Ultra-Efficient Neural Algorithm Accelerator Using Processing With Memory

Matthew J. Marinella, Sapan Agarwal, Robin Jacobs-Gedrim, Alex Hsia, David Hughart, Elliot Fuller, Steve J. Plimpton, A. Alec Talin, and Conrad D. James

Sandia National Laboratories

*matthew.marinella@sandia.gov
Why do we need more efficient computers?

- **Google Deep Learning Study**
  - 16000 core, 1000 machine GPU cluster
  - Trained on 10 million 200x200 pixel images
  - Training required 3 days
  - Training dataset size: no larger than what can be trained in 1 week

- **What would they like to do?**
  - ~2 billion photos uploaded to internet per day (2014)
  - Can we train a deep net on one day of image data?
  - Assume 1000x1000 nominal image size, linear scaling (both assumptions are unrealistically optimistic)
  - **Requires 5 ZettaIPS to train in 3 days**
    (ZettaIPS=10^{21} IPS; ~5 billion modern GPU cores)
  - World doesn’t produce enough power for this!
  - Data is increasing exponentially with time

- **Need >10^{16}-10^{18} instruction-per-second on 1 IC**
  - Less than 10 fJ per instruction energy budget

Q. Le, IEEE ICASSP 2013
Inspired by Hasler and Marr, Frontiers in Neuroscience, 2013

Neural Inspired Computational Elements

Evolution of Computing Machinery

<table>
<thead>
<tr>
<th>Year</th>
<th>Efficiency (pJ/MAC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1980’s</td>
<td>100</td>
</tr>
<tr>
<td>1990’s</td>
<td>10</td>
</tr>
<tr>
<td>2000’s</td>
<td>1</td>
</tr>
<tr>
<td>2010-Present</td>
<td>100</td>
</tr>
<tr>
<td>2025?</td>
<td>100 fJ/MAC</td>
</tr>
<tr>
<td>2035?</td>
<td>1 fJ/MAC</td>
</tr>
</tbody>
</table>

Moore’s Law Era: Density scaling and Dennard Power Density Scaling

Modern Computer

Single GPU Card, Late 2016

Exascale System Goal

Heterogeneous Integration Era: Close integration of emerging memory, low voltage logic, photonics

Neuromorphic Target

Efficiency only possible with new computing paradigm

Biological Neurons*

“Let physics do the computation” Our brain is the ultimate example of this paradigm

*Caveat: Biological neurons probably do not perform MACs
Metal Oxide Resistive RAM (ReRAM)

- Example: Sandia TiN/Ta/TaOx/TiN device
- Starts as insulating MIM structure
- Forming: remove $O^{2-} \rightarrow$ soft breakdown
- Bipolar resistance modulation
- Excellent memory attributes: Switching in less than 1 ns, less than 1 pJ demonstrated, scaling to 5 nm, $>10^{12}$ write cycles

**Pre-Form I/V**

**Forming**

**SET-RESET**

Example: Sandia TiN/Ta/TaOx/TiN device
- Starts as insulating MIM structure
- Forming: remove $O^{2-} \rightarrow$ soft breakdown
- Bipolar resistance modulation
- Excellent memory attributes: Switching in less than 1 ns, less than 1 pJ demonstrated, scaling to 5 nm, $>10^{12}$ write cycles

**Pre-Form I/V**

**Forming**

**SET-RESET**
Crossbar Theoretical Limits

- Potential for 100 Tbit of ReRAM on chip
- If each can perform 1M computations of interest per second (1 M-op):
  - $10^{12}$ active devices/chip x $10^6$ cycle per second $\rightarrow 10^{18}$ comps per second per chip
  - Exascale-computations per sec on one chip!
- In order to not melt the chip, entire area must be limited to $\sim 100$W
- Allowed energy per operation = $P \times t/\text{op}$
  - $100W / 10^{18} = 10^{-16} = 100\text{ aJ/operation}$
- 10nm line capacitance = 10 aF
- Can charge line to 1V with 10 aJ
- Drawback: “only” $\sim 100B$ transistors/chip
How does a crossbar perform a useful computation per device?

- Electronic Vector Matrix Multiply

Mathematical

\[
V^TW = I
\]

\[
\begin{bmatrix}
v_1 \\
v_2 \\
v_3
\end{bmatrix}
\begin{bmatrix}
w_{1,1} & w_{1,2} & w_{1,3} \\
w_{2,1} & w_{2,2} & w_{2,3} \\
w_{3,1} & w_{3,2} & w_{3,3}
\end{bmatrix}
= 
\begin{bmatrix}
i_1=\Sigma v_{i,1}w_{i,1} \\
i_2=\Sigma v_{i,2}w_{i,2} \\
i_3=\Sigma v_{i,3}w_{i,3}
\end{bmatrix}
\]

Electrical

\[
V^TW = I
\]

\[
\begin{bmatrix}
v_1 \\
v_2 \\
v_3
\end{bmatrix}
\begin{bmatrix}
G_{1,1} & G_{1,2} & G_{1,3} \\
G_{2,1} & G_{2,2} & G_{2,3} \\
G_{3,1} & G_{3,2} & G_{3,3}
\end{bmatrix}
= 
\begin{bmatrix}
i_1=\Sigma v_{i,1}G_{i,1} \\
i_2=\Sigma v_{i,2}G_{i,2} \\
i_3=\Sigma v_{i,3}G_{i,3}
\end{bmatrix}
\]
Vector Matrix Multiply, Rank 1 Update:
Key kernel used in many algorithms

Convert analog inputs to varying length voltage pulses

Integrate current to get an analog output value
Accelerator Architecture

Network is actually a grid, omitted here for simplicity

On-Chip Network

Chip Control & Bus Interface
Core
Core
Core
Memory (buffers, training sets, ...)
[might be added to package via 2.5D integration]
Core
Core
Core
Core

Data Flow & Synchronization

Local Vector Cache
SIMD Engine
Co-located Pre- and Post-processing
Row Values
Row Drivers
Neuron Function
ReRAM Array w/ peripherals
ReRAM Array w/ peripherals
Column Values
Column Drivers
Column Drivers
MUX & ADC
Neuron Function
LUT
Outputs
Integrators
ReRAM Crossbar
Integrators
ReRAM Array w/ Peripherals
Neuron Function

Core (multiple crossbars, some digital SIMD operations)
Device to Algorithm Model

What device properties are needed?

Neural Algorithm Level Model

Computer Architecture Level Model

Circuit Level Models

Device Level Models

How do specific devices work in system?
Experimental Device Nonidealities

- Ideally weight would increase and decrease linearly proportional to learning rule result
- Experimental devices have several nonidealities: Write Variability, Write Nonlinearity, Asymmetry, Read Noise
- Circuits also have A/D, D/A noise, parasitics
ReRAM Analog Characterization

- Use as a neuromorphic weight requires precise analog tuning
- Dataset requires 1000 repeated SET and RESET pulses
- Nominal pulse values
  - SET: +1V 10ns RT/PW/FT
  - RESET: -1V 10ns RT/PW/FT
  - READ: 100 mV 1 ms RT/PW/FT
Repeated Pulsed Cycling

100 on→off cycles, (200k pulses)

10 ns PW

100 ns PW

1000 ns PW

10 ns

100 ns

1 µs

RESET

SET

Conductance (S)

Pulse Number (#)

Delta G (µS)

Conductance (µS)

CDF

Delta G (µS)

Conductance (µS)

CDF

Delta G (µS)

Conductance (µS)

CDF
How can training accuracy be improved?

### TaOx ReRAM in Backprop Training

<table>
<thead>
<tr>
<th>TaOx</th>
<th>Large Images</th>
<th>Small Images</th>
<th>File Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 ns</td>
<td>84.45%</td>
<td>71.40%</td>
<td>77.67%</td>
</tr>
<tr>
<td>100 ns</td>
<td>78.48%</td>
<td>89.48%</td>
<td>67.78%</td>
</tr>
<tr>
<td>1 μs</td>
<td>71.48%</td>
<td>71.84%</td>
<td>56.33%</td>
</tr>
</tbody>
</table>

Increasing Network Size
Li-Ion Synaptic Transistor for Analog Computation (LISTA)

E. Fuller et al, *Adv Mater*, accepted 2017
Analog State Characterization

LISTA > 200 states

TaOx ReRAM

PCM Array

E. Fuller et al, Adv Mater, accepted 2017

GW Burr et al, IEEE TED 2015

Neural Inspired Computational Elements
LISTA-device Performance for Backprop Algorithm

<table>
<thead>
<tr>
<th>Data set</th>
<th># Training Examples</th>
<th># Test Examples</th>
<th>Network Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>UCI Small Digits[1]</td>
<td>3,823</td>
<td>1,797</td>
<td>64×36×10</td>
</tr>
<tr>
<td>MNIST Large Digits[3]</td>
<td>60,000</td>
<td>10,000</td>
<td>784×300×10</td>
</tr>
</tbody>
</table>

See Poster for Detail!
E. Fuller et al, Adv Mater, accepted 2017
van de Burgt ... Saleo, *Nature Mater.*, 2017 in press
Electrochemical Neuromorphic Organic Device (eNode)

See Poster for Detail!
van de Burgt ... Saleo, Nature Mater., 2017 in press
Circuit-Level Improvement

- Allows much closer to ideal with high variability TaOx device
- LISTA achieves essentially perfect accuracy
- Requires tradeoff of energy/latency for accuracy – exact tradeoff depends on algorithm reqs.

Agarwal et al, submitted 2017
## Energy and Latency Analysis

<table>
<thead>
<tr>
<th></th>
<th>SRAM</th>
<th>Digital ReRAM</th>
<th>Analog ReRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Equivalent Area</strong></td>
<td>400 mm²</td>
<td>32 mm²</td>
<td>11 mm²</td>
</tr>
<tr>
<td>~450 1k × 1k matrices</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Time</strong></td>
<td>~100 µs</td>
<td>~60 µs</td>
<td>~5 µs</td>
</tr>
<tr>
<td>per 1-layer cycle</td>
<td>Transpose read dominated</td>
<td>Update dominated: 10 ns write</td>
<td>Temporal coding dominated: 256 levels</td>
</tr>
<tr>
<td>3 Ops: 2 reads, 1 write</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total Energy</strong></td>
<td>~1000 nJ</td>
<td>~700 nJ</td>
<td>~15 nJ</td>
</tr>
<tr>
<td>per 1-layer cycle</td>
<td>Multiply dominated</td>
<td>Multiply dominated</td>
<td></td>
</tr>
<tr>
<td><strong>Matrix Storage Area</strong></td>
<td>95%</td>
<td>50%</td>
<td>17%</td>
</tr>
<tr>
<td><strong>Periphery Area</strong></td>
<td>5%</td>
<td>50%</td>
<td>100%</td>
</tr>
<tr>
<td><strong>Matrices per</strong></td>
<td>~450</td>
<td>~5,500</td>
<td>~15,000</td>
</tr>
<tr>
<td>400 mm² Chip</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Energy / Operation</strong></td>
<td>330 fJ</td>
<td>230 fJ</td>
<td>5 fJ</td>
</tr>
<tr>
<td><strong>Operations/Second</strong></td>
<td>14 TeraOps/S</td>
<td>270 TeraOps/S</td>
<td>9,000 TeraOps/S</td>
</tr>
</tbody>
</table>
## Energy and Latency Analysis

<table>
<thead>
<tr>
<th>Matrix Storage 1024x1024 [Values are per-array]</th>
<th>SRAM</th>
<th>Digital ReRAM</th>
<th>Analog ReRAM Crossbar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Area</strong></td>
<td>800,000 µm²</td>
<td>35,000 µm²</td>
<td>Array: 4,300 µm² Periphery: 8,460 µm²</td>
</tr>
<tr>
<td><strong>Read</strong></td>
<td>30 nJ / 15 µs</td>
<td>15 nJ / 4 µs</td>
<td>~ 3 nJ / ~ 1.5 µs</td>
</tr>
<tr>
<td><strong>Read Transpose</strong></td>
<td>300 nJ / 65 µs</td>
<td>15 nJ / 4 µs</td>
<td>~ 3 nJ / ~ 1.5 µs</td>
</tr>
<tr>
<td><strong>Write</strong></td>
<td>30 nJ / 15 µs</td>
<td>50 nJ / 45 µs</td>
<td>3 nJ / ~ 1.5 µs</td>
</tr>
</tbody>
</table>

| Multiply Accumulators [256 in parallel]       |      |               | FREE |
| **Area**                                     |      | 19,000 µm²    |      |
| **Run: 1M ops**                              |      | 200 nJ / 4 µs |      |

| Output LUT                                   |      |               | Uses Digital Methods |
| **Area**                                     |      | 1,400 µm²     |      |
| **Read**                                     |      | 1 nJ / 1 µs   |      |

| Input/Output Buffers [8 bits]                |      |               | Uses Digital Methods |
| **Area**                                     |      | 13,000 µm²    |      |
| **Per Run**                                  |      | ~ 0.1 nJ      |      |

| Vector Cache* 16 entries 1024x8-bit [Values are per 1024x8 vector] |      |               | Uses Digital Methods |
| **Area**                                     |      | 11,250 µm²    | 500 µm² |
| **Read**                                     |      | ~ 0.1 nJ / ~ 0.2 µs | ~ 1 nJ / 4 ns |
| **Write**                                    |      | ~ 0.1 nJ / ~ 0.2 µs | ~ 1 nJ / 50 ns |
Conclusion

- Dennard (constant power density) scaling has ceased and Moore’s law is slowing
- New paradigms like neuromorphic computing will be required for sub-fJ computing
- We now require a device through system design mentality
  - Motivation behind CrossSim
  - See poster for more detail on CrossSim
- Oxide-based resistive memory offers intriguing device options for both eras
- Novel LISTA and eNode devices, offer significant potential in the development of a low energy neural accelerator
  - See LISTA and eNode posters for more detail on these
Thank you!
Acknowledgements

- This work is funded by Sandia’s Laboratory Directed Research and Development as part of the Hardware Acceleration of Adaptive Neural Algorithms Grand Challenge Project

- Many shared ideas among collaborators:
  - Alberto Saleo, Yoeri van de Burgt, Stanford
  - Engin Ipek, U Rochester
  - Dave Mountain, Mark McLean, US Government
  - Stan Williams, John Paul Strachan, HPL
  - Dhireesha Kudithipudi, RIT
  - Jianhua Yang, U Mass
  - Hugh Barnaby, Mike Kozicki, Sheming Yu, ASU
  - Tarek Taha, U Dayton
  - Paul Franzon, NC State University
  - Sayeef Salahuddin, UC Berkeley
  - Dozens of others...

- **We are especially interested in collaborations on cross-sim!**
ENIAC

Electronic Numerical Integrator And Computer
Developed by US Army/U Penn – 1946
150 kW, 357 FLOPs
400 J/FLOP (10 bit)
Where Are we Today?

- **Single Unit: Nvidea Tesla P100 GPU**
  - Most advanced GPU processor specs, released late 2016
  - Target’s deep learning and neural applications
  - 20 TFLOPs 16 bit peak performance w/ peak power dissipation of 300W
  - 70 GFLOPs/watt or about 15 pJ/FLOP (16 bit)

- **Supercomputer: Sunway TaihuLight (China)**
  - Top supercomputer in the world
  - ShenWei processor
  - 90 PFLOPs peak, 15 MW power
  - 6 GFLOPs/W or about 170 pJ/FLOP

- Need >1000x improvement to tackle internet-scale problems
Basics of Neural Networks

Basic Building Block

\[ y = \frac{1}{1 + e^{-z}} \]

Neuron

\[ z = \sum_{i=0}^{n} w_i x_i \]

Weights

Inputs

\[ x_1, x_2, \ldots, x_n \]

Simple Network: Backpropagation

Incorrect – adjust

Correct – no adjustment

Outputs

Hidden Layer

Inputs

0 1 2 3
Another Analogy

Mathematical

\[ y = \frac{1}{1 + e^{-z}} \]

\[ z = \sum_{i=0}^{n} w_i x_i \]

Electrical

\[ I = \sum_{i=0}^{n} G_i V_i \]
Why is it essential to cram so many computations on a single chip?

Can you simply connect millions of ultra-efficient chips?

Yes, but every time data leaves the chip, it is elevated in the comm hierarchy

→ Energy efficiency per operation is reduced
TaOx ReRAM in Backprop Training

### Increasing Network Size

#### Small Digits

- **Data set:** UCI Small Digits[1]
- **# Training Examples:** 3,823
- **# Test Examples:** 1,797
- **Network Size:** $64 \times 36 \times 10$

#### File Types

- **Data set:** File Types[2]
- **# Training Examples:** 4,501
- **# Test Examples:** 900
- **Network Size:** $256 \times 512 \times 9$

#### Large Digits

- **Data set:** MNIST Large Digits[3]
- **# Training Examples:** 60,000
- **# Test Examples:** 10,000
- **Network Size:** $784 \times 300 \times 10$

1 µs pulses
ReRAM Measurements

- DC Current-voltage “loops” sweeps are not time-controlled
  - Excessive heating and early wearout
  - Do not provide info on dynamics
- Physical switching < 10ns
- Need pseudo RF setup to measure
  - Ground/signal, conductor backed
  - Agilent B1530 module
  - 10 ns RT/FT, 10 ns PW
  - 1 V nominal, ~140 mV overshoot

\[
\begin{align*}
\text{Rise} &= 12.8 \text{ ns} \\
\text{Fall} &= 11.4 \text{ ns} \\
\text{Amp} &= 1.14 \text{ V}
\end{align*}
\]
Shorter pulses may be employed to lower conductance switching range

Linearity qualitatively similar across Pulse Width (PW) and Edge Time (ET)
- Best for SET at 100 ns
- Best for RESET at 1 us

Relative conductance change increased with shorter Pulse Width / Edge Time

Nominal Pulse Voltage Values: SET: +1 V RESET: -1 V
Analog Core: Forward Propagation

\[ z_j = \sum_i y_i \times w_{ij} \]

\[ y_j = \frac{1}{1 + e^{-z_j}} \]

O(N^2) Operations

O(N) Operations
Analog Core: Back Propagation

\[ \delta_k = \sum_j w_{jk} \delta_k \]

\[ \Delta_k = \sum_k \delta_k \]

\[ \eta \times \delta_j \]

O(N^2) Read Operations

O(N) Operations

O(N^2) Write Operations

Neural Inspired Computational Elements