Spin Electronics

Implementation of Artificial Neural Networks using Magnetoresistive Random-Access Memory-based Stochastic Computing Units

Yixin Shao(邵奕鑫)[1], Sisilia Lamsari Sinaga [1], Idris O. Sunmola[1], Andrew S. Borland[1], Matthew J. Carey[2], Jordan A. Katine[2]**, Victor Lopez-Dominguez[1], Pedram Khalili Amiri[1]*

[1] *Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208, USA.*
[2] *Western Digital Corporation, San Jose, CA 95119, USA.*
** Senior Member, IEEE*
*** Fellow, IEEE*

Received 1 Apr 2020, revised 15 Apr 2020, accepted 20 Apr 2020, published 1 Jun 2020, current version 15 Jun 2020. (Dates will be inserted by IEEE; "published" is the date the accepted preprint is posted on IEEE Xplore®; "current version" is the date the typeset version is posted on Xplore®).

Abstract— Hardware implementation of Artificial Neural Networks (ANNs) using conventional binary arithmetic units requires large area and energy, due to the massive multiplication and addition operations in the inference process, limiting their use in edge computing and emerging Internet of Things (IoT) systems. Stochastic computing (SC), where the probability of 1s and 0s in a randomly generated bit-stream is used to represent a decimal number, has been proposed as an alternative for compact and low-energy arithmetic hardware, due to its ability to implement basic arithmetic operations using far fewer logic gates than binary operations. To realize SC in hardware, however, tunable true random number generators (TRNGs) are needed, which cannot be efficiently realized using existing CMOS technology. Here we address this challenge by using magnetic tunnel junctions (MTJs) as TRNGs, the stochasticity of which can be tuned by an electric current via spin-transfer torque. We demonstrate the implementation of ANNs with SC units, using stochastic bit-streams experimentally generated by a series of 50 nm perpendicular MTJs. The numerical value (1 to 0 ratio) of the bit-streams is tuned by the current through the MTJs via spin-transfer torque, with an ultralow current of < 5 μA (= 0.25 MA cm^{-2}). The MTJ-based SC-ANN achieves 95% accuracy for handwritten digit recognition on the MNIST database. MRAM-based SC-ANNs provide a promising solution for ultra-low-power machine learning in edge, mobile and IoT devices.

Index Terms—Spin Electronics, Artificial Neural Network, Stochastic Computing, MRAM

I. INTRODUCTION

Machine learning in portable systems and edge devices is emerging as a critical enabler of new applications in internet of things (IoT) [Li 2018], autonomous driving [Chen 2015, Sallab 2017, Shalev-Shwartz 2016], health [Beam and Kohane 2018, Farrar and Worden 2012, Ravì 2016], wearables [Hammerla 2016], augmented/virtual reality (AR/VR) [Wu 2019] and other areas. However, existing hardware implementations of Artificial Neural Networks (ANNs) using conventional binary arithmetic units require large area and energy, due to the massive multiplication and addition operations in the inference process [Li 2017, Li 2017, Ren 2017, Sim and Lee 2017]. This limits their use in low-power portable systems, edge, and IoT devices.

Stochastic computing (SC) [Li 2017, Li 2017, Ren 2017, Sim and Lee 2017, Gaines 1969, Brown and Card 2001, Wang 2017, Lv and Wang 2017, Li 2018, Daniels 2020, Kim 2016] has been proposed as an alternative for compact and low-energy arithmetic hardware. SC uses the probability of 1s or 0s in a randomly generated bit-stream to represent a decimal number. This allows it to implement basic arithmetic operations using fewer logic gates than binary operations. However, to efficiently realize SC in hardware, a key requirement is the existence of tunable true random number generators (TRNGs), which cannot be efficiently realized using existing CMOS technology. As an example, a conventional 32-bit linear feedback shift register (LFSR) used for RNG operation in CMOS requires more than 1000 transistors [Borders 2019].

Here we address this challenge by using a series of magnetoresistive random-access memory (MRAM) bits – i.e. magnetic tunnel junctions (MTJs) [Daniels 2020, Nishimura 2002, Ikeda 2010, Gallagher 1997, Mizrahi 2018, Mizrahi 2016, Vodenicarevic 2017] – as TRNGs. The TRNG operation is based on the thermal fluctuations at room temperature of the MTJ free layer [Camsari 2017, Brown Jr 1963, Fukushima 2014]. The stochasticity of this process can be tuned by an ultralow current of < 5 μA (= 0.25 MA cm^{-2}) via spin-transfer torque (STT) [Fuchs 2004], to generate tunable stochastic bit-streams representing the entire range of numbers from -1 to 1. By using the bit-streams that are experimentally generated from these MTJs, we demonstrate a SC-based ANN using MTJ-TRNGs that performs handwritten digit recognition on the MNIST database [Li 2018, Daniels 2020, Lecun 1999] with accuracy of 95% using a 1024-bit stochastic bit-stream length.

II. GENERATION OF BIT-STREAMS

Corresponding author: Pedram Khalili (pedram@northwestern.edu). Digital Object Identifier: 10.1109/LMAG.XXXX.XXXXXXX (inserted by IEEE).

A. Device Structure and Physical Mechanism

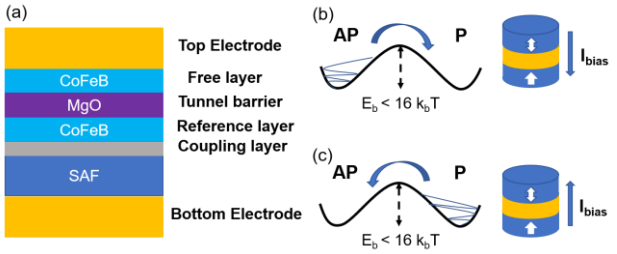

Fig. 1. (a) Cross-section illustration of the MTJs. The devices have a bottom-pinned configuration with a Co/Pt multilayer-based synthetic antiferromagnetic (SAF) structure. (b)-(c) Energy diagram for a stochastic MTJ under bias current. Spin-transfer torque (STT) acting on the MTJ free layer in (b) favors the parallel state, while in (c) it favors the antiparallel state.

The structure of the perpendicular MTJs used in this work is illustrated in Fig. 1(a). The MTJ consists of two ferromagnetic layers separated by an oxide layer. Depending on the direction of the magnetization in the two ferromagnetic layers, the device has a low-resistance parallel state (P) and a high-resistance antiparallel state (AP) [Yuasa and Djayaprawira 2007, Yuasa 2004, Yuasa 2004], resulting in a tunnel magnetoresistance (TMR) ratio of ~ 130% and a parallel-state resistance-area (RA) product of ~ 440 Ω-μm^2. Our devices were circular and had a diameter of 50 nm.

The two states of an MTJ are separated by an energy barrier E_b which is proportional to the free layer volume and anisotropy. The retention time can be written as $\tau = \tau_0 \exp(E_b/k_BT)$, where τ_0 is the characteristic attempt time (on the order of 1 ns), k_B is the Boltzmann constant and T is temperature. For a large MTJ where E_b is large enough, the retention time is long resulting in nonvolatile memory operation. In the present experiment, however, the free layer thickness and anisotropy were adjusted so that the retention time was reduced to ~ 5 ms, corresponding to an energy barrier < 16 k_BT. As a result, the MTJ stochastically switched between its two states at room temperature due to thermal fluctuations. In the presence of a current, one state or the other would be preferred by STT [Fuchs 2004], as shown in Fig. 1(b)-(c).

B. Measurement of Bit-streams

Stochastic bit-streams were generated experimentally by measuring the resistance of the MTJs in time domain under different voltage bias conditions. In total, six nominally similar MTJs with a diameter of 50 nm were used in this work, as explained in detail below. The resistance of a representative 50 nm MTJ as a function of external magnetic field, under different bias voltages is shown in Fig. 2(a)-(c). An offset field of approximately -35 mT was observed in the loop measured at 1 mV, due to the stray field from the uncompensated reference layer. The MTJ did not show a significant coercivity, consistent with its small energy barrier. Due to the STT effect, the offset field shifted in opposite directions depending on the applied bias voltage. Based on this, we fixed the external magnetic field at -35 mT in our experiment and measured the resistance under different bias voltages for a period of ~ 2 minutes, in intervals of 100 ms, which provided ~ 1200 data points for each voltage. Fig. 2(d)-(f) show the results under three different bias voltages applied to the MTJ. Tunability from > 95% AP to > 95% P was experimentally achieved

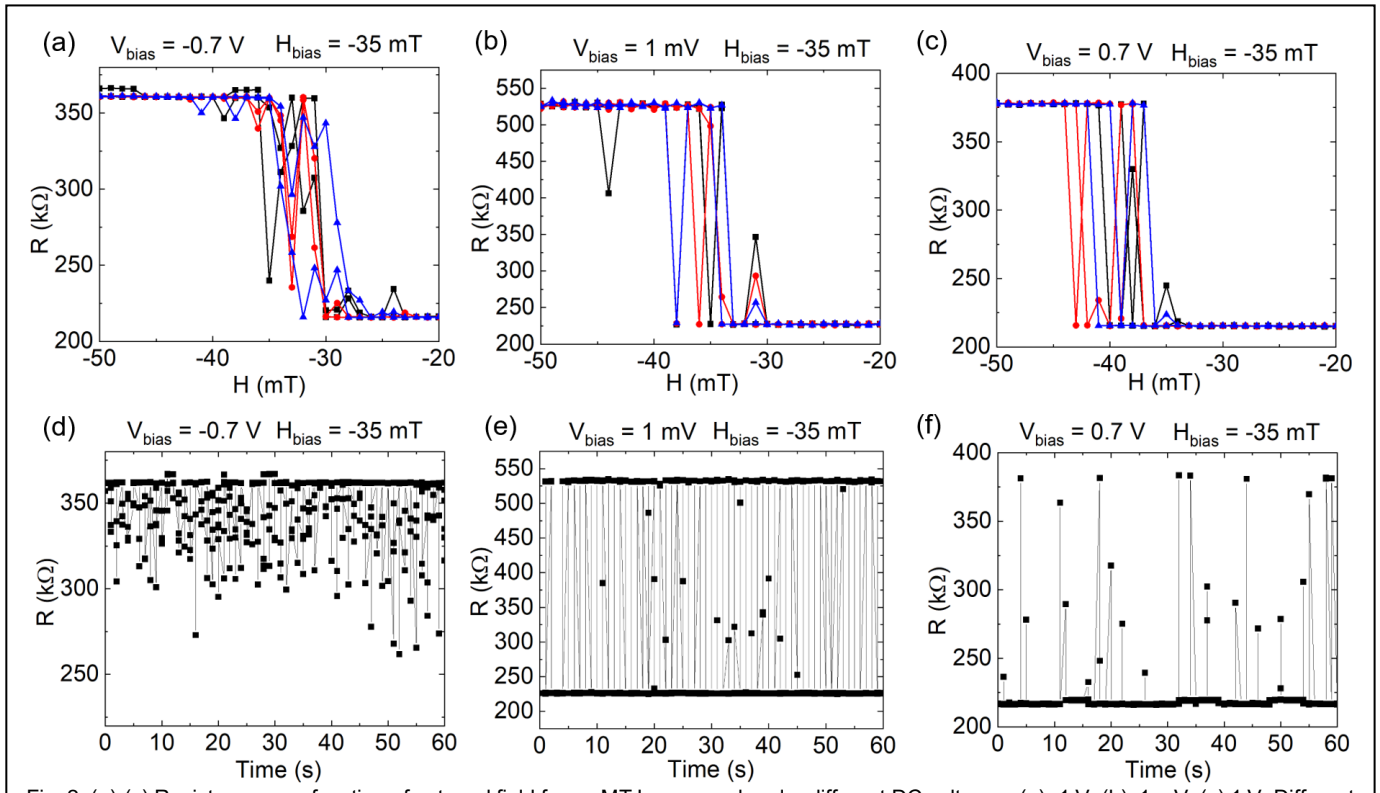

Fig. 2. (a)-(c) Resistance as a function of external field for an MTJ measured under different DC voltages: (a) -1 V, (b) 1 mV, (c) 1 V. Different colors represent different measurement repetitions. (d)-(f) MTJ resistance oscillations measured as a function of time, under a fixed external magnetic field H = -350 Oe but different bias voltages: (d) V = -1 V, (e) V = 1 mV and (f) V = 1 V.

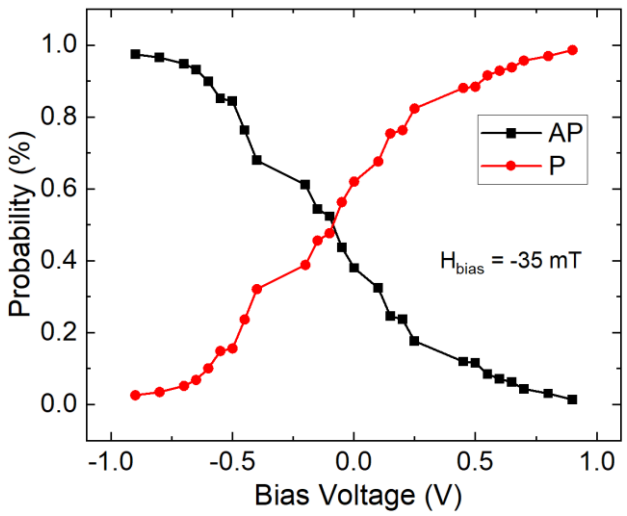

Fig. 3. Probabilities of 1s and 0s (parallel and antiparallel states) generated by an MTJ under different bias voltages.

by a voltage less than 1 V, as shown in Fig. 3, corresponding to an ultralow current less than 5 μA (= 0.25 MA cm^{-2}). Using this procedure, bit-streams were generated representing the entire range of numbers from -1 to 1.

III. MTJ-BASED STOCHASTIC COMPUTING UNITS

In the SC paradigm, numbers are represented by the probability of 1s in a bit-stream [Gaines 1969, Brown and Card 2001]. In this work, bipolar mapping was used to map real numbers x within the range of [-1, 1], to bit-streams X, via the relation $P(X=1) = (x+1)/2$. Using this approach, the key arithmetic operations in the ANN were implemented as follows.

Multiplication was implemented with an XNOR gate [Brown and Card 2001], as shown in Fig. 4(a). The output of the XNOR gate is $P(Y) = P(A) \cdot P(B) + \overline{P(A)} \cdot \overline{P(B)}$. For bipolar mapping, this can be rewritten as $(y + 1) / 2 = [(a + 1) / 2][(b + 1) / 2] + [1 - (a + 1) / 2][1 - (b + 1) / 2]$, which can be reduced to $y=ab$.

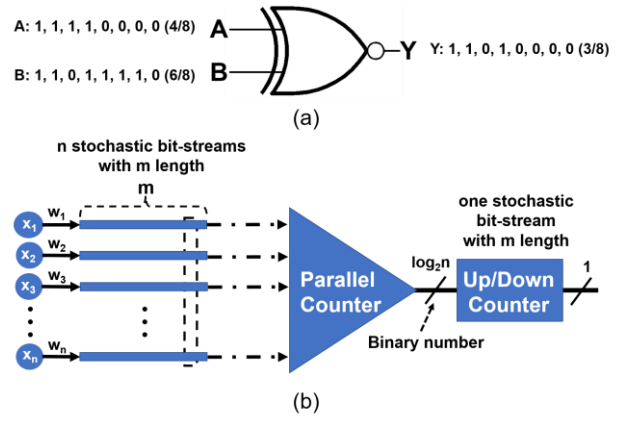

Fig. 4. (a) Stochastic multiplication using bipolar mapping within the [-1, 1] range. (b) APC-based neuron for stochastic dot product and activation functions.

The addition and the following activation operation were implemented by an approximate parallel counter (APC)-based neuron design, following an approach similar to [Kim 2016]. As shown in Fig. 4(b), the multiplication of n inputs and weights was performed through XNOR gates as described above, which resulted in n bit-streams with bit-stream length m. The addition was then done by the APC, where the sum of 1s in each column (dashed square in Fig. 4(b)) was accumulated. However, since the output from the APC was a binary number, to convert it again into a stochastic bit-stream, a saturated up/down counter was exploited to approximate a hyperbolic tangent function $Btanh(n, K, x) \approx \tanh(x)$ [Kim 2016], where K is the number of states for the saturated counter and $K = 2n$ in this work. This is similar to a finite state machine, except that the amount of increase or decrease for the states in each cycle is decided by the counted number in the APC for each column. Given K states in the counter, half of them generate 0 and the other half generate 1. The output bit-stream is thus an approximation of the hyperbolic tangent of the result of the dot product.

IV. IMPLEMENTATION OF SC-ANN

A. Training

The ANN architecture demonstrated in this work had one hidden layer with 128 neurons, as shown in Fig. 5. The inputs were grayscale images of handwritten digits from the MNIST database [Lecun 1999], whose values were pre-scaled to [0, 1] to be compatible with the stochastic bit-streams. The ANN parameters (weights and biases) were trained using TensorFlow [Abadi 2016], on floating point numbers with 32 bits, during which L-2 regularization was employed to ensure the trained weights and biases also sat within the [-1, 1] range. The resulting training accuracy was 97%.

B. Inference on Stochastic Computing ANN

The inference process was then performed using the stochastic computing approach discussed above, by mapping the inputs and trained parameters to corresponding stochastic bit-streams, which were generated experimentally by the 50 nm MTJs. For each MTJ, data under ~30 different bias voltages were obtained, resulting in ~30 different bit-streams per MTJ. The products (XNOR) of every two MTJs were then used to generate bit-stream sets with deeper number resolution. Furthermore, to make sure the bit-streams involved in each operation were statistically independent of each other, data from different pairs of MTJs were used to map the values for inputs and weights in different layers of the SC-ANN. Thus, six MTJs in total were used where each two of them were responsible for one of the three statistically independent bit-stream sets used in the network.

It is worth noting that as a consequence of the relatively small number of MTJs that generate our bit-streams, the number of synaptic weights in each layer is still much larger than the number of sampled bit-streams. To reduce the resulting correlations of bit-streams of the same value in the same layer, we additionally used one of our MTJs to introduce a random reshuffling mechanism. The 1024-bit long bit-streams were cut into eight segments, where each one of them was 128-bit long. Each time when a number was to be mapped by the corresponding bit-stream, the bit-stream was rotated and restarted from the i-th segment, where i is a random integer from 0 to 7. Importantly, to generate the random integers from 0 to 7 with the same probability, a bit-stream with 50% probabilities of 1s and 0s from one of the MTJs was used. In principle, the need for this reshuffling

mechanism is eliminated in real applications when the bit-streams are generated in real time.

The structure for the resulting SC-ANN is shown in Fig. 5. The result of inference with SC on 1024-bit long bit-streams is shown in Fig. 6(a) in the form of a confusion matrix. The numbers of correct

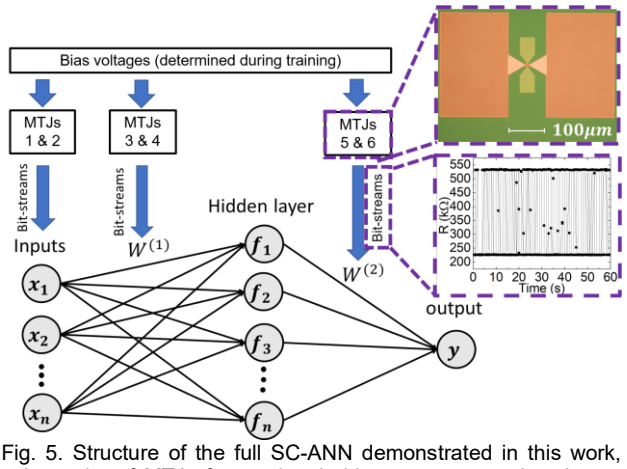

Fig. 5. Structure of the full SC-ANN demonstrated in this work, using pairs of MTJs for stochastic bit-stream generation. Insets show the photograph of one of the MTJ devices used (upper right inset) and the stochastic data measured (lower right inset). The big and small rectangular pads in the upper right inset are the bottom and top electrodes, respectively.

and incorrect classifications are summarized and normalized for each class. It can be seen that the ANN successfully classifies the handwritten digits.

Classification accuracy for the inference run with SC using bit-streams of different lengths is shown in Fig. 6(b). It is evident that longer bit-streams provide better classification accuracy, which is understandable because the accuracy of each bit-stream is proportional to its length.

It is worth comparing the proposed SC-ANN to recent works on CMOS [[Li 2018]] and hybrid spintronic-CMOS [Daniels 2020] SC-based neural networks and RNGs [Zhakatayev 2018] in terms of energy dissipation. Specifically, while the circuit design and simulation of a complete SC-ANN are beyond the scope of the present paper, here we focus on comparing the performance of our MTJ-based TRNG to the RNGs discussed in these recent reports.

For a conventional CMOS-based LFSR RNG, the energy per bit is on the order of ~10 fJ [Daniels 2020]. In our case, the energy dissipation of the TRNG depends on the retention time τ of the MTJs, which itself is determined by the energy barrier E_b. Although the retention time reported in this work is relatively long, it could in principle be reduced by reducing the perpendicular magnetic anisotropy or reducing the diameter of the MTJs. Assuming a reduction of the diameter of our MTJs from 50 nm to 20 nm, we would expect a ~ 6.25× reduction of the free layer volume, which results in $E_b \sim 2.5\ k_BT$. This corresponds to a reduction of the retention time (and associated increase of the bit generation rate) to $\tau \sim 10$ ns. We note that this is a conservative estimate, and recent works have indicated the possibility of retention times even smaller than 1 ns [Kaiser 2019, Hassan 2019, Desplat and Kim 2020]. Nonetheless, even with $\tau \sim 10$ ns, the energy per bit reduces to ~ 20 fJ assuming an applied voltage of ~1 V and device resistance of 500 kΩ, which is comparable to CMOS-only RNGs [Li 2018, Zhakatayev 2018]. It is worth noting that the analog control circuitry required for biasing the MTJs will add to this energy consumption. This contribution, which would depend on the required bit-stream resolution and number of MTJs, is likely highly dependent on details of the circuit implementation and is therefore not quantified here.

It is also worth comparing this type of TRNG to the MTJ-based TRNGs proposed in [Daniels 2020]. The latter approach uses a digitally controlled circuit to convert the oscillations of a superparamagnetic MTJ into stochastic bit-streams. This is qualitatively different from the TRNG discussed in the present paper, which is essentially analog (similar to the circuits previously proposed for probabilistic (p-) bit generation [Camsari 2017]), thus representing different tradeoffs and suitable application scenarios. Firstly, the energy dissipation of the pre-charge sense amplifier (PCSA) method used in [Daniels 2020] is essentially independent of the MTJ device size, in contrast to the present approach where the switching rate directly affects the energy dissipation. Hence, while the PCSA approach is expected to provide superior energy efficiency for longer clock cycles (150 ns in [Daniels 2020]), as clock speed is increased, one can expect the analog TRNG approach to achieve similar, if not

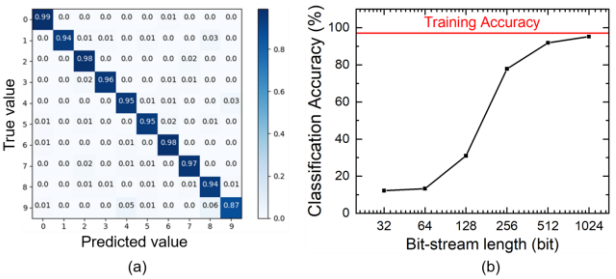

Fig. 6. (a) Confusion matrix of the result of the inference operation, with stochastic computing on 1024-bit long bit-streams. (b) Classification accuracy achieved on the SC-ANN using different stochastic bit-stream lengths.

better, energy efficiency. A second difference is that in the analog TRNG used in this work, the representation accuracy can be controlled by the length of the bit-streams. On the other hand, for the PCSA method, the representation accuracy is determined by the number of programmable bits in the bit-stream generators, thus determined by the number of transistors and MTJs in the circuit. Hence, for the same representation accuracy, the present method is likely to have an overall lower component count. Finally, we note that while the work by [Daniels 2020] showed a similar increase of the accuracy with bit-stream length as that shown in Fig. 6, it achieved comparable accuracies with shorter bit-streams than in the present work, which is a consequence of the more complex LeNet5 neural network structure with six hidden layers that was considered in [Daniels 2020].

V. CONCLUSION

We have demonstrated MRAM-based SC-ANNs which successfully classify handwritten digits with accuracy up to 95%. The SC-ANNs use experimentally measured stochastic bit-streams generated by 50 nm MTJ-based TRNGs that are tuned by an ultralow electric current (< 5 μA). The accuracy of the classification can be

adjusted in real time by changing the length of the bit-streams. Our results provide a promising solution for ultra-low-power machine learning in edge, mobile and IoT devices.

ACKNOWLEDGMENT

This work was supported by a grant from the National Science Foundation, Division of Industrial Innovation and Partnerships (NSF IIP-1919109).

REFERENCES

H. Li, K. Otaand M. Dong, "Learning IoT in edge: Deep learning for the Internet of Things with edge computing," *IEEE network,* vol. 32, no. 1, pp. 96-101, 2018.

C. Chen, A. Seff, A. Kornhauserand J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2722-2730.

A. E. Sallab, M. Abdou, E. Perotand S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging,* vol. 2017, no. 19, pp. 70-76, 2017.

S. Shalev-Shwartz, S. Shammahand A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295,* 2016.

A. L. Beam and I. S. Kohane, "Big data and machine learning in health care," *Jama,* vol. 319, no. 13, pp. 1317-1318, 2018.

C. R. Farrar and K. Worden, *Structural health monitoring: a machine learning perspective*. John Wiley & Sons, 2012.

D. Ravì, C. Wong, F. Deligianni, M. Berthelot, J. Andreu-Perez, B. Loand G.-Z. Yang, "Deep learning for health informatics," *IEEE journal of biomedical and health informatics,* vol. 21, no. 1, pp. 4-21, 2016.

N. Y. Hammerla, S. Halloranand T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," *arXiv preprint arXiv:1604.08880,* 2016.

C.-J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jiaand B. Jia, "Machine learning at facebook: Understanding inference at the edge," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2019: IEEE, pp. 331-344.

J. Li, A. Ren, Z. Li, C. Ding, B. Yuan, Q. Qiuand Y. Wang, "Towards acceleration of deep convolutional neural networks using stochastic computing," in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2017: IEEE, pp. 115-120.

J. Li, Z. Yuan, Z. Li, C. Ding, A. Ren, Q. Qiu, J. Draperand Y. Wang, "Hardware-driven nonlinear activation for stochastic computing based deep convolutional neural networks," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017: IEEE, pp. 1230-1236.

A. Ren, Z. Li, C. Ding, Q. Qiu, Y. Wang, J. Li, X. Qianand B. Yuan, "Sc-dcnn: Highly-scalable deep convolutional neural network using stochastic computing," *ACM SIGPLAN Notices,* vol. 52, no. 4, pp. 405-418, 2017.

H. Sim and J. Lee, "A new stochastic computing multiplier with application to deep convolutional neural networks," in *Proceedings of the 54th Annual Design Automation Conference 2017*, 2017, pp. 1-6.

B. R. Gaines, "Stochastic computing systems," in *Advances in information systems science*: Springer, 1969, pp. 37-172.

B. D. Brown and H. C. Card, "Stochastic neural computation. I. Computational elements," *IEEE Transactions on computers,* vol. 50, no. 9, pp. 891-905, 2001.

S. Wang, S. Pal, T. Li, A. Pan, C. Grezes, P. Khalili-Amiri, K. L. Wangand P. Gupta, "Hybrid VC-MTJ/CMOS non-volatile stochastic logic for efficient computing," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 2017: IEEE, pp. 1438-1443.

Y. Lv and J.-P. Wang, "A single magnetic-tunnel-junction stochastic computing unit," in *2017 IEEE International Electron Devices Meeting (IEDM)*, 2017: IEEE, pp. 36.2. 1-36.2. 4.

Z. Li, J. Li, A. Ren, R. Cai, C. Ding, X. Qian, J. Draper, B. Yuan, J. Tangand Q. Qiu, "HEIF: Highly efficient stochastic computing-based inference framework for deep neural networks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 38, no. 8, pp. 1543-1556, 2018.

M. W. Daniels, A. Madhavan, P. Talatchian, A. Mizrahiand M. D. Stiles, "Energy-efficient stochastic computing with superparamagnetic tunnel junctions," *Physical Review Applied,* vol. 13, no. 3, p. 034016, 2020.

K. Kim, J. Kim, J. Yu, J. Seo, J. Leeand K. Choi, "Dynamic energy-accuracy trade-off using stochastic computing in deep neural networks," in *Proceedings of the 53rd Annual Design Automation Conference*, 2016, pp. 1-6.

W. A. Borders, A. Z. Pervaiz, S. Fukami, K. Y. Camsari, H. Ohnoand S. Datta, "Integer factorization using stochastic magnetic tunnel junctions," *Nature,* vol. 573, no. 7774, pp. 390-393, 2019.

N. Nishimura, T. Hirai, A. Koganei, T. Ikeda, K. Okano, Y. Sekiguchiand Y. Osada, "Magnetic tunnel junction device with perpendicular magnetization films for high-density magnetic random access memory," *Journal of applied physics,* vol. 91, no. 8, pp. 5246-5249, 2002.

S. Ikeda, K. Miura, H. Yamamoto, K. Mizunuma, H. Gan, M. Endo, S. Kanai, J. Hayakawa, F. Matsukuraand H. Ohno, "A perpendicular-anisotropy CoFeB–MgO magnetic tunnel junction," *Nature materials,* vol. 9, no. 9, pp. 721-724, 2010.

W. J. Gallagher, J. H. Kaufman, S. S. P. Parkinand R. E. Scheuerlein, "Magnetic memory array using magnetic tunnel junction devices in the memory cells," ed: Google Patents, 1997.

A. Mizrahi, T. Hirtzlin, A. Fukushima, H. Kubota, S. Yuasa, J. Grollierand D. Querlioz, "Neural-like computing with populations of superparamagnetic basis functions," *Nature communications,* vol. 9, no. 1, pp. 1-11, 2018.

A. Mizrahi, N. Locatelli, R. Lebrun, V. Cros, A. Fukushima, H. Kubota, S. Yuasa, D. Querliozand J. Grollier, "Controlling the phase locking of stochastic magnetic bits for ultra-low power computation," *Scientific reports,* vol. 6, no. 1, pp. 1-7, 2016.

D. Vodenicarevic, N. Locatelli, A. Mizrahi, J. S. Friedman, A. F. Vincent, M. Romera, A. Fukushima, K. Yakushiji, H. Kubotaand S. Yuasa, "Low-energy truly random number generation with superparamagnetic tunnel junctions for unconventional computing," *Physical Review Applied,* vol. 8, no. 5, p. 054045, 2017.

K. Y. Camsari, S. Salahuddinand S. Datta, "Implementing p-bits with embedded MTJ," *IEEE Electron Device Letters,* vol. 38, no. 12, pp. 1767-1770, 2017.

W. F. Brown Jr, "Thermal fluctuations of a single-domain particle," *Physical review,* vol. 130, no. 5, p. 1677, 1963.

A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasaand K. Ando, "Spin dice: A scalable truly random number generator based on spintronics," *Applied Physics Express,* vol. 7, no. 8, p. 083001, 2014.

G. Fuchs, N. Emley, I. Krivorotov, P. Braganca, E. Ryan, S. Kiselev, J. Sankey, D. Ralph, R. Buhrmanand J. Katine, "Spin-transfer effects in nanoscale magnetic tunnel junctions," *Applied Physics Letters,* vol. 85, no. 7, pp. 1205-1207, 2004.

Y. Lecun, C. Cortesand C. Burges, "The MNIST Dataset of Handwritten Digits(Images)," ed, 1999.

S. Yuasa and D. Djayaprawira, "Giant tunnel magnetoresistance in magnetic tunnel junctions with a crystalline MgO (0 0 1) barrier," *Journal of Physics D: Applied Physics,* vol. 40, no. 21, p. R337, 2007.

S. Yuasa, A. Fukushima, T. Nagahama, K. Andoand Y. Suzuki, "High tunnel magnetoresistance at room temperature in fully epitaxial Fe/MgO/Fe tunnel junctions due to coherent spin-polarized tunneling," *Japanese Journal of Applied Physics,* vol. 43, no. 4B, p. L588, 2004.

S. Yuasa, T. Nagahama, A. Fukushima, Y. Suzukiand K. Ando, "Giant room-temperature magnetoresistance in single-crystal Fe/MgO/Fe magnetic tunnel junctions," *Nature materials,* vol. 3, no. 12, pp. 868-871, 2004.

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irvingand M. Isard, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265-283.

A. Zhakatayev, K. Kim, K. Choiand J. Lee, "An efficient and accurate stochastic number generator using even-distribution coding," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems,* vol. 37, no. 12, pp. 3056-3066, 2018.

J. Kaiser, A. Rustagi, K. Y. Camsari, J. Z. Sun, S. Dattaand P. Upadhyaya, "Subnanosecond fluctuations in low-barrier nanomagnets," *Physical Review Applied,* vol. 12, no. 5, p. 054056, 2019.

O. Hassan, R. Faria, K. Y. Camsari, J. Z. Sunand S. Datta, "Low-barrier magnet design for efficient hardware binary stochastic neurons," *IEEE Magnetics Letters,* vol. 10, pp. 1-5, 2019.

L. Desplat and J.-V. Kim, "Entropy-reduced retention times in magnetic memory elements: A case of the Meyer-Neldel Compensation Rule," *Physical Review Letters,* vol. 125, no. 10, p. 107201, 2020.