Towards a chip architecture for acceleration of Deep Neural Networks using Analog Memory

Stefano Ambrogio
Pritish Narayanan
Hsinyu Tsai
Charles Mackin
An Chen
Robert M. Shelby
Geoffrey W. Burr
IBM Research – Almaden
Outline

- Introduction
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
- Circuit design considerations
- Conclusion
Outline

- Introduction
  - 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

- **Data**
- **Compute**
- **Algorithms**

ImageNet Classification Error

- 2010
- 2011
- 2012: Deep Neural Networks
- 2013
- 2014
- 2015
- Human
AI hardware, present & near-future: high-level view

- **Forward Inference** (in the cloud & at the edge)
  - TODAY: CPUs & GPUs
  - VERY SOON: TPU1
  - LATER ON…?: Custom digital accelerators
- **Training** (mostly in the cloud)
  - TODAY: CPUs & GPUs
  - VERY SOON: TPU1
  - LATER ON…?: TPU2
  - LATER ON…?: Analog-memory-based accelerators?
Outline

- Introduction
- Analog memory for training Neural Networks
  - Software-equivalent accuracy with novel unit cell
  - Circuit design considerations
- Conclusion
Computation needed: “Multiply-accumulate”

With a GPU, matrix-multiplication is fast and parallel

\[ y_j = f \left( \sum x_i w_{ij} \right) \]

but \( x \) and \( w \) values must arrive from DRAM and new \( y \) values sent back to DRAM
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

**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.

![Diagram](image_url)

- **Address decoder**
- **V\text{read}**
- **Selector device**
- **Analog resistors**
- **Sense-Amplifiers** (analog current $\rightarrow$ 0s and 1s)
Multiply-accumulate with NVM: computed at the data, by physics

\[ y_j^B = f(\sum x_i \cdot w_{ij}) \]

1) Different peripheral circuitry
2) Weights \( w \rightarrow \) conductances \( G^+, G^- \)
   (Ohm’s Law: \( V = I \cdot R \rightarrow I = G \cdot V \))
3) Apply “\( x \)” voltages to every row
   (Kirchhoff’s Current Law \( \sum I \))
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...)

<table>
<thead>
<tr>
<th>Of zero interest</th>
</tr>
</thead>
</table>

### Accuracy

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

**Of zero interest**

### Faster

(circuitry must be massively parallel)

**Of zero interest**

| Sweet spot: rather than buy GPUs, people buy this chip instead for training of Deep-NN's |
| Still of interest for power-constrained situations: learning-in-cars, etc. |
| Still of interest for some situations: learning-in-server-room |

**Of zero interest**
Outline

- Introduction
- Analog memory for training Neural Networks
- Software-equivalent accuracy with novel unit cell
  - Circuit design considerations
- Conclusion
Where we were in 2014

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

“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

- 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

\[ W = G^+ - G^- \]
2-PCM scheme: dependence on applied pulses

- $\Sigma \Delta W$ 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.
Novel 2T2R + 3T1C unit cell

\[ W = F \times (G^+ - G^-) + g^+ - g^- \]

- **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)

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

- 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

**Dynamic Range Fraction**

-100 net pulses

**PDF**

-100 net pulses

+100 net pulses

**MNIST Accuracy**

TensorFlow: 97.94%

2T2R+3T1C: 98.10%
Novel unit cell: 2T2R + 3T1C, CMOS variability

- 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 $\sum \Delta W$ 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

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

\[ W = F \times (G^+ - G^-) + (g^+ - g^-) \]

or

\[ W = F \times (G^+ - G^-) - (g^+ - g^-) \]


Stronger PFET

Equal PFET and NFET

Stronger NFET

+100 net pulses

Transfer

Read current ADDs to weight

Read current SUBTRACTS from weight

INCREASE weight

INCREASE weight

DECREASE weight

DECREASE weight

Increase weight

Decrease weight

SUBTRACTS from weight
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

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

Transfer learning from ImageNet to CIFAR-10/100

Mixed hardware-software experiment

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

Only train last fully-connected layer

Convolutional and Subsampling layers

Fully Connected layer
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

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

Infrequent transfer from \( g^+ \) and \( g^- \)

Weight update

Most Significant Pair (MSP)

Least Significant Pair (LSP)
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

## Comparison of device specifications for MSP and LSP

<table>
<thead>
<tr>
<th>Specifications</th>
<th>Parameter</th>
<th>MSP</th>
<th>LSP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Step-size</td>
<td>$\Delta G_0$</td>
<td>$&lt; 21 \mu S$ (42%)</td>
<td>$&lt; 1.4 \mu S$ (2.8%)</td>
</tr>
<tr>
<td></td>
<td>$\Delta G_0^*$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intra-device Variability</td>
<td>$\sigma_{\text{intra}}$</td>
<td>$&lt; 1.5 \mu S$</td>
<td>$&lt; 0.8 \mu S$</td>
</tr>
<tr>
<td>Inter-device Variability</td>
<td>$\sigma_{\text{Gmax}}$</td>
<td>$&lt; 10 \mu S$</td>
<td>$&lt; 12 \mu S$</td>
</tr>
<tr>
<td></td>
<td>$\sigma_{\Delta G0}^*$</td>
<td>$&lt; 200%$</td>
<td>$&lt; 95%$</td>
</tr>
<tr>
<td>Faulty devices</td>
<td>Dead C.R.</td>
<td>$&lt; 7%$</td>
<td>$&lt; 7%$</td>
</tr>
<tr>
<td></td>
<td>Stuck On C.R.</td>
<td>$&lt; 35%$</td>
<td>$&lt; 10%$</td>
</tr>
<tr>
<td>Dynamic range</td>
<td>Number of levels</td>
<td>$&gt; 13$</td>
<td>$&gt; 110$</td>
</tr>
<tr>
<td>Retention</td>
<td>Time before data loss</td>
<td>Higher</td>
<td>Lower</td>
</tr>
<tr>
<td>Endurance</td>
<td>Number of Set/Reset</td>
<td>Lower</td>
<td>Higher</td>
</tr>
</tbody>
</table>

---

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

Outline

- Introduction
- 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...)

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

**Faster**
(circuitry must be massively parallel)

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

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

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

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

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

- Introduction
- 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
Photos of us with “our first wafer of PCM-based circuit designs”

Sidney Tsai
Bob Shelby
Stefano Ambrogio
Kohji Hosokawa

Geoffrey Burr
Pritish Narayanan
G. Burr
P. Narayanan

Not shown: Scott C. Lewis (YKT)
Thank you!

stefano.ambrogio@ibm.com
What do we mean by “mixed-hardware-software experiment”?

- **Full software simulation**
  - NVM devices
  - CMOS Periphery, Neurons, etc.
  - Modeled in software (SPICE) → accurate!

- **Mixed-hardware-software experiment**
  - On-chip memory array
  - (the real yield, variability, non-ergodic statistics, etc.)
  - Modeled in software (SPICE) → accurate!

- **Full hardware experiment**
  - On-chip memory array
  - Real CMOS implementation

- **Make a few NVM & measure, then capture in a statistical model → not very accurate!**
Impact of different techniques

- Polarity inversion shows the largest impact on accuracy
- Other techniques show varying importance depending on the training dataset (MNIST, MNIST backrand, CIFAR-10/100)