Building Next-generation AI systems: Co-optimization of Algorithms, Architectures, and Nanoscale Memristive Devices

Bipin Rajendran*, Abu Sebastian†, and Evangelos Eleftheriou†
Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, 07102, USA
†IBM Research – Zurich, 8803 Rüschlikon, Switzerland
Email: bipin@njit.edu

***Abstract*—Computing systems inspired by the architecture of the human brain is poised to revolutionize the engines for information processing and data analytics. However, the efficiency and performance of these platforms pale in comparison with the human brain, especially when benchmarked in terms of metrics such as intelligence per Watt per square mm. In this paper, we review some recent progress and future prospects of building artificial intelligence systems that target the efficiency of the brain, leveraging the unique properties of nanoscale memristive device technologies.**

***Index Terms*—Spiking neural network, memristive devices, crossbar array, in-memory computing, on-chip learning**

I. Introduction

The past decade has seen significant advances in the ability of algorithms powered by deep learning techniques to execute complex cognitive tasks, often rivaling human performance [1], [2]. These learning models employ large artificial neural networks, with several layers of local compute units (neurons) that interact with each other through dedicated connections with adjustable weights (synaptic memory). Since each compute node connects to hundreds or thousands of other nodes, the number of synaptic parameters far exceeds the number of neuronal parameters. Neurons in adjacent layers in these networks can have all-to-all, convolutional, or recurrent connectivity and can be employed for supervised, unsupervised, and reinforcement learning tasks.

The optimization of these networks for supervised learning proceeds in three phases (Fig. 1): (i) During the forward pass, the neuronal activation values of a layer are propagated to the next layer through the synaptic weights, which is mathematically equivalent to the multiplication of the vector of neuronal activations with the connectivity matrix representing the synaptic conductances, in a layer-by-layer fashion. At the final layer, the generated output can then be compared with the desired network response and an error signal can be calculated. (ii) During the back-propagation phase, the error from a layer is propagated back to the previous layer through the same synaptic weights. This is mathematically equivalent to the multiplication of the vector of errors with the transpose of the connectivity matrix representing the synaptic conductances. Finally, in step (iii), the synaptic weight w_{ij} between two neurons i and j in adjacent layers is updated by an amount, $\Delta w_{ij} = \eta x_i \delta_j$, where η is an appropriately chosen learning rate, x_i is the activation of the of neuron i in the pre-synaptic layer, and δ_j is the back-propagated error of the neuron j in the post-synaptic layer. While the above description of the back-propagation algorithm is based on fully-connected networks, the implementation is conceptually similar and can be extended to convolutional networks and recurrent networks in a straightforward manner.

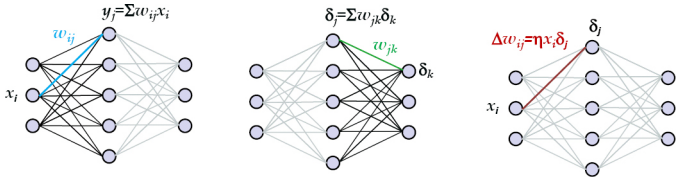

Fig. 1: The three steps used in supervised learning in deep learning networks based on the back-propagation algorithm.

The reasons why today's computational systems based on the traditional von Neumann architecture are ill-suited for implementing large multi-layered deep networks should be now evident. The real-valued signals that propagate through the network have to be represented using floating point or fixed point numbers, unlike the binary action potentials (spikes) that are used exclusively in the brain for communication. Further, the synaptic parameters have to be fetched from memory to the processor at each of these steps and stored back after weight update, which significantly impacts overall performance due to the limitations imposed by the von Neumann bottleneck.

This analysis also reveals the potential solutions to the problem. First, instead of memory-less real-valued neuronal models used in deep learning, third generation Spiking Neural Networks (SNNs) that mimic the asynchronous signaling and information processing capabilities of the brain can significantly reduce the communication requirements of the network, provided efficient event-driven learning algorithms can be devised [3]. This is a major challenge, as the traditional methods of back-propagation which are based on gradient descent of continuous-valued cost functions in deep networks do not carry over to spiking networks whose dynamics involve discontinuities due to the abrupt reset in membrane potential during spike events. Moreover, it is also crucial that new

978-1-7281-0981-7/19/$31.00 ©2019 IEEE

learning algorithms that are developed for artificial SNNs mimic the low spike probabilities and sparse signaling nature of their biological counterparts. We will discuss some recent efforts in this direction in section II.

The second avenue for improvement is at the device level. Digital CMOS circuits, as well as nanoscale emerging devices, are optimized to operate at relatively large currents and voltages (typically exceeding $10\,\mu$A and 1 V), at switching time scales below $1-100$ ns. In comparison, neurons and synapses in the brain compute using nano-ampere scale currents and spike signals of 100 mV amplitude, which are communicated to 1000s of other nodes in the network, at slow signaling rates ($10-100$ Hz). No semiconductor device has managed to demonstrate reliable operating characteristics at these power/energy budgets and is hence an important research direction to build systems approaching the efficiency of the brain. We will discuss some of the features of nanoscale emerging memories that are leading candidates for implementing neuronal and synaptic dynamics, and the ongoing efforts in improving their efficiency in section III.

In addition to algorithms and devices, system-level architectures that efficiently implement the required functions also need to be devised. In-memory computing architectures based on cross-bar arrays with programmable analog memory devices is promising to realize large learning networks. With this architecture, vector-matrix multiplication operations could be executed in place and in parallel, leveraging Kirchhoff's laws. This would avoid the need to constantly shuttle synaptic data stored in memory to the processor units, resulting in significant improvements in performance and energy efficiency. However, in order to reap the associated benefits of efficient algorithms and devices, the underlying system architecture should be developed and optimized based on the limitations imposed by the building blocks. We discuss some recent advances in system architectures for on-chip learning and inference using nanoscale memories in section IV, before concluding the paper by laying out the future outlook for the field in section V.

II. Spike based learning algorithms

While there is a large body of research aimed at improving the capabilities of deep learning algorithms [1], we focus here on recent developments on the algorithms for spiking networks due to the potential advantages of event-triggered learning. Despite the great promise of SNNs in terms of efficient implementation, rich dynamics and learning capabilities, it has been unclear how to effectively train large deep networks of spiking neurons to reach the accuracy of Deep Neural Networks (DNNs) for common machine learning tasks. The various SNN training or learning approaches that have been developed over the past few years can be classified into five main categories, as described in [4]. The simplest method is the so-called binarization of ANNs, where standard deep ANNs are trained with binary activations maintaining their synchronous mode of information processing [5]. The second category relies on converting fully trained ANNs using traditional backpropagation algorithms into SNNs, i.e., the analog neurons are converted into spiking neurons [6]. This conversion has been traditionally based on rate codes, but recently researchers have investigated the use of temporal coding including rank order coding. A third approach is based on the notion of constrain-then-train developed in [7], i.e., prior to conversion, conventional ANN training rules (such as backpropagation) are used, taking into account the constraints arising from the spiking neuron models. For example, one such constraint is the need to transform the spiking neuronal dynamics into a differentiable form and then apply backpropagation. Another category is based on supervised learning with spikes, in which gradient descent approaches are employed on cost-functions written in terms of spike-rate or spike times, and learning rules derived using approximate dynamics of spiking neurons [8] or probabilistic formulations [9]. Finally, the last category includes the use of biologically-inspired unsupervised Hebbian learning rules, such as Spike-Timing-Dependent Plasticity (STDP) for learning algorithms [10]. Recently, it was shown that recurrent SNNs with adapting neurons can achieve classification performance comparable to state-of-the-art LSTM networks by using backpropagation through time (BPTT) [11]. So to summarize, even though SNNs have not yet found widespread acceptance for machine learning applications, recent developments in building the fundamental algorithms and training approaches have buoyed the hope for their adoption for many applications, and especially those that have stringent memory and power constraints.

III. Nanoscale Memristive devices

Significant research efforts have been directed at developing post-CMOS nanoscale non-volatile memory (NVM) devices, targeting the replacement of flash memory and DRAM in the traditional compute stack as well as for certain compute applications [12]. These devices have a two-terminal structure, with an active material (usually a dielectric) with certain specific properties sandwiched between two metal electrodes. Many of these devices, based on the characteristics of the applied programming pulses, can also have stable internal states corresponding to intermediary values, in between the standard **0** and **1** states, which offers an excellent medium to store the synaptic weights of a network (See [13] for a review). The most prominent examples of devices with the above mentioned memristive programmability are phase change memory (PCM) [14], and resistive random access memories (RRAM) [15]. Spin-transfer torque RAM (STTRAM) devices, though traditionally used as binary storage devices, have recently been optimized as stochastic or memristive switches, as well as microwave oscillators, which provides an alternate computational element for learning systems [16].

These nanoscale devices offer the ability to store synaptic conductance values in a compact form factor - typically $20\,\mathrm{F}^2$, but potentially as low as $4\,\mathrm{F}^2$, where F is the smallest patternable feature size in a lithography node; note that SRAM cell size typically exceed $150\,\mathrm{F}^2$. However, the conductance modulation characteristics of these devices are non-linear, stochastic, and asymmetric, which introduces several challenges in

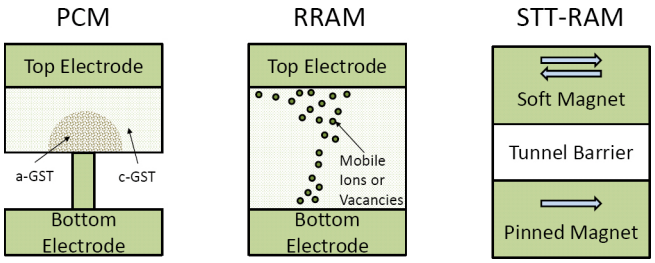

Fig. 2: Schematic of PCM, RRAM, and STTRAM devices.

using them for learning applications [17]. Furthermore, several reliability issues are common to all these devices, including drift in the programmed conductance state, read disturb, and device-to-device variability. Hence, using these devices to implement synaptic weights without architectural or algorithmic optimization leads to drastic reductions in accuracy compared to what is attainable in software. For instance, as demonstrated in [18], the accuracy of a 3-layered artificial neural network implemented using PCM synapses drops to about 83% for the MNIST hand-written digit recognition problem, compared to the software baseline of 97%.

In addition to using the conductance of memristive devices to represent synaptic weights, it is also possible to leverage other internal device operating mechanisms to implement more complex neuronal and synaptic functions. We highlight two approaches here: (a) By the careful design of programming waveforms, inspired, for instance, by the shape of the action potential observed in biology, it is possible to implement more complex learning rules such as spike-timing-dependent plasticity in these devices, where the conductivity of the device changes in a natural fashion as a function of the time of spikes of the pre- and post-synaptic neurons [19]; and (b) it has been recently demonstrated that the dynamical behavior of spiking neurons can be mimicked leveraging the accumulative behavior of conductance change in PCM [20] and metal-insulator transitions in phase-transition oxides [21].

While most of the above mentioned emerging memory technologies operate above $1\,\mathrm{V}$ and $100\,\mu\mathrm{A}$, there are also several proof-of-concept experiments that aim to build devices at lower energy and power budgets. For instance, memristive devices based on the movement of ionic species such as Cu^{2+} have been demonstrated to exhibit quantized conductance states at room temperature and below $300\,\mathrm{mV}$, and could be used for synaptic learning [22]. Similarly, nanoscale two-dimensional materials also hold great promise for achieving sub-femtojoule energy operations as demonstrated in [23], although these scaled devices require further optimization for meeting other required specifications on reliability such as endurance and retention.

IV. System Architecture

A fundamental aspect of learning networks is the large fan-out of the compute nodes - each neuron transmits its output to more than 100 other neurons in most networks; fan-outs of 1000 is not uncommon. Hence, the hardware architecture needs to support mechanisms for large and flexible high fan-out connectivity between the neurons in the different layers. A tiled array of cross-bars is an excellent approach to achieve this high fanout connectivity; a digital CMOS implementation of this scheme was used in the TrueNorth chip from IBM [24]. Computational memories which use emerging memories at the cross-point can also be used to execute the various steps needed for network emulation such as neuronal communication, backpropagation, and weight update, although such architectures can also be used for other applications such as solving systems of linear equations using Kirchhoff's laws [25]. The peripheral real-valued signals for deep networks can be implemented as stochastic pulse streams for communication and weight-update [26], while signals for communication in SNNs can be binary pulses that read the cells in the array. Local computation within the core can be hence analog or digital, while communications between the core are based on digital routing networks with packets containing information about spikes or other neuronal parameters.

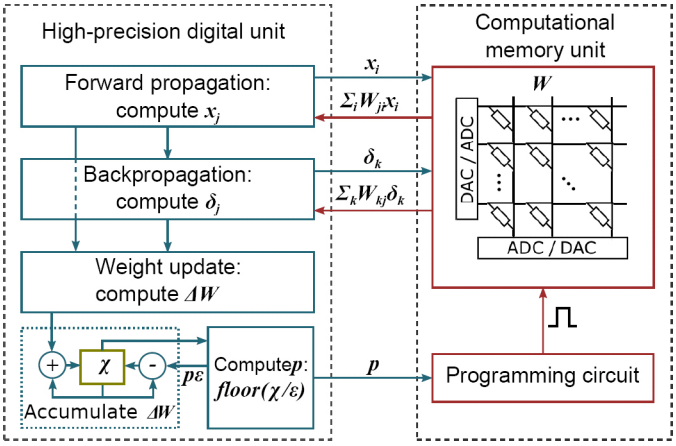

Fig. 3: Mixed-precision architecture proposed to meet the weight-update requirements of learning algorithms using low-precision nanoscale devices, adapted from [27].

However, one of the crucial aspects that the architecture should support is the mapping of synaptic weights and conductance modulations required by the algorithm faithfully into the nanoscale devices in the cross-bar. This is a significant challenge: for instance, most deep learning algorithms require $\Delta w/w < 10^{-3}$ (i.e., 10-bit resolution for the weight update), while most nanoscale devices only have a bit capacity of $3-5$ bits. The recently proposed mixed precision architecture addresses this issue by using a high precision digital memory block to accumulate the small weight-updates, and transferring the accumulated value to the nanoscale device only when it exceeds the update granularity [27]. The matrix-vector multiplications needed for neuronal communication and error back-propagation is implemented using nanoscale cross-bars in a parallel fashion; it has been projected that this architecture can mitigate the issues due to the non-ideal behaviors of nanoscale devices and deliver software-equivalent performance at higher efficiency (Fig.3). The recently demonstrated analog

memory architecture which uses a 3 Transistor - 1 Capacitor cell to accumulate the required updates before transferring it to a memristive device is another noteworthy example analogous to the digital approach described above [28].

V. Future Outlook

Most of the efforts today for building neuromorphic hardware are based on digital CMOS technologies - IBM's TrueNorth [24], Intel's Loihi [29], and Google's Tensor Processing Unit [30] are notable examples. These chips, fabricated at advanced technology nodes, can achieve between $10^{10}-10^{12}$ operations per second per Watt. Though impressive for Silicon, it is approximately three orders of magnitude below the estimates for the performance of the human brain.

We described three high-level research directions that are being pursued today to bridge this gap. These research efforts have seen a convergence of ideas from neuroscience, nanotechnology, and computer architecture, and the joint co-optimization of algorithms, architectures, and nanoscale devices which efficiently enable the parallel computation of complex functions have buoyed the hopes for realizing large-scale AI systems that approach the efficiency of the brain in the not too distant future.

Acknowledgment

B.R acknowledges support from National Science Foundation Award 1710009, Semiconductor Research Corporation and Cisco.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436 – 444, 2015.

[2] W. Xiong, L. Wu, F. Alleva, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5934–5938.

[3] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural networks*, vol. 10, no. 9, pp. 1659–1671, 1997.

[4] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers in Neuroscience*, vol. 12, p. 774, 2018.

[5] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems*, 2016.

[6] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, "Going deeper in spiking neural networks: VGG and residual architectures," *Frontiers in Neuroscience,*, vol. 13, no. 9, 2019.

[7] S. K. Esser, R. Appuswamy, P. A. Merolla, J. V. Arthur, and D. S. Modha, "Backpropagation for energy-efficient neuromorphic computing," in *Neural Information Processing Systems*, ser. NIPS'15, 2015.

[8] N. Anwani and B. Rajendran, "NormAD - normalized approximate descent based supervised learning rule for spiking neurons," in *International Joint Conference on Neural Networks*, 2015.

[9] A. Bagheri, O. Simeone, and B. Rajendran, "Training Probabilistic Spiking Neural Networks with First-to-spike Decoding," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018.

[10] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S. J. Thorpe, and T. Masquelier, "Combining STDP and reward-modulated STDP in deep convolutional spiking neural networks for digit recognition," *CoRR*, vol. abs/1804.00227, 2018.

[11] G. Bellec, D. Salaj, A. Subramoney, R. A. Legenstein, and W. Maass, "Long short-term memory and learning-to-learn in networks of spiking neurons," in *Neural Information Processing Systems (NIPS)*, 2018.

[12] B. Rajendran, R. Cheek, L. Lastras, M. Franceschini, M. Breitwisch, A. Schrott, J. Li, R. Montoye, L. Chang, and C. Lam, "Demonstration of CAM and TCAM using phase change devices," in *IEEE International Memory Workshop*, 2011.

[13] B. Rajendran and F. Alibart, "Neuromorphic computing based on emerging memory technologies," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. PP, no. 99, pp. 1–14, 2016.

[14] I. Boybat, M. Le Gallo, S. Nandakumar, T. Moraitis, T. Parnell, T. Tuma, B. Rajendran, Y. Leblebici, A. Sebastian, and E. Eleftheriou, "Neuromorphic computing with multi-memristive synapses," *Nature communications*, vol. 9, no. 1, p. 2514, 2018.

[15] N. Panwar, D. Kumar, N. Upadhyay, P. Arya, U. Ganguly, and B. Rajendran, "Memristive synaptic plasticity in $Pr_{0.7}Ca_{0.3}MnO_3$ RRAM by bio-mimetic programming," in *Device Research Conference (DRC), 2014 72nd Annual*, June 2014.

[16] M. Romera, P. Talatchian, S. Tsunegi, F. Abreu Araujo, V. Cros, P. Bortolotti, J. Trastoy, K. Yakushiji, A. Fukushima, H. Kubota, S. Yuasa, M. Ernoult, D. Vodenicarevic, T. Hirtzlin, N. Locatelli, D. Querlioz, and J. Grollier, "Vowel recognition with four coupled spin-torque nano-oscillators," *Nature*, vol. 563, no. 7730, pp. 230–234, 2018.

[17] S. R. Nandakumar, M. Le Gallo, I. Boybat, B. Rajendran, A. Sebastian, and E. Eleftheriou, "A phase-change memory model for neuromorphic computing," *Journal of Applied Physics*, vol. 124, 2018.

[18] G. Burr, R. Shelby, C. di Nolfo, J. Jang, R. Shenoy, P. Narayanan, K. Virwani, E. Giacometti, B. 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," in *IEEE International Electron Devices Meeting*, 2014.

[19] N. Panwar, B. Rajendran, and U. Ganguly, "Arbitrary spike time dependent plasticity (STDP) in memristor by analog waveform engineering," *IEEE Electron Device Letters*, vol. 38, no. 6, pp. 740–743, June 2017.

[20] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, and E. Eleftheriou, "Stochastic phase-change neurons," *Nature Nanotechnology*, vol. 11, pp. 693–699, 2016.

[21] M. Jerry, W. y. Tsai, B. Xie, X. Li, V. Narayanan, A. Raychowdhury, and S. Datta, "Phase transition oxide neuron for spiking neural networks," in *Device Research Conference (DRC)*, June 2016.

[22] S. R. Nandakumar, M. Minvielle, S. Nagar, C. Dubourdieu, and B. Rajendran, "A 250 mV $Cu/SiO_2/W$ memristor with half-integer quantum conductance states," *Nano Letters*, vol. 16, no. 3, 2016.

[23] H. Zhao, Z. Dong, H. Tian, D. DiMarzi, M.-G. Han, L. Zhang, X. Yan, F. Liu, L. Shen, S.-J. Han, S. Cronin, W. Wu, J. Tice, J. Guo, and H. Wang, "Atomically thin femtojoule memristive device," *Advanced Materials*, vol. 29, no. 47, p. 1703232, 2017.

[24] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, B. Brezzo, I. Vo, S. K. Esser, R. Appuswamy, B. Taba, A. Amir, M. D. Flickner, W. P. Risk, R. Manohar, and D. S. Modha, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[25] M. L. Gallo, A. Sebastian, R. Mathis, M. Manica, T. Tuma, C. Bekas, A. Curioni, and E. Eleftheriou, "Mixed-precision in-memory computing," *Nature Electronics*, pp. 246–253, 2018.

[26] T. Gokmen and Y. Vlasov, "Acceleration of deep neural network training with resistive cross-point devices: Design considerations," *Frontiers in Neuroscience*, vol. 10, p. 333, 2016.

[27] S. R. Nandakumar, M. L. Gallo, I. Boybat, B. Rajendran, A. Sebastian, and E. Eleftheriou, "Mixed-precision architecture based on computational memory for training deep neural networks," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2018.

[28] S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat, C. di 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.

[29] M. Davies, N. Srinivasa, T. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, Y. Liao, C. Lin, A. Lines, R. Liu, D. Mathaikutty, S. McCoy, A. Paul, J. Tse, G. Venkataramanan, Y. Weng, A. Wild, Y. Yang, and H. Wang, "Loihi: A neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, January 2018.

[30] N. P. Jouppi, C. Young, N. Patil, and D. Patterson, "A domain-specific architecture for deep neural networks," *Commun. ACM*, vol. 61, no. 9, pp. 50–59, Aug. 2018.