



ICFP M2 Internship Report

Spin-orbit torque magnetic tunnel junctions
for neuromorphic computing

Dashiell Harrison

April 2025 - June 2025

Internship supervisor :

PR. PIETRO GAMBARDELLA

Internship supervisor :

MARCO HOFFMANN

HOME UNIVERSITY :



HOST DEPARTMENT :

DMATL

Table of contents

Introduction	1
1 Nonlinear dynamics in magnetic tunnel junctions	2
1.1 Spin-orbit torque magnetic tunnel junctions	2
1.2 Free layer magnetization dynamics	3
1.2.1 Landau-Lifshitz-Gilbert equation	3
1.2.2 Spin-transfer torque	4
1.2.3 Spin-orbit torque	4
1.2.4 Voltage-controlled magnetic anisotropy	4
1.2.5 Pure SOT	5
1.2.6 Tunneling magneto-resistance	5
1.2.7 Switching probability	6
1.3 Reservoir computing	8
2 Methods	9
2.1 Experimental setup	9
2.2 Protocol	10
2.2.1 Handwritten digits database	10
2.2.2 Encoding	11
2.2.3 Training	11
2.2.4 Example	12
2.2.5 Optuna grid search	13
3 Results	14
3.1 Binary classification	14
3.1.1 First experiment	14
3.1.2 Grid search	15
3.2 Multinomial classification	17
3.2.1 Five emulated MTJs	17
3.2.2 Decomposition of an image in 8 features	17
Conclusion	20
Appendix	21
Abbreviations	23

Introduction

In recent years, artificial intelligence models have improved greatly. Today, these models mostly rely on matrix multiplications, which are carried out in parallel in a graphic processing unit and then transmitted to a central processing unit. The computation and transmission of the information introduces many bottlenecks between the input and output (see appendix 1). This inefficiency in the architecture leads to a high energy consumption and a glass ceiling for the theoretical power of deep neural networks [1]. This drives the effort to rethink neural networks and the hardware on which they operate[2].

The human brain is an excellent example of how powerful a deep learning model can become, instead of matrix multiplication it uses spike trains that go from neuron to neuron through synapses which keep the memory of past data. As it happens, the goal of neuromorphic computing is to build a brain-like machine [3]. This defines the larger context of this report.

Neuromorphic computing can take various forms and is primarily based on materials that display memory of its past dynamics (e.g. hysteresis), which, in turn, can be leveraged to reproduce synaptic weight. Some examples of devices include memristive oxides [4], superconducting circuits [5], and magnetic tunnel junctions (MTJ) [6]. The work presented here investigates the latter. As for the architecture, while many companies are already marketing neuromorphic chips (Loihi 2 by Intel, TrueNorth by IBM, SpiNNaker by Dresden University, ...), it is still an active field of research. The implementation presented in this report is through reservoir computing [7].

Reservoir computing originates in 2001 under the name of "Echo State Network" [8], in which only the output layer of the neural network is modified to achieve learning tasks. The reservoir processes the information through a fixed network of nonlinear units. Mathematically, a reservoir is a "nonlinear generalization of standard bases" [9]. In part 1, we will explain how a single magnetic tunnel junction can be considered as a reservoir. Although benchmark tests for reservoir computing rather involve time series predictions [10], many other tests have been performed in the past to evaluate learning efficiency in reservoir computing such as audio recognition tasks [11], signal recognition tasks [12], and computer vision tasks both when the reservoir is an algorithm [13] or a simulated skyrmion reservoir [14]. In part 2, we will show how we put our MTJ to the test with handwritten digit recognition tasks. We transformed these digits into voltage pulses that, when applied to the MTJ, dynamically change its state. Finally, in part 3, we will present intermediate results.

Part 1 :

Nonlinear dynamics in magnetic tunnel junctions

1.1 Spin-orbit torque magnetic tunnel junctions

A magnetic tunnel junction is made up of two magnetic *CoFeB* layers separated by a *MgO* tunnel barrier. In figure 1.1a, the upper red layer is called the reference layer (RL) because its magnetization (symbolized by a bold arrow) is pinned by its exchange interaction with the synthetic anti-ferromagnet (SAF) layer above it (figure 1.1b). Conversely, the lower blue layer's magnetization (fine arrow) is easier to switch : this is the free layer (FL).

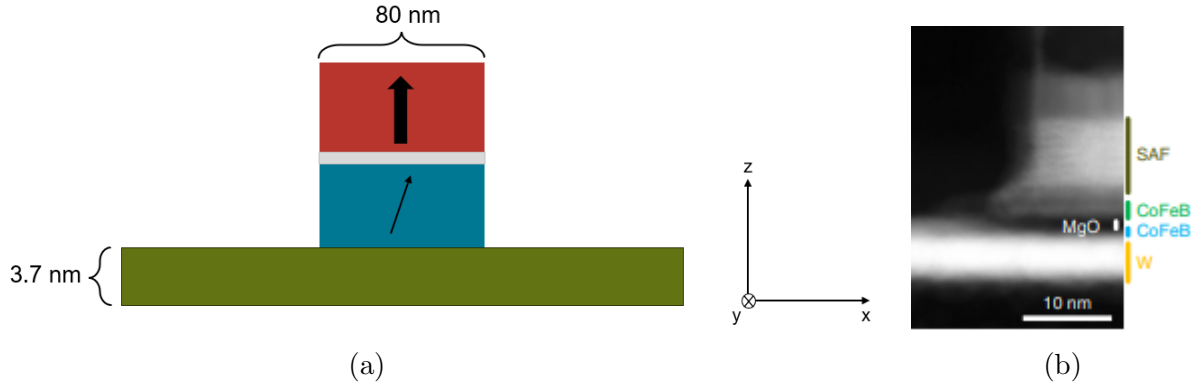


Figure 1.1: (a) Outline of a spin-orbit torque magnetic tunnel junction and (b) Scanning Electron Microscope image of an SOT MTJ from Grimaldi et al., *Nature nanotechnology* 15.2 (2020), pp. 111–117 [15].

The Interuniversity Microelectronics Centre (Imec) produces the specific type of MTJ studied in this report. They are composed of an additional Tungsten track (in green in 1.1a) under the FL that can assist the switching by spin-orbit torque effect (see part 1.2.3). These devices have been engineered by Imec to stay in stable states for non-volatile memory applications in MRAMs. Therefore these MTJs can be in one of two states. If the free (blue) layer's magnetization is parallel along $+\mathbf{z}$ with the reference (red) layer's magnetization, it is the P state. If they are anti-parallel, it is the AP state. In order to understand how to switch from one state to the other, we can dive deeper into the underlying physics of magnetization dynamics.

1.2 Free layer magnetization dynamics

This section draws heavily from the following references : [16], [17], [18].

Magnetization is the density of magnetic moments. We can consider the magnetization of our FL as a single magnetic moment or as a collection of many magnetic moments. Regardless, the following equations apply to it.

1.2.1 Landau-Lifshitz-Gilbert equation

The FL magnetization dynamics is governed by the Landau-Lifshitz-Gilbert (LLG) equation :

$$\frac{d\mathbf{M}}{dt} = -\gamma\mathbf{M} \times \mathbf{B} + \frac{\alpha}{M_s}\mathbf{M} \times \frac{d\mathbf{M}}{dt} + \gamma\mathbf{T} \quad (1.1)$$

where :

- \mathbf{M} is the magnetization [$A.m^{-1}$]
- γ is the absolute value of the gyromagnetic ratio [$s^{-1}.T^{-1}$]
- \mathbf{B} is the effective magnetic field [T]
- M_s is the magnetization saturation value [$A.m^{-1}$]
- α is the phenomenological Gilbert damping parameter [\emptyset]
- \mathbf{T} represents all the torques that do not derive from a magnetic energy density

The first part of the equation is just the rotational version of Newton's second law. Indeed, a magnetic moment is first of all an angular momentum and a magnetic field torques it as $-\mathbf{m} \times \mathbf{B}$. In the present case, \mathbf{B} comes from the upper magnetic material and all applied fields. Hence the first part.

The second part of the equation is phenomenological. Indeed, solving the first part results in a magnetic moment precessing around the effective field with a frequency of $f = \frac{\gamma B}{2\pi}$ called the Larmor frequency. Phenomenologically, the magnetic moment relaxes and aligns with the field at some point. Thus a damping parameter is introduced to take into account this relaxation due to, for example, interactions with the lattice.

The third part of the equation repeats the first two for torques that, unlike \mathbf{B} do not originate from the magnetic energy density, these are the torques that one leverages to act on the state of the MTJ :

$$\mathbf{T} = \tau_{FL}\mathbf{M} \times \boldsymbol{\xi} + \frac{\tau_{DL}}{M_s}\mathbf{M} \times (\mathbf{M} \times \boldsymbol{\xi}) \quad (1.2)$$

where :

- τ_{FL} is the field-like torque coefficient
- $\boldsymbol{\xi}$ is a unit vector that depends on the macroscopic origin of the torque

- τ_{DL} is the damping-like torque coefficient

In this framework, we can explain how to switch the FL of a MTJ.

1.2.2 Spin-transfer torque

By applying a voltage V_{STT} across the junction itself, one can switch the magnetization of the FL by spin-transfer torque (STT). A current of spin-polarized electrons from the RL will flow through the MgO by tunneling into the FL. The spins that they carry exert torque onto the FL spins in the way described by equation 1.2. In STT, ξ is the polarization vector and is oriented along the magnetization direction of the RL. For STT-MRAMs, this direction corresponds to the \mathbf{z} -axis in figure 1.1a.

1.2.3 Spin-orbit torque

By adding a third terminal to the device in figure 1.1a at the beginning of the Tungsten track, one can apply a current flowing through the non-magnetic metal. By spin Hall effect¹ [20], the electrical current produces a spin current that flows upwards to the interface with the FL (see figure 1.2). These spins torque the magnetization of the FL through an effect called spin-orbit torque (SOT). This is described by equation 1.2 where ξ is determined by the charge-spin conversion process and is typically oriented along the \mathbf{y} -axis in 1.1a .

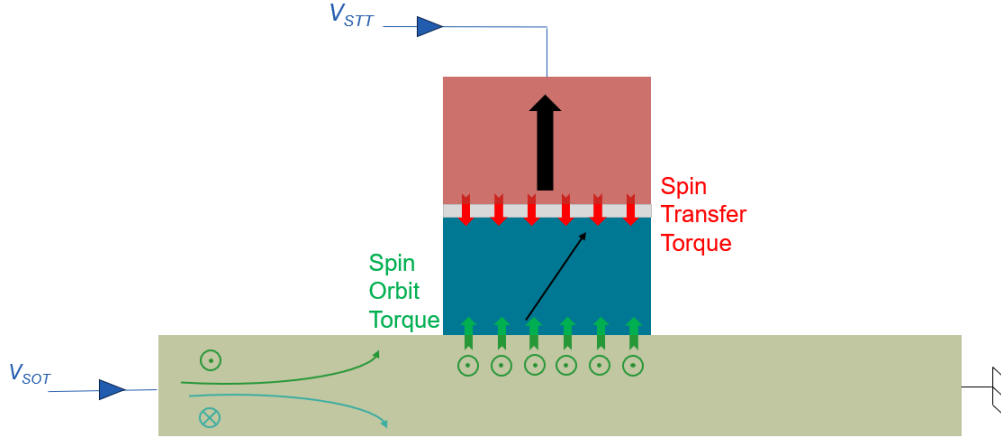


Figure 1.2: Three-terminal SOT MTJ and the torques assisting the switching of the FL.

1.2.4 Voltage-controlled magnetic anisotropy

Moreover, the ratio of the applied STT voltage and the applied SOT voltage :

$$x = \frac{V_{STT}}{V_{SOT}} \quad (1.3)$$

¹The spin Hall effect is just an example here. It is not the only effect that can produce a spin-polarized current and many others are at play [19].

can be changed to promote or inhibit the FL magnetization switching[15]. This is due to STT effects and voltage-controlled magnetic anisotropy (VCMA) : when a voltage is applied to the MTJ (through V_{STT}), it acts as a bias voltage that displaces ions at the interface with the MgO layer. This decreases or increases the perpendicular magnetic anisotropy (PMA) of the FL. Stronger (weaker) PMA implies that a higher (lower) V_{SOT} is needed to switch. Adjusting the ratio x further tunes the FL switching.

1.2.5 Pure SOT

One can also choose to set $V_{STT} = 0$ V. In that case, the switching is purely due to the V_{SOT} . Practical implementation of pure SOT current is done by imposing a geometry-related fixed value of the x ratio (see part 2). In this case, however, the FL magnetization may terminate in both equilibrium states with equal probability, either up or down along \mathbf{z} -axis in figure 1.1a. To break this symmetry, one can introduce a gradient of atom concentration, an anisotropy gradient, a bent track, or anything that favors one configuration over the other. In the present device, a magnetic field along the current direction (the \mathbf{x} -axis) breaks the symmetry. In this pure SOT case, the relative sign of the SOT voltage and magnetic field determines which switching we are attempting. These cases are summarized in figure 1.3.

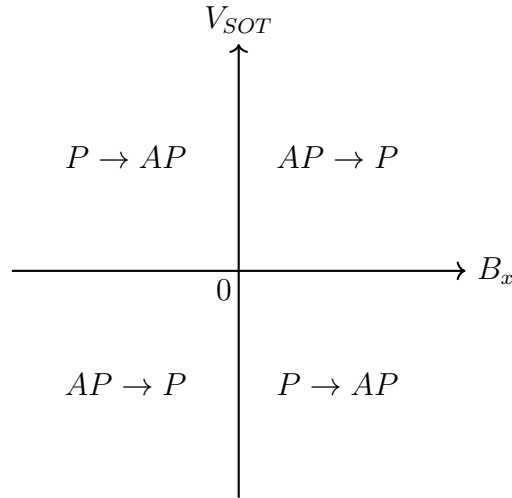


Figure 1.3: Switching between MTJ states.

1.2.6 Tunneling magneto-resistance

The state of the MTJ (P or AP) changes its resistance through tunneling magneto-resistance (TMR). Qualitatively, the resistance is lower if both the RL and FL have the same magnetization, as electrons with identical spins will flow more easily from one to the other. However, this is not always true. Quantitatively, one can specifically measure the spin-polarized electronic density of states in both layers and find the tunneling probability in each configuration. It can happen that the AP configuration is less resistive. It is not

the case for the device studied here. The relative resistance difference between one state and the other is described by:

$$TMR = \frac{R_{AP} - R_P}{R_P} \quad (1.4)$$

where :

- TMR is the relative tunneling magneto-resistance [%]
- R_{AP} is the AP state resistance [Ω]
- R_P is the P state resistance [Ω]

The higher the TMR, the easier it is to detect it.

For our samples, which rely on PMA along the \mathbf{z} -axis, the TMR is directly measured by applying an incremental magnetic field along the \mathbf{z} -axis to record a hysteresis loop (see figure 1.4).

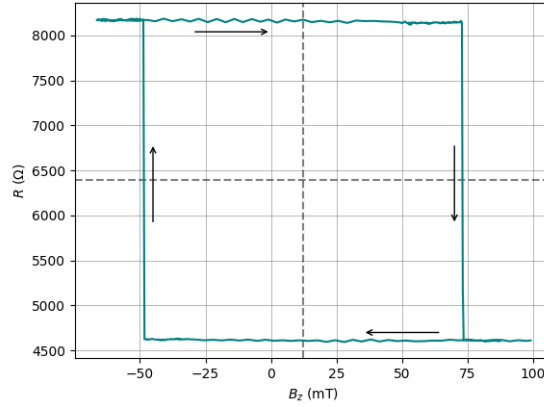


Figure 1.4: Complete hysteresis loop of a single MTJ. $TMR = 79\%$

We can notice in figure 1.4 that the center of the loop is offset by $+12.26$ mT : this is the opposite value of the pinning field produced by the SAF mentioned in part 1.1. In order to compensate for it, the applied field required to change the FL magnetization is higher in that direction, hence the offset.

Detecting a MTJ switch is detecting a difference in TMR : the resistance read at the center of the hysteresis loop is our reference point to know if we are in P or AP state.

1.2.7 Switching probability

These nonlinear dynamics lead the MTJ to switch probabilistically. The switching probability can be described as a sigmoidal of the applied voltage[19]. This sigmoidal changes significantly with respect to the x ratio, the applied magnetic field (for pure SOT) and the pulse's temporal width.

In particular, the pulse's temporal width has been thoroughly investigated by Grimaldi and colleagues [15]. It has been shown that the nucleation of the reversal of a magnetic

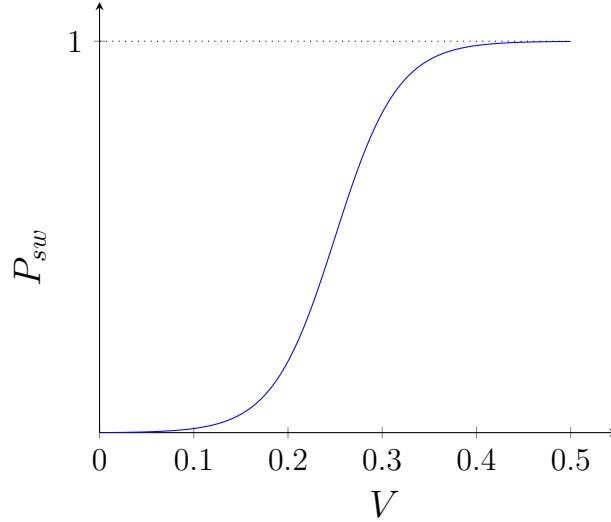


Figure 1.5: Switching probability as a function of the pulse voltage.

domain is closely related to the pulse duration. Without nucleation, the FL doesn't switch. Once the nucleation is initialized, it propagates through the FL which will switch on a nanosecond time scale. This is also the time scale corresponding to local heating. Indeed, applying a voltage to an MTJ heats locally the FL. Heating allows for easier switching. Among other physical quantities, the magnetic anisotropy is the most affected by it :

$$K(T) = K_0 \left(1 - \frac{T}{T_c}\right)^{2.5} \quad (1.5)$$

and a weak magnetic anisotropy means that the configuration is less stable and enables easier switching.

In pure SOT, the interplay between the pulse's voltage V_{SOT} , time duration t_q , and symmetry-breaking magnetic field B_x already allows for exceptional tunability of the FL switching probability. This is why the switching probability, arising from nonlinear interactions inside the MTJ, will be the reservoir state of the MTJ.

1.3 Reservoir computing

In the introduction, reservoir computing was described as a neural network of which only the last layer is trainable whilst the bulk is a reservoir of nonlinear units. Here, our reservoir is a single MTJ composed of many magnetic domains, connected to each other by nonlinear interactions governed by the LLG equation.

In figure 1.6, the group of green nodes is the reservoir : the MTJ. The group of blue nodes is the input : a handwritten image. The image is mapped to the reservoir using a fixed matrix W^{in} . The mapping is done by transforming the image into a sequence of voltage pulses (cf. part 2.2.1). The physical input of the reservoir then modifies the reservoir state : the scalar switching probability.

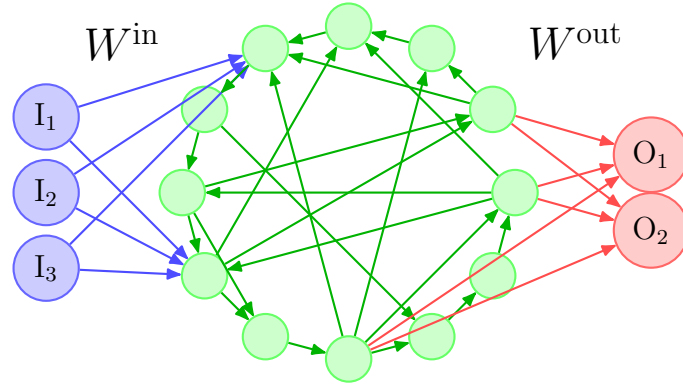


Figure 1.6: Reservoir computing architecture

The switching probability is then mapped by W^{out} back onto the output layer : target vectors of our handwritten digits. W^{out} is the only trained layer. It is obtained by minimizing a loss function. The study used the following [9]:

$$W^{\text{out}T} = \min_w (||\mathcal{P}_{sw}w - y||^2 + \beta ||w||^2) \quad (1.6)$$

where :

- \mathcal{P}_{sw} is the switching probability
- y is the 1×10 target vector with 0 everywhere except at the digit position (onehot vector)
- $\beta \in [0, 1]$ is a regularization parameter

For $\beta = 0$, minimizing the above loss function amounts to computing a pseudo-inverse matrix. With $\beta \neq 0$, minimizing the above loss function is called a ridge regression.

Part 2 : Methods

2.1 Experimental setup

The setup (figure 2.1) starts with a 25 GHz-bandwidth arbitrary waveform generator (AWG) from which two pulses V_0 and V_1 are sent with no relative delay. The pulses can be observed on a 10 GHz-bandwidth scope (see figure 2.2) thanks to pick-off tees. The setup continues with bias-tees which enable resistance measurements through the 10 Hz PXI (almost direct current) and the 1 k Ω reference resistance (of the same order of magnitude as the MTJ's resistance, cf. figure 1.4).

To prevent the insulator layer from burning when applying excessive voltage, we impose $V_{STT} = 0$ V. To do so, since the MTJ is located at the middle of the SOT track and $V_{out} = 0$ V at the end of the track, we need to apply :

$$V_1 = \frac{V_0}{2} \quad (2.1)$$

in other words $x = 0.5$.

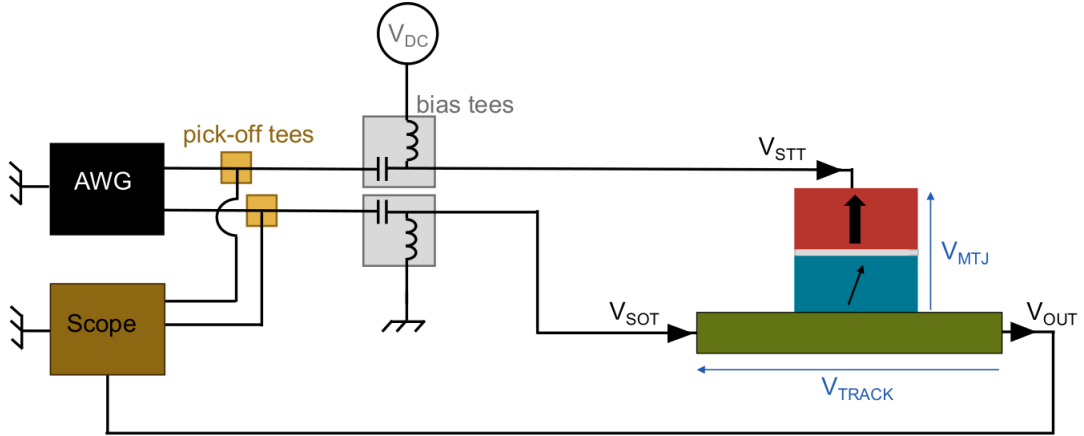


Figure 2.1: Experimental setup diagram. When $\frac{V_1}{V_0} = 0.5$ we have pure SOT switching.

Pure SOT switching means that we do not take advantage of any STT or VCMA effect that would inhibit or promote switching. However, these effects can be implemented by changing the value of the ratio x .

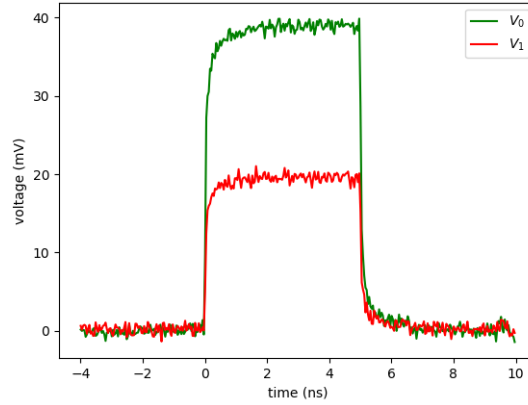


Figure 2.2: Example of 5 ns pulse sent from the AWG and read on the oscilloscope with $V_0 = 0.45 V$ and $x = 0.5$.

After the pulse, measuring the resistance of the MTJ with the PXI gives access to the state (either P or AP) of the MTJ.

2.2 Protocol

This section describes the database, encoding for the input layer, and training of the output layer.

2.2.1 Handwritten digits database

The MNIST database of handwritten digits is a common benchmark for computer vision tasks (see [13] and [14]). However, we used the simpler version from the scikit-learn library in Python. It is composed of 1797 total digits ranging from 0 to 9 each represented by a 64-pixel grayscale array (see figure 2.4). These digits are smaller than the 784 pixels per digit of the actual MNIST, but the methods exposed here are transposable to the MNIST database.

Moreover, for the first experiment, an even simpler database was created : horizontal and vertical bars inscribed in a 2x2 array.

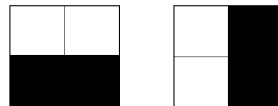


Figure 2.3: First experiment database classes.

2.2.2 Encoding

Each grayscale array is converted into a spike train through a type of rate coding scheme [21].

First, a quantification step t_q is defined. Each pixel of the handwritten image represents a pulse of duration t_q .

Second, each pixel digital value is treated as a voltage analog value. The pixel array becomes effectively a t_q -pulse sequence with each voltage corresponding to the pixel intensity.

Either the sequence is applied to the MTJ as is, or it goes through a preprocessing filter before. Sending an entire array as a pulse sequence might invariably switch the MTJ and prevent it from correctly classifying the different images. The filters tried were :

- "averaging" filter averages every 8 pixels into 1 pixel intensity to have a shorter pulse
- "sparse" filter picks out randomly 60 % of the pixel values to make a sequence of the same length but with only 60 % non-zero values
- "long" random filter which multiplies the array by a 1×100 random matrix with values between -1 and 1
- "short" random filter which multiplies the array by a 1×8 random matrix with values between 0 and 1

2.2.3 Training

The spike train is then sent \mathcal{N} number of times from the AWG to the MTJ. Each time the MTJ switches, a reset pulse is sent to reset the MTJ state. After the \mathcal{N} attempts, the MTJ will have switched N_{sw} many times. Thus, the MTJ's switching probability \mathcal{P}_{sw} associated with this spike train writes :

$$\mathcal{P}_{sw} = \frac{N_{sw}}{\mathcal{N}} \quad (2.2)$$

The switching probability is the final state of the reservoir that corresponds to the spike train.

The W^{out} is then trained by minimizing the loss function 1.6. This is rigorously done by the scikit-learn modules Linear Regression (for $\beta = 0$) and Ridge Regression (for $0 < \beta < 1$). If the switching probability becomes a vector (see 3.2), and not a scalar anymore, these modules still work perfectly.

2.2.4 Example

The number 3 from the scikit-learn library can be plotted as a 8×8 grayscale pixels image.

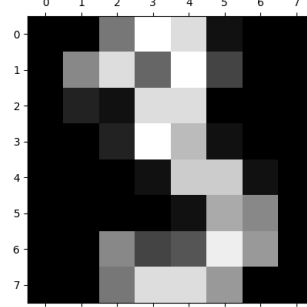


Figure 2.4: Number 3 from scikit-learn handwritten digits dataset.

In Python, it will be represented as a 1×64 matrix, each matrix value carries the grayscale value of the image.

$$[0. \ 2. \ 9. \ 15. \ 14. \ 9. \ 3. \ 0. \ \dots] \quad (2.3)$$

The matrix is then normalized and multiplied by the maximum pulse voltage. Henceforth, it becomes a $64 \times t_q$ voltage sequence.

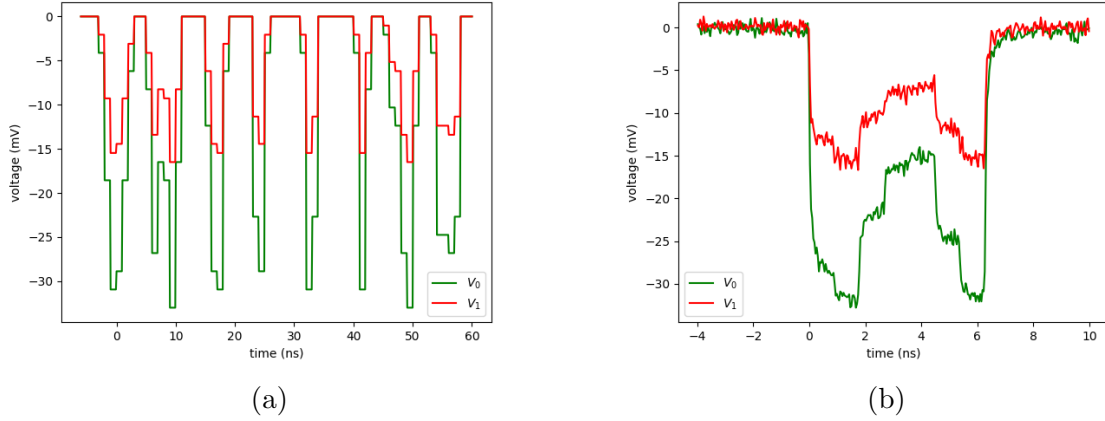


Figure 2.5: (a) theoretical full pulse of the handwritten 3 and (b) "average"-filter preprocessed signal read on the oscilloscope and sent to the MTJ.

$V_0 = -0.36 \text{ V}$, $t_q = 0.9 \text{ ns}$ and $x = 0.5$.

To leverage the MTJ's nonlinearity, the pulses go through one of the aforementioned preprocessing filters before being applied to the MTJ.

2.2.5 Optuna grid search

The probability of switching of the MTJ is sensitive to V_{SOT} , t_q , B_x . These are the hyperparameters of the reservoir. To find the best combination of these hyperparameters, one can implement a grid search algorithm. The one used in this study is the hyperparameter optimization framework Optuna implemented in Python. Optuna uses sequential algorithms that optimize the grid search [22]. The specific algorithm used in this study is a tree-structured Parzen estimator [23]. On each trial, for each parameter, the tree-structured Parzen estimator fits one Gaussian mixture model $l(x)$ to the set of parameter values associated with the best accuracies, and another Gaussian mixture model $g(x)$ to the remaining parameter values. It chooses the parameter value x that maximizes the ratio $l(x)/g(x)$.

Part 3 :

Results

3.1 Binary classification

In all the following, the accuracy given by the ridge or the linear regression was the same. No cross-validation or confusion matrix and no other score than the accuracy was implemented. Switching attempts were from P to AP with a positive magnetic field, hence negative pulse voltages 1.3.

3.1.1 First experiment

The first experiment data set consisted of 100 samples of horizontal and vertical bars with a test ratio of 20%. The switching probabilities were determined with $\mathcal{N} = 100$ attempts. The values of the field, the voltage and the quantification step were all tuned by hand. The goal was to separate the vertical from the horizontal bar on the \mathcal{P}_{sw} scale. The best results corresponded to a sub-ns quantification step at low magnetic field and high voltage. This takes advantage of the heating timescale of the FL : the vertical bar doesn't heat the FL enough to make it consistently switch while the horizontal bar corresponds to a super-ns pulse which heats the FL enough to switch its magnetization.

The separation between the two classes is so clear that one can observe it with his bare eyes (see figure 3.1 and 3.2). Unsurprisingly, the accuracy score is therefore 100 % .

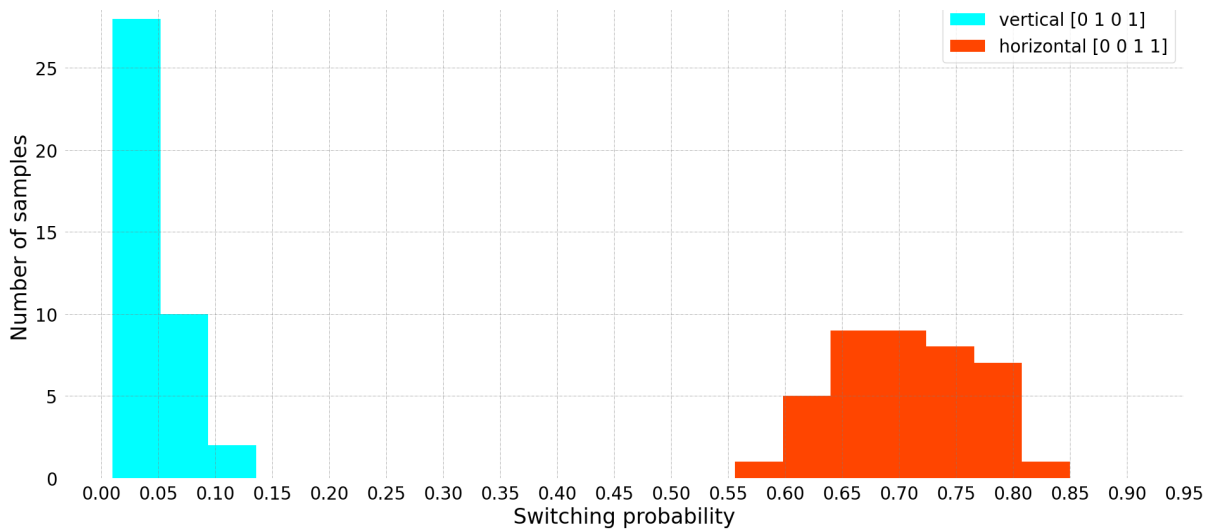


Figure 3.1: Training results of the first experiment with $\mathcal{N} = 100$.

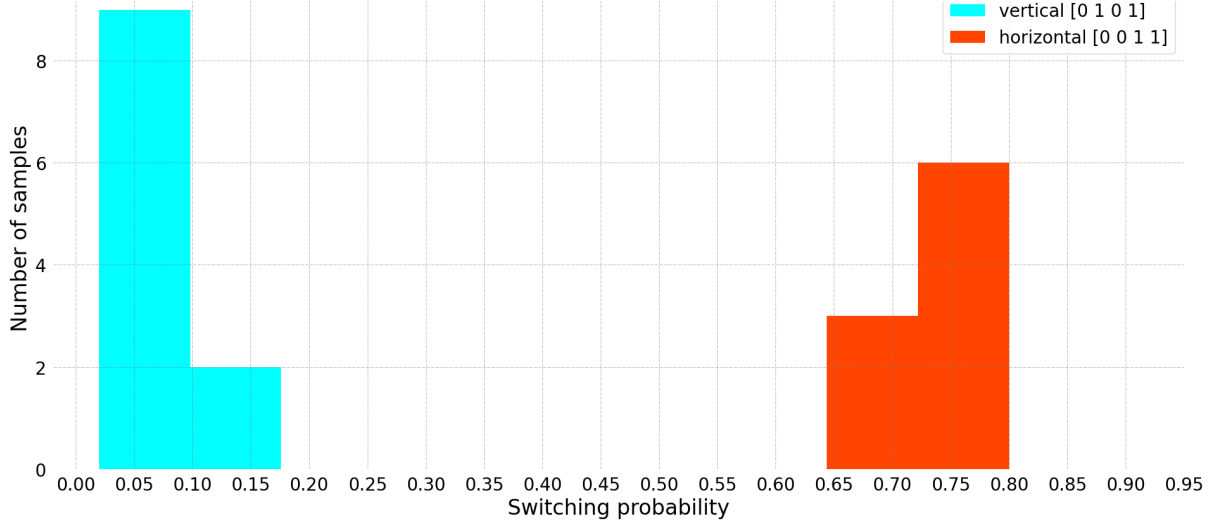


Figure 3.2: Testing results of the first experiment with $\mathcal{N} = 100$.

These preliminary results show that a single MTJ can classify two simple shapes by projecting each on a different value range between 0 and 1. This only indicates that the MTJ is a good representation of these shapes but doesn't give information on the learning capability of the whole system. We can put it to the test with a harder binary classification task : classifying two numbers of the scikit-learn digits database.

In the following, all the switching probabilities were determined with $\mathcal{N} = 10$ attempts for time-saving reasons.

3.1.2 Grid search

The next data set consisted of 100 samples with a 20 % test ratio of 3 and 8 digits of the scikit database. Finding the correct hyperparameters for these more complex sequences was done by the Optuna grid search algorithm. The search went over 250 different combinations of values (each called a trial) from the following grid :

- a magnetic field B_x between 0.5 A and 2.5 A corresponding to a physical range of [8.35, 41.75] mT (1 A is 16.7 mT)
- a maximum voltage V_{max} between 0.3 V and 0.9 V corresponding to a physical range of [0.15, 0.45] V (physical voltage value divided by 2 compared to the plot 3.3 for setup reasons)
- a quantification step t_q in the range of [0.5, 7.5] ns
- a choice of preprocessing filter between all the available ones (see part 2.2.2)

The best accuracy the algorithm found was 85 % . However, 7 other trials gave 80 % accuracy. On a training dataset of 20 samples, this corresponds to a single sample difference. Without any cross-validation it is hard to say if this is a true difference or just the regression method that has a 50 % chance of assigning the correct target to its

reservoir state.

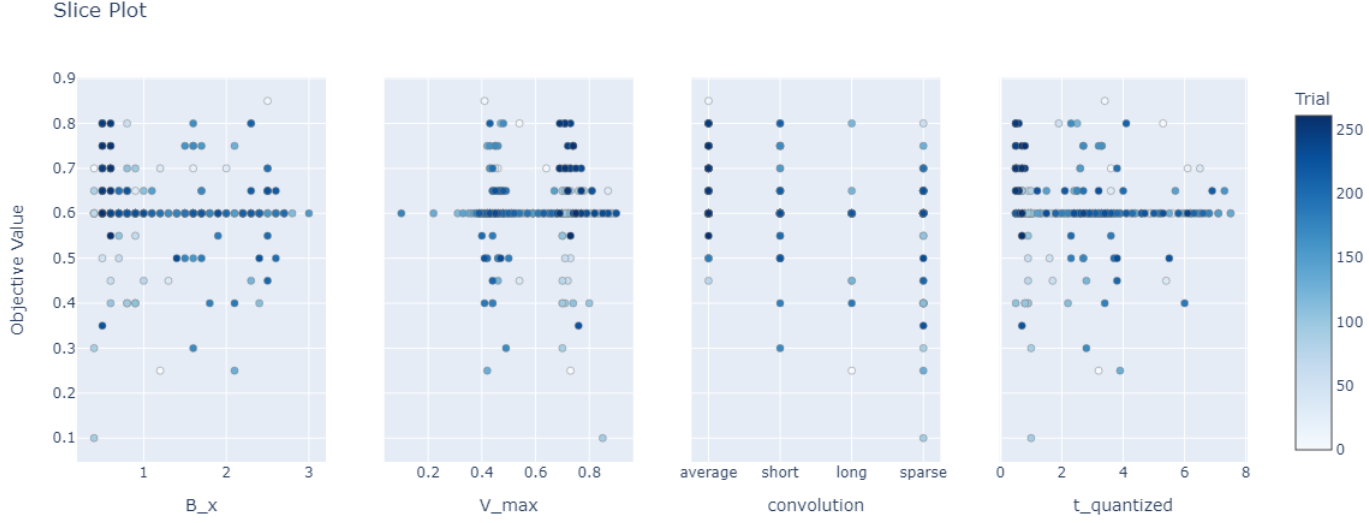


Figure 3.3: Accuracy (objective value) as a function of the hyperparameter.

The Optuna search algorithm showed no tangible difference between each pre-processing filter, except for the "short" one which is *a priori* less effective. The accuracy ceiling might also be due to a poor choice of filters.

However, the search showed that longer than 5 ns quantification steps and lower than 20 mV voltages seem to render a lower accuracy. For low voltages, the FL does not switch and therefore the MTJ can't discriminate two distinct signals. For long quantification steps, it doesn't take advantage of heating in the short time scales and might make the FL switch too frequently such that the MTJ still can't separate two distinct signals.

Concerning the influence of the magnetic field, figure 3.4 displaying the contour plots of the hyperparameters shows that we could infer the existence of two regions yielding higher accuracies. In one region, the quantification step can be very short: in that case the weak magnetic field doesn't assist the switching considerably and the switching mechanism is mainly due to heating, different signals will heat the FL differently and it will separate more efficiently different types of signals. In the other region, the quantification step is longer: in that case a stronger magnetic field assists the switching which isn't discriminated through heat anymore.

Simultaneously, a shorter quantification step calls for a higher voltage, so high voltages go with weak fields, while a stronger field assists the switching enough such that low voltages go with long quantification steps.

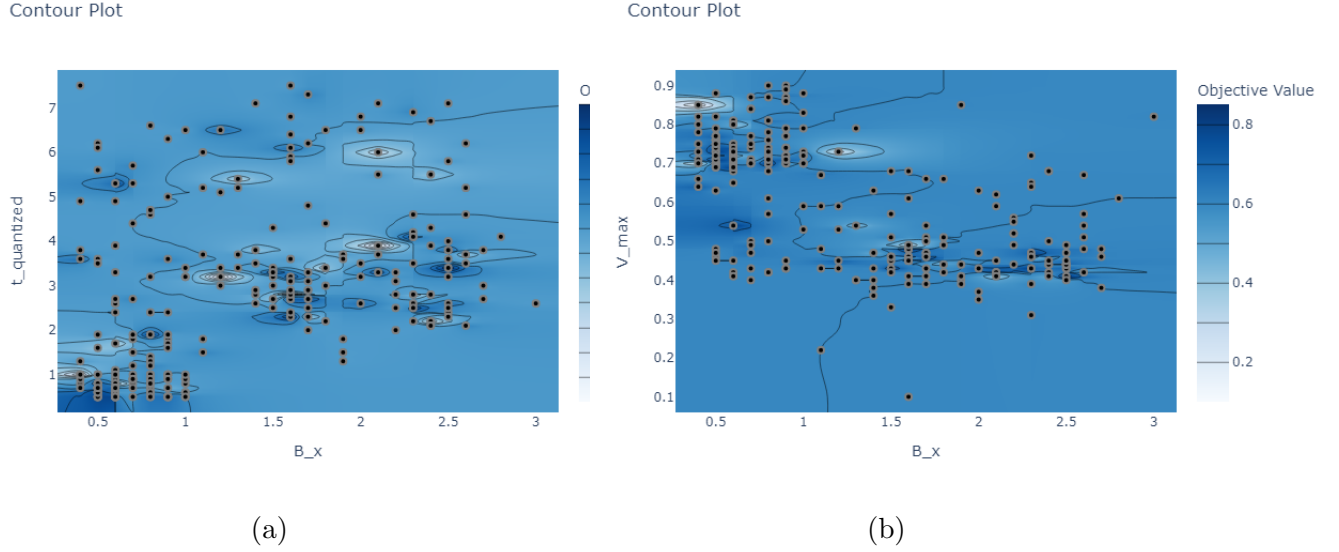


Figure 3.4: Darker blue regions correspond to higher accuracy. Dots are the actual combinations (trials) that were tested. (a) is the quantification step plotted with the field and (b) is the voltage plotted with the field.

3.2 Multinomial classification

The grid search shows that a scalar switching probability might not be enough to classify the elements of the handwritten digits database. To add features to the training data set, one can assign a switching probability vector to each digit. The regression model can then predict the targets of the testing data set by using not one but several distinct switching probabilities. The following section describes attempts to classify the ten different digits of the scikit database.

3.2.1 Five emulated MTJs

The first way to test this hypothesis is to emulate MTJs with different hyperparameters each. This effectively expands the size of our reservoir. For this experiment, the data set consisted of 100 samples and 20 % test ratio all filtered by the "averaging" preprocessing filter over sequences of 8 pixels. The data set was sent through the single MTJ 5 times, changing the system's hyperparameters (t_q , V_{max} , B_x) for each time. The resultant accuracy was 10 %, which is exactly random guessing. This method doesn't seem like the path to follow. For a plot of the results, please see appendix 2.

3.2.2 Decomposition of an image in 8 features

The data set for the last experiment consisted of 250 samples with a test ratio of 20 %. In this experiment, each 1×64 matrix is decomposed into eight 1×8 matrices which all yield a switching probability. This assigns 8 scalars to each digit and the reservoir state

is now an 1×8 vector of switching probabilities.

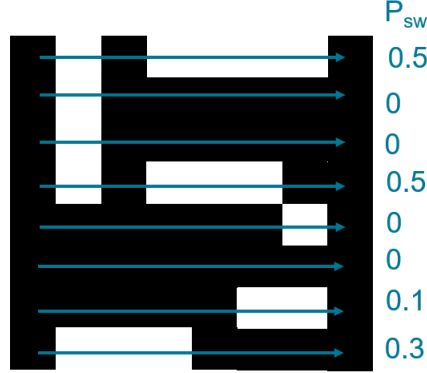


Figure 3.5: Schematic example of assigning 8 switching probability values to a digit 5 of the database.

The values chosen for this experiment were those that yielded 80 % accuracy in the Optuna grid search experiment in the weak B_z region.

The result is an accuracy of 40 % which is much better than the 10 % random guessing of our first attempt to classify all the digits. In figure 3.6, the probabilities are much more spread out than in appendix 2 (emulated MTJs). This may explain in part why the accuracy is better. The fact that there are 8 features and not 5 also helps increase the accuracy.

To improve the accuracy, a grid search would help adjust the MTJ hyperparameters. There is room for improvement in that regard because many of the 8-sequence lines still switch all the time ($\mathcal{P}_{sw} > 0.9$). By expanding the training data set, the accuracy might also increase. With a MNIST database, one could add more features to the dataset since the images are much bigger. This could also increase the accuracy. In either case, implementing cross-validation will make the analysis of the accuracy much more rigorous.

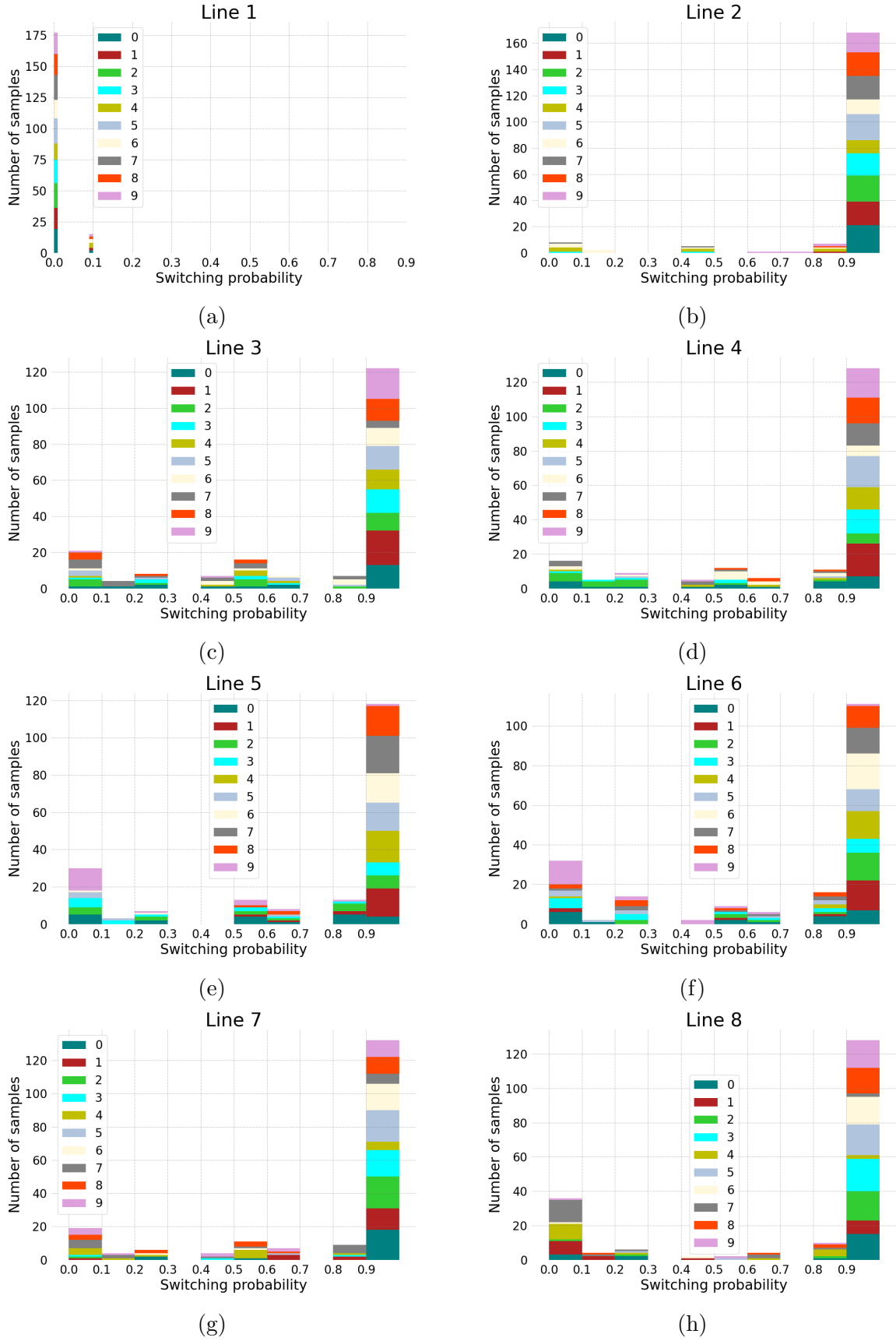


Figure 3.6: Switching probabilities of all 8 lines of all 10 digits in the last experiment. The first line almost never switches because of only black pixels.

Conclusion and outlook

In conclusion, through this report we argued that a single SOT MTJ can be considered as a reservoir and that it can be implemented in a reservoir computing scheme which may be able to classify handwritten digits.

The incremental results presented here can be improved by better leveraging the nonlinearity of the MTJ and by improving the encoding scheme, for example, using temporal encoding schemes [21].

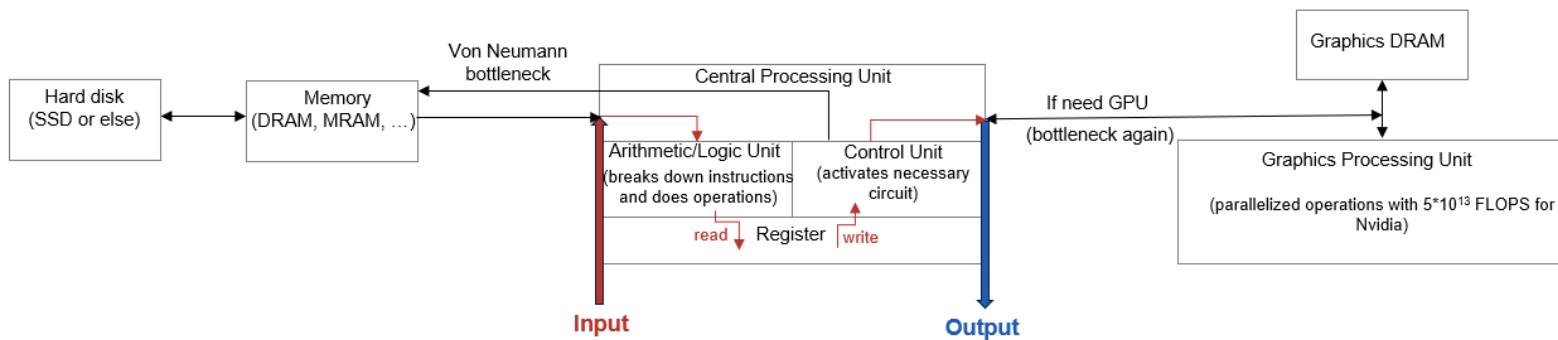
Cross-validation, confusion matrix and precision score will be implemented shortly. A comparison to a standard echo state network should also follow when the experiments show higher scores.

In the future, a possible bigger step is to classify the MNIST handwritten digits data set or to predict time series using established benchmark tests [10].

Appendix

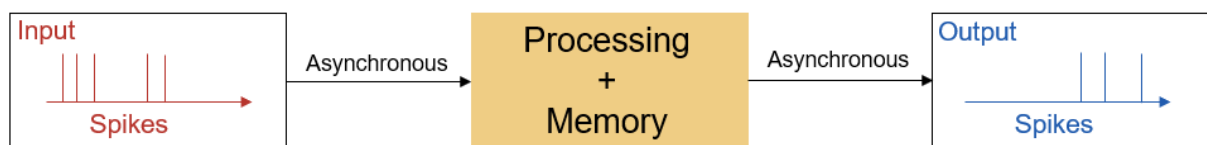
1. A bit more on computer architecture

The von Neumann computer architecture dates back to 1945 and hasn't been modified since. The fact that the memory and the processor are separated introduces several bottlenecks where information transmission is slowed down. In most computers today, it looks like this :



Other computer architectures exist like the Harvard architecture but it still has the same types of bottlenecks. CPU architectures have greatly been improved through CISC and RISC architectures which reduces the amount of bottlenecks but still doesn't totally bring the memory and processing together and are still limited by the frequency (the clock) at which information is transmitted.

In contrast, the brain doesn't have a clock to transmit information (asynchronous) and keeps together memory and processing. This is what neuromorphic computing aims to reproduce and it would look more like this :



2. Emulated MTJs

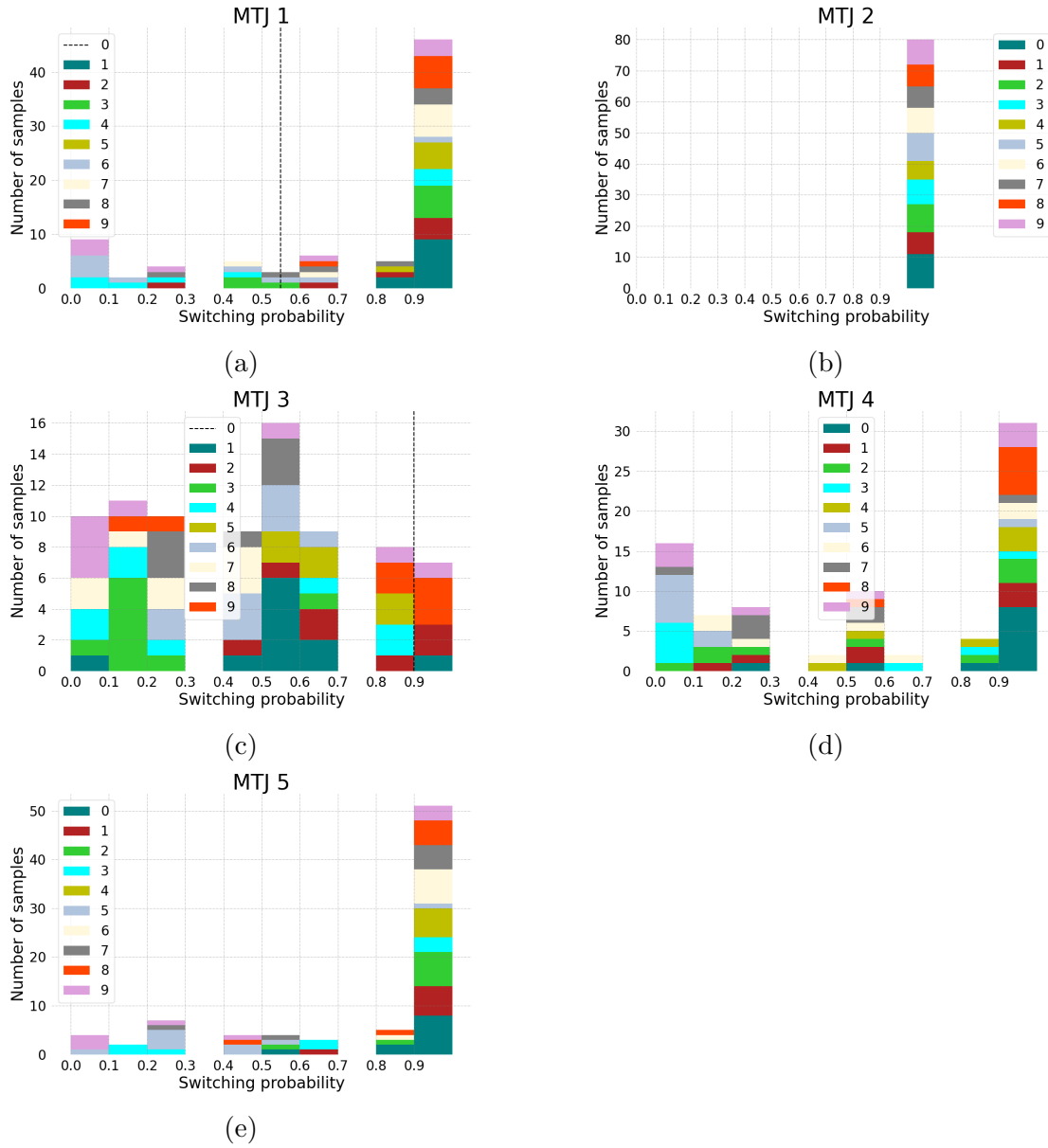


Figure : 10-class classification : accuracy is 10% (random guessing). Binary classification : if the same "8" arrays yielding $P > 0.55$ in (a) also yield $P > 0.9$ in (c) then we can separate the digits 8 and 3.

Abbreviations

- MRAM : Magnetic Random Access Memory
- MTJ : magnetic tunnel junction
- FL : free layer
- RL : reference layer
- SAF : synthetic anti-ferromagnet
- P : parallel state of the MTJ
- AP : anti-parallel state of the MTJ
- LLG : Landau-Lifshitz-Gilbert equation
- SOT : spin-orbit torque
- STT : spin-transfer torque
- VCMA : voltage-controlled magnetic anisotropy
- PMA : perpendicular magnetic anisotropy
- TMR : tunneling magneto-resistance
- AWG : arbitrary waveform generator
- PXI : peripheral component interconnect extension for instrumentation

Bibliography

- [1] Neil C Thompson et al. “The computational limits of deep learning”. In: *arXiv preprint arXiv:2007.05558* 10 (2020).
- [2] Leon Chua. “Memristor-the missing circuit element”. In: *IEEE Transactions on circuit theory* 18.5 (2003), pp. 507–519.
- [3] Dmitri Strukov et al. “Building brain-inspired computing”. In: *Nature Communications* 10 (2019), pp. 4838–2019.
- [4] Laura Bégon-Lours et al. “Scaled, Ferroelectric Memristive Synapse for Back-End-of-Line Integration with Neuromorphic Hardware”. In: *Advanced Electronic Materials* 8.6 (2022), p. 2101395.
- [5] Danijela Marković and Julie Grollier. “Quantum neuromorphic computing”. In: *Applied physics letters* 117.15 (2020).
- [6] Baofang Cai et al. “Unconventional computing based on magnetic tunnel junction”. In: *Applied Physics A* 129.4 (2023), p. 236.
- [7] Jie Cao et al. “Emerging dynamic memristors for neuromorphic reservoir computing”. In: *Nanoscale* 14.2 (2022), pp. 289–298.
- [8] Herbert Jaeger. “The “echo state” approach to analysing and training recurrent neural networks-with an erratum note”. In: *Bonn, Germany: German national research center for information technology gmd technical report* 148.34 (2001), p. 13.
- [9] Min Yan et al. “Emerging opportunities and challenges for the future of reservoir computing”. In: *Nature Communications* 15.1 (2024), p. 2056.
- [10] Hugo Cisneros, Tomas Mikolov, and Josef Sivic. “Benchmarking learning efficiency in deep reservoir computing”. In: *Conference on Lifelong Learning Agents*. PMLR. 2022, pp. 532–547.
- [11] Miguel Romera et al. “Vowel recognition with four coupled spin-torque nano-oscillators”. In: *Nature* 563.7730 (2018), pp. 230–234.
- [12] Danijela Marković et al. “Reservoir computing with the frequency, phase, and amplitude of spin-torque nano-oscillators”. In: *Applied Physics Letters* 114.1 (2019).
- [13] Nils Schaetti, Michel Salomon, and Raphaël Couturier. “Echo state networks-based reservoir computing for mnist handwritten digits recognition”. In: *2016 IEEE Intl conference on computational science and engineering (CSE) and IEEE Intl conference on embedded and ubiquitous computing (EUC) and 15th Intl symposium on distributed computing and applications for business engineering (DCABES)*. IEEE. 2016, pp. 484–491.

- [14] Mu-Kun Lee and Masahito Mochizuki. “Handwritten digit recognition by spin waves in a Skyrmion reservoir”. In: *Scientific reports* 13.1 (2023), p. 19423.
- [15] Eva Grimaldi et al. “Single-shot dynamics of spin-orbit torque and spin transfer torque switching in three-terminal magnetic tunnel junctions”. In: *Nature nanotechnology* 15.2 (2020), pp. 111–117.
- [16] John MD Coey. *Magnetism and magnetic materials*. Cambridge university press, 2010.
- [17] Aurelien Manchon et al. “Current-induced spin-orbit torques in ferromagnetic and antiferromagnetic systems”. In: *Reviews of Modern Physics* 91.3 (2019), p. 035004.
- [18] Arne Brataas, Andrew D Kent, and Hideo Ohno. “Current-induced torques in magnetic materials”. In: *Nature materials* 11.5 (2012), pp. 372–381.
- [19] Ioan Mihai Miron et al. “Perpendicular switching of a single ferromagnetic layer induced by in-plane current injection”. In: *Nature* 476.7359 (2011), pp. 189–193.
- [20] Mikhail I Dyakonov and VI Perel. “Current-induced spin orientation of electrons in semiconductors”. In: *Physics Letters A* 35.6 (1971), pp. 459–460.
- [21] Daniel Auge et al. “A survey of encoding techniques for signal processing in spiking neural networks”. In: *Neural Processing Letters* 53.6 (2021), pp. 4693–4710.
- [22] James Bergstra et al. “Algorithms for hyper-parameter optimization”. In: *Advances in neural information processing systems* 24 (2011).
- [23] Shuhei Watanabe. “Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance”. In: *arXiv preprint arXiv:2304.11127* (2023).