Solving the Non-Volatile Memory Conundrum for Deep Learning Workloads

Ahmet Inci and Diana Marculescu
Department of Electrical and Computer Engineering
Carnegie Mellon University
ainci@andrew.cmu.edu

Architectures and Systems for Big Data (ASBD)
Los Angeles, CA – June 2, 2018
Hitting the "Walls"

- "Hitting the memory wall"
  - [Wulf et al., CAL’95]

- End of Dennard scaling
  - Power wall
    - [Dennard et al., SSC’74]

- von Neumann architecture
  - Data movement

- What’s next?
Hitting the "Walls"

- "Hitting the memory wall"
  - [Wulf et al., CAL’95]

- End of Dennard scaling
  - Power wall
    - [Dennard et al., SSC’74]

- von Neumann architecture
  - Data movement

- What’s next?
  - Non-volatile memory (NVM) [Chi et al., ISCA’16]
  - Processing-in memory [Gu et al., ISCA’16]
  - 3D-stacking [Gao et al., ASPLOS’17]
NVM for Deep Learning

- **On-going debate** regarding whether and how NVM can be used for data-intensive applications, particularly for deep learning
  - Non-volatile memory (NVM)
  - Processing-in memory
  - Analog computation
  - Spiking

- In this work, we show **actual platform measurements** for memory footprints of various DNNs

- Our results show that NVM can be a promising solution for **data-intensive applications** in certain cases
Background: The Good, The Bad, and the Ugly
The Good

✓ Near-zero standby power
✓ High storage density
✓ Low read energy/latency
✓ Non-volatility
✓ Scalability
✓ Disruptive technology

[Source: Intel]
The Bad

✗ High write energy/latency
  ♦ STT-MRAM
    ✗ Single read/write current path
    ✗ Read disturb errors

  ♦ SOT-MRAM
    ✓ Separate read/write current paths
    ✓ Resilient to read disturb errors

✗ Endurance
  ♦ Limited write numbers (>10^{15})
    • Comparable to SRAM (≈10^{16})

[Oboril et al., TCAD'15]
The Ugly

- NVM is not commercially available

- Integrating NVM technology into cycle-accurate simulators for current architectures is non-trivial
  - Furthermore, cycle-accurate architectural simulators such as gem5 are inaccurate when varying simulator modes are used [Komalan et al., DATE’18]

- Obvious need for actual platform measurements to perform a limit study for NVM improvements!
Contributions

- **Profiling memory footprints** of various DNNs for both training and inference in different configurations

- Our results show that **NVM** can be a promising solution for data-intensive applications in certain cases

- Our analysis shows the cases where one would gain advantage of using **NVM** compared to conventional memory technologies such as **SRAM**

- This work is a roadmap for both **computer architecture** and **device technology** fields as our analysis relies on actual platform, fine-grain measurements
Infrastructure

- **Dataset**
  - ImageNet

- **DNNs (training & inference)**
  - AlexNet, GoogLeNet, ResNet-50, VGG16, MobileNet_v1

- **Platform**
  - Nvidia Titan X® GPU

- **Framework**
  - Caffe (with cuDNN)
Impact of batch size on L2 read/write

- **Read/write ratio** significantly increases as batch size increases
  - (e.g. bs 8 = 1.72, bs 512 = 9.49)

- **Due to its high cell density, NVM enables the use of larger batch sizes** [Smith et al., ICLR’18]
  - Faster inference
  - Faster convergence time
  - Potentially higher accuracy
VGG16 and ResNet-50 give sufficient motivation to use NVM.
NVM/SRAM Comparison

- **SRAM** occupies significant on-chip area

- **STT-MRAM** is 1.7x more area efficient compared to SRAM
  - *(i.e. 1.7x more storage within the same area budget)*

- **SOT-MRAM** is 1.54x more area efficient compared to SRAM

- **Increasing batch size** also becomes a better use case for NVM since it increases *read/write ratio*

---

### Power and Performance Parameters for Various Memory Technologies

<table>
<thead>
<tr>
<th></th>
<th>SRAM</th>
<th>STT-MRAM</th>
<th>SOT-MRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Read Latency (ns)</strong></td>
<td>2.17</td>
<td>1.2</td>
<td>1.13</td>
</tr>
<tr>
<td><strong>Write Latency (ns)</strong></td>
<td>2.07</td>
<td>11.22</td>
<td>1.36</td>
</tr>
<tr>
<td><strong>Read Energy (pJ)</strong></td>
<td>587</td>
<td>260</td>
<td>247</td>
</tr>
<tr>
<td><strong>Write Energy (pJ)</strong></td>
<td>355</td>
<td>2337</td>
<td>334</td>
</tr>
<tr>
<td><strong>Area (mm²)</strong></td>
<td>2.78</td>
<td>1.63</td>
<td>1.8</td>
</tr>
</tbody>
</table>
Limit Analysis for NVM

- **SOT-MRAM** would produce 1.7x-2.5x energy reduction and 1.7x-2x speedup when compared to SRAM

- **STT-MRAM** would produce up to 1.3x energy reduction and 1.1x speedup due to its single read/write current path
Future Work

- Extending DNN workloads, NVM types, and HW platforms
- Impact of framework on memory behavior
- Architectural simulations calibrated with platform measurements
Summary

- We investigate whether emerging NVM would be useful for deep learning workloads

- Contrary to common belief, our results show that in certain cases, using NVM would be beneficial for deep learning in terms of energy and performance

- Using large batch size is a better use case for NVM since it increases read/write ratio which increases overall improvement

- It might be a promising solution for other data-intensive applications as well
Thank you!