, Y. Bengio “Low precision storage for deep learning” ICLR 2015*

Concept of Dot-Product Engine with memristor crossbar

– Memristor crossbar as computing memory

Input 2: Arrary of conductances $\{G_{ij}\}$

Input 1: Vector of voltages $\{V_i^I\}$



Output: Vector of currents $\{I_i^O\}$

Ideally: $I_j^O = \sum_j G_{ij} \cdot V_i^I$

- ❑ **Crossbar array naturally represents a matrix**
- ❑ **Compute dot product through Ohm's Law**
- ❑ **Highly parallel multiply & accumulate – favorable scaling with array size**

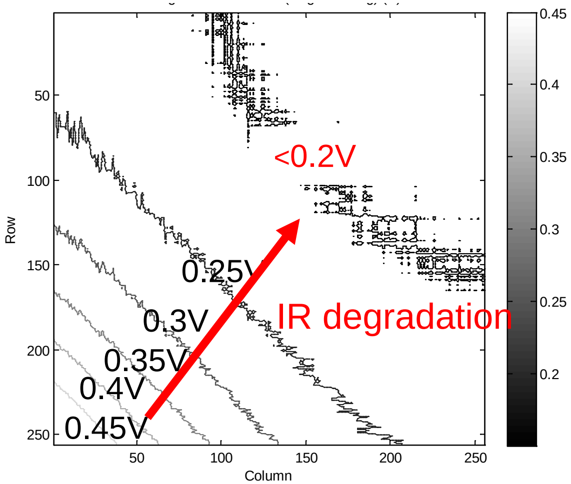
However…
Many challenges to implement!

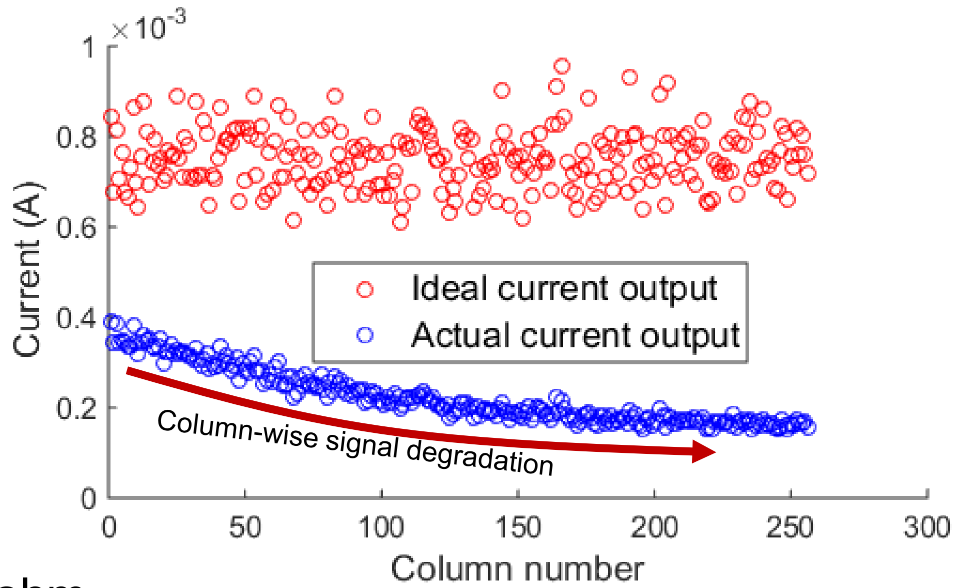
Problem

–In a real crossbar, $\mathbf{I_j^O} \neq \sum_j \mathbf{G_{ij}} \cdot \mathbf{V_i^I}$

–Because of nonlinear device resistance, input/output resistance, wire resistance, temperature and etc...

Device voltage (All 0.5V input)





Input/output resistance: 100 ohm
Wire segment resistance: 10 ohm
Device resistance: 50k ohm to 1M ohm

Challenges to implement DPE with crossbar array

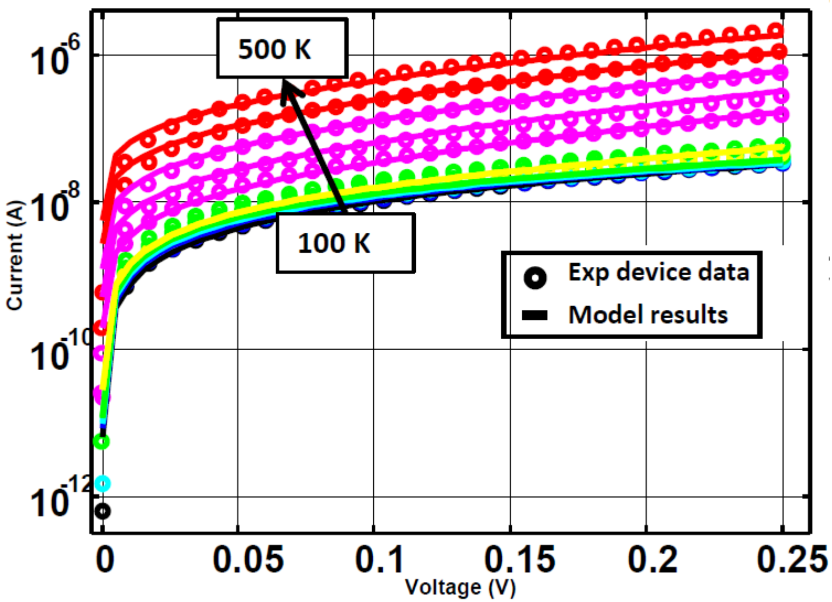
–A realistic DPE needs to address following challenges:

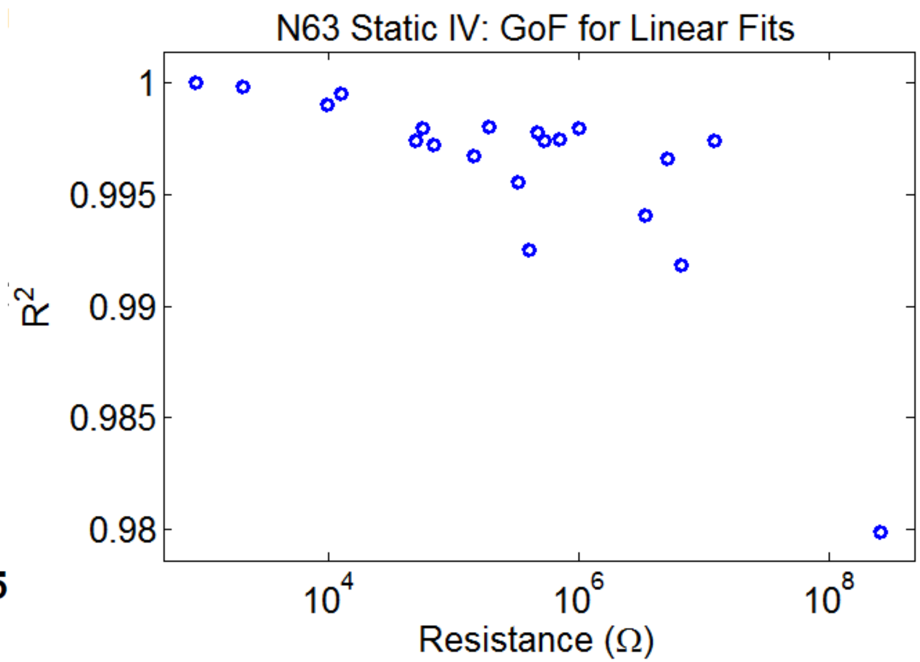
1. A stable and programmable analog device with linear resistance
2. A Transistor-like selector
3. An analog programming scheme
4. An algorithm to mapping mathematic variables with circuit parameters
5. Target applications with fixed matrix values

TaOx memristor device with linear static resistance

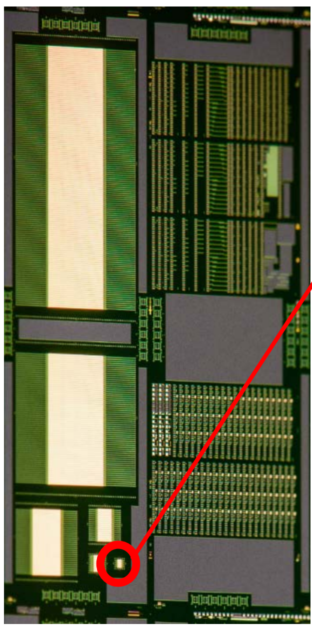
–Good linearity of stats up to 10 MΩ

–Good linearity of states up to +/- 0.3V



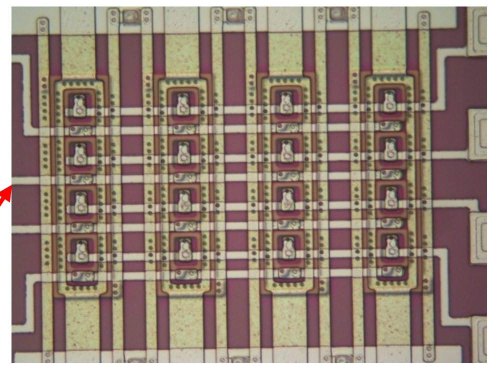


1T1R crossbar array with linear and stable analog states

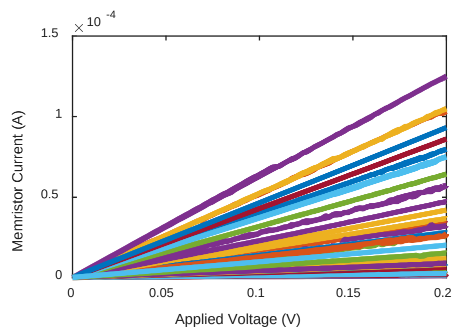


Actual wafer image
(Richard Lewington)

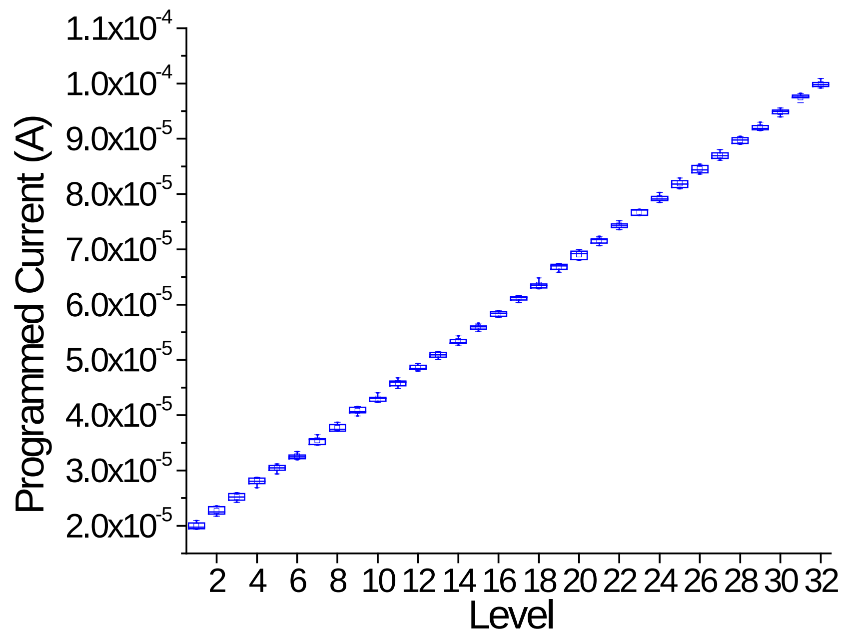
4x4 1T1R array



IV DC curves

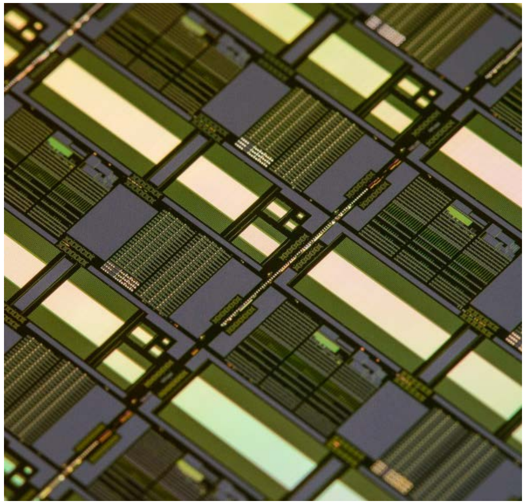


Use feedback programming algorithm to acheieve 32 levels with 1uA tolerance



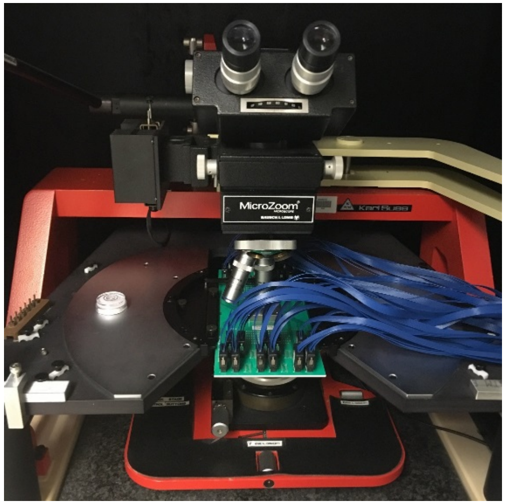
Programming signal: B1530
Current read: B1500 SMUs

DPE demonstrator – parallel array tester



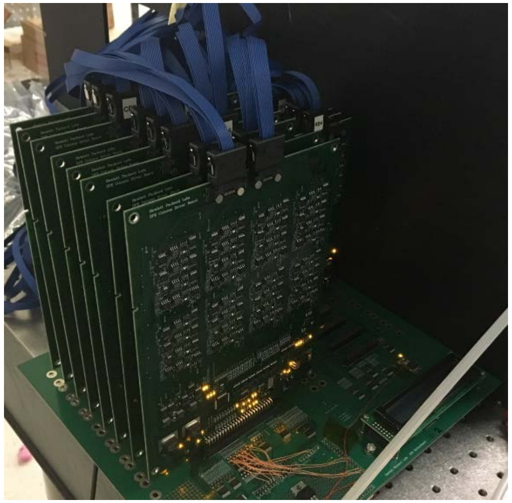
1T1M crossbar wafer
Array size:

4×4 to 128×64



Pin pad:

Maximum 260 pins:
- 128 row pins
- 64 column pins
- 64 selector pins
- 4 ground pins



DPE boards:

Parallel signal support:
Functions:
- Dot-product operation
- Single/Multi device read/write
- Pulse width > 160 ns
- Voltage: -10 to +10

Summary of the 1T1R devices

– - What is the status of achieving linear repeatable response, low power, sufficiently long retention, fast writes, sufficiently distinguishable resistances in different states, and long write endurance in one nanoscale device?

– - Is the access device issue solved? What are the remaining issues?

– For Dot-product Engine 1T1R devices:
  - Linear repeatable response: Good enough < 0.3V
  - Long retention: days and weeks, and it's overall stable.
  - Fast writes: < 100 ns
  - Distinguishable resistances: 5~6 bits
  - Long write endurance: > 10^8
  - Access device: transistor is the best solution so far.
    - Problem of existing selectors for analog computing:
      - Variation & stability
      - Nonlinear ON state
      - Yield
      - Require high read voltage for computing

Challenges to implement DPE with crossbar array

–A realistic DPE needs to address following challenges:

1. A stable and programmable analog device with linear resistance: ✓
2. A Transistor-like selector: use transistor ✓
3. An analog programming scheme: transistor-assisted close-loop tuning ✓
4. Target applications with fixed matrix values: DFT, Deep networks ✓

5. An algorithm to mapping mathematic variables with circuit parameters ?

Conversion algorithm – Basic idea

–Find $\mathbf{G}_{\mathbf{new}}$ satisfy the following equation:

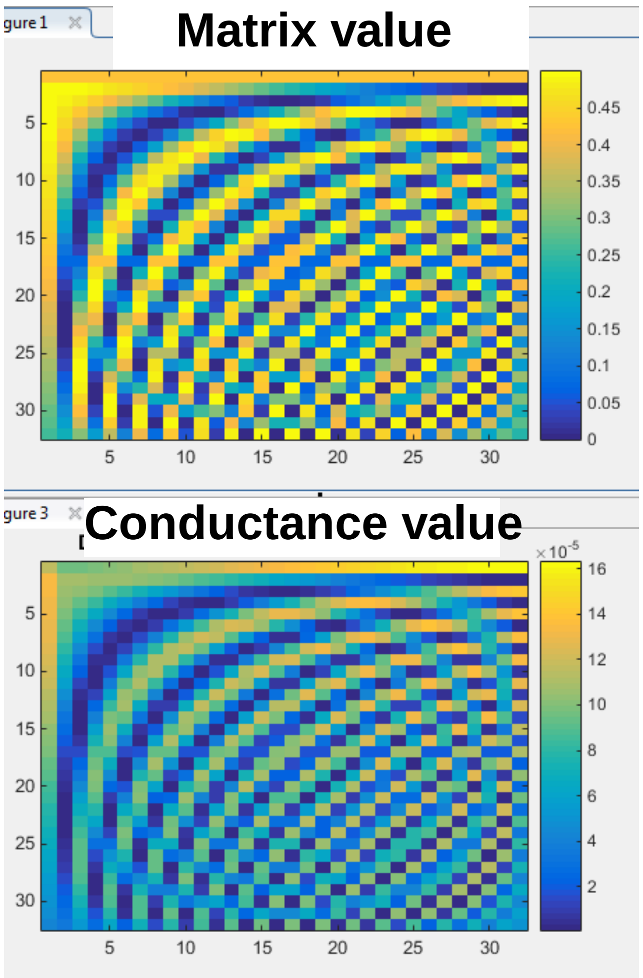
For arbitrary $\mathbf{V}_{\mathbf{in}}$, **Vin * G** ≈ crossbar($\mathbf{G}_{\mathbf{new}}$, **Vin**, etc..)

–**Benefits:**

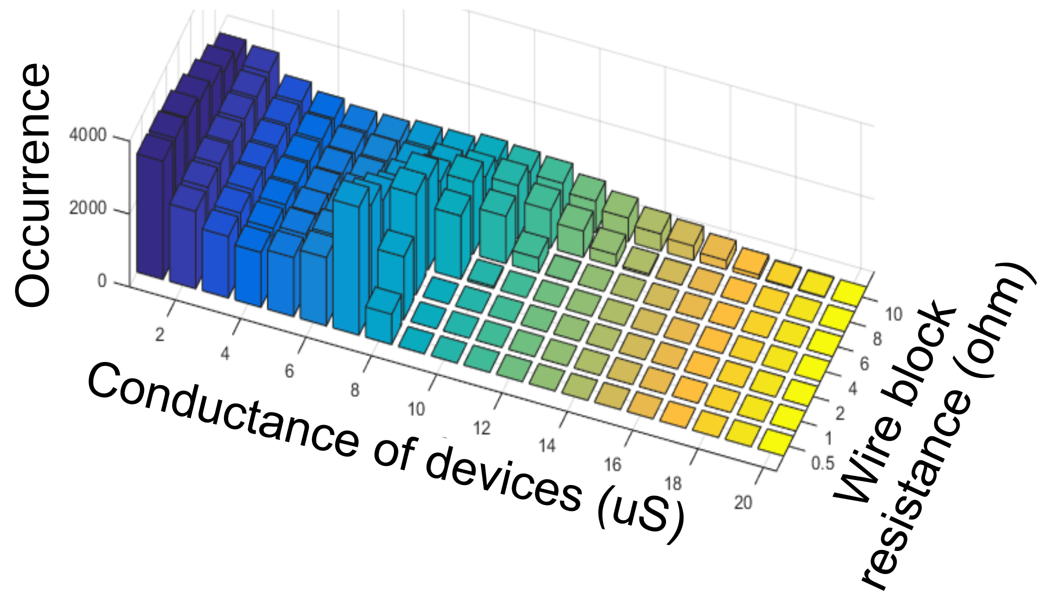
- –Minimize circuit cost and programming cost
- –Can tolerate most circuit issues
- –Can apply to general matrix.

Result of conversion algorithm

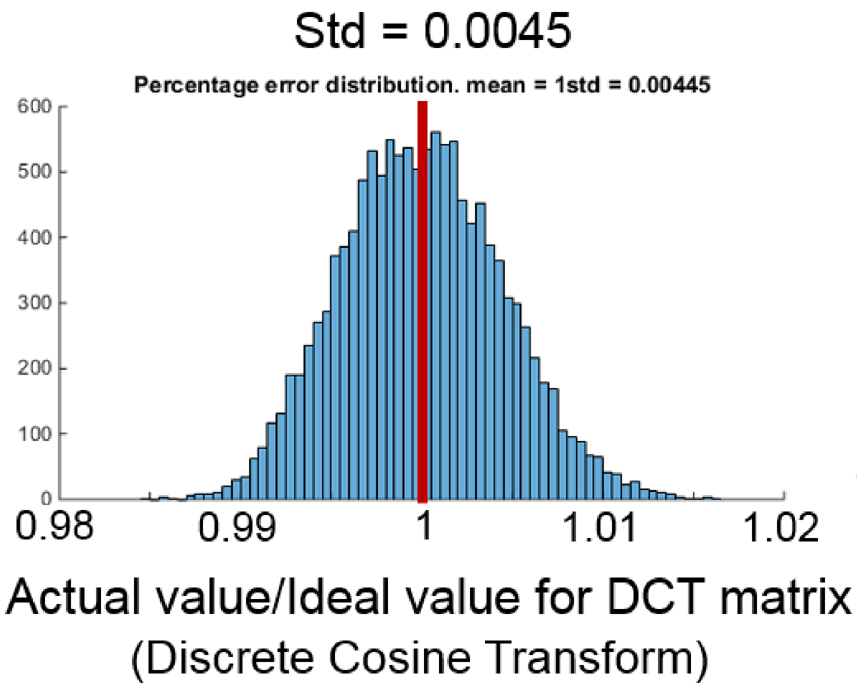
–Use Discrete Cosine Transform as example:

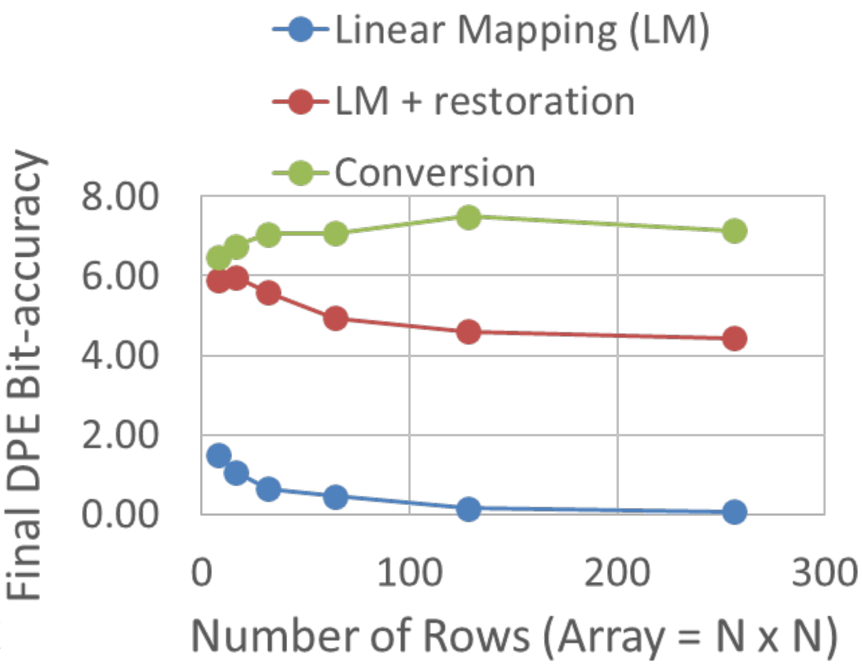


Conversion algorithm tunes device conductance to compensate wire resistance, sneak current, device nonlinearity and yield.



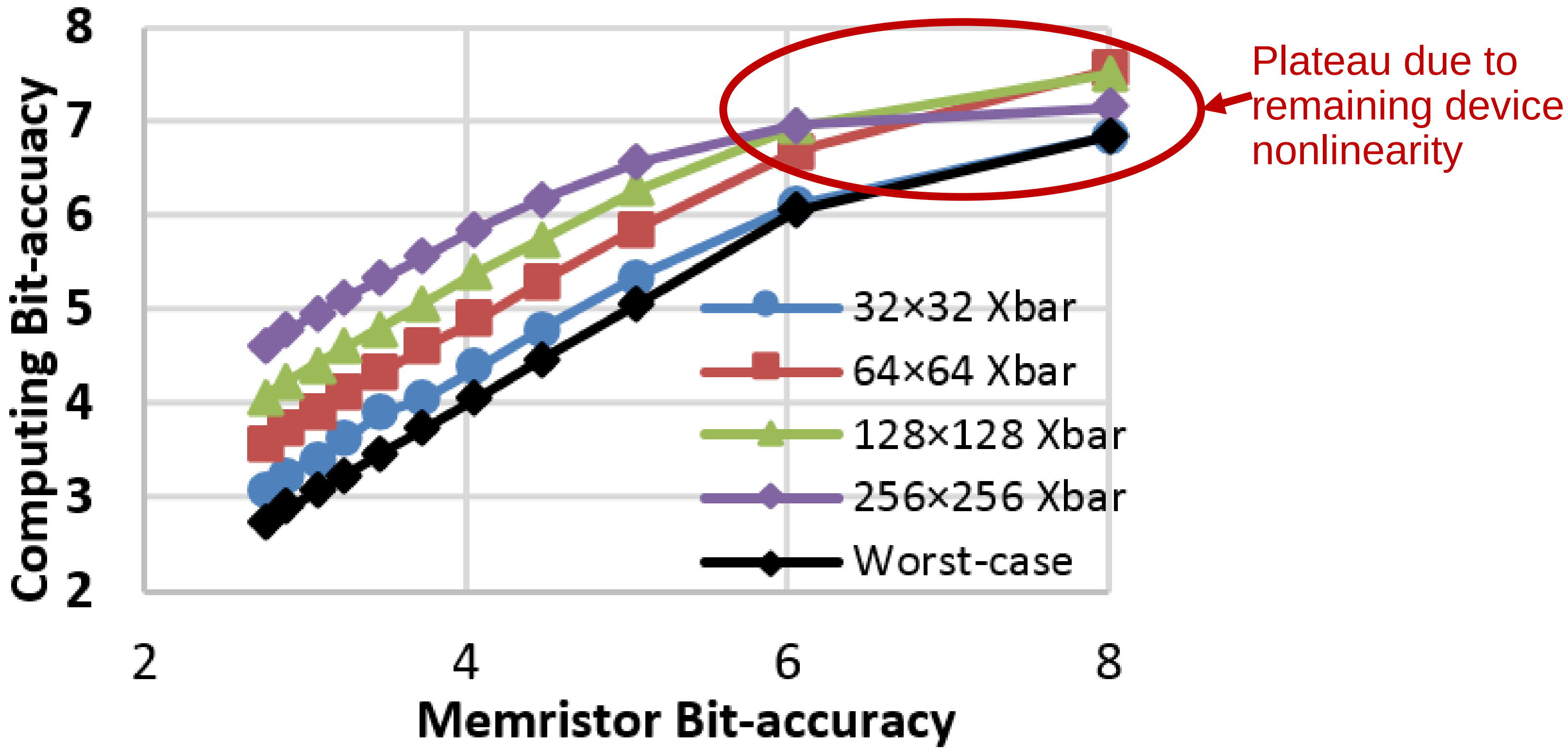
DPE computing accuracy with the conversion algorithm





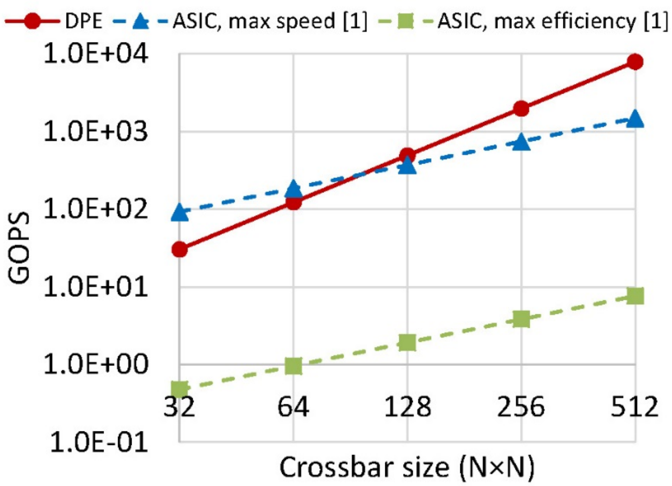
* LM + restoration assume complex restoration circuit can be afforded for the specific matrix, which is not practical.

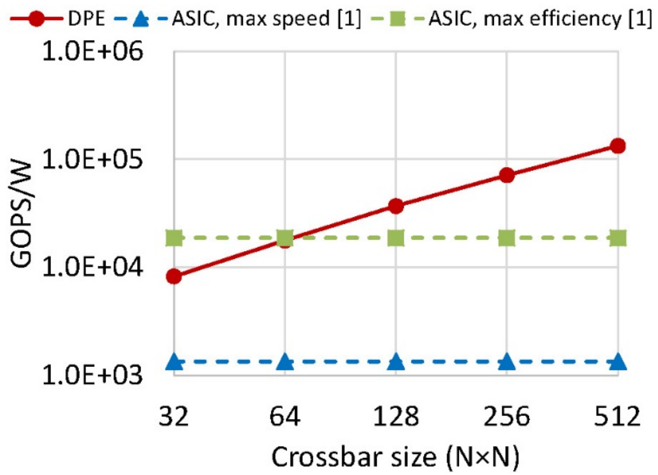
DPE computing accuracy vs. Memristor accuracy



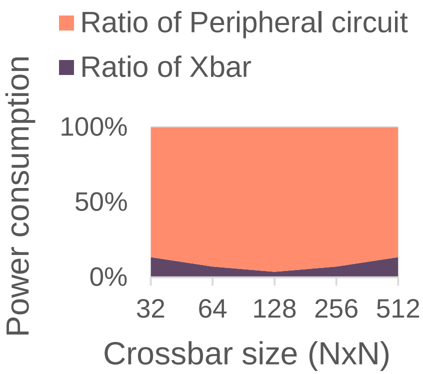
Energy efficiency and speed estimation

- Circuit assumptions: Pdiss = 100 uW per channel, fs = 10MHz, B = 8-bit resolution
  - FoM: **P = 2^B*fs = 2.56e9; F = P/Pdiss = 2.56e13, can be achieved since 2005.**
    - Le, Bin, et al. "Analog-to-digital converters." *Signal Processing Magazine, IEEE* 22.6 (2005): 69-77





>90% energy is consumed by peripheral circuits

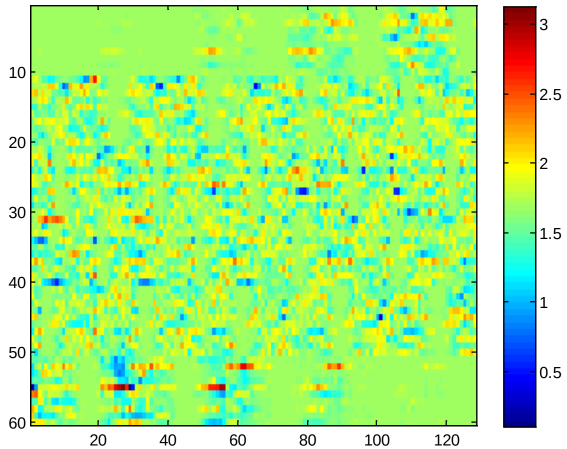


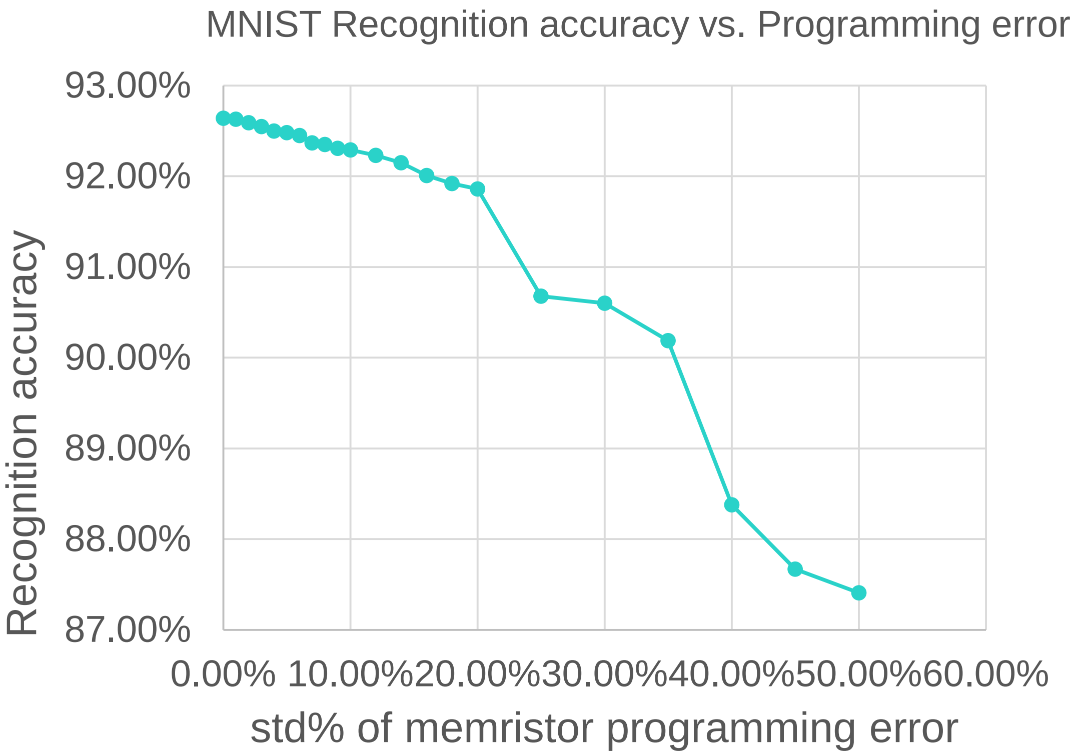
[1] S. K. Hsu, et al. "A 280 mV-to-1.1 V 256b reconfigurable SIMD vector permutation engine with 2-dimensional shuffle in 22 nm tri-gate CMOS.Solid-State Circuits," IEEE Journal of, 48(1), 118-127.

Softmax neural network on crossbar

–Very slight performance degradation even with large device error

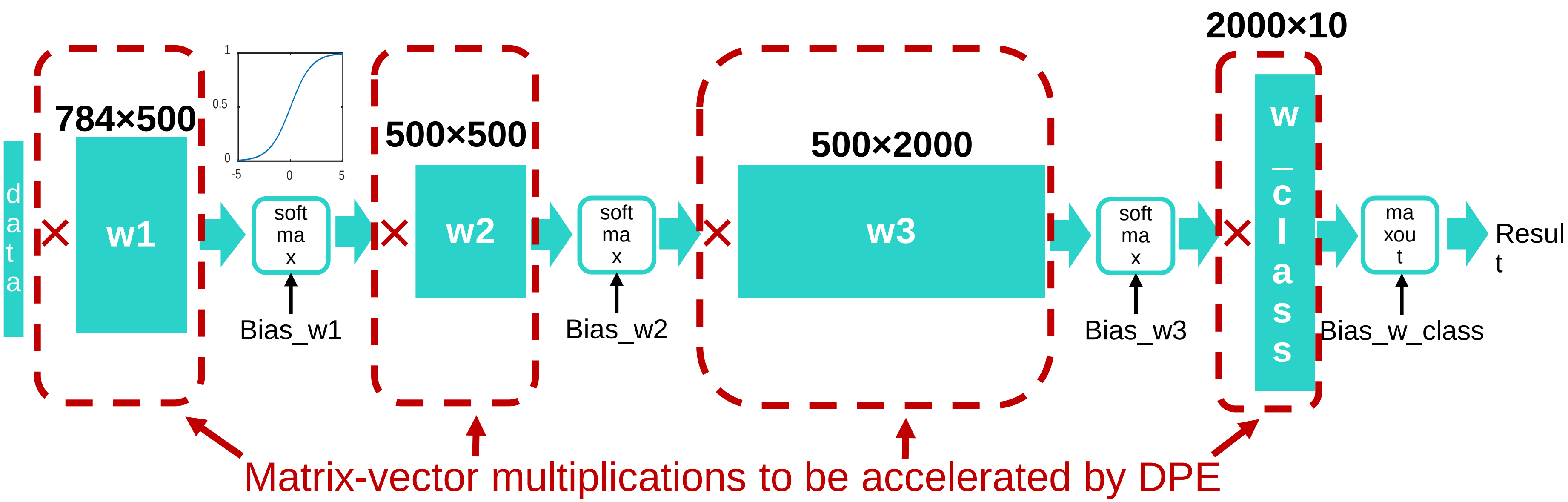
–Because of well-trained weight matrix.





A deep nonlinear encoder network for MNIST (Salakhutdinov and Hinton, AI-Stats 2007)

– **~1% error**, 100 misclassification in 10k test samples.

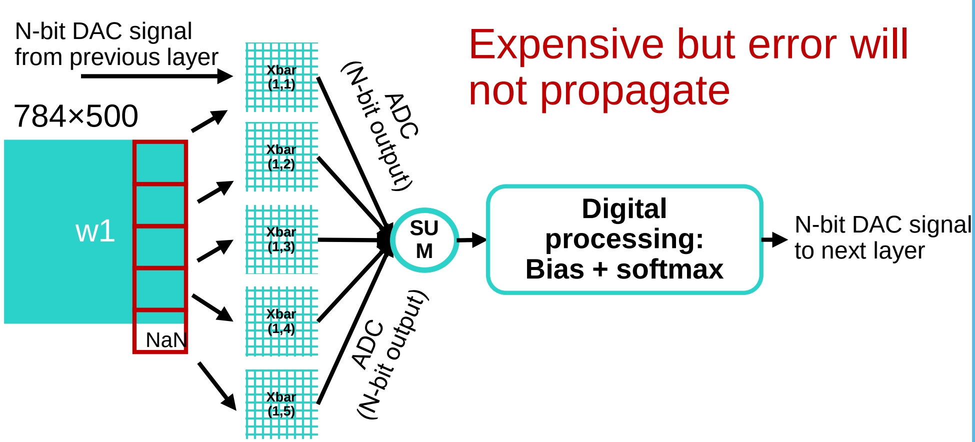


Apply DPE in machine learning

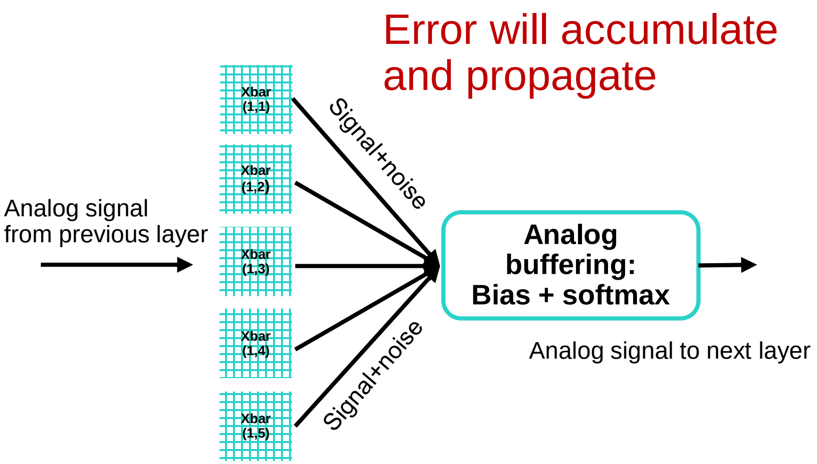
–Partition the matrix with 128×128 crossbars

–Two approaches:

full DAC+ADC support (use DPE core)



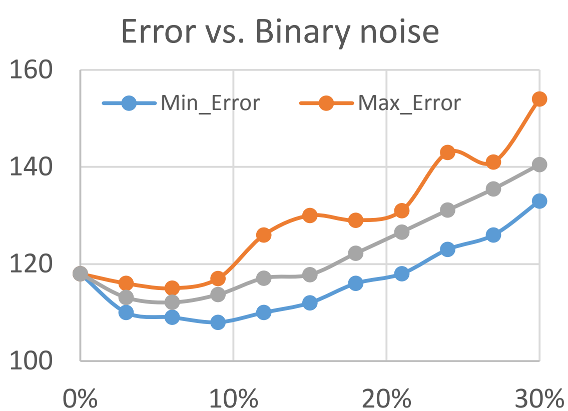
Pure analog implementation



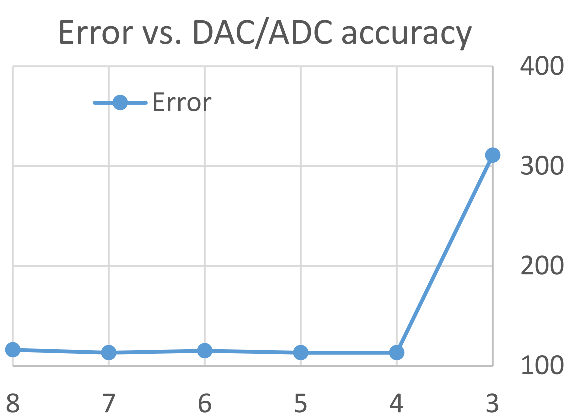
Tune memristor to very resistive state (like 100 M) to match NaN value

Result comparsion

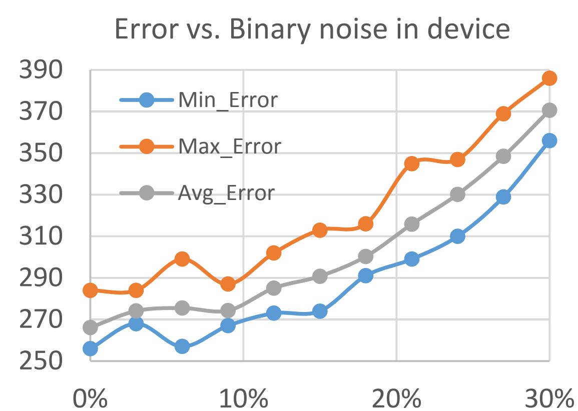
- Performance, tests are repeated 10 times for each noise setting:
  - **1.18%** error for software,
  - **1.13%** error for DPE with 4-bit DAC+ADC
  - **3.70%** error for DPE with pure analog implementation.
- RTN(Binary noise) degrades the system accuracy up to ~0% (DAC+ADC) or ~1% (Pure Analog)
- Pure analog implementation hits the accuracy by 1.48%



Binary noise in weight matrix



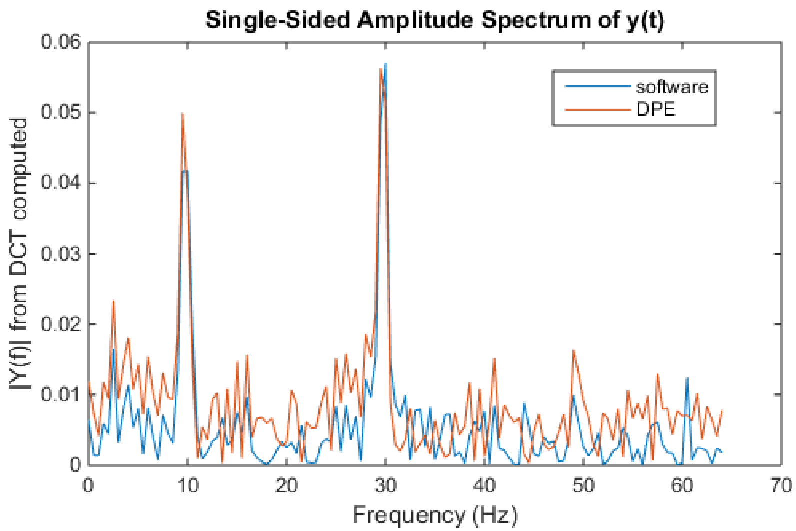
Bit-accuracy of DPE



Binary noise in device conductance (Analog implementation)

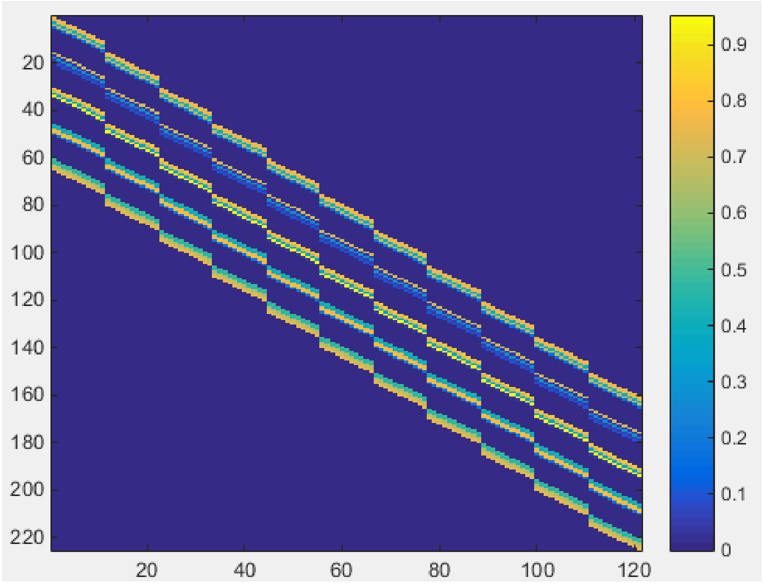
Other applications

–Discrete Fourier Transform , Convolution, IoT



Miao, ICRC 2016

Matrix T for convolution kernel



Sensors → Analog Buffer → DPE → MP → Blue Tooth

NNs
Filters

Conclusion

- We analyzed the challenges for a practical Dot-Product Engine implementation on nano-crossbars
- We present a conversion algorithm with near-zero overhead:
  - Scalable up to 512x512 crossbar model or even more
  - Up to 8-Bit output accuracy
  - <2 second on a normal desktop workstation for 128x128 crossbar
- DPE is excellent as accelerators for off-line machine learning algorithms:
  - More than 3 orders of magnitude improvement comparing to the best possible ASIC
    - 1,000 to 10,000 better speed-energy efficiency product
  - Enough and flexible computing accuracy for trained NNs (no training yet)

Thank you!

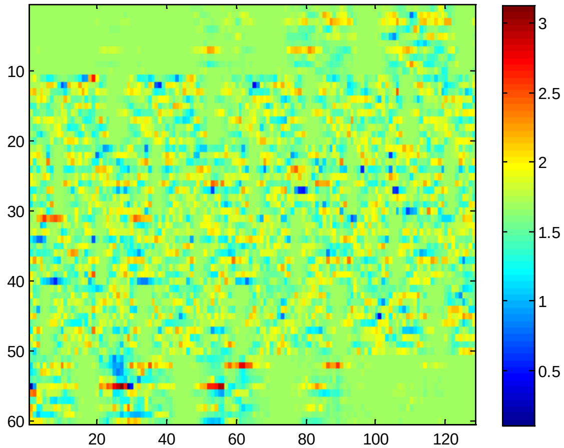
For more detail please refer:

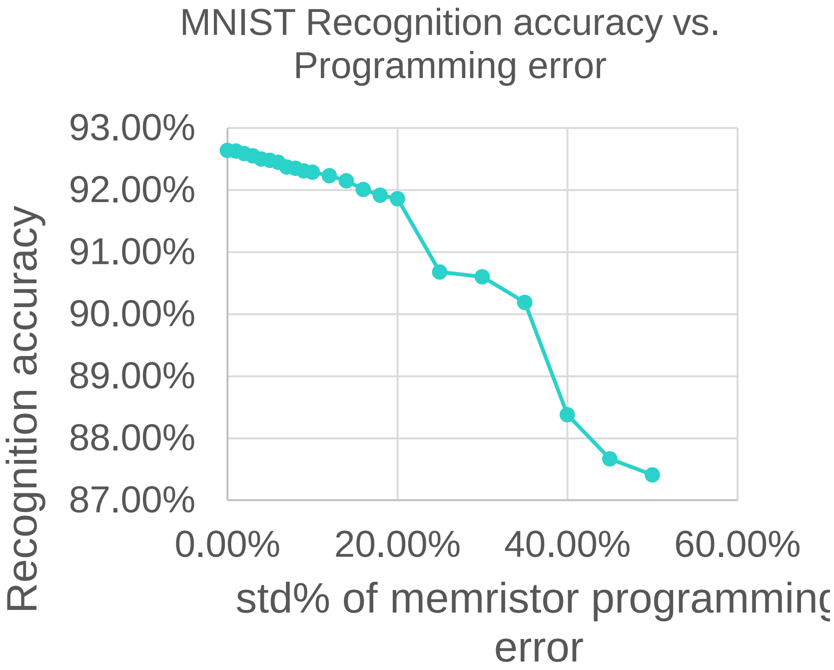
Miao et. al, “Dot-product engine for neuromorphic computing: programming 1T1M crossbar to accelerate matrix-vector multiplication,” DAC 2016.

Simulated accuracy for softmax neural network on crossbar

– Very slight performance degradation even with large device programming error

– Because of well-trained weight matrix.

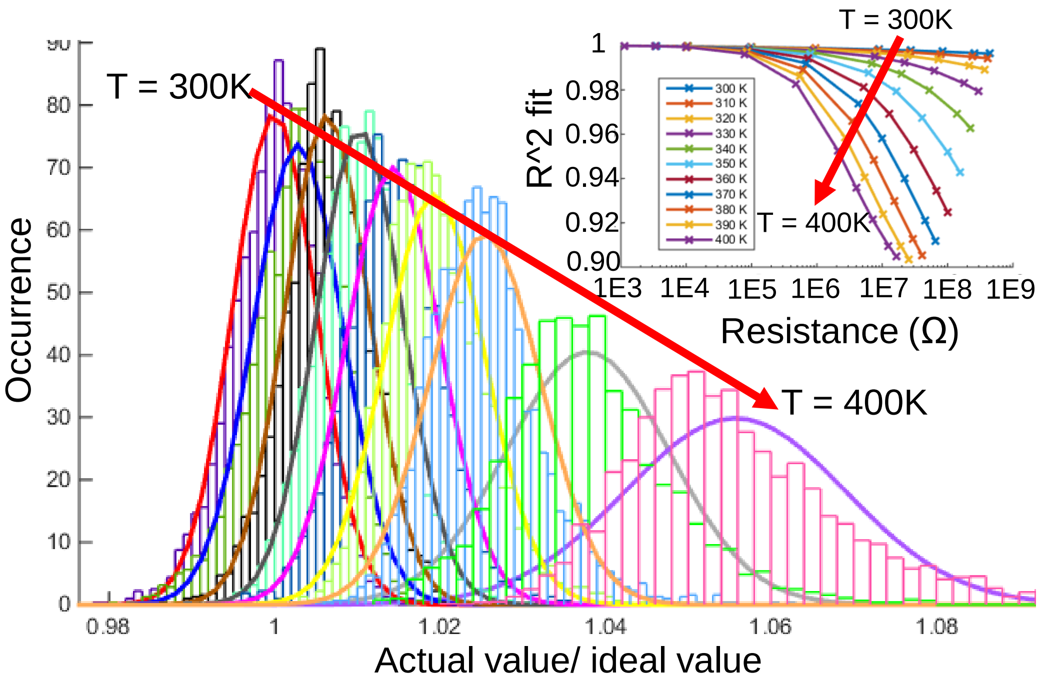


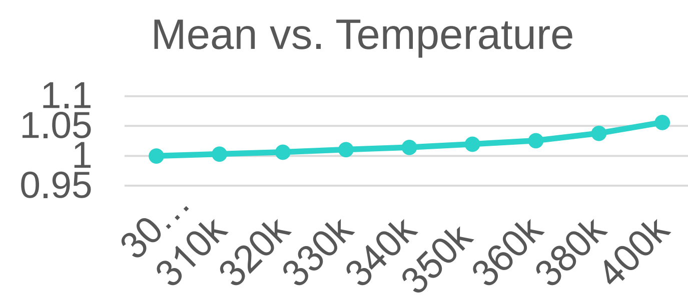


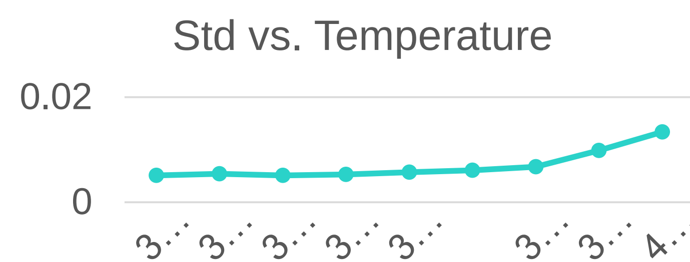
Conversion algorithm optimization : Temperature

– Best to calibrate the temperature to the working temperature, but it has a large tolerance margin.

– Test case: 32x32 crossbar, 10 ohm wire segment, calibrated at 0.25V, Tested at 0~0.5V.







Theoretical analysis of conversion algorithm for W to G’

–Problem definition:

–Assume **W** is positive, can we use a crossbar with wire block resistance Gw and linear devices with tuned conductance map **G’**, to realize ideal calculation Iout = Vin * **G** = a*X***W +** b for any input vector X with zero error? a and b are coefficients to linearly map X to Vin and **W** to **G**.

–Answer:

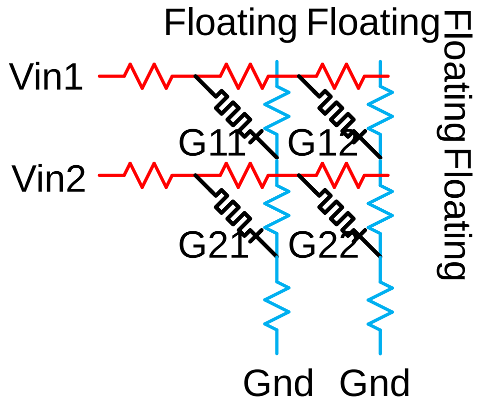
–Yes, there is **G’** for ideal matrix vector multiplication with arbitrary inputs.

–However, this **G’** will be extremely difficult to be analytically calculated.

Start with the simplest example (2x2 crossbar)

Top node KCL equations:
(Vt11-Vin1)*Gw + (Vt11-Vt12)*Gw + (Vt11-Vb11)*G11 = 0;
(Vt21-Vin2)*Gw + (Vt21-Vt22)*Gw + (Vt21-Vb21)*G21 = 0;
(Vt12-Vt11)*Gw + (Vt12-Vb12)*G12 = 0;
(Vt22-Vt21)*Gw + (Vt22-Vb22)*G22 = 0;

Bot node KCL equations:
(Vb11-Vt11)*G11 + (Vb11-Vb21)*Gw = 0;
(Vb21-Vt21)*G21 + (Vb21-Vb11)*Gw + Vb21*Gw = 0;
(Vb12-Vt12)*G12 + (Vb12-Vb22)*Gw = 0;
(Vb22-Vt22)*G22 + (Vb22-Vb12)*Gw + Vb22*Gw = 0;



Variable definition:
Gw: Wire block resistance;
Vtij: top voltage of the cross-point at ith row and jth column
Vbij: bottom voltage of the cross-point at ith row and jth column
Gij: conductance of the cross-point device at ith row and jth column

Calculate voltage across devices V_{device}

$$\underbrace{\begin{bmatrix} 2Gw+G11 & & -Gw & & -G11 & & & \\ & 2Gw+G21 & & -Gw & & -G21 & & \\ -Gw & & Gw+G12 & & & & -G12 & \\ & -Gw & & Gw+G22 & & & & -G22 \\ -G11 & & & & Gw+G11 & -Gw & & \\ & -G21 & & & -Gw & 2Gw+G21 & & \\ & & -G12 & & & & Gw+G12 & -Gw \\ & & & -G22 & & & -Gw & 2Gw+G22 \end{bmatrix}}_{A} \begin{bmatrix} Vt11 \\ Vt21 \\ Vt12 \\ Vt22 \\ Vb11 \\ Vb21 \\ Vb12 \\ Vb22 \end{bmatrix} + \underbrace{\begin{bmatrix} -Gw & \\ & -Gw \\ & \\ & \\ & \\ & \\ & \\ & \end{bmatrix}}_{B} \begin{bmatrix} Vin1 \\ Vin2 \end{bmatrix} = 0$$

$$\boldsymbol{A \begin{bmatrix} Vt \\ Vb \end{bmatrix} + BV_{in}^{T} = 0 \rightarrow \begin{bmatrix} Vt \\ Vb \end{bmatrix} = -A^{-1}BV_{in}^{T}}$$

$$\boldsymbol{V_{device} = \begin{bmatrix} I & -I \end{bmatrix} \begin{bmatrix} Vt \\ Vb \end{bmatrix} = -\begin{bmatrix} I & -I \end{bmatrix} A^{-1}BV_{in}^{T} = CV_{in}^{T}}$$

To realize ideal matrix vector multiplication

- Sufficient condition: Multiplication at each entry is accurate, V_{device} .* G' = V_{device_ideal} .* G.

Current through the first device in actual condition/ ideal condition (Gw = +∞)

$$\begin{bmatrix} C11 * G'11 & C12 * G'11 \\ C21 * G'21 & C22 * G'21 \\ C31 * G'12 & C32 * G'12 \\ C41 * G'22 & C42 * G'22 \end{bmatrix} \begin{bmatrix} Vin1 \\ Vin2 \end{bmatrix} = \begin{bmatrix} G11 & 0 \\ 0 & G21 \\ G12 & 0 \\ 0 & G22 \end{bmatrix} \begin{bmatrix} Vin1 \\ Vin2 \end{bmatrix}$$

N^3 nonlinear equations with N^2 variables.

- Sufficient and necessary condition: Only matrix vector multiplication result is accurate, $I_{actualoutput} = I_{idealoutput}$

Current through the first column in actual condition/ ideal condition (Gw= +∞)

$$\begin{bmatrix} C11 * G'11 + C21 * G'21 & C22 * G'21 + C12 * G'11 \\ C31 * G'12 + C41 * G'22 & C32 * G'12 + C42 * G'22 \end{bmatrix} \begin{bmatrix} Vin1 \\ Vin2 \end{bmatrix} = \begin{bmatrix} G11 & G21 \\ G12 & G22 \end{bmatrix} \begin{bmatrix} Vin1 \\ Vin2 \end{bmatrix}$$

N^2 nonlinear equations with N^2 variables

Numerical method to approximate G'

- The main issue of ideal equations is lack of direct physical representation

This term stands for the contribution of conductance by Vin1 to the first column, but there is no physical term stands for that in the crossbar simulation!

$$\begin{bmatrix} C11*G'11+C21*G'21 & C22*G'21+C12*G'11 \\ C31*G'12+C41*G'22 & C32*G'12+C42*G'22 \end{bmatrix} \begin{bmatrix} Vin1 \\ Vin2 \end{bmatrix} = \begin{bmatrix} G11 & G21 \\ G12 & G22 \end{bmatrix} \begin{bmatrix} Vin1 \\ Vin2 \end{bmatrix}$$

- With device models, it still the best to tune current at every cross-point to the ideal value because:
  1. Current through each device is well-defined in the simulation, it makes calculation much easier.
  2. Since device has voltage dependence, even ideal equations will not guarantee zero error for arbitrary inputs.

$$\begin{bmatrix} C11*G'11*Vin1+C12*G'11*Vin2 \\ C21*G'21*Vin1+C22*G'21*Vin2 \\ C31*G'12*Vin1+C32*G'12*Vin2 \\ C41*G'22*Vin1+C42*G'22*Vin2 \end{bmatrix} = \begin{bmatrix} G11*Vin1 \\ G21*Vin2 \\ G12*Vin1 \\ G22*Vin2 \end{bmatrix}$$