



**HAL**  
open science

# Overcoming device unreliability with continuous learning in a population coding based computing system

Alice Mizrahi, Julie Grollier, Damien Querlioz, M. Stiles

## ► To cite this version:

Alice Mizrahi, Julie Grollier, Damien Querlioz, M. Stiles. Overcoming device unreliability with continuous learning in a population coding based computing system. *Journal of Applied Physics*, 2018, 124 (15), pp.152111. <10.1063/1.5042250>. <hal-02350218>

**HAL Id: hal-02350218**

**<https://hal.science/hal-02350218v1>**

Submitted on 14 Jan 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Overcoming device unreliability with continuous learning in a population coding based computing system

Alice Mizrahi,<sup>1,2, a)</sup> Julie Grollier,<sup>3</sup> Damien Querlioz,<sup>4</sup> and M.D. Stiles<sup>1</sup>

<sup>1)</sup>*National Institute of Standards and Technology, Gaithersburg, USA*

<sup>2)</sup>*Maryland NanoCenter, University of Maryland, College Park, USA*

<sup>3)</sup>*Unité Mixte de Physique CNRS, Thales, Univ. Paris-Sud, Université Paris-Saclay, 91767, Palaiseau, France*

<sup>4)</sup>*Centre de Nanosciences et de Nanotechnologies, Univ. Paris-Sud, CNRS, Université Paris-Saclay, 91405, Orsay, France*

(Dated: 4 June 2018)

The brain, which uses redundancy and continuous learning to overcome the unreliability of its components, provides a promising path to building computing systems that are robust to the unreliability of their constituent nanodevices. In this work, we illustrate this path by a computing system based on population coding with magnetic tunnel junctions that implement both neurons and synaptic weights. We show that equipping such a system with continuous learning enables it to recover from the loss of neurons and makes it possible to use unreliable synaptic weights (*i.e.* low energy barrier magnetic memories). There is a tradeoff between power consumption and precision because low energy barrier memories consume less energy than high barrier ones. For a given precision, there is an optimal number of neurons and an optimal energy barrier for the weights that leads to minimum power consumption.

## I. INTRODUCTION

The small size and low energy consumption of nanoelectronics devices make them ideal components for novel forms of computing. However, these advantages come with drawbacks: device to device variability, possibly stochastic behavior, and sometimes, device failure. Biological systems, like our brains, give hope that these drawbacks can be overcome because they operate in noisy environments with components, neurons and synapses, that are variable and normally exhibit stochastic behavior.<sup>1</sup> Our brain is capable of recovery after massive loss of neurons, as can happen in an accident.<sup>2</sup> Two factors contribute to these abilities. First, the brain exhibits high redundancy, which is studied by neuroscientists within the framework of population coding.<sup>3-5</sup> Second, the brain never stops learning.<sup>6-8</sup> In contrast to most artificial neural networks in which the synaptic weights are set after an initial training phase,<sup>9</sup> synaptic weights in the brain constantly adjust to adapt to changes.

We show that equipping an artificial neural network with redundancy and, most importantly, continuous learning allows it to overcome the unreliability of nanodevices. We illustrate this idea with a system that uses population coding with stochastic magnetic tunnel junctions as spiking neurons and stable magnetic tunnel junctions to store synaptic weights.<sup>10</sup> We first present the system and summarize the relevant physics of magnetic tunnel junctions. Numerical simulations of this system show that continuous learning enables it to recover from the catastrophic loss of neurons (caused by the breakdown

of the tunnel barrier of junctions for instance), whether the loss occurs before or after training. With continuous learning, this type of system can learn in spite of unreliable synaptic weights. Here, magnetic tunnel junctions with energy barriers as low as  $15k_B T$  lead to the same system precision as perfectly reliable weights. Because writing low barrier junctions consumes less energy than writing high barrier ones, there is a tradeoff between power consumption and precision. For a given precision, there is an optimal number of neurons and weights and an optimal barrier for the weight junctions, for which power consumption of the system is minimal.

## II. DESCRIPTION OF THE SYSTEM

To illustrate the power of population coding, which provides intrinsic redundancy, we consider the computing system depicted in Fig. 1.<sup>10</sup> It is based on controllable-rate stochastic devices as neurons and (mostly) non-volatile devices to store synaptic weights as well as control circuitry realized with conventional electronic circuits. An analog input value is encoded by the firing rates of the population of spiking neurons.<sup>3-5</sup> This system can perform nonlinear transformations of the input value by connecting two populations linearly by the synaptic weights.<sup>11</sup> The rates of the output junctions, which encode the transformed value, satisfy:

$$\langle R_j^{\text{out}} \rangle = \sum_{i=1}^{N_{\text{in}}} W_{ij} \langle R_i^{\text{in}} \rangle \quad (1)$$

where  $N_{\text{in}}$  is the number of input junctions,  $R_j^{\text{out}}$  is the rate of the  $j$ -th output junction,  $R_i^{\text{in}}$  is the rate of the  $i$ -th input junction and  $W_{ij}$  is the synaptic weight connecting the  $i$ -th input junction to the  $j$ -th output junction. The

<sup>a)</sup>alice.mizrahi@nist.gov

value encoded by the output population is a nonlinear function of the value encoded by the input population, and this function depends on the synaptic weights. Details about the encoding are given in the Appendix.

Due to their unlimited endurance, magnetic tunnel junctions are promising devices to implement such systems where learning is key. They consist of two thin ferromagnetic layers separated by a tunnel barrier. The state of the device can be read through its tunneling magnetoresistance,<sup>12,13</sup> *i.e.* each magnetic configuration (whether the magnetizations in the layers are parallel (P) or anti-parallel (AP)) has a different resistance. The probability to switch from one state to another after a given time  $\Delta t$  is driven by a thermally activated Poisson process with escape rates  $\phi_{P/AP}$ .<sup>14,15</sup>

$$P_{P/AP}(\Delta t) = 1 - \exp(-\phi_{P/AP} \Delta t) \quad (2)$$

$$\phi_{P/AP} = \phi_0 \exp\left(-\frac{\Delta E}{k_B T} \left(1 \pm \frac{V}{V_c}\right)\right) \quad (3)$$

where  $\phi_0 = 10^9 \text{ s}^{-1}$  is the attempt frequency,  $\Delta E$  is the energy barrier between the two stable states,  $k_B$  is the Boltzmann constant,  $T$  is the temperature,  $V_c$  is the critical voltage and  $V$  is the voltage applied across the junction. High energy barriers lead to long retention of the state: junctions with high energy barriers function as binary non-volatile memory, such as in Magnetic Random Access Memories.<sup>16</sup> Low energy barriers cause the junction to randomly switch between the two states.<sup>14,17</sup> The switching rate can be controlled by applying a voltage across the junction through the phenomenon of spin transfer torque<sup>18,19</sup>, as captured by the voltage dependent factor in Eq. 3. This control of the spiking rate is reminiscent of sensory neurons and a key feature for population coding.<sup>10</sup>

Several recent approaches to novel computing use magnetic tunnel junctions in their stochastic regime.<sup>20–28</sup> The present system uses one single stack of materials to implement both neurons and synapses, as depicted in Figure 1. Small junctions have low barriers and emulate neurons, while large junctions have high barriers and store the synaptic weights (several junctions per weight are required since they are binary).

Figure 2 shows the results of numerical simulations illustrating learning with this system, for the case of perfectly reliable components. In this paper, we use a *sine* transformation to demonstrate this system because it is non-linear and non-monotonic. Learning is done by trial and error. At first the weights are random, which gives an error of about 29%. At each learning step a random input value is chosen. If the output is close enough to the desired value, the weights are left unchanged, otherwise, they are modified according to a learning rule detailed in the Appendix. The learning is evaluated by computing the error, set to be the mean distance between the value encoded by the output population and

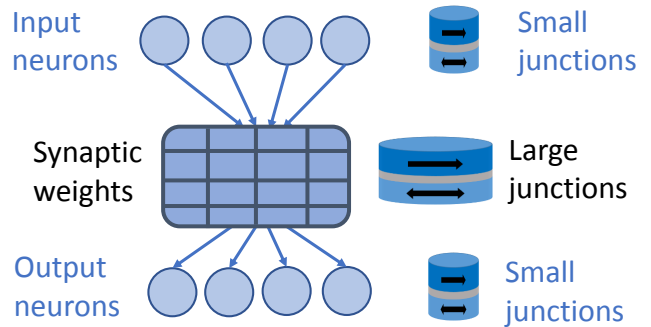


FIG. 1. Schematic of the considered system. Two neuron populations – emulated by small unstable magnetic tunnel junctions – are connected by an array of synaptic weights – stored in large stable magnetic tunnel junctions.

its target divided by the range of possible output values. In Fig. 2, the error is plotted versus the number of learning steps. It progressively decreases down to a final minimum error of approximately 2% of the output range. This minimum error depends among others on the number of junctions used and the range of voltages needed to tune the switching rates.<sup>10</sup> Parameters used in these simulations are given in the Appendix.

Ref. 10 uses circuit level simulations to demonstrate that a hardware implementation of this system would consume little energy only require a small area. Most importantly, the learning of the weights would consume sufficiently low energy and require sufficiently low area to be implemented on-chip and proceed continuously during the use of the system. The present results illustrate how continuous learning can overcome unreliability of components.

### III. ROBUSTNESS TO THE LOSS OF NEURONS

Magnetic tunnel junctions can occasionally fail, due to the breakdown of the tunnel barrier. To investigate the effects of such failure, we consider a system where both the input and output populations are composed of  $N = 100$  junctions. We model catastrophic component failure by selecting at random a fixed number of junctions in both the input and output populations and assigning them a null rate. The solid lines in Fig. 3(a) show how the error evolves with the number of learning steps when neuron failure occurs before any learning. The different colors correspond to different loss levels and thus to different resulting population sizes. As expected from standard population coding works<sup>11,29</sup> and Ref. 10, the final error is smaller for larger populations. The dashed lines in Fig. 3(a) and (b) show how the error evolves when neural loss occurs after initial learning. The system is initially trained with 4000 learning steps (reaching the steady state), and then, the neural loss is simulated. We

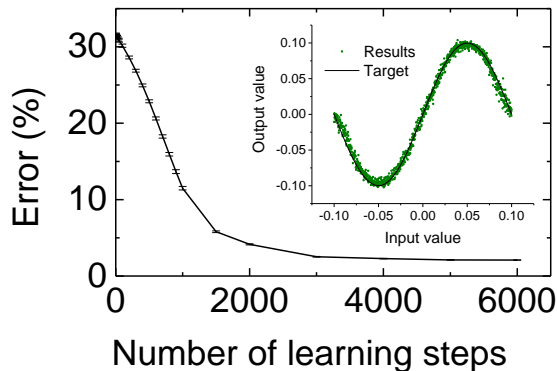


FIG. 2. Main panel: Error versus the number of learning steps for the *sine* transformation. Each population is composed of 100 neurons. The error is computed as the distance between the output value and target value and expressed in percentage of the range of output values. Each data point corresponds to an average over 50 learning processes and for each of them the error was calculated as an average over 50 different input values. The error bar indicates the single standard deviation of the error in the mean, computed over the 50 learning processes. Inset: output value versus input value. The black solid line is the target *sine* function and the green squares are the simulation results for a single learning process after 4050 learning steps.

observe that the loss increases the error of the system. However, this error is much lower than it was before the initial training (even with 80 % loss, the error is less than half of what it is without training).

After neuron loss, we continue to train the resulting system. We observe recovery, that is, the error decreases to a steady-state value that depends on the loss level. This steady-state value is the same as it would be for a system starting with that number of neurons as seen by comparing the recovery with the training of a system composed of a comparably reduced population of neurons, plotted as solid lines in Fig. 3(a). Furthermore, this state is reached much faster than through initial learning (a few hundred steps versus thousands). This rapid re-learning shows the ability of the system to adapt to radical changes by re-learning, building on previous knowledge. The results in this paper demonstrate how continuous learning makes our system resilient to the loss of neurons.

#### IV. ROBUSTNESS TO UNRELIABLE SYNAPTIC WEIGHTS

The main reliability concern for the synaptic weights  $W_{i,j}$ , is the ability of the devices to retain information. In the case of magnetic tunnel junctions, reducing energy consumption requires lowering energy barriers, which reduces the long term non-volatility of these devices. Quan-

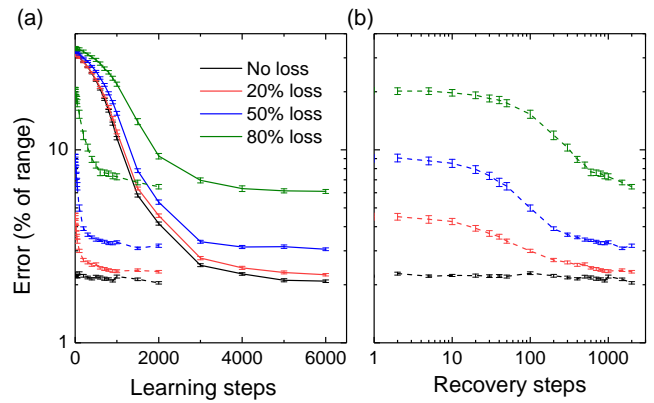


FIG. 3. Relative output error versus the number of learning steps. (a) Dashed lines: recovery after the loss of neurons. Solid lines: learning starting from a reduced population. (b) Recovery after the loss of neurons (enlargement of the dashed lines on panel (a)). The different colors correspond to different amounts of loss, on an initial system where input and output population each have 100 neurons. Each data point corresponds to an average over 50 learning instances and for each of them the error was estimated as an average over 50 different input values. The error bar indicates the single standard deviation of the error in the mean, computed over the 50 learning instances.

tifying the effect of this volatility on performance is crucial to understanding how the system behaves. First, we study the initial learning process for different energy barrier heights in the devices storing the weights. Each finite barrier height  $\Delta E_w$  leads to a probability  $P_{\text{loss}}$  for the weight to lose its information at each learning step. Because several binary junctions are necessary to store one weight, this loss probability is computed as follows:

$$P(\Delta t, \Delta E_w) = 1 - \exp\left(-N_{\text{bits}} \Delta t \phi_0 \exp\left(-\frac{\Delta E_w}{k_B T}\right)\right) \quad (4)$$

Where  $N_{\text{bits}}$  is the number of magnetic tunnel junctions (bits) per weight. We have observed that 8 bits are sufficient to store the weights with the same system precision as with real valued analog weights.  $\Delta t = 10 \mu\text{s}$  is the duration of the learning step. It consists on the observation phase where switches of the neurons are counted, followed by a computing phase in which equation 1 is evaluated. We neglect the length of the computing phase since digital electronics is very fast compared to the observation phase. Note that if the energy barrier of the neuron junctions was lower, the switching rates would be higher and therefore the required duration of the observation phase would be lower.

Fig. 4 presents the evolution of the error versus the number of learning steps, for various energy barrier heights. We observe that each case reaches a steady state: after sufficient initial learning, continuous learning enables the system to adapt to dynamical changes of the weights. The steady-state error decreases with the en-

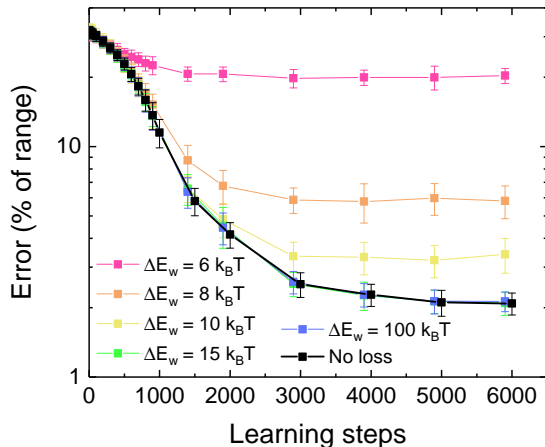


FIG. 4. Relative output error versus the number of learning steps, for various energy barrier heights. Each data point correspond to an average over 50 learning instances and for each point, the error was estimated as an average over 50 different input values. The error bar indicates the standard deviation, computed over the 50 learning processes. Note that the plots for  $\Delta E_w = 15k_B T$ ,  $\Delta E_w = 100k_B T$  and no loss overlap.

ergy barrier height (increases with the loss probability). When the energy barrier height is too low, the system is unable to learn. Energy barriers above  $15k_B T$  provide as good precision as perfectly reliable weights (“no loss” label, which would correspond to an infinite energy barrier), within the statistical uncertainties. For reference, here  $\Delta E_w = 15k_B T$  corresponds to an individual weight loss probability of  $1.5 \times 10^{-7}$  per learning step. We have assumed that the unexpected reversal of one magnetic tunnel junction leads to the synaptic weight taking a random value.

This complete randomization corresponds to a worst-case scenario as reversal of one of the less significant bits would only slightly change the weight. Furthermore, storing the weights in memristors or chained magnetic tunnel junctions would induce less drastic losses of the weights: drifts to close values rather than full randomization. Such implementations would make the system even more robust to synaptic unreliability.

## V. POWER CONSUMPTION VERSUS PRECISION TRADEOFF

We compute how the unreliability of the devices storing the synaptic weights affects the power consumption of the system during continuous learning (i.e. during the steady state observed in Fig. 4). The current, and therefore energy, required to write a magnetic tunnel junction scales with the critical current of the device, which is proportional to the energy barrier height.<sup>30</sup> We are thus

interested in studying the trade-off between the precision of the system and the power consumption. Similar trade-offs have been considered in the context of energy efficient faulty computing.<sup>31–33</sup> The power consumption of updating the synaptic weights depends on three factors: the rate at which the weights are modified during continuous learning (i.e. how often the target is missed), the number of synapses (i.e. the number of input neurons times the number of output neurons) and the power required to write a weight.

The power consumption will be proportional to the rate at which individual weights need to be updated, times the number that need updating, and the square of the energy barrier of the devices:

$$Power(N, \Delta E_w) \propto R_{\text{update}}(N, \Delta E_w) \times N^2 \times \Delta E_w^2 \quad (5)$$

$N$  is the number of neurons in each population, therefore there are  $N^2$  synapses.  $R_{\text{update}}$  is the rate at which the weights need to be updated.

We use numerical simulations to compute – for various numbers of neurons and energy barrier heights – both the error and the rate at which the weights must be updated. In each case, the power consumption is normalized by that of a system with 100 neurons in each population and an energy barrier of  $\Delta E_w = 20k_B T$ , which in our case is effectively the same as perfectly reliable weights. Fig. 5 (a) shows the normalized power consumption for various barrier heights and neuron numbers (different colors).

For each number of neurons (colored curves in Fig. 5(a)) the error and the power consumption vary with the energy barrier height. Increasing the number of neurons decreases the error because in larger systems the information carried by each synapse has a smaller relative importance. Therefore, the system is more robust to synaptic unreliability but requires greater power as there are more weights to update. Varying both the number of synapses and their reliability is necessary to minimize the power consumption required to reach a given error. This optimal trade-off between power consumption and precision is plotted in black squares in Fig. 5(a). The minimum power consumption for a given error corresponds to a specific number of neurons and energy barrier height, which are plotted in Fig. 5(b) and 5(c) respectively. As expected, lower targeted errors require more neurons and more reliable synapses.

These results provide guidance on designing a population coding based systems for particular applications and neurons with particular characteristics. Given the desired precision, one can compute the optimal number of neurons as well as the optimal synaptic reliability. For example, when building a system for applications in which 3 % error is acceptable, 54 stochastic magnetic tunnel junctions (neurons) in each population and magnetic tunnel junctions with an energy barrier  $\Delta E = 12k_B T$  as weights would provide the lowest power consumption. These values of the optimal energy barriers are very low compared to the values used for traditional memory applications: retaining the information of a single magnetic

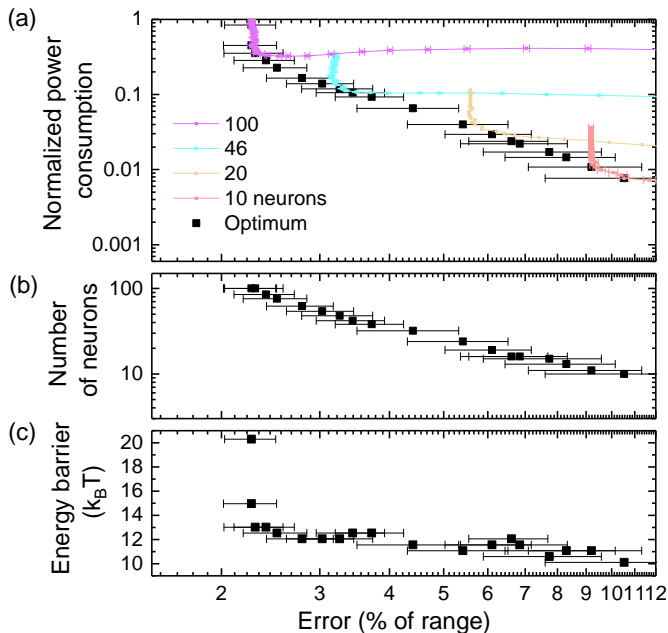


FIG. 5. (a) Normalized power consumption  $Power(N, \Delta E_w) / Power(100, 20k_B T)$ , plotted versus the error. Each color corresponds to the number of neurons in each population. For all curves in this figure each data point corresponds to an average over 200 learning processes and for each of them the error was estimated as an average over 50 different input values. The error bar indicates the single standard deviation of the error in the mean, computed over the 50 learning processes. For the colored curves, the error bar corresponds to a single standard deviation of the error in the mean. Only a few of the calculations for different numbers of neurons are shown. The black squares represent the minimum power consumption versus the error. The error bars for this curve correspond to the standard deviation of the distribution. (b)-(c) Number of neurons in each population (b) and energy barrier height (c) corresponding to the minimum power consumption to reach a given error. The error bars correspond to the standard deviation of the distribution.

tunnel junction for a few days with 99.9% certainty already requires energy barriers above  $40k_B T$ .

## VI. CONCLUSION

Equipping a nanodevice-based population coding system with continuous learning makes it able to adapt to changes and compensate for the unreliability of its components. Continuous learning is viable as it is low power and does not require much circuit area, thus allowing online learning while the system is active. Our results suggest that there is a tradeoff that can be tuned between precision and power consumption. We show that for a given desired precision, it is power efficient to use unreliable weights that are energy efficient to write. When using magnetic tunnel junctions to store the weights, this

translates into the possibility of using energy barriers much lower than those traditionally required for memory applications. Relaxing this constraint on the energy barrier should allow for easier device fabrication as well as improved performance with respect to other parameters.

These results are promising for hardware implementations of population coding schemes with magnetic tunnel junctions as neurons and synaptic weights. They should remain valid qualitatively independent of the devices used, highlighting the idea that using brain inspired approaches, such as continuous learning can allow a system with unreliable nanodevices to perform reliably.

## ACKNOWLEDGMENTS

A.M. acknowledges support under the Cooperative Research Agreement between the University of Maryland and the National Institute of Standards and Technology, Center for Nanoscale Science and Technology, Grant No. 70NANB10H193, through the University of Maryland. This work was supported in part by the European Research Council ERC under Grant bioSPINspired 682955.

## APPENDIX: DETAILS OF THE SIMULATIONS

For all simulations, the evolution of the state of each neuron junction (in both the input and output populations) is simulated using Eq. 2 and 3. The energy barrier of the neuron junctions  $\Delta E_n$  is so that  $\frac{\Delta E_n}{k_B T} = 6$  and the critical voltage is  $V_c = 0.1V$ . The input voltage  $V$  can take values between  $-0.1V$  and  $0.1V$ . In each neuron population, the junction of index  $i$  receives an individual bias voltage  $V_{0i}$  spanning from  $-0.15V$  to  $0.15V$  across the population. The rates of the neurons in each populations are computed by counting the number of switches over  $10\mu s$ . Computing the rate by counting stochastic switches makes it intrinsically inexact. The observation length leads to a tradeoff between energy consumption and precision as a longer observation consumes more but provides a better estimation of the rate. This tradeoff is typical of stochastic computing.<sup>34</sup> Here, for simplicity, we keep the observation length fixed through the paper.

The voltage  $V_{out}$  corresponding to the output of the system is deduced from them by a standard population decoding technique:<sup>11</sup>

$$V_{out} = \frac{\sum_{i=1}^{N_{out}} V_{0i} R_i^{out}}{\sum_{i=1}^{N_{out}} R_i^{out}} \quad (6)$$

The expression of the desired target output is:

$$V_{Target} = V_{max} \times \sin(V_{Input} / (V_{max} \pi)) \quad (7)$$

where  $V_{max} = 0.1V$  being the boundary of the input voltage range.

At each learning step, the learning rule is implemented as follows. If the output matches the target within a chosen range (i.e. the size of the target), the weights are not modified. In this paper we chose the size of the target to be 5% of the possible output values range. If the output is higher than the target: the weights connecting the input population to the junctions which bias voltages correspond to values higher than the output are decreased and the weights connecting the input population to the junctions which bias voltages correspond to values lower than the output are increased. If, in contrast, the output is lower than the target, the opposite is implemented.

The increase or decrease of the weights connecting the  $j$ -th output junction to the input population corresponds to:

$$W_{ij} \rightarrow \left( W_{ij} \pm \alpha \frac{R_i^{\text{in}}}{f_0} \right) \frac{1}{1 + \alpha} \quad (8)$$

where  $\alpha = 0.001$  is the learning rate.

- <sup>1</sup>A. A. Faisal, L. P. J. Selen, and D. M. Wolpert, *Nature Reviews Neuroscience* **9**, 292 (2008).
- <sup>2</sup>R. A. Mason, C. S. Prat, and M. A. Just, *Cerebral Cortex* **24**, 1474 (2014).
- <sup>3</sup>A. P. Georgopoulos, R. E. Kettner, and A. B. Schwartz, *Journal of Neuroscience* **8**, 2928 (1988).
- <sup>4</sup>A. Pouget, P. Dayan, and R. Zemel, *Nature Reviews Neuroscience* **1**, 125 (2000).
- <sup>5</sup>A. Pouget, J. M. Beck, W. J. Ma, and P. E. Latham, *Nature Neuroscience* **16**, 1170 (2013).
- <sup>6</sup>M. M. Merzenich, W. M. Jenkins, and S. Levy-Reiner, *The Adaptable Brain*, levy-reiner ed., Vol. Vol. II (Library of Congress, Washington, DC, 1999).
- <sup>7</sup>A. Pascual-Leone, A. Amedi, F. Fregni, and L. B. Merabet, *Annual Review of Neuroscience* **28**, 377 (2005).
- <sup>8</sup>S. B. Hofer, T. D. Mrsic-Flogel, T. Bonhoeffer, and M. Hübner, *Current Opinion in Neurobiology* *Sensory systems*, **16**, 451 (2006).
- <sup>9</sup>Y. LeCun, Y. Bengio, and G. Hinton, *Nature* **521**, 436 (2015).
- <sup>10</sup>A. Mizrahi, T. Hirtzlin, A. Fukushima, H. Kubota, S. Yuasa, J. Grollier, and D. Querlioz, *Nature Communications* **9**, 1533 (2018).
- <sup>11</sup>E. Salinas and L. F. Abbott, *The Journal of Neuroscience* **15**, 6461 (1995).
- <sup>12</sup>S. Yuasa, T. Nagahama, A. Fukushima, Y. Suzuki, and K. Ando, *Nature Materials* **3**, 868 (2004).
- <sup>13</sup>S. S. P. Parkin, C. Kaiser, A. Panchula, P. M. Rice, B. Hughes, M. Samant, and S.-H. Yang, *Nature Materials* **3**, 862 (2004).
- <sup>14</sup>W. Rippard, R. Heindl, M. Pufall, S. Russek, and A. Kos, *Physical Review B* **84**, 064439 (2011).
- <sup>15</sup>A. Mizrahi, N. Locatelli, R. Matsumoto, A. Fukushima, H. Kubota, S. Yuasa, V. Cros, J.-V. Kim, J. Grollier, and D. Querlioz, *IEEE Transactions on Magnetics* **51**, 1 (2015).
- <sup>16</sup>D. Apalkov, B. Dieny, and J. M. Slaughter, *Proceedings of the IEEE* **104**, 1796 (2016).
- <sup>17</sup>D. Vodenicarevic, N. Locatelli, A. Mizrahi, J. Friedman, A. Vincent, M. Romera, A. Fukushima, K. Yakushiji, H. Kubota, S. Yuasa, S. Tiwari, J. Grollier, and D. Querlioz, *Physical Review Applied* **8**, 054045 (2017).
- <sup>18</sup>J. C. Slonczewski, *Physical Review B* **71**, 024411 (2005).
- <sup>19</sup>D. Ralph and M. Stiles, *Journal of Magnetism and Magnetic Materials* **320**, 1190 (2008).
- <sup>20</sup>A. Mizrahi, N. Locatelli, R. Lebrun, V. Cros, A. Fukushima, H. Kubota, S. Yuasa, D. Querlioz, and J. Grollier, *Scientific Reports* **6**, 30535 (2016).
- <sup>21</sup>B. Sutton, K. Y. Camsari, B. Behin-Aein, and S. Datta, *Scientific Reports* **7**, 44370 (2017).
- <sup>22</sup>C. M. Liyanagedera, A. Sengupta, A. Jaiswal, and K. Roy, *Physical Review Applied* **8**, 064017 (2017).
- <sup>23</sup>K. Y. Camsari, R. Faria, B. M. Sutton, and S. Datta, *Physical Review X* **7**, 031014 (2017).
- <sup>24</sup>A. Fukushima, T. Seki, K. Yakushiji, H. Kubota, H. Imamura, S. Yuasa, and K. Ando, *Applied Physics Express* **7**, 083001 (2014).
- <sup>25</sup>W. H. Choi, Y. Lv, J. Kim, A. Deshpande, G. Kang, J.-P. Wang, and C. H. Kim, in *2014 IEEE International Electron Devices Meeting* (2014) pp. 12.5.1–12.5.4.
- <sup>26</sup>B. Parks, M. Bapna, J. Igbokwe, H. Almasi, W. Wang, and S. A. Majetich, *AIP Advances* **8**, 055903 (2017).
- <sup>27</sup>H. Lee, F. Ebrahimi, P. K. Amiri, and K. L. Wang, *AIP Advances* **7**, 055934 (2017).
- <sup>28</sup>N. Rangarajan, A. Parthasarathy, and S. Rakheja, *Journal of Applied Physics* **121**, 223905 (2017).
- <sup>29</sup>L. F. Abbott, *Quarterly Reviews of Biophysics* **27**, 291 (1994).
- <sup>30</sup>H. Sato, E. C. I. Enobio, M. Yamanouchi, S. Ikeda, S. Fukami, S. Kanai, F. Matsukura, and H. Ohno, *Applied Physics Letters* **105**, 062403 (2014).
- <sup>31</sup>K. Palem and A. Lingamneni, *ACM Transactions on Embedded Computing Systems* **12**, 1 (2013).
- <sup>32</sup>J. Han and M. Orshansky, in *2013 18th IEEE European Test Symposium (ETS)* (2013) pp. 1–6.
- <sup>33</sup>P. Korkmaz, B. E. S. Akgul, and K. V. Palem, *IEEE Transactions on Circuits and Systems I: Regular Papers* **55**, 2249 (2008).
- <sup>34</sup>A. Alaghi and J. P. Hayes, *ACM Transactions on Embedded Computing Systems* **12**, 1 (2013).