SHE-MTJ Circuits for Convolutional Neural Networks

Andrew W. Stephan and Steven J. Koester, Fellow, IEEE

Abstract—We report the performance characteristics of a neural network model based on the proposed Multiply-Accumulate-Activate-Pool set, an MTJ-based spintronic circuit made to compute multiple neural functionalities in parallel. A study of image classification with the MNIST handwritten digits dataset using this network is provided via simulation. The effect of changing the weight representation precision, the severity of device process variation within the MAAP sets and the computational redundancy is provided. The emulated network achieves between 90 and 95% image classification accuracy at a cost of 100 nJ per image.

Index Terms—Neuromorphic Computing, Convolutional Neural Network, Spintronics, Spin Hall, Magnetic Tunnel Junction.

I. INTRODUCTION

Convolutional neural networks (CNNs) are a powerful tool for beyond-Boolean computing such as data classification, whether it be text, audio or visual. [1], [2] Their complexity makes in-hardware implementations difficult and costly beyond that of the simpler fully-connected network. We propose to use the Multiply-Accumulate-Activate-Pool (MAAP) sets described in [3] to reduce the complexity of convolutional neural network implementation. The MAAP sets will limit the number of unique operations required by the CNN by condensing the convolution, activation and pooling operations into one circuit. In so doing, the number of required peripheral operations such as memory is also reduced compared to other hardware-based implementations that incorporate all CNN functions individually [2].

II. BACKGROUND

A. Convolutional Neural Networks

Typical CNNs consist of one or more sequences of convolution, activation and pooling layers followed by one or more fully-connected (FC) layers as shown in Fig. 1. Each neuron in the convolutional layers applies a certain weight template to a subset of values in the input space, with neighboring neurons applying the same template to neighboring, possibly overlapping, subsets. Since multiple values in the input contribute to a single value in the convolution layer, this results in downsampling of the image size. Each template corresponds to one full set of neurons, or one convolutional image map. Activation layers pass each value contained in the convolutional layers through some nonlinear activation function in a 1-to-1 fashion. The rectified linear unit (ReLU)

\[ R(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \]  

(1)

is commonly used for this purpose. In max-pooling, each neuron chooses the maximum value from its unique subset of the input space. This further downsamples the data and also introduces some translation-invariance. The fully-connected layer consists of a one-dimensional vector of neurons, each of which takes a weighted sum of all values in the previous layer. The convolution, activation and pooling layers prior to the final fully-connected layer comprise a significant portion of the computational cost of a CNN.

B. MAAP Sets

The spin-torque-controlled magnetic tunnel junction (MTJ) is a well-known basic element of spintronic computing. [4]–[8] Depending on the circuit layout, geometry and specific application of spin torque, these versatile spin-MTJs can be used as analog or digital programmable synapse memristors, spiking neurons or artificial neurons. In [3] a useful application of MTJ cells manipulated via the spin-Hall effect (SHE) is proposed. Utilizing in-plane fieldlike spin-torque along the hard axis of the free layer (FL), a linear hysteresis loop is produced [9]. With the appropriate choice of circuit parameters, a voltage divider composed of one SHE-MTJ cell and a reference resistor with an inverter to read the output can produce a linear output—with saturation—as a function of the charge current passing through the SHE layer. This structure is effectively a three-terminal device with input, output and constant terminals in which the potential across the output and constant terminals depends on the charge current injected from the input terminal to the constant terminal. Crossbar arrays are commonly used in neuromorphics to perform the multiply-and-accumulate operation by summing up parallel currents, each of which represents a single product of a
Fig. 1. The CNN structure used for training in TensorFlow. The network consists of two convolutional layers with their subsequent activation and pooling layers, followed by one final fully-connected layer. Each convolution contains four kernels.

Fig. 2. The decomposition of the CNN in Fig. 1 to a MAAP-CNN. Each convolution-activation-pooling layer sequence has been replaced by a matching group of MAAP sets which perform all three functions.

III. MAAP-CNNs

We applied the MAAP set concept to the problem of classifying the MNIST handwritten digits dataset. The HSPICE/Matlab simulator used in [3], [10] was used to generate a simplified input-to-output simulation module, vastly decreasing the required compute time. This module is used in Matlab to emulate the CNN in Fig. 1 as shown in Fig. 2 using weight templates trained in Tensorflow. To store intermediate results in the course of CNN processing the MAAP data is sent to analog-to-digital converters (ADCs) and stored in static-RAM (SRAM). The convolutional templates are also digitally stored in SRAM. These quantities are digitally multiplied, requiring approximately $B^2$ gates for $B$ bits, and supplied to $B$ operational transconductance amplifier (OTA) based current sources that provide the weighted input to the MAAP sets in a fashion similar to the OTA usage in [3], [11], [12]. The low input impedance of the MAAP set circuit makes summing up parallel inputs in a crossbar-like manner quite accurate [3]. A block diagram of this layout is shown in Fig. 3. We note that with sufficiently large memory, only one MAAP set is needed to emulate an entire CNN. However, since the processing time for a single MAAP operation is on the order of ns [3] it is much more time-efficient to complete the ~4000 operations necessary with many sets (see Fig. 2).

IV. RESULTS AND DISCUSSION

In this section we report the relationship between terms such as the image classification accuracy, redundancy factor $R$, bit representation precision $B$, energy and process variation. In the figures referenced herein, the voltage deviation term refers to threshold deviation. The other process variation terms related to the MTJs are assumed to follow constant Gaussian...
distributions with values drawn from those distributions for each device in each iteration. The width of the Gaussian threshold distribution is a variable in some figures.

A. Energy

Apart from peripheral circuitry, the energy dissipation is largely independent of the number of physical MAAP circuits, being dependent on the number of MAAP and memory operations to be computed instead. The total energy dissipated during the processing of one image is:

$$E = N_M E_M + N_{MEM} E_{MEM} + N_A E_A + N_{MUL} E_{MUL},$$

(2)

where $N_M$, $N_{MEM}$, $N_A$ and $N_{MUL}$ are the number of MAAP, digital memory, ADC and digital multiply operations, respectively.

| TABLE I | Operation | Energy/Op | Iterations |
|---------|-----------|-----------|------------|
| MAAP    | 0.9 - 7.2 pJ | 3904 - $R$ |            |
| Memory  | 41.8 fJ    | 85328     |            |
| ADC     | 1 pJ       | 3904 - $B$ |            |
| Multiply| 125.4 fJ   | 3088 - $B^2$ |          |

Quantity of each operation needed to classify an image. $B$ and $R$ are the number of bits used for representation and the MAAP operation redundancy, respectively.
respectively. These quantities are given in Table I with the values ultimately deriving from the dimensions of the MAAP set arrays shown in Fig. 2. The energy terms $E_M$, $E_{MEM}$, $E_A$ and $E_{MUL}$ are the costs of a single instance of each operation. The MAAP operation cost $E_M$ accounts for leakage across the MTJ stacks, as calculated in [3] assuming a TMR of 1.5 and 40 $\times$ 40 nm$^2$ MTJs, as well as the static and dynamic dissipation of the OTAs [12]. $E_{MEM}$ is based on 16 nm node transistor data for 6T SRAM. [13] $E_{MUL}$ is estimated with the same transistor data, assuming $B^2$ gates per B-bit multiplication. The ADC energy is taken from the Stanford ADC Survey, with a power to Nyquist sampling frequency ratio of 1 pJ. [14] We note that the majority of the total energy is taken up by the digital processing (see Fig. 4). The energy usage depends upon the level of precision used to encode the weights and store the MAAP outputs. The assumed level of process variation that needs to be corrected and the MAAP circuit runtime have some effect as well. [3] Fig. 3(a) compares the MAAP and memory operational costs for 4-bit, 8-bit and 32-bit networks, assuming 0.6 ns runtime per MAAP set. The memory and digital operations dominate the MAAP processing cost. Fig. 4(b) shows the rise in total energy required to correct the error in the mean MAAP output due to process variation, assuming a 4-bit network. The dashed line assumes no variation, while the data points all correspond to a set level of MTJ variation with standard deviations matching those reported in [3], [15]–[17]. The standard deviation for the transistor threshold potentials is varied with multiple values. Although the process variation-induced mean error is easily dealt with, the process variation introduces a significant deviation in the error as well. In order to reduce this effect, redundant sampling is used. Each MAAP set output is measured $R$ times with the same inputs and the final stored value is the average of the measurements. The variation in the expected mean of $R$ redundant measurements is less than the variation of a single measurement by a factor of $\sqrt{R}$. Multiplying the number of MAAP operations by the redundancy factor $R$ comes at the cost of additional energy for MAAP sampling, and the need for at least $R$ physically distinct MAAP sets as the different measurements must be performed on different devices to ensure a unique random sampling each time. Fig. 5 shows the increase in total energy caused by operation redundancy.

B. Accuracy

In Fig. 6 we display the results pertaining to image classification accuracy. The values shown are statistical estimates. Each simulation included 100 test images and the values shown in the figure are the mean result of five or more simulations with error bars included. Fig. 6(a) indicates the accuracy vs. $B$ using non-ideal simulated devices with redundancy $R = 10$ to indicate the upper bound on performance. As the representation precision grows the accuracy increases from an average of about 90% to 95%. Fig. 6(b) shows the accuracy vs. $R$. With $R = 1$, the high error deviation introduced by process variation significantly lowers accuracy; however, with a modest $R = 5$ the accuracy is almost entirely recovered. At $R = 10$ the results are indistinguishable from those of ideal devices. Surprisingly, the particular level of transistor threshold deviation appears insignificant to the accuracy as indicated in Fig. 7. The results do not significantly differ between devices with 0 mV or 20 mV of threshold deviation. The existence of any significant process variation— in the MTJs, if not the transistors—is sufficient to necessitate some redundancy. However, increasing the level of variation has little to no incremental effect, although we surmise that extreme amounts beyond what was tested would cause further noticeable deterioration of the accuracy.

Finally, Fig. 8 indicates the classification accuracy vs. the energy cost for three different bit precision levels. Using a 4-bit network with $R = 5$ is sufficient to reach nearly 90% accuracy at a cost of about 80 nJ per image. The incremental cost of increasing accuracy to 95% is quite large, requiring 32-bit precision and $R = 10$ with a cost of about 1000 nJ per image.

C. Discussion

This work demonstrates that spintronic circuits based on the increasingly well-understood MTJ neuron model can be used to effectively implement the complex functions involved in CNNs and accurately classify images. The MAAP architecture condenses the many different CNN layers into fewer, and more cohesive sets of calculations. The modularity of the MAAP set model also detaches the physical circuitry from the number of conceptual neurons involved in the CNN model. Depending on $R$ and $N$ the energy required to process an image varies between about $40 - 1000$ nJ, comparable to the networks reviewed in [2]. At a maximum accuracy of between 90%–95% the performance is also comparable. We also note that, assuming sufficient devices to compute an entire sequence of layer operations in parallel, the fully charge-based network in [2] requires between 101.5 and 139 ns to calculate the output of a single convolution-ReLU-pooling sequence. By reducing the number of times the network must pass its intermediary outputs through the ADC to memory and back, we achieve a significant speed-up. Each MAAP set operation takes on the order of 1 ns, saving a great deal of time if we assume identical delays in the ADC and digital processing peripherals between the two networks. Finally, comparing the number of MAAP operations $N_M = 3904$ to the number of CeNN operations in an equivalent network in [2], [18] $N_C \sim 56000$ shows a significant reduction in complexity, even with redundancy. We also note that condensing the neuromorphic layer operations reduces the number of ADC and memory operations that are required to store and access intermediate data compared to a system which explicitly computes each operation, especially those which compute the operations via multiple sub-steps which themselves may require digital processing. This should yield additional savings. We hope this achievement will help spark continued interest and study of spintronic materials and circuits to unlock their great potential.

References

[1] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-Based Learning Applied to Document Recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, DOI:10.1109/5.726791.
[2] Q. Lou, C. Pan, J. McGuinness, A. Horvath, A. Naeemi and X. S. Hu, “A Mixed Signal Architecture for Convolutional Neural Networks,” ACM J. Emerg. Technol. Comput. Syst., vol. 15, no. 2, pp. 19:1–26, Mar. 2019, DOI:10.1145/3304410.

[3] A. W. Stephan and S. J. Koester, “Spin Hall MTJ Devices for Advanced Neuromorphic Functions,” IEEE Trans. Elec. Dev., vol. 67, no. 2, pp. 487–492, Jan. 2020, DOI:10.1109/TED.2019.2959732.

[4] W. Kang, Z. Wang, Y. Zhang, J.-O. Klein, W. Ly and W. Zhao, “Spintronic Logic Design Methodology Based on Spin Hall Effect-Driven Magnetic Tunnel Junctions,” J. Phys. D: Appl. Phys., vol. 49, pp. 065008–1–11, Jan. 2016, DOI:10.1088/0022-3727/49/6/065008.

[5] A. Sengupta and K. Roy, “Encoding Neural and Synaptic Functionalities in Electron Spin: A Pathway to Efficient Neuromorphic Computing,” Appl. Phys. Rev., vol. 4, no. 4, pp. 041105–1–25, Dec. 2017, DOI:10.1063/1.5012763.

[6] C. Pan and A. Naeemi, “A Proposal for Energy-Efficient Cellular Neural Network Based on Spintronic Devices,” IEEE Trans. Nanotech., vol. 15, pp. 820–827, Sept. 2016, DOI:10.1109/TNANO.2016.2598147.

[7] C. Pan and A. Naeemi, “Non-Boolean Computing Benchmarking for Beyond-CMOS Devices Based on Cellular Neural Network,” IEEE J. Expl. Sol.-Stat. Computat. Dev. and Circ., vol. 2 pp. 36–43, Nov. 2016, DOI:10.1109/JXICDC.2016.2633251.

[9] H. B. M. Saidouai and A. Manchon, “Spin-Swapping Transport and Torques in Ultrathin Magnetic Bilayers,” Phys. Rev. Lett., vol. 117, no. 3, pp. 036601–1–5, Jul. 2016, DOI:10.1103/PhysRevLett.117.036601.

[10] Predictive Technology Model. Accessed: Jul. 25, 2017. [Online]. Available: [http://ptm.asu.edu](http://ptm.asu.edu)

[11] Q. Lou, C. Pan, J. McGuinness, A. Horvath, A. Naeemi, M. Niemier and X. S. Hu, “A Mixed Signal Architecture for Convolutional Neural Networks,” ACM J. Emerg. Technol. Comput. Syst., vol. 15, no. 2, pp. 19:1–26, Mar. 2019, DOI:10.1145/3304410.

[12] Q. Lou, I. Palit, A. Horvath, X. S. Hu, M. Niemier and J. Nahas, “TFET-based Operational Transconductance Amplifier Design for CNN Systems,” in Proc. 25th Ed. Great Lakes Symp. VLSI, vol. 20–22, pp. 277–272, May 2015, DOI:10.1109/2742060.2742089.

[13] A. Pushkarna, S. Raghavan and H. Mahmoodi, “Comparison of Performance Parameters of SRAM Designs in 16nm CMOS and CNTFET Technologies,” 23rd IEEE Int. SOC Conf., pp. 339–342, Sept. 2010, DOI:10.1109/SOCC.2010.5784690.

[14] “Stanford ADC Survey.” [Online]. Available: [web.stanford.edu/~murmann/publications/ADCSurvey_rev20190802.xls](http://web.stanford.edu/~murmann/publications/ADCSurvey_rev20190802.xls). [Accessed 19 Dec 2019].

[15] W. Kang, L. Zhang, J.-O. Klein, Y. Zhang, D. Ravelosona, W. Zhao, “Reconfigurable Codesign of STT-MRAM Under Process Variations in Deeply Scaled Technology,” IEEE Trans. Elec. Dev., vol. 62, no. 6, pp. 1769–1777, Jun. 2015, DOI:10.1109/TED.2015.2412960.

[16] P. Wang, E. Eken, Z. W. Zhang, R. Joshi, R. Kanj and Y. Chen, “A Thermal and Process Variation Aware MTJ Switching Model and its Applications in Soft Error Analysis,” More than Moore Technologies for Next Generation Computer Design, Chapter 5, pp. 101–125, Springer New York, 2015.

[17] M. D. Giles, N. Arkali Radhakrishna, D. Becher, A. Kornfeld, K. Maurice, S. Mudanai, S Natarajan, P. Newman, P. Packan and T. Raksit, “High Sigma Measurement of Random Threshold Voltage Variation in 14nm Logic FinFET technology,” Proc. of VLSI Technology, pp. T150–T151, Aug. 2015, DOI:10.1109/VLSI.2015.7223657.

[18] A. W. Stephan, Q. Lou, M. T. Niemier, X. S. Hu and S. J. Koester, “Nonvolatile Spintronic Memory Cells for Neural Networks,” IEEE J. Expl. Sol.-Stat. Comput. Dev. and Circ., vol. 5, no. 2, pp. 67–73, Dec. 2019, DOI:10.1109/JXICDC.2019.2932992.