A SiO_x RRAM-based hardware with spike frequency adaptation for power-saving continual learning in convolutional neural networks

I. Muñoz-Martin[1*], S. Bianchi[1*], E. Covi[1], G. Piccolboni[2], A. Bricalli[2], A. Regev[2], J. F. Nodin[3], E. Nowak[3], G. Molas[3] and D. Ielmini[1]

[1] Politecnico di Milano, Milan, Italy. [2] Weebit Nano, Hod Hasharon, Israel. [3] CEA, LETI, Grenoble, France. * Contributed equally.

E-mail: irene.munoz@polimi.it

Abstract. Biological systems autonomously evolve to maximize their efficiency in a continually changing world. On the other hand, artificial neural networks (ANNs) outperform the human ability of object recognition but cannot acquire new information without forgetting trained tasks. To introduce resilience in ANNs, we present a SiO_x RRAM-based inference hardware capable of merging the efficiency of convolutional ANNs and the plasticity of spiking networks. We validate the accuracy of the system with MNIST (99.3%), noisy N-MNIST (96%), Fashion-MNIST (93%) and CIFAR-10 (91%) datasets. We demonstrate that the circuit plastically adapts its operative frequency for power saving and enables continual learning of up to 50% non-trained classes. This optimizes the classification and enables the re-training of the filters, thus overcoming the catastrophic forgetting of standard ANNs.

Lifelong learning. The experience-based knowledge transfers and adapts the incoming information to achieve continual learning throughout life, Fig. 1a. Artificial neural networks (ANNs) show great accuracy in object recognition [1], but lack of the necessary plasticity, typical of spiking neural networks (SNNs), for enabling lifelong learning. This arises the so called "stability-plasticity dilemma", Fig. 1b [2]. In order to overtake this limitation, we propose to merge the accuracy of standard convolutional neural networks (CNNs), and the plasticity of unsupervised learning SNNs, Fig. 1c, [3].

RRAM-based system. The network is implemented with SiO_x RRAM devices by Weebit, Fig. 2a. Fig. 2b shows the I-V characteristics with different compliance currents I_C. Fig. 2c presents the low resistive state (LRS) distributions, with mean μ_R (d), and standard deviation σ_R, (e). Fig. 2f shows the high resistive state (HRS) distributions versus V_{STOP}. The RRAMs are used (i) in the supervised section for the convolutional filters and (ii) in the unsupervised block for classification via winner-take-all (WTA), Fig. 3a. Fig. 3b highlights the 20x20 convolutional filters, which are either "class filters" (CFs) or "feature filters" (FFs), [4]. CFs are specialized in recognizing a specific trained class respect to all the others while FFs extract generic features from the training dataset which, for transfer learning, can be found even in non-trained objects [5]. Each weight of the filters is digitalized in 4 bits inside the RRAM array, Fig.3c, and used for matrix vector multiplication (MVM), (d). The large dimension of the filters reduces the number of convolutional steps. The results after MVM are processed into a Zync 7000 FPGA, which, for every filter, applies max pooling and compares the outcome with the fixed bias threshold V_{REF} of that filter. If V_{REF} is overcome, the corresponding pixel is set high. The combination of the responses of the filters constitutes a "feature map", original for every class, which is classified by the asynchronous spike-timing-dependent plasticity block (A-STDP). Fig. 4a shows the circuital scheme of the network. The outputs of the MVM enter the 8x8 SiO_x crossbar (64 filters are used) after every firing activity. The internal threshold of each neuron relies on spike-frequency adaptation (SFA) [6], Fig. 4b. When a neuron fires, it partially sets a further SiO_x RRAM which, in turn, increases its internal threshold V_{TH}. Thus, each neuron specializes in a feature map of a certain input class, avoiding confusion with other neurons and saving power for the reduced switching activity. Note that a further bit-line in Fig. 4a is used for pattern detection, since each STDP activity relies on uncorrelated signals (noise) for synaptic background depression, Fig. 4c [7].

Accuracy of the filters. The accuracy of the convolutional filters depends on the number of bits used for mapping each weight, Fig.5. The programming I_C and V_{STOP} of the RRAMs affect the accuracy of both CFs, Fig. 6a-b, and FFs, Fig. 6c-d. The average results for CFs and FFs are gathered in the contour plots of Fig. 6e-f. The filters are trained with respect to V_{REF}.

Dynamic continual learning. Fig. 7 shows the time evolution of the signals at the input of the A-STDP block during the testing of the CIFAR-10 dataset, showing the average trained (a) and non-trained (b) feature maps. The feature maps are blurred, since there is variability within the input classes. Fig. 7c shows the time evolution of the scatter plot for the non-trained class "dog". Note that the feature map varies in time, as visible in the snapshots t_1, t_2 and t_3. This affects the final accuracy, since not every "dog" pattern hits the threshold of the neuron. Only limited variations (< 20%) of the feature map density are tolerated by A-STDP. Fig. 8 shows the synaptic evolution of 5 non-trained classes (a) from the MNIST, with the synaptic evolutions for pattern and background (b) and the SFA (c). The highest spiking rate occurs when new feature maps appear (d), as the output neurons have not specialized yet.

Accuracy and variability. Fig. 9 shows the results for the accuracy of full-trained MNIST (a), noisy N-MNIST (20% of random pixels in each MNIST pattern) (b), Fashion MNIST (c) and CIFAR-10 (d) varying the I_C current used for programming the filters and the A-STDP synapses. Once selected the best I_C condition, we tested the network with 1 to 5 non-trained classes, i.e. progressively decreasing the trained part of each dataset. Fig. 10 shows the accuracy of non-trained classes (a) and the global accuracy (b). Note that when a new class appears, the spike frequency of the further classification neuron is initially higher, causing an increase of the power consumption, Fig. 11.

Re-training. Some feature filters may have a negligible rate of appearance, as shown in Fig. 12 for MNIST. Once neglected, these filters do not catastrophically reduce the accuracy, since they do not modify the threshold hitting of the neurons, as in the case of neuron 4, Fig. 13a. These filters can be re-trained accordingly to the clustered patterns recorded by the classification neurons during continual learning, Fig. 13b, thus obtaining a significant improvement of the accuracy, Fig.14.

Conclusions. We presented a novel SiO_x RRAM-based hardware that combines the efficiency of supervised architectures and the adaptation of unsupervised learning to solve the stability-plasticity dilemma. The system provides high accuracies for full testing (99.3% for MNIST, 91% for CIFAR-10) and learning of up to 50% non-trained classes. The use of power saving spike-frequency modulation enables feasible solutions for lifelong learning in autonomous systems.

References. [1] A. Krizhevsky et al., NIPS, vol. 1. pp. 1097–1105, 2012. [2] M. McCloskey et al., Psyc. of learn. and mot. 24, 109, 1989. [3] S. Bianchi et al., VLSI 2019, 10.23919/VLSIT.2019.8776559. [4] I. Munoz et al., IEEE JXCDC, vol. 5, pp. 58-66, 2019. [5] L. Torrey et al., Transfer learning, 2009, 10.4018/978-1-60566-766-9.ch011. [6] G. Indiveri et al., Frontiers in Neurosci., vol. 5:73, 2011. [7] M. Prezioso et al., Nature communications, vol. 9, 5311, 2018.

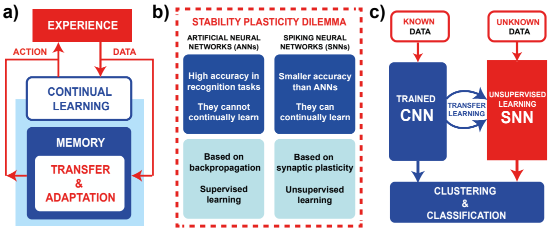

Fig.1. (a) Experience-based knowledge for continual learning in nature. (b) Stability-plasticity dilemma. (c) Merge CNNs and SNNs to achieve resilient ANNs.

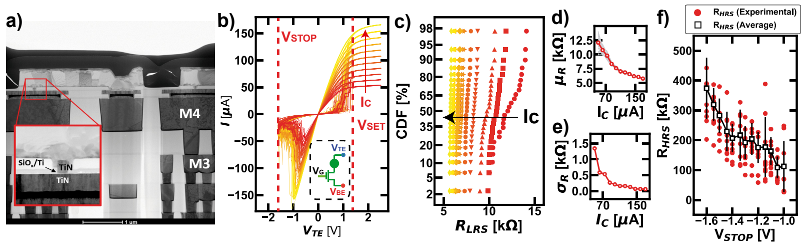

Fig.2. (a) SiO_x RRAM devices. (b) I-V characteristics for different compliance currents I_C. (c) Low resistive distributions as a function of I_C, with mean μ_R(d) and standard deviation σ_R (e). (f) High resistive states as a function of V_{STOP}.

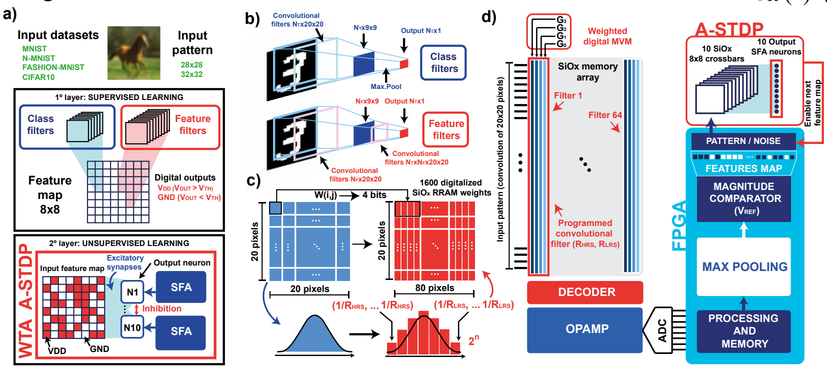

Fig.3. (a) The network consists of a convolutional layer and a winner-take-all (WTA) with spike frequency adaptation (SFA). (b) All the filters are trained with a fully convolutional approach. (c) The weights are digitalized in 4 bits and referred to values > 0. (d) The convolution is performed with RRAM-based matrix-vector multiplication (MVM). The results are stored into an FPGA and feed the A-STDP block.

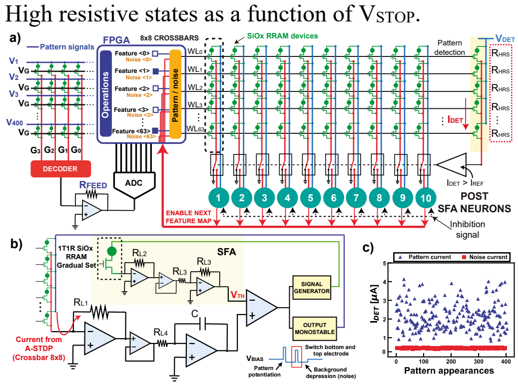

Fig.4. (a) The convolutional results after MVM feed the A-STDP after a neuron fires. (b) Each neuron self-adapts its V_{TH} as a function of its firing frequency acting on the multilevel LRS of a RRAM (SFA). (c) Role of the pattern detection for improved STDP.

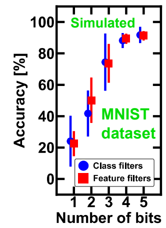

Fig.5. Accuracy versus binary weight precision.

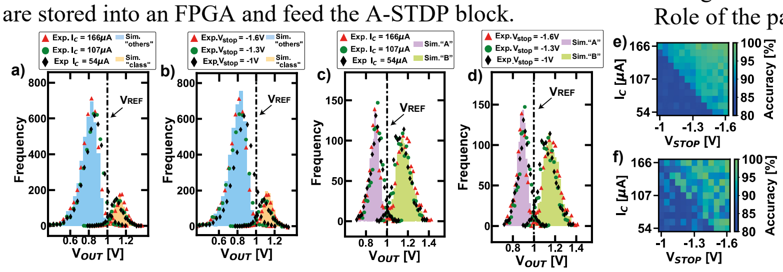

Fig.6. Experimental and simulated results of accuracy for a class filter (a,b) and a feature filter (c,d) as a function of I_C and V_{STOP}, respectively. Color maps for the best average programming condition of class filters (e) and feature filters (f).

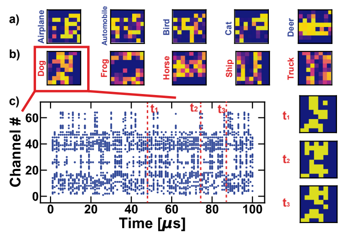

Fig.7. Measured trained (a) and non-trained (b) CIFAR-10 feature maps. (c) Time-evolution for "dog" class.

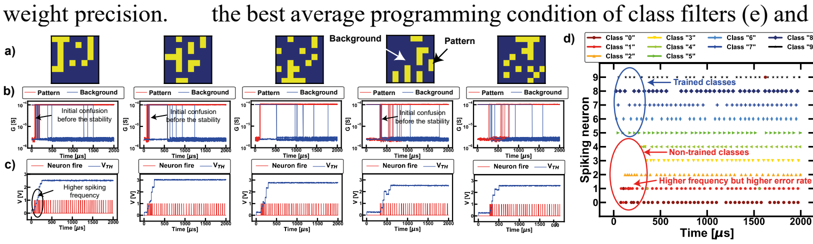

Fig.8. (a) Experimental examples of feature maps for 5 non-trained classes of MNIST with (b) the synaptic evolution in time and (c) the spike-frequency adaptation. (d) Raster plot of the fire activity of trained and non-trained neurons.

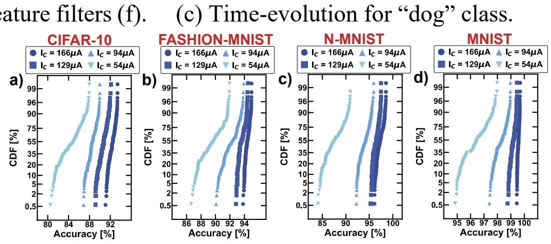

Fig.9. Full testing accuracy of the 4 datasets at various programming I_C for the weights and synapses (V_{STOP} = -1.6 V). A low I_C negatively affects the accuracy.

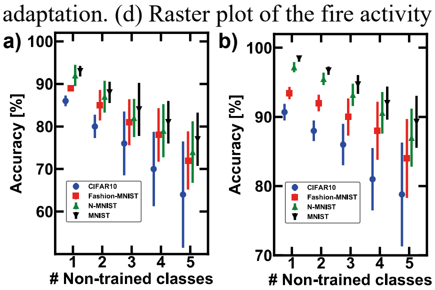

Fig.10. (a) Experimental average results for the accuracy of 1 to 5 non-trained classes of the 4 datasets. (b) Global average accuracies. Note that the accuracy decreases as the number of non-trained classes increases.

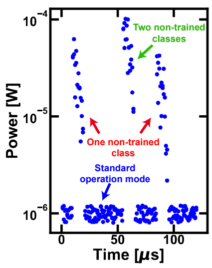

Fig.11. The power consumption depends on the number of new classes to classify by continual learning.

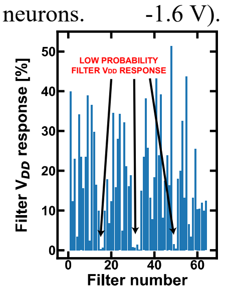

Fig.12. Experimental analysis of the use of trained filters during MNIST continual learning of 5 classes.

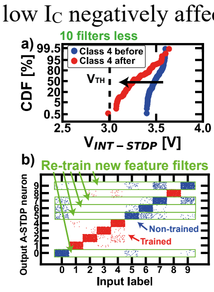

Fig.13. Measured impact of erased filters on neuron 4 (a). Re-training of the new filters (b).

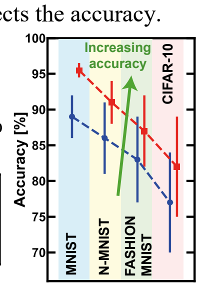

Fig.14. Improved results after re-training the filters with the clusters of non-trained classes.