IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 11, 2019, accepted May 12, 2019, date of publication May 30, 2019, date of current version June 19, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2920076

Contrasting Advantages of Learning With Random Weights and Backpropagation in Non-Volatile Memory Neural Networks

CHRISTOPHER H. BENNETT[1,3], (Member, IEEE), VIVEK PARMAR[2], (Student Member, IEEE), LAURIE E. CALVET[1], (Senior Member, IEEE), JACQUES-OLIVIER KLEIN[1], (Member, IEEE), MANAN SURI[2], (Member, IEEE), MATTHEW J. MARINELLA[3], (Senior Member, IEEE), AND DAMIEN QUERLIOZ[1], (Member, IEEE)

[1]Centre de Nanosciences et de Nanotechnologies, University of Paris-Sud, Universite Paris-Saclay, 91120 Palaiseau, France
[2]Department of Electrical Engineering, IIT Delhi, New Delhi 110016, India
[3]Sandia National Laboratories, Albuquerque, NM 87185, USA

Corresponding author: Christopher H. Bennett (cbennett10@gmail.com)

The work of C. H. Bennett was supported in part by the Paris-Saclay Nanodesign Lidex, in part by the Chaire nanofiabilite, and in part by the Sandia's Laboratory-Directed Research and Development Program. The work of V. Parmar and M. Suri is supported by DST-SERB-Core Research Grant (Number CRG/2018/001901), Government of India, CNRS-PICS Grant and CYRAN AI Solutions. The work of M. Marinella was supported by the Sandia's Laboratory-Directed Research and Development Program. The work of D. Querlioz was supported in part by the European Research Council Grant NANOINFER under Grant 715872 and in part by the CNRS PICS Grant.

ABSTRACT Recently, a Cambrian explosion of a novel, non-volatile memory (NVM) devices known as memristive devices have inspired effort in building hardware neural networks that learn like the brain. Early experimental prototypes built simple perceptrons from nanosynapses, and recently, fully-connected multi-layer perceptron (MLP) learning systems have been realized. However, while backpropagating learning systems pair well with high-precision computer memories and achieve state-of-the-art performances, this typically comes with a massive energy budget. For future Internet of Things/peripheral use cases, system energy footprint will be a major constraint, and emerging NVM devices may fill the gap by sacrificing high bit precision for lower energy. In this paper, we contrast the well-known MLP approach with the extreme learning machine (ELM) or NoProp approach, which uses a large layer of random weights to improve the separability of high-dimensional tasks, and is usually considered inferior in a software context. However, we find that when taking the device non-linearity into account, NoProp manages to equal hardware MLP system in terms of accuracy. While also using a sign-based adaptation of the delta rule for energy-savings, we find that NoProp can learn effectively with four to six 'bits' of device analog capacity, while MLP requires eight-bit capacity with the same rule. This may allow the requirements for memristive devices to be relaxed in the context of online learning. By comparing the energy footprint of these systems for several candidate nanosynapses and realistic peripherals, we confirm that memristive NoProp systems save energy compared with MLP systems. Lastly, we show that ELM/NoProp systems can achieve better generalization abilities than nanosynaptic MLP systems when paired with pre-processing layers (which do not require backpropagated error). Collectively, these advantages make such systems worthy of consideration in future accelerators or embedded hardware.

INDEX TERMS Hardware neural networks, memristive devices, online learning, edge computing.

I. INTRODUCTION

In recent years, artificial intelligence based on neural networks has experienced considerable progress and achievements. However, training and using neural networks is associated with high energy consumption, which limits their use in low power embedded applications. Now, an emerging class of nanoelectronic elements can physically emulate synaptic features using various types of internal switching and diffusive dynamics, thus, ''nanosynapes'' [1], [2]. Using these elements as a building block, several neuromorphic architectures are possible [3]. Nanosynapses can be used for

The associate editor coordinating the review of this manuscript and approving it for publication was Laxmisha Rai.

systems trained off-line (*ex-situ*), used only for inference. Nevertheless, their most attractive use would be in learning-capable systems, *e.g*, those which can learn to generalize on new tasks in real-time using *in-situ* learning rules [4]. Notably, learning-capable systems reduce or entirely avoid reliance on external servers and/or Graphics Processing Units (GPUs) clusters which provide local accelerators pre-computed weights. In this context, *in-situ* memristor learning systems would have a broad range of applications, from implementing mathematical kernels, to building bio-realistic neural networks [5].

While simulated works explore various architectures, hardware-implemented systems with *in situ* learning are typically one-layer neural networks [4], [6]–[8]. Recently, these initial attempts have been extended to multi-layer perceptron (MLP) learning systems [9], [10], which implement approximate versions of the backpropagation algorithm [11]. Despite these successes, such systems have important constraints: slow training/convergence especially as system size (number of layers) grows [12], and deterioration relative to floating point performance due to imperfect device effects. These include device non-linearity, asymmetry, limited writable resolution, and conductance drift/endurance concerns [9], [13]–[15]. However, brains are able to learn with synapses that are highly non-linear, have limited resolution, and behave stochastically [16], [17]. This suggests that transposing learning techniques used in industrial machine learning might not be the best way to make use of nanodevices which play the role of synapses in learning systems.

One fast-learning alternative to fully back-propagating neural networks that can nonetheless achieve promising performance on a wide set of tasks in high-dimensional space, and may require less stringent synapse properties, is the Extreme Learning Machine or NoProp approach [18], [19]. Previous work has been done on porting ELM-inspired systems to nanotechnological substrates in order to take advantage of high density and energy-efficiency, such as the use of domain-wall nanowire logic systems [20], or resistive memory crossbar systems [21], [22]; however, all previous proposals critically relied upon the energy-intensive offsite computing requirement of computing a large matrix inverse. As first proposed in [23], this inverse can be locally and sequentially approximated on-chip; however, until now this approach has never been rigorously benchmarked compared to ELM software performance, or competing nanotechnology-ready algorithms *i.e.* approximate backpropagation.

Concretely, we contrast the efficiency of these categories of neural networks built from memristive nanosynapses:

- Online MLP systems trained by standard backpropagation, that learn according to a modern cost function, and with contrasting differential neuron designs
- Online NoProp systems, where the first layer realizes random weights given intrinsic device-to-device variation, a simple neuron design optimized to reduce energy consumption provides projections, and where the output layer is trained sequentially via a sign-based implementation of the classic Widrow-Hoff learning rule.

Our studied neural networks always perform both weight adaptation and inference operations *in situ*. We assess their overall performance on a standard machine learning task, with special attention paid to their requirements in terms of nanodevice bit resolution, area overhead, and final projected energy consumption over the learning experience.

We find that even if it requires more synapses and area, and therefore might appear as a wasteful idea, the NoProp/ELM system, by taking advantage of the natural properties of nanodevices to realize random projection layers, can become notably less demanding in terms of required second-layer nanosynapse properties. Moreover, it can be trained using less energy to reach similar performances than standard backpropagation systems, once device imperfections are taken into account.

The paper consists of three sections: methodology, where generic synapse models are introduced and the simulated nanoelectronic ELM and MLP systems specified; results, where the performance of these systems is demonstrated and contrasted across a variety of parameters; and discussion, where major themes are highlighted. A brief conclusion reiterates the implications of the work.

II. METHODOLOGY

A. ANALOG NANOSYNAPSES

Analog nanosynapses are nanoelectronic elements capable of storing the weights of neural network architectures. They should feature a dynamic mode (weights/conductances are written) as well as a non-volatile mode (weights/conductance are read). Device candidates for nanosynapses include floating gate transistor structures [24], scaled capacitive crosspoint arrays [25], and in our case, resistive crosspoint arrays where the analog state is stored physically as the conductance of a nanodevice. In this case, weights/conductances can be selectively modified based on the magnitude and/or polarity of applied biases [26], [27]. In order to effectively use memristive nanodevices as analog nanosynapses, devices must possess a large enough workable range, *i.e.*, usable margin to read and write between maximum conductance $G_{\max}$ and minimum conductance $G_{\min}$. They should also feature a large number of analog addressable levels (multi-bit weight resolution) within this range and good endurance, *i.e.* capability for repeatable learning cycles without quickly aging or easily destroying the device. Several varieties of industrial and academic devices including phase change memories [28]–[30], filamentary migrating oxide devices [31], filamentary devices exploiting polymeric redox mechanisms [8], [32], and ferroelectric tunnel junction synapses [33], meet these criteria and may subsequently be integrated into dense crossbars capable of on-chip learning in neuromorphic architectures.

In this work, we make use of generic nanodevice behavioral models that can nonetheless account for critical physical constraints confronting designers and users of memristive devices as nanosynapses [14], [34], consisting notably of two

mathematical models: "perfect" (linear), and "non-linear". In the "perfect" model, conductance G evolves by a fixed amount whenever the device experiences a programming operation; it is changed by

$$\Delta G_L = \frac{r}{g-1}. \tag{1}$$

when a "positive" (or SET) programming operation is performed, and by $-\Delta G_L$ when a "negative" (or RESET) programming operation is performed ($r = G_{\max} - G_{\min}$, and g is the number of writable levels). In this model, all nanosynapses may have identical $G_{\min}$ and $G_{\max}$ value and writable number of levels g; alternatively, the perfect model can also consider variability by individualizing each device with different extrema and writable depth parameters. There are some emerging nanosynaptic devices which evolve approximately linearly and symmetrically [32], [35], [36], and this model serves as their approximation.

However, in the majority of candidate analog nanosynapses, SET and RESET behavior is non-linear as a function of number of programming operations and asymmetric between these two modes for a variety of physical reasons. In the first case, SET pulses have more adaptive power (large ΔG) when the device is low conductance and less adaptive power (small ΔG) as the device approaches its maximum conductance; inversely, RESET pulses are more effective when the device is in a high conductance state [37]. In the second, diverse classes of devices contrast a more gradual/linear SET behavior with abrupt/non-linear RESET behavior, hence they have asymmetric behavior between their adaptive modes [8], [28]. To account for these phenomena, we make use of a second non-linear and state-dependent model. Conductance evolution now follows:

$$\Delta G_{NL} = \beta(G, r, g) exp(\frac{-G}{r}), \tag{2}$$

with β dependent on all terms and unique for SET and RESET:

$$\beta_{\text{SET}} = \alpha_{SET}(G_{\max} - G)\Delta G_L \tag{3}$$

$$\beta_{\text{RESET}} = \alpha_{RESET}(G - G_{\min})\Delta G_L, \tag{4}$$

where α_{SET} and α_{RESET} are adjustable parameters of the model, and thus, can be modified to implement various levels of asymmetry.

In such a non-linear model, device states can easily cluster near the conductance extrema of the devices. To form a basis of comparison between the different models of nanosynapse, α_{SET} and α_{RESET} constants were used to fit the curve with more than g discrete states along the entire space, but only g writable levels within the remaining, quasi-linear weight space (clipped within 10% of either extremum).

B. NANOELECTRONIC LEARNING IMPLEMENTATION

To explore computing possibilities of networks built from nanodevices the MNIST database of handwritten digits [38], consisting of 60, 000 training digits and 10, 000 test digits, was selected. The neural network architecture consists of:

- A number of input neurons, L. For the MNIST task, $L = 784$ is set.
- A number of hidden layer neurons, M performing projection and/or gradient accumulation, where M can vary as a parameter.
- A number of output neurons N. Again for MNIST, $N = 10$ is set according to the ten digit classes.

Like in standard computer vision tasks, images are vectorized before system presentation. A simple analog encoding scheme maps images on a pixel-by-pixel basis into voltage values. Given a numeric input channel or pixel of index i, X_i a voltage within the range $-V_{\text{read}} < V_i < +V_{read}$ is assigned:

$$V_i = 2V_{\text{read}}(X_i/L_{\max}) - V_{\text{read}}, \tag{5}$$

where $L_{\max}$ is the maximum pixel intensity, and V_{read} is a voltage that does not alter nanodevice conductances.

1) IN-SITU CROSSBAR INFERENCE AND ADAPTATION

As neural networks require negative weights, and conductances are physically positive values, we associate nanosynapses in pairs of differentially accessed devices. A crossbar of closely connected paired nanosynapses can be sequentially or simultaneously accessed in both programming modes, when applied voltage pulses are greater than thresholds, or inference mode, when applied pulses are below them. Crossbars in this configuration can perform *in-situ* dot product operations [31], [39], making them a natural building block for on-chip neural networks.

This works because a collective output current is obtained naturally through Kirchhoff's laws when an array is voltage-biased. For instance, if the differential synapses are arranged along two output lines, one positive and one negative, then the output at a given output neuron of index k will be:

$$Y_k = \sum_{i=1}^{N} W_{i+,k} X_i - \sum_{i=1}^{N} W_{i-,k} X_i. \tag{6}$$

where $W_{i+,k}$ and $W_{i-,k}$ are physically analog conductance values ($G_{i+,k}$, $G_{i-,k}$) (the bias can classically be included by including a constant input).

Online learning in such a physical neural network takes place by repeatedly alternating between inference and programming modes. In the former, training images are applied and the system output immediately demonstrates the error cases; in the latter, appropriate programming pulses are applied. During training stages, the final logic output of each neuron, O_k, is a logic signal representing its sign $(-1, 1)$. This can be obtained using a circuit with a transimpedance amplifier [8], [31]. When compared to expected T_k, the appropriate error case, if any, is revealed. For ANN applications, during testing or inference stages, the sign-based output is no longer appropriate. The analog values amongst all output neurons of the ultimate layer must be compared, and the output neuron with the highest value should be selected as the output of the network. This approach, also referred to as

max-out or winner take all, is a staple of learning operations in neuro-computing.

Simultaneous pre-synaptic and post-synaptic voltage programming pulses can naturally implement learning or adaptation across a crossbar array. The precise voltage pulse magnitude, polarity, and number of steps applied to the periphery of the array to properly implement learning depends on the chosen nanodevice and learning policy. Bipolar memristive devices require two consecutive array-wide programming steps, while unipolar memristive devices require only one, as illustrated in [40]. The combination of post-synaptic programming voltage pulses V_p and pre-synaptic, V_n, reduce error within all pair weights simultaneously, satisfying a simplified version of the Widrow-Hoff learning algorithm:

$$\Delta W_{i,k} = \Delta G sign(X_i(T_k - O_k)), \tag{7}$$

where $\Delta W_{i,k}$ is the weight change at each programming step for the synapse connecting input i to output k, T_k is the expected output, O_k the actual output, X_i the input value, and ΔG the conductance change. Since ΔG varies on a device-by-device basis, every device has an intrinsic learning rate. The rule is therefore not physically binary, but update polarity relative to the combination of post and pre-synaptic error is binary (has two directions). For clarity, we refer to the rule hereafter as sign Widrow-Hoff (s-WH).

2) PROJECTION/REGRESSION SYSTEM (NoProp)

Our first considered system, NoProp uses a two-layer neural network (Fig. 2(a)). The weights from the first layer W_{in} are random, and the state of the jth hidden neuron (of M total) is given by:

$$H_j = f(\sum_{i=1}^{N} W_{in,ij} X_i), \tag{8}$$

where f is a non-linear activation function. As noted in [22], in a nanodevice-based implementation, first layer random weights can be efficiently physically realized by using the natural dispersion of conductances around G_{on} and G_{off}. For our simulations, we used the dispersion around the G_{off} state, using measurements extracted from the physical devices of [8]. Projections from this first layer are then transformed from analog currents to voltages. Here, we used the simplest activation function to implement: $f(x) = sign(x)$.

In *ex situ* learning, the weights of the second layer W_{out} are computed using a Moore-Penrose pseudo-inverse, or a regularized form of ridge regression. Solving this pseudo-inverse all at once or incrementally [41] in an off-chip context involves substantial memory and computing overhead. In addition, transferring collected activation functions to the external computing cell and returning computed weights incurs substantial energy expenditure. Here, we propose a learning scheme especially adapted for *in situ* learning on-chip. Our rule uses the high-dimensional projections from W_{in} and the labels supplied at the network's output to implement the binary Widrow Hoff throughout the regression layer W_{out}. This set-up is very similar to the rule proposed in [42] for hardware ELM learning, but requires no external normalization or learning rate parameters (these are provided naturally by nanosynapse properties).

As in Fig. 2(a), when W_{out} is biased, post-synaptic currents (Y_k) are obtained and converted into voltage outputs O_k. When compared to expected T_k, error cases for every post-synaptic neuron are immediately available and a conditional pulse programming scheme can then correct the error case in each pair. Note this learning rule is also sparse in the sense that, if $sign(T_k) = sign(Y_k)$, as in the right-most column of 2(a), no adjustments are taken in the synapses connected to that sign-equivalent output neuron.

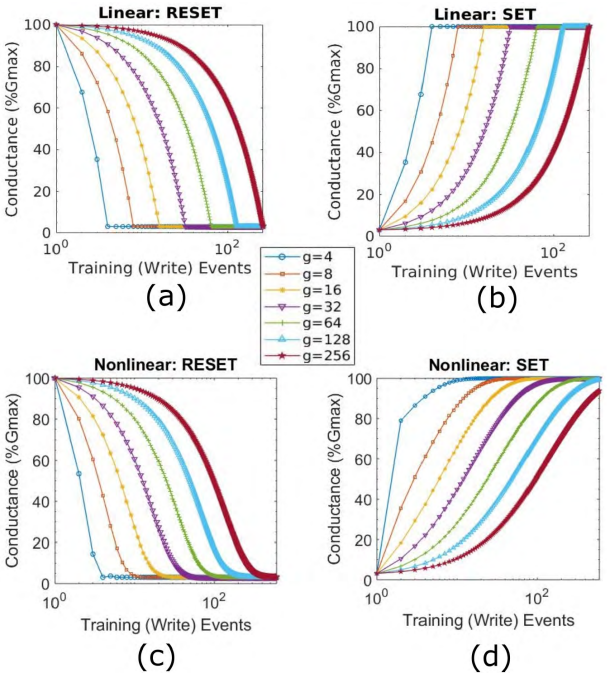

FIGURE 1. (a) and (b) show jump tables for device evolution starting at $G_{\text{on}}/Gmax$ and $G_{\text{off}}/Gmin$ conductance respectively using the linear model (Eqn. 1); (c) and (d) depict the same but for the non-linear case (Eqn. 2) where ΔG is now modulated by the device's state relative to its extrema.

In order to build a full range of projection weights available for the hidden layer, we achieved best results with a scheme called Simultaneous read-out (SRO). In SRO, pairs of projection lines are used, as in 2(a) (this doubles the area overhead required for the projection crossbar). Read out in this case is instantaneous, but the overhead is higher (crossbar requires $2M$ hidden layer read-out wires). This scheme is pictured in Fig. 1(a).

3) MULTI-LAYER PERCEPTRON (MLP) BACKPROPAGATION SYSTEMS

Our second considered system implements a multilayer perceptron trained by backpropagation in nanoelectronic form

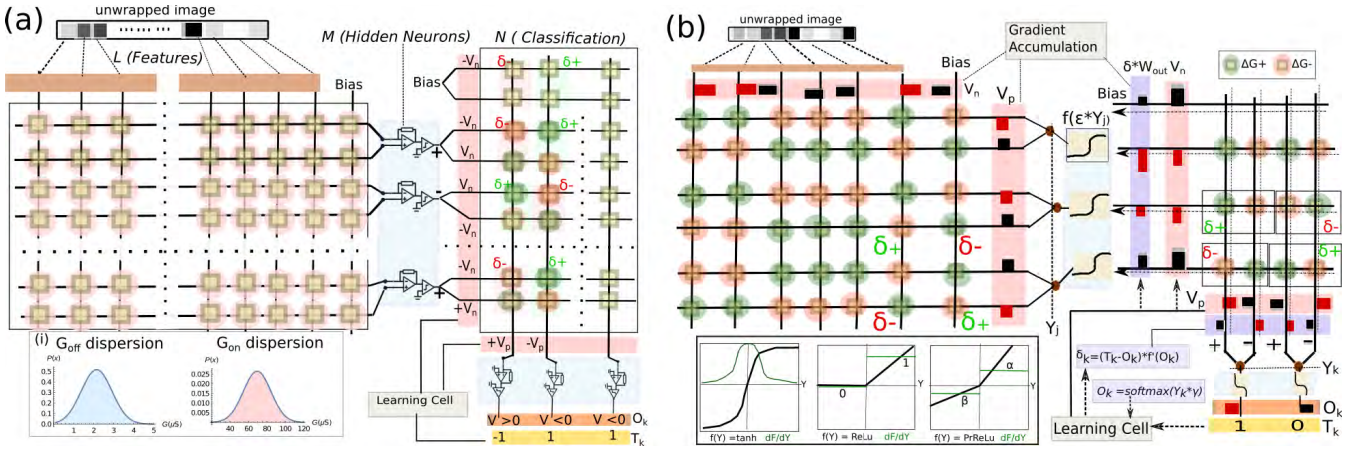

FIGURE 2. **(a) Shows NoProp system, with input (dimension *L*, index *i*) being projected to an *M* dimensional space (index *j*) via fixed weights matrix *Lx2M*; adaptive read-out performs multiple linear regression from *M* to the classification boundaries provided by the *N* ultimate output neurons (index *k*). The bottom inset shows the Gaussian dispersions of conductance that can be used to naturally set the fixed weights; example data from [8]. (b) Shows MLP system with analog neuron and softmax training design, where error is now being backpropagated (arrows and purple computations). The bottom insets show the differentiable functions and first derivatives we have considered.**

(Fig. 1(b)). The architecture of the system is similar to the NoProp system, with additional requirement that W_{in} is trained via backpropagated error. In this work, we chose a cross-entropy (log-loss) cost function, with softmax one-hot encoded outputs. The circuit output O_k is obtained as

$$O_k = \frac{\exp^{\gamma Y_k}}{\sum \exp^{\gamma Y_k}}, \tag{9}$$

where Y_k is obtained as in Eq. 6, and γ is a normalizing parameter. This choice indeed leads to excellent performance in terms of machine learning [43], and can map relatively naturally to crossbars of memristive nanodevices. The choice of softmax leads to learning rules that are simplified with regards to apparently simpler output choices. Following standard backpropagation calculations, the weights in the output layer are adapted as:

$$\Delta w_{j,k} = -\eta \delta_{j,k}, \tag{10}$$

where η is a learning rate, and $\delta_{j,k} = H_j(O_k - T_k)$. For the input layer,

$$\Delta w_{i,j} = -\eta \sum_k \delta_{j,k} w_{j,k} f'(H_j), \tag{11}$$

where H_j is the activation of that middle layer neuron, and $f'(H_j)$ its derivative. A drawback is that performing softmax computations on-chip can be a significant cost in terms of circuit overhead and energy. One such implementation that is relatively efficient has been described in [44], yet the computation cost of 690 μW is still high. For our energy analysis, we have assumed a parallel and energy favorable alternative, which exploits the natural exponential ability of sub-threshold CMOS devices [45].

A critical constraint in on-chip backpropagation is that activation functions must be differentiable. The *sign* function used in the projection network cannot meet this criteria; therefore, we introduce and compare two additional neural activation functions in the context of our MLP systems: *Analog* neurons, which may be the *tanh* activation function, a rescaling of the classic invert logit (sigmoid) function, or the sigmoid function itself, or *Digital* neurons, in particular the rectifying linear function. To electrically realize analog neural functions, we discovered a trade-off between the smallest area circuits, *e.g*, a resistor and two transistors [46], and those which provide the closest match to the mathematical activation function. For the purpose of our energy calculations, we then used a scaled version of the circuit proposed in [47], which achieves reasonable efficiency in energy and area (6 transistors) while better emulating the function using triode and saturation regimes.

Another difference between the binary and analog differentiable neurons involves the ease of computing and storing their derivatives. While analog neurons can easily have complex ($e.g\, f'(h_j) = 1 - tanh(x)^2$) derivatives requiring many bits to store, rectifier gradients are binary, which reduces constraints on accompanying analog CMOS circuitry. For systems using the rectifying linear function, which has achieved state of the art results in machine learning [48], positive outputs are instead projected by a constant α, hence $\frac{dF}{dx} = \alpha$, and negative ones by a constant β, thus $\frac{dF}{dx} = \beta$. Both are visible in the bottom inset, Fig. 2(b).

The sum in Eqn. 11 used to propagate error backward, can also be performed *in-situ* using the dot product operation [49], as also visible in Fig. 2(b). Nevertheless, the calculation of gradients for every pair of nanosynapses and its on-chip implementation is non-trivial, and is a considerable circuit overhead cost relative regards to the simpler NoProp system.

During learning, we adjust weights in accordance with a backpropagation-friendly version of sign-WH ($sign(\delta_k x_{j,k})$) within both layers [50]. Additionally, in the following simulations we always implement mini-batch stochastic gradient descent by grouping training images into mini-batches of $b = 100$ images, accumulating error while the n samples

are presented, and applying a single programming cycle on both layers (crossbars) after this multiple-inference period. However, in hardware, minibatches requires associated memory or capacitor devices to store intermediate error gradients (by storing only the sign, this complexity is greatly reduced).

C. TESTING AND SIMULATION METHODOLOGY

The testing phase reveals the generalization ability of the system and consists of repeated inference operations. Unknown samples are applied one at a time for T total tests, passing through W_{in}, activating each respective hidden layer neuron, and passing through W_{out} to produce $\vec{O}$. The answer is $max(\vec{O})$; if predicted neuron k is the actual label, counter c is incremented. The final testing percentage is $\frac{100\ c}{T}$.

Our nano-synapse models have been integrated within a crossbar simulator. Because the devices are non-volatile during the inference steps, only active moments (programming windows) track the evolution of weights and use fixed time integration. Devices in learning layers are always initialized in a random state between each of their respective extrema. To smooth the effect of random initial conditions (device conductances) upon learning outcomes, Monte Carlo simulations were run with different seeds. The mean of five system's learning performances is shown in the following section unless otherwise noted. For off-chip comparisons, scripts were written in Python, integrating functions available in the TensorFlow packages to build software ANNs with floating point synapses [51]. Software gradient learning system weights were always obtained with the same activation function and hidden layer dimensions, so as to make a fair comparison with the companion nanosystem, and an appropriate, constant learning rate. For off-chip imported or software comparisons, NoProp second-layer weights were always obtained analytically via regularized ridge regression.

III. RESULTS

A. EFFECTIVENESS SIGN-BASED ON-CHIP TRAINING

Previous work [52], [53] has argued that sign-based gradient learning can serve as a satisfactory on-chip approximation for fully featured gradient descent in multilayer perceptron systems. We confirm this result, as our MLP results with a constant learning rate and using *tanh* and *ReLU* hidden neurons can achieve within 0.2% and 0.8% of the results obtained with software synapses in Tensorflow, as visible in Table 1. For the first time, we analogously confirm the validity of our sign-WH appropriate on-chip learning with NoProp/ELM systems. We have directly contrasted the ability of our proposed system, which learns locally with only pre-synaptic activation and post-synaptic error values obtained from labels as in Fig. 2(a), to a comparison system which collects hidden layer activations, computes output weights off-chip analytically and then imports them before inference. As visible in Figure 3, we find that the on-chip method can perfectly approximate this result with linear synapses, while requiring about twice as many examples. In addition,

TABLE 1. Direct comparison of systems.

Nanosynapse System	Linear synapses	Non-linear synapses
Single crossbar/layer	87.1%	84.9%
NoProp[a], SRO	94.5%	92.1%
MLP, best$tanh$[b]	96.3%	91.7%
MLP best Rectifier [c]	93.9%	91.1%
MLP $ReLu$ [d]	93.2%	90.2%
NoProp Software	94.3%	n/a
MLP Software: $tanh$	96.5%	n/a
MLP Software: $ReLu$	94.0%	n/a

Performances for all examined systems are directly contrasted on the same challenge (10k digits of the MNIST testing set), with mean values ($n = 10$) for the linear and non-linear synapse models.
[a] $M = 3000$ hidden layer, system given 3 epochs to learn, $g = 256$.
[b] $M = 300$ hidden layer, $tanh$ function uses $gf = 550$, system given 8 epochs to learn, $g = 256$., samples presented in mini-batches of $b = 100$)
[c] $M = 800$ hidden layer, Hidden layer function uses linear derivative $\alpha = 1.5, \beta = 0.1$ system given 8 epochs to learn, $g = 256$.,samples presented in mini-batches of $b = 100$,
[d] Same as[c] except hidden layer functions implement $\alpha = 1, \beta = 0$.

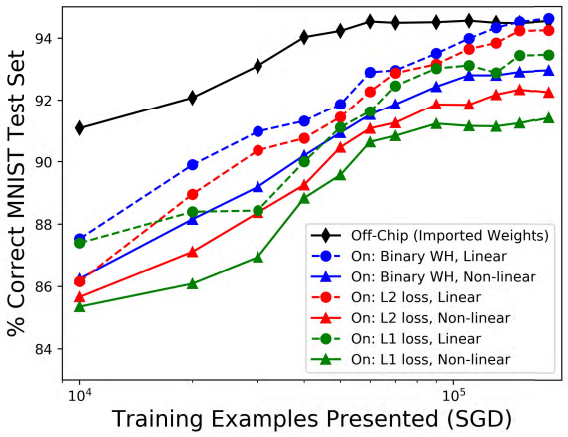

FIGURE 3. On-chip learning policies: sign Widrow-Hoff, L2, and L1, are compared to each other and inference results when output weights computed off-chip. In all cases, $M = 3000$, $g = 256$, no learning rate used. Normalization is provided by the device properties.

we have contrasted our proposed sign-WH approach with two competing and more complex on-chip learning policies. In contrast to learning by the proposed $sign(X_i(T_k - O_k))$, in L1 learning, weight updates are scaled on a neuron-by-neuron basis by $X_i abs(T_k - O_k)$; and L2 learning, or mean square error, where weight updates are scaled as $X_i(T_k - O_k)^2$. In addition to their extra complexity, as visible in Figure 3, these rules perform inferior to sign-WH learning for both linear and non-linear devices in the second layer.

B. BEST PERFORMANCE ON TASK

Fig. 4(a) demonstrates the convergence of *in-situ* NoProp systems of varying synapse quality alongside a fully software neural network. As visible, this *ex-situ* learning approach is faster to converge; at $M = 3000$, maximum 94.3% performance is analytically obtained given around 50, 000 samples. In contrast, a full three epochs or 180, 000 training samples

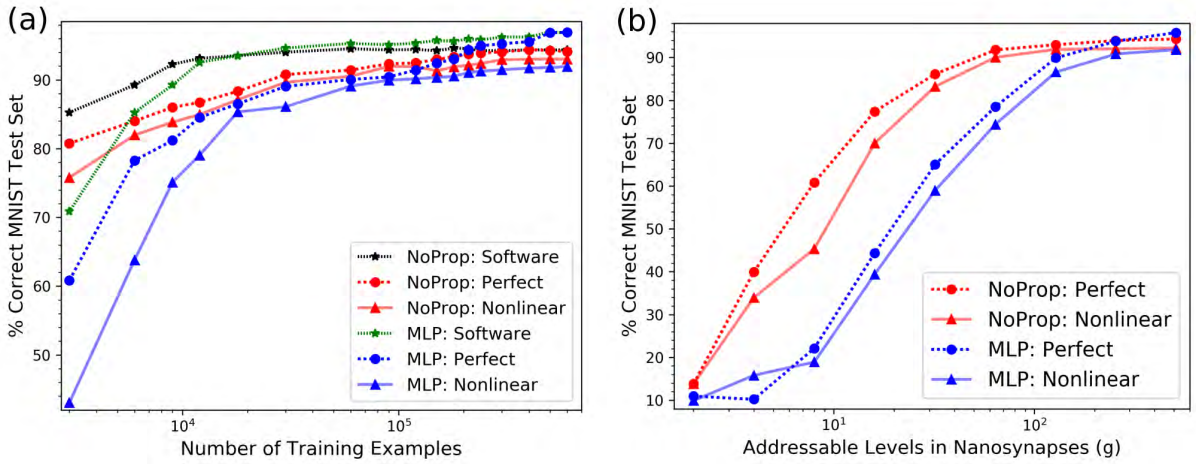

FIGURE 4. (a) Shows convergence of several studied online-learning systems shown in contrast to their software cousin systems, which learn with floating point software synapses and gradients. Pictured MLP systems use analog (*tanh*) activation (b) Shows reliance of the two considered online algorithm(s) and their two varieties of synaptic evolution on synaptic resolution g in the adaptive layer(s) (all synapses have equivalent g).

are required to achieve on-chip convergence, which slightly outperforms the software system at 94.5%.

Using the mini-batch stochastic gradient descent learning policy (mini-batches of $b = 100$ images), cross-entropy and softmax error as the cost function and following sign-based gradient descent, we examined the success of several MLP systems using varying hidden layer activation functions. The strongest performing system used *tanh*; this system achieved superior performance on the test set of the MNIST database as compared to the NoProp system as well as others, at mean 96.3% recognition- but only when the system is built from perfect (linear) devices. As visible in Fig 4(a), this system approaches the performance of a software batch learning system after 6-7 epochs (360k-420k training samples). When using the non-linear model, this performance advantage disappears. Moreover, as visible in Table 1, the generic (*ReLU*) and best rectifier system struggle to completely match software results. Moreover, these systems slightly under-perform the NoProp systems when the perfect model, and substantially under-perform them when using the non-linear model.

C. DEPENDENCE ON DEVICE/SYSTEM PARAMETERS

In the earlier results, $g = 256$ or 8 bit synapse resolution was assumed, but it is not always the case that emerging nanosynapses are of this quality. In addition, estimations of emerging nanosynapse bit resolution can substantively vary depending on the device class and the measurement style. As a concrete example, for filamentary ReRAM devices using oxygen anions, values as high as 6-7 bit [54] and more modestly 4-5 bit [55] have been reported. Given this uncertainty and the importance of this constraint, we vary g between 2 bits (binary operation) and 9 bits ($g = 512$). Fig. 4(b) shows the dependence of the best performing MLP system (using *tanh* activation) and NoProp system within this range, for both linear and non-linear synaptic behavior cases. At 4 bits, NoProp systems approach 80% accuracy, which is still poor, but MLP systems fail catastrophically. At 5 bits, NoProp systems perform within 5% of their maximum possible accuracy, while MLP systems still miss 20% of the examples on the test set. At 6 bits, NoProp systems are only within 1-2% of their maximum achievable values, even given non-linear device behavior; MLP systems learning with linear synapses require 8 bits ($g = 256$) to do the same, while 9 bit ($g = 512$) is required to compensate for learning with non-linear devices. This is a significant contrast which is explained by the varying requirements of the two learning algorithms. While Widrow-Hoff constructs a hyperplane to separate classes, gradient systems traverse a more complex error landscape, and dependencies are created between the layers during the training process.

Uniquely for MLP learning systems, synapse requirements for W_{in} and W_{out} can be contrasted. Fig. 5(a) demonstrates this analysis visually for the case of the *tanh* hidden layer activation and perfect synapses, where the x axis is the first (input) layer depth g_1, the y axis is the second(output) layer depth g_2, and the z value at each place on the grid search is the generalization ability on the MNIST test set. As visible, the requirements for synapse depth are notably higher in the first (W_{in}) layers rather than the second (W_{out}) layers. For instance, less than $g = 100$ can be an appropriate value for the second layer weight resolution if first layer resolution is high; however, any system with such low resolution in the first layer would necessarily perform poorly. At this stage, we are not certain whether the higher synapse quality requirement of the first layer is a property of the backpropagation algorithm itself, or a numerical effect as $\delta_{\vec{k}}$ is transformed into $\delta_{\vec{j}}$ via the vector-matrix multiply ($\delta_{\vec{k}} * W_{\text{out}}$).

Varying the hidden layer size M can have a strong influence on the quality of the learning outcomes. Fig. 5(b) shows that only NoProp systems with larger hidden layers generalize well on the test set. Only when $M = 2L$ is benefit over the

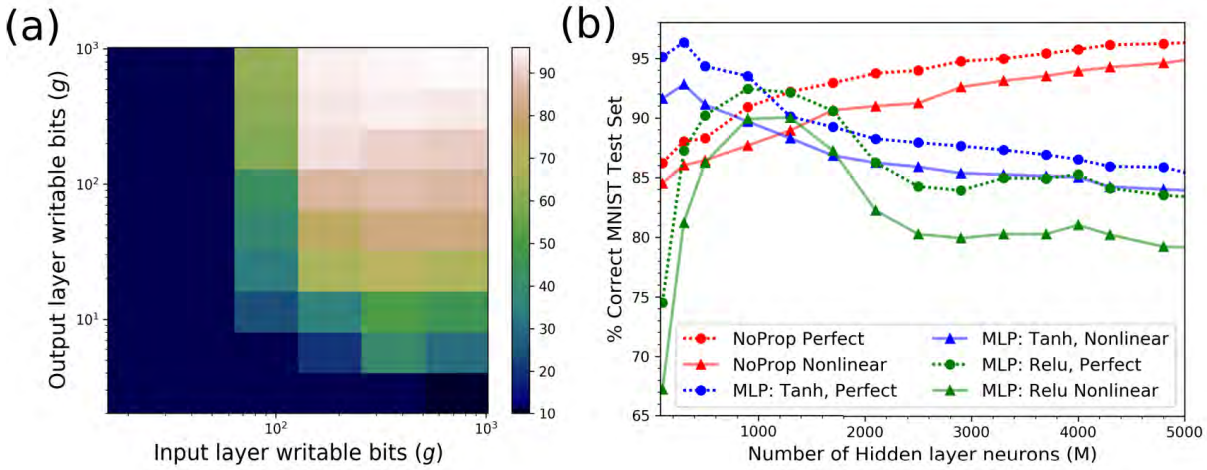

FIGURE 5. **(a) Demonstrates a grid search of possible synapse depths in two adaptive layers of an MLP system; the coloring represents the average performance of $n = 5$ systems learning with synapses at the resolution of those coordinates, as noted in the colorbar. (b) Demonstrates dependence of all considered systems on the hidden layer dimension M; g is always fixed at 8 bits for all systems in these simulations.**

one-layer parallel perceptron system evident. At very small hidden layer sizes, performance is actually inferior to one-layer, direct task presentation (Table 1). Yet at dimensions of $3L$ and higher, the system improves towards 95% classification accuracy, in the case of linear devices, and 93.5%, for non-linear devices.

The quality of generalization of gradient learning systems also depends on M. As visible in Fig. 5(b), analog differentiable activation functions perform optimally at a smaller size, between $M = 100$ and $M = 500$, whereas rectifier systems perform at their best between $M = 600$ to $M = 1500$. In the former case, too large systems suffer from over-fitting, as a number of over-complete neurons leads to confusion in learning outcomes. In the latter case, rectifier hidden layers implement a form of implicit "drop-out"- a form of normalization that battles over-fitting. Explicitly, synapses in the first layer connected to activations for 'dead' (negative) neurons - ReLu ($\beta = 0$), or asymmetrically small outputs ($\beta < \alpha$) do not adjust, or hardly adjust to weight updates, respectively.

D. RESILIENCE TO DEVICE NON-IDEALITIES

As suggested in [56], non-volatile memory neural networks are relatively in-sensitive to global device imperfections *e.g.* inter-device variability, while being very sensitive to global effects that steer the ability of all synaptic devices to adapt appropriately. In this section, we quantify these sensitivities in terms of the two considered learning algorithms.

1) SENSITIVITY TO INTER-DEVICE VARIABILITY

We consider two components of device-to-device variability: increasing variability within device-to-device writable range as extrema values (G_{Max}, G_{Min}) become increasingly disperse, and increasing variability within device-to-device writable capacity g. In the first case, every nanosynapse now samples randomly from a Gaussian distribution of possible maximal and minimal conductance values; in the second, every device draws a unique g value from a Gaussian distribution centered around $g = 256$. In comparison to the simulations producing Fig. 4(b) and Fig. 5(a), which assigned g uniformly to assess overall impact of synaptic depth, this assesses sensitivity to g's inter-device variability.

As variability in these parameters increases, every device will reach increasingly different extrema during the process of its adaptation, and adapt at different rates, respectively. As visible in Fig 6. (a),(b), the NoProp systems are notably more resilient to both variability effects, with the MLP-inspired system being mildly harmed by g dispersion, but significantly harmed by the extrema dispersion. Extrema variability is especially difficult to cope with in systems with non-linear synapses since many devices can easily become 'stuck-on' at high G_{Max} values, disturbing correct inference. This can be partially accounted for by clipping the weights to a common, quasi-linear region away from these extrema, as in [8].

2) SENSITIVITY TO PROGRAMMING MODE ASYMMETRY

During programming-mode asymmetry, devices fundamentally respond differently when being SET (increasing conductance) as opposed to RESET (decreasing conductance). We have simulated this effect by assuming RESET-mode scaling by an asymmetric variable ζ. For non-linear devices:

$$\beta_{\text{RESET}} = \zeta * \alpha_{RESET}(G - G_{\text{min}})\Delta G_L, \qquad (12)$$

And for linear ones, which is normally symmetric, asymmetry has been added by specifying a different ΔG_L value for RESET operations, as weighted by the respective ζ value. As visible in Fig. 7(a), even small asymmetry values can quickly overwhelm the adaptive power of both considered online learning systems, but gradient learning systems suffer

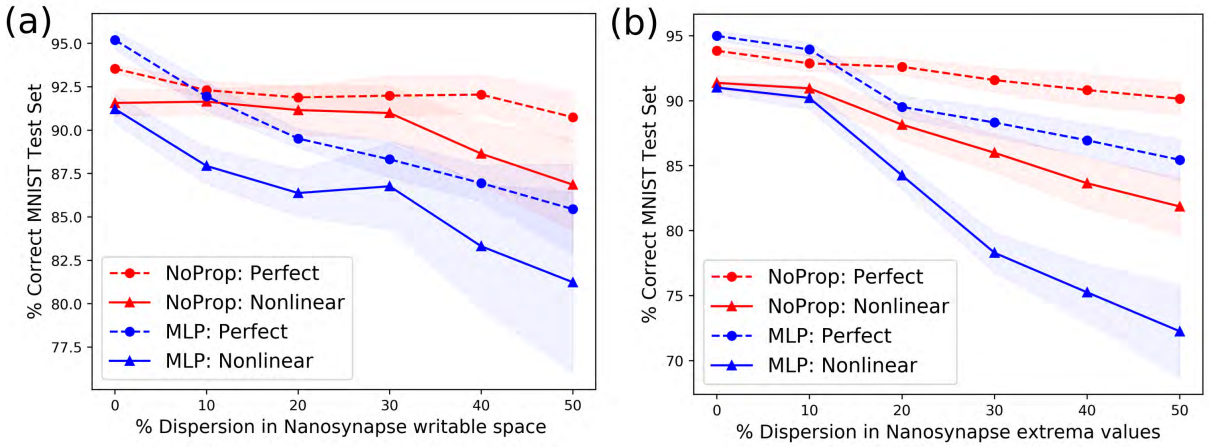

FIGURE 6. (a) Demonstrates resilience to increasing variaiblity in nanosynapse analog capacity; every adaptive device possesses slightly different g values at the σ dispersion shown around mean $g = 256$ (b) Demonstrates resilience to increasing variability in extrema values G_{max}, G_{min}, which both vary at the σ dispersion shown. Shaded regions depict standard deviations from mean points.

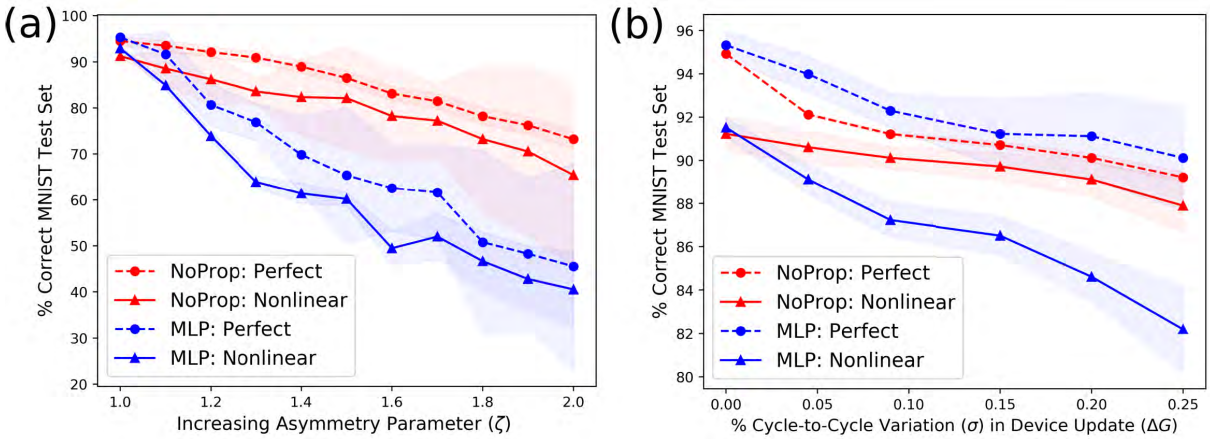

FIGURE 7. (a) Shows system performance as ζ scales RESET behavior asymmetrically to SET; (b) shows effect of cycle-to-cycle noise on every device's individual update at distribution width σ. All systems learn with $g = 256$; NoProp has $M = 3000$, MLP $M = 300$.

more in comparison. For instance, when ζ is 1.25, or RESET operations are 125% more powerful than SET operations, MLP performances with non-linear synapses fall below 75%, while the best performing, NoProp with linear synapses, is still at about 91%. At more dramatic values, e.g. $\zeta = 1.5$, all considered systems generalize less well than a single layer network learning with symmetric synapses.

3) SENSITIVITY TO WRITE-CYCLE STATISTICAL VARIATIONS

Finally, we consider the case where individual device-by-device updates are different than the ideal ones implemented according to the algorithm, whether due to random device effects, or imperfect electrical behaviors in the crossbar, e.g. parasitic capacitance or currents. This cycle-to-cycle or write variability is added by

$$\Delta W_{i,k} = c * \Delta G sign(X_i(T_k - O_k)), \tag{13}$$

where c is a small scalar value that has been drawn from a Gaussian distribution with its mean from an average ΔG value considered by calculating globally, for the perfect model, or from the initial device value, for the non-linear one, and the σ as varied. Note that, as the device updates are very small, at large σ, sign-flips contrary to the s-WH rule are possible. As visible in Fig. 7(b), broadening the write-cycle variability has a non-negligible effect on the ability of the network to generalize. However, only in the non-linear MLP case does the performance drop off drastically, *i.e.* to substantially less than a single layer network could achieve.

E. LEARNING ENERGY ANALYSIS

A primary energy cost during the learning phase of the system comes from the programming of the memories. We collected write energy budgets, along with typically smaller read energy budgets, from the literature or computed it using the approach in [57], for a suite of candidate emerging nanosynapses that have sufficiently low-voltage operation modes, and may be scaled appropriately for ultra-dense learning set-ups. Specifically, we examined two 'perfect' or highly linear emerging devices: the polymeric ENODe device described

in [32] and a novel lithium non-volatile transistor (NVT) device called 'LISTA' [58], alongside two more non-linear but fully scaled existing NVM devices, the ferroelectric tunnel memristor [33], and the tantalum oxide ReRAM device [59]. Notably, switching programming pulses for each of these (V_{prog}) are 10 pJ, 10 aJ, 1.48 pJ, and 12.5 pJ respectively (note that LISTA's low write pulse is somewhat compensated by a far larger read pulse energy). Assuming an access device on each output neuron minimizes parasitic current losses [60], [61], each learning mode step requires a serial (column-by-column) write in both analog arrays for the MLP system and the second only for NoProp, and fully parallel inference/read operation in both systems. By scaling these basic operation costs up to respective system dimensions, as well as including the additional energy costs for the studied CMOS companion systems (ultra-low power *sign* neurons for NoProp systems, digital or analog hidden layer neurons for MLP systems, and additional soft-max operation for MLP), we have estimated energy costs for learning.

Regarding the neuron designs, our NoProp digital design uses the simple, low-power CMOS inverter design proposed and successfully demonstrated in [62]. This system has a total energy footprint of less than 10 fJ per neuron. For analog gradient systems, our studied sigmoid circuit uses input in the form of current and gives a output in form of a voltage closely modeling the sigmoid function. It is based on a modified version of the circuit provided in [47] at the CMOS design node of 90 *nm*. In order to verify functionality, DC analysis was performed while varying input current from $-100\mu A$ to $100\mu A$; the results showed an almost complete swing between $0V$ and $1.25V$. For obtaining power values, transient analysis was performed using a pulsed current source with 100 ns period for a duration of 10 μs (100 cycles). These yielded the following leakage and dynamic energy values, respectively: $40\mu W$, and 43.8 pJ.

For the digital hidden layer neuron design, a 16-bit fixed-point ReLU VHDL activation function block was designed based on an earlier version proposed in [63]. Estimates for the block design were extracted with dynamic power based on activity file extracted from testbench, taking into account leakage power. The energy estimates for leakage power and dynamic energy were $4.2\mu W$, and 0.151 pJ, respectively.

Table 2 shows final energy costs for a variety of devices and highlighted learning policies (note that, as the ENODe device had almost equivalent energy costs as the *TaOx*, it is not shown). It demonstrates that No-Prop learning systems always save energy compared to MLP systems within the same device class. The energy superiority of NoProp systems stems from three basic advantages: energy-savings of faster convergence (Fig. 4(a)),, energy-savings from only having to adjust weights in one of two layers, and energy-savings from ultra-low power *sign* neurons, which almost wholly offset the impact of the very large M in these systems. Table 2 also emphasizes that the choice between single-example and batch-style programming, as well as the chosen device's elementary programming costs, can have more

TABLE 2. Energy fingerprint of systems.

System	Ferroelectric	TaOx	LISTA
NoProp, SGD[a]	$6.28mJ$	$0.053J$	$1.8\mu J$
NoProp, Batch[a]	$127.68\mu J$	$1.09mJ$	$1.78\mu J$
MLP, SGD Analog[b]	$0.75J$	$6.31J$	$7.94mJ$
MLP, SGD Digital[c]	$3.41J$	$28.86J$	$184.39mJ$
MLP, Batch Analog[b]	$15.3mJ$	$70.9mJ$	$7.91mJ$
MLP, Batch Digital[c]	$34mJ$	$0.288J$	$91.52\mu J$

Every value shows the total energy cost of learning considering core energy costs of powering the two neuromorphic arrays and the studied companion CMOS circuitry (note: not all accessory systems were considered). First column shows computed energy costs with fast, intermediate-voltage, high resistance ferroelectric devices; second, with very fast, intermediate voltage TaOx devices; and last, very low voltage, very fast LISTA NVRT devices.
[a] $M = 3 * L$ hidden layer, system given 3 epochs to learn, $g = 256$.
[b] $M = 300$ hidden layer, $tanh$ function with $gf = 550$, 8 epochs training $g = 256$; batch size $b = 100$ (where used)
[c] Same parameters as analog learning but ReLu activation function is used ($\alpha = 1, \beta = 0$) and larger hidden layer $M = 800$.

significant implications on the final energy costs of learning than just this learning policy choice, however.

Fig 8(a) shows the final energy and performance fingerprint of all considered nanosynapse multi-layer learning systems as a function of the device's core read/write costs: driven by learning policy and the device model (TaOx and ferroelectric employed the non-linear analytical model, while ENODe and LISTA used the perfect/linear one) along with the required system periphery. As visible, a wide variety of outcomes are possible, with the most promising ones represented by the academic LISTA device. However, NoProp learning with the Ferroelectric and TaOx synapses represents a promising application point as well, as it avoids the low-performance, high-energy area that several of the systems built with the same synapses fall into.

Fig. 8(b),(c) demonstrates a per-batch energy budget ($b = 100$, that is 99 inference steps followed by one programming event) for the Ferroelectric NVM device for NoProp (b) and MLP analog (c) learning policy. Meanwhile, Fig. 8(d),(e) demonstrate the same policy contrast but for the ENODe NVM device. Note that, in both cases peripheral neuron energy is negligible in the NoProp case but significant in the MLP case. However, the relative dominance of read v. write energy budgets is device-dependent; for instance, ENODe has far more equivalent read-write operation costs. In the case of batch-learning, this information could be used to calibrate trade-offs between accuracy or energy-savings. Promisingly for on-chip MLP set-ups, the softmax operation, which may have been a large budget with a standard CMOS design, constitutes a negligible part of the per-batch programming budget when using the proposed sub-threshold design.

Note that our objective here was not to provide a definitive estimate of power for these systems by including all accessory sub-circuits and modules as in [64]. However, as these outstanding sub-systems, e.g., accessory circuits for batch-learning, or the logic core for on-chip learning, would be constantly implemented regardless of policy, these power estimates should still serve as meaningful benchmarks for base operations of these NVM accelerators.

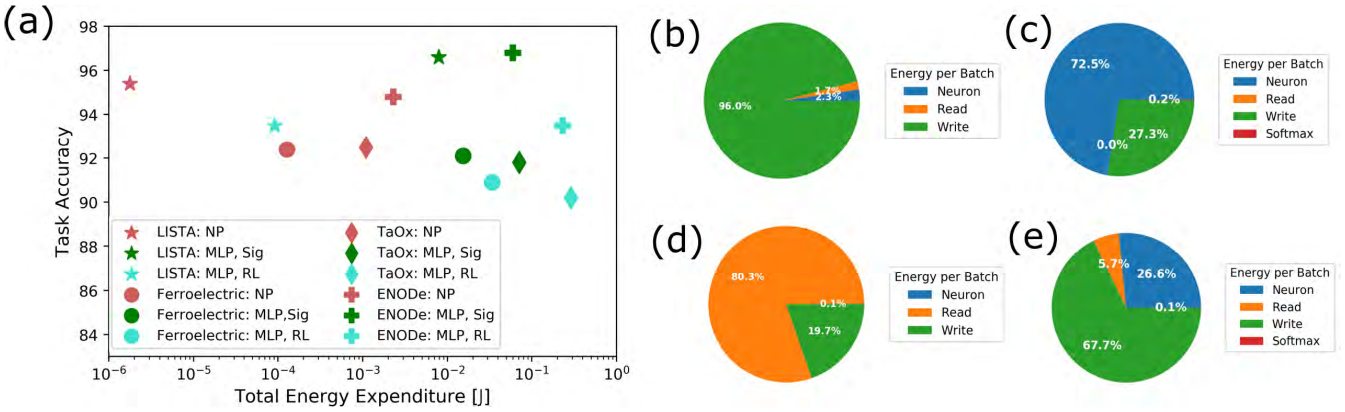

FIGURE 8. (a) Scatter plot showing policy and device accuracy and energy outcomes for all considered families of analog nanosynapses and the two contrasting learning policies (along with contrasting activation choice for MLP); for these estimates, candidate systems have been calibrated with different hidden layer M and number of training examples, according to results shown in Table. 1. Only batch learning estimates are shown. (b)-(e) show single batch energy budgets; (b) and (c) for the ferroelectric memristor, and (d)-(e) for the polymeric ENODe NVM.

F. IMPROVING GENERALIZATION ABILITIES OF MEMRISTIVE NoProp SYSTEMS

Deep networks have achieved better generalization and hence, performance on competitive tasks, in comparison to single-layer (shallow) networks [65]–[67]. Recently, there has been significant interest in building deeper ELM-like systems- or Multiple Hidden-Layer (M-HL) ELM systems that allow for depth and online training, while reducing training iterations as compared to full backpropagation networks. In order to achieve this, such systems require an auto-encoder system [68] at an earlier part of the network to encode representations of the task data more informative than random weights. While typical auto-encoder systems require the equivalent of training labels during pre-training, recent proposals have demonstrated a hierarchical auto-encoder approach to M-HL ELM learning in which unsupervised filter banks, rather than fine-tuned backpropagated gradients, set projections used by the final regression system [69], [70]. This approach bears some similarity to convolutional neural networks (CNNs), and indeed, has been demonstrated to achieve state-of-the-art performance on typical image tasks. We have designed and tested a partially and fully on-chip version of an M-HL ELM learning system which uses a single unsupervised auto-encoder as pre-processing for the later primary systems, as shown in Fig. 9. This architecture is similar to that of deep belief networks (DBNs), however it does not require the use of contrastive divergence.

In our proposed M-HL ELM architecture, a random layer is followed by an auto-encoder layer of dimension M_{ae}, which trains prior to the second system. Subsequently, the training data is pre-processed by the auto-encoder layer and presented to a ELM classifier with hidden layer dimension M_{elm}. In our simulations, weights for projection layers are randomly initialized based on device-to-device variability (OFF-state) for memristive devices as before. Each layer-to-layer connection can be realized using separate crossbars, though a single large layer with random device states can be re-used during the prior and primary training stages.

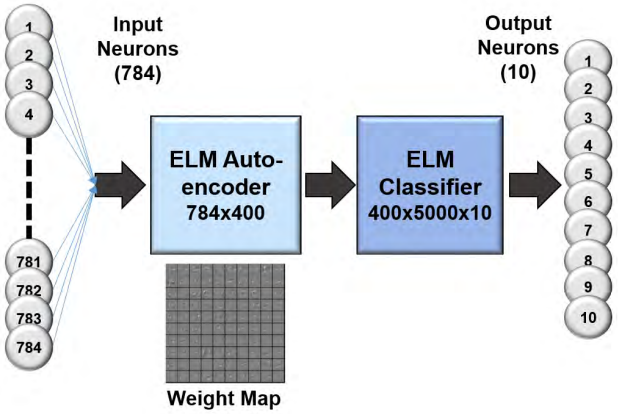

FIGURE 9. This figure demonstrates the M-HL ELM system, which uses a random initial projection, an ELM auto-encoder, and finally, a standard random projection and regression system (equivalent to the SHL-ELM system already detailed). Temporally, the Auto-encoder is trained first, and then serves as pre-processor to the secondary classifier during primary learning.

TABLE 3. Performance of multiple hidden layer (M-HL) memristive systems.

System Name	Configuration	Training Accuracy	Test Accuracy
Software Synapse	$784x400x5000x10$	99.16%	98.70%
Memristive Synapse: Off-Chip[a]	$784x400x5000x10$	98.29%	97.53%
Memristive Synapse: On-Chip[b]	$784x400x5000x10$	97.51%	96.83%
Memristive Synapse: On-Chip[b]	$784x400x3000x10$	96.11%	95.63%

[a]Regularized ridge-regression was used to compute the weights and was subsequently written to devices with $g = 256$ resolution.
[b]150k samples for training; nanosynapses have $g = 256$ resolution

We consider two cases: one in which the devices in the final output (regression) layer are written following collected hidden layer activation (off-chip); a second, in which $s-WH$ policy implements local learning (on-chip). As seen in Table 3, minimal loss in learning performance ($< 1\%$) exists between the software version of the M-HL ELM system, and the

system using off-chip written memristive synapses. In the online scheme, an additional 1% accuracy is lost, considering a large ultimate hidden layer size M, and assuming the nanosynapses in the ultimate layer have a resolution of 8 bits and are linear/quasi-linear. This case is somewhat different than the single hidden layer nanodevice ELM systems, where we were able to show complete convergence between off-chip collected/computed weights and the online approach with s-WH rule. Preliminary analysis suggests this slight loss can be ameliorated by modulating auto-encoder M_{ae} and final layer M_{elm} dimension. For instance, as in Table 3, expanding M_{elm} brings more accuracy; we expect a broader parameter search could show parity. Nonetheless, even at the smaller M_{elm} dimension case shown, we still see a greater than 1% gain in generalization on the test set relative to a simple memristive ELM. While the proposed M-HL ELM system achieves inferior results to alternative memristve CNN proposals, the system we simulated was a proof of concept and atypically used only one unsupervised AE unit; these units can be stacked/cascaded, expanding computational capabilities. Thus, while these results confirm the possibility of deeper memristive NoProp architectures and exceed MLP system performance, comparisons with more elaborate backprop models (e.g., CNN) and DBNs remain a future research task.

IV. DISCUSSION

A. RELATIONSHIP TO STATE-OF-THE-ART NVM GRADIENT LEARNING

A key result is that, unlike NoProp, memristive gradient systems learning with a sign approximate of the gradient (s-WH in both layers as $sign(\delta_k x_{j,k})$), require $g > 256$ or eight bit weight resolution to do so effectively (losing 1% or less compared to software comparison systems). This observation is consistent with the machine learning literature, which has suggested that less than 8 bit weight resolution can significantly impede network generalization and accuracy when gradient updates are binarized [71], [72]. In [37] a parameter analysis shows that $g = 64$ or 6 bits are sufficient for NVM backprop online learning. This is not necessarily an inconsistent result as the updates used to manifest backprop in that case are precisely calibrated, which requires additional overhead on the core to calculate magnitudes of $\delta_k x_{j,k}$ for every row and/or column. Lastly, by separating out the relative synapse depth requirements of the first and second layer, we have also shown that the stringency requirement for g is higher in the earlier layers than it will be in the later ones. A natural extension of this work would be to examine if the trend continues in deeper networks (*i.e.* more than 1 hidden layer) using gradient learning; this could limit the possibility of building deeper gradient learning systems using our proposed energy-efficient update scheme. On the other hand, more complicated and energy expensive choices could be taken.

As in other works, we show MLP gradient learning systems are especially fragile to non-linear weight updates. For instance, the best-performing MLP system (containing fine-tuned analog differentiable neuron) lost nearly 5% when taking this natural effect in many emerging nanosynapses into account. While there are potential mitigation strategies on both the device-level, *e.g.* building more linear synapses [58], and on the circuit level, *e.g.* designing multiple weight-range synapses [73], they may require unacceptably slow system operation or significant additional peripheral circuitry, respectively. Other studies have shown that, when nanodevices are paired with additional CMOS devices such as linear capacitors and access transistors [74], or a non-linear cell-selector [75], a complete mitigation of asymmetric non-linearity issues can be obtained. While the energy and area overhead of these approaches at the systems-level is unclear, it is certain an increasing bit cell size larger than 1R or 1T1R could limit the ultimate density of memory accelerators.

In contrast to proposals for deep neural networks [72], which binarize neuron activations and weights but require highly analog gradient values to be accumulated, our approach binarized batch gradients but always maintains highly analog weights. This contrast highlights that ultra-dense implementations of gradient learning can binarize in most places yet must store a highly analog value somewhere. Our work highlights that doing so within emerging nanosynapses themselves is a good strategy, given that these devices are reasonably linear and symmetric.

The idea of implementing differentiable systems with only scalar derivatives is appealing and has worked well in standard machine learning. However, our analysis has highlighted the difficulty in incorporating this approach in *in-situ* nanosynaptic learning systems, whether binary (ReLu) or quasi-binary (PrReLU) activation nodes are chosen. Other works have suggested memristive MLP systems using binary activation functions should perform better than we showed here, yet these assumed either analog updates [76] or complicated pulse-based learning schemes [52]. In our ultra-low power set-up, we find that rectifier systems require an intermediate layer size between analog MLP and NoProp; this outweighs any energy benefit gained from the simpler neuronal circuitry itself.

B. ADVANTAGES AND CONSTRAINTS OF NVM NoProp LEARNING

The online learning NoProp learning systems we analyzed were only able to converge to software quality results with sufficient writable depth, and perfect (linear) evolution. Yet, unlike gradient-learning alternatives, the required writable depth requirement is gentle, with 6 bits $g = 64$ being ideal but 4-5 bits ($g = 16 - 32$) performing well too. In [77], 4 bits were shown sufficient to implement weights executing simple STDP (temporal correlation) learning, which is a similar operation to s-WH. In addition, a memristive two-layer system using the locally competitive algorithm in the first layer (unsupervised) and supervised learning in the second layer also required between 4-6 bits depending on device considerations [78]. These consistent results could inspire further

research on the potential advantages of random/unsupervised pre-processing layers cooperating with later supervised ones.

We have shown that the NoProp systems were more resilient to learning with devices that update weights non-linearly than their gradient-learning competitors, and that such systems are comparatively resilient as all critical nanodevice parameters (G_{Max}, G_{Min}, writable depth g) become increasingly variable (Fig. 6). Lastly, our energy analysis confirmed that NoProp systems always use less energy to perform *in-situ* learning when considering the same nanosynapse and learning policy, but that accessory circuit considerations as well as learning style (batch v. single-example) can easily have as much of an affect on the final energy fingerprint as this choice.

One of the difficulties involved in realizing NVM-powered NoProp/ELM hardware systems will be to mitigate the significant additional area overhead need to realize larger hidden-layer neuronal arrays. Possible routes to confront this challenge are time-multiplexing, to allow a smaller physical layer to do the work of many more neuronal functions, or further exploring the use of ultra-dense emerging devices as nano-neurons.

V. CONCLUSION

In this work, we have designed an energy-efficient *in-situ* learning rule and system inspired by software-based NoProp/ELM systems, and found that it can successfully compete with imported weight (inference-only) memristive as well as software alternatives. Using the rule, we obtained state-of-the-art results for single hidden-layer memristive ELM: 94.5%, and multiple hidden-layer memristive ELM (M-HL ELM): 96.83% on the MNIST task, the latter of which was competitive with results obtained with a memristive MLP system with similar constraints (96.3%.)

We also found that NoProp/ELM systems learning with this rule obtain special properties. Foremost, they manage to learn with synapses with less than 8 bit quality, while this serves as a necessary minimal requirement for MLP online learning systems learning with the sign-based learning rule. Additionally, we discovered that device non-linearity has a differential impact between NoProp/ELM v. MLP online learning styles. While only perfect/linear devices can fully emulate software results in all cases, NoProp/ELM systems are notably more resilient to systems learning with non-linear synapses. Lastly, we found that device-to-device variability, asymmetry and cycle-to-cycle noise again have a differential impact between the two learning approaches. NoProp/ELM systems seem to be more resilient to several of the effects; however, no system is immune to them.

An energy estimation of online learning in memristive MLP and NoProp systems- taking into the array's core operation costs as well as key peripheral CMOS systems for a few candidate nanosynapses classes- showed that NoProp systems save energy head-to-head versus competitor MLP systems. Total savings varied between 70x-120x, depending on the device class. However, the highest performing system with only one hidden-layer still uses backpropagation of error, and the effect of chosen nanodevice synapse and CMOS neuron designs can easily outweigh learning policy gains.

Overall, our results suggest that the NoProp/ELM learning system we have introduced here may be an unconventional but attractive option for neuromorphic designers. By exploiting a natural device property usually seen as an enemy - device-to-device variability- and locally performing computation based on their projections with an energy-efficient learning policy, we have suggested a new pathway towards low-cost inference and learning in edge computing systems.

ACKNOWLEDGMENT

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

REFERENCES

[1] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Lett.*, vol. 10, no. 4, pp. 1297–1301, 2010.

[2] J. J. Yang, D. B. Strukov, and D. R. Stewart, "Memristive devices for computing," *Nature Nanotechnol.*, vol. 8, no. 1, pp. 13–24, 2013.

[3] S. Saïghi, C. G. Mayr, T. Serrano-Gotarredona, H. Schmidt, G. Lecerf, J. Tomas, J. Grollier, S. Boyn, A. F. Vincent, D. Querlioz, S. La Barbera, F. Alibart, D. Vuillaume, O. Bichler, C. Gamrat, and B. Linares-Barranco, "Plasticity in memristive devices for spiking neural networks," *Frontires Neurosci.*, vol. 9, p. 51, Mar. 2015. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2015.00051

[4] F. Alibart, E. Zamanidoost, and D. B. Strukov, "Pattern classification by memristive crossbar circuits using *ex situ* and *in situ* training," *Nature Commun.*, vol. 4, Jun. 2013, Art. no. 2072.

[5] O. Kavehei, S. Al-Sarawi, K.-R. Cho, and N. Iannella, "Memristor-based synaptic networks and logical operations using *in-situ* computing," in *Proc. IEEE 7th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process. (ISSNIP)*, Dec. 2011, pp. 137–142.

[6] M. Prezioso, F. Merrikh-Bayat, B. D. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, pp. 61–64, May 2015.

[7] S. Park, M. Chu, J. Kim, J. Noh, M. Jeon, B. H. Lee, H. Hwang, B. Lee, and B.-G. Lee, "Electronic system with memristive synapses for pattern recognition," *Sci. Rep.*, vol. 5, May 2015, Art. no. 10123.

[8] Y.-P. Lin, C. H. Bennett, T. Cabaret, D. Vodenicarevic, D. Chabi, D. Querlioz, B. Jousselme, V. Derycke, and J.-O. Klein, "Physical realization of a supervised learning system built with organic memristive synapses," *Sci. Rep.*, vol. 6, Sep. 2016, Art. no. 31932.

[9] G. W. Burr, R. M. Shelby, S. Sidler, C. Nolfo, J. Jang, I. Boybat, R. S. Shenoy, P. Narayanan, K. Virwani, E. U. Giacometti, B. N. Kurdi, and H. Hwang, "Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element," *IEEE Trans. Electron Devices*, vol. 62, no. 11, pp. 3498–3507, Nov. 2015.

[10] C. Li, D. Belkin, Y. Li, P. Yan, M. Hu, N. Ge, H. Jiang, E. Montgomery, P. Lin, Z. Wang, W. Song, J. P. Strachan, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, "Efficient and self-adaptive *in-situ* learning in multilayer memristor neural networks," *Nature Commun.*, vol. 9, no. 1, Jun. 2018, Art. no. 2385.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, p. 533, 1986.

[12] T. D. Sanger, "Optimal unsupervised learning in a single-layer linear feedforward neural network," *Neural Netw.*, vol. 2, no. 6, pp. 459–473, 1989.

[13] R. M. Shelby, G. W. Burr, I. Boybat, and C. di Nolfo, "Non-volatile memory as hardware synapse in neuromorphic computing: A first look at reliability issues," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2015, pp. 6A.1.1–6A.1.6.

[14] P.-Y. Chen, B. Lin, I.-T. Wang, T.-H. Hou, J. Ye, S. Vrudhula, J.-S. Seo, Y. Cao, and S. Yu, "Mitigating effects of non-ideal synaptic device characteristics for on-chip learning," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2015, pp. 194–199.

[15] M. Ueda, Y. Nishitani, Y. Kaneko, and A. Omote, "Back-propagation operation for analog neural network hardware with synapse components having hysteresis characteristics," *PLoS ONE*, vol. 9, no. 11, 2014, Art. no. e112659.

[16] M. Tsodyks, K. Pawelzik, and H. Markram, "Neural networks with dynamic synapses," *Neural Comput.*, vol. 10, no. 4, pp. 821–835, 1998.

[17] H. S. Seung, "Learning in spiking neural networks by reinforcement of stochastic synaptic transmission," *Neuron*, vol. 40, no. 6, pp. 1063–1073, 2003.

[18] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[19] B. Widrow, A. Greenblatt, Y. Kim, and D. Park, "The no-prop algorithm: A new learning algorithm for multilayer neural networks," *Neural Netw.*, vol. 37, pp. 182–188, Jan. 2013.

[20] Y. Wang, H. Yu, L. Ni, G.-B. Huang, M. Yan, C. Weng, W. Yang, and J. Zhao, "An energy-efficient nonvolatile in-memory computing architecture for extreme learning machine by domain-wall nanowire devices," *IEEE Trans. Nanotechnol.*, vol. 14, no. 6, pp. 998–1012, Nov. 2015.

[21] M. Suri and V. Parmar, "Exploiting intrinsic variability of filamentary resistive memory for extreme learning machine architectures," *IEEE Trans. Nanotechnol.*, vol. 14, no. 6, pp. 963–968, Nov. 2015.

[22] M. Suri, V. Parmar, G. Sassine, and F. Alibart, "OXRAM based ELM architecture for multi-class classification applications," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

[23] C. H. Bennett, S. La Barbera, A. F. Vincent, J.-O. Klein, F. Alibart, and D. Querlioz, "Exploiting the short-term to long-term plasticity transition in memristive nanodevice learning architectures," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2016, pp. 947–954.

[24] J. Hasler and B. Marr, "Finding a roadmap to achieve large neuromorphic hardware systems," *Frontiers Neurosci.*, vol. 7, p. 118, Sep. 2013. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2013.00118

[25] Y. Li, S. Kim, X. Sun, P. Solomon, T. Gokmen, H. Tsai, S. Koswatta, Z. Ren, R. Mo, C. C. Yeh, W. Haensch, and E. Leobandung, "Capacitor-based cross-point array for analog neural network with record symmetry and linearity," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2018, pp. 25–26.

[26] Y. V. Pershin and M. Di Ventra, "Practical approach to programmable analog circuits with memristors," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 1857–1864, Aug. 2010.

[27] G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, and L. L. Sanches, "Neuromorphic computing using non-volatile memory," *Adv. Phys. X*, vol. 2, no. 1, pp. 89–124, 2017.

[28] M. Suri, O. Bichler, D. Querlioz, B. Traoré, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Physical aspects of low power synapses based on phase change memory devices," *J. Appl. Phys.*, vol. 112, no. 5, 2012, Art. no. 054904.

[29] Y. Li, Y. P. Zhong, L. Xu, J. J. Zhang, X. H. Xu, H. J. Sun, and X. S. Miao, "Ultrafast synaptic events in a chalcogenide memristor," *Sci. Rep.*, vol. 3, Mar. 2013, Art. no. 1619.

[30] W. Wang, D. Loke, L. Shi, R. Zhao, H. Yang, L.-T. Law, L.-T. Ng, K.-G. Lim, Y.-C. Yeo, T.-C. Chong, and A. L. Lacaita, "Enabling universal memory by overcoming the contradictory speed and stability nature of phase-change materials," *Sci. Rep.*, vol. 2, Apr. 2012, Art. no. 360. doi: 10.1038/srep00360.

[31] F. Alibart, L. Gao, B. D. Hoskins, and D. B. Strukov, "High precision tuning of state for memristive devices by adaptable variation-tolerant algorithm," *Nanotechnology*, vol. 23, no. 7, 2012, Art. no. 075201.

[32] Y. Van de Burgt, E. Lubberman, E. J. Fuller, S. T. Keene, G. C. Faria, S. Agarwal, M. J. Marinella, A. A. Talin, and A. Salleo, "A non-volatile organic electrochemical device as a low-voltage artificial synapse for neuromorphic computing," *Nature Mater.*, vol. 16, no. 4, pp. 414–418, Apr. 2017.

[33] A. Chanthbouala, V. Garcia, R. O. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, S. Xavier, H. Yamada, C. Deranlot, N. D. Mathur, M. Bibes, A. Barthélémy, and J. Grollier, "A ferroelectric memristor," *Nature Mater.*, vol. 11, pp. 860–864, Sep. 2012.

[34] D. Querlioz, P. Dollfus, O. Bichler, and C. Gamrat, "Learning with memristive devices: How should we model their behavior?" in *Proc. IEEE/ACM Int. Symp. Nanoscale Archit.*, Jun. 2011, pp. 150–156.

[35] J. Woo, K. Moon, J. Song, S. Lee, M. Kwak, J. Park, and H. Hwang, "Improved synaptic behavior under identical pulses using AlO_x/HfO_2 Bilayer RRAM array for neuromorphic systems," *IEEE Electron Device Lett.*, vol. 37, no. 8, pp. 994–997, Aug. 2016.

[36] X. Sun, P. Wang, K. Ni, S. Datta, and S. Yu, "Exploiting hybrid precision for training and inference: A 2T-1FeFET based analog synaptic weight cell," in *IEDM Tech. Dig.*, Dec. 2018, pp. 1–3.

[37] S. Yu, "Neuro-inspired computing with emerging nonvolatile memorys," *Proc. IEEE*, vol. 106, no. 2, pp. 260–285, Feb. 2018.

[38] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[39] L. Gao, F. Alibart, and D. B. Strukov, "Analog-input analog-weight dot-product operation with Ag/a-Si/Pt memristive devices," in *Proc. IEEE/IFIP 20th Int. Conf. VLSI Syst.-Chip (VLSI-SoC)*, Oct. 2012, pp. 88–93.

[40] D. Chabi, W. Zhao, D. Querlioz, and J.-O. Klein, "On-chip universal supervised learning methods for neuro-inspired block of memristive nanodevices," *ACM J. Emerg. Technol. Comput. Syst.*, vol. 11, no. 4, 2015, Art. no. 34.

[41] A. van Schaik and J. Tapson, "Online and adaptive pseudoinverse solutions for ELM weights," *Neurocomputing*, vol. 149, pp. 233–238, Feb. 2015.

[42] C. S. Thakur, R. Wang, S. Afshar, G. Cohen, T. J. Hamilton, J. Tapson, and A. van Schaik, "An online learning algorithm for neuromorphic hardware implementation," 2015, *arXiv:1505.02495*. [Online]. Available: https://arxiv.org/abs/1505.02495

[43] M. Joost and W. Schiffmann, "Speeding up backpropagation algorithms by using cross-entropy combined with pattern normalization," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 6, no. 02, pp. 117–126, 1998.

[44] R. Zunino and P. Gastaldo, "Analog implementation of the SoftMax function," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, vol. 2, May 2002, pp. II–II.

[45] I. M. Elfadel and J. L. Wyatt, Jr, "The 'softmax' nonlinearity: Derivation using statistical mechanics and useful properties as a multiterminal analog circuit element," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, pp. 882–887.

[46] J. Shamsi, A. Amirsoleimani, S. Mirzakuchaki, A. Ahmade, S. Alirezaee, and M. Ahmadi, "Hyperbolic tangent passive resistive-type neuron," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 581–584.

[47] G. Khodabandehloo, M. Mirhassani, and M. Ahmadi, "Analog implementation of a novel resistive-type sigmoidal neuron," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 750–754, Apr. 2012.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

[49] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, and S. Kvatinsky, "Memristor-based multilayer neural networks with online gradient descent training," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2408–2421, Oct. 2015.

[50] I. Kataeva, F. Merrikh-Bayat, E. Zamanidoost, and D. Strukov, "Efficient training algorithms for neural networks based on memristive crossbar circuits," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.

[51] M. Abadi *et al.* (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: http://tensorflow.org/

[52] D. Negrov, I. Karandashev, V. Shakirov, Y. Matveyev, W. Dunin-Barkowski, and A. Zenkevich, "An approximate backpropagation learning rule for memristor based neural networks using synaptic plasticity," *Neurocomputing*, vol. 237, pp. 193–199, May 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231216313042

[53] Q. Zhang, H. Wu, P. Yao, W. Zhang, B. Gao, N. Deng, and H. Qian, "Sign backpropagation: An on-chip learning algorithm for analog RRAM neuromorphic computing systems," *Neural Netw.*, vol. 108, pp. 217–223, Dec. 2018.
[54] S. Stathopoulos, A. Khiat, M. Trapatseli, S. Cortese, A. Serb, I. Valov, and T. Prodromakis, "Multibit memory operation of metal-oxide bi-layer memristors," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 17532.
[55] H. Jiang, L. Han, P. Lin, Z. Wang, M. H. Jang, Q. Wu, M. Barnell, J. J. Yang, H. L. Xin, and Q. Xia, "Sub-10 nm Ta channel responsible for superior performance of a HfO_2 memristor," *Sci. Rep.*, vol. 6, Jun. 2016, Art. no. 28525.
[56] S. Sidler, I. Boybat, R. M. Shelby, P. Narayanan, J. Jang, A. Fumarola, K. Moon, Y. Leblebici, H. Hwang, and G. W. Burr, "Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: Impact of conductance response," in *Proc. IEEE 46th Eur. Solid-State Device Res. Conf. (ESSDERC)*, Sep. 2016, pp. 440–443.
[57] S. Agarwal, T.-T. Quach, O. Parekh, A. H. Hsia, E. P. DeBenedictis, C. D. James, M. J. Marinella, and J. B. Aimone, "Energy scaling advantages of resistive memory crossbar based computation and its application to sparse coding," *Frontiers Neurosci.*, vol. 9, p. 484, Jan. 2015.
[58] E. J. Fuller, F. El Gabaly, F. Léonard, S. Agarwal, S. J. Plimpton, R. B. Jacobs-Gedrim, C. D. James, M. J. Marinella, and A. A. Talin, "Li-ion synaptic transistor for low power analog computing," *Adv. Mater.*, vol. 29, Jan. 2016, Art. no. 1604310.
[59] J. P. Strachan, A. C. Torrezan, G. Medeiros-Ribeiro, and R. S. Williams, "Measuring the switching dynamics and energy efficiency of tantalum oxide memristors," *Nanotechnology*, vol. 22, no. 50, 2011, Art. no. 505402.
[60] S. H. Jo, T. Kumar, S. Narayanan, W. D. Lu, and H. Nazarian, "3D-stackable crossbar resistive memory based on Field Assisted Superlinear Threshold (FAST) selector," in *IEDM Tech. Dig.*, Dec. 2014, pp. 6–7.
[61] G. W. Burr, R. S. Shenoyc, K. Virwani, P. Narayanan, A. Padillad, and B. Kurdi, "Access devices for 3D crosspoint memory," *J. Vac. Sci. Technol. B, Microelectron.*, vol. 32, no. 4, 2014, Art. no. 040802.
[62] D. Chabi, Z. Wang, C. Bennett, J.-O. Klein, and W. Zhao, "Ultrahigh density memristor neural crossbar for on-chip supervised learning," *IEEE Trans. Nanotechnol.*, vol. 14, no. 6, pp. 954–962, Nov. 2015.
[63] V. Parmar and M. Suri, "Design exploration of IoT centric neural inference accelerators," in *Proc. ACM Great Lakes Symp. VLSI*, 2018, pp. 391–396.
[64] M. J. Marinella, S. Agarwal, A. Hsia, I. Richter, R. Jacobs-Gedrim, J. Niroula, S. J. Plimpton, E. Ipek, and C. D. James, "Multiscale co-design analysis of energy, latency, area, and accuracy of a ReRAM analog neural training accelerator," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 1, pp. 86–101, Mar. 2018.
[65] R. Salakhutdinov and G. Hinton, "Deep Boltzmann machines," in *Proc. Mach. Learn. Res.*, Clearwater Beach, FL, USA, 2009, pp. 448–455. [Online]. Available: http://proceedings.mlr.press/v5/
[66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2015, pp. 1–9.
[67] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[68] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, Dec. 2010.
[69] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 809–821, Apr. 2015.
[70] W. Zhu, J. Miao, L. Qing, and G.-B. Huang, "Hierarchical extreme learning machine for unsupervised representation learning," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2015, pp. 1–8.
[71] V. Vanhoucke, A. Senior, and M. Z. Mao, "Improving the speed of neural networks on CPUs," in *Proc. Deep Learn. Unsupervised Feature Learn. NIPS Workshop*, vol. 1, 2011, p. 4.
[72] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or −1," 2016, *arXiv:1602.02830*. [Online]. Available: https://arxiv.org/abs/1602.02830
[73] S. Agarwal, R. B. J. Gedrim, A. H. Hsia, D. R. Hughart, E. J. Fuller, A. A. Talin, C. D. James, S. J. Plimpton, and M. J. Marinella, "Achieving ideal accuracies in analog neuromorphic computing using periodic carry," in *Proc. IEEE Symp. VLSI Technol.*, Jun. 2017, pp. T174–T175.
[74] S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat, C. Nolfo, S. Sidler, M. Giordano, M. Bodini, N. C. P. Farinha, B. Killeen, C. Cheng, Y. Jaoudi, and G. W. Burr, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, pp. 60–67, 2018.
[75] S. T. Keene, A. Melianas, E. J. Fuller, Y. van de Burgt, A. A. Talin, and A. Salleo, "Optimized pulsed write schemes improve linearity and write speed for low-power organic neuromorphic devices," *J. Phys. D, Appl. Phys.*, vol. 51, no. 22, 2018, Art. no. 224002.
[76] C. Liu, Q. Yang, C. Zhang, H. Jiang, Q. Wu, and H. H. Li, "A memristor-based neuromorphic engine with a current sensing scheme for artificial neural network applications," in *Proc. IEEE 22nd Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2017, pp. 647–652.
[77] T. Pfeil, T. Potjans, S. Schrader, W. Potjans, J. Schemmel, M. Diesmann, and K. Meier, "Is a 4-bit synaptic weight resolution enough?—Constraints on enabling spike-timing dependent plasticity in neuromorphic hardware," *Frontiers Neurosci.*, vol. 6, p. 90, Jul. 2012. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2012.00090
[78] W. Woods, J. Bürger, and C. Teuscher, "Synaptic weight states in a locally competitive algorithm for neuromorphic memristive hardware," *IEEE Trans. Nanotechnol.*, vol. 14, no. 6, pp. 945–953, Nov. 2015.

CHRISTOPHER H. BENNETT (M'14) received the B.Sc./M.Sc. degree from Stanford University, in 2011, a joint M.S. degree from the Erasmus Mundus Masters in Nanoscience and Nanotechnology (EMM-NANO), host schools K.U. Leuven, Belgium, and the Chalmers University of Technology, Sweden, in 2014, and the Ph.D. degree from the Centre de Nanosciences et Nanotechnologies (C2N), Nanoarchitectures team, Université Paris-Sud/Université Paris-Saclay, for his research, in 2018. During this thesis, he designed and built hardware learning systems with memory nanodevices, in particular, a new class of highly analog polymeric nanodevices, and modeled new nanodevice architectures. He is currently with Sandia National Laboratories in the research on ReRAM accelerators.

VIVEK PARMAR received the M.Tech. degree in electrical engineering from the IIT Delhi, India, in 2016, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include computer architecture, neuromorphic computing, embedded systems, VLSI design, and machine learning.

LAURIE E. CALVET (M'97–SM'18) received the B.S. degree in applied physics from Columbia University, New York, NY, USA, in 1995, and the Ph.D. degree from Yale University, New Haven, CT, USA, in 2001. In 2007, she joined the French Centre Nationale la Recherche Scientifique (CNRS), Paris, France, and carries out research at the Center for Nanoscience and Nanotechnology, Université Paris-Sud, Orsay, France. Her current research interests include exploring the use of nano-devices and materials to realize new technologies, and understanding the physics of such devices and their use in realizing new computing paradigms.

JACQUES-OLIVIER KLEIN (M'90) received the Ph.D. and Habilitation degrees in electronic engineering from the University of Paris-Sud, Orsay, France, in 1995 and 2009, respectively, where he is currently a Professor with the Center for Nanoelectronics and Nanotechnology. He has authored or coauthored over 80 publications and 70 conference papers. His current research interests focus on architecture of circuits and systems based on emerging nanocomponents in the field of nano-magnetism, and bio-inspired nanonoelectronics.

MANAN SURI (M'08) received the bachelor's and master's degrees from Cornell University, USA, in 2009, and the Ph.D. degree from INP-Grenoble, France, in 2013. He has also served as an Advisor/Steering Committee Member of some leading AI/neuromorphic and NVM hardware companies. He was with NXP Semiconductors, Belgium, and CEA-LETI, France. He is currently an Assistant Professor with the Department of Electrical Engineering, IIT Delhi. He is also a Visiting Scientist with CNRS, France. He has filed several patents, authored over 50 publications, and delivered over 30 invited talks. His research interest includes semiconductor non-volatile memory (NVM) technology and its advanced applications (neuromorphic, AI, security, computing, and sensing). He has been globally recognized as a leading DeepTech Innovator. He was selected by MIT Technology Review as one of the Top 35 Global Innovators Under 35 (MIT-TR 35 Global List) and as one of the Top 10 Indian Innovators Under 35 (MIT-TR 35 India List). He received the prestigious IEEE EDS Early Career Award, in 2018, the Young Scientist Award from the National Academy of Sciences, in 2017, the Young Engineers Award from the Institution of Engineers, in 2016, and the Laureat du Prix from the French Nanosciences Foundation, in 2014.

MATTHEW J. MARINELLA (M'04–SM'17) received the Ph.D. degree in electrical engineering from Arizona State University under Dieter K. Schroder, in 2008. He is currently a Principal Member of the Technical Staff with Sandia National Labs. He is also a Principal Investigator for Sandia's Nonvolatile Memory Program and leads research projects on neuromorphic, radiation hard, and energy efficient computing. He Chairs the Emerging Memory Devices Section for the IRDS Roadmap Beyond CMOS Chapter, serves on various technical program committees.

DAMIEN QUERLIOZ (M'08) received the Predoctoral degree from the Ecole Normale Supérieure, Paris, and the Ph.D. degree from the Université Paris-Sud, in 2008. After postdoctoral appointments at Stanford University and CEA, he became a Permanent Researcher with the Centre for Nanoscience and Nanotechnology, Université Paris-Sud, where he is currently a CNRS Research Scientist. He focuses on novel usages of emerging non-volatile memory, in particular, relying on inspirations from biology and machine learning. He coordinates the INTEGNANO interdisciplinary research group. He received the European Research Council Starting Grant to develop the concept of natively intelligent memory, in 2016 and the CNRS Bronze medal, in 2017.