Analog weight updates with compliance current modulation of binary ReRAMs for on-chip learning

Melika Payvand[1*], Yigit Demirag[1], Thomas Dalgaty[2], Elisa Vianello[2] and Giacomo Indiveri[1]
[1] *Institute of Neuroinformatics, University of Zurich and ETH Zurich, Switzerland*
[2] *CEA Leti, Grenoble, France*

Abstract—**Many edge computing and IoT applications require adaptive and on-line learning architectures for fast and low-power processing of locally sensed signals. A promising class of architectures to solve this problem is that of in-memory computing ones, based on event-based hybrid memristive-CMOS devices. In this work, we present an example of such systems that supports always-on on-line learning. To overcome the problems of variability and limited resolution of ReRAM memristive devices used to store synaptic weights, we propose to use only their High Conductive State (HCS) and control their desired conductance by modulating their programming Compliance Current (I_{CC}). We describe the spike-based learning CMOS circuits that are used to modulate the synaptic weights and demonstrate the relationship between the synaptic weight, the device conductance, and the I_{CC} used to set its weight, with experimental measurements from a 4kb array of HfO$_2$-based devices. To validate the approach and the circuits presented, we present circuit simulation results for a standard CMOS 180 nm process and system-level behavioral simulations for classifying hand-written digits from the MNIST data-set with classification accuracy of 92.68% on the test set.**

I. Introduction

While deep learning has been demonstrating impressive results on computing systems that are not resource constrained (e.g. in terms of power, memory, or resolution), the problem of learning on low power computing hardware still remains unsolved. Many edge devices will benefit (or even require) online learning, to adapt to the changes of the input signals, fine-tune pre-trained networks, or to implement semi-supervised learning algorithms on the field. A promising class of ultra-low power architectures that offers online learning is that of the hybrid memristive-CMOS ones implementing Spiking Neural Network (SNN) [1], [2]. Here we propose novel learning circuits for SNN and 1T1R arrays, that allow analog weight updates on synaptic binary ReRAM devices by controlling their SET operation I_{CC}. In addition to increasing the bit precision of the synaptic weights in online learning, the proposed strategy allows compact, fast, and scalable event-based learning scheme compatible with the Address-Event Representation (AER) [3]. There have been significant efforts from multiple fronts to increase the bit precision of memristive devices for online learning applications, both with research and development on the single device level, and on the architecture.

Material Optimization: Previously, several groups reported TiO$_2$-based [4]–[6] and HfO$_2$-based [7] ReRAM devices with up to 8 bits of precision. However, in all these works, the analog behavior is traded off with the lower available ON/OFF ratio. While the analog behavior is an important concern for training neural networks, cycle-to-cycle and device-to-device variability harms the effective number of bits further when ON/OFF ratio is small. Also, tuning the precise memory state is not always easily achievable in a real-time manner, requiring recursively tuning with an active feedback scheme [7], [8]. Furthermore, some efforts have been focused on carefully designing a barrier level using exhaustive experimental search over a range of materials [4], [5] which makes it difficult to fabricate.

Architecture Optimization: Increasing the “effective” bit resolution has also been done with an architectural optimization. An example of this approach involves using n binary switches to emulate an n-bit synapse as is reported in [9]. Another scheme involving binary devices shows that by using their stochastic switching properties, it is possible to reproduce effective analog-like synaptic adaptation [10], [11]. Alternatively, IBM has used a capacitor next to two devices (in the form of Phase Change Memory (PCM)) which acts as an analog volatile memory, integrating the weight updates and thus increasing the combined capacitor+PCM bit precision while learning [12]. This approach suffers from the large area overhead of the capacitor that is needed per synapse. Also, a mixed-precision approach has been employed to train the networks using a computer for weight update accumulation and transferring the weights to the PCM devices [13]. This solution is applicable for inference-only and not for online-learning applications.

In summary, both device and architecture optimization strategies have not yet led to optimal online learning mechanisms that fully resolve the problems related to low bit precision of memristors. It has been reported and analyzed in [14] that after a SET operation, the resistance value of the memristors, R, follows a linear relationship with the I_{CC} in a log-log scale. I_{CC} provides control over the size of the filament tuning the value of the set resistance. This implies that tuning the I_{CC} can control the resistance of the device.

To minimize the effect of variability, in this work we follow a co-design approach by restricting the devices to stay only in their HCS and controlling their conductance by modulating their programming I_{CC}. Specifically, for the online learning application, we calculate the weight update using the Delta rule learning algorithm [15] and map it onto the current compliance used for setting the device. We derive a

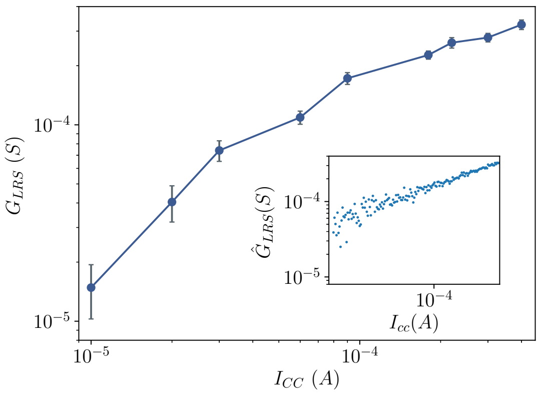

Fig. 1: Mean and STD of the device conductance as a function of the I_{CC}s. The inset shows the samples from the fitted mean and STD used for the simulations.

technologically plausible learning algorithm which makes use of the programming conditions of the memristive devices.

This kind of algorithm-device co-design provides the opportunity to (i) relax the fabrication constraints that is required for multi-bit devices and (ii) provide more state stability compared to the multi-bit approaches by practically using 2 levels per device and only changing the value of the levels.

II. ReRAM Measurements

To find the average relationship between the mean of the cycle-to-cycle distribution of the HCS and the SET programming I_{CC}, we performed measurements on a 16 × 256 (4kb) array of HfO$_2$-based ReRAM devices integrated onto a 130 nm CMOS process between metal layers 4 and 5 [16]. Each device is connected in series to the drain of an n-type selector transistor which allows the SET programming I_{CC} to be controlled based on the voltage applied to its gate. The 1T1R structure allows a single device to be selected for reading or programming by applying appropriate voltages to a pair of Source/Bit Lines (SL/BL) and a single Word Line (WL).

All 4kb devices were initially formed in a raster scan fashion by applying a large voltage (4V typical) between the SL and BL to induce a soft breakdown in the oxide layer and introduce conductive oxygen vacancies. After forming, each device was subject to sets of 100 RESET/SET cycles over a range of SET I_{CC}s between 10μA and 400μA, where the resistance of each device was recorded after each SET operation. The mean of all devices' median resistances over the 100 cycles, at a single I_{CC}, gives the average relationship between HCS median and SET I_{CC} as in Fig. 1. The relationship is seen to follow a line in the log-log plot (power law) and over this I_{CC} range, it allows precise control of the conductance median of the cycle-to-cycle distribution between 50kΩ and 2kΩ.

III. Technologically Plausible Learning Algorithm

The learning algorithm is based on the Delta rule, the simplest form of the gradient descent for single-layer networks.

In our implementation, the objective function is defined as the difference between the desired target output signal y and the network prediction signal $\hat{y}$, for a given set of input patterns signals x, weighted by the synaptic weight parameters w. Then the Delta rule can be used to calculate the change of the weights connecting a neuron i in the input layer and a neuron j at the output layer as follows:

$$\Delta w_{ji} = \eta(y_j - \hat{y}_j)x_i = \eta\delta_j \; x_i, \tag{1}$$

where δ_j is the error, and η is the learning rate. To implement this using memristive synaptic architecture, we represent each synaptic weight w, by the combined conductance of two memristors, w_{ji1} and w_{ji2}, arranged in a push-pull differential configuration. This scheme extends the effective dynamic range of a single synapse to capture the negative values.

During the network operation, the target and the prediction signals are compared continuously to generate the error signal. With the arrival of a pre-synaptic event, if the error signal is larger than a small error threshold, the weight update process is initiated. The small error threshold that creates the "stop-learning" regime has been proposed to help the convergence of the neural networks with stochastic weight updates [17].

The implementation of the synaptic plasticity consists of three phases (Alg. 1). First, a READ operation is performed on every excitatory and inhibitory memristor to determine their conductance. Then the resulting current values (I_{ji1} and I_{ji2}) are scaled to the level of the error signal. Second, the current value proportional to the amount of the weight change $\eta\delta_j x_i$ is summed up with the scaled READ current to represent the desired conductance strength to be programmed. Finally,

Algorithm 1: Delta Rule implementation with dual-memristors

```
$w_{ji1} = rand(),\ w_{ji2} = rand();$
while $t < simDuration$ do
    $\delta_j = |\hat{y} - y|;$
    if @Pre and $\delta_j > \delta_{th}$ then
        forall $w_{ji}$ do
            $I_{ji1}, I_{ji2} = READ(w_{ji1}, w_{ji2});$
            $I_1 = I_{ji1} * scale\_const;$
            $I_2 = I_{ji2} * scale\_const;$
            if $(\hat{y} - y) > 0$ then
                $S_1 = I_1 + \eta\delta_j;$
                $ICC_{ji1} = S_1 * scale\_const2;$
                $S_2 = I_2 - \eta\delta_j;$
                $ICC_{ji2} = S_2 * scale\_const2;$
            else
                $S_1 = I_1 - \eta\delta_j;$
                $ICC_{ji1} = S_1 * scale\_const2;$
                $S_2 = I_2 + \eta\delta_j;$
                $ICC_{ji2} = S_2 * scale\_const2;$
            end
            $RESET(w_{ji1}, w_{ji2});$
            $SET(w_{ji1}, w_{ji2});$
        end
    end
end
```

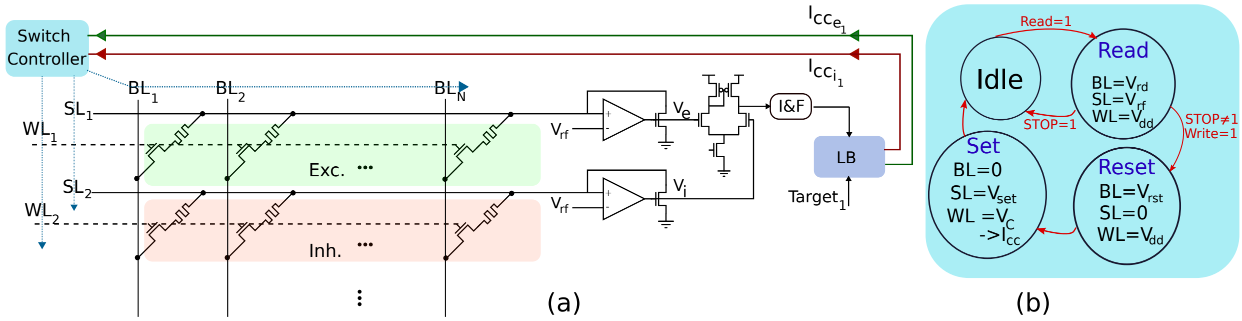

Fig. 2: Event-based neuromorphic architecture using on-line learning in a 1T1R array (a), and the asynchronous state machine used as the switch controller applying the appropriate voltages on the BL, SL and WL of the array for online learning.

these currents are further scaled to a valid I_{CC} range. To provide a larger dynamic range per synapse, the conductance of both memristors are updated with a push-pull mechanism considering the sign of the error (i.e. if the conductance of one memristor increased, the conductance of the complementary memristor is decreased, and vice-versa).

IV. Learning Circuits and Architecture

Neuromorphic Architecture: Figure 2a illustrates the event-based neuromorphic architecture encompassing the learning algorithm. It consists of a 1T1R array, a Switch Controller, Leaky Integrate and Fire (I&F) neurons and a learning block (LB). Every neuron receives excitatory and inhibitory currents from two rows of the 1T1R array respectively.

With the arrival of every event through the AER interface (not shown), two consecutive READ and WRITE signals are generated [2]. Based on these signals, the asynchronous state machine in Fig. 2b controls the sequence so that the SLs, BLs and the WLs of the array are driven by the appropriate voltages such that: device is read, its value is integrated by the I&F neuron; the error value is updated through the learning block (LB), generating I_{CC1} and I_{CC2} (section III); Based on these values, the excitatory and inhibitory devices are programmed.

Learning Circuits: Based on Alg. 1 and data from Fig. 1, we have designed circuits that generate the appropriate I_{CC}, based on the firing rate distance between the neuron and its target. Figure 3 presents these circuits. The spikes from the neurons and the target are integrated using subthreshold Gm-C filters highlighted in red generating V_N and V_T. These voltages are subtracted from one another using a subthreshold ”Bump” (*subBump*) circuit [18] highlighted in green, and an above-threshold ”Bump” circuit (*abvBump*) in orange.

subBump circuit compares V_N and V_T giving rise to the error currents when neuron and the target frequencies are far apart and generates the STOP signal when the error is small and in the stop-learning range (δ_{th}) [17], [19]. STOP signal gates the tail current of all the above-threshold circuits and thus substantially reduce the power consumption when the learning is stopped. Moreover, input events are also used as another gating mechanism. *abvBump* circuit subtracts V_N and V_T and scales it to I_{scale}, equal to the maximum I_{CC} required based on Fig. 1. Based on the error sign (UP), the scaled error current is summed with or subtracted from the scaled device current generating the desired I_{CC} (Alg. 1). This circuit is highlighted in purple.

Circuit Simulations Results: Figure 4 depicts the positive and negative error currents, STOP-learning signal, and the I_{CC1} and I_{CC2} currents. The error currents follow a Sigmoid which can be approximated by a line for error values between -1 and 1. As is explained in Alg. 1, for positive errors, I_{CC2} (I_{CC1}) follows the summation (subtraction) of the error current with the scaled device current, while for the negative errors, it is the opposite. Figure 5 illustrates the dependence of the I_{CC} on the current value of the devices which shifts the error current curve up or down.

V. System-level simulations

We performed SNN simulations with BRIAN2 [20] to test the performance of the presented update scheme using the fitted device data in Fig. 1 with stochastic weight changes. Our goal was to have a comparable test accuracy with the artificial neural networks classically trained using 32b floating point precision on the digital hardware. We tested our network on the MNIST handwritten digits dataset [21] using the first five classes, which is composed of 30596 training and 5139 testing images of 28×28 pixels. We trained a fully-connected single-layer network with 784 input LIF neurons in the first layer and 5 LIF neurons at the output layer.

Each input image is presented to the network for 100 ms. We encoded the input pixel intensity as $[0, 200]$ Hz Poisson spikes. At the output layer, spikes are counted per neuron during each stimulus, and the neuron with the maximum firing rate is selected as the network prediction. The error signal is calculated as the difference between low-pass filtered network output spikes and low-pass filtered target spikes, which is encoded as Poisson spikes with a rate of 40 kHz.

The cycle-to-cycle variability of I_{CC} dependent G_{LRS} conductance follows a Gaussian distribution with I_{CC} dependent mean and standard deviation (std) (Sec. II). This relationship is modeled using the power-law $y = Ax^B$ for both mean and the std. This device variability model is assigned to all synaptic devices in the simulation. After training three epochs on five classes, a test accuracy of 92.68% is achieved.

VI. Discussion

There is a significant effort in developing learning algorithms for SNNs given their potential for highly-parallel and

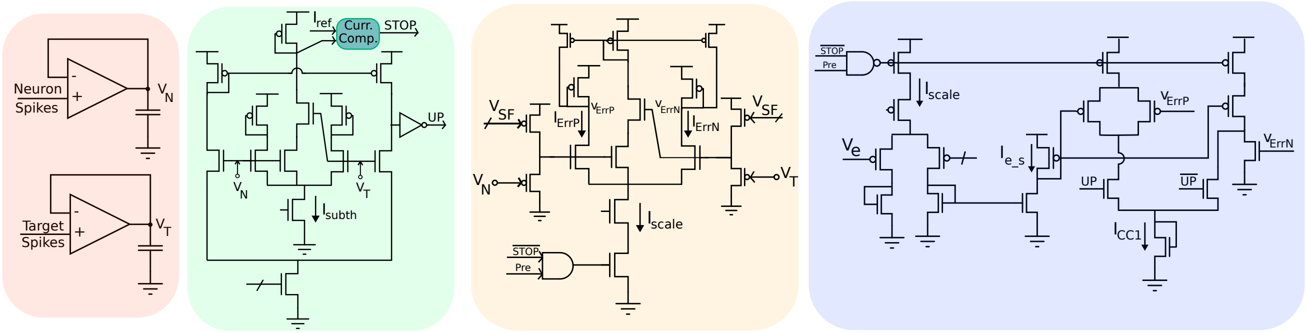

Fig. 3: Learning circuits generating the I_{CC} for updating the devices based on the distance between the neuron and its target frequency. Highlighted in red is the Gm-C filters, low pass filtering the neuron and target spikes giving rise to V_N and V_T. In green and orange, the error between the two is calculated generating positive (I_{ErrP}) and negative (I_{ErrN}) errors, unless error is small and STOP signal is high. In purple, V_e, the excitatory voltage from Fig.2, regenerates the read current and is scaled to I_{scale} producing I_{e_S}. Based on the error sign (UP), I_{CC1} is either the sum of I_{e_S} and I_{Err} or the subtraction of the two.

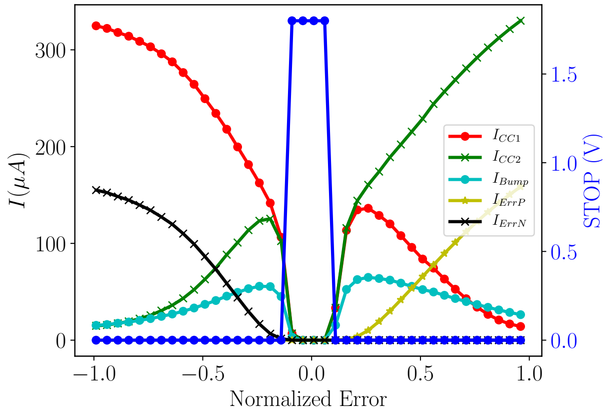

Fig. 4: Error current, STOP learning signal and I_{CC} as a function of the normalized error between the target and the neuron frequencies.

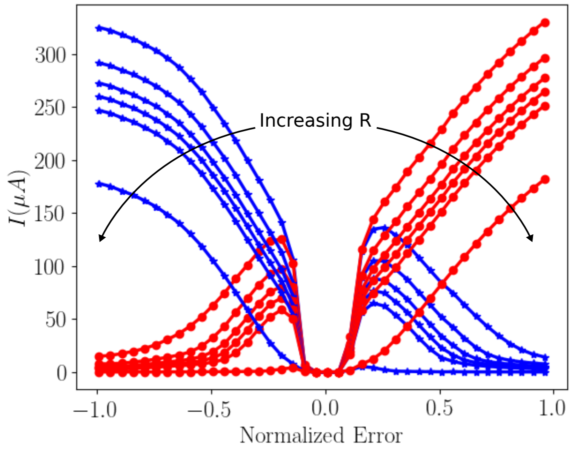

Fig. 5: Change of the I_{CC1} (red) and I_{CC2} (blue) as a function of the error and the resistance value of the devices.

low-power processing. However, there is a large gap between these algorithms and the feasibility of their implementation on hardware given the noise, variability and the available bit precision. This gap calls for the importance of technologically plausible learning algorithms developed from the device physics/measurements [22]. This paper proposes a step in this direction exploiting the I_{CC} of ReRAMs for weight updates. Some important factors to consider are:

Power consumption and Scalability: As the LB generates up to 100s of μAs of I_{CC} for large errors, design considerations must be given such as our event and STOP-learning signal gating, to reduce the average power consumption. Each LB has a peak current between 1 to 600 μA depending on the network error. Thanks to the Poisson distribution of the events (due to thermal noise), we can assume that at each instance only the devices in one column are programmed, and thus, the peak power grows slower than linearly with the number of neurons (linear in the worst case). Hence, the scalability is not mainly limited by the power consumption. However, with Poisson distributed input events and a maximum frequency for each input channel, an upper bound for the array size exists which can be calculated based on the pulse width of the events, and the tolerance to missing events [11].

The non-linear effect: The power-law nature of $I_{CC} \rightarrow G_{LRS}$ transformation in Fig. 1 non-linearly maps the updated weights to the actual updates. This introduces a slight bias in the weight update deviating from the optimal weights calculated by Delta rule.

VII. Conclusions

Here, we proposed a technologically plausible learning algorithm that takes advantage of the compliance current of binary ReRAMs to effectively generate variable, multi-level conductance changes. We have presented a complete co-design approach within multiple levels of abstractions, ranging from device measurements, algorithm, architecture and circuits. We argue that this work takes a significant step toward building always-on on-chip learning systems.

Acknowledgment

This work is supported by funding from Toshiba corporation, the Swiss Forschungskredit grant FK-19-106, and by the H2020 ERC Project NeuroAgents under Grant 724295.

References

[1] G. Indiveri, B. Linares-Barranco, R. Legenstein, G. Deligeorgis, and T. Prodromakis, "Integration of nanoscale memristor synapses in neuromorphic computing architectures," *Nanotechnology*, vol. 24, no. 38, p. 384010, 2013.

[2] M. Payvand, M. V. Nair, L. K. Müller, and G. Indiveri, "A neuromorphic systems approach to in-memory computing with non-ideal memristive devices: From mitigation to exploitation," *Faraday Discussions*, vol. 213, pp. 487–510, 2019.

[3] S. Deiss, R. Douglas, and A. Whatley, "A pulse-coded communications infrastructure for neuromorphic systems," in *Pulsed Neural Networks*, W. Maass and C. Bishop, Eds. MIT Press, 1998, ch. 6, pp. 157–78.

[4] S. Stathopoulos, A. Khiat, M. Trapatseli, S. Cortese, A. Serb, I. Valov, and T. Prodromakis, "Multibit memory operation of metal-oxide bi-layer memristors," *Scientific reports*, vol. 7, no. 1, p. 17532, 2017.

[5] M. Prezioso, F. Merrikh-Bayat, B. Hoskins, G. C. Adam, K. K. Likharev, and D. B. Strukov, "Training and operation of an integrated neuromorphic network based on metal-oxide memristors," *Nature*, vol. 521, no. 7550, p. 61, 2015.

[6] F. M. Bayat, M. Prezioso, B. Chakrabarti, H. Nili, I. Kataeva, and D. Strukov, "Implementation of multilayer perceptron network with highly uniform passive memristive crossbar circuits," *Nature communications*, vol. 9, no. 1, p. 2331, 2018.

[7] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves *et al.*, "Analogue signal and image processing with large memristor crossbars," *Nature Electronics*, vol. 1, no. 1, p. 52, 2018.

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

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

[10] M. Suri, O. Bichler, D. Querlioz, G. Palma, E. Vianello, D. Vuillaume, C. Gamrat, and B. DeSalvo, "Cbram devices as binary synapses for low-power stochastic neuromorphic systems: auditory (cochlea) and visual (retina) cognitive processing applications," in *2012 International Electron Devices Meeting*. IEEE, 2012, pp. 10–3.

[11] M. Payvand, L. K. Muller, and G. Indiveri, "Event-based circuits for controlling stochastic learning with memristive devices in neuromorphic architectures," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[12] S. Ambrogio, P. Narayanan, H. Tsai, R. M. Shelby, I. Boybat, C. di Nolfo, S. Sidler, M. Giordano, M. Bodini, N. C. Farinha *et al.*, "Equivalent-accuracy accelerated neural-network training using analogue memory," *Nature*, vol. 558, no. 7708, p. 60, 2018.

[13] S. Nandakumar, M. Le Gallo, I. Boybat, B. Rajendran, A. Sebastian, and E. Eleftheriou, "Mixed-precision architecture based on computational memory for training deep neural networks," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2018, pp. 1–5.

[14] D. Ielmini, "Resistive switching memories based on metal oxides: mechanisms, reliability and scaling," *Semiconductor Science and Technology*, vol. 31, no. 6, p. 063002, 2016.

[15] B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition," *Computer*, vol. 21, no. 3, pp. 25–39, 1988.

[16] A. Grossi, E. Nowak, C. Zambelli, C. Pellissier, S. Bernasconi, G. Cibrario, K. E. Hajjam, R. Crochemore, J. F. Nodin, P. Olivo, and L. Perniola, "Fundamental variability limits of filament-based rram," in *2016 IEEE International Electron Devices Meeting (IEDM)*, Dec 2016, pp. 4.7.1–4.7.4.

[17] M. Payvand and G. Indiveri, "Spike-Based Plasticity Circuits for Always-on On-Line Learning in Neuromorphic Systems," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. Sapporo, Japan: IEEE, May 2019, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8702497/

[18] T. Delbruck, "'bump'circuits for computing similarity and dissimilarity of analog voltages," in *Neural Networks, 1991., IJCNN-91-Seattle International Joint Conference on*, vol. 1. IEEE, 1991, pp. 475–479.

[19] M. Payvand, M. Fouda, F. Kurdahi, A. Eltawil, and E. O. Neftci, "Error-triggered three-factor learning dynamics for crossbar arrays," *arXiv preprint arXiv:1910.06152*, 2019.

[20] D. Goodman, "Brian: a simulator for spiking neural networks in Python," *Frontiers in Neuroinformatics*, vol. 2, 2008. [Online]. Available: http://journal.frontiersin.org/article/10.3389/neuro.11.005.2008/abstract

[21] 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. 2278–2324, Nov. 1998. [Online]. Available: http://ieeexplore.ieee.org/document/726791/

[22] T. Dalgaty, M. Payvand, F. Moro, D. R. Ly, F. Pebay-Peyroula, J. Casas, G. Indiveri, and E. Vianello, "Hybrid neuromorphic circuits exploiting non-conventional properties of rram for massively parallel local plasticity mechanisms," *APL Materials*, vol. 7, no. 8, p. 081125, 2019.