Low-Energy Deep Belief Networks using Intrinsic Sigmoidal Spintronic-based Probabilistic Neurons

Ramtin Zand[1], Kerem Yunus Camsari[2], Steven D. Pyle[1], Ibrahim Ahmed[3],
Chris H. Kim[3], and Ronald F. DeMara[1]
[1]Department of Electrical and Computer Engineering, University of Central Florida, Orlando, FL, 32816 USA
[2]Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, 47906 USA
[3]Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA

ABSTRACT

A low-energy hardware implementation of deep belief network (DBN) architecture is developed using near-zero energy barrier probabilistic spin logic devices (p-bits), which are modeled to realize an intrinsic sigmoidal activation function. A CMOS/spin based weighted array structure is designed to implement a restricted Boltzmann machine (RBM). Device-level simulations based on precise physics relations are used to validate the sigmoidal relation between the output probability of a p-bit and its input currents. Characteristics of the resistive networks and p-bits are modeled in SPICE to perform a circuit-level simulation investigating the performance, area, and power consumption tradeoffs of the weighted array. In the application-level simulation, a DBN is implemented in MATLAB for digit recognition using the extracted device and circuit behavioral models. The MNIST data set is used to assess the accuracy of the DBN using 5,000 training images for five distinct network topologies. The results indicate that a baseline error rate of 36.8% for a 784×10 DBN trained by 100 samples can be reduced to only 3.7% using a 784×800×800×10 DBN trained by 5,000 input samples. Finally, Power dissipation and accuracy tradeoffs for probabilistic computing mechanisms using resistive devices are identified.

KEYWORDS

Deep belief network (DBN), Boltzmann machine (BM), p-bit, proba-bilistic spin logic (PSL), resistive network.

1 INTRODUCTION

The interrelated fields of machine learning (ML), artificial intelligence, and artificial neural networks (ANN) have grown significantly in previous decades due to the availability of computing systems powerful enough to train and simulate large scale ANNs within reasonable time-scales, as well as the abundance of data available to train such networks in recent years. The resulting research has realized a bevy of ANN architectures that have performed incredible feats including a wide range of classification problems [1], and various recognition tasks [2].

Most ML techniques in-use today rely on supervised learning, where the systems are trained on patterns with a known desired output, or label. However, intelligent biological systems exhibit unsupervised learning whereby statistically correlated input modalities are associated within an internal model used for probabilistic inference and decision making [3]. One interesting class of unsupervised learning approaches that has been extensively researched is the Restricted Boltzmann machine (RBM) [4]. RBMs can be hierarchically organized to realize deep belief networks (DBNs) that have demonstrated unsupervised learning abilities, such as natural language understanding [5]. Most RBM and DBN research has focused on software implementations, which provides flexibility, but requires significant execution time and energy due to large matrix multiplications that are relatively inefficient when implemented on standard Von-Neumann architectures due to the memory-processor bandwidth bottleneck when compared to hardware-based in-memory computing approaches [6]. Thus, research into hardware-based RBM designs has recently sought to alleviate these constraints.

Previous hardware-based RBM implementations have aimed to overcome software limitations by utilizing FPGAs [7, 8] and stochastic CMOS [9]. In recent years, emerging technologies such as resistive RAM (RRAM) [10, 11] and phase change memory (PCM) [12] are proposed to be leveraged within the DBN architecture as weighted connections interconnecting building blocks in RBMs. While most of the previous hybrid Memristor/CMOS designs focus on improving the synapse behaviors, the work presented herein overcomes many of the preceding challenges by utilizing a novel spintronic p-bit device that leverages intrinsic thermal noise within low energy barrier nanomagnets to provide a natural building block for RBMs within a compact and low-energy package. The contribution of this paper is go to beyond using low-energy barrier magnetic tunnel junctions (MTJs), as has been previously introduced for a neuron in spiking neuromorphic systems [13, 14]. To the best of our knowledge this paper is the first effort to use MTJs with near-zero energy barriers as neurons within an RBM implementation. Additionally, various parameters of a hybrid CMOS/spin weight array structure are investigated for metrics of power dissipation, and error rate using the MNIST digit recognition benchmarks.

2 FUNDAMENTALS OF RBM

Boltzmann Machines (BM) are a class of recurrent stochastic ANNs with binary nodes whereby each possible state of the network, v, has an energy determined by the undirected connection weights between nodes and the node bias as described by (1), where s_i^v is the state of node i in v, b_i is the bias, or intrinsic excitability of node i, and w_{ij} is the connection weight between nodes i and j [15].

$$E(v) = -\sum_i s_i^v b_i - \sum_{i<j} s_i^v s_j^v w_{ij} \tag{1}$$

$$P(s_i = 1) = \sigma(b_i + \sum_j w_{ij} s_j) \tag{2}$$

Each node in a BM has a probability to be in state 1 according to (2), where σ is the logistic sigmoid function. BMs, when given enough time, will reach a Boltzmann distribution where the probability of the system being in state $\boldsymbol{v}$ is found by $P(v) = \frac{e^{-E(v)}}{\sum_u e^{-E(u)}}$, where $\boldsymbol{u}$ could be any possible state of the system. Thus, the system is most likely to be found in states that have the lowest associated energy. Restricted Boltzmann machines (RBMs) are BMs constrained to two fully-connected non-recurrent layers called the *visible layer*, where salient inputs clamp nodes to output levels of either zero or one, and the *hidden layer*, where associations between input vectors are learned. By enforcing the conditional independence of the visible and hidden layers, unbiased samples from the input can be obtained in one time-step, which enhances the learning process.

The most widely used method for training RBMs is contrastive divergence (CD), which is an approximate gradient descent procedure using Gibbs sampling [16]. CD operates in three phases: *(1) Positive Phase:* A training input vector, v, is applied to the visible layer by clamping the nodes to either 1 or 0 levels, and the hidden layer is sampled, h. *(2) Negative Phase:* by clamping the hidden layer to h, the

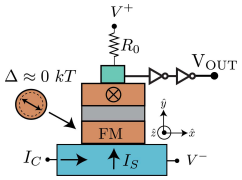

Figure 1: Structure of a p-bit.

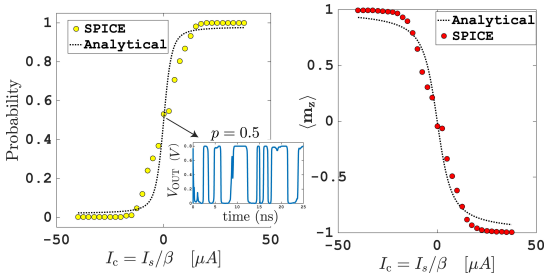

Figure 2: Time-averaged results over 100 ns for p-bit.

reconstructed input layer is sampled, v'. Then, clamp the visible layer to v' and sample the hidden layer to obtain h'. *(3)* Update the weights according to $\Delta W = \eta(vh^T - v'h'^T)$, where η is the learning rate and W is the weight matrix.

DBNs are realized when additional hidden layers are stacked on top of an RBM, and can be trained in a very similar way to RBMs. Essentially, training a DBN involves performing CD on the visible layer and the first hidden layer for as many steps as desired, then fixing those weights and moving up a hierarchy as follows. The first hidden layer is now viewed as a visible layer, while the second hidden layer acts as a hidden layer with respect to the CD procedure identified above. Next, another set of CD steps are performed, and then the process is repeated for each additional layer of the DBN.

3 SPIN-BASED BUILDING BLOCK FOR RBM

In this section, we provide a detailed description of the p-bit that provides the building block for our proposed spin-based BM architecture. Individual building blocks are interconnected by networks of memristive devices whose resistances can be programmed to provide the desired weights. For instance, in this paper, we will assume that the memristive devices are implemented using the three terminal spin-orbit torque (SOT)-driven domain wall motion (DWM) device proposed in [17].

The activation function is achieved by a spintronic building block that has been used in the design of probabilistic spin logic devices (p-bits) for a wide variety of Boolean and non-Boolean problems [18–21]. The basic functionality of the p-bit shown in Fig. 1 [18] is to produce a stochastic output whose steady-state probability is modulated by an input current to generate a sigmoidal activation function. For instance, a high positive input current produces a stochastic output with a high probability of "0", and vice versa. In the absence of any input current, the device generates either 0 or VDD outputs with roughly equal probability of 0.5, as shown in Fig. 2. This device consists of a 3-terminal, spin-Hall driven MTJ [22] that uses a circular, unstable nanomagnet ($\Delta \ll 40kT$), whereby its output is amplified by CMOS inverters as shown in Fig. 1. This MTJ with an unstable free layer can be fabricated using standard technology such that the surface anisotropy to achieve perpendicular magnetic anisotropy (PMA) that is not strong enough to overcome the demagnetizing field. Thus, the magnetization stochastically rotates in the plane, due to the presence of thermal fluctuations.

The charge current that is injected to the spin-Hall layer creates a spin-current flowing into the circular FM (in the +y direction), which does not have an axis with any preferential geometry. The spin-polarization of this spin-current is in the ($\pm z$) direction, and pins the magnetization in the (+z) or (-z) direction depending on the direction of the charge current, through the spin-torque mechanism [20]. The inherent physics of the spin-current driven low-barrier nanomagnet provides a natural sigmoidal function when a long time average of magnetization is taken. Through the tunneling magnetoresistance effect, a charge current flowing through the MTJ with a stable fixed layer detects the modulated magnetization as a voltage change. To achieve this, a small read voltage V_R is applied between the $V+$ and $V-$ terminals through a reference resistance R_0, adjusted to the average conductance of the MTJ ($R_0^-1 = GP + GAP/2$) where GP and GAP represented conductance in parallel (P) and anti-parallel (AP) states, respectively. This voltage becomes an input to the CMOS inverters that are biased at the middle point of their DC operating point, creating a stochastic output whose probability can be tuned by the input charge current.

Each component of the device is represented by an independent spin-circuit based on experimentally-benchmarked models that have been established in [23] and simulated as a spin-circuit in a SPICE-like platform. Here, we obtain an analytical approximation to the time-averaged

Table 1: Parameters for p-bit Based Activation Function.

Parameter	Description	Value
Circular FM		
ϕ	Diameter	$100nm$
t	Thickness	$2nm$
α	Damping coefficient	0.01
MTJ		
$G0$	Conductance	$150e^{-6}S$
P	Spin Polarization	0.52
Giant Spin Hall Layer(GSHE)		
λ	Spin-diffusion length	$2.1nm$
θ	Spin Hall Angle	0.5
$Volume$	$l \times w \times t$	$100nm \times 100nm \times 3.15nm$

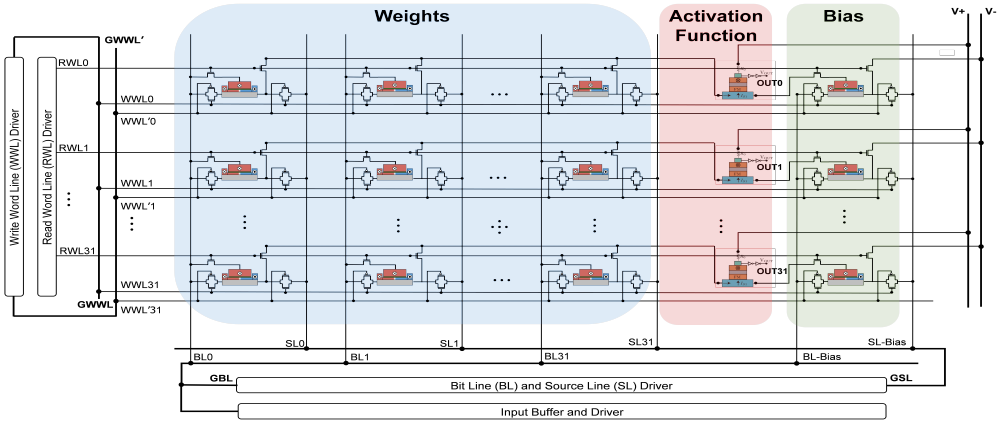

Figure 3: Proposed 32×32 hybrid CMOS/spin-based weighted array structure for RBM implementation.

behavior of the output characteristics. We start by relating the charge current flowing in the spin Hall layer to the spin-current absorbed by the magnet, assuming short-circuit conditions for simplicity, i.e. 100% spin absorption by the FM:

$$I_s/I_c = \beta = \frac{L}{t}(\theta)(1 - sech(\frac{t}{\lambda})) \tag{3}$$

where I_s is the spin-current, I_c is the charge current, θ is the spin-Hall angle, L, t, λ are the length, thickness and spin diffusion lengths for the spin-Hall layer. The length and width of the GSHE layer are assumed to be the same as the circular nanomagnet. With a suitable choice of the L and t, the spin-current generated can be greater in magnitude than the charge current generating "gain." For the parameters used in this paper, which are listed in Table 1, the gain factor β is ~ 10. Next, we approximate the behavior of the magnetization as a function of an input spin-current, polarized in the $(\pm z)$ direction. For a magnet with only a PMA in the $\pm z$ direction, a distribution function at steady state can be written analytically as below, as long as the spin-current is also fully in the $\pm z$ direction:

$$\rho(m_z) = \frac{1}{Z} exp(\Delta m_z^2 + 2 i_s m_z) \tag{4}$$

where Z is a normalization constant, m_z is the magnetization along $+z$, is the thermal barrier of the nanomagnet, and i_s is a normalization quantity for the spin-current such that $i_s = I_s/(4q/\hbar\alpha kT)$, α being the damping coefficient of the magnet, q the electron charge and $\hbar$ the reduced Planck constant. It is possible to use (4) to obtain an average magnetization $< m_z >= \int_{-1}^{+1} dm_z m_z \rho(m_z) / \int_{-1}^{+1} dm_z \rho(m_z)$. Assuming $\Delta \ll kT$, $< m_z >$ can be evaluated to give the Langevin function, $< m_z >= L(i_s)$ where $L(x) = \frac{1}{x} - coth\frac{1}{x}$, which is an exact description for the average magnetization in the presence of a z-directed spin-current for a low barrier PMA magnet.

In the present case, however, the nanomagnet has a circular shape with a strong in-plane anisotropy and no simple analytical formula can be derived, thus We use the Langevin function with a fitting parameter that adjusts the normalization current by a factor η, so that the modified normalization constant becomes $(4q/\hbar\alpha kT)(\eta)$. This factor increases with elevating the shape anisotropy ($H_d \sim 4\pi M_s$) and becomes exactly one when there is no shape anisotropy. Once the magnetization and charge currents are related, we can approximate the output probability of the CMOS inverters by a phenomenological equation along with fitting parameter χ as follows, $p = \frac{V_{OUT}}{VDD} \approx \frac{1}{2}[1 - tanh(\chi < m_z >)]$, which allows us to relate the input charge current to the output probability, with physical parameters. Fig. 2 shows the comparison of the

Table 2: Signaling to Control The Array Operations.

Operation	WWL	RWL	BL	SL	V+	V-
Increase Weight	VPULSE	GND	VDD	GND	Hi-Z	Hi-Z
Decrease Weight	VPULSE	GND	GND	GND	Hi-Z	Hi-Z
Read	GND	VDD	VIN	Hi-Z	VDD	VDD/2

Table 3: Relation between the input currents of activation functions and array size for $R_P = 1M\Omega$.

Features	Array Size			
	8×8	16×16	32×32	64×64
Max. Positive Current (μA)	2.71	5.14	10.79	21.46
Max. Negative Current (μA)	3.57	7.14	14.23	28.28
Max. output "0" Probability	0.77	0.88	0.95	0.97
Min. output "0" Probability	0.175	0.08	0.038	0.026

full SPICE-model with respect to aforementioned equations showing good agreement with two fitting parameters η and χ, which fit the magnetization and CMOS components, respectively.

4 PROPOSED WEIGHTED ARRAY DESIGN

Figure 3 shows the structure of the weighted array proposed herein to implement the RBM architecture including the SOT-DWM based weighted connections and biases, as well as the p-bit based activation functions. Transmission gates (TGs) are utilized in write circuits within the bit cells of the weighted connection to adjust weights by moving the DW position. As investigated in [24], TGs can provide energy-efficient and symmetric switching operation for SOT-based devices, which are desirable during the training phase. Table 2 lists the required signaling for controlling the training and read operations in the weighted array structure. Herein, a chain of inverters are considered to drive signal lines, in which each successive inverter is twice as large as the previous one.

During the read operation, write word line (WWL) is connected to ground (GND) and the source line (SL) is in high impedance (Hi-Z) state, which disconnects the write path. The read word line (RWL) for each row is connected to VDD, which turns ON the read transistors in the weighted connection bit cell. The bit line (BL) will be connected to the input signal (VIN), which results in producing a current that affects the output probability of the p-bit device. The direction of the generated current relies on the VIN signal. In particular, since V- is supplied by a voltage source equal to VDD/2, if VIN is connected to VDD the injected current to the p-bit based activation function will have positive value, and if VIN is zero the input current will be negative. The amplitude of the generated current depends on the resistance of the weighted connection which is defined by the position of the DW in the SOT-DWM device.

During the training operation, the RWL is connected to GND, which turns OFF the read transistors and disconnects the read path. The WWL is connected to an input pulse (VPULSE) signal which activates the write path for a short period of time. The duration of the VPULSE should be designed in a manner such that it can provide the desired learning rate, η, to the training circuit. For instance, a high VPULSE duration results in a significant change in the DW position in each training iteration, which effectively reduces the number of different resistive states that can be realized by the SOT-DWM device. Resistance of the weighted connections can be adjusted by the BL and SL signals, as listed in Table 2. A higher resistance leads to a smaller current injected to the p-bit device. Therefore, the input signal connected to the weighted connection will have lower impact on the output probability of the p-bit device, which means the input signal exhibits a lower weight. The bias nodes can also be adjusted similar to the weighted connection.

5 SIMULATION RESULTS AND DISCUSSION

To analyze the RBM implementation using the proposed p-bit device and the weighted array structure, we have utilized a hierarchical simulation framework including circuit-level and application-level simulations. In circuit level simulation, the behavioral models of the p-bit and SOT-DWM devices were leveraged in SPICE circuit simulations using 20nm CMOS technology with 0.9V nominal voltage to validate the functionality of the designed weighted array circuit. In application-level simulation, the results obtained from device-level and circuit-level simulations are used to implement a DBN architecture and analyze its behavior in MATLAB.

5.1 Circuit-level simulation

The device-level simulations shown in Fig. 2 verified a sigmoidal relation between the input current of the p-bit based activation function and its output probability. The shape of the activation on function is one of the major factors affecting the performance of the RBM. Therefore, we have provided comprehensive analyses on the impacts of weighted connection resistance and weighted array dimensions on the input currents of the p-bit based activation functions, and the power consumption of the weighted array.

Table 3 lists the range of the activation function input currents for various weighted array dimensions, while the resistance of the SOT-DWM device in parallel state (RP) is constant and equals $1M\Omega$. The experimental results provided in [19, 28] exhibit that an MTJ

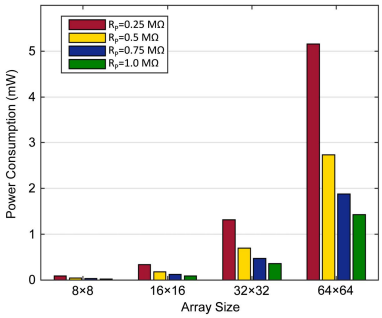

Figure 4: Weighted array power consumption versus the resistance of the weighted connections and array size.

Table 4: Relation between the input currents of activation functions and R_P in a 32×32 array.

Features	$R_P(M\Omega)$			
	0.25	0.5	0.75	1
Max. Positive Current (μA)	36.56	20.02	13.97	10.79
Max. Negative Current (μA)	54.95	28.12	18.9	14.23
Max. output "0" Probability	0.98	0.965	0.96	0.95
Min. output "0" Probability	0.01	0.026	0.032	0.038

Table 5: Comparison between various RBM implementations with an emphasis on activation function structure.

Design	[7]	[8]	[9]	[10]	[11]	[12]	Proposed Herein
Weighted Connection	Embedded multipliers	Embedded multipliers	- LFSR - AND/OR gates	RRAM memristor	RRAM	PCM	SOT-DWM
Activation Function	CMOS-based LUTs	-2-kB BRAM - Picewise Linear Interpolator - Random number Generator	- LFSR - Bit-wise AND - tree adder - FSM-based tanh unit	Off-chip	- 64×16 LUTs - Pseudo Random Number Generator - Comparator	Off-chip	- near-zero energy barrier probabilistic spin logic device
Energy per neuron	N/A	$\sim 10 - 100nJ$	$\sim 10 - 100pJ$	N/A	$\sim 1 - 10nJ$	N/A	$\sim 1 - 10fJ$
Normalized area per neuron	N/A	$\sim 3000\times$	$\sim 90\times$	N/A	$\sim 1250\times$	N/A	$\sim 1\times$

resistance in the $M\Omega$ range can be obtained by increasing the oxide thickness in an MTJ structure. The highest positive and negative currents can be achieved while the weighted connections are in parallel state, i.e. lowest resistance, and all of the input voltages (VIN) are equal to VDD and GND, respectively. The difference between the amplitude of positive and negative currents in a given array size with constant RP is caused by the different pull-down and pull-up strengths in NMOS read transistors. The maximum and minimum output-level "0" probabilities are listed in Table 3, which can be obtained according to the measured input currents and the sigmoidal activation function shown in Fig. 2.

Moreover, Table 4 illustrates the relation between the R_P values and input currents of the activation functions, and their corresponding output probabilities, for a given 32×32 weighted array. The lower RP resistance and higher array size provides a wider range of output probabilities which can increase the RBM performance. However, this is achieved at the cost of higher area and power consumption. The trade-offs between the array size, weighted connection resistance, and average power consumption in a single read operation is shown in Fig. 4. The lowest power consumption of 22.6 μW is realized by an 8×8 array with $R_P = 1M\Omega$. However, this array provides the narrowest range of the output probabilities, which significantly reduces the performance of the DBN.

5.2 Application-level simulation

In the application-level simulation, we have leveraged the obtained device and circuit behavioral models to simulate a DBN architecture for digit recognition. In particular, learning rate and the shape of the sigmoid activation function is extracted by the SOT-DWM and p-bit device-level simulations, respectively, while the circuit-level simulations defines the range of the output probabilities. To evaluate the performance of the system, we have modified a MATLAB implementation of DBN by Tanaka and Okutomi [25] and used the MNIST data set [26] including 60,000 and 10,000 sample images with 28×28 pixels for training and testing operations, respectively. We have used Error rate (ERR) metric to evaluate the performance of the DBN, as expressed by $ERR = N_F/N$, where, N is the number of input data, N_F is the number of false inference [25].

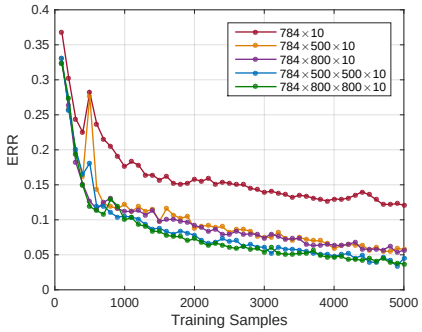

Figure 5: ERR for various DBN topologies.

The simplest model of the DBN that can be implemented for MNIST digit recognition consists 784 nodes in visible layer to handle 28×28 pixels of the input images, and 10 nodes in hidden layer representing the output classes. Fig. 5 shows the relation between the performance of various DBN topologies, and the number of input training samples ranging from 100 to 5,000, which is obtained using 1,000 test samples. The ERR and RMSE metrics can be improved by enlarging the DBN structure through increasing the number of hidden layers, as well as the number of nodes in each layer. This improvement is realized at the cost of larger area and power consumptions. Increasing the input training samples can improve the DBN performance as well, however it will quickly converge due to the limited weight values that can be provided by SOT-DWM based weighted connections. As shown in Fig. 5, some random behaviors are observed for networks with smaller sizes that are trained by lower number of training samples, which will be significantly reduced by increasing the number of training samples.

The simulation results exhibit the highest error rate of 36.8% for a 784×10 DBN that is trained by 100 training samples. Meanwhile, the lowest error rate of 3.7% was achieved using a $784 \times 800 \times 800 \times 10$ DBN trained by 5,000 input training samples. This illustrates that the recognition error rate can be decreased by increasing the number of hidden layers, and training samples, which is also realized at the cost of higher area and power overheads.

5.3 Disucussion

Table 5 lists previous hardware-based RBM implementations, which have aimed to overcome software limitations by utilizing FPGAs [7, 8], stochastic CMOS [9], and hybrid memristor-CMOS designs [10–12]. FPGA implementations demonstrated RBM speedups of 25-145 over software implementations [7, 8], but had significant constraints such as only realizing a single 128×128 RBM per FPGA chip, routing congestion, and clock frequencies limited to 100MHz [8]. The stochastic CMOS-based RBM implementation proposed in [9] leveraged the low-complexity of stochastic CMOS arithmetic to save area and power. However, the need for extremely long bit-stream lengths negate energy savings and lead to very long latencies. Additionally, a significant amount of Linear Feedback Shift Registers (LFSRs) were required to produce the uncorrelated input and weight bit-streams. In both the FPGA and stochastic CMOS designs, improvements were achieved by implementing parallel Boolean circuits such as multipliers and pseudo-random number generators for probabilistic behavior, which has significant area and energy overheads compared to leveraging the physical behaviors of emerging devices to perform the computation intrinsically. Bojnordi et al. [11] leveraged resistive RAM (RRAM) devices to implement efficient matrix multiplication for weighted products within Boltzmann machine applications, and demonstrated significant speedup of up to 100-fold over single-threaded cores and energy savings of over 10-fold. Similarly, Sheri et al. [10] and Eryilmaz et al. [12] utilized RRAM and PCM devices to implement matrix multiplication, while the corresponding activation function circuitry is still based on the CMOS technology, which suffers from the aforementioned area and power consumption overheads.

While most of the previous hybrid Memristor/CMOS designs focus on improving the performance of weighted connections, the work presented herein overcomes many of the preceding challenges of generating sigmoidal probabilistic activation functions by utilizing a novel p-bit device that leverages intrinsic thermal noise within low energy barrier nanomagnets to provide a natural building block for RBMs within a compact and low-energy package. As listed in Table V, the proposed design can achieve approximately three orders of magnitude improvement in term of energy consumption compared to the most energy-efficient designs, while realizing at least 90X device count reduction for considerable area savings. Note that these calculations do not take into account the weighted connections, since the main focus of this paper is on the activation function. While SOT-DWM devices are utilized herein for the weighted connections, any other memristive devices could be utilized without loss of generality.

6 CONCLUSION

In this paper, we developed a hybrid CMOS/spin-based DBN implementation leveraging p-bit based activation functions, and SOT-DWM based weighted connections. First, a p-bit device is used to produce a probabilistic output, which can be modulated by an input current. The device-level simulations exhibited a sigmoid relation between the input currents and output probability of the p-bit device. Next, a SPICE-based behavioral model of the p-bit device was developed to design a weighted array structure for implementing RBM. The proposed

array was examined using circuit-level simulations. The results showed that the performance of the array can be improved by enlarging the array size, as well as reducing the resistance of the weighted connections. However, these improvements can be achieved at the cost of increased area and power consumption. For instance, the lowest power dissipation among the examined designs belongs to an 8×8 array with the maximum resistance of $1M\Omega$ for weighted connections. However, this array structure can only provide the output probabilities ranging from 0.175 to 0.77, which is the narrowest range among the examined designs resulting in a DBN implementation with lowest accuracy.

Finally, we have simulated a DBN for digit recognition application in MATLAB based on the extracted device and circuit-level behavioral models. Trade-offs include the relations between the recognition accuracy of the designed DBN architecture and the number of training samples, which are comparable to conventional hardware implementations. As depicted in the results, the recognition error rate decreased substantially for the first thousand training samples, regardless of the size of the array, while benefits continue through several thousand inputs. However, at least two hidden layers are desirable to achieve suitable error rates. Finally, we have provided a comparison between previous hardware-based RBM implementations and our proposed design with an emphasis on the probabilistic activation function realization within the neuron structure. The results exhibited that the p-bit based activation function can achieve roughly three orders of magnitude energy improvement, while realizing at least 90X reduction in terms of device count, compared to the previous most energy-efficient designs.

REFERENCES

[1] I. Basheer and M. Hajmeer, "Artificial neural networks: fundamentals, computing, design, and application," *Journal of microbiological methods*, vol. 43, no. 1, pp. 3–31, 2000.
[2] C. M. Bishop, *Neural networks for pattern recognition*. Oxford university press, 1995.
[3] L. Buesing, J. Bill, B. Nessler, and W. Maass, "Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons," *PLoS computational biology*, vol. 7, no. 11, p. e1002211, 2011.
[4] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
[5] R. Sarikaya, G. E. Hinton, and A. Deoras, "Application of deep belief networks for natural language understanding," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 4, pp. 778–784, Apr. 2014.
[6] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.
[7] S. K. Kim, P. L. McMahon, and K. Olukotun, "A large-scale architecture for restricted boltzmann machines," in *2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, May 2010, pp. 201–208.
[8] D. L. Ly and P. Chow, "High-performance reconfigurable hardware architecture for restricted boltzmann machines," *IEEE Transactions on Neural Networks*, vol. 21, no. 11, pp. 1780–1792, Nov 2010.
[9] A. Ardakani, F. Leduc-Primeau, N. Onizawa, T. Hanyu, and W. J. Gross, "Vlsi implementation of deep neural network using integral stochastic computing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 10, pp. 2688–2699, Oct 2017.
[10] A. M. Sheri, A. Rafique, W. Pedrycz, and M. Jeon, "Contrastive divergence for memristor-based restricted boltzmann machine," *Engineering Applications of Artificial Intelligence*, vol. 37, no. Supplement C, pp. 336 – 342, 2015.
[11] M. N. Bojnordi and E. Ipek, "Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2016.
[12] S. B. Eryilmaz, E. Neftci, S. Joshi, S. Kim, M. BrightSky, H. L. Lung, C. Lam, G. Cauwenberghs, and H. S. P. Wong, "Training a probabilistic graphical model with resistive switching electronic synapses," *IEEE Transactions on Electron Devices*, vol. 63, no. 12, pp. 5004–5011, Dec 2016.
[13] A. Sengupta, P. Panda, P. Wijesinghe, Y. Kim, and K. Roy, "Magnetic tunnel junction mimics stochastic cortical spiking neurons," *Scientific reports*, vol. 6, p. 30039, 2016.
[14] A. Sengupta, M. Parsa, B. Han, and K. Roy, "Probabilistic deep spiking neural systems enabled by magnetic tunnel junction," *IEEE Transactions on Electron Devices*, vol. 63, no. 7, pp. 2963–2970, July 2016.
[15] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cognitive science*, vol. 9, no. 1, pp. 147–169, 1985.
[16] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning." in *Aistats*, vol. 10, 2005, pp. 33–40.
[17] A. Sengupta, A. Banerjee, and K. Roy, "Hybrid spintronic-cmos spiking neural network with on-chip learning: Devices, circuits, and systems," *Phys. Rev. Applied*, vol. 6, p. 064003, Dec 2016.
[18] K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, "Stochastic p-bits for invertible logic," *Phys. Rev. X*, vol. 7, p. 031014, Jul 2017.
[19] R. Faria, K. Y. Camsari, and S. Datta, "Low-barrier nanomagnets as p-bits for spin logic," *IEEE Magnetics Letters*, vol. 8, pp. 1–5, 2017.
[20] B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, "Intrinsic optimization using stochastic nanomagnets," *Scientific Reports*, vol. 7, 2017.
[21] B. Behin-Aein, V. Diep, and S. Datta, "A building block for hardware belief networks," *Scientific reports*, vol. 6, 2016.
[22] L. Liu, C.-F. Pai, Y. Li, H. W. Tseng, D. C. Ralph, and R. A. Buhrman, "Spin-torque switching with the giant spin hall effect of tantalum," vol. 336, no. 6081, pp. 555–558, 2012.
[23] K. Y. Camsari, S. Ganguly, and S. Datta, "Modular approach to spintronics," *Scientific reports*, vol. 5, 2015.
[24] R. Zand, A. Roohi, and R. F. DeMara, "Energy-efficient and process-variation-resilient write circuit schemes for spin hall effect mram device," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 9, pp. 2394–2401, 2017.
[25] M. Tanaka and M. Okutomi, "A novel inference of a restricted boltzmann machine," in *2014 22nd International Conference on Pattern Recognition*, 2014, pp. 1526–1531.
[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov 1998.