Implementation of a Binary Neural Network on a Passive Array of Magnetic Tunnel Junctions

Jonathan M. Goodwill1, Nitin Prasad2,3, Member, IEEE, Brian D. Hoskins1, Matthew W. Daniels1, Advait Madhavan2,4, Lei Wan5, Tiffany S. Santos5, Member, IEEE, Michael Tran5, Jordan A. Katine5, Fellow, IEEE, Patrick M. Braganca5, Senior Member, IEEE, Mark D. Stiles1, Fellow, IEEE, and Jabez J. McClelland1

1Physical Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, 20899, USA
2Associate, Physical Measurement Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, 20899, USA
3Department of Chemistry and Biochemistry, University of Maryland, College Park, MD, USA
4Institute for Research in Electronics and Applied Physics, University of Maryland, College Park, MD, USA
5Western Digital Research Center, Western Digital Corporation, San Jose, California, 95119, USA

Magnetic tunnel junctions (MTJs) provide an attractive platform for implementing neural networks because of their simplicity, non-volatility, and scalability. However, in hardware realizations, device variations, write errors, and parasitic resistance degrade performance. To quantify such effects, we perform inference experiments on a 2-layer perceptron constructed from a 15 × 15 passive array of MTJs, examining classification accuracy and write fidelity. Despite imperfections, we achieve median accuracy of 95.3 % with proper tuning of network parameters. The success of this tuning process shows that new metrics are needed to characterize and optimize networks reproduced in mixed signal hardware.

Index Terms—Magnetic tunnel junctions, neural networks, vector matrix multiplication, inference.

I. INTRODUCTION

The increasing scale of neural networks and their growing application space have produced demand for more energy- and memory-efficient artificial-intelligence-specific hardware [1]. Avenues to mitigate a key issue, the von Neumann bottleneck, include in-memory and near-memory architectures, as well as algorithmic approaches [2], [3]. To realize an in-memory approach and examine some of the practical issues involved, we leverage the inherently low-power, non-volatile, binary operation [4] of a passive array of magnetic tunnel junctions (MTJs) to demonstrate neural network hardware inference. Compared to active or transistor-integrated arrays, passive, transistorless arrays are potentially an even more efficient way to implement these networks, as they significantly reduce the overhead of transistor capacitances and could be implemented at significantly higher density, while freeing up space for additional transistors that might be needed in peripheral circuitry [5]. However, lack of selectors in these passive arrays makes it difficult to write individual devices in the presence of significant line resistances [6].

In this work, we implement a neural network on passive 15 × 15 crossbar arrays of MTJs [7], as shown in Fig. 1(a). Using the Wine dataset [8], we demonstrate high inference accuracies even in the presence of hardware imperfections such as device-to-device variations and line resistances. This dataset consists of 178 samples of wine with each sample consisting of 13 recorded characteristics (for example, alcohol concentration, color intensity, etc.) and an associated label for the cultivar from which the wine was produced.

The architecture of the implemented neural network is...
We obtain a median accuracy of 95.3 %, which is lower than formed better than others. Over these 300 different solutions, training into the MTJ array. As expected, certain solutions per-
unique weight matrix solutions obtained through offline-
and 93.3 % on the test set. We characterize the network performance by writing 300 unique weight matrix solutions obtained through offline-training into the MTJ array. As expected, certain solutions performed better than others. Over these 300 different solutions, we obtain a median accuracy of 95.3 %, which is lower than the median offline-trained accuracy of 99.3 %. These results demonstrate that a high-accuracy inference binary neural network can be realized using a non-ideal passive MTJ hardware array. This high level of performance is obtained after optimizing the normalization parameter $g_{\text{norm}}$ (see Fig. 1(c)) which maps conductances to dimensionless weights. Surprisingly, the value of normalization conductance that minimizes the root-mean-square deviation between the ideal weights and the realized weights is not the same value that maximizes the median classification accuracy over all 300 weight solutions. We reproduce this mismatch in SPICE simulations in which we model the MTJ array using passive elements and include both independently measured device-device variations and line resistances. This finding provides insight into the problem of embedded inference with MTJ-based hardware accelerators and is an integral step forward on the pathway toward large-scale integration of hardware devices with imperfections and variations.

**ACKNOWLEDGEMENT**

We acknowledge Michael Grobis, Chris Petti, and Alexei Bogdanov from Western Digital for their contributions in the design of the MTJ arrays used in this study. This work was funded by the National Institute of Standards and Technology. Nitin Prasad is supported by Quantum Materials for Energy Efficient Neuromorphic Computing, an Energy Frontier Research Center funded by the U.S. DOE, Office of Science, Basic Energy Sciences, under Award DE-SC0019273. Advaith Madhavan acknowledges support under the Cooperative Research Agreement Award No. 70NANB14H209, through the University of Maryland. This work benefited in part from materials and software developed under DARPA / ONR grant No. N00014-20-1-2031.

**REFERENCES**

[1] M. Capra, B. Bussolino, A. Marchisio, G. Masera, M. Martina, and M. Shafique, “Hardware and software optimizations for accelerating deep neural networks: Survey of current trends, challenges, and the road ahead”, *IEEE Access* 8 (2020).
[2] G. W. Burr et al., “Neuromorphic computing using non-volatile memory”, *Advances in Physics: X* 2, 89 (2017).
[3] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, “Neuromorphic computing using non-volatile memory”, *IEEE High Performance Extreme Computing Conference (HPEC)* 1 (2020).
[4] D. Apalkov, et al., “Spin-transfer torque magnetic random access memory (STT-MRAM)”, *ACM Journal on Emerging Technologies in Computing Systems* 9, 1 (2013).
[5] Q. Xia and J. J. Yang, “Memristive crossbar arrays for brain-inspired computing” *Nature Materials*, 18 4 (2019).
[6] Jiantao Zhou, Kuk-Hwan Kim, and Wei Lu, “Crossbar RRAM Arrays: Selector Device Requirements During Read Operation” *IEEE Transactions on Electron Devices* 61, 5 (2014).
[7] J. M. Goodwill, et al., “Implementation of a binary neural network on a passive array of magnetic tunnel junctions” *Physical Review Applied*, 18, 1 (2022).
[8] UCI Machine Learning Repository: Wine Data Set.