Neuromorphic Architectures with Electronic Synapses

Sukru Burc Eryilmaz[1], Siddharth Joshi[2], Emre Neftci[3], Weier Wan[1],
Gert Cauwenberghs[2], H.-S. Philip Wong[1]
[1]Department of Electrical Engineering and Stanford SystemX Alliance,
Stanford University, Stanford, CA USA
[2]Department of Electrical and Computer Engineering,
University of California San Diego, San Diego, CA USA
[3]Department of Cognitive Sciences, University of California Irvine, Irvine, CA USA
E-mail: eryilmaz@stanford.edu, sijoshi@eng.ucsd.edu

Abstract

This paper gives an overview of recent progress on 1) online learning algorithms with spiking neurons 2) neuromorphic platforms that efficiently run these algorithms with a focus on implementation using analog-non-volatile memory (aNVM) as electronic synapses. Design considerations and challenges for using aNVM synapses such as requirements for device variability, multilevel states, programming energy, array-level connectivity, wire energy, fan-in/fan-out, and IR drop are presented. Future research directions and integration challenges are summarized. Algorithms based on spiking neural networks are promising for energy efficient real-time learning, but cycle-to-cycle device variations can significantly impact learning performance. Our analysis suggests that wires are increasingly important for energy considerations, especially for large systems.

Keywords

Cognitive computing, neuromorphic hardware, non-volatile memory, phase change memory, resistive switching memory, monolithic integration, device variability

1. Brain-inspired Algorithms

Two broad classes of brain-inspired algorithms include: 1) biology-based learning models (examples: spike-timing-dependent plasticity (STDP) and models of neurons and neural populations [1-3] that are derived from neuroscience research), and 2) artificial neural networks (ANNs) that are used to solve machine learning (ML) tasks. Algorithms in class 2 (ANNs) do not strictly mimic biology, but get inspiration from the biological brain to some extent in terms of connectivity and topology [4,5]. ANNs are primarily designed for conventional digital processors. They use batch-based discrete time updates that lack temporal locality (due to iterative updates) and spatial locality (due to shared parameters). On the other hand, biologically realistic spike-based algorithms, which is the focus of neuromorphic engineering, offer online real time learning through continuous weight updates that operate on local synaptic weights through local neural events [6]. In this section we give an overview of algorithmic frameworks for biologically realistic spiking neural network models that allow them to perform learning/inference in online fashion.

Concurrent with advances in Machine Learning (ML), where neural network models are currently the state-of-the-art in several practical tasks [7,8], in the past decade, biologically realistic algorithms have also made significant advances [9,10], with spiking networks approaching state-of-the-art in performance [11] on tasks such as MNIST character recognition. These works have employed spiking neurons, similar not only to what is found in biological systems, but also implemented in very power efficient neuromorphic hardware systems [1,12-14]. Many advances in the algorithmic aspect of neuromorphic engineering have also helped focus efforts on understanding the requirements for the underlying hardware computing units [15].

Probabilistic graphical models, which include Bayesian networks (directed graphical models) and random Markov fields (undirected graphical models) are employed by many artificial intelligence and expert systems [16]. While the link between populations of biological neurons and the mathematics of Bayesian inference is currently an open problem [17], progress is being made in this direction. One recently proposed hypothesis argues that spikes should be viewed as samples from an underlying a target probability distribution [10,18]. Neural Sampling (NS), in this form, proves to be an attractive neural information encoding strategy using substantially fewer spiking neurons than an alternative where neurons encode probabilities [19]. Abstract model neurons consistent with the behavior of biological neurons have been shown to implement Markov Chain Monte Carlo (MCMC) sampling [20], and Restricted Boltzmann Machines (RBMs) sampled in this way can be efficiently trained using Contrastive Divergence (CD), with almost no loss in performance [21]. Ref. [22] employs a stochastic approximation of Integrate and Fire (I&F) neurons

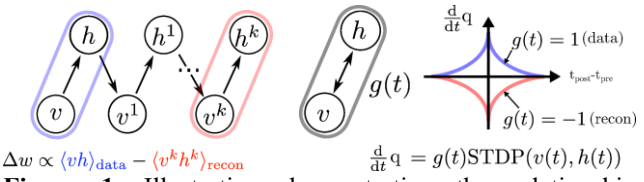

Figure 1. Illustration demonstrating the relationship between Contrastive Divergence (CD) [7] and event driven Contrastive Divergence (eCD) [24].

to compute CD updates, they utilize a relationship between firing rates of I&F neurons and the firing rates of their inputs

in order to perform off-line learning of the parameters of a Boltzmann Machine. Neftci et al. [24] implemented an online variant of CD [7], termed Event Driven Contrastive Divergence (eCD). The event driven nature of this algorithm (Fig. 1), makes this a particularly attractive candidate for adoption in neuromorphic spiking systems. Event Driven Contrastive Divergence is capable of emulating a Boltzmann Machine in a network driven by spiking I&F neurons, resulting in performance very competitive with the standard Boltzmann Machine as seen in Fig. 2.

The source of stochasticity in NS based algorithms (i.e., whether it is from neurons or synapses) is generally left unspecified, thus since there are a larger number of synapses

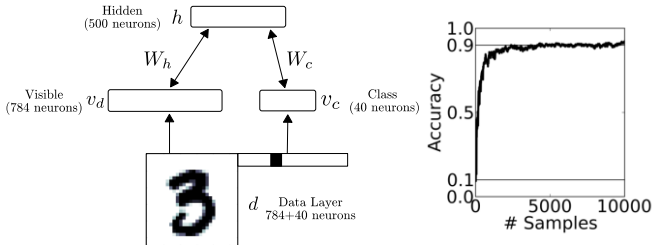

Figure 2. Training a spiking network of 784 neurons with eCD [24] results in better than 90% accuracy on the MNIST dataset.

available in a neural network, there has been recent interest [25] in exploiting synaptic stochasticity as this source via synaptic sampling (SS). Ref. [26] have demonstrated results on learning Winner-Take-All networks exploiting the stochasticity in synaptic plasticity. Ref. [6] have published the synaptic sampling machine (SSM), a class of stochastic spiking neural networks exploiting synaptic unreliability for computation. The use of a DropConnect [23] like noise model enables SSMs to outperform RBMs given the same number of neurons, truly highlighting the capabilities of spiking neural networks.

2. Brain-inspired hardware

Conventional hardware such as supercomputers, GPUs and CPUs have been employed for brain-inspired algorithms, both for ANNs and also for spiking neural networks [27-31]. Such implementations consume 100s of kWs of power in large scale [32]. As a result, such implementations in large scale are not available to researchers/engineers except the few that can afford this huge amount of computing power. In order to scale up both types of algorithms in systems size, hardware customization is essential [33].

Neuromorphic hardware employs connectivity, processing, and communication schemes that are inspired from real brain on the device, circuit [34-37], and architecture level [14,38-40]; and aims to mimic the real time processing power and energy efficiency of biological brain. Its processing nodes and memory are referred as neurons and synapses, respectively; through analogy to biological brain.

2.1. Neurons (processing)

Neuron implementations can be either analog [36] or digital [41]. Digital neuron design has the advantage of being easily programmable, scalable, robust and portable. On the other hand, analog neuron can operate with very low power, but scaling can be problematic because of the capacitors used in the design. Furthermore, since low power analog neurons rely heavily on leakage characteristics of transistors that control the membrane voltage on the capacitor, porting an analog design to more advanced processes might require significant effort to mitigate the effects of larger leakage in advanced technology nodes.

2.2. Asynchronous address-event-representation and routing schemes (communication)

Neurons and synapses operate at speeds much slower than typical silicon circuits, due to this, arrays of them can generally be serialized and multiplexed onto single fast communication busses. Thus communicating with asynchronous spike packets between many neural arrays is typically implemented using digital asynchronous communication between neural arrays for energy efficiency (no energy overhead for a global clock) [42]. Besides, digital schemes enable robust communication and mitigates mismatch problems between arrays and chips that is typical in analog implementations. Communication happens through address event representation [42], where a digital bus carries the address of the spiking neuron within one neural array and other relevant information to other neural array nodes using asynchronous handshaking protocols. Tree routing [35,43] or mesh routing [14] and N-dimensional Taurus [40] is used to route spike events between several nodes. In general, meshes have high bandwidth and high latency, whereas trees have low bandwidth and low latency [35]. Both meshes and trees support deadlock-free multicasting [44]. Since there is a higher multicasting overhead on meshes [35], tree based networks are more favorable.

2.3. Synapses (memory)

Because the number of synapses in a neural network is typically much larger than the number of neurons (a factor of $\sim 10^4$), the density, power, performance and wiring of the electronic synapses require special attention. IBM's all digital design TrueNorth employs SRAM synapses [14]. However, the SRAM synapse is volatile which might potentially cause issues; and is very area inefficient (SRAM area/cell > 120 F^2, where F is the feature size). The area inefficiency has limited the of number of bits/synapse (TrueNorth implements 1 bit/cell). DRAM has also been employed as synaptic memory, but DRAM is often off-chip due to process technology limitations, which makes it energetically expensive to access.

"New" (or "emerging") non-volatile resistive memory elements are two terminal memory devices and have characteristics that are very desirable as electronic synapses. These include resistive switching memory (RRAM) [45], phase change memory (PCM) [46], conductive bridge memory (CBRAM) [47], and ferroelectric memory (FeRAM) [48]. These devices have excellent size scalability, low energy operation, and analog programmability [49,50]. They can be monolithically integrated on top of CMOS in 3-dimension [51,52]. This allows designers to hide the CMOS neuron circuitry underneath multiple layers of synaptic memory [32]. Numerous studies have utilized the gradual resistance change behavior of these devices as synaptic elements in neural networks (ANNs or spiking networks), both in experiment (see fig. 3) [53-56] or simulation [57-59]. Many variations of the STDP rule can be implemented with these devices, making them attractive for spiking networks [46]. Recently, learning with backpropagation was also demonstrated with PCM devices, potentially useful for ANN implementations [55].

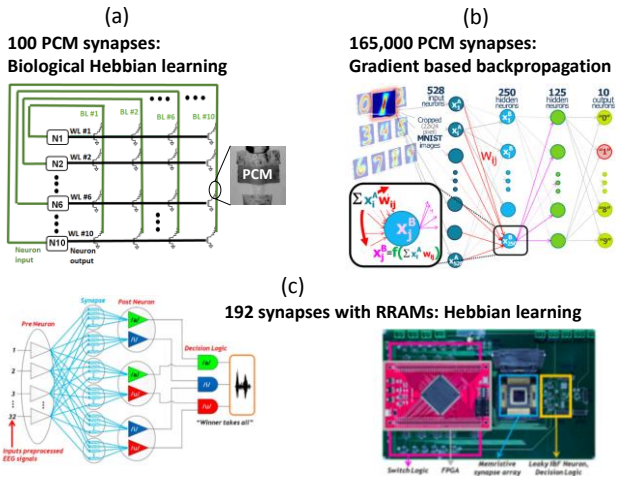

Figure 3. Experimental demonstrations of small and mid-scale networks using resistive memory devices: (a) Pattern recognition in a Hopfield network with 100 synapses (PCM) [53], (b) Digit recognition with a multilayer feedforward network with 165,000 synapses (PCM) [55] (c) WTA (winner-takes-all) architecture for EEG signal recognition with 192 synapses (RRAM) [54].

3. Design Considerations for aNVM synapses

While previous studies have shown that using aNVM device as synaptic element gives good performance in many learning/inference tasks, only very few of these simulation studies include non-idealities of real RRAM devices, such as non-uniform and non-deterministic conductance change as well as device-to-device variations, into consideration. In addition, device level and array level trade-offs that involve energy consumption, latency and circuit complexity should be considered for determining a design architecture.

3.1. Electronic Synaptic Device

Previous studies have shown that for some ML tasks, 5- or 6-bit digital synapses are able to perform as comparably to 64-bit (double precision) synapses [57,24]. Using 5-bit synapse reduces the recognition accuracy only slightly from 91.9% obtained with 64-bit synapse to 89.4%, although this study uses 64-bit synapse during training, and down-sample it to 5-bit only right before inference [24]. While NVM devices with more than 100 or more levels (equivalently 6-7 bits) have been demonstrated, it is typically impossible to precisely control the conductance level using single shot programming [46,50]. Moreover, the conductance does not change linearly over the entire conductance range. Furthermore, some devices show gradual resistance change only in one direction (only from high resistance to low resistance, etc.). To mitigate these effects, besides pursuing device engineering for more precise control of the conductance, solutions on both algorithm and task level should be investigated. On the algorithm level, using stochastic binary switching of RRAM and CBRAM during training has been proposed [60,61]. Alternatively, two devices with a differential read-out can be used to store synaptic weight such that gradual weight change can happen in both directions, while trading off synapse density [59]. When a realistic RRAM model [62] calibrated to experimental measurements is used where 2-RRAM synapse is employed, an R_{off}/R_{on} ratio of 500 can achieve an 88% recognition accuracy (out-of-sample) on MNIST dataset using a supervised contrastive divergence training of a 2-layer (1 hidden layer) restricted Boltzmann machine (RBM) [32]. Although this accuracy is lower than the result with state-of-the-art neural networks in ML literature (99.79%) [23], it is comparable to a 2-layer RBM (with same topology) using 64-bit digital synapses (92%) [24]. In the same study [32], RRAM is initialized to low-R state and gradual RESET is performed during training. It is observed that the differential read-out with 2-RRAM synapses could relax R_{off}/R_{on} requirement to much less than 10^3 suggested in previous studies [60,63], because the conductance of two cells can be brought very close to each other through learning to realize very small differential weight. Meanwhile, we observe that device-to-device variations are largely ameliorated in differential readout synapses since the training can reduce the effect of device-to-device variations in a way to make the differential weight minimize the overall error. This way, the accuracy of weight update is limited not by device-to-device variation but rather by the precision of conductance change with one pulse and cycle-to-cycle

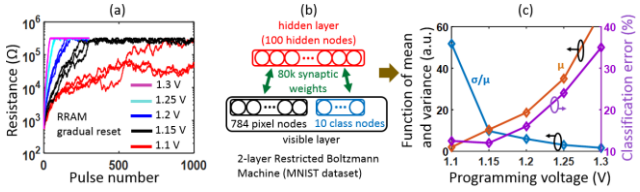

Figure 4. (a) Gradual resistance change in RRAM device with single pulses for different programming voltages (b)RBM architecture used in [32] (c) Effect of programming voltage on classification accuracy, average log-conductance change(μ), and variation in conductance change (σ/μ). After [32]

variation. Furthermore, intrinsic device-to-device variation can be advantageous for weight initialization due to symmetry breaking, which results in more efficient learning than initializing all weights to the same value [64]. Generating randomness requires pseudo-random number generator hardware on conventional computer, but can be easily done with NVM devices [65]. The effect of device-to-device variations can also be mitigated by trading off latency, energy or area in some cases. Using a Hopfield network, it was previously shown that an increase in device-to-device variability from 9% to 60% could be tolerated by using more learning iterations (from 1 to 11) and therefore consuming more energy (4nJ to 54nJ) [66]. Using redundant memory cells can also reduce variation, but trades off area and energy [67].

The cycle-to-cycle variations and number of gradual levels can have a strong impact on learning performance. Fig. 4c shows that the classification accuracy of a RBM (see Fig. 4b) trained on MNIST dataset using CD algorithm heavily depends on both conductance change rate (which is modulated by programming voltage) and cycle-to-cycle variation (which again depends on programming voltage, as shown in fig 4a). Also it is noteworthy that the learning performance can degrade when nonlinearity in conductance change is large [55]. Therefore, it is important to employ a device model that can accurately capture these non-ideal device behaviors. This emphasizes the algorithm-device interplay in the analysis of aNVM based neuromorphic hardware.

For choosing the device to be used as electronic synapse, besides aforementioned variability and non-linearity, programming energy is also crucial. RRAM can realize more than 100 gradual level and can go down to a few ns of write time with <pJ of programming energy [68], and therefore is a good candidate for synaptic device if its variability can be improved or tolerated. PCM has less conductance variation, but integration on smaller technology nodes is needed to reduce write energy [49,68].

3.2. Array size and energy considerations

Energy consumption plays an essential role in determining RRAM programing voltage and network architecture. Energy consumption in synaptic arrays consists of two major parts: wire energy (CV^2) and programming energy ($V^2t_{pulsewidth}/R$). Wire energy is crucial because even for a 1k×1k array, wire energy (order of 1pJ) starts dominating [32,68]. RRAM programming energy scales roughly inversely with resistance. When low-R of RRAM cells is ~600 Ω, RRAM energy is 3 orders of magnitude higher than wire energy, again for the case study with RBM training [32]. The wire energy becomes comparable to RRAM energy when low-R is ~600 kΩ (see Fig. 5b). In most cases, low programming voltage is desirable because it lowers both programming and wire energy. Wire energy directly scales quadratically with the programming voltage, whereas RRAM energy has more complicated dependencies [32]. If training only takes a few epochs, and the RRAM cells are initialized to low-R state before training, lowering programming voltage too much results in RRAM cells remaining in low-R state for an extended period due to lower conductance change per pulse, which significantly increases energy consumption (see Fig. 5a). This effect disappears after 50 epochs.

In neural networks whether they are biologically realistic or belonging to the ANN class, neurons usually have large fan-in and fan-outs. Using a huge array to implement the largest fan-out within one array is not energy efficient because it requires continuously charging and discharging long wires in the array. Instead, if the connectivity of the network or neuron activations are sparse, replacing big arrays with multiple small corelets can significantly reduce wire energy. Wire energy is reduced by 60% when four 32×32 arrays are used in place of one 128×128 array for the case of 256-4-256 deep autoencoder [32]. However, the overhead from the communication between corelets attenuates the energy benefits of using smaller array sizes. Furthermore, smaller arrays might require time-multiplexing of input neurons to accommodate large fan-ins, which introduces latency. Therefore, the tradeoffs between energy consumption, latency and circuit complexity should be considered when determining the array size [32].

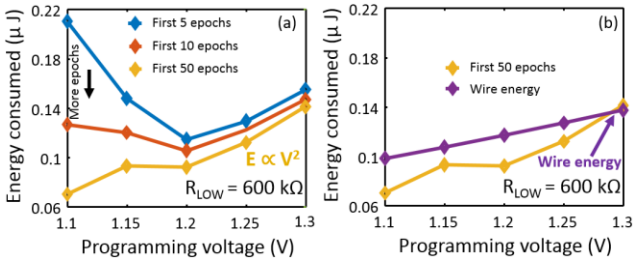

Figure 5. (a) Energy consumption in training phase for RBM training within RRAM devices. Blue, red and orange curves are energy consumed per epoch (averaged over first 5, 10 and 50 epochs). (b) Wire energy assumes 1k×1k array with 100 nm full pitch. After [32]

When a large array is used, IR drop along the wire can significantly degrade the RRAM read accuracy [67] as well as introduce variations in programming due to degraded programming pulse amplitude across the cells away from the power supply [32]. To mitigate IR drop, RRAM cells should be operated in higher R regimes, which also lowers the energy consumption. When the analog current is converted to digital data using an ADC, increasing R_{on} from 10 kΩ to 1 MΩ reduces read inaccuracy from 20% to less than 1%, while lowering read energy from 1 pJ to 20 fJ [67]. Increased read latency should be taken into account when R_{on} is higher. In summary, the connectivity of network architecture, R_{on} and R_{off}/R_{on} ratio, area of neuron vs synapses should be analyzed jointly to understand the tradeoffs of different array sizes.

Acknowledgements

This work is supported in part by SONIC, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA, the NSF (award EFRI-1137279), the Office of Naval Research (ONR

MURI 14-13-1-0205), the NSF Expedition on Computing (Visual Cortex on Silicon, award 1317470), Intel Corporation (ISRA on Neuromorphic Architectures for Mainstream Computing), and the member companies of the Stanford Non-Volatile Memory Technology Research Initiative (NMTRI) and the Stanford SystemX Alliance. Collaborations and discussions with Jinfeng Kang from PKU; Chung Lam and SangBum Kim from IBM are highly appreciated.

References

[1] G. Indiveri et al., "Neuromorphic Silicon Neuron Circuits," *Front. Neurosci*., vol. 5, 2011.

[2] C. Ramos et al., "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Front. Neurosci*., vol. 5, 2011.

[3] D. George et al., "Towards a mathematical theory of cortical micro-circuits," *PLoS Comput. Biol*., vol. 5, no. 10, 2009.

[4] Y. LeCun et al., "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436-444, 2015.

[5] G. Hinton, S. Osindero and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comp*., vol. 18, no. 7, pp. 1527-1554, 2006.

[6] E. Neftci et al. "Unsupervised Learning in Synaptic Sampling Machines," *arXiv preprint arXiv:1511.04484*, 2015.

[7] G. E. Hinton and R.R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[8] Y. Bengio, "Learning deep architectures for AI," *Found. Trends. Mach. Learn.*, vol. 2, pp. 1–127.

[9] P. Merolla, T. Ursell, and J. Arthur, "The thermodynamic temperature of a rhythmic spiking network," *ArXiv e-prints*, September 2010.

[10] J. Fiser, P. Berkes, G. Orbán, and M. Lengyel, "Statistically optimal perception and learning: from behavior to neural representations," *Trends in Cognitive Sciences*, vol. 14, no. 3, pp. 119–130, 2010.

[11] S. Esser et al., "Backpropagation for energy-efficient neuromorphic computing", NIPS, Montreal, 2015.

[12] T. Yu et al., "65k-neuron integrate-and-fire array transceiver with address-event reconfigurable synaptic routing, " *BioCAS, IEEE*, pp.21-24, 28-30 Nov. 2012.

[13] J. Park et al., "A 65k-neuron 73-mevents/s 22-pj/event asynchronous micro-pipelined integrate-and-fire array transceiver," *BioCAS, IEEE*, Oct. 2014.

[14] P. A. Merolla et al., "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[15] S. Sheik, M. Pfeiffer, F. Stefanini, G. Indiveri, "Spatio-temporal spike pattern classification in neuromorphic systems," *Biomimetic and Biohybrid Systems*, Springer Berlin Heidelberg, pp. 262-2732013.

[16] D. Koller, and N. Friedman, "Probabilistic Graphical Models: Principles and Techniques," MIT press, 2009.

[17] K. Doya, S. Ishii, A. Pouget, and R. Rao, "Bayesian Brain: Probabilistic Approaches to Neural Coding," Cambridge, MA, MIT Press, 2007.

[18] P. Berkes, G. Orbán, M. Lengyel, J Fiser, "Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment," *Science*, vol. 331, no. 6013, pp. 83-87, 2011.

[19] C. Eliasmith, and C. S. Terrence, "Nengo and the Neural Engineering Framework: From Spikes to Cognition." Cognitive Science Society, 2012.

[20] L. Buesing, J. Bill, B. Nessler, and W. Maass, "Neural dynamics as sampling: A model for stochastic computation in recurrent networks of spiking neurons," *PLoS Comput. Biol*., vol. 7, no. 11, 2011.

[21] B. U. Pedroni et al., "Neuromorphic adaptations of restricted boltzmann machines and deep belief networks," *Neural Networks (IJCNN), The 2013 International Joint Conference on, IEEE*, pp. 1-6, 2013.

[22] P. O'Connor et al., "Real-time classification and sensor fusion with a spiking deep belief network," *Front. Neurosci*., vol. 7, 2013.

[23] L. Wan et al., "Regularization of neural networks using dropconnect," ICML, 2013.

[24] E. Neftci et al., "Event-driven contrastive divergence for spiking neuromorphic systems," *Front. Neurosci*., vol. 7, 2013.

[25] L. Aitchison, and P. E. Latham, "Synaptic sampling: A connection between PSP variability and uncertainty explains neurophysiological observations," *arXiv preprint arXiv:1505.04544*, 2015.

[26] D. Kappel, S. Habenschuss, R. Legenstein, and W. Maass, "Synaptic sampling: A Bayesian approach to neural network plasticity and rewiring," *NIPS*, 2015.

[27] R.Ananthanarayanan et al., "The cat is out of the bag: cortical simulations with 109 neurons, 1013 synapses," *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis.* New York, NY, USA: ACM, 2009.

[28] Q. V. Le et al., "Building high-level features using large scale unsupervised learning," *ICML*, 2012.

[29] A. Krizhevsky et al., "Imagenet classification with deep convolutional neural networks," *NIPS*, 2012.

[30] R. Raina, A. Madhavan, and A. Y. Ng, "Large-scale deep unsupervised learning using graphics processors," *ICML,* 2009.

[31] V. Vanhoucke et al., "Improving the speed of neural networks on CPUs," *Deep Learning and Unsupervised Feature Learning NIPS Workshop*, 2010.

[32] S. B. Eryilmaz, et al., "Device and system level design considerations for analog-non-volatile-memory based neuromorphic architectures," *Electron Devices Meeting, 2015. IEDM'15 Technical Digest. IEEE International*, pp. 4.1.1-4, 2015.

[33]http://users.ece.gatech.edu/mrichard/ExascaleComputingStudyReports/exascale_final_report_100208.pdf

[34] C. Mead, "Neuromorphic electronic systems," *Proc. IEEE*, vol. 78, no. 10, pp. 1629-1636, 1990.

[35] B. V. Benjamin et al., "Neurogrid: A mixed-analog-digital multichip system for large-scale neural simulations," *Proc. IEEE*, vol. 102, no. 5, pp. 699-716, 2014.
[36] R. J. Vogelstein et al., "Dynamically reconfigurable silicon array of spiking neurons with conductance-based synapses," *IEEE Trans. Neural Netw.*, vol. 18, no. 1, pp. 253-265, 2007.
[37] G. Indiveri et al., "A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity," *IEEE Trans. Neural Netw*., vol. 17, no. 1, pp. 211-221, 2006.
[38] A.S. Cassidy et al., "Design of silicon brains in the nano-CMOS era: Spiking neurons, learning synapses and neural architecture optimization," *Neural Networks*, vol. 45, pp. 4-26, 2013.
[39] J. Gehlhaar, "Neuromorphic processing: a new frontier in scaling computer architecture," *ASPLOS*, 2014.
[40] M.M. Khan et al.,"SpiNNaker: mapping neural networks onto a massively-parallel chip multiprocessor," *IJCNN*, 2008.
[41] J. Seo et al., "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," *CICC*, 2011.
[42] K. Boahen, "Point-to-point connectivity between neuromorphic chips using address events," *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on,* vol 47, no. 5, pp. 416-434, 2000.
[43] S. Joshi et al., "Scalable event routing in hierarchical neural array architecture with global synaptic connectivity," *Cellular Nanoscale Networks and Their Applications (CNNA), 12th International Workshop on*., IEEE, 2010.
[44] E. A. Carara, G. M. Fernando, "Deadlock-free multicast routing algorithm for wormhole-switched mesh networks-on-chip," *Symposium on VLSI, 2008, IEEE Computer Society Annual*, 2008.
[45] S. Yu et al., "An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation," *IEEE Trans. Electron Devices*, vol. 58, no. 8, pp. 2729-2737, 2011.
[46] D. Kuzum et al., "Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing," *Nano Letters*, vol. 12, no. 5, pp. 2179-2186, 2011.
[47] T. Ohno et al., "Short-term plasticity and long-term potentiation mimicked in single inorganic synapses," *Nature Mat*., vol. 10, no. 8, pp. 591-595, 2011.
[48] A. Chanthbouala et al.," A ferroelectric memristor," *Nature Mat.,* vol. 11, no. 10, pp. 860-864, 2012.
[49] D. Kuzum et al., "Synaptic electronics: materials, devices and applications," *Nanotechnology*, vol. 24, no. 38, pp. 382001, 2013.
[50] S. Yu et al., "A Low Energy Oxide-Based Electronic Synaptic Device for Neuromorphic Visual Systems with Tolerance to Device Variation," *Adv. Mat*., vol. 25, no. 12, pp. 1774-1779, 2013.
[51] M. M. Shulaker et al., "Monolithic 3D integration of carbon nanotube FETs, resistive RAM, silicon FETs," *IEDM*, 2014.
[52] Y. Liauw et al., "Nonvolatile 3D-FPGA with monolithically stacked RRAM-based configuration memory," *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012.*
[53] S. B. Eryilmaz et al., "Experimental demonstration of array-level learning with phase change synaptic devices," *IEDM*, 2013.
[54] S. Park et al., "Electronic system with memristive synapses for pattern recognition," *Scientific Reports*, vol. 5, 2015.
[55] G. Burr et al., "Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element," *IEDM*, pp. 29-5, 2014.
[56] M. Prezioso et al., "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, pp. 61-64, 2015.
[57] P. Y. Chen et al., "Technology-design co-optimization of resistive cross-point array for accelerating learning algorithms on chip," *DATE*, pp. 854-859, 2015.
[58] D. Kuzum, et al. "Energy efficient programming of nanoelectronic synaptic devices for large-scale implementation of associative and temporal sequence learning," *IEDM*, 2011.
[59] O. Bichler et al., "Visual Pattern Extraction Using Energy-Efficient "2-PCM Synapse" Neuromorphic Architecture," *IEEE Trans. Electron Devices*, vol. 59, no. 8, pp. 2206-2214, 2012.
[60] S. Yu et al., "Stochastic learning in oxide binary synaptic device for neuromorphic computing," *Front. Neurosci.,* vol. 7, 2013.
[61] M. Suri et al.," CBRAM devices as binary synapses for low-power stochastic neuromorphic systems: auditory (cochlea) and visual (retina) cognitive processing applications," *IEDM*, 2012.
[62] Z. Jiang, S. Yu, Y. Wu, J. H. Engel, X. Guan, H.-S. P. Wong, "Verilog-A Compact Model for Oxide-based Resistive Random Access Memory (RRAM)," *International Conference on Simulation of Semiconductor Processes and Devices (SISPAD),* paper 3-3, pp. 41 – 44, Yokohama, Japan, September 9 – 11, 2014. Model available at: https://nanohub.org/publications/19
[63] D. Querlioz et al., Simulation of a memristor-based spiking neural network immune to device variations," *IJCNN*, 2011.
[64] G. E. Hinton, "A practical guide to training restricted Boltzmann machines," *Tech. Rep. UTML TR 2010-003*, Dept. Comput. Sci., Univ. Toronto, 2010.
[65] A. Chen et al., "Comprehensive assessment of RRAM-based PUF for hardware security applications," *IEDM*, pp. 10.7.1-4, 2015.
[66] S. B. Eryilmaz et al., "Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array," *Front. Neurosci.*, vol. 8, 2014.
[67] S. Yu et al., "Scaling-up resistive synaptic arrays for neuro-inspired architecture: Challenges and prospect," *IEDM*, pp. 17.3.1-4, 2015.
[68] https://nano.stanford.edu/stanford-memory-trends