Spiking neural networks — Algorithms, hardware implementations and applications

Conference Paper · August 2017

DOI: 10.1109/MWSCAS.2017.8052951

3 authors:

Shruti Kulkarni
New Jersey Institute of Technology
22 PUBLICATIONS **149** CITATIONS
SEE PROFILE

Anakha V B
New Jersey Institute of Technology
8 PUBLICATIONS **49** CITATIONS
SEE PROFILE

Bipin Rajendran
125 PUBLICATIONS **4,898** CITATIONS
SEE PROFILE

Spiking Neural Networks – Algorithms, Hardware Implementations and Applications

Shruti R. Kulkarni, Anakha V. Babu, Bipin Rajendran
Department of Electrical and Computer Engineering,
New Jersey Institute of Technology, Newark, NJ, 07102, USA
Email: {srk68, av442, bipin}@njit.edu

***Abstract*—Spiking Neural Networks (SNNs) are the third generation of artificial neural networks that closely mimic the time encoding and information processing aspects of the human brain. It has been postulated that these networks are more efficient for realizing cognitive computing systems compared to second generation networks that are widely used in machine learning algorithms today. In this paper, we review the learning algorithms, hardware demonstrations and potential applications of SNN based learning systems.**

I. Introduction

The organization of the human brain motivated and inspired the pioneers of computing in designing the architecture of modern microprocessors [1]. However, even after several decades of research and progress, human brain still outperforms modern processors in both speed and power on many tasks involving unstructured data classification and pattern recognition. The human brain is a seamlessly interconnected network of neurons and synapses. Most computing systems, on the other hand, are based on the von-Neumann architecture, with physically separated memory and execution units. In this model, data is fetched from the memory unit to the execution unit for processing and the results of the computation are stored back in the memory. Hence, the performance of computations that involve large data movement between memory and processor is limited not by the processor speed but by the memory access time and associated power consumption [2]. In order to bridge this 'von Neumann bottleneck' between the processor and the memory, several solutions have been proposed such as monolithic 3D integration and storage class memories [3], [4]. But even with these solutions, memory access continues to be the bottleneck for many 'big data' analytics applications which involve large data movement.

Inspired by the computational capability of the human brain, several neural networks based on the brain's architecture have been explored to naturally discover inherent structure in large data streams. Among those, the most popular are the *second* generation Artificial Neural Networks (ANN) used for deep learning [5]. These networks are trained by the back-propagation algorithm which computes the gradient of some objective function with respect to the weights in the network. These networks have established themselves as the state of the art in a number of high-level cognitive applications [6]–[8].

Spiking neural networks (SNNs) are the *third* generation of artificial networks and they mimic their biological counterparts more closely [9]. The neurons used in these networks issue spikes as a communication token and the modulation of synaptic strength depends on the arrival time of spikes. This time-dependent synaptic plasticity is believed to be a fundamental mechanism for learning in the brain [10]. SNNs have more computational power compared to non-spiking network due to their ability to incorporate temporal dimension in information representation [9].

SNNs can also potentially enable highly power efficient cognitive processing systems [11]. The neurons in the brain spike at a rate of $10 - 100\,\text{Hz}$ and process information in an event-driven manner. Different schemes such as rate coding, temporal coding, population coding and first-time-to-spike coding have been explored to encode information in the spike domain [12]. Hardware chips such as the IBM's TrueNorth [13], Qualcomm's Zeroth [14], and the ROLLS chip from INI Zurich [15] are neuromorphic processors that compute with spikes and minimize the communication bottleneck in von-Neumann architectures. The TrueNorth chip, for example, consumes less than $100\,\text{mW}$ when realizing a network with close to 256 million synapses, although it does not support on-chip and real-time learning.

In this paper, we first introduce the basic units and the mathematical models for neurons and synapses of spiking neural networks (SNNs) in Section II. We then discuss supervised and unsupervised learning algorithms that have been published for SNN training in Section III. Some acceleration strategies in both software (with parallel programming on GPUs) and with custom neuromorphic architectures are discussed in section IV. Section V presents some emerging nanoscale devices that are currently being explored for emulating the behavior of SNNs. Section VI presents an overview of exemplary applications that employ SNNs for information processing. Lastly, we discuss the challenges that need to be addressed in designing neuromorphic systems approaching the efficiency of the human brain in section VII.

II. Spiking Neural Networks

Figure 1 illustrates two spiking neurons connected by a synapse with strength of w. The neurons' membrane potential $V(t)$ can be modeled by the computationally simple leaky-integrate-and-fire (LIF) model [16] described by the equation,

$$C\frac{dV(t)}{dt} = -g_L(V(t) - E_L) + I(t) \tag{1}$$

for an input excitation current $I(t)$. Here, C and g_L represents the membrane capacitance and leak conductance, respectively. Once $V(t)$ crosses a threshold V_T, it is reset to its resting state E_L and remains there till the neuron comes out of its refractory period t_{ref}.

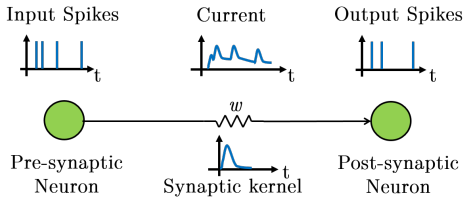

Fig. 1: Basic communication framework in a spiking neural network; reproduced from [17]. Neurons communicate to each other via issuing spikes, and the strength of the communication signal is encoded in the effective conductance of the synapse.

Mathematically, the synapses can be thought of as performing low-pass filtering of the input spikes, each arriving at time t^i and then weighting them to generate the current.

$$c(t) = \sum_i \delta(t - t^i) * (\exp(-t/\tau_1) - \exp(-t/\tau_2)) \quad (2)$$

$$I(t) = w \times c(t) \quad (3)$$

Here, $c(t)$ represents the double exponential post-synaptic kernel (as shown in Figure 1), defined by the rising and falling time constants τ_2 and τ_1 respectively. Neuro-biological studies have suggested that correlation in timing of the spikes arriving on a synapse play a key role in adjusting its strength (Hebb's law [18]). The phenomenon of spike timing dependent plasticity (STDP) which adjusts the synaptic strength based on the precise timing of spikes arriving from the pre and post-synaptic neurons is believed to be fundamental to the brain's learning abilities [19] (Figure 2). STDP is an unsupervised learning rule and has been employed in SNNs for cognitive tasks such as pattern recognition [20], [21].

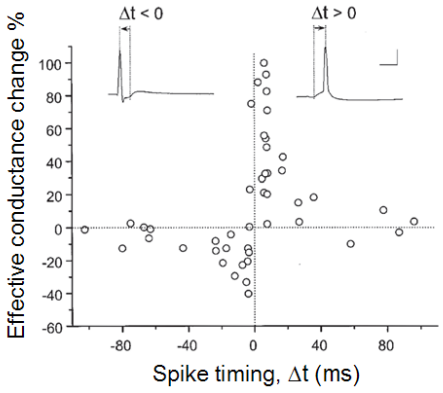

Fig. 2: Biological spike timing dependent plasticity showing that causal spike pairs lead to increase in effective conductivity of synapses and vice versa; reproduced from [19].

III. Learning algorithms

Supervised training in the second generation ANNs involves evaluating the network's output error with the desired output and back-propagating the error derivatives for updating the parameters. The cost function is a continuous convex function of the network's training error. Typically gradient descent is used to iteratively adjust the network's parameters over multiple training epochs.

Training SNNs to generate desired spike patterns in response to input excitation is a harder mathematical problem. In SNNs, the training error is the difference in the spike timings of the actual and desired spike trains, both of which are discrete signals in time. The training algorithms have to be tuned to account for this discontinuity. Real-world signals also have to be first translated into spike domain, by some spike encoding strategy. Though algorithms for this are not well established, some time-encoding strategies have been proposed for band-limited signals [22].

Several algorithms have been proposed to train SNNs that use the precise timings of the input and output spikes as the error for adjusting the network's weights. One of the earliest proposals was the SpikeProp algorithm, which described the cost function in terms of the difference between desired and actual spike times [23]. However, this rule was limited to learning a single spike. There have been modifications to this rule that incorporate learning with multiple spikes demonstrating convergence speedup such as QuickProp and RProp [24], [25]. Algorithms such as the spike pattern association (SPAN), Precise spike driven synaptic plasticity (PSD) and the remote supervised (ReSuMe) are based on the Widrow-Hoff rule [26]–[28]. SPAN and PSD have their cost functions defined in terms of the difference in spike trains converted to analog signals. The ReSuMe algorithm uses local spike timing dependent plasticity (STDP) rules for weight update [19]. The DL-ReSuMe algorithm uses the transmission delays of synapses interconnecting the neurons, in addition to their weights as learning parameters [29]. However, none of these have algorithms have been used to demonstrate learning in large benchmark data-sets used in machine learning.

Our group has developed a gradient descent based learning rule for SNNs called the normalized approximate descent (NormAD) that casts the cost function in terms of the membrane potential, instead of the time of spikes [17]. The weight update is still spike triggered and is given by:

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \int_0^T e(t) \frac{\hat{\mathbf{d}}(t)}{|\hat{\mathbf{d}}(t)|} dt \quad (4)$$

where the factor $\hat{\mathbf{d}}(t)$ represents the convolution between the incoming synaptic input and the neuron's impulse response and $e(t)$ is the sampling function that detects points of discrepancy between desired and observed spikes. NormAD algorithm has shown faster convergence over other bio-inspired algorithms such as ReSuMe [17] and has been used to classify handwritten digits from the MNIST database with over 98% test accuracy [30].

Lee, et al., have demonstrated SNN back-propagation for both fully connected and convolution network architectures [31]. The cost function is defined in terms of the error between observed and desired spike times by treating the discontinuous nature of the cost function as noise for gradient computation. Their convolutional SNN has achieved a classification accuracy of 99.31% on the MNIST dataset.

There are also several efforts to convert a trained second generation ANN into an equivalent SNNs [32]–[34]. In such networks, the firing rates of spiking neurons correspond to the activations of ANN neurons. Techniques such as the addition of noise, replacing max-pooling with averaging, the introduction of lateral inhibition, weight normalization, etc. have been employed in converting the non-spiking networks to spiking ones. It has been shown that for problems involving static inputs such as images, the accuracy loss after such conversions is negligible. However, it is not clear if such conversion strategies are applicable for networks that need to process time-varying signals.

IV. Hardware architecture and implementations

SNNs require parallel synaptic signal transmission and weight update and implementing them on CPUs that are inherently serial limits their speed. Hence, strategies to parallelize their operations are necessary. General purpose graphical processing units (GP-GPUs) with multiple scalar processors and streaming multiprocessors are ideal for implementing deep learning networks. There have been a number of hardware-software based co-simulation approaches for realizing SNNs that include the use of GPUs, FPGAs and microcontrollers that have shown significant speedup over CPUs [35]–[37]. Several GPU based simulators for the SNNs have also been developed [35], [38], [39]. Most of these simulators implement the basic real-time simulation of large SNNs and local STDP-like weight update rules.

We recently developed a parallel implementation of a three layer SNN on a GPU platform for supervised learning using the NormAD algorithm [30]. Our implementation achieved a speedup of ~ 6 over a single core CPU implementation. An FPGA-based implementation of an SNN with two million neurons was shown to achieve a $38\times$ speedup compared to a serial implementation on a CPU [40].

There have also been efforts to design custom neuromorphic architectures with spiking neurons and synapses using both digital and analog circuits [41]–[44]. The most prominent among these is a 5.4 billion transistor chip called TrueNorth with 1 million spiking neurons, 4096 neurosynaptic cores and 256 million configurable synapses (1 bit SRAM) fabricated with an area of 4.3 cm^2 in 28 nm CMOS [13]. The basic building block of TrueNorth is a neurosynaptic core which has 256 input lines (axons) for communication, 256 output neurons for computation and 256×256 synapses as memory (schematically shown in Figure 3). Spike events are transmitted between cores by time multiplexed wires using interconnected routers. TrueNorth architecture is highly scalable as the cores can be tiled in 2 dimensions similar to the mammalian cortex. The

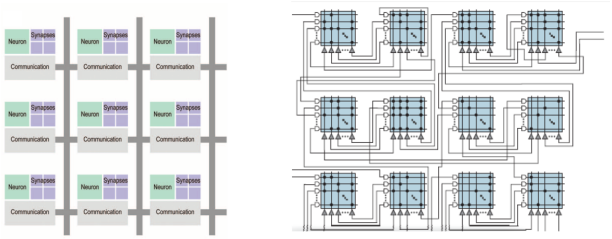

Fig. 3: Conceptual blue print of TrueNorth architecture localizing computation, communication, and memory (left); spike routing allows interconnection of neurosynaptic cores to form long-range networks; reproduced from [13].

average power density of TrueNorth is $20\,mW/cm^2$, which is comparable with the brain (vs. a typical CPU power density of $100\,W/cm^2$). Even though TrueNorth does not support on-chip learning, there are no fundamental limitations in realizing learning systems in CMOS hardware [45].

The most notable analog implementation is the Reconfigurable On-line Learning (ROLLS) spiking neuromorphic processor that more closely captures the physics of biological neurons and synapses [15]. ROLLS chip can implement spike based adaptation and plasticity mechanisms and can support on-chip learning. The block diagram of ROLLS microprocessor is shown in the Figure 4; it has separate synapse arrays to implement long term and short term plasticity.

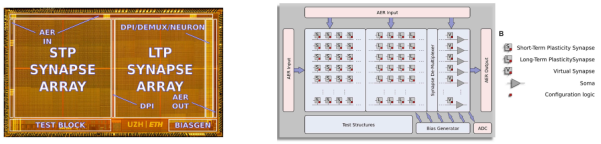

Fig. 4: The micro photograph of ROLLS neuromorphic processor fabricated using $180\,nm$ CMOS technology with an area of $51.4\,mm^2$ and approximately 12.2 million transistors (left); Block diagram of ROLLS neuromorphic processor with separate synapse arrays for long term and short term plasticity, virtual synapses and row of neurons; reproduced from [15].

V. Nanoscale devices for SNNs

There are also significant efforts to build dedicated nanoscale devices that naturally mimic the dynamics of biological neurons and synapses. Cross bar arrays integrated with analog memristive devices as synapses at each junction and nanoscale neuron devices/circuits at the periphery is an efficient scheme for realizing large neuromorphic systems due to its parallel processing capabilities [46].

One of the earliest demonstrations of synaptic plasticity in a nanoscale device was the STDP demonstration in chalcogenide based Phase Change Memory (PCM) based on waveform engineering (Figure 5) [47], [48]. Recently, there have also been proposals to implement stochastic integrate and fire neurons leveraging the crystallization dynamics of these nanoscale materials [43] (Figure 6).

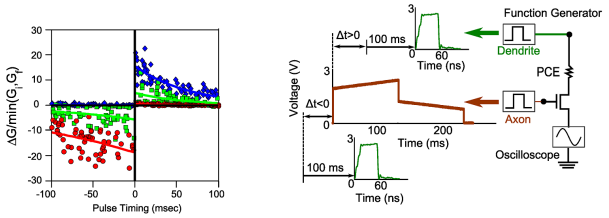

Fig. 5: Demonstration of STDP in phase change memory (left); waveforms that encode the relative timing of spikes of pre and post neurons are used for programming; reproduced from [48].

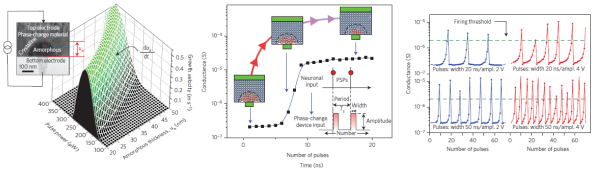

Fig. 6: The dynamics of crystallization in phase change alloys (left); device conductance changes gradually as a function of the number of pulses (center); mimicking integrate and fire dynamics; reproduced from [43].

Building up on these ideas, several other materials systems have been explored to improve the efficiency of neuronal and synaptic devices. $Pr_{0.7}Ca_{0.3}MnO_3$ (PCMO) based alloys that exhibit bipolar memristive switching below $\pm 1.5\,V$ [49] and Mn-doped HfO_2 RRAM devices consuming about $500\,fJ$ [50] are notable examples of synaptic demonstrations from our research. In addition to the above examples, emerging memories such as PCM [47], [51], RRAM [52], MemFET [53] and OxRAM [54] have also been used for synaptic applications. For neuronal dynamics, insulator-metal transitions in VO_2 also appear promising [44], [55], [56].

Based on the phenomenological models of these devices, there are also emulations of applications such as sequence learning [48], visual pattern recognition in videos [57], audio sequence learning [50], and unsupervised learning [58]. Though large arrays of such emerging memory devices (in particular, PCM [59]) have been used for learning demonstration of second generation networks, large-scale experimental demonstrations of SNNs using these devices are awaited; see [60] for a review.

VI. Applications

Various groups have demonstrated applications that use SNNs for cognitive processing tasks and bio-medical applications. Some of these include movement generation and control for neuro-prosthetics [61], spatio-temporal brain data processing with EEG [62], obstacle avoidance based on processing real-time video [63], path planning [64], financial data analysis in an unsupervised manner with polychronous SNN [65], gait-event detection in real time in a supervised manner [66], and music composer classification [67]. ROLLS microprocessor along with a Dynamic Vision Sensor (DVS) camera was used for image classification tasks, with the DVS camera sending in spike trains to the synaptic array [15]. The TrueNorth chip with a pre-trained network has demonstrated superior performance in tasks such as multi-object detection and classification from a video scene entirely with spikes, and also in realizing orientation-selective filters for feature extraction [13].

As a proof of concept of the efficiency of SNNs, our group developed an adaptive SNN with 10 neurons to control the motion of a robot inspired by the thermo-taxis behavior and network of the nematode *C. elegans* [68], [69] (Figure 7 (a)). The control circuitry of the robot consists of Izhikevich neurons, which makes spike based navigational decisions so that the robot tracks a path of a particular intensity in a given region. This intensity is pre-configured into the system and only the local intensity at the location of the robot is measured using a single sensor. The neurons and synapses of the circuit are implemented using Atmega and AT-tiny microcontrollers and mounted atop a mobile platform of the robot. Control signals from the three output neurons of the circuit, for turning left, right or random foraging, control the motors of the robot. The robot performs all the sensing, processing and motor actuation in real-time, using spikes. Figure 7(b) shows a sample trajectory (simulated in MATLAB) of the 'software worm' starting from an arbitrary initial point, tracking a set temperature of 20° C. We have recently shown that SNNs for navigation that are inspired by the echo-locating circuits of the bat are at least 200x more efficient than simple PID based control in decision efficiency [70].

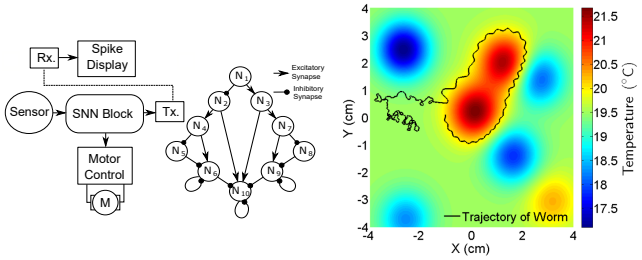

Fig. 7: (a) The control circuitry of the path tracking robot consisting of 10 spiking neurons with a single sensor measuring local intensity (Left); reproduced from [69] (b) Simulated trajectory traced by the network for a set temperature of $T_s = 20^\circ$ C starting from a random initial point (Right); reproduced from [68].

Our group has also developed a three layer SNN for handwritten digit recognition, which achieved over 98% classification accuracy on the MNIST dataset [30]. We are currently implementing this as a stand-alone embedded system for Optical Character Recognition (OCR).

VII. Challenges and Future Outlook

Significant progress has been achieved in the last decade in building new learning algorithms and hardware implementation strategies employing nanoscale CMOS and post-CMOS devices for various cognitive applications using spiking neural

networks (SNNs). However, these demonstrations have not yet achieved the level of adoption that second generation networks that are used in deep learning enjoy today.

The most important challenge in our opinion is the lack of learning algorithms that are efficient for training deep spiking networks. The prospect for realizing large scale neuromorphic systems leveraging adaptive nanoscale devices appear promising, though several reliability challenges need to be addressed together with the optimization of the algorithms used in these applications. Advances in new material systems, architectures that enable arbitrary connectivity patterns, and algorithmic principles for spike-based information encoding and processing will be required to bridge the substantial gap in efficiency between today's neuromorphic processors and the human brain.

ACKNOWLEDGMENT

We gratefully acknowledge the collaborators who were at IBM T. J. Watson Research Center, I.I.T. Bombay, Ecole Centrale Lyon and University of Toledo who contributed to the research projects discussed in this paper. This work was supported in part by a grant from CISCO and the Semiconductor Research Corporation.

REFERENCES

[1] J. von Neumann, "First draft of a report on the EDVAC," *IEEE Ann. Hist. Comput.*, vol. 15, no. 4, pp. 27–75, Oct. 1993. [Online]. Available: http://dx.doi.org/10.1109/85.238389

[2] D. Patterson, T. Anderson, N. Cardwell, R. Fromm, K. Keeton, C. Kozyrakis, R. Thomas, and K. Yelick, "A case for intelligent RAM," *Micro, IEEE*, vol. 17, no. 2, pp. 34–44, Mar 1997.

[3] B. Rajendran, R. Shenoy, D. Witte, N. Chokshi, R. De Leon, G. Tompa, and R. Fabian, "Low thermal budget processing for sequential 3-D IC fabrication," *Electron Devices, IEEE Transactions on*, vol. 54, no. 4, pp. 707 –714, April 2007.

[4] G. W. Burr, B. N. Kurdi, J. C. Scott, C. H. Lam, K. Gopalakrishnan, and R. S. Shenoy, "Overview of candidate device technologies for storage-class memory," *IBM Journal of Research and Development*, vol. 52, no. 4.5, pp. 449–464, july 2008.

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

[6] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's neural machine translation system: Bridging the gap between human and machine translation," *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information and Processing Systems (NIPS)*, 2012, pp. 1097–1105.

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

[10] D. Purves, G. J. Augustine, and D. Fitzpatrick, Eds., *Neuroscience*. Sunderland: Sinauer Associates, 2001.

[11] K. Boahen, "Neuromorphic microchips," *Scientific American*, vol. 292, no. 5, pp. 56–63, 2005.

[12] S. Panzeri, N. Brunel, N. K. Logothetis, and C. Kayser, "Sensory neural codes using multiplexed temporal scales," *Trends in neurosciences*, vol. 33, no. 3, pp. 111–120, 2010.

[13] 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.

[14] J. Gehlhaar, "Neuromorphic processing: A new frontier in scaling computer architecture," in *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2014, pp. 317–318. [Online]. Available: http://doi.acm.org/10.1145/2541940.2564710

[15] N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, and G. Indiveri, "A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses," *Frontiers in Neuroscience*, vol. 9, p. 141, 2015.

[16] L. F. Abbott, "Lapicques introduction of the integrate-and-fire model neuron (1907)," *Brain research bulletin*, vol. 50, no. 5, pp. 303–304, 1999.

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

[18] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[19] G. Q. Bi and M. M. Poo, "Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, pp. 10 464–10 472, Dec 1998.

[20] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, vol. 9, no. 99, 2015.

[21] J. M. Allred and K. Roy, "Unsupervised incremental stdp learning using forced firing of dormant or idle neurons," in *2016 International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 2492–2499.

[22] A. A. Lazar, E. K. Simonyi, and L. T. Tóth, "Time encoding of bandlimited signals, an overview," in *Proceedings of Conference on Telecommunication Systems, Modeling and Analysis*. Citeseer, 2005.

[23] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1, pp. 17–37, 2002.

[24] O. Booij and H. tat Nguyen, "A gradient descent rule for spiking neurons emitting multiple spikes," *Information Processing Letters*, vol. 95, no. 6, pp. 552–558, 2005.

[25] S. McKennoch, D. Liu, and L. G. Bushnell, "Fast modifications of the spikeprop algorithm," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006, pp. 3970–3977.

[26] A. Mohammed, S. Schlibebs, S. Matsuda, and N. Kasabov, "SPAN: Spike pattern association neuron for learning spatio-temporal spike patterns," *International Journal of Neural Systems*, vol. 22, no. 04, p. 1250012, 2012.

[27] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting," *Neural Computation*, vol. 22, no. 2, pp. 467–510, Feb. 2010.

[28] Q. Yu, H. Tang, K. C. Tan, and H. Li, "Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns," *PLoS ONE*, vol. 8, no. 11, p. e78318, 11 2013.

[29] A. Taherkhani, A. Belatreche, Y. Li, and L. P. Maguire, "Dl-resume: A delay learning-based remote supervised method for spiking neurons," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 12, pp. 3137–3149, Dec 2015.

[30] S. Kulkarni and B. Rajendran, "Supervised learning with spiking neural networks on a GPU platform," *unpublished*, 2017.

[31] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers in Neuroscience*, vol. 10, p. 508, 2016.

[32] P. U. Diehl, D. Neil, J. Binas, M. Cook, S.-C. Liu, and M. Pfeiffer, "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–8.

[33] E. Hunsberger and C. Eliasmith, "Training spiking deep networks for neuromorphic hardware," *arXiv preprint arXiv:1611.05141*, 2016.

[34] B. Rueckauer, I.-A. Lungu, Y. Hu, and M. Pfeiffer, "Theory and tools for the conversion of analog to spiking convolutional neural networks," *arXiv preprint arXiv:1612.04052*, 2016.

[35] R. Brette and D. F. Goodman, "Simulating spiking neural networks on GPU," *Network: Computation in Neural Systems*, 2012.

[36] F. M.Kraft and A.Kasinski, "Fpga implementation of the resume learning method for snn," in *Proceedings of EPFL LATSIS Symposium 2006, Dynamical Principles for Neuroscience and Intelligent Biomimetic Devices*, 2006, pp. 97–98.

[37] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker project," *Proceedings of the IEEE*, vol. 102, no. 5, pp. 652–665, May 2014.

[38] J. M. Nageswaran *et al.*, "A configurable simulation environment for the efficient simulation of large-scale spiking neural networks on graphics processors," *Neural networks*, vol. 22, no. 5, pp. 791–800, 2009.

[39] A. K. Fidjeland and M. P. Shanahan, "Accelerated simulation of spiking neural networks using GPUs," in *The 2010 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2010, pp. 1–8.

[40] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong, "Optimizing FPGA-based accelerator design for deep convolutional neural networks," in *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '15. New York, NY, USA: ACM, 2015, pp. 161–170.

[41] G. Indiveri, B. Linares-Barranco, T. J. Hamilton, A. van Schaik, R. Etienne-Cummings, T. Delbruck, S.-C. Liu, P. Dudek, P. Hafliger, S. Renaud, J. Schemmel, G. Cauwenberghs, J. Arthur, K. Hynna, F. Folowosele, S. Saighi, T. Serrano-Gotarredona, J. Wijekoon, Y. Wang, and K. Boahen, "Neuromorphic silicon neuron circuits," *Frontiers in Neuroscience*, vol. 5, 2011.

[42] V. Ostwal, R. Meshram, B. Rajendran, and U. Ganguly, "An ultra-compact and low power neuron based on SOI platform," in *VLSI Technology, Systems and Application (VLSI-TSA), 2015 International Symposium on*, April 2015, pp. 1–2.

[43] T. Tuma, A. Pantazi, M. L. Gallo, A. Sebastian, and E. Eleftheriou, "Stochastic phase-change neurons," *Nature Nanotechnology*, vol. 11, pp. 693–600, May 2016.

[44] 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 *2016 74th Annual Device Research Conference (DRC)*, June 2016, pp. 1–2.

[45] J. S. Seo, B. Brezzo, Y. Liu, B. D. Parker, S. K. Esser, R. K. Montoye, B. Rajendran, J. A. Tierno, L. Chang, D. S. Modha, and D. J. Friedman, "A 45nm CMOS neuromorphic chip with a scalable architecture for learning in networks of spiking neurons," in *Custom Integrated Circuits Conference (CICC), 2011 IEEE*, Sept 2011, pp. 1–4.

[46] G. W. Burr, R. M. Shelby, A. Sebastian, S. Kim, S. Kim, S. Sidler, K. Virwani, M. Ishii, P. Narayanan, A. Fumarola, L. L. Sanches, I. Boybat, M. L. Gallo, K. Moon, J. Woo, H. Hwang, and Y. Leblebici, "Neuromorphic computing using non-volatile memory," *Advances in Physics: X*, vol. 2, no. 1, pp. 89–124, 2017.

[47] D. Kuzum, R. G. D. Jeyasingh, B. Lee, and H.-S. P. Wong, "Nano-electronic programmable synapses based on phase change materials for brain-inspired computing," *Nano Letters*, vol. 12, no. 5, pp. 2179–2186, 2012.

[48] B. L. Jackson, B. Rajendran, G. S. Corrado, M. Breitwisch, G. W. Burr, R. Cheek, K. Gopalakrishnan, S. Raoux, C. T. Rettner, A. Padilla, A. G. Schrott, R. S. Shenoy, B. N. Kurdi, C. H. Lam, and D. S. Modha, "Nano-scale electronic synapses using phase change devices," in *ACM Journal of Emerging Technologies for Computing*, vol. 9, no. 2, Sept. 2013, pp. 12:1–12:20.

[49] 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, pp. 135–136.

[50] S. Mandal, A. El-Amin, K. Alexander, B. Rajendran, and R. Jha, "Novel synaptic memory device for neuromorphic computing," *Nature Scientific Reports*, vol. 4, no. 5333, 2014.

[51] S. Ambrogio, N. Ciocchini, M. Laudato, V. Milo, A. Pirovano, P. Fantini, and D. Ielmini, "Unsupervised learning by spike timing dependent plasticity in phase change memory (pcm) synapses," *Frontiers in neuroscience*, vol. 10, 2016.

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

[53] F. Alibart, S. Pleutin, O. Bichler, C. Gamrat, T. Serrano-Gotarredona, B. Linares-Barranco, and D. Vuillaume, "A memristive nanoparticle/organic hybrid synapstor for neuro-inspired computing," *Advanced Functional Materials*, vol. 22, pp. 609–616, 2011.

[54] S. Yu, Y. Wu, R. Jeyasingh, D. Kuzum, and H.-S. Wong, "An electronic synapse device based on metal oxide resistive switching memory for neuromorphic computation," *Electron Devices, IEEE Transactions on*, vol. 58, no. 8, pp. 2729–2737, Aug 2011.

[55] K. Moon, E. Cha, D. Lee, J. Jang, J. Park, and H. Hwang, "Reram-based analog synapse and imt neuron device for neuromorphic system," in *2016 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*, April 2016, pp. 1–2.

[56] J. Lin, Annadi, S. Sonde, C. Chen, L. Stan, K. V. L. V. Achari, S. Ramanathan, and S. Guha, "Low-voltage artificial neuron using feedback engineered insulator-to-metal-transition devices," in *IEEE International Electron Devices Meeting*, 2016, pp. 34.5.1–34.5.4.

[57] M. Suri, O. Bichler, D. Querlioz, O. Cueto, L. Perniola, V. Sousa, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Phase change memory as synapse for ultra-dense neuromorphic systems: Application to complex visual pattern extraction," in *Electron Devices Meeting (IEDM), 2011 IEEE International*, Dec 2011, pp. 4.4.1–4.4.4.

[58] E. Covi, S. Brivio, A. Serb, T. Prodromakis, M. Fanciulli, and S. Spiga, "Analog memristive synapse in spiking networks implementing unsupervised learning," *Frontiers in Neuroscience*, vol. 10, p. 482, 2016. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2016.00482

[59] 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 *Electron Devices Meeting (IEDM), 2014 IEEE International*, 2014, pp. 29.5.1–29.5.4.

[60] 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.

[61] F. Ponulak and A. Kasinski, "Resume learning method for spiking neural networks dedicated to neuroprostheses control," in *Proceedings of EPFL LATSIS Symposium 2006, Dynamical Principles for Neuroscience and Intelligent Biomimetic Devices*. Citeseer, 2006, pp. 119–120.

[62] N. K. Kasabov, "Neucube: A spiking neural network architecture for mapping, learning and understanding of spatio-temporal brain data," *Neural Networks*, vol. 52, pp. 62–76, 2014.

[63] C. Ge, N. Kasabov, Z. Liu, and J. Yang, "A spiking neural network model for obstacle avoidance in simulated prosthetic vision," *Information Sciences*, vol. 399, pp. 30–42, 2017.

[64] T. Hwu, A. Wang, N. Oros, and J. Krichmar, "Adaptive robot path planning using a spiking neuron algorithm with axonal delays," *IEEE Transactions on Cognitive and Developmental Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[65] D. Reid, A. J. Hussain, and H. Tawfik, "Financial time series prediction using spiking neural networks," *PloS one*, vol. 9, no. 8, p. e103656, 2014.

[66] W. W. Lee, S. L. Kukreja, and N. V. Thakor, "Cone: Convex-optimized-synaptic efficacies for temporally precise spike mapping," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 4, pp. 849–861, April 2017.

[67] C. Prasad, K. Saboo, and B. Rajendran, "Composer classification based on temporal coding in adaptive spiking neural networks," in *Neural Networks (IJCNN), 2015 International Joint Conference on*, July 2015.

[68] A. Bora, A. Rao, and B. Rajendran, "Mimicking the worm - an adaptive spiking neural circuit for contour tracking inspired by C. elegans thermotaxis," in *Neural Networks (IJCNN), 2014 International Joint Conference on*, July 2014, pp. 2079–2086.

[69] C. Shetty, S. Nitchith, R. Rawat, S. Nandakumar, P. Shah, S. Kulkarni, and B. Rajendran, "Live demonstration: Spiking neural circuit based navigation inspired by c. elegans thermotaxis," in *Circuits and Systems (ISCAS), 2015 IEEE International Symposium on*, May 2015, pp. 1905–1905.

[70] P. Tandon, Y. Malviya, and B. Rajendran, "Efficient and robust spiking neural circuit for navigation inspired by echolocating bats," in *Neural Information Processing Systems*, 2016.