IMCE: Energy-Efficient Bit-Wise In-Memory Convolution Engine for Deep Neural Network

Deliang Fan

Assistant Professor
dfan@ucf.edu
http://www.eecs.ucf.edu/~dfan/
Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL
OUTLINE

- Motivation
- STT-MRAM and SOT-MRAM
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion
Convolutional neural networks (CNNs) are reaching record-breaking accuracy in image recognition on small data sets like MNIST, SVHN and CIFAR-10 with accuracy rates of 99.79%, 98.31% and 96.53% [1].

On the large data-sets like ImageNet, ResNet shows a prominent recognition accuracy (96.43%) even higher than humans! (94.9%).

Following the trend, when going deeper in CNNs (e.g. ResNet employs 18-1001 layers), memory/computational resources and their communication have faced inevitable limitations called “CNN power and memory wall”) [1,2].

Several methods have been proposed to break the wall:

A. Compressing pre-trained networks,
B. Quantizing parameters, and
C. Binarization
D. Prunning

Visualization of Inference in CNN

Energy efficient and high performance computing hardware development is beginning to stall fundamentally due to limitations in both devices and architectures.

First, the current computing platforms primarily depend on Complementary Metal Oxide Semiconductor (CMOS) technology, which is reaching its power wall.
MOTIVATION (ARCHITECTURE)

Von-Neumann architecture

- Energy hungry data transfer
- Long memory access latency
- Limited memory bandwidth

In-Memory Computing Cluster

- Parallel, local data processing
- Short memory access latency
- Ultra-low energy
- Programmable, Low cost/area

There is an urgent need to investigate fundamentally different devices and architectures for information processing and data storage with the ability to continuously deliver energy efficient and high performance computing solutions.

OUTLINE

- Motivation
- STT-MRAM and SOT-MRAM
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion
STT-MRAM

- Magnetic Tunnel Junction (MTJ) owns high/low resistance with respect to its free layer magnetization configuration.
- Current-induced Spin Transfer Torque (STT) MTJ switching scheme

**Key Advantages**
- Non-volatility
- High density
- No leakage power

**Limitations**
- Write asymmetry
- Reliability-limited write speed
- Read write optimization conflicts

<table>
<thead>
<tr>
<th>Operations</th>
<th>Write ‘1’ (‘0’)</th>
<th>Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>WL</td>
<td>$V_{DD}$</td>
<td>$V_{DD}$</td>
</tr>
<tr>
<td>BL</td>
<td>GND ($V_{DD}$)</td>
<td>$I_{sense}$</td>
</tr>
<tr>
<td>SL</td>
<td>$V_{DD}$ (GND)</td>
<td>GND</td>
</tr>
</tbody>
</table>
The bit-cell structure of 2T1R SOT-MRAM and its biasing conditions

Key Advantages

- Energy-efficient write
- Decoupled R/W current paths
- Separate optimization for Read and for Write

Limitations

- Requires two access transistors
- Switching PMA MTJ requires FL engineering that involves fabrication challenge

<table>
<thead>
<tr>
<th>Operations</th>
<th>Write ‘1’(‘0’)</th>
<th>Read</th>
</tr>
</thead>
<tbody>
<tr>
<td>WWL</td>
<td>$V_{DD}$</td>
<td>0</td>
</tr>
<tr>
<td>RWL</td>
<td>0</td>
<td>$V_{DD}$</td>
</tr>
<tr>
<td>RBL</td>
<td>0</td>
<td>$I_{READ}$</td>
</tr>
<tr>
<td>WBL</td>
<td>$V_{WP}(V_{WN})$</td>
<td>0</td>
</tr>
<tr>
<td>SL</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The bit-cell structure of 2T1R SOT-MRAM and its biasing conditions

OUTLINE

- Motivation
- STT-MRAM and SOT-MRAM
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion
IN-MEMORY PROCESSING PLATFORM

- Dual mode architecture that perform both memory read-write and AND/OR logic operations.

- **Memory Write:**
  1. WWL1 should be activated by the Row Decoder and SL1 is grounded.
  2. To write ‘1’ (‘0’), the voltage driver (V1) connected with WBL1 is set to positive (negative) write voltage.

- **Memory Read:**
  1. RWL1 is activated while SL1 is grounded.
  2. The Column Decoder activates the RBL1 line to be connected to the SA.
  3. A read current flows from the SOT-MRAM cell to ground, generating a sense voltage, which is compared with $V_{\text{ref}} : V_{\text{sense},P} < V_{\text{ref}} < V_{\text{sense},AP}$

- **Computing Mode:** Every two bits stored in the identical column can be selected and sensed simultaneously. Through selecting different reference resistances ($EN_M$, $EN_{\text{AND}}$, $EN_{\text{OR}}$), the SA can perform basic in-memory Boolean functions (i.e. AND and OR).
IN-MEMORY PROCESSING PLATFORM

- For AND operation, $R_{\text{ref}}$ is set at the midpoint of $R_{AP} \parallel R_p = (1,0)$ and $R_{AP} \parallel R_{AP} = (1,1)$
- For OR operation, $R_{\text{ref}}$ is set at the midpoint of $R_p \parallel R_p$ and $R_p \parallel R_{AP}$
- We have performed Monte-Carlo simulation with **100000 trials**. A $\sigma = 5\%$ variation is added on the Resistance-Area product ($RA_p$), and a $\sigma = 10\%$ process variation is added on the TMR.

- Sense Margin will be reduced by increasing the logic fan-in (i.e. number of parallel memory cells).

- To avoid read failure, only **two fan-in in-memory logic** is used in this work.

Monte Carlo simulation result
OUTLINE

- Motivation
- Spintronic Devices
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion
BIT-WISE CONVOLUTIONAL NEURAL NETWORK

- DoReFa-Net [1] proposes to employ low bitwise convolutions (weights and activations/grads) during forward/backward passes to accelerate both training and inference.

- Quantize floating point convolution into bit-wise convolution with limited accuracy loss for large scale IMAGENET benchmark in different configurations.

- Such bit-wise convolution could be totally implemented within our proposed accelerator using IMCE.

\[
O[n][k][x][y] = \text{ReLU}(B[k] + \sum_{i=0}^{F_h-1} \sum_{j=0}^{F_w-1} \sum_{z=0}^{C-1} I[n][z][U_x+i][U_y+j] \times W[k][z][i][j]),
\]

\[
I \ast W = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} 2^{m+n} \text{bitcount}((C_n(W), C_m(I)))
\]

Typical dot-product operation of convolutional layer

Low bit-width fixed-point integers dot-product developed by DoReFa-NET

IN-MEMORY CONVOLUTION ENGINE CONCEPT AND IMPLEMENTATION

- A potential solution to better address storage, computation, and data transfer bottlenecks of CNNs.
- This architecture mainly consists of Image Bank, Kernel Bank, bit-wise In-Memory Convolution Engine (IMCE), and Digital Processing Unit (DPU).
- Preprocessing:
  - Assume Input fmaps (I) and Kernels (W) are stored in Image Banks and Kernel Banks of memory.
  - Inputs need to be constantly quantized before mapping into computational sub-arrays. This step is performed using DPU’s Quantizer and then the results are mapped to IMCE’s sub-arrays.
  - IMCE is realized through the proposed SOT-MRAM based computational sub-array.

(a) General overview of the proposed CNN accelerator with image bank, kernel bank, computational sub-arrays, and DPU,
(b) Bit-wise IMCE’s sub-array.
IN-MEMORY CONVOLUTION ENGINE
CONCEPT AND IMPLEMENTATION

- The main idea is to exploit logic \textit{AND}, \textit{bitcount}, and \textit{bitshift} as rapid and parallelizable operations to accelerate the MACs in convolutional layers.

- Operation:
  - $I$ is a sequence of $M$-bit input integers (3-bit, here) located in input fmap covered by sliding kernel of $W$, such that $I_i \in I$ is a 3-bit vector representing a fixed-point integer (e.g. 3 = “011”).
  - We index the bits of each $I_i$ element from LSB to MSB with $m = [0, M-1]$, such that $m = 0$ and $m = M - 1$ are corresponding to LSB and MSB.
  - A second sequence $C_m(I)$ including the combination of $m$th bit of all $I_i$ elements (shown by colored elliptic) (e.g. $C_0(I)$ vector is “0110”). Thus $I$ can be written as:
    \[ I = \sum_{m=0}^{M-1} 2^m C_m(I) \]
  - Same procedure is applied to $W$ as a sequence of N-bit weight integers (3-bit, herein) to make:
    \[ W = \sum_{n=0}^{N-1} 2^n C_n(W) \]
  - In this way, the convolution between $I$ and $W$ can be defined as:
    \[ I \ast W = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} 2^{m+n} \text{bitcount}(\text{and}(C_n(W), C_m(I))) \]
IN-MEMORY CONVOLUTION ENGINE
CONCEPT AND IMPLEMENTATION

1) Mapping $C_2(W) - C_0(W)$ to the designated sub-array rows,
2) Mapping $C_2(I) - C_0(I)$ in the following memory rows in the same way,
3) Performing bit-wise parallel AND operation of $C_n(W)$ and $C_m(I)$ using the proposed SOT-MRAM computational sub-array,
4) The Bit-Counter readily counts the number of “1”s and passes it to the Shifter unit,
5) The Shifter left-shifts input data by specific number of bits, here “0001”, as result of Bit-Counter is left-shifted by 3-bit ($x2^{2+1}$) to “1000”,
6) Eventually, Sum unit adds the Shifter unit’s outputs to produce the output fmaps,
7) The output fmaps coming from convolutional layer can be later processed for down-sampling using average pooling performed using Sum and Shifter units.
DIGITAL PROCESSING UNIT (DPU)

- **Quantizer**: This unit quantizes a real number input $r_i$ [0, 1] to a $k$-bit number output $r_o$ [0, 1] using quantization Function:

  $$r_o = \frac{1}{2^k - 1} \text{round}((2^k - 1)r_i)$$

- **Batch-Norm.**: Batch Normalization layer alleviates the information loss during quantization by normalizing the input batch to have zero mean and unit variance.

  $$I_o(R) = \frac{I_i(R) - \mu}{\sqrt{\sigma^2 + \epsilon}} + \beta$$

- **Activ. Function**: The proper selection of activation function has a profound impact on network prediction accuracy specially in lower bit-width CNN. This unit can be reconfigured to perform two distinct activation functions ($\frac{\text{tanh}(x) + 1}{2}$, $\text{sign}(x)$).
OUTLINE

- Motivation
- STT-MRAM and SOT-MRAM
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion
PERFORMANCE EVALUATION

Device to System Level Simulations:

Device Level:
Verilog-A model of 3T SHE-MTJ device was developed to co-simulate with the interface CMOS circuits in SPICE to validate the functionality and evaluate performance of the proposed design.

Circuit Level:
45nm North Carolina State University (NCSU) Product Development Kit (PDK) [1] library is used in SPICE to verify the proposed design and evaluate the performance.

System Level:
We employ the modified self-consistent NVSim [2] along with an in-house developed C++ code to verify the performance of memory.

[1] www.eda.ncsu.edu/wiki/FreePDK45
PERFORMANCE EVALUATION

Accuracy:

- **Bit-width configuration**: Six bit-width configurations of W:I (32:32, 1:1, 1:2, 1:3, 1:4, and 2:2) are considered. The 8-bit gradient is applied to all configurations except 32:32.

- **Model**: A CNN with 6 (bit-wise) convolutional layers, 2 (average) pooling layers and 2 FC layers that cost about 80 FLOPs for a 40×40 image. FC layers are equivalently implemented by convolutions.

- **Data-set**: The SVHN. The cropped format of colored images (32×32) centered around each single digit is selected.

- **Training**: Modification on open source algorithm by DoReFa-Net, we adopt batch normalization and different dropout techniques to accelerate and avoid over-fitting. The model is trained on TensorFlow [1].

![Test error of CNN model for processing SVHN in different bit-width configuration and Evolution of prediction accuracy vs. epoch.](image)

- Complexity of inference and training are achieved using $W \times I$ and $W \times I + W \times G$, respectively.

- Experiments show that inputs and gradients are progressively more sensitive to bit-width changes.

- The accuracy in different configurations after modifications is almost matched with reported data in DoReFA-NET.

---

PERFORMANCE EVALUATION

Main Memory Storage:
- As shown, lower bit-width the CNN model is, less memory storage is required.
- AlexNet-BCNN mapped to our proposed accelerator requires 39.7MB memory storage which is $12\times$ and $6\times$ less compared to double precision (DP) and single precision (SP) CNNs.

Energy Consumption Estimation:
- The intensity of AND operations processed by IMCE makes the most fraction of energy consumption (up to 65%).
- $1^{st}$ un-quantized convolutional layer is the most computationally intensive one and the size of computations is diminished for next layers after quantization.
- According to system and application constraints, the designer can choose different configurations considering the accuracy and energy consumption trade-offs.
The experimental results show that the last two conv layers (converted from fully-connected layers) take up the most part of the area due to the high number of weight parameters.

- IMCE imposes 15.4% area overhead to original memory chip. It can be seen that modified row decoder and Sum unit contribute more than 60% of area overhead.

(a) Area distribution of conv. layers mapped to IMCE for processing a single image of SVHN, (b) Breakdown of area overhead of IMCE.
PERFORMANCE EVALUATION

Hardware Mapping Comparison:

- Comparison of two promising resistive memories (i.e. RRAM [8] and SOT-MRAM herein) over three different data-sets in terms of energy and area under 45nm technology node.

- The proposed accelerator exploiting bit-wise IMCE can process Binary CNN (BCNN) over different datasets very efficiently.

- It processes binary-weight AlexNet [1] for ImageNet favorably with 785.25μJ/img where ~ 3× and ~ 4× lower energy and area are achieved, respectively, compared to RRAM-based design.

![Performance Estimation of CNN and BCNN accelerators.](image)

<table>
<thead>
<tr>
<th>Designs</th>
<th>ImageNet</th>
<th>SVHN</th>
<th>MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Energy (μJ/img)</td>
<td>Area (mm²)</td>
<td>Energy (μJ/img)</td>
</tr>
<tr>
<td>CNN-RRAM [8]</td>
<td>5444.85</td>
<td>21.25</td>
<td>850.42</td>
</tr>
<tr>
<td>BCNN-RRAM [8]</td>
<td>2275.34</td>
<td>9.19</td>
<td>425.21</td>
</tr>
<tr>
<td>BCNN-SOT-MRAM</td>
<td>785.25</td>
<td>2.12</td>
<td>135.26</td>
</tr>
</tbody>
</table>


OUTLINE

- Motivation
- Spintronic Devices
- In-Memory Processing Platform based on SOT-MRAM
- In-Memory Convolution Engine
- Performance Evaluation
- Conclusion
CONCLUSION

- In this work, we develop a new in-memory processing architecture based on SOT-MRAM, which could be used as both non-volatile memory and reconfigurable in-memory logic.

- The new concept of bit-wise In-Memory Convolution Engine (IMCE) is then developed based on the proposed in-memory processing architecture achieving 3 goals:
  
  (1) All bitwise computation can be implemented within the proposed in-memory accelerator by eliminating massive energy consumption of data communication in traditional architecture between memory and computing units (i.e. CPU/GPU);

  (2) Reducing the energy consumption of convolutional layers through utilizing energy efficient intrinsic in-memory computation;

  (3) Accelerating the inference task by employing in-memory parallelism.

- Our accelerator can process low bit-width AlexNet on ImageNet data-set favorably with 785.25μJ/img, which consumes ~3× less energy than that of recent RRAM based counterpart. Besides, the chip area is ~4× smaller.
THANKS