A Robust Learning Rule for Memristor-based Synapses Competitive with Supervised Learning in Standard Spiking Neural Networks

Thomas F. Tiotto*, Jelmer P. Borst† and Niels A. Taatgen‡
Groningen Cognitive Systems and Materials Center
University of Groningen
Groningen, The Netherlands
Email: *t.f.tiotto@rug.nl, †j.p.borst@rug.nl, ‡n.a.taatgen@rug.nl

***Abstract*—One influential view in theoretical neuroscience sees the brain as a function-computing device, thus being able to approximate functions is fundamental to build future brain research and to derive efficient computational machines. Here we do so by applying a novel learning algorithm based on controlling memristive synapses, which is able to match the performance of standard methods.**

***Index Terms*—Beyond CMOS, Neuromorphics, Memristors, Hebbian theory**

I. Introduction

Theoretical Neuroscience has mostly adopted the view that sees the brain as a function approximator [1], defining and studying it as a device which applies functions to its inputs in order to generate outcomes in the form of new internal states and motor outputs.

Memristors are a novel class of device that has attracted great research interest since its realisation [2] due to its capacity to maintain a resistance state in absence of external stimuli. The fact that the physical state of this device can be directly changed by applying voltage pulses to the terminals allows its programming with a fraction of the power needed for silicon transistors [3].

When looking to advance beyond the Von Neumann architecture and Moore's law, it comes quite natural to look to the brain for inspiration as it is maybe the most remarkable computational device we are aware of. If we want to start to narrow the gap between our computers and the brain, we need both new materials and novel computing paradigms: memristors can give us the former, while we can start working towards the latter by adopting a function-based computational approach as the brain's is believed to be.

In a previous paper we presented a novel supervised learning algorithm controlling the resistance of *nickel/niobium-doped strontium titanate* (Ni/Nb-doped $SrTiO_3$) memristive synapses, which we named *memristor PES* (mPES). The implementation was done in Nengo, a spiking neural network simulator representing information and determining the synaptic weights using the principles of the *Neural Engineering Framework* (NEF) [4]. In this work our methodology is extended by applying mPES to learning non-trivial, non-linear functions of varying dimensionality. It is shown that the performance of the MacNeil & Eliasmith general error-based learning rule *Prescribed Error Sensitivity* (PES) [5] can be matched by reproducing and extending the benchmarks by Bekolay [6].

The aim of this paper is to show that memristors can be used as substrate to approximate complex functions, setting the grounds for functional brain simulations and computational research.

II. Methods

The PES learning rule [5] is incorporated into Nengo and accomplishes online error minimization by solving

$$\Delta\omega_{ij} = \kappa\alpha_j e_j \boldsymbol{E} a_i \tag{1}$$

where ω_{ij} is the weight of the connection between pre-synaptic neuron i and post-synaptic neuron j, κ is the learning rate, α_j and e_j are NEF-specific parameters, $\boldsymbol{E}$ is the global d-dimensional error to minimize, and a_i the activity of neuron i.

Memristive devices change their internal state and resistance in response to voltages above a threshold. Here, simulated Ni/Nb-doped $SrTiO_3$ devices were utilised, where resistive switching results from changes occurring at the interface [7]. To derive a model of the device, it was subjected to a series of electrical measurements and it was found that the power-law in (2) could explain the change in resistance $R(n)$ as a function of the number n of $+0.1$ V pulses applied to the device terminals

$$R(n) = 200 + 2.3 \times 10^8 n^{-0.146} \tag{2}$$

mPES is a novel learning rule that operates on the memristors' resistance and is - essentially - a discretised version of PES. To be able to represent negative network weights - as resistance and its reciprocal conductance are positive physical quantities - each synaptic weight is represented by the difference in conductance between two memristors $M_{ij}^{\pm}$ and mPES applies voltage pulses to either one or the other in order to minimise the error $\boldsymbol{E}$ in (1). The overall effect of this procedure is that synapses whose neurons have a beneficial participation to the error are facilitated, and those whose neurons have a negative effect are depressed so as to decrease the probability of them re-activating.

A simple network topology is defined specifically to test the learning rule: it consists of a noisy `input` signal x projecting to a `pre`-synaptic neuronal ensemble, a `post`-synaptic ensemble representing y - which is connected to `pre` via a plastic connection - a `ground truth` ensemble representing the transformed input $f(x)$, and finally an `error` ensemble

 ISBN: 978-1-61208-950-8

TABLE I
FUNCTIONS f UNDER TEST. DIMENSIONALITY d AND NUMBER OF NEURONS # OF THE MAIN NEURONAL ENSEMBLES. SIMULATED TIME FOR EACH NETWORK RUN. ERROR AND 95 % CONFIDENCE INTERVAL MEASURED ON THE FINAL TESTING BLOCK.

Function f	`Pre [d/#]`	`Error [d/#]`	`Post [d/#]`	**Sim. time**	**Error (CI $\pm$) [mPES/PES/NEF]**
$f(x_1, x_2) = x_1 \times x_2$	2-D / 200	1-D / 100	1-D / 200	50 s	216 ($\pm$31) / 227 ($\pm$40) / 141 ($\pm$20)
$f(x_1, x_2, x_3, x_4) = x_1 \times x_2 + x_3 \times x_4$	4-D / 400	1-D / 100	1-D / 400	100 s	394 ($\pm$32) / 446 ($\pm$60) / 261 ($\pm$39)
$f(x_1, x_2, x_3) = [x_1 \times x_2, x_1 \times x_3, x_2 \times x_3]$	3-D / 300	3-D / 300	3-D / 300	100 s	745 ($\pm$83) / 773 ($\pm$92) / 483 ($\pm$60)
$f(x_1, x_2, x_3, x_4) = [x_1, x_2] \otimes [x_3, x_4]$	4-D / 400	2-D / 200	2-D / 200	200 s	771 ($\pm$70) / 813 ($\pm$73) / 710 ($\pm$63)
$f(x_1, x_2, x_3, x_4, x_5, x_6) = [x_1, x_2, x_3] \otimes [x_4, x_5, x_6]$	6-D / 600	3-D / 300	3-D / 300	400 s	1364 ($\pm$64) / 1317 ($\pm$68) / 1258 ($\pm$70)

comparing the activity in `post` and in `ground truth` as $\boldsymbol{E} = y - f(x)$. The number of neurons in each neuronal ensemble and the simulation run time are all altered depending on the specific function being tested, with the specific values being reported in Table I. Learning is tested by running mPES, PES, and NEF ten times each for each of the functions f in Table I, with f applied as the transformation on the `input`-to-`ground truth` connection.

The mPES and PES learning rules use the information received from `error` to act on the weights on the plastic connection between `pre` and `post`, resulting in the `pre`-to-`post` connection matrix being progressively tuned to represent the transformation f. Thus as the network learns f, the value y in `post` comes to approximate that in `ground truth`, i.e., $y \sim f(x)$.

The learning performance of mPES and PES are compared to the NEF baseline by breaking up a simulation run into an equal number of 2.5 s *learning* and *testing* blocks. After each simulation run, the values represented by `post` and `ground truth` during each testing block are subtracted and these differences are summed to give the absolute total error for that block.

III. Results and Discussion

When learning the five different functions f in Table I, mPES is able to modulate the memristive synapses' resistances in order to derive a synaptic weight matrix that implements f as well or better than PES. The variations of multiplication and circular convolution learned with mPES have a total error within the confidence interval of that of PES on the final testing block. Across the spectrum, as expected, the performance of the NEF analytically-determined network weight matrix is superior to that of those obtained via online learning.

Thus, mPES is able to match the learning performance of PES, which is remarkable given the restriction imposed by having to operate on non-ideal, stochastic items - as are the simulated memristors - instead of real-valued, continuous network weights. It is also notable that mPES reaches this level of performance without having any information about the magnitude of the updates happening on the underlying memristors: the memristive devices' resistance follows a power-law so subsequent voltage pulses have a monotonically decreasing effect and each update to the resistance is stochastic. This leads us to speculate that mPES could also be applied to systems based on different memristive synapses, not just to the ones utilised in this work.

IV. Conclusion

Using analogue components as basis for neural network weights, especially when paired with a biologically-plausible learning algorithm, is one way of improving the energy efficiency of present-day computers. Memristors are inherently stochastic and this characteristic, if properly harnessed, could turn out to be important to deal with the randomness present in all data resulting from real-world interactions.

The brain is probably the most extraordinary computational device we know of, but how it carries out its feats of intelligence is still mostly a mystery. One way of understanding the brain is to view it as a function-computing machine able - for example - to apply a function to the inputs received from our retinae and decide that we are looking at a cat. Therefore, having a memristor-based neuromorphic system that is able to learn to approximate non-trivial functions - as the ones tested in this work - could prove to be a valuable tool to start to functionally reproduce some of the tasks that the brain seems to carry out so easily and that still elude our best computers and learning models.

References

[1] A. M. Zador, "The basic unit of computation," *Nature neuroscience*, vol. 3, no. 11, pp. 1167–1167, 2000.

[2] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[3] D. S. Jeong, K. M. Kim, S. Kim, B. J. Choi, and C. S. Hwang, "Memristors for energy-efficient new computing paradigms," *Advanced Electronic Materials*, vol. 2, no. 9, p. 1600090, 2016.

[4] C. Eliasmith and C. H. Anderson, *Neural engineering: Computation, representation, and dynamics in neurobiological systems*. MIT press, 2003.

[5] D. MacNeil and C. Eliasmith, "Fine-tuning and the stability of recurrent neural networks," *PloS one*, vol. 6, no. 9, p. e22885, 2011.

[6] T. Bekolay, "Learning nonlinear functions on vectors: examples and predictions," Technical Report CTN–TR–20101217–010, Centre for Theoretical Neuroscience . . . , Tech. Rep., 2010.

[7] A. Goossens, A. Das, and T. Banerjee, "Electric field driven memristive behavior at the schottky interface of nb-doped srtio3," *Journal of Applied Physics*, vol. 124, no. 15, p. 152102, 2018.