Impact of Increasing Number of Neurons on Performance of Neuromorphic Architecture

Mahyar Shahsavari*, Pierre Boulet*, Asadollah Shahbahrami†, Said Hamdioui‡
*Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL - Centre de Recherche
en Informatique Signal et Automatique de Lille, F-59000 Lille, France
†Department of Computer Engineering, Faculty of Engineering, University of Giulan, Rasht, Iran
‡ Computer Engineering, Delft University of Technology, Delft, The Netherlands
Email: Mahyar.Shahsavari@ed.univ-lille1.fr, Pierre.Boulet@univ-lille1.fr,
Shahbahrami@guilan.ac.ir, S.Hamdioui@tudelft.nl

***Abstract*—Pattern recognition is used to classify the input data into different classes based on extracted key features. Increasing the recognition rate of pattern recognition applications is a challenging task. The spike neural networks inspired from physiological brain architecture, is a neuromorphic hardware implementation of network of neurons. A sample of neuromorphic architecture has two layers of neurons, input and output. The number of input neurons is fixed based on the input data patterns. While the number of outputs neurons can be different. The goal of this paper is performance evaluation of neuromorphic architecture in terms of recognition rates using different numbers of output neurons. For this purpose a simulation environment of N2S3 and MNIST handwritten digits are used. Our simulation results show the recognition rate for various number of output neurons, 20, 30, 50, 100, 200, and 300 is 70%, 74%, 79%, 85%, 89%, and 91%, respectively.**

I. Introduction

One of the key factors for evaluating the performance of Artificial Neural Network (ANN) is the ability of pattern recognition known as recognition rate. Choosing the proper number of neurons to reach optimum recognition rate, is one of the major issues that researchers in ANN domain are facing [1]. In machine learning approaches, not only an efficient processing algorithm is necessary but also finding a fast, parallel and power-efficient hardware architecture to implement these algorithms is unavoidable [2]. Neuromorphic architectures are cognitive computing platforms to process data to classify into different classes based on extracted key features from data [3], [4], [5]. Spiking Neural Network (SNN) is a promising approach for designing an electronic hardware neuromorphic model of biological brain. However, increasing the classification rate of pattern recognition applications in SNN neuromorphic platform is a challenging task.

There are several factors affect the recognition capability of neural network such as different learning algorithms, activation function of neuron, network connection topologies, training samples, the number of layers in network and the number of neurons. The random selection of number of neurons might lead to either overfitting or underfitting problems [1]. The approximation of number of neurons in hidden layer to achieve better performance is introduced in previous research in artificial neural networks [1], [6], [7]. However in Spiking Neural Network, there are few works that discussed this issue. Tavanaei and Maida [8] presented the minimal number of neurons in SNN for pattern classification of MNIST dataset of handwritten digits using 100 Izhikevich [9] spiking neurons. In [8] the number of neuron was fixed and recognition rate with different iterations of presenting samples was explored. Parameter exploration to improve the performance of SNN-based neuromorphic architecture is presented in [10]. The impact of number of neurons on network rate recognition is explored in this research. However changing the number of neurons is aligned with changing the other parameters such as STDP window, neuron threshold and distribution of input spikes.

The goal of this paper is to evaluate the recognition performance of SNN-based architecture with increasing the number of neurons in the output layer. We use a spiking neural network hardware model, as a neuromorphic architecture for classification of handwritten digits. For this purpose, a SNN-based neuromorphic architecture with two layers of neurons which are fully connected from first to second layer is used. A threshold-based model of spiking neuron is used known as *Leaky-Integrate-and-Fire (LIF)* neuron [11]. We used spike-timing-dependent plasticity (STDP) unsupervised learning method for adjusting the synaptic weights. In this hardware simulation, we consider the model of memristor [12], [13] as an artificial synapse. Unique properties in memristor, such as scalability, flexibility because of their analog behavior, manufacturability on top of CMOS technology and ability to remember the last state make this nanodevice a proper alternative to play the role of artificial synapse [14]. We used Neural Network Scalable Spiking Simulator (N2S3) [15], a simulation framework for architecture exploration of neuromorphic platforms. Our simulation results on MNIST handwritten data set show that using different number of output neurons from 20 to 300 leads to different recognition rate ranging from 70 to 91%. It means that increasing the number of output neurons increases the recognition rate.

This paper is organized as follows. Neuromorphic and memristor concepts are discussed in Section 2. The architecture of neural network and learning algorithm are presented in Section 3. Section 4 presents experimental results. Finally conclusions

are drawn in Section 5.

II. Neuromorphic and Memristor

The first neuromorphic term was coined by Carver Mead [16] using Very Large Scale Integration (VLSI) technology to propose an implementation of neural system hardware. Merolla et al. [17] in an IBM research was sponsored by DARPA, have demonstrated a computing hardware consist of the compact modular core for large-scale neuromorphic system architecture. The cores combine digital neurons with the large synaptic array. This general purpose neuromorphic processor was built using thousands of neurosynaptic cores are involved one million neurons and 256 million of reconfigurable synapses. SpiNNaker project [18] aims to deliver a massively parallel million core architectures whose interconnections are inspired by the connectivity properties of the mammalian brain. The hardware platform is suitable to model the large-scale spiking neural networks in biological real time. Neuromorphic and neuro-inspired computing is now being adapted by an increasing number of academic and industrial different research teams. In recent few years, there have been many valuable publications explaining the use of novel materials such as memristors which are able to emulate some of the properties observed in biological synapses [19], [20], [21], [22], [23].

Recently, emerging nano-scale devices such as memristors have demonstrated novel properties for making new memories and unconventional processing units. Memristor was hypothetically presented by Leon Chua in 1971 [12] and after few decades, HP was the first to announce the successful fabrication of memristor [13]. The plasticity characteristic in Memristor open the new windows toward researchers in neural network and artificial intelligence domain to continue their research studies with more confident. The unique properties in memristor nano-devices such as, extreme scalability, flexibility, and ability to remember the last state make the memristor a very promising candidate to be applied as a synapse in Artificial Neural Network (ANN) [14]. Recent advances in nanotechnology have provided neuromorphic computing architecture with novel memristive devices which have the capability of mimicking synaptic plasticity, such as resistive switching memory (RRAM) [24], [25], [26], phase change memory (PCM) [27], [28], Conductive Bridge memory (CBRAM) [29], [30], [31], and ferroelectric memory (FeRAM) [32], [33].

III. Neuromorphic Architecture and Learning Algorithm

In this section neuromorphic architecture, model of neuron and learning algorithms are presented.

A. Neuromorphic Architecture

A sample of neuromorphic architecture is depicted in Figure 1 which depicted two layers of input and output neurons. These layers are connected using artificial synapses which are called memristors. Input neurons received spikes which are a representative of the image pixel intensities. These spikes are connected to the neuron units to fire toward the memristors.

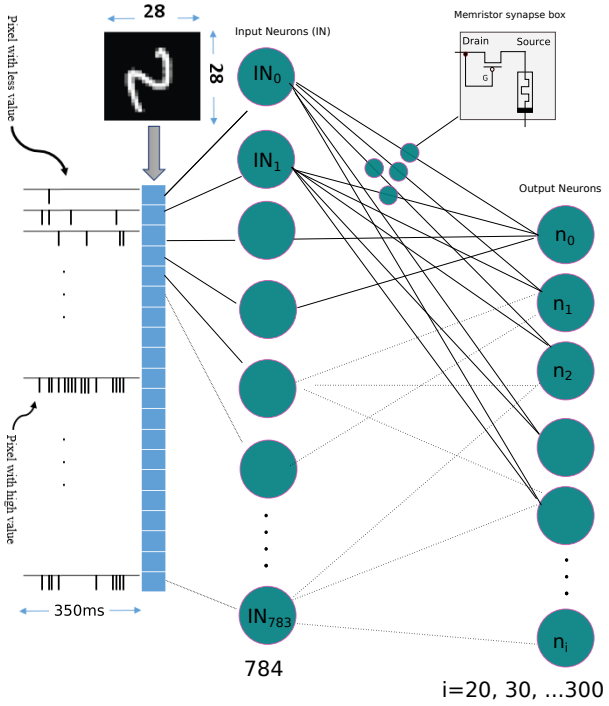

Fig. 1. Each pixel is transfered to spikes based on its intensity, therefore we have 784 pixel connected to the input neurons.

In other words, neuron integrates the spike inputs from other neurons that are connected to it. These input spikes change the internal potential of the neuron, it is known as neuron's membrane potential or state variable. When this membrane potential passes a threshold voltage due to integrated inputs, the action potential occurs. The function of the neurons is depicted in Equation 1 and 2.

$$\tau_n \frac{\mathrm{d}v}{\mathrm{d}t} = -v(t) + RI_{syn}(t) \tag{1}$$

$$I_{syn}(t) = \sum_j g_{ij} \sum_n \alpha(t - t_j^{(n)}) \tag{2}$$

where $v(t)$ represents the membrane potential at time t, $\tau_n = RC$ is the membrane time constant and R is the membrane resistance. Equation 1 describes a simple parallel resistor-capacitor (RC) circuit where the leakage term is due to the resistor and the integration of $I_{syn}(t)$ is due to the capacitor. $I_{syn}(t)$ is generated by the activity of pre-synaptic neurons. In fact, each pre-synaptic spike generates a post-synaptic current pulse. The total input current to a neuron is the sum over all current pulses which is calculated in Equation 2, $t_j^{(n)}$ represents the time of the n_{th} spike of post-synaptic neuron j and g_{ij} is the conductance of synaptic efficacy between neuron i and neuron j. $\alpha(t) = q\delta(t)$, where q is injected charge to the the artificial synapse and $\delta(t)$ is Dirac pulse function. If $I_{syn}(t)$ is big enough where action potential

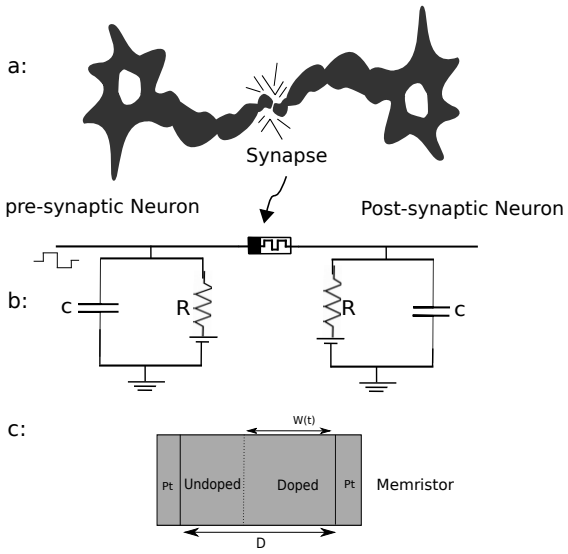

Fig. 2. **a)** A synapse connects two neurons. **b)** A simple electronic model of a, which neurons are modeled using a capacitor and resistor and synapse is modeled using a memristor. **c)** A simple physical model of a memristor.

can pass the threshold voltage, neuron fires. It means there are enough input spikes in a short time window. When there is no or few spikes in a time window the neuron is in leaky phase and state variable decreases exponentially. This spiking model of neuron is known as *Leaky-Integrate-and-Fire (LIF)* model [11]. The duration of this time window is depend on $\tau_n = RC$. The equation is analytically solvable however we apply the final solution answer of equation in large networks with presenting input spike. It improves the delay time and network performance as we do not consist extra time for solving the equation numerically. The fabricated model with recent CMOS technology is available for LIF neuron [34], [35]. Two biological neuron with a simple synaptic connection between neurons are shown in Figure 2.a. The first electrical model of neuron is known as Hodgkin-Huxley neuron model [36] which is depicted in Figure 2.b. In fact, LIF model of neuron is derived from the Hodgkin-Huxley neuron model.

Memristor adjusts the weight between two input and output neurons. By changing the doped-undoped regions of device, the conductance will be changed. Bigger doped region leads to more conductivity. Therefore by controlling of this boundary between two regions, the conductivity is controlled which is depicted in Figure 2.c. In this work, we used memristive synapse box as a connection between two neurons. More details about memristive synapse box can be found in [37], [38].

The output neurons of the neuromorphic architecture in Figure 1, receive spikes from memristors and process them to make a decision for firing or not. After unsupervised training, the output of these neurons are labeled to different classes for testing phase. Finally, the evaluation of impacts of increasing different number of output neurons on recognition rate are performed.

B. Learning Algorithm

We used a fully connected network architecture which means each neuron in input layer is connected to the all output neurons. It is widely believed that plasticity is the key for learning in the biological brain [39]. With the latest proposals to use memristive nano-devices as synapses, we implement efficient unsupervised learning rules known as Spike Timing Dependent Plasticity (STDP) [40]. By using unsupervised learning inspired from biological neural network, we apply STDP learning method for adjusting the synaptic weights [40], [41]. If there is an output spike in pre-synaptic neuron and shortly after in post-synaptic neuron, the conductance of the memristor between two neurons increases. On the other hand, if the post-synaptic neuron spikes and shortly after the pre-synaptic neuron spikes, the conductance of memristor between two neurons decreases. Additionally, inspiring from biologic behavior of brain computing, we apply lateral inhibition to reduce the activity of neighbor neurons of the winner. This method is known as winner-takes-all strategy [42]. The neuron which reaches the threshold first, sends an inhibitory signal to all other neurons in the same layer to reset their states during inhibition time. This neuron called winner in the firing competition.

IV. Experimental Results

In this section, our data set, simulation environment and results are presented.

A. Datasets

To train and test the neural network, the MNIST dataset of handwritten digits [43] is used for experimental evaluations. MNIST set consists of 60000 digits between 0 to 9. Each handwritten number image is 28 $\times$ 28 =784 pixels. In this simulation, we present full dataset (60000) and full image (28 $\times$ 28) to the input of neural network. Each pixel is connected to one input buffer neuron to receive the intensity of pixel as spikes (see Figure 1). To transfer dataset of handwritten digit pixel intensity to spikes train, we tried several spike distributions such as Poission, Uniform, and Gaussian. Pixel intensity is ranging from 0 to 255 that are transfered to 0 to 22 Hertz of spiking rate frequency. The duration of presenting a sample input to neural network is 350 ms. Based on previous similar work [44], 150 ms is considered as an interval time between presenting two sample inputs. Therefore, the membrane potentials of the neurons have enough time to be reseted to initial value. For training phase unsupervised learning has been used. The network connection weights are between zero and one. For weight initialization, random weights are applied using Gaussian distribution. Among 60000 samples, 50000 samples are used for training phase and 10000 samples are used for testing phase. In order to validate testing results simulation is executed ten times and the median results are presented.

Output labels for digit classification based on number of firing for each neuron (n=20)

	0	1	2	3	4	5	6	7	8	9	Total
N0:0	0	1	0	2	51	18	0	37	2	187	187
N0:1	1	0	2	2	0	0	0	395	0	1	395
N0:2	0	143	0	0	2	1	2	11	1	2	333
N0:3	0	0	0	0	133	4	0	0	0	2	133
N0:4	22	0	3	12	1	150	8	0	6	1	150
N0:5	0	7	1	2	2	7	0	4	96	1	96
N0:6	0	1	473	5	0	0	0	6	4	0	473
N0:7	5	1	27	51	2	15	0	11	286	6	396
N0:8	160	0	6	0	0	0	0	0	0	2	160
N0:9	0	0	164	2	0	0	0	4	0	0	164
N0:10	11	2	10	0	5	4	499	1	1	2	499
N0:11	6	0	1	0	4	1	223	0	0	0	223
N0:12	0	0	0	154	0	0	0	0	0	0	154
N0:13	0	0	0	0	1	0	0	157	1	12	157
N0:14	3	0	39	2	264	19	20	39	4	112	264
N0:15	0	724	16	26	8	23	17	40	21	15	724
N0:16	33	0	0	9	0	154	6	0	10	0	154
N0:17	21	0	12	414	0	157	3	1	75	4	414
N0:18	0	2	2	20	262	41	1	92	13	447	447
N0:19	519	0	2	5	0	6	9	1	0	0	519
Total	781	1071	758	706	735	600	788	799	630	794	

Fig. 3. Each neuron has been labeled a class from zero to nine based on the maximum firing rate in configuration with 20 output neurons.

Output labels for digit classification based on number of firing for each neuron (n=30)

	0	1	2	3	4	5	6	7	8	9	Total
N0:0	0	0	2	5	170	12	1	80	9	263	263
N0:1	0	0	3	2	0	0	0	202	2	21	202
N0:2	2	0	20	0	234	14	6	22	9	112	234
N0:3	1	0	1	1	0	2	0	228	2	2	228
N0:4	0	0	149	2	0	0	0	1	2	0	149
N0:5	1	1	2	1	1	1	0	1	99	0	99
N0:6	1	0	0	7	0	242	5	0	11	8	242
N0:7	8	0	13	145	3	137	5	2	151	5	151
N0:8	5	2	4	0	8	1	390	1	2	2	390
N0:9	2	1	1	0	4	1	249	0	4	0	249
N0:10	0	0	0	0	130	0	0	0	1	65	130
N0:11	0	263	1	0	2	0	1	7	1	2	263
N0:12	0	0	0	0	164	3	0	1	0	27	164
N0:13	164	0	6	0	0	0	2	0	0	1	164
N0:14	0	1	0	263	0	6	0	2	28	3	263
N0:15	2	0	0	0	17	4	202	0	4	0	202
N0:16	0	2	388	7	4	1	0	16	3	1	388
N0:17	0	0	0	10	0	203	2	0	15	1	203
N0:18	383	0	24	6	1	5	23	2	13	8	383
N0:19	2	0	12	23	0	10	2	5	479	9	479
N0:20	0	1	312	9	0	1	1	4	11	0	312
N0:21	0	0	1	115	0	0	0	0	3	0	115
N0:22	0	0	0	1	27	0	0	3	2	143	143
N0:23	0	0	0	1	0	1	0	283	1	10	283
N0:24	0	285	1	1	3	0	0	10	0	1	285
N0:25	1	0	0	8	140	18	2	37	15	238	238
N0:26	67	0	5	22	1	34	7	0	9	0	67
N0:27	325	0	3	2	0	5	12	0	3	0	325
N0:28	0	425	2	7	5	5	1	8	5	7	425
N0:29	1	0	3	298	0	55	1	0	10	1	298
Total	965	981	953	936	914	761	912	915	894	930	

Fig. 4. Each neuron has been labeled a class from zero to nine based on the maximum firing rate in configuration with 30 output neurons.

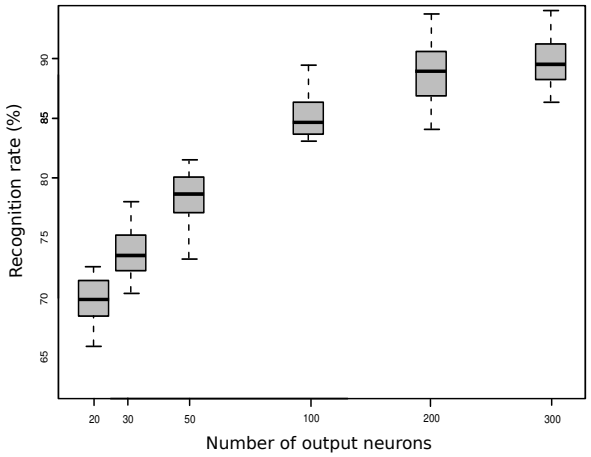

Fig. 5. The recognition rate of different output neurons from 20 to 300. For each number the simulation is run ten times.

B. Simulation Environment

In order to simulate the spiking neural network, we used a Neural Network Scalable Spiking Simulator (N2S3), a simulation framework for architecture exploration of neuromorphic architecture (https://sourcesup.renater.fr/projects/n2s3) [15]. This simulation platform is event-driven, concurrent, scalable, and adaptable to model different synapses, neurons, and topologies.

The main reason for using N2S3 simulator in this work, is that other popular neural network simulators such as Neuron [45], Brian [46] or NEST [47] are not fit our purposes for this implementation appropriately . They are clock-driven and the model of memristor as a synapse is not considered. Xnet [48] is an event-driven simulator but it is not available to use publicly as well as its capabilities do not address some of our requirements such as scalability and concurrency. The hardware platform is an Intel cores i7-3687U CPU (2.10GHz $\times$ 4).

C. Simulation Results

The number of input neurons is fixed to 784, while the number of output neurons is varying from 20, 30, 50, 100, 200, to 300.

First, the number of output neurons was set to 20. Labeling results of twenty neurons is depicted in Figure 3. Obviously in this figure, there are 20 rows for output neurons and ten columns for different classes. A suitable class should be assigned to each neuron. For this purpose, most frequent firing rate of each row is labeled to the column numbers. For instance, the most frequent firing rate for the neuron number six (N0:6) is 473, that assigned to label two. Each label is assigned to two neurons. The average of recognition rate is 70%.

Second, the number of output neurons was set to thirty. The output of testing phases is depicted in Figure 4. Each output neuron should be labeled to a class. The class assignment is the same as Figure 3 while the assignment is not symmetric. For example four neurons, N0:13, 18, 26, and 27 are labeled to class zero. The reason is the frequent firing rate for those neurons in column zero is higher than other columns. Class five has two neurons, N0:6, 17, while other classes has three neurons. The average of recognition rate is 74%. Compare to the Figure 3, recognition rate is increased. It means that increasing the number of output neurons leads to increasing the recognition rate.

Finally , the number of neurons in the output layer of the spiking neural network architecture has also been set to 50, 100, 200, and 300. Recognition rate of different output neurons is depicted in Figure 5.

As this figure shows, the recognition rate is increased by the increasing of the number of output neurons. This recognition

Fig. 6. Recognized classes by spiking neural network with different output neurons, 20, 30, 50 and 100.

rate is ranging from 70% to 91% for 20 and 300 neurons, respectively. Increasing the recognition rate from 20 to 100 neurons is much higher than 100 to 300 neurons. It means that there is a limitation on increasing the number of output neurons due to the overfitting issue. Figure 6 depicts the assigned classes to different numbers of output neurons from 20 to 100.

V. Conclusions

Recently hardware architectures based on neural networks are used for pattern recognition applications. One of these architecture is Spiking neuromorphic platform. A sample of neuromorphic architecture has two layers of neurons. These layers are connected using memristor synapse box. Memristor conductance adjusts the weights between the neurons using spikes. Recognition rate of neuromorphic architecture was evaluated using different number of output neurons. Our simulation results using MNIST handwritten data set show that increasing the number output neurons increases the recognition rate. In addition, results show that increasing recognition rate from 20 to 100 in much higher than 100 to 300 neurons. Increasing the number of output neurons more than 300, the architecture is overfitted.

Acknowledgments

This work has been supported by European Network on High Performance and Embedded Architecture and Compilation (HiPEAC) in collaboration grant agreement H2020-687698.

References

[1] K. G. Sheela and S. N. Deepa, "Review on Methods to Fix Number of Hidden Neurons in Neural Networks," 2013. DOI: 10.1155/2013/425740.

[2] C. D. James, J. B. Aimone, N. E. Miner, C. M. Vineyard, F. H. Rothganger, K. D. Carlson, S. A. Mulder, T. J. Draelos, A. Faust, M. J. Marinella, J. H. Naegle, and S. J. Plimpton, "A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications," *Biologically Inspired Cognitive Architectures*, vol. 19, pp. 49–64, Jan. 2017.

[3] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in Computational Neuroscience*, p. 99, 2015.

[4] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning*, 2007.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998.

[6] Y. Liu, J. A. Starzyk, and Z. Zhu, "Optimizing Number of Hidden Neurons in Neural Networks," ACTA Press, Jan. 2007.

[7] D. Hunter, H. Yu, M. S. P. III, J. Kolbusz, and B. M. Wilamowski, "Selection of Proper Neural Network Sizes and Architectures #x2014;A Comparative Study," *IEEE Transactions on Industrial Informatics*, vol. 8, pp. 228–240, May 2012.

[8] A. Tavanaei and A. S. Maida, "A Minimal Spiking Neural Network to Rapidly Train and Classify Handwritten Digits in Binary and 10-Digit Tasks," *International Journal of Advanced Research in Artificial Intelligence (ijarai)*, vol. 4, no. 7, 2015.

[9] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 14, no. 6, pp. 1569–1572, 2003.

[10] M. Shahsavari and P. Boulet, "Parameter Exploration to Improve Performance of Memristor-Based Neuromorphic Architectures," *IEEE Transactions on Multi-Scale Computing Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[11] W. Maass and C. M. Bishop, eds., *Pulsed Neural Networks*. Cambridge, MA, USA: MIT Press, 1999.

[12] L. Chua, "Memristor-The missing circuit element," *Circuit Theory, IEEE Transactions on*, vol. 18, pp. 507–519, Sept. 1971.

[13] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008.

[14] 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. PMID: 20192230.

[15] M. Shahsavari, P. Devienne, and P. Boulet, "N2s3, a Simulator for the Architecture Exploration of Neuromorphic Accelerators," in *2nd International Workshop on Neuromorphic and Brain-Based Computing Systems (NeuComp 2015) in DATE Conference, Grenoble, France*, 2015.

[16] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, pp. 1629–1636, Oct. 1990.

[17] 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, pp. 668–673, Aug. 2014.

[18] S. B. Furber, F. Galluppi, S. Temple, and L. A. Plana, "The SpiNNaker Project," *Proceedings of the IEEE*, vol. 102, pp. 652–665, May 2014.

[19] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale Memristor Device as Synapse in Neuromorphic Systems," *Nano Lett.*, vol. 10, pp. 1297–1301, Mar. 2010.

[20] B. Rajendran and F. Alibart, "Neuromorphic Computing Based on Emerging Memory Technologies," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, pp. 198–211, June 2016.

[21] L. Shi, J. Pei, N. Deng, D. Wang, L. Deng, Y. Wang, Y. Zhang, F. Chen, M. Zhao, S. Song, F. Zeng, G. Li, H. Li, and C. Ma, "Development of a neuromorphic computing system," in *2015 IEEE International Electron Devices Meeting (IEDM)*, pp. 4.3.1–4.3.4, Dec. 2015.

[22] P. O'Connor, D. Neil, S.-C. Liu, T. Delbruck, and M. Pfeiffer, "Real-time classification and sensor fusion with a spiking deep belief network," *Neuromorphic Engineering*, vol. 7, p. 178, 2013.

[23] B. V. Benjamin, P. Gao, E. McQuinn, S. Choudhary, A. R. Chandrasekaran, J. M. Bussat, R. Alvarez-Icaza, J. V. Arthur, P. A. Merolla, and K. Boahen, "Neurogrid: A Mixed-Analog-Digital Multichip System for Large-Scale Neural Simulations," *Proceedings of the IEEE*, vol. 102, pp. 699–716, May 2014.

[24] J. Yang, M. Pickett, X. Li, D. Ohlberg, D. Stewart, and R. Williams, "Memristive switching mechanism for metal/oxide/metal nanodevices," *Nature Nanotechnology*, vol. 3, no. 7, pp. 429–433, 2008.

[25] C. Xu, X. Dong, N. P. Jouppi, and Y. Xie, "Design implications of memristor-based rram cross-point structures.," in *DATE*, pp. 734–739, IEEE, 2011.

[26] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. Adam, K. K. Likharev, and D. B. Strukov, "Training and Operation of an Integrated Neuromorphic Network Based on Metal-Oxide Memristors," *Nature*, vol. 521, pp. 61–64, May 2015. arXiv: 1412.0611.

[27] S. B. Eryilmaz, D. Kuzum, R. Jeyasingh, S. Kim, M. BrightSky, C. Lam, and H.-S. P. Wong, "Brain-like associative learning using a nanoscale non-volatile phase change synaptic device array," *Neuromorphic Engineering*, vol. 8, p. 205, 2014.

[28] 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," *Neuromorphic Engineering*, p. 56, 2016.

[29] Q. Liu, S. Long, H. Lv, W. Wang, J. Niu, Z. Huo, J. Chen, and M. Liu, "Controllable Growth of Nanoscale Conductive Filaments in Solid-Electrolyte-Based ReRAM by Using a Metal Nanocrystal Covered Bottom Electrode," *ACS Nano*, vol. 4, pp. 6162–6168, Oct. 2010.

[30] O. Bichler, D. Roclin, C. Gamrat, and D. Querlioz, "Design exploration methodology for memristor-based spiking neuromorphic architectures with the Xnet event-driven simulator," in *2013 IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH)*, pp. 7–12, July 2013.

[31] M. Suri, D. Querlioz, O. Bichler, G. Palma, E. Vianello, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Bio-Inspired Stochastic Computing Using Binary CBRAM Synapses," *IEEE Transactions on Electron Devices*, vol. 60, pp. 2402–2409, July 2013.

[32] A. Chanthbouala, V. Garcia, R. O. Cherifi, K. Bouzehouane, S. Fusil, X. Moya, S. Xavier, H. Yamada, C. Deranlot, N. D. Mathur, M. Bibes, A. Barthélémy, and J. Grollier, "A ferroelectric memristor," June 2012.

[33] Y. Nishitani, Y. Kaneko, and M. Ueda, "Artificial synapses using ferroelectric memristors embedded with CMOS Circuit for image recognition," in *72nd Device Research Conference*, pp. 297–298, June 2014.

[34] E. Chicca, D. Badoni, V. Dante, M. D'Andreagiovanni, G. Salina, L. Carota, S. Fusi, and P. D. Giudice, "A VLSI recurrent network of integrate-and-fire neurons connected by plastic synapses with long-term memory," *IEEE Transactions on Neural Networks*, vol. 14, pp. 1297–1307, Sept. 2003.

[35] S.-C. Liu and R. Douglas, "Temporal coding in a silicon network of integrate-and-fire neurons," *IEEE Transactions on Neural Networks*, vol. 15, pp. 1305–1314, Sept. 2004.

[36] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, pp. 500–544, Aug. 1952.

[37] M. Shahsavari, P. Falez, and P. Boulet, "Combining a Volatile and Nonvolatile Memristor in Artificial Synapse to Improve Learning in Spiking Neural Networks," in *12th ACM/IEEE International Symposium on Nanoscale Architectures (Nanoarch 2016)*, (Beijing, China), July 2016.

[38] M. Shahsavari, M. Faisal Nadeem, S. Arash Ostadzadeh, P. Devienne, and P. Boulet, "Unconventional digital computing approach: memristive nanodevice platform," *physica status solidi (c)*, vol. 12, pp. 222–228, Jan. 2015.

[39] W. Gerstner, R. Ritz, and J. L. Hemmen, "Why spikes? hebbian learning and retrieval of time-resolved excitation patterns," *Biological Cybernetics*, vol. 69, no. 5-6, pp. 503–515, 1993.

[40] H. Markram, J. Lbke, M. Frotscher, and B. Sakmann, "Regulation of synaptic efficacy by coincidence of postsynaptic aps and epsps," *Science*, vol. 275, no. 5297, pp. 213–215, 1997.

[41] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, "STDP and STDP variations with memristors for spiking neuromorphic learning systems," *Frontiers in Neuroscience*, vol. 7, no. 2, 2013.

[42] B. Nessler, M. Pfeiffer, and W. Maass, "Stdp enables spiking neurons to detect hidden causes of their inputs," in *Advances in Neural Information Processing Systems 22* (Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, eds.), pp. 1357–1365, Curran Associates, Inc., 2009.

[43] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[44] 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.

[45] M. Hines, "Neuron a program for simulation of nerve equations," in *Neural Systems: Analysis and Modeling* (F. Eeckman, ed.), pp. 127–136, Springer US, 1993.

[46] D. F. M. Goodman and R. Brette, "Brian: a simulator for spiking neural networks in python," *Frontiers in Neuroinformatics*, vol. 2, no. 5, 2008.

[47] M.-O. Gewaltig and M. Diesmann, "Nest (neural simulation tool)," *Scholarpedia*, vol. 2, no. 4, p. 1430, 2007.

[48] O. Bichler, D. Roclin, C. Gamrat, and D. Querlioz, "Design exploration methodology for memristor-based spiking neuromorphic architectures with the xnet event-driven simulator," in *IEEE/ACM International Symposium on Nanoscale Architectures, NANOARCH 2013, Brooklyn, NY, USA, July 15-17, 2013*, pp. 7–12, 2013.