Delta-Sigma Modulation Neurons for High-Precision Training of Memristive Synapses in Deep Neural Networks

Loai Danial, *Student Member, IEEE*, Sidharth Thomas, and Shahar Kvatinsky, *Senior Member, IEEE*
The Andrew and Erna Viterbi Faculty of Electrical Engineering
Technion - Israel Institute of Technology, Haifa 3200003, ISRAEL
Email: sloaidan@campus.technion.ac.il, shahar@ee.technion.ac.il

Abstract— **The spike generation mechanism and information coding process of biological neurons can be emulated by the amplitude-to-frequency modulation property of delta-sigma modulators (ΔΣ). Oversampling, averaging, and noise-shaping features of the ΔΣ allow high neural coding accuracy and mitigate the intrinsic noise level in neural networks. In this paper, a ΔΣ is proposed as a neuron activation function for inference and training of artificial analog neural networks. The inherent dithering of the ΔΣ prevents the weights from being stuck in a spurious local minimum, and its non-linear transfer function makes it attractive for multi-layer architectures. Memristive synapses are used as weights, which are trained by supervised/unsupervised machine learning (ML) algorithms, using stochastic gradient descent (SGD) or biologically plausible spike-time-dependent plasticity (STDP). Our ΔΣ networks outperform the prevalent power-hungry pulse width modulator counterparts, with 97.37% training accuracy and 3.2X speedup in MNIST using SGD. These findings constitute a milestone in closing the cultural gap between brain-inspired models and ML using analog neuromorphic hardware.**

Keywords— **Artificial neural networks, backpropagation, delta-sigma modulation, memristive systems, machine learning, neuromorphic computing, stochastic gradient descent, STDP.**

I. Introduction

The impending end of Moore's law and Dennard scaling requires rethinking the way computing is performed. Thus, several neuro-inspired architectures have been proposed, shifting the spotlight from the traditional von Neumann paradigm to neuromorphic computing [1-5]. Designed to accelerate real-time data-driven applications, this new paradigm mimics intrinsic properties of the human brain to handle the ever-growing data volumes by means of adaptive capabilities, massively parallel interconnections, noise tolerance, and energy efficiency. To implement this new paradigm, researchers have turned to machine learning (ML) for inspiration, as it has already achieved adaptive and error-tolerant training in a software environment.

ML is now prevalent in all domains of engineering. The trainability feature of ML algorithms allows them to be used independently in continuously varying conditions. Artificial Neural Networks (ANNs) are an example of such trainable architectures [6]. The building blocks of these architectures are synapses that could be trained to store weights for the network functionality, and neurons that collectively interact to encode and transmit information. Deep neural networks are modularly constructed using a large number of massively interconnected layers trained to perform inference. A layer is an atomic neural entity, comprising trainable synapses (matrix) and neurons (input vector), abstracted by the vector-matrix multiplication (VMM) dot product. It is commonly trained either using ML optimization techniques, *e.g.*, stochastic gradient descent (SGD) [6], or using neuro-inspired heuristics, *e.g.*, spike-time-dependent plasticity (STDP) [7].

Unfortunately, handling these computationally intensive arithmetic operations, even in custom designed hardware (ASIC), is constrained due to the excessive data movement between the memory elements and the processing units [2]. Implementing hardware for ANNs requires novel circuits and devices, capable of handling fast VMMs with added non-volatile storage capabilities. The last decade has witnessed a technological breakthrough in non-volatile nanoscale memory technologies [8]. Memristors are now being widely adopted in the design of synapses for ANNs because of their small size, energy efficiency, and non-volatility. These characteristics allow for synapse-like behavior, where the conductance of the memristor is considered as the synaptic weight [7]. Using a resistive platform for computation can accelerate VMM as a result of Ohm's and Kirchhoff's laws, and reconfigure it using memristor-compatible ML algorithms. Furthermore, physically realizable memristive synapses have sparked neuroscience in spike-based learning techniques [7].

However, several challenges have hindered the practical use of memristor technology. The main obstacles in this regard are variability, the stochastic nature of memristor switching, and integration with existing technologies [9]. Many efforts have been made to overcome these obstacles, but here we apply a novel approach, inspired by the fault-tolerant biological neuron [10][11]. The most popular previous attempt to achieve the functionality of the biological neuron using a mature CMOS technology is the leaky integrate and fire (LIF) model [12]. Pulse width modulation (PWM) neurons have been also suggested [13]. While the LIF neuron does not shape white noise sufficiently, due to its backward reset, the PWM neuron does not filter noise, due to its feedforward structure. Furthermore, the neural spike code remains a central problem in neuroscience: there is no consensus as to whether information is encoded in firing rates, or in individual spike timing [11].

In this paper, we utilize the resemblance between a biological neuron and a delta-sigma (ΔΣ) modulator [14] to improve training and inference accuracy, by precise programming control of memristive synapses. This will overcome the intrinsic variability of the memristors, encode information using frequency of a pulse-train, and achieve high training and inference accuracy. ΔΣ modulators are traditionally used as building blocks in high resolution data converters with remarkable noise tolerance. We demonstrate the proposed ΔΣ neurons for performing supervised learning using SGD and unsupervised learning using STDP.

The remainder of this paper is organized as follows. In Section II, background on the existing neuron models and the building blocks of the ΔΣ ANNs are provided. Section III presents the proposed single-layer and multi-layer ΔΣ neural network architecture. In Section IV, the supervised and unsupervised learning capabilities of the proposed ANN are evaluated. The paper is summarized in Section V.

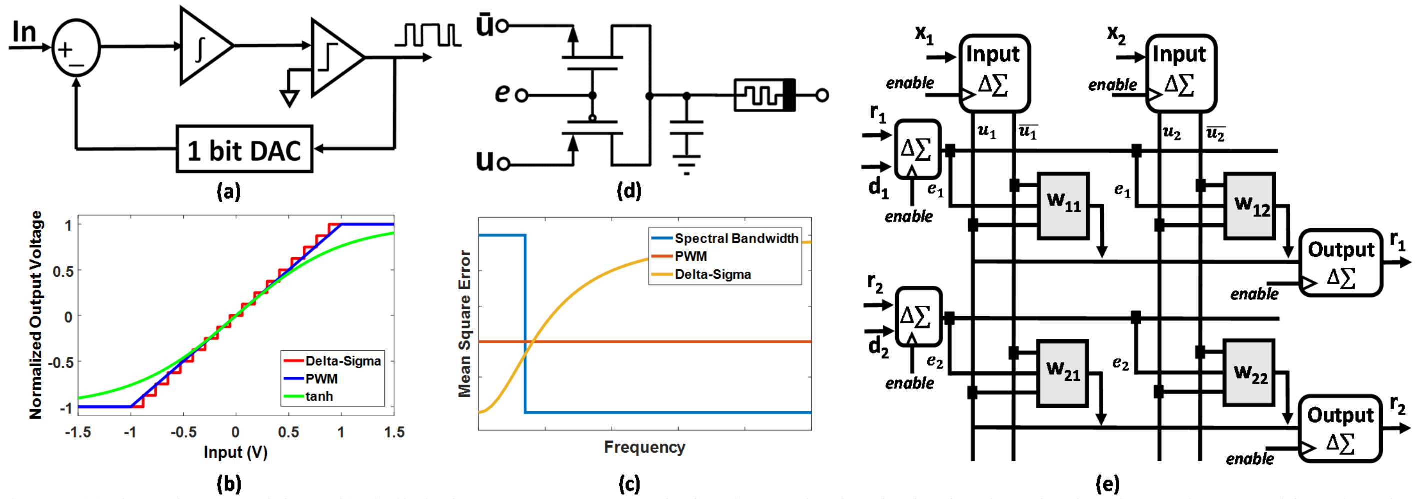

Figure 1: (a) First order ΔΣ modulator. (b) Similarity between ΔΣ, PWM transfer functions, and tanh activation function. The plotted output is extracted from the pulses' product in response to an input and normalized. The discrete steps in the ΔΣ, makes the weights intolerant to small input variations so that they move in the direction of the maximum gradient. (c) Spectral noise shaping of ΔΣ. The spectral bandwidth is derived from the spectral representation of a cost function (d) 2T-1R synapse implements non-volatile weight, where the shock capacitor is added to eliminate noise [20]. (e) Proposed single layer architecture of a 2x2 ANN with weights $w_{i,j}$ and x_j inputs ($\boldsymbol{i,j} \le \mathbf{2}$), designed using 2T–1R synapses and ΔΣ neurons. The enable signal determines the operation and synchronizes the read, update, and reset signals.

II. Background

In this section, we present basic background on the building blocks of the proposed ΔΣ ANNs: memristive synapses, neuron models, and training algorithms. We emphasize their correlated characteristics to facilitate understanding of the proposed architecture.

A. Memristive Synapses

Memristors are two-terminal passive devices with varying resistance, which changes according to the current flowing through the device, or alternatively, the voltage across the device. Memristors primarily serve as non-volatile memory and can be used for both digital and analog applications [15]. The activation-dependent dynamics of memristors make them a promising feature for registering and updating synaptic weights. The high-to-low resistance (HRS/LRS) ratio is an important factor in determining the maximum number of available resistive levels, with device non-linearity, noise, endurance, and Poisson distributed stochastic switching placing a limit on exercising these levels. Our VTEAM memristor model [16] is used with parameters fitted for a linearized Pt/HfOx/Hf/TiN RRAM device [17] with a metal buffer layer. The device provides low forming and programming voltage with a high-to-low resistance state (HRS/LRS) ratio of approximately 50 [5].

Memristive crossbar arrays inherently implement VMM by Ohm's and Kirchhoff's laws for ANN hardware realization. The output vector $\boldsymbol{r}$ corresponding to a layer is determined as

$$\boldsymbol{r} = \boldsymbol{Wx}, \qquad (1)$$

where $\boldsymbol{W}$ is the synaptic weight matrix, realized by the conductance values of memristors inside the crossbar, and $\boldsymbol{x}$ is the layer's input neuron vector, computed by the activations of the input neurons, *e.g.*, LIF, PWM or ΔΣ (Fig. 1(e)).

B. CMOS Neuron Models: LIF and PWM

The LIF neuron model uses a leaky capacitor to model the neuron membrane [12]. A spike is generated if the capacitor voltage crosses a pre-defined threshold, followed by a gradual reset. LIF remains the preferred model for spiking neural networks (SNNs) that rely on spike-based learning rules for unsupervised learning [18].

PWM [15] is a technique for encoding an analog input of a varying magnitude to output pulses with constant amplitude, and varying pulse width proportional to the analog input magnitude. The PWM activation function has been used to model neurons which have successfully trained ANNs using supervised learning [13] [19].

C. Training Algorithms: SGD and STDP

Stochastic gradient descent (SGD) [6] is a popular supervised learning algorithm for training ANNs by updating the synaptic weights W. The input vector is primarily randomized. The weight update (after each instance) aims to minimize a cost function, for example, the mean square error:

$$E_j = \tfrac{1}{2}\textstyle\sum_{k=1}^{K}(d_j^k - r_j^k)^2, \qquad (2)$$

where E_j is the mean square error (MSE) of the j^{th} neuron, k is the iteration count, and d is the expected value of the output [20]. The weight update rule is used with an η learning rate:

$$\Delta W_{ji} = \eta \frac{\partial E_j}{\partial W_{ji}}. \qquad (3)$$

STDP is a biological process that adjusts synaptic strengths in the brain [18]. The STDP algorithm is applied locally to a synapse with an emphasis on spike timings. If the presynaptic neuron fires before (after) the postsynaptic neuron, the synapse is strengthened (weakened). Studies have shown that memristors can accurately emulate a biological synapse, with several efforts to demonstrate STDP locally on memristive synapses [7], [21].

SGD and STDP learning rules are functionally different, with the latter requiring purely spike-encoded information. SGD is widely used in supervised learning for classification tasks, with a teacher signal label to iteratively optimize the cost function in (2), using the backpropagation algorithm. STDP, on the other hand, can be efficiently used in unsupervised learning for clustering tasks [22], to perform online training and simultaneous synaptic updates.

D. Delta-Sigma (ΔΣ) Modulator

A ΔΣ modulator, shown in Fig. 1(a), is a circuit topology used to modulate the amplitude of input signals to a binary sequence, with the percentage of 1's proportional to the amplitude [23]. It is a building block in high-resolution ADCs, where the conversion is done by cascading the modulator and an averaging filter. The ΔΣ modulator uses noise shaping and oversampling techniques to achieve high signal-to-noise ratio (SNR). If X_i , q , s and X_o are,

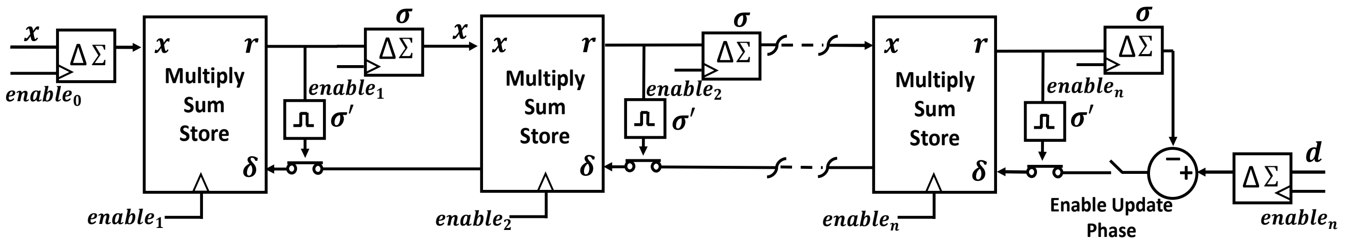

Figure 2: Proposed multi-layer neural network architecture using ΔΣ neurons. Each of the 'multiply, sum, store' units resemble the units in Fig. 1(e). Additional blocks which implement multiplication with $\boldsymbol{\sigma}'$are used. Inputs are fed forward to calculate the result. Backpropagation is used to train the weights.

TABLE I: CIRCUIT PARAMETERS

Parameter	Value	Parameter	Value
Power Supply			
V_{DD}	+1.8 V	V_{EE}	−1.8 V
NMOS		**PMOS**	
W/L	10	W/L	20
V_{TN}	0.56 V	V_{TP}	-0.57 V
Memristor			
$V_{on/off}$	-0.3V, 0.4V	$R_{on/off}$	2 $k\Omega$, 100 $k\Omega$
$K_{on/off}$	-4.8 mm/s, 2.8mm/s	$\alpha_{on/off}$	1,1
Operating Point		**Delta – Sigma Modulator**	
R_{out}	104 $k\Omega$	t_{pulse}	100 ns
R_{ref}	51 $k\Omega$	# pulses	32

respectively, the input, quantization noise, frequency and the output, at the Laplace domain, then

$$X_o(s) = \frac{X_i(s)\cdot s}{1+s} + \frac{q(s)}{1+s}. \tag{4}$$

III. TRAINABLE ANN ARCHITECTURE

In this section, we present our neural network architecture, including the proposed ΔΣ neuron and other building blocks and peripheral circuitry.

A. Delta-Sigma (ΔΣ) Neuron

We propose using a ΔΣ modulator as the artificial neuron due to its similarity to a biological neuron. The action potential spike [12] of a biological neuron has the same shape irrespective of the input. This suggests that the information is encoded in the frequency of spikes, which is similar to a ΔΣ modulation. Moreover, the action potential has periodic binary stages: a fired state and a resting state, just like the spectrally-distributed V_{high} and V_{low} pulse-train output of the ΔΣ modulator. The transfer function of the ΔΣ is similar to widely-used ANN activation functions, *e.g.*, tanh and sigmoid [14], as shown in Fig. 1(b).

The spectral bandwidth, derived from the cost function in (2), is limited to low frequencies in the spectral domain. The presence of noise within this bandwidth can adversely affect training time and accuracy. The noise-shaping property of ΔΣ can push the noise out of the required low-frequency spectral bandwidth, according to (4), as shown in Fig. 1(c). This provides higher SNR compared to PWM, where the noise floor is relatively unaltered [24]. Moreover, the inherent dithering property of the ΔΣ may prevent the weights from being stuck at a locally stable state during training [14].

B. 2T – 1R Synapse

The 2T-1R synapse, proposed in our previous work [20], as shown in Fig. 1(d), is used in our ANN design to support SGD; the synapse consists of a single memristor, PMOS and NMOS transistors, with gates connected to a common enable input e. When $e = V_{DD}$ $(-V_{DD})$, the NMOS (PMOS) turns on and $\bar{u}$ (u) is passed to the output. When $e = 0$, neither input is passed, and the output is zero. Note that u and $\bar{u}$ are complements. Further details regarding keeping the transistors in the proper regime are discussed in our previous work [20].

C. Single-Layer ANN Trained by SGD

We propose using ΔΣ neurons to achieve supervised learning using SGD. An architectural schematic of the single-layer circuit is given in Fig. 1(e). For a single-layer ANN using ΔΣ neurons, (3) is reduced to

$$\Delta W_{ji} = \eta x_i(d_j - r_j + q). \tag{5}$$

The training period consists of three phases: a read phase, an update phase, and a reset phase. During the read phase, the analog inputs x_i are fed into the ΔΣ modulator. Each modulator output is multiplexed and shorted to u input of the (i, j) 2T-1R synapse, which has its enable e latched to $-V_{DD}$ during the read phase. After being multiplied by the weights, the inputs are then summed. The sum, $r_j = W_{ij}x_i$, is held by the integrator of ΔΣ. During this phase, the read enable is set high, while the update and reset enables are set low.

During the update phase, only the update enable is set high. The data stored in the memory element, r_j, is fed into a ΔΣ$_j$. Each desired value d_j is also ΔΣ modulated. The two modulated outputs are passed through a subtractor [5], whose output is now connected to enable e. If the output of the ΔΣ modulator has the stages, V_{DD} and $-V_{DD}$, then e can have three possible values: $V_{DD,}-V_{DD}$ and 0. Each input x_i is attenuated, ensuring that the transistors conduct in the ohmic regime, and is connected to u and $\bar{u}$ inputs of the synapse. The overall learning rate η is determined by the physical properties of the memristor, represented by the model parameters, and the ΔΣ pulse width. Each update phase is followed by a reset phase, which clears the memory elements and the integrators in ΔΣ so that different input instances will not interfere.

D. Multi-Layer ANN Trained by SGD

For a ΔΣ neural network with *N* layers as shown in Fig. 2, (3) is reduced to

$$\Delta W_{ji} = \eta \delta_j x_i, \tag{6}$$

where in the outermost layer, $\delta_j^{(o)}$ corresponds to

$$\delta_j^{(o)} = \left(d_j - r_j + q\right) \cdot \frac{\partial f_j}{\partial x_j}, \tag{7}$$

and in the remaining hidden layers, $\delta_j^{(h)}$ corresponds to

$$\delta_j^{(h)} = \sum_m W_{jm}^{(h+1)} \cdot \delta_m^{(h+1)} \cdot \frac{\partial f_j}{\partial x_j}. \tag{8}$$

The quantization noise is negligible, and the derivatives of the ΔΣ transfer function can be approximated as 1 within the interval [-1, 1] and 0 otherwise. Similarly to the single-layer design, the training consists of three phases, with the signals local to each layer. During the read phase of the i^{th} layer, the

TABLE II: DATASETS USED FOR TRAINING & TESTING [26]

Dataset	Training Samples	Test Samples	Size of network	# Epochs
W. D. Breast Cancer	399	170	30 x 2	10
Wine	130	48	13 x 3	10
Iris	120	30	4 x 4 x 3	10
MNIST	60,000	10,000	784 x 100 x 100 x 10	10

TABLE III: TESTING ACCURACY (Error %)

Dataset	MATLAB Model	Delta-Sigma Activation		PWM Activation	
		Ideal	Noisy	Ideal	Noisy
W. D. Breast Cancer	2.604 %	2.447%	4.235 %	2.647 %	4.27 %
Wine	1.115 %	1.125 %	2.083 %	1.791 %	2.166 %
Iris	2.432 %	2.666 %	3.333 %	2.76 %	3.413 %
MNIST	2.54 %	2.63 %	3.851 %	2.65 %	4.012 %

remaining read, update and reset enable of the other layers are set low. The inputs to a layer are fed into a ΔΣ modulator. The outputs of the modulator are multiplied by weights, summed, and then stored, serving as the input to the next layer. This way, the input propagates from the first layer to the last.

During the update phase, the delta values corresponding to each layer are successively computed layer by layer, starting from the last. Once this is done, (6) is used to update the weights. The update phase is followed by a reset phase to flush the storage units and integrators in ΔΣ. Assume that the ΔΣ modulator output is set at a width of τ seconds. The time required to perform each of the phases on an N - layer network will be $\mathrm{N}\cdot\tau$, $(\mathrm{N}+1)\cdot\tau$, and τ_{reset} seconds, respectively.

IV. TRAINING EVALUATION

A. *Supervised Learning using SGD*

The proposed circuit was designed in Cadence Virtuoso using 180nm CMOS technology and the VTEAM model [16]. The design parameters are provided in Table I. We tested the circuit model for different scaled fully-connected networks by computer simulations (MATLAB) using the four datasets specified in Table II. The weights are calculated around an operating point, as specified in [25]. Simulations were also performed considering noise and process variations, using parameters and evaluation methodology similar to [19].

The results suggest that ΔΣ is a better contender than PWM, as can be seen from Table III and Fig. 3(a). ΔΣ neural networks offer a significant advantage over PWM in terms of speed. While the convergence during training takes a similar number of steps, each step is shorter for the ΔΣ; the PWM pulses had 10 μs duration, while the ΔΣ output had multiple pulses with a maximum total duration of 3.2 μs, providing, at least, a speedup of 3.125X, thanks to the noise shaping property, which enables ΔΣ to achieve higher accuracy in shorter time. Furthermore, the transfer function of the PWM is linear, as can be seen from Fig. 1(b), in contrast to the ΔΣ, which significantly benefits from additional non-linearity due to the quantization error. For deeper ANNs, we believe that the linearity of PWM could drastically hamper the training performance. PWM also consumes more power than ΔΣ due to the ramp function needed to generate the PWM signal. Finally, ΔΣ more closely emulates the natural spiking that occurs in the brain. We believe that the improvements will dramatically increase with the scale and depth of ANNs.

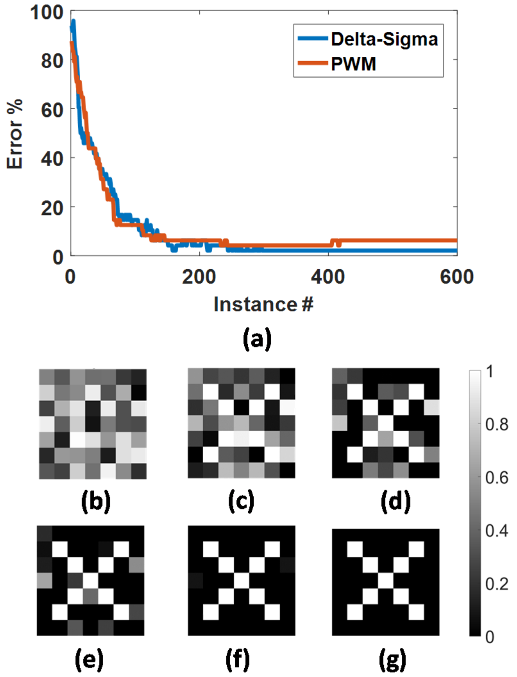

Figure 3: Learning results. (a) Weight evolution curve obtained by training for the wine dataset, where ΔΣ outperforms PWM due to error noise-shaping. (b-g) Weight matrix after 0 trials, 10 trails, 20 trails, 30 trials, 40 trials and 50 trials. After 50 trials, the weight matrix resembles the input, meaning that the network has been successfully trained.

B. *Unsupervised Learning using STDP*

To demonstrate STDP using ΔΣ, we modeled a two-layer neural network with 49 neurons in the first layer and one output neuron in the second layer, similar to [21]. The first layer, which acts as a retina, is fed a pattern resembling the one in Fig. 3 (g). Each instance of the pattern is composed of 1's and 0's, and the overall pattern resembles an 'X' when displayed as a 7x7 matrix. The input is alternated between the pattern and random noise, with each having respectively 60% and 40% probability of occurrence in an instance. Additionally, each of the individual noise sources in the 7x7 matrix have 50% chance of being '1' and 50% of being '0'.

Fig. 3 (b-g) depicts the evolution of weights over time, demonstrating the training. After 50 trials, the weight matrix retrieves the input, and the network is trained successfully.

V. CONCLUSIONS

We proposed a ΔΣ neuron for implementing real-time training in reconfigurable memristive arrays. This building block can be used for different ANN applications, as demonstrated in a multi-layer ANN, and could be scaled for deep ANNs, using SGD and in SNN using STDP. We showed that ΔΣ neurons are better than PWM, in terms of training accuracy, convergence speed, and power consumption, owing to their higher non-linearity and quantization noise.

We have explored novel ML techniques to create a modular neuromorphic platform, which is intolerant to most device and circuit variations and relies on small-scale observations in biology. In future experiments, we will analyze the effects of oversampling, noise-shaping, and using higher order modulators in deep ANNs, which can bring about a metamorphosis in real-time unsupervised learning.

ACKNOWLEDGEMENTS

This research was partially supported by the Israeli Planning and Budgeting Committee Fellowship, and by the Israeli Ministry of Science and Technology. The authors would like to thank K. Tiwari and H. A. Hanna for their help.

REFERENCES

[1] S. Yu, “Neuro-Inspired Computing with Emerging Nonvolatile Memory,” *Proceedings of the IEEE*, Vol. 106, No. 2, pp. 260-285, Feb. 2018.

[2] B. V. Benjamin *et al*., "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," *Proceedings of the IEEE*, Vol. 102, No. 5, pp. 699-716, May 2014.

[3] P. A. Merolla *et al.*, “A Million Spiking-Neuron Integrated Circuit with a Scalable Communication Network and Interface,” *Science*, Vol. 345, No. 6197, pp. 668-673, Aug. 2014.

[4] S. Park *et al*., "Neuromorphic Speech Systems Using Advanced ReRAM-based Synapse," *Proceedings of the IEEE International Electron Devices Meeting*, Vol. 25, No. 6, pp. 1-4, Dec. 2013.

[5] L. Danial, N. Wainstein, S. Kraus, and S. Kvatinsky, “Breaking through the Speed-Power-Accuracy Tradeoff in ADCs Using a Memristive Neuromorphic Architecture,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, Vol. 2, No. 5, pp. 396-409, 2018.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, The MIT Press, 2016.

[7] S. H. Jo *et al.*, “Nanoscale Memristor Device as Synapse in Neuromorphic Systems,” *Nano Letters,* Vol. 10, No. 4, pp. 1297-1301, March 2010.

[8] S. Yu and P. Chen, "Emerging Memory Technologies: Recent Trends and Prospects," *IEEE Solid-State Circuits Magazine*, Vol. 8, No. 2, pp. 43-56, Spring 2016.

[9] P. Pouyan, E. Amat, and A. Rubio, "Reliability Challenges in Design of Memristive Memories," *Proceedings of the European Workshop on CMOS Variability (VARI)*, pp. 1-6, Sept. 2014.

[10] M. B. Leslie and R. J. Baker, "Noise-Shaping Sense Amplifier for MRAM Cross-Point Arrays," *IEEE Journal of Solid-State Circuits*, Vol. 41, No. 3, pp. 699-704, March 2006.

[11] D. B. Chklovskii and D. Soudry, "Neuronal Spike Generation Mechanism as an Oversampling, Noise-Shaping A-to-D Converter," *Advances in Neural Information Processing Systems 25*, pp. 503-511, Dec. 2012.

[12] E. M. Izhikevich, “Which Model to use for Cortical Spiking Neurons,” *IEEE Transactions on Neural Networks*, Vol. 15, pp. 1063-1070, Sept. 2004.

[13] H. Jiang *et al.*, “Pulse-Width Modulation based Dot-Product Engine for Neuromorphic Computing System Using Memristor Crossbar Array,” *Proceedings of the IEEE International Symposiun on Circuits and Systems (ISCAS)*, pp. 1-4, May 2018.

[14] K. F. Cheung and P. Y. H. Tang, “Sigma-Delta Modulation Neural Networks,” *Proceedings of the IEEE International Conference on Neural Networks*, Vol. 1, pp. 489-493, Mar. 1993.

[15] J. J. Yang, D. B. Strukov, and D. R. Stewart. "Memristive Devices for Computing," *Nature Nanotechnology,* Vol. 8, No.1, pp. 13-24, Dec. 2012.

[16] S. Kvatinsky, M. Ramadan, E. G. Friedman, and A. Kolodny, “VTEAM: A General Model for Voltage-Controlled Memristors,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 62, No. 8, pp. 786-790, Aug. 2015.

[17] J. Sandrini *et al.*, “Effect of Metal Buffer Layer and Thermal Annealing on HfOx-Based ReRAMs,” *Proceedings of the IEEE International Conference on the Science of Electrical Engineering (ICSEE),* pp. 1-5, Nov. 2016.

[18] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski,, *Neuronal Dynamics: From Single Neurons to Networks and Models of Cognition,* Cambridge University Press, 2014.

[19] L. Danial, N. Wainstein, S. Kraus, and S. Kvatinsky, “DIDACTIC: A Data-Intelligent Digital-to-Analog Converter with a Trainable Integrated Circuit Using Memristors,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, Vol. 8, No. 1, pp. 146–158, Mar. 2018.

[20] D. Soudry *et al.*, “Memristor-Based Multilayer Neural Networks with Online Gradient Descent Training,” *IEEE Transactions on Neural Network. Learning Systems*, Vol. 26, No. 10, pp. 2408-2421, Oct. 2015.

[21] S. Ambrogio *et al.,* “Neuromorphic Learning and Recognition With One-Transistor-One-Resistor Synapses and Bistable Metal Oxide RRAM,” *IEEE Transactions on Electron Devices*, Vol. 63, No. 4, pp.1508-1515, April 2016.

[22] G. Indiveri, F. Corradi and N. Qiao, “Neuromorphic Architectures for Spiking Deep Neural Networks,” *Proceedings of the IEEE International Electron Devices Meeting*, Vol. 4, No. 2, pp. 1-4, Dec. 2015.

[23] R. Schreier and G. C. Temes, *Understanding Delta-Sigma Data Converters*, Wiley, 2005.

[24] S. Chakrabartty, R. K. Shaga, and K. Aono, "Noise-Shaping Gradient Descent-Based Online Adaptation Algorithms for Digital Calibration of Analog Circuits," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 24, No. 4, pp. 554-565, April 2013.

[25] E. Rosenthal, S. Greshnikov, D. Soudry, and S. Kvatinsky, “A Fully Analog Memristor-based Neural Network with Online Gradient Training,” *Proceedings of the IEEE International Symposiun on Circuits and Systems (ISCAS)*, pp. 1394-1397, May 2016.

[26] D. Dua and K. Taniskidou, UCI Machine Learning Repository, *[http://archive.ics.uci.edu/ml]*, CA: University of California, School of Information and Computer Science.