

POLITECNICO DI TORINO

Master degree in Nanotechnologies for ICTs

Master Thesis

# Nanoarchitectonics based on Memristive Nanowire Networks



**Supervisor:**

Prof. Carlo Ricciardi

**Co-supervisors:**

Prof. Daniele Ielmini

Dr. Gianluca Milano

**Candidate**

Kevin Montano

---

October 2020



*To all people who still believe in altruism.*

# Abstract

Transistor-based architectures in the traditional *von Neumann* architecture are reaching their limit exhibiting a wide gap in performances comparing the CPU and the memory addressing, where the latter represents a big constraint in operational frequency. New computing paradigms are necessary to overcome these limitations, where bio-mimetic approaches come to help: brain-inspired paradigms suggest to perform storage and processing operations in a spatial and physical correlated framework. One of the devices which lets this approach possible is a novel analog device: the memristor. Acting as an artificial synapse, this component exhibits resistive switching and memory properties, which allow the mimicking of brain plasticity. Arranged in different architectures they can be involved in the building up of new computing paradigms, such as *reservoir computing*. This approach basically refers to the mapping of an input on a higher dimensional space dynamical system (reservoir) to emphasize and extrapolate spatio-temporal input correlations, which are classified by a simple readout function (usually, a one-layer neural network).

This work deals with self-organizing memristive nanowire networks, which show high connectivity and the capability to replicate some of the characteristics of biological neural networks: homo- and hetero-synaptic plasticity, paired pulse facilitation and short-term plasticity. After discussing the fabrication and experimental measurements, the compact model discussed in the framework of this thesis is exploited to extrapolate phenomenological behavior of network internal state and a quantitative analysis by fitting experimental curves. Model replication of experimental homo- and hetero-synaptic data is demonstrated, both in a qualitative and quantitative point of view up to a certain discussed degree of confidence. By using this model, it was evaluated the possibility to perform reservoir computing on these kind of devices through a simulation approach, exploiting the nanowire network as a reservoir and a one-layer neural network as a readout function. Written digit recognition task is demonstrated and optimized by considering different degrees of freedom of the implemented process, such as electrodes configuration, input processing, managing of reference voltages. Such developed system implements simultaneously, however, memristive and CMOS technology. In order to exploit energy and speed advantages of memristive technology over the traditional one, the possibility to build up a fully-memristive system is demonstrated, again, through simulations of real

hardware components: the memristive crossbar array. Also, a discussion on the energy consumption of the above described architecture is provided.

As the early stage of an ongoing work, moreover, experimental reservoir computing is demonstrated considering 4 different input patterns, paving the way to more complex future analysis.

This work serves as a demonstration of one possible computing approach exploiting self-organizing memristive nanowires network devices, which have been modeled in their relevant behaviors and experimentally exploited to demonstrate actual reservoir computing feasibility. As a future perspective, models developed in the framework of this thesis can support and complement more complex experimental activity towards the implementation of new computing paradigms in self-organized NW networks.

# Acknowledgements

A special thank goes to my supervisor Prof. Carlo Ricciardi, source of inspiration and humanity. A deep gratitude for positive critical eye and the trust in my work, which has made possible my academic and personal growth during last months.

I am also strongly grateful for the special coordination of my co-supervisor Dr. Gianluca Milano. Working *side by side* with you during last months has been a big pleasure, as well as a great occasion.

A deep honor to have collaborated with Prof. Daniele Ielmini and Dr. Giacomo Pedretti, punctual references, sources of knowledge and consulting.

A heartfelt thank goes to all my friends I have met along the university career, who have shared with me feelings, emotions, knowledge and, most important, time. I am thankful to all of you to have ironed out sides of my being.

A special gratitude to my family, with a constant and extraordinary spirit of sacrifice. Your sustain has been, since always, a source of motivation and a spur not to give up in difficult moments.

Last, but not least, an intimate gratitude to Raffaella, who has been, since the beginning, a reference point in my life, my complementary accomplice.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background	1
1.2	Memristor: a fundamental circuit element	2
1.2.1	Memristor and resistive switching devices	3
1.3	Resistive Switching Phenomenon	4
1.3.1	ECM	5
1.3.2	VCM	5
1.4	Memristive devices as artificial synapses	7
1.4.1	Biological synapse	7
1.4.2	Artificial synapse	8
1.5	Modeling memristive behavior	8
1.5.1	Linear Drift Model	10
1.5.2	Non-Linear Drift Model	10
1.5.3	Exponential Model	11
1.5.4	Balanced Rate Equation for STP modeling	12
1.6	Device Applications - Nanoarchitectonics	13
1.6.1	Memristive Cross-Bar Array	13
1.6.2	Self-organizing Memristive Structures	14
<b>2</b>	<b>Memristive Nanowire Networks</b>	<b>17</b>
2.1	Experimental resistive switching effect in memristive nanowire networks	17
2.1.1	Device fabrication	17
2.1.2	Device Characterization	18
	Two-terminal measurements	18
	Multi-terminal measurements	23
2.2	Modeling: Network as a single <i>effective</i> memristor	23
2.2.1	Software Implementation	24
2.2.2	Results	25
2.2.3	Model Limits	26
2.3	Modeling: Network as a grid of memristor	28
2.3.1	Model Structure	29
2.3.2	Homo-synaptic plasticity	32

2.3.3	Hetero-synaptic plasticity . . . . .	35
	Set 1 (N8-W5) . . . . .	35
	Set 2 (N8-S7) . . . . .	37
	Set 3 (N8-S10) - Set 4 (N8-S13) . . . . .	38
2.3.4	Model Assessment . . . . .	40
<b>3</b>	<b>Reservoir Computing</b> . . . . .	<b>41</b>
3.1	Introduction . . . . .	41
3.2	NW network reservoir for written digit recognition . . . . .	43
3.2.1	Input and pulse stream . . . . .	44
3.2.2	NW network reservoir . . . . .	45
3.2.3	Readout function . . . . .	47
3.2.4	Training . . . . .	48
3.3	Results . . . . .	49
3.3.1	Testing . . . . .	49
3.3.2	Optimization of electrodes configuration . . . . .	49
3.3.3	Effect of input processing . . . . .	50
3.3.4	Managing of ground and floating nodes . . . . .	52
3.3.5	Effect of different readout functions . . . . .	53
3.4	Discussion . . . . .	54
<b>4</b>	<b>Fully-Memristive Classification</b> . . . . .	<b>57</b>
4.1	Memristive Cross-Bar Array as a Hardware Neural Network . . . . .	57
4.1.1	Cross-bar array architecture for <i>on-chip</i> training . . . . .	58
4.1.2	Device Simulations . . . . .	60
4.2	Results . . . . .	62
4.2.1	One-layer NN classification . . . . .	62
4.2.2	Two-layers NN classification . . . . .	64
4.3	Discussion . . . . .	65
<b>5</b>	<b>Experimental Reservoir Computing</b> . . . . .	<b>71</b>
5.1	Experimental setup . . . . .	71
5.2	Model simulation . . . . .	73
5.3	Experimental data analysis . . . . .	74
5.4	Discussion . . . . .	80
<b>6</b>	<b>Conclusions and future perspectives</b> . . . . .	<b>81</b>
	<b>Bibliography</b> . . . . .	<b>83</b>

# Chapter 1

## Introduction

### 1.1 Background

Digitalization is the great challenge we are facing nowadays, which permeates in a variety of fields: Internet of Things (which is evolving in *Internet of Everything*), healthcare monitoring and diagnosis, Information and Communication Technology (ICT), Digital Twin as a virtual *alter-ego* to make predictions.

The concept of digitalization is intimately connected to Artificial Intelligence (AI), through which big amount of data are processed to classify information, extrapolate correlations in data-patterns and make predictions on the basis of learned information (from here *Machine Learning*).

Two big limits, however, are worrying scientists about this rapid evolving scenario: power consumption and complex tasks performances. Moreover, the throughput of the traditional *von Neumann architecture* has reached its limit: nowadays the time to address the memory is way longer than the CPU time performance, generating a *bottleneck* for current computing paradigm.

Following the bio-mimetic approach, human brain performances is capable of solving complex tasks with a very low power consumption. The high connectivity of neurons, which communicate by means of synapses, does not show a separation in memory and processing unit, but computation and storage are spatially and physically correlated: this is the reason why the most promising paradigm refers to In-Memory-Computing. The mimicking of neural synapses activity was born in the 1980s with the concept formalization of neuromorphing computing by Carver Mead. The research, involved in looking for nano-electronic devices to reproduce neurons, considers the *memristor* one of the most promising hardware implementations of biological neurons. The memristor, theorized in 1971 by Leon Chua and realized in 2008 in HP labs, relies on the resistive-switching mechanism which makes it a candidate for efficient memories. Moreover, the great technological hype is generated by its capability of reproducing synaptic plasticity, the fundamental phenomenon allowing in-memory computing.

## 1.2 Memristor: a fundamental circuit element

According to a work from Leon Chua (1971) [1], the memristor was first theorized on the basis of mathematical symmetry related to the fundamental circuit variables.

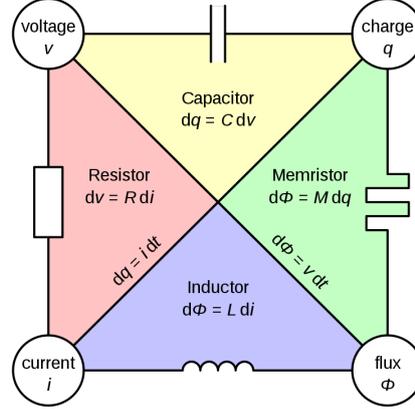


Figure 1.1: Relations among fundamental circuit elements variables.

Before that moment, the four state variables *charge* ( $q$ ), *voltage* ( $V$ ), *flux* ( $\phi$ ) and *current* ( $i$ ) were connected by flux and charge definitions:

$$dq = i dt \quad (1.1)$$

$$d\phi = v dt \quad (1.2)$$

and by the RLC circuit elements state equations:

$$dV = R di \quad (1.3)$$

$$d\phi = L di \quad (1.4)$$

$$dq = C dV \quad (1.5)$$

In this set of equations, the one linking the charge and the flux was missing:

$$d\phi = M(q) dq \quad (1.6)$$

The quantity  $M(q)$  is the so called memristance of the theorized missing passive element. By combining eq. (1.2) and (1.6), the memristor state equations reduces to Ohm's law:

$$V = M(q)i \quad (1.7)$$

It is worth noticing that, due to Ohm's law, the behavior is similar to the resistor one, but with the difference of the resistance state depending on the charge. Otherwise

stated, it is a resistor with memory on previous state history: from here the name *memristor* as a combination of *memory* and *resistor*.

In order to generalize the concept of memristor to a memristive system, the memristance argument is expressed in function of the state variable which has its own equation depending on the particular system [2]:

$$\begin{cases} V(t) = M(q, s(t))i(t) \\ \frac{ds(t)}{dt} = g(s(t), i(t)) \end{cases} \quad (1.8)$$

### 1.2.1 Memristor and resistive switching devices

From an electrical characterization point of view, the voltage-current plane exhibits a pinched (in the origin) hysteresis loop covering the I and III quadrants, as presented in Figure 1.2, where high resistance state (HRS) and low resistance state (LRS) typical of resistive switching devices are highlighted.

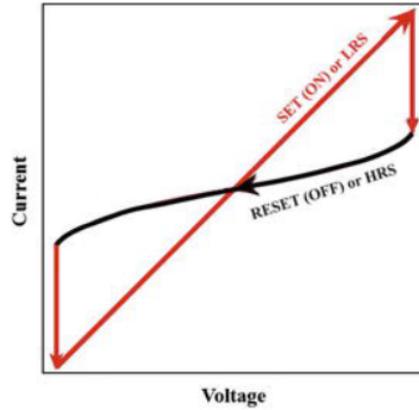


Figure 1.2: Pinched hysteresis loop, highlighting the resistive switching phenomenon. Reprinted from [3].

Actually, the concept of memristor has generated and continues to rise misconceptions about it.

The presented ideal memristor theorized by Chua in 1971 has been demonstrated impossible to realize by Vongehr et al. [4]. Successively, Abraham et al. [5] highlighted its infeasibility.

However, as already mentioned, the first memristor was realized in HP labs thanks to Williams et al. [6]. Its working principle was presented to be the resistive switching one driven by atomic rearrangement as an effect of applied electric field.

It is important to remark that resistive switching phenomenon was already known and studied in previous works, as by Waser et al. [7]. However, the HP memristor generated high attraction on resistive switching effect, giving to the related work a

high importance.

Stated the difficulty in realizing the ideal memristor, Chua associated all the non-volatile resistive switching devices to memristors, despite the physics behind the phenomenon [8].

In a more general framework, despite the still opened debate, memristive systems (or devices) concept is associated to those devices whose resistance depends on applied voltage (or current) history.

### 1.3 Resistive Switching Phenomenon

Resistive switching effect in electronic devices refers to their capability to exhibit (at least) two different non-volatile states associated to their electrical resistance value. In particular, they are characterized by a high resistance state, behaving as an insulating material, and a low resistance state which allows electrons transport. The bi-stable behavior, seen as the two logic states '0' and '1', is highly exploited in the fabrication of non-volatile memory devices.

The nature of this phenomenon may rely on different physical properties of materials: magnetism, electrostatic and atomic reconfiguration. However, whatever the particular device, the structure fingerprint is a sandwiched structure *metal-insulator-metal*, where the dielectric choice, the interface properties and the structure size influence the behavior. The whole scenario is schematized in Figure 1.3.

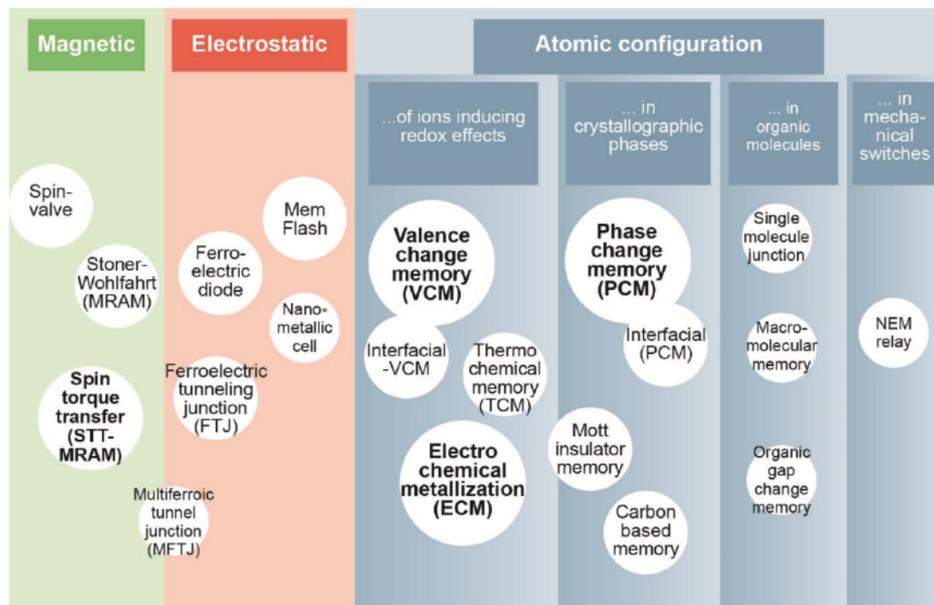


Figure 1.3: Resistive switch effect classification on the basis of physical principle. Reprinted from [9].

Nowadays, although efficient applications of magnetic and electrostatic devices exist, great attention is paid to the atomic reconfiguration class. The latter is further differentiated on the base of the material where the organization occurs: organic molecules, crystallographic phases, mechanical switches and ions inducing redox reactions. Memristive device covers the atom reconfiguration scenario, where two different frameworks are dominant: *electro-chemical metalization* (ECM) and *valence change memory* (VCM).

### 1.3.1 ECM

The electro-chemical metalization, as the name anticipates, is based on the generation of a conductive filament across a dielectric layer (in the role of a solid electrolyte) starting from electro-chemical reactions. The phenomenon requires an initial oxidation of the anode metal atoms as a consequence of an applied voltage:



Then, a migration of the metal cation across the dielectric is necessary to reach the cathode. It is worth noticing that the migration is possible thanks both to the presence of an electric field and the amorphousness of the dielectric layer: a crystalline insulating layer, otherwise, would not permit ions to migrate.

Finally, a reduction reaction at the cathode allows the creation of a metal hillock which soon becomes a filament:



The whole process exhibits a dual behavior with applied voltage opposite in sign, which allows the restoring of the initial configuration. The physics of the process relies on the coupling of ions and electrons transport: when no filament exists the memory cell is in a high-resistance state, when a conductive bridge is formed, instead, it switches to a low resistance state.

In order to emphasize a connection between the described phenomenon and the pinched hysteresis loop proposed above, the figure 1.4 is reported.

### 1.3.2 VCM

The valence change memory effect, as ECM, relies on the atomic reconfiguration in order to provide a conductive path from the cathode to the anode. In this framework, however, the migration is associated to oxygen ions (and corresponding oxygen vacancies) of the metal-oxide structure sandwiched between the two electrodes. Typical used metal-oxide are  $TiO_2$ ,  $HfO_2$ ,  $Ta_2O_5$ . Stated otherwise, a change in the local stoichiometry generates conductive regions.

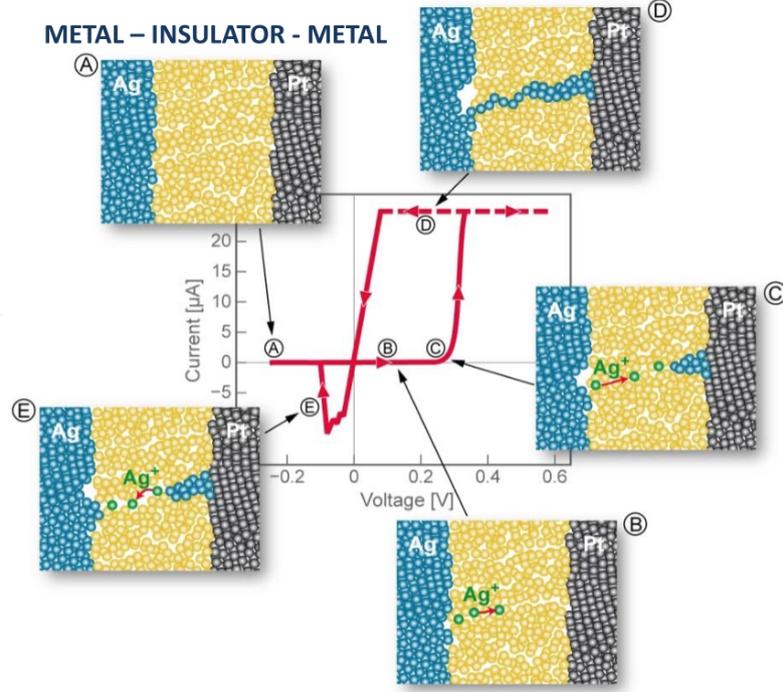
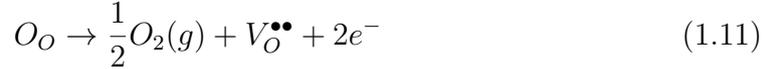


Figure 1.4: Electro-chemical metalization process, highlighting the connection with the pinched loop. Reprinted from [10].

Applying an electric field, in fact, the anodic oxidation of the metal-oxide happens according to:



with  $O_O$  the oxygen ions as a part of the metal-oxide lattice and  $V_O^{\bullet\bullet}$  the oxygen vacancies. This reaction has been experimentally proven observing the formation of liquid water bubbles in humidity controlled environment, highlighting the production of gaseous oxygen [11]. The generation of a conductive region for electrons is responsible for the change of the device state from an high to a low resistance configuration, exhibiting the resistance switching effect. The connection of the migration dynamic with the pinched loop is highlighted in Figure 1.5, where it is shown also the reversibility of the process.

**The nano-scale nature of the process** An important remark about the ECM and VCM resistive switching processes is their intimate nanoscopic nature.

The electric field necessary to activate metal ions dissolution and providing their migration is about  $E \simeq 10^7 \div 10^8 \text{ V/m}$ .

Considering operating voltages of few Volts, the dielectric thickness must be *few tens of nanometers*.

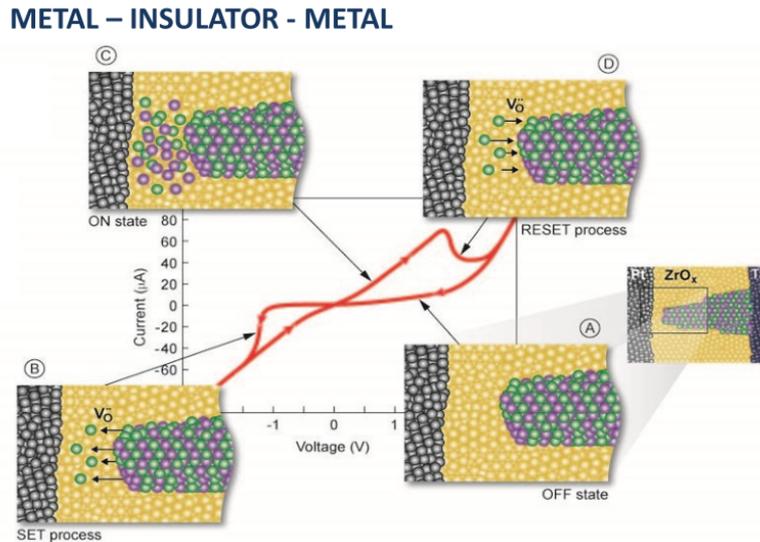


Figure 1.5: Valence change memory process, highlighting the connection with the pinched loop. Reprinted from [10].

## 1.4 Memristive devices as artificial synapses

### 1.4.1 Biological synapse

In the human brain, a synapse occurs between two connected neurons, namely pre and post-synaptic ones. Neuron's physical structure is made up of an axon, a soma and dendrites. A biological synapse occurs between the terminal region of the pre-synaptic neuron axon and the post-synaptic dendrite. Actually, the dendrites of a neuron are able to receive information from multiple neurons axons and produce a signal following the *integrate and fire* rule. The signal to integrate, delivered by each axon, is the *action potential* characterized by a spike waveform. The inputs integration operation is not a democratic one: a weighted sum is performed on the basis of the *synaptic weight*.

Neuronal plasticity is at the base of the synaptic weight modulation: frequently stimulated synapses show an increased weight over the less stimulated ones. The weight information is associated to the concentration of different ionic species ( $K^+$ ,  $Ca^{2+}$ ,  $Na^+$  etc.).

Two main plasticity classes regulate the human brain with its memory and processing capability:

- Short-term synaptic plasticity (STSP)
- Long-term synaptic plasticity (LTSP)

STSP refers to  $\sim 10 - 100$  millisecond time scale associated to the efficacy of state

modulation of a post-synaptic neurons induced by a firing of a pre-synaptic one [22]. LTSP, on the contrary, deals with hours or longer time.

In both cases the modulation may be either potentiation or depression of synaptic strength.

### 1.4.2 Artificial synapse

Memristive devices represent an important element for neuromorphic implementations due to their capability of reproducing biological synapses behavior.

This is linked to the resistive switching effect, which allows the device to have multiple states with different conductivity, mimicking the biological synaptic strength. As in human brain the plasticity is linked to ions concentration, so in memristive devices it is associated to ion migration, modulating the filament structure.

In particular, it exhibits conductance modulation upon multiple voltage or current stimuli. Depending on the input sign, both conductance increasing and decreasing can be obtained. Depression may also be linked to a spontaneous relaxation of the conductive filament, leading to the so called *short-term plasticity* (STP). When the potentiation occurs in an almost non-reversible way it refers to long-term plasticity (LTP).

The same device, if properly designed, can be used to implement both STP and LTP depending on the number of pulses sent: Zhang et al. [12] have shown both effects on a memristive device characterized by the stacked structure Cu/a-Si/Pt differently operating on the basis of the number of stimuli.

In bio-neural world, while LTP is important for memory purposes, STP is supposed to be involved in solving critical tasks [13].

Nanowire networks proposed in this work only exhibit short term plasticity due to their volatile behavior. This is a fundamental property which can be exploited to perform reservoir computing: it has been adopted in other works for temporal filtering [13] [14], transient memory buffer [15], pattern completion [16] and other neural functions [17].

Here it is important to provide a relative fast computation paradigm with a millisecond timescale for the network to adapt to repeated external stimuli.

## 1.5 Modeling memristive behavior

As in every electronic device, modeling is a fundamental tool to describe basic processes and simultaneously access more complex properties and behaviors.

Thinking of memristive systems as the unit of new computing paradigms, simulations play an important role to examine the potentiality of new architectures.

As already mentioned, the particular model has to be defined referring to the analyzed system, according to eq. (1.8). Different models have been proposed in

literature, both physics-based and semi-empiric ones, with the aim of finding the most simple model to reproduce experimental data.

The choice of a model is done on the basis of the system feature it is able to reproduce with a trade-off with its computational cost.

Concerning memristive devices, the modeling state of art is schematized in figure 1.6.

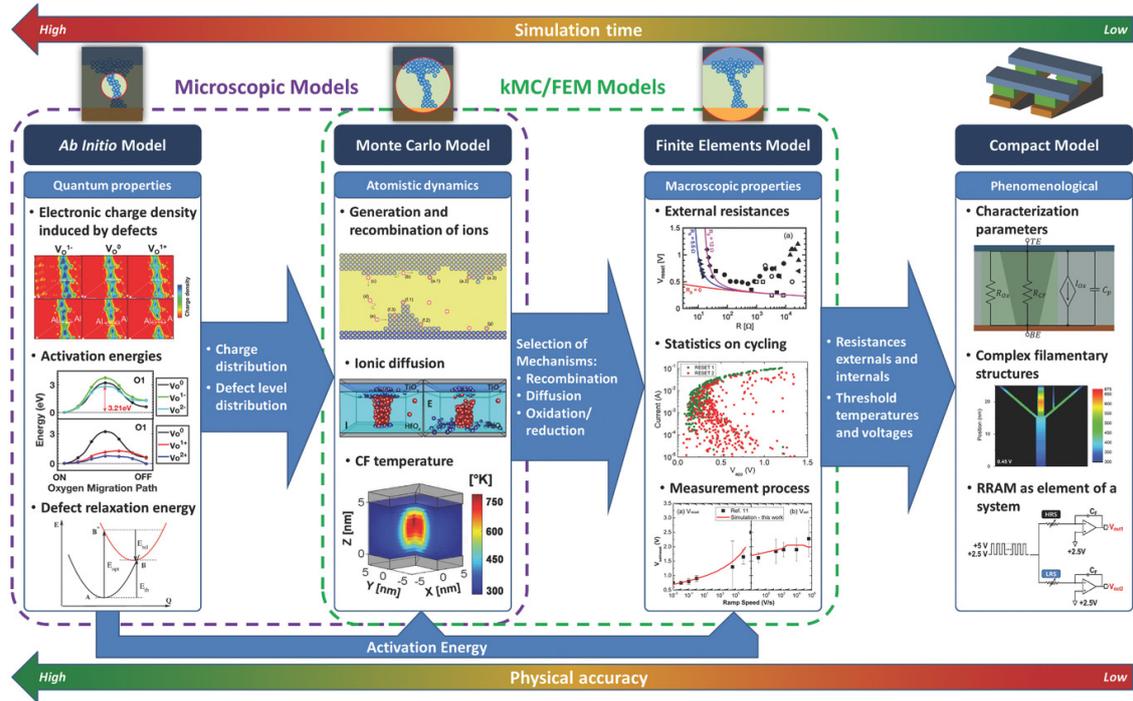


Figure 1.6: State of the art concerning modeling memristive devices. Reprinted from [18].

Models are differentiated on the basis of their scale of analysis. As the scale becomes lower, the physical accuracy increases, with the prize of a higher computational cost.

On the other hand, dealing with phenomenological approaches, the computational cost becomes lower with a partial disconnection from the physical nature of the working principle.

Following this approach, models can be clustered in four different classes:

- Ab-initio: atomic level is investigated dealing with electronic charge density due to defects, activation energies and defect relaxation energies
- Monte Carlo: generation/recombination and diffusion dynamics of ions and investigation on the conductive filament temperature

- Finite element: macroscopic properties as cycling statistics, measurement analysis and external resistances
- Compact: phenomenological approach through parameters characterization, filament structure and high-scale behavior as an element of a larger system

It is worth noticing that a lower scale model provides the necessary parameters to perform the subsequent higher scale analysis.

This work will deal with a compact model one: a balanced rate equation will be used to extrapolate model parameters on the basis of experimental data with the aim of exploring new nano-architectures for the implementation of unconventional computing paradigms. The low computation cost, here, is a key requirement.

### 1.5.1 Linear Drift Model

The first model developed is connected to the first physical memristor realized at HP lab. It is a two-terminal device with a semiconductor ( $TiO_2$ ) film in between electrodes ( $Pt$ ) made up of two differently oxygen-vacancies doped regions. Exploiting the different conductance of the semiconductor depending on doping level, an electric field is used to let the vacancies migrate and change the width of the doped region. According to this behavior, the overall memristance can be modeled as an average between the two regions resistance:

$$M(q(t)) = R_{on}x(t) + R_{off}(1 - x(t)) \quad (1.12)$$

where  $x(t) = \frac{w(t)}{D}$  is the normalized doped thickness with respect to the overall semiconductor region. The state variable  $x(t)$ , then, is linked to the current flowing across the device by:

$$\frac{dx(t)}{dt} = \mu \frac{R_{on}}{D^2} i(t) \quad (1.13)$$

with  $\mu$  the average drift mobility of charges. In its explicit expression, equation (1.13) turns out to be a linear equation between the state variable and the charge:

$$x(t) = \mu \frac{R_{on}}{D^2} q(t) \quad (1.14)$$

Equations (1.12) and (1.13) complete the generalized description proposed by equations (1.8).

### 1.5.2 Non-Linear Drift Model

The linear behavior shown in equation (1.14) has, basically, two main limits. First, the drift of oxygen vacancies may be not linear at the boundary due to the fact that strong electric fields may arise even from small signals. Second,  $x(t)$  will never

reach zero value, meaning no oxygen vacancies present in the device. The mathematical way to generalize it is to introduce a non linearity in equation (1.13):

$$\frac{dx(t)}{dt} = \mu \frac{R_{on}}{D^2} i(t) f(x(t)) \quad (1.15)$$

where  $f(t)$  is the so called *window function*. The choice of the best window function has been widely studied, however all of them must have some properties to provide a correct model [19] [20]:

- zero value at the boundary and maximum value in the middle region
- it should belong to the interval  $[0, 1]$
- it should provide a non linear drift across the whole device
- it should match the boundary properties
- it must be a generalization of linear drift and not a distortion of it
- it should be dependent on a control parameter to tune the model

Just to name some adopted window functions, the most used ones have been described by *Joglekar et al.* [23], *Biolek et al.* [24], *Prodromakis et al.* [25] and *Zha et al.* [26].

### 1.5.3 Exponential Model

With the described non linear model, even by looking for the best window function, is difficult to reproduce the strong electric field effect into real devices. In order to reproduce it, an exponential model has been developed by Yang et al. [21], according to which the current is:

$$i(t) = x(t)^n \beta \sinh(\alpha V(t)) + \chi(e^{\gamma V(t)} - 1) \quad (1.16)$$

with  $\alpha$ ,  $\beta$ ,  $\gamma$  and  $\chi$  fitting parameters.

According to authors, the model choice is linked to the phenomenological capability of reproducing I-V curves, rather than the physical background. Stated otherwise it belongs to the class of compact models: its simplicity has influenced the authors to adopt it.

Considering an equivalent circuitual representation, it is described by a parallel between a diode-like rectifier and a memristor modeled by a tunneling behavior in the ON-state. Looking at equation (1.16), in fact, the total current is given by the sum of a tunneling current through a thin residual barrier plus a diode current, respectively.

### 1.5.4 Balanced Rate Equation for STP modeling

All the here discussed models deal with the first HP memristive structure. This work, however, will provide insights concerning memristive nanowires network randomly organized by means of a self assembly growing.

A suitable model for describing memristive devices with short term plasticity has been described by Miranda et al. [27]: a balanced voltage-controlled rate equation developed in connection with memristive structure made up by a ZnO nanowire connected to Ag and Pt electrodes. The change in conductance, here, is associated to the migration of  $Ag^+$  ions, differently from HP device which is based on oxide filament. Despite the model has been tested on single nanowire memristive system, it can be adopted to describe all memristive systems exhibiting STP property based on Ag dynamics.

This simple and analytical model, differently from previous cited ones, is able to reproduce short-term synaptic plasticity, which suggests its utility in neuromorphic application.

In accordance with Chua's framework, two equations are defined to model the electronic transport and the memory state dynamic (associated to ions displacement). The electronic transport is described by a linear equation (1.17):

$$I = [G_{min}(1 - g) + G_{max}g]V \quad (1.17)$$

with  $g$  the normalized conductance acting as a state variable, whose dynamic is defined by:

$$\frac{dg}{dt} = k_P(1 - g) - k_Dg \quad (1.18)$$

The two coefficients  $k_P$  and  $k_D$  are the potentiation and depression coefficients which, for simplicity, are modeled as a function of the voltage only:

$$\begin{cases} k_P = k_{P0} e^{\eta_P V} \\ k_D = k_{D0} e^{-\eta_D V} \end{cases} \quad (1.19)$$

The latter equation system is an approximation of the ionic diffusive dynamics which follows a hyperbolic sinus behavior [28][29][30].

It is worth noticing that, although a negative voltage is able to induce a relaxation, also a zero voltage will result in a spontaneous decaying of conductance state to the high resistance state, in accordance with the experimental observations of STP.

**Other Models** The memristive modeling scenario is much depth than the proposed one. As already mentioned, the detailed model should be built up on the given particular device. However a model can be customized to other devices by means of fitting parameters of the model itself.

The most difficult task is to describe to the best the experimental data, but adopting

a simple compact model. Physical arguments can be easily added to the model (such as quantum-point contact [31], Landauer nanowire approach, Schottky barrier modulation [32] and so on), however the simplicity is difficult to maintain.

## 1.6 Device Applications - Nanoarchitectonics

To fully exploit the memristor component, different nanoarchitectonics have been developed. The idea is to mimic the human brain to reach its efficiency in terms of power consumption, neurons connectivity, chip-area, operational frequency and complex task managing.

The most exploited one is memristive cross-bar array, which can be used to perform *one-shot* different operations involved in machine learning algorithms. However, despite crossbar arrays have demonstrated the possibility to accelerate different computing paradigm, more and more attention is devoted to new architectures which exhibits a closer brain-inspired structure: this is the case of random self-organizing memristive systems, which can be adopted to reproduce biological effect such as short-term and long-term plasticity, hetero-synaptic plasticity, paired pulse facilitation, while managing multi-terminal inputs.

The basic principles of these two architectures are discussed in the following.

### 1.6.1 Memristive Cross-Bar Array

The memristive crossbar array is basically a 2-dimensional matrix of memristive devices which are placed at each row-column node. The row and column electrodes behave as input and output pads to send and collect signals. Each memristive node, as highlighted in figure 1.7, represents the synapse between a pre-synaptic neuron and a post-synaptic one. Moreover, by exploiting the Ohm's law, it is able to obtain an output which is a weighted sum of input values, according to memristive conductance weights. This behavior can be assimilated to a certain extent to the biological function *integrate and fire*. At each cross point, in fact, the current is given by the applied voltage divided by the memristor resistance and those contributions in current are summed over a column.

Thanks to described behavior, several mathematical operations can be implemented on the crossbar hardware.

*Matrix-vector multiplication (MVM)* can be obtained by sending an input voltage vector on row electrodes, storing the matrix values on memristor conductances, finally collecting output current vector on columns electrodes. Current values turn out to be the result of the MVM [38].

Through similar consideration, outer products can be realized to write new conductance state of memristive nodes.

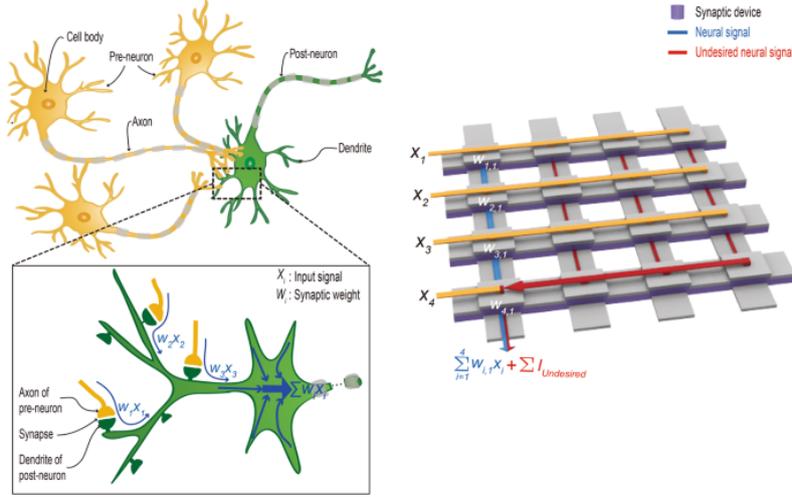


Figure 1.7: Memristive crossbar array structure and its biological meaning. Reprinted from [33].

These two operations are fundamental to realize a neural network (NN) algorithm: cross-bar arrays, in fact, find their major exploitation in realizing on-chip trained NN providing orders of magnitude energy efficiency and increased speed with respect to CMOS technology [34].

Involved in supervised and unsupervised learning, memristive crossbar array are a natural choice for machine learning applications, as image recognition, language processing, decision making, healthcare and brain-machine interface [37].

Yao et al. [35] were the first to implement face recognition on a 128x8 array based on  $HfAl_yO_x/TaO_x$  memristive cross point. A 16x32 crossbar sub-array (from a 32x32 structure)  $WO_x$ -based, in addition, has been adopted by Sheridan et al. [36] for image processing.

By adopting a closed loop architecture, then, it is also possible to perform *inversion of matrix* [38].

In addition, as highlighted in [39], one shot linear regression algorithm training can be accelerated observing the possibility to in-memory compute the pseudo-inverse matrix of the problem.

It is important to remark that all these processes, regardless of the problem dimension, are performed in a single operation, leading to the fastest computing approach. Moreover, since memristors are analogue by nature, this device device is able to receive analog signals from sensors and other devices, avoiding an ADC stage.

## 1.6.2 Self-organizing Memristive Structures

By further taking inspiration from human brain, new architectonics are based on a random configuration of memristive structure, obtained by self-assembly techniques.

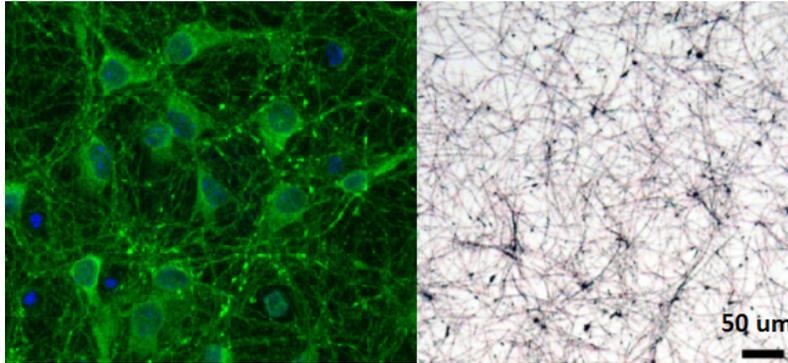


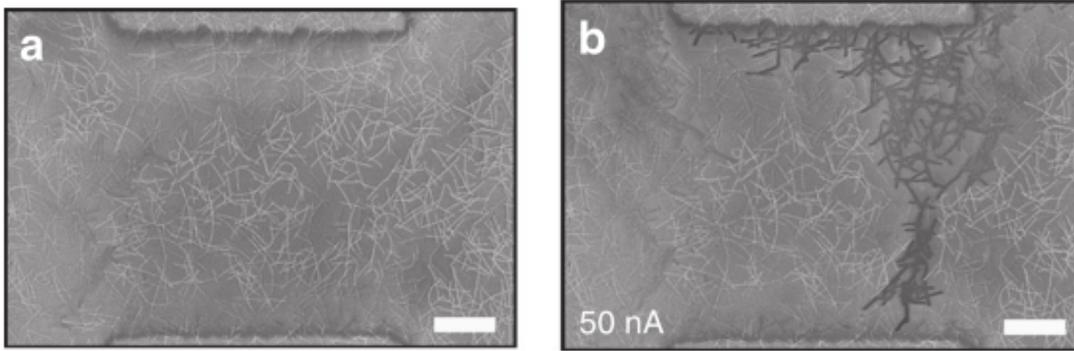
Figure 1.8: **Left.** Human brain, fluorescence imaging. **Right.** Self-organizing memristive nanowire network, SEM image. Reprinted from [40].

This production approach is the reason why these structures are extremely cheap. As studied by Diaz-Alvarez et al. [41], these kind of structures exhibit complex dynamics as collective memory response in the sub-threshold voltage region, LRS and HRS characterized by different power-law fluctuation scaling, resilience and adaptation behavior as in biological neuronal systems.

Another important property of nanowire networks has been highlighted by Gomes da Rocha et al. [43]: during potentiation, the network is capable of self select the most energy efficient path connection among electrodes, as presented in figure 1.9. Moreover, this behavior leads to conductance plateaus which are stable over a certain range in current compliance. Stated otherwise, inputs can be mapped in the network internal state by means of conductive paths. These results generates great attraction concerning neuromorphic computing. Self-organizing structures also refer to other nano-systems: Pike et al. [42] have shown how self organizing nanoparticles can reproduce neuronal behavior by conductance modulation through tunneling gaps between particles.

Among these self-organized architecture, random interconnected nanowires are promising structures since they exhibit high surface to volume ratio, making possible analysis about the effect of the surroundings on their behavior: they can be exploited through surface functionalization or light stimuli [45; 46; 47; 48].

Based on interconnected nanowires, these 2-dimensional networks are able to exhibit a high connectivity, with *millions of synapses per square millimeter*. Different biological phenomena can be reproduced: short and long term plasticity, paired-pulse facilitation and hetero-synaptic plasticity. Applying a proper stimulus in a two terminal fashion, in fact, it has been seen that a not only a conductive path is generated among these two as already discussed, with a retention of a certain time depending on the device and the applied stimulus, but also other neighbor regions can be stimulated to a high resistance state.



(a) Unperturbed nanowire network, SEM image (scale bar:  $2\ \mu\text{m}$ ).

(b) Stimulated network with current compliance  $I_C = 50\ \text{nA}$ , SEM image (scale bar:  $2\ \mu\text{m}$ ).

Figure 1.9: Winner takes all phenomenon in a  $10\ \mu\text{m} \times 10\ \mu\text{m}$  Ag nanowire network. Reprinted from [43].

While homo-synaptic plasticity serves to associative modulation of synaptic weights, hetero-synaptic plasticity counter-acts runaway dynamics introduced by Hebbian rules and balances synaptic modulations. Stated otherwise, hetero-synaptic plasticity provides stable learning systems and enhances synaptic competition [44]. Moreover, the interconnection of non-linear components with a memory effect may result useful in new computing paradigm, such as reservoir computing.

## Chapter 2

# Memristive Nanowire Networks

This work deals with memristive nanowire networks architectonic with the aim to connect the experimental data to a mathematical model. Two different approaches have been evaluated, both adopting the balanced rate equation model (eq. (1.18)): first, considering the whole network as a single effective memristor device and second, building up a simplified graph model with each edge following the cited state equation.

### 2.1 Experimental resistive switching effect in memristive nanowire networks

#### 2.1.1 Device fabrication

The analyzed data refer to a device made up of Silver nanowires covered by a thin layer ( $1 \div 2 \text{ nm}$ ) of Polyvinylpyrrolidone (PVP), an insulating polymer. PVP is the residue of the nanowire synthesis, since it acts as a surfactant to obtain high aspect ratio structures. In this specific case, nanowires are  $115 \text{ nm}$  in diameter and  $20 \div 50 \mu\text{m}$  in length purchased from *Sigma-Aldrich* chemical company. The presence of an insulating polymer, however, is fundamental to obtain a resistive switching behavior providing a *metal-insulator-metal* structure.

The fabrication [40] has been obtained by drop-casting Silver nanowires in alcohol (IPA) suspension on a  $\text{SiO}_2$  substrate. The areal mass density (AMD) of the deposited nanowires can be controlled by tuning the mass ratio between suspended nanowire and IPA. In particular, lower AMD can be obtained with higher levels of IPA, as shown in figure 2.1. It is remarked that nanowires are purchased in alcohol suspension: if a lower AMD is desired, dilution in IPA is increased.

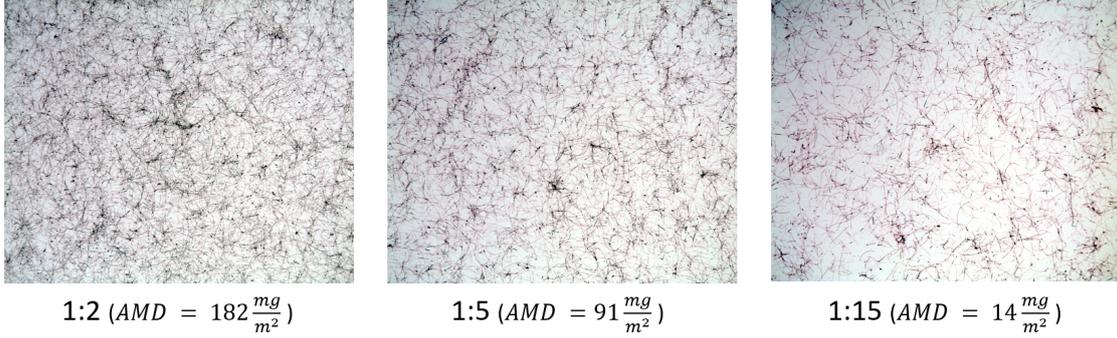


Figure 2.1: Nanowire density controlled by the ratio Ag NW: IPA.

Subsequently, gold pads are realized along the periphery of the network through sputtering with the use of a shadow mask. A zoomed view of pads is highlighted in figure 2.2. It is important to remark that there is no need of cleanroom facilities or lithographic processes, making the process simple and cheap.

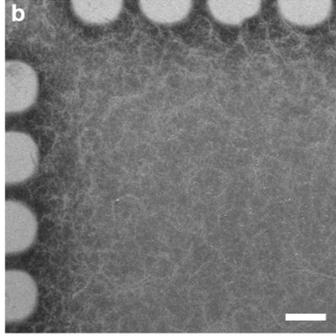


Figure 2.2: Gold electrodes deposited through sputtering. SEM image, scale bar  $250\ \mu\text{m}$ . Reprinted from [40].

### 2.1.2 Device Characterization

In the following, experimental results from the group exploited for modeling are presented and discussed.

Experimental results acquired in two-terminal and multi-terminal configurations are reported in the following.

#### Two-terminal measurements

The two-terminal configuration characterization has been performed by means of a Keithley 4200 analyzer coupled with Pulse Measurement Units (PMUs). Au electrodes 7mm-spaced have been considered for this analysis.

In order to highlight the memristive behavior of the network while testing cycling endurance, figure 2.3 depicts the pinched hysteresis loop over 300 cycles. It is evident that a clear separation between HRS and LRS still holds after 300 cycles. The network areal mass density is an important process variation to explore.

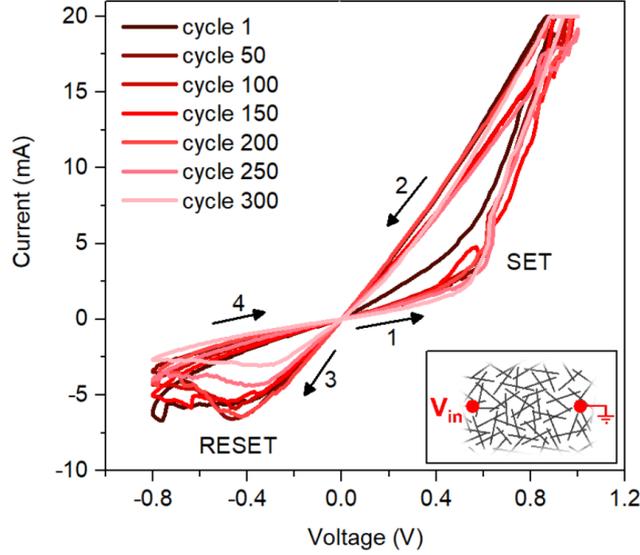
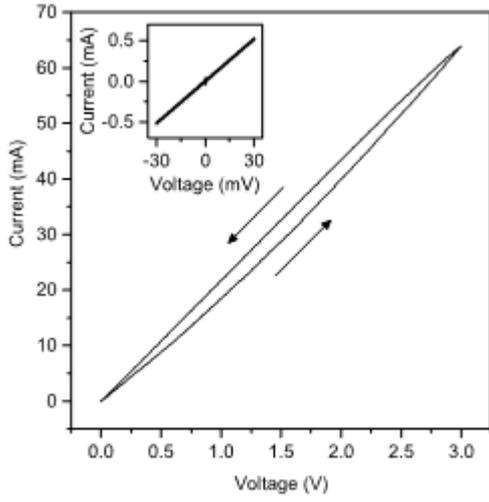


Figure 2.3: Pinched hysteresis loop over 300 cycles, with a sweep rate of  $0.27 \frac{V}{s}$  and a compliance current of  $20 \text{ mA}$ . Reprinted from [40].

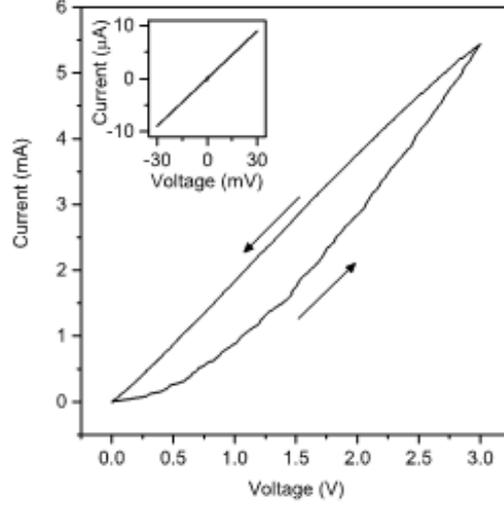
Figure 2.4 presents a comparison of two network with different nanowires density when subjected to the same stimulus. By looking at figures 2.4a and 2.4b a wider cycle can be observed for lower density network. From a physical point of view this is a way to state that lower density network exhibits a wider conductance range: the derivative of the curve in each point, in fact, represents the conductance value in that state. For higher density network, the HRS and LRS tend to be closer to each other. By cycling over a smaller voltage input range, going from  $-30 \text{ mV}$  to  $30 \text{ mV}$  it can be seen that the resistive switching effect is absent, obtaining a resistive behavior and exploiting these voltages for reading process.

In addition, figures 2.4c and 2.4d show the response of the network in case of a square voltage stimulus. The conductance increase with respect to pristine state is much higher in case of lower density network, obtaining an increasing of  $153\%$ , against the  $0.47\%$  of higher density sample. Moreover, the latter does not exhibit quantized conductance level: this is linked to the higher number of memristive connections, whose connection/disconnection probability distribution tends to smooth the relaxing dynamic.

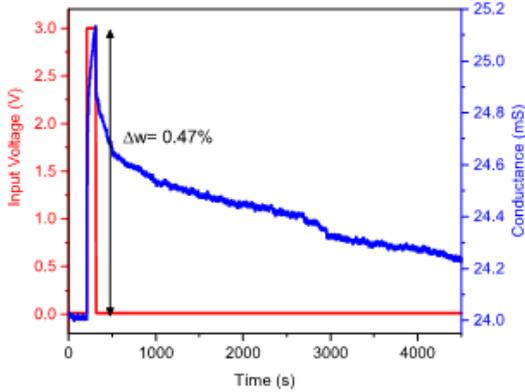
By referring to the lower density sample, to further investigate the resistive switching property of the network, a step signal with different amplitude has been sent, as shown in figure 2.5a, with conductance values obtained from the measured current.



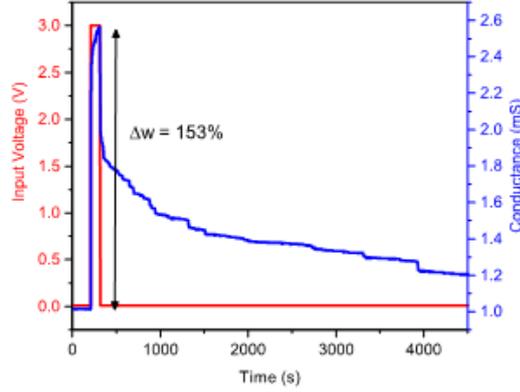
(a) AMD:  $91 \frac{mg}{m^2}$ . I-V cycle: sweep-rate  $0.3 \frac{V}{s}$ .



(b) AMD:  $14 \frac{mg}{m^2}$ . I-V cycle: sweep-rate  $0.3 \frac{V}{s}$ .



(c) AMD:  $91 \frac{mg}{m^2}$ .  $V_{pulse} = 3V$ ,  $V_{read} = 1mV$ ,  $t = 100s$ .



(d) AMD:  $14 \frac{mg}{m^2}$ .  $V_{pulse} = 3V$ ,  $V_{read} = 1mV$ ,  $t = 100s$ .

Figure 2.4: Areal mass density effect on Ag nanowire network behavior.

As it can be seen, when a high signal is applied, the conductance exhibits a non-linear increasing, emphasizing a transition from the high-resistance state to the low-resistance state. It is remarked that the behavior is analog, rather than digital: a smooth transition among all intermediate conductance states is observed. As experimental demonstrated [43] and reported in figure 1.9, a conductive path is created between the stimulated electrodes. Higher flown charge quantities (related to higher voltage input in this case) during the forming process lead to a wider conductive path.

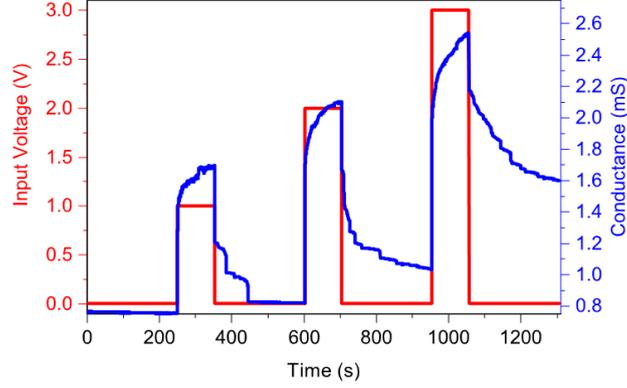
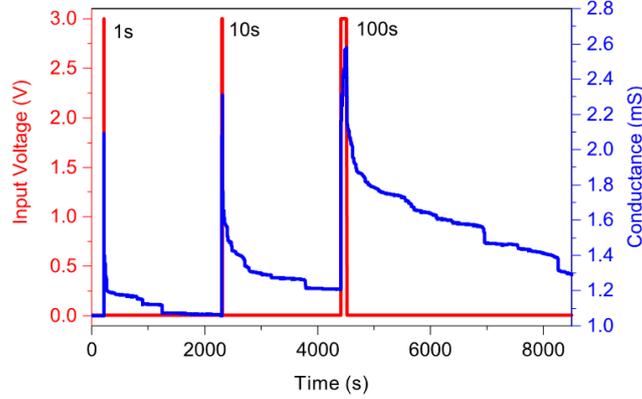

 (a) Step-wise input signal and measured conductance ( $\text{AMD} = 14 \frac{\text{mg}}{\text{m}^2}$ ).

 (b) Different timing input signal and measured conductance ( $\text{AMD} = 14 \frac{\text{mg}}{\text{m}^2}$ ).

Figure 2.5: Conductance variation behavior with respect to pulse amplitude and pulse time.

As the input voltage becomes low ( $10 \text{ mV}$ ), instead, the reverse process occurs. It is important to remark that a *spontaneous relaxation* is typical of these devices: there is no need of negative voltage to restore the high resistance state condition. This is due to a spontaneous filament atoms diffusion in each memristive junction of the network proportional to the gradient of surface atomic chemical potential, according to the Gibbs-Thomson effect as discussed in [49]. This process, in case of isotropic surface diffusion assumption, can be modeled by the surface atomic flux along a certain surface  $s$ :

$$J_s = - \left( \frac{D_s \gamma \delta^4}{kT} \right) \nabla_s \chi \quad (2.1)$$

where  $D_s$ ,  $\gamma$ ,  $\delta$ ,  $k$ ,  $T$  are the surface diffusion coefficient, surface energy, inter-atomic distance, Boltzmann constant and temperature, respectively.

$\chi$  represents the surface curvature, where a spatial variation of it drives the Gibbs-Thomson effect.

The volatility, thus, suggests that these kind of structures cannot be used for memory storage. However, the spontaneous relaxation is important to emulate STP phenomenon, which is at the base of learning processes as already discussed in previous chapter.

The relaxation pattern, moreover, highlights the presence of *quantized steps of conductance*. In case of single memristor, quantized states are observed due to the nanoscopic nature of the filament, mathematically described by the Landauer approach. In this framework, however, the quantized steps are likely to be linked to the discrete disconnection of memristive junctions, which results in a different effective conductive path.

Not only the voltage amplitude, but also the pulse duration influences the network resistive switching as depicted in figure 2.5b. Previous considerations holds, including the effect of charge flow into the device affecting the potentiation behavior (longer pulse time means higher charge).

In order to explore the biological similarity of this network, repeated stimuli have been sent. The conductance behavior emphasized in figure 2.6 follows the so called paired pulse facilitation: two subsequent stimuli, due to plasticity of the network, produce a higher post-synaptic potential if the stimuli are sufficiently close in time [29; 50].

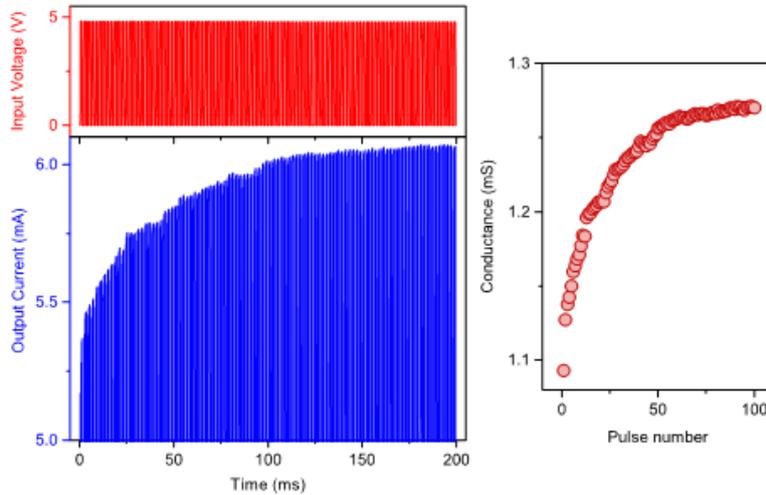


Figure 2.6: Paired pulse facilitation (5V amplitude, 100 stimuli).

## Multi-terminal measurements

The multi-terminal characterization of the device has been conducted by means of multiple electrical probe tips driven by micro-manipulators. A Keithley 707 switch matrix has been adopted to select the proper combination of electrodes, while keeping the others floating. A TTI-TGA 1202 signal generator (100MHz bandwidth) has been used to send rectangular wave of different amplitude and duration. The current, in the meanwhile, has been measured by a Lecroy Wavesurfer 3024 oscilloscope (200MHz bandwidth).

Multi-terminal configuration measurements, moreover, have highlighted the hetero-synaptic behavior of the network: by stimulating a couple of electrodes, the conductance variation also influences other region of the networks, observed as a variation of conductance across two different electrodes. The sent pulse structure is a 8V voltage input of 1s duration. Figure 2.7 depicts some experimental results, where figure 2.7a shows the terminal configuration and figure 2.7b highlights the resistance variation after stimulus for each couple of electrodes combination. The latter results are arranged in correlation matrices. As it can be seen, hence, the resistance variation of non stimulated terminals has non-zero values. In biological environment, the hetero-synaptic plasticity is the key element for associative learning.

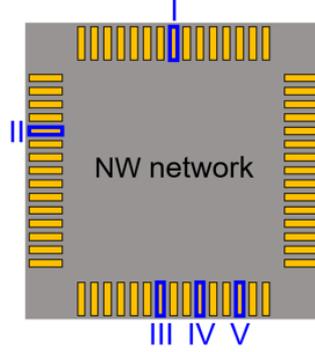
## 2.2 Modeling: Network as a single *effective memristor*

The aim of the following discussion is to analyze the capability of a model to reproduce the experimental data presented above. Looking for the simplest mathematical approach, the nanowires network has been considered (in first approximation) as a single memristive device described by the balanced rate equation (1.18). The great advantage of this model, is the possibility to describe the conductance evolution with respect to the applied voltage with an analytical and recursive equation:

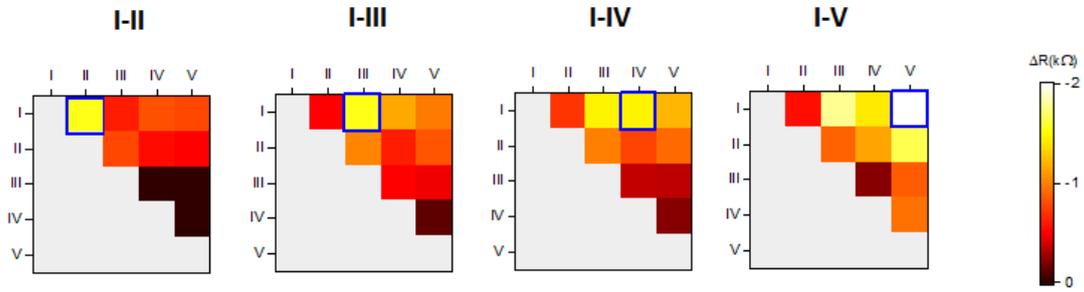
$$g(t) = \frac{k_P}{k_P + k_D} \left\{ 1 - \left[ 1 - \left( 1 + \frac{k_P}{k_D} \right) g(t - \Delta t) \right] e^{-(k_P + k_D)\Delta t} \right\} \quad (2.2)$$

where the voltage dependence is hidden in  $k_P$  and  $k_D$ , according to equation (1.19). Stated otherwise, the network is seen as a *black box* governed by the presented state equation.

The effective memristor approximation is not so rash in principle: the conductive path formation in nanowires network can be seen, on a different spatial scale, as the filament growing in single memristor. The different spatial scale of the two phenomena can be included in the model parameters which are fitted on the basis of experimental data.



(a) Pad configuration for hetero-synaptic plasticity analysis.



(b) Resistance variation for each couple of network's electrodes after stimulus.

Figure 2.7: Experimental hetero-synaptic plasticity analysis in multi-terminal configuration. The four cases refer to different stimulated electrodes configurations. Reprinted from [40].

### 2.2.1 Software Implementation

*Python* environment has been adopted to implement the fitting analysis of experimental data, in accordance to equation (2.2), in order to extrapolate the model parameters:  $k_{P0}$ ,  $\eta_p$ ,  $k_{D0}$ ,  $\eta_d$ ,  $g_0$ ,  $g_{min}$ ,  $g_{max}$ . The last two parameters are essential to define the allowed conductance range for the system, while  $g_0$ , due to the recursive model, is the initial conductance value.

A code excerpt with the model definition and the fitting function is reported:

```
[...]
def model(time, kp0, kd0, eta_p, eta_d, g0, g_min, g_max):
    g = zeros(len(time),)
    for i in range(0, len(time)):
        signal = V[i]
        kp = kp0*exp(eta_p*signal)
        kd = kd0*exp(-eta_d*signal)
        if i == 0:
            g[i] = g0
```

```

else:
    delta_t = time[i] - time[i-1]
    g[i] = (kp/(kp+kd)*(1-(1-(1+kd/kp)*(g[i-1]-g_min)/
        (g_max - g_min))*exp(-(kp+kd)*delta_t)))*
        (g_max - g_min) + g_min
return g

[...]

best_val, covar = curve_fit(model, time, G_exp, p0=init_val, method='trf',
    bounds=(0, max_val), maxfev=600)

[...]

```

The adopted function for fitting procedure is `curve_fit()` from *SciPy* library. The method for this function has been set to `trf`, i.e. *Trust Region Reflective* algorithm. It is a simple but powerful method adopted in solving non linear programming problems and it belongs to the class of least square algorithms. The problem of fitting reduces to find the set of parameters which minimize a function  $f(x)$  which is a sum of squares (function which represents the distance between the given data to fit and the model curve). The problem of fitting, thus, reduces to a minimization problem: the basic approach of Trust-Region-Reflective method is to approximate  $f(x)$  which a second order series expansion over a certain interval of confidence (trust region) and perform the minimization.

## 2.2.2 Results

In order to test the conformity of the model to the experimental data, a first analysis has been conducted on a single potentiation-depression pattern resulting from a single voltage pulse. Figure 2.8a depicts the system input signals and figure 2.8c the respective conductance state of the system. It is stressed that the depression region correspondent voltage input is  $V_{off} = 10\text{ mV}$ , a non-zero value which allows the conductance reading while keeping the system as much as possible unperturbed. As it can be observed in figure 2.8c, the presented model is able to reproduce experimental data with a good conformity both from a phenomenological and quantitative point of view. However, since the model is basically an exponential function of the voltage, no quantized states can be observed, obtaining a smooth relaxation behavior.

Identical considerations hold for potentiation-depression pattern given by 1V and 3V stimulation as presented in figures 2.8b and 2.8d. However, fitting parameters in these three scenarios, presented in figure 2.9, turn out to be quite different from each others. This is the cause for a bad fitting of more complex pulse patterns, as in case of a sequence of 1V, 2V and 3V potentiation, as it emerges in figure 2.10. However, given the non-exact match of model and data, the former is a valid description of the experimental system up to a certain degree of confidence.

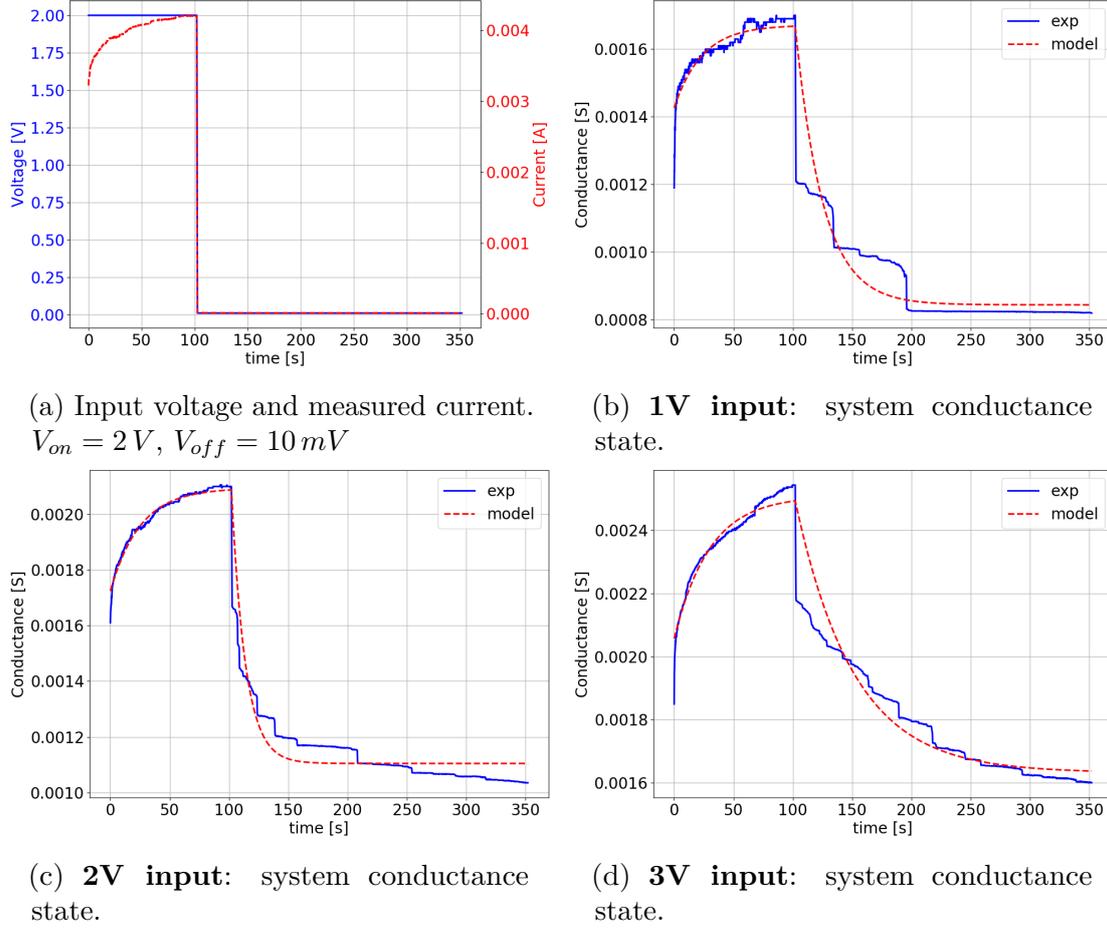


Figure 2.8: Conductance state evolution with a 1V, 2V and 3V input voltage stimulation.

This is highlighted also in paired-pulse facilitation fitting, presented in figure 2.11. As it emerges from figure 2.11a, the qualitative behavior is fitted, even if the model is not able to intercept correctly the peak values. However, the model is able to quantitatively follow the facilitation given by a train of pulses, as depicted in figure 2.11b.

### 2.2.3 Model Limits

The main limitation of the model is linked to the absence of dependence of conductance evolution from the current flown into the device, but only on the previous conductance state.

In single memristor the filament morphology is strongly determined by the imposed compliance current (the maximum current flowing into the device).

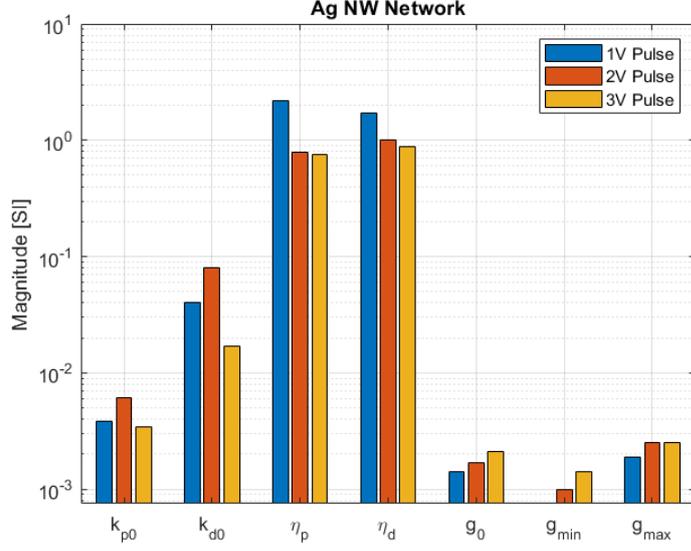


Figure 2.9: Histogram of fitting parameters for 1V, 2V and 3V stimulation.

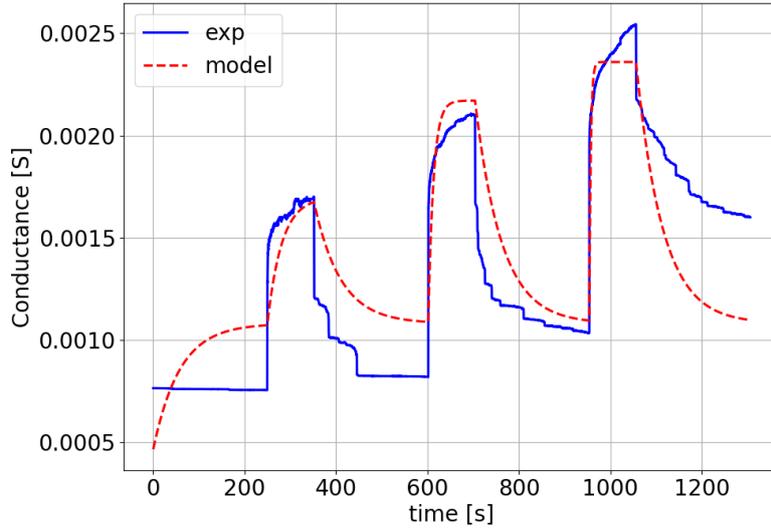


Figure 2.10

In these devices, established its qualitative similarity with a single effective memristor, the flown current should determine the conductive path width. A certain conductance state is not in one-to-one relationship with the low-resistance path morphology, suggesting a lack of information in the model equation. By looking in literature [51], modeling of single memristor includes the dependence on the compliance current (or, in general, the maximum current flown before the relaxation)

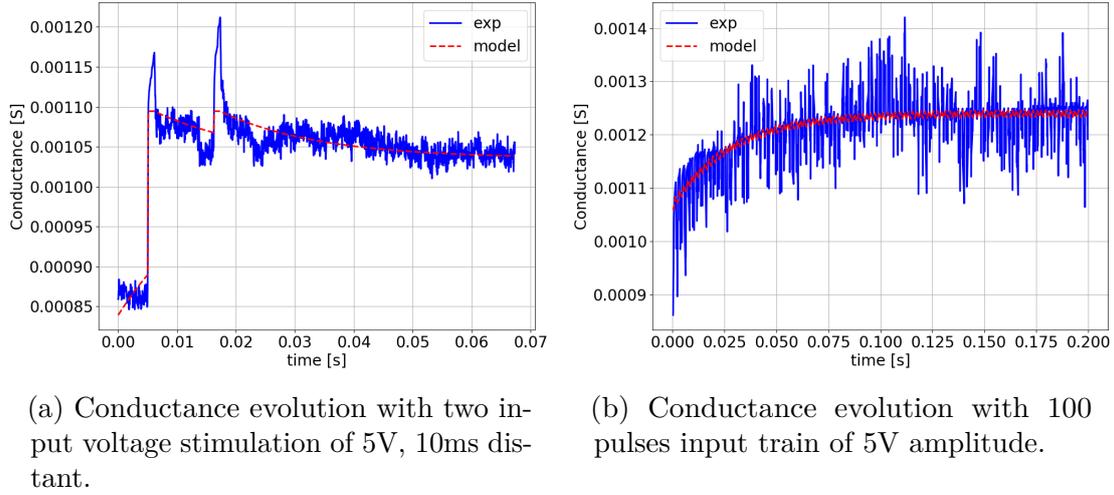


Figure 2.11: Conductance state evolution exploiting repeated input pulses.

when defining the relaxation time constant. As in the referring study, for example:

$$\tau_{reset} \propto \left( \frac{I_C}{V_C} \right)^2 \quad (2.3)$$

with  $I_C$  the compliance current and  $V_C$  the correspondent voltage value. However, since the model is able to correctly reproduce experimental data in case of fixed input structure (in terms of voltage amplitude and timing) by extrapolation of respective fitting parameters, its mathematical simplicity represents a major advantage in terms of computational cost with respect to other models. This is a relevant advantage for a compact model in order not to have long computational times, while reproducing the most important behaviors of the modeled system.

Second, this model only allows two-terminal data fitting, excluding multi-terminal analysis: device spatial information is loosened by considering a single effective memristor.

## 2.3 Modeling: Network as a grid of memristor

In order to overcome the previous discussed limit, a grid model has been implemented in order to map the spatial features of the device. As already proposed in [40], Ag nanowires can be mapped on the nodes of the graph and the PVP insulating layer can be associated to the graph edges. The step forward with respect to the cited study relies on the state equation associated to the conductance of the edges. The previous simulation work relies on ad-hoc design of the shortest conductive path between stimulated electrodes with a subsequent exponential decaying. However it was not able to reproduce the device experimental dynamics due to manual design

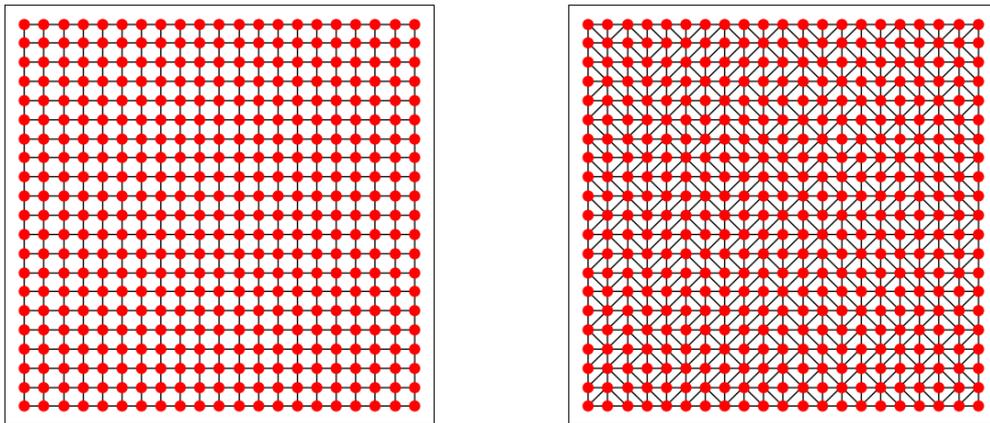
of the conductive path. Here, on the other hand, each memristive edge of the graph evolves according to the balanced rate equation (1.18).

With this new approach is possible to simulate potentiation dynamic to investigate the actual conductive path morphology and, second, to provide a quantitative analysis which was not possible in previous work.

It is important to stress that the aim of the model is not to map each memristive connection, which are of the order of millions per square millimeter, since it would be unreliable. However, it is possible to map a subset of artificial synapses on a single edge to reduce complexity. In other words, the idea is similar to before but on a substantial different scale.

### 2.3.1 Model Structure

The model has been built up by defining first a square grid of nodes connected by edges by exploiting the *Python* library *Networkx*. Two different edges arrangement have been implemented: one by inserting only rows and column edges, the other by also placing edges along diagonals with a random orientation as shown in figure 2.12 (a random seed approach has been adopted to have always the same randomness). The simpler grid is useful to reduce computational costs in case of two stimulation electrodes arranged on the same row or column. The random diagonal structure, instead, is useful to guarantee a plausible conductive path morphology in case of diagonal arranged electrodes.



(a) **Structure 1:** Row and column edges.

(b) **Structure 2:** Random diagonals graph.

Figure 2.12: Two different grid model structures.

The dimension of the grid depends on the physical device to map, in particular on the number of electrodes to place. It is important to preserve the experimental relative distances between electrodes, avoiding a single edge connecting each pair of them.

Once defined the structure backbone, its evolution has been simulated by implementing the *modified voltage node analysis* (MVNA) algorithm. The idea is to place voltage signal generators between stimulated electrodes and solve the electrical circuit of a grid of resistances. Moreover, at each computational time step, each edge resistance value needs to be updated according to equation (1.18) depending on the voltage drop across the particular memristive edge. The MVNA provides a great advantage in the whole analysis if compared with *voltage node analysis* (VNA) and *mesh current analysis* (MCA). The last two, in fact, can only deal with current and voltage generators respectively, without the possibility to work with them simultaneously, allowed by MVNA.

In this scenario, since the MVNA only works with passive circuit elements, each memristor is considered as a resistor of a certain resistance value for each time step. The algorithm consists in solving a linear system of equations, which in case of independent current and voltage sources results to be:

$$Ax = z \tag{2.4}$$

Considering a graph with  $N$  nodes and  $M$  sources, the matrices definition is:

$$A = \begin{bmatrix} G & B \\ C & D \end{bmatrix} \tag{2.5}$$

$$x = \begin{bmatrix} v \\ j \end{bmatrix} \tag{2.6}$$

$$z = \begin{bmatrix} i \\ e \end{bmatrix} \tag{2.7}$$

where:

- $G$  is a  $N \times N$  matrix containing along the diagonal the sum of elements conductance connected to a each node and off-diagonal elements are the negative value of element conductance connected to each pair of nodes
- $B$  is a  $N \times M$  matrix containing 0, 1, -1 values corresponding to the presence of a source between two nodes and in which orientation
- $C$  is a  $M \times N$  matrix corresponding to the transpose of  $B$
- $D$  is a  $M \times M$  matrix full of zeros in case of independent sources
- $v$  is a  $N \times 1$  matrix with each element corresponding to node voltages

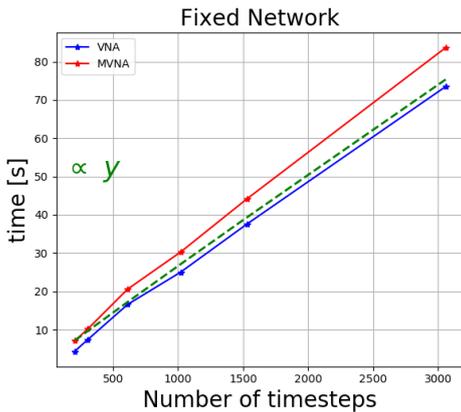
- $j$  is a  $M \times 1$  matrix where each entry is the current flowing through each voltage source
- $i$  is a  $N \times 1$  matrix with each entry equal to the sum of currents through each element connected to a certain node
- $e$  is a  $M \times 1$  matrix corresponding to independent voltage sources

Stated otherwise, the system implements the Kirchoff current law at each node of the graph, while introducing additional equations for each source of the circuit. The great advantage of implementing the MVNA with respect to VNA will be clearer when dealing with reservoir computing in next chapters. However, the advantage introduced has to be paid with a slightly higher computational costs. In figure 2.13 is presented a comparison among the two methods.

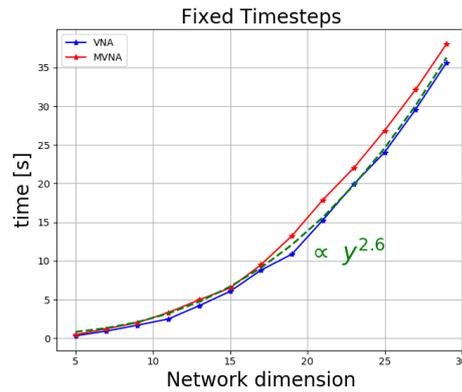
The algorithm bottleneck, in fact, is linked to the matrix inversion operation, performed by means of Python function `numpy.linalg.inv()`.

The matrix inversion with the cited algorithm requires, in general,  $O(N^3)$  operations, with  $N$  the matrix dimension. Since the relation between the number of nodes ( $n$ ) and the matrix dimension is  $n = N^2$ , a  $O(N^3)$  behavior should reflect in a  $O(n^{3/2})$  behavior.

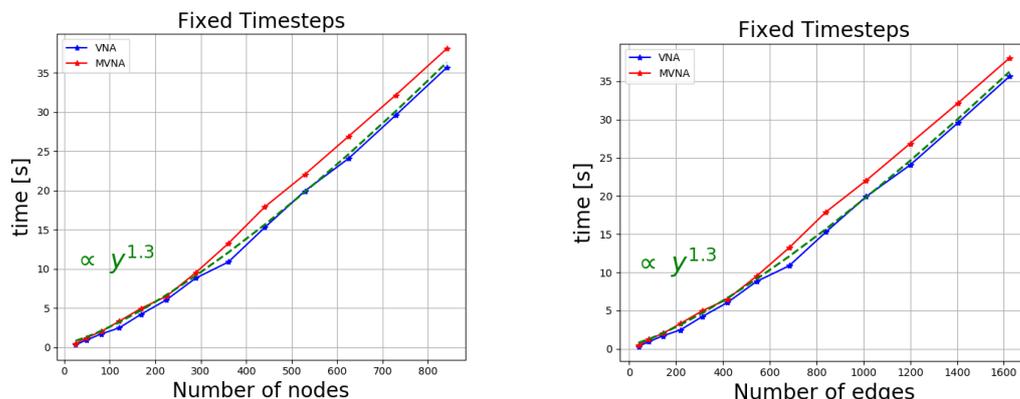
Moreover, the number of edges ( $e$ ) goes as  $e = 3N^2 - 4N + 1$  in case of random diagonal case and  $e = 2N^2 - 2N$  in case of simple grid graph. This means that for low  $n$  the linear increasing of edges is dominant, reflecting in a  $O(e^3)$  behavior. For high matrix dimension the quadratic term is dominant, so the increasing follows the same of nodes, i.e.  $O(e^{3/2})$ , but with a different constant factor.



(a) Computational cost at fixed network dimension and variable time steps



(b) Computational cost at fixed time steps and variable dimension of the network



(c) Computational cost at fixed time steps and variable number of nodes of the network

(d) Computational cost at fixed time steps and variable number of edges of the network

Figure 2.13: **MVNA** and **VNA**: computational cost comparison.  $y$  is just an arbitrary name to show the curve increasing with respect to the independent variable. These results have been obtained by stimulating the network with input voltage presented in figure 2.8c.

Both methods show a dependence of computational time with respect to network characteristics just described, but with a slightly better performance: being the matrix symmetric it will require less effort to invert it. The performance with respect to network dimension, in fact, falls from  $O(N^3)$  to  $O(N^{2.6})$  (figure 2.13b), reflecting in a  $O(n^{1.3})$  (figure 2.13c) and  $O(e^{1.3})$  (figure 2.13d) behavior considering nodes and edges, respectively.

Moreover the computational time increases linearly with the considered simulation time-steps (figure 2.13a): this is reasonable since a fixed dimension matrix needs to be inverted for  $N_t$  times, with  $N_t$  the number of time-steps.

Here it is also understood why, when possible, the analysis without random diagonals has to be preferred.

### 2.3.2 Homo-synaptic plasticity

The potentiation-depression pattern already discussed with the previous modeling (figure 2.8c) is a benchmark for this new simulation scenario.

It is important to remark that the fitting parameters are the same for each edge. The fitting procedure is similar to before with some differences: first the initial conductance point (high resistance state, with absence of conductive path) is fitted initializing each edge with the same value, then the parameters which better reproduce the network evolution are found.

Figure 2.14b shows the fit obtained with a 21x21 grid without random diagonals, where figure 2.14a depicts the stimulated pads.

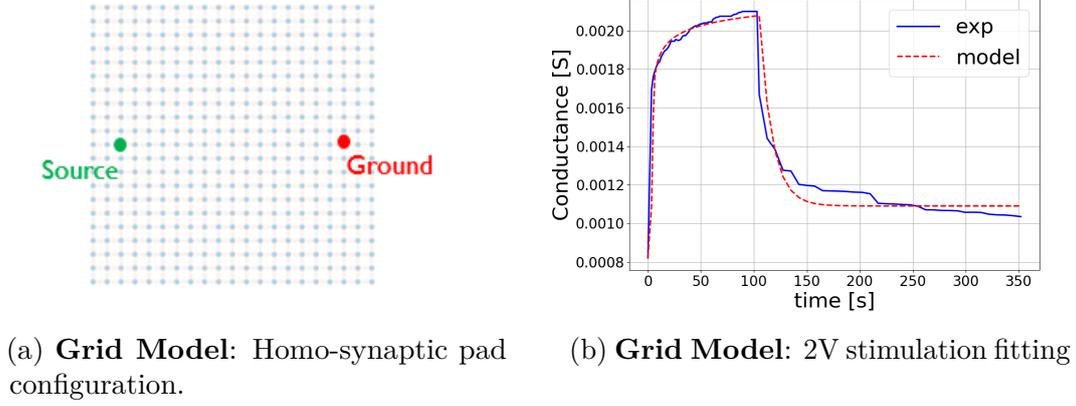


Figure 2.14: Homo-synaptic model fit for 2V pulseshape.

The model is able to correctly reproduce the experimental data both in potentiation and depression region.

As already stated, we are now able to investigate also the internal properties of the network, such the filament formation dynamic. Figure 2.15 shows the state of the network in 3 different time instants during potentiation.

The first frame refers to  $t = 5s$ , when the network starts to be potentiated: as it can be seen a filament is not present, but it starts to grow from the electrode nodes positions toward the center of the network.

Next, at  $t = 9.59s$ , a conductive path is present with a certain width which covers about a third of the network: figure 2.15e shows the edge values along the central vertical section of the network.

At  $t = 89.22s$  the maximum potentiation of the network is reached, where it can be seen that the conductive path section is wider than before, covering now about half of the network: a persisting voltage, or equivalently flown charge, has the effect of enlarging the conductive path section, similarly as in single memristor where the filament grows in section.

Dealing with the spontaneous relaxation region, figure 2.16 depicts the conductive path depression in three different time instants. Differently than potentiation, this dynamics, as it has been modeled, involves all the path volume *almost* at the same rate. It would ideally relax at the same rate in case of identically zero input voltage, losing the dependence of the conductance update (equation (2.2)) on the voltage drop across each edge. Since a low, but non-zero, reading voltage is provided, the relaxation rate slightly varies on the network region.

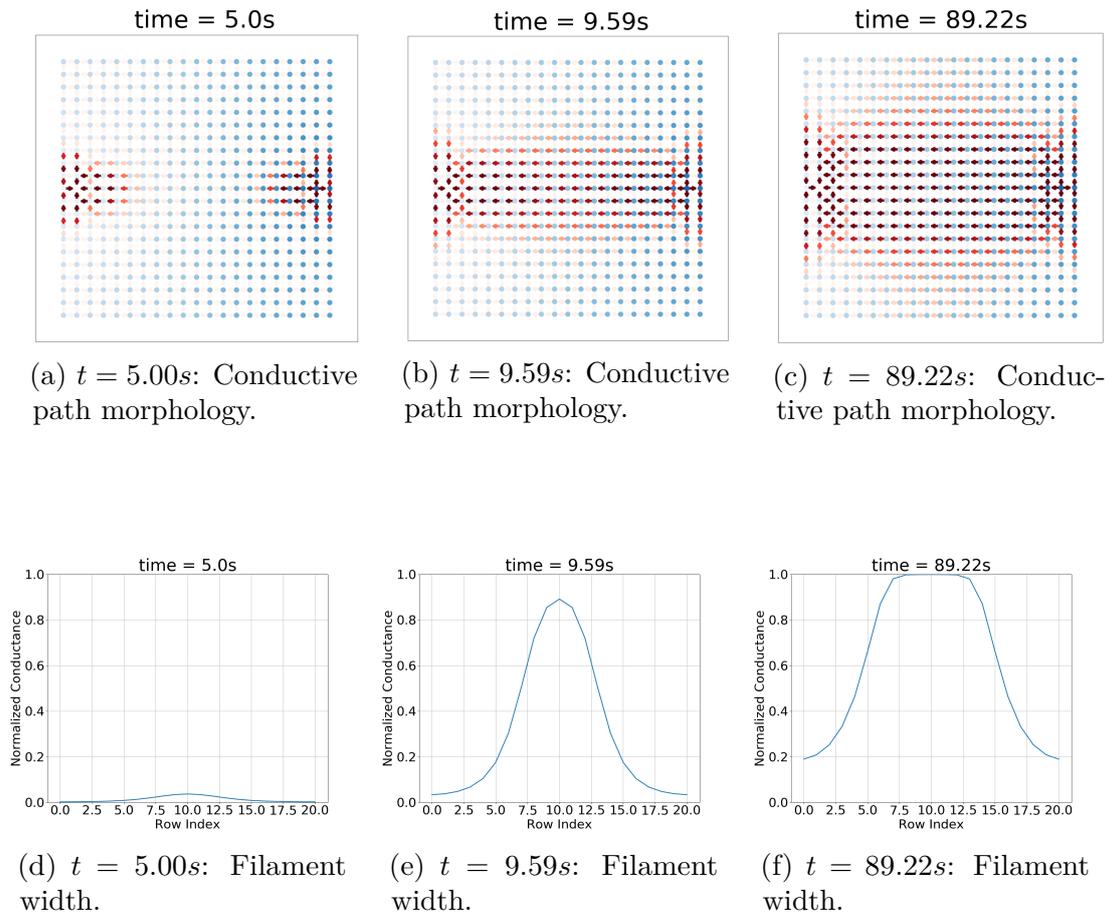


Figure 2.15: Spatial information on filament formation dynamic.

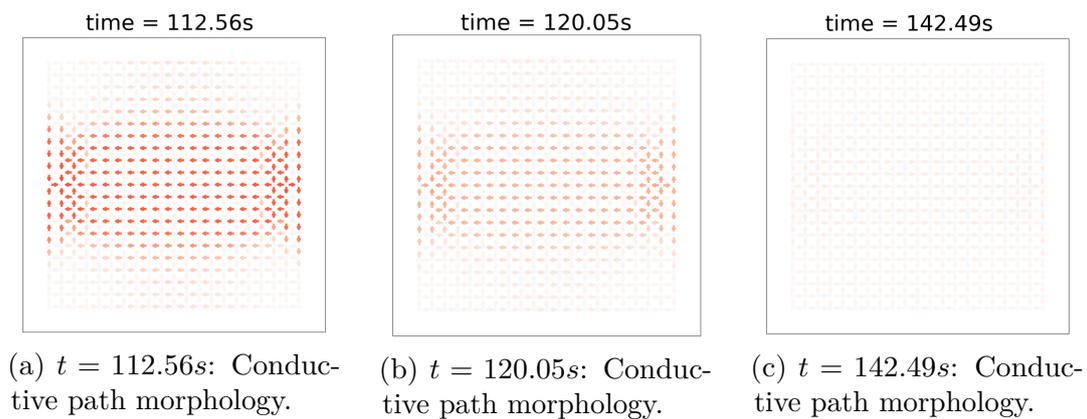


Figure 2.16: Spatial information on filament relaxation dynamic.

### 2.3.3 Hetero-synaptic plasticity

The grid model is a powerful tool to perform multi-terminal analysis and investigate hetero-synaptic properties of the network.

The experimental data are composed by the measurement of potentiation-depression conductance pattern across two stimulated electrodes and by the depression curve of each other electrode pairs. The pad mapping is depicted in figure 2.17, where the minimum grid 19x19 has been implemented. Four sets of data from [40] have been

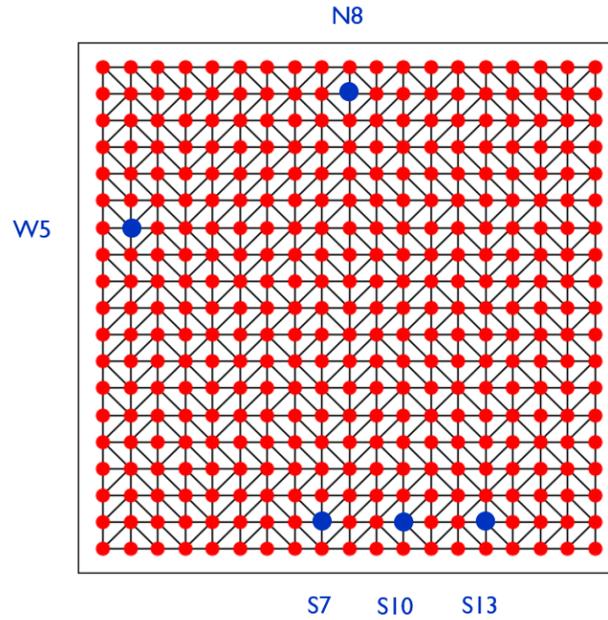


Figure 2.17: 19x19 grid and pad configuration for hetero-synaptic plasticity analysis.

analyzed, corresponding to the stimulation of pad N8 with respect to the others. For all the four cases, the followed approach has been the fitting of homo-synaptic potentiation-depression data in order to extrapolate both the parameters and the network dynamics. Second, the conductance dynamic for each pair of electrodes has been read and compared to experimental measurements [40] to check the model validity.

#### Set 1 (N8-W5)

Figure 2.18 shows both the fitted data and the conductive path formation after potentiation with a 8V input pulse, one second long.

The read data can be well visualized through correlation maps and colored plot: figure 2.19a and 2.19b depict resistance variation due to potentiation and relaxation dynamics for each pad pair, respectively. As it can be seen, here proposed *Model II* correlation matrix is able to match the experimental data from a qualitative and

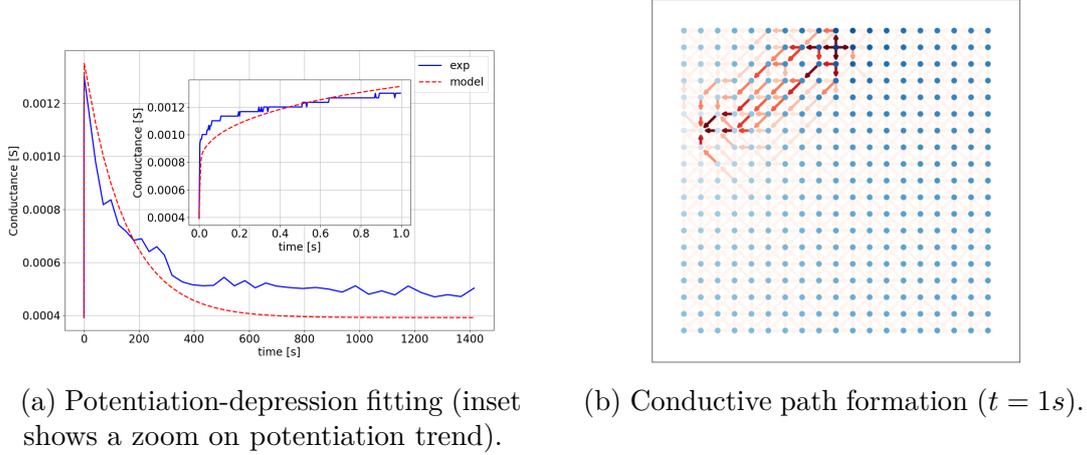


Figure 2.18: **Set 1**: Fitted data.

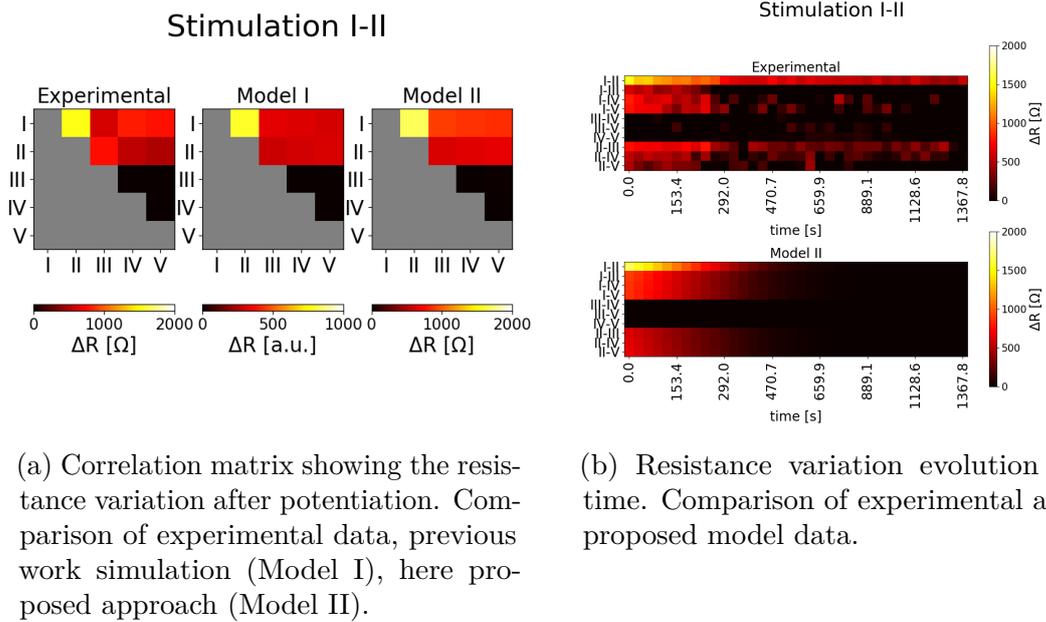


Figure 2.19: **Set 1**: Read data.

quantitative point of view, going further the *Model I* proposed in previous work [40], which is able to just describe the system phenomenologically (data expressed in arbitrary units). Moreover, simulated time dynamics is able to well reflect the experimental relaxing time with the correct resistance magnitude.

**Set 2 (N8-S7)**

Following the same analysis, figures 2.20 and 2.21 depict results for the *N8-S7* stimulation data. Having a look to the conductive path morphology, it is glaring that the potentiation of the network follows the field lines of the input electric field, along which the higher voltage gradient is translated in a higher stimulation probability.

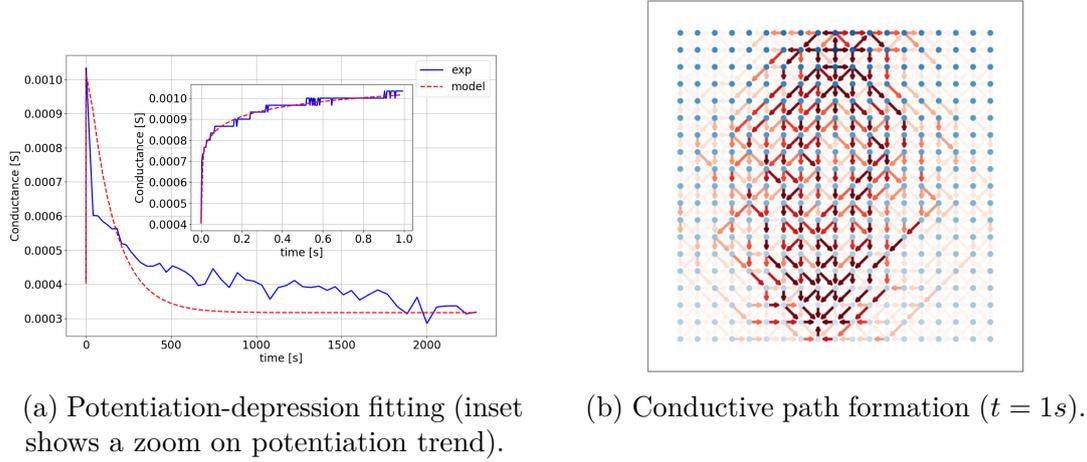


Figure 2.20: **Set 2**: Fitted data.

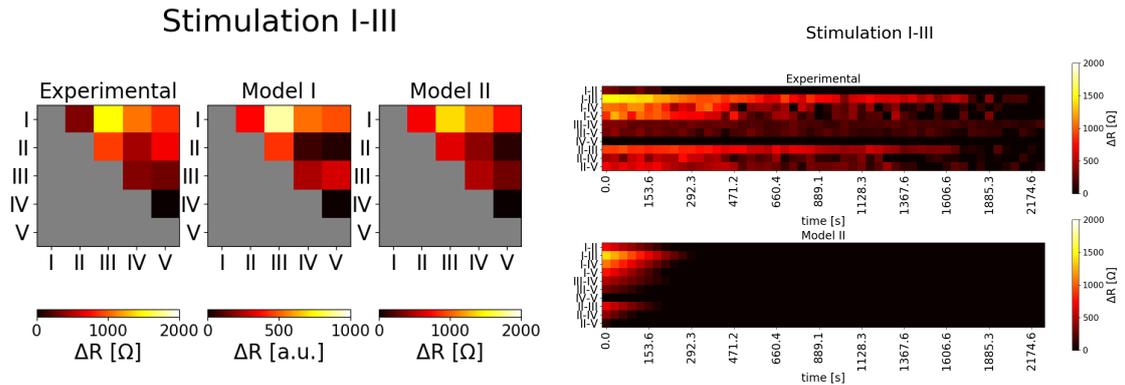


Figure 2.21: **Set 2**: Read data.

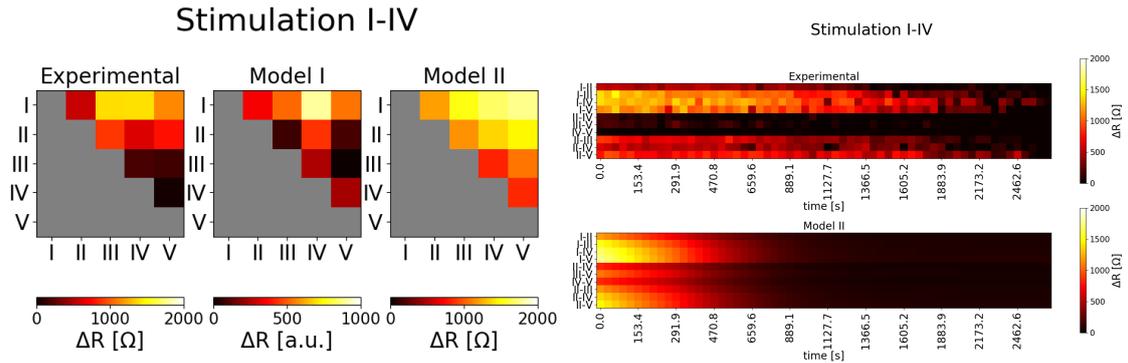
*Model II* correlation matrix in figure 2.19a, in addition, provides better phenomenological results than *Model I*, while adding an accurate quantitative description of experimental results, validating the proposed model. Relaxation behavior depicted in figure 2.21b provides a good quantitative description, but with some mismatch about the decaying time: this effect is already clear in the fitting of figure 2.20a, where the model curve reaches soon the minimum of conductance range. This is both linked to noisy experimental data and the approximation which are introduced by the model itself, which do not allow a perfect fit in depression region.

### Set 3 (N8-S10) - Set 4 (N8-S13)

If the first two sets of data are really encouraging, the last two highlight some issues about the model. Figure 2.22 reports the correlation matrices and time dynamics in the two frameworks.

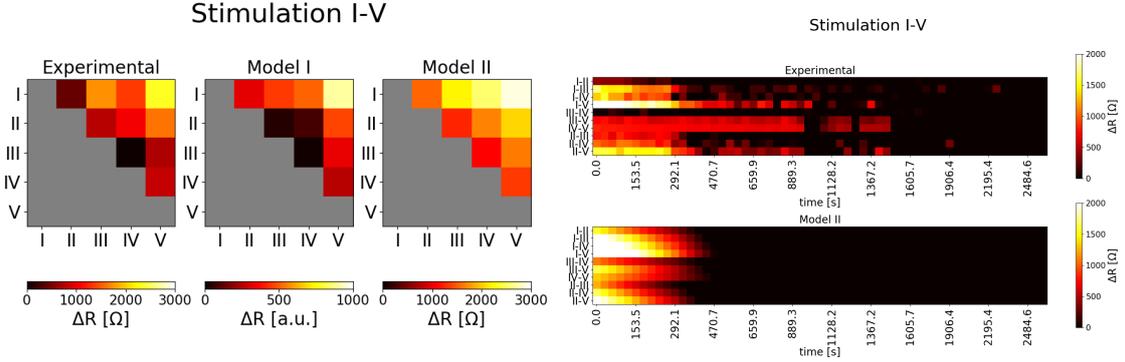
What is clear by looking at the correlation matrices, is the high degree of potentiation of the network with respect to experimental data. The qualitative behavior, however, is preserved, as the time dynamics also suggest, but on a slightly different resistance scale. The origin of this issue has to be looked for in the conductive path morphology, presented in figure 2.23 for both sets of data.

The network reaction to stimulus is not a generation of a conductive path, but almost the whole network is potentiated. These simulation results are in contrast with experimental observation and they are probably a result of the fitting process.



(a) Correlation matrix showing the resistance variation after potentiation. Comparison of experimental data, previous work simulation (Model I), here proposed approach (Model II).

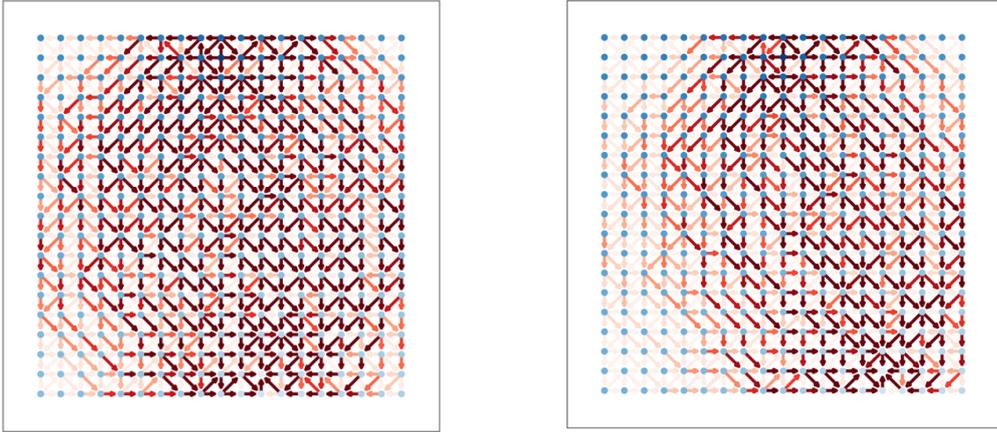
(b) Resistance variation evolution in time. Comparison of experimental and proposed model data.



(c) Correlation matrix showing the resistance variation after potentiation. Comparison of experimental data, previous work simulation (Model I), here proposed approach (Model II).

(d) Resistance variation evolution in time. Comparison of experimental and proposed model data.

Figure 2.22: **Set 3-4:** Read data.



(a) **Set 3:** Conductive path formation ( $t = 1s$ ).

(b) **Set 4:** Conductive path formation ( $t = 1s$ ).

Figure 2.23

The idea is, as already discussed, finding the set of parameters which minimize the error with the experimental data, but actually the space of parameters which approximate (up to a certain tolerance) the given data is not a point in the parameters space, but actually a volume containing different valid solutions. The correct fitting procedure, thus, should consider this effect by defining good starting fitting point to reach a meaningful behavior. This additional procedure has not been performed in this work for computational costs constraints.

### 2.3.4 Model Assessment

The analyzed grid model provides interesting insights about the network dynamics. First, it is able to reproduce the experimental observed conductive path formation to obtain meaningful information about its morphology and time evolution with persisting input signal. Second, it allows the measurement of conductance dynamics across non stimulated terminals. This will be of fundamental importance in reservoir computing simulations.

Overall the model behaves well, reproducing experimental data from a quantitative point of view, even if with some limitations. Of course it is a simplified model, which does not take into account advanced physical arguments. However, for this work purpose, with its simplicity, it is enough to reproduce the relevant network behavior at reasonable computational costs.

As outlined in the effective memristor model, also in this case it is difficult to identify a universal set of parameters for the data fitting, but they should be tailored on a particular stimulus shape. Even if this is not a negligible model limit, it is also true that devices usually work with standard input signals, for which fitting parameters can be induced.

In any case, the proposed model aim is not to describe at all the device, but to give some insights about the network dynamics, allowing more complex simulations. Being a compact model, indeed, the computational cost is much cheaper than low-level models described in Section 1.5 and this is fundamental to implement computing simulations.

# Chapter 3

## Reservoir Computing

### 3.1 Introduction

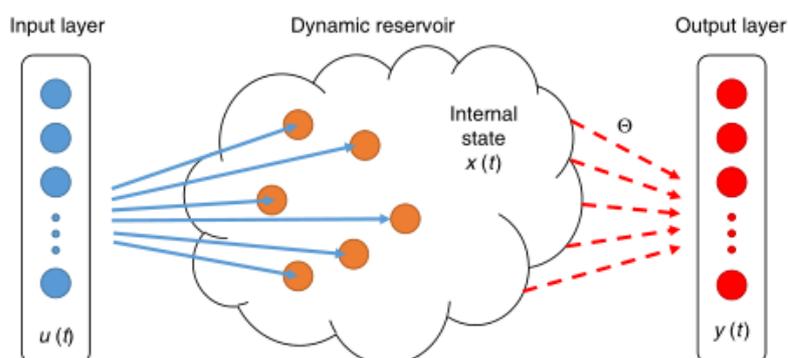


Figure 3.1: Reservoir computing general work principle. Reprinted from [69].

The reservoir computing (RC) paradigm was born with the need of solving complex problems by means of recursive neural networks (RNN). The use of recursive connections makes the network enough articulated so that a variety of tasks can be performed given a proper training process.

The latter, however, would result to be too computationally expensive when the network degree of complexity increases.

RNNs basically refer to a couple of training algorithms: *backpropagation through time* (BPTT) [53; 54] and *real-time recurrent learning* (RTRL) [55; 56]. The former works by unfolding the RNN in time and training it as if it was a forward neural network (FNN) with a backpropagation method [55], showing issues in long-term dependencies learning. The latter shows advantage in online learning, but with a too high computational cost.

For this reason, the introduction of a *reservoir*, seen as a black box, could mimic

the recurrent network in a more rapid, efficient and natural way.

The first approaches in RC were independently developed by Jaeger et al. [57] and Maass et al. [58]. The former dealt with the *echo state networks* (ESN), while the latter with the *liquid state machine* (LSM).

The reservoir may be physical or virtual: this work deals with the simulation of a physical one.

The competitive advantage of this computing approach relies in the training operation, which acts only on the readout function (one-layer neural network) which takes information from the reservoir and translate them into problem solution by a linear transformation. The great advantage of this computing approach relies on the lower computational cost and time to perform training. Moreover, in principle, a given reservoir may be equipped with several readout functions to have a general purpose system exploiting the same reservoir [71].

The basic idea, in other words, is to map the problem input into a larger dimension space identified by the reservoir, where the latter should be, then, the *magic hat* where some information are read and trained to get the output solution.

In order to perform RC, the reservoir should satisfy three important requirements:

- its elements must be able to store information
- it must be made up of independent units exhibiting a non-linear behavior
- it must exhibit the separation property
- it has to be designed such that the effect of an input on the reservoir must vanish after a certain time (*fading memory* [58])

The first is fundamental to ensure a memory effect to highlight input time correlations, the second is essential to solve complex tasks, the third is important to map different inputs on different reservoir states and the latter ensures a memory of recent past and not of distant past. The latter is also known as *echo state memory* [57].

Recent trends on RC outline how, despite it was born to deal with temporal pattern recognition, it can be exploited with many other machine learning problems by properly transforming the input data into temporal pattern.

The relevant applications of RC are linked to spoken digit recognition [59], human activity recognition [60], handwritten digit recognition [61], waveform classification [62], sine-wave generation [63] and so on.

Moreover, in principle any dynamical system can be used as a reservoir if it satisfy the previous cited requirements. Literature is available concerning mechanical [64], electronic [65], photonic [66], spintronic [67] and biological [68] reservoir type.

According to the definition of RC, self-assembly nanowires network represents a good candidate for the reservoir. The memory effect is guaranteed by the memristive nature of the single connected elements, the non-linearity behavior has been

experimentally proven and mathematically described in the previous chapter. Moreover, the separation property can be investigated by using different configurations setups as it will be discussed and the fading memory effect is guaranteed by the STP property of the network. Without the latter feature, this paradigm would not be implementable.

The idea to exploit these architectonics is to send pulse patterns to the network by means of multi-terminal configuration: the network would, then, be potentiated depending on the particular input structure exploiting the already discussed phenomena of paired-pulse facilitation, homo and hetero-synaptic plasticity.

Stated otherwise, each input produces different network evolution according to its spatio-temporal structure. Once the input is sent, conductances across multiple electrodes can be read and used as the input for the readout function.

## 3.2 NW network reservoir for written digit recognition

Image recognition is one of the most diffused task performed by means of neural network, thus represents a good benchmark analysis to test nanowire networks used in advantage to reservoir computing. In order to understand the potentialities and the limits of this approach, 5x4 written digits maps have been analyzed, taking inspiration from [69], where a five discrete memristors reservoir is used.

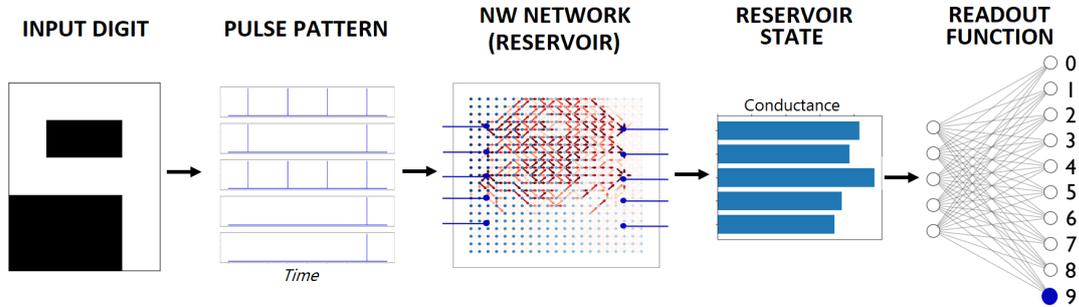


Figure 3.2: Implemented RC process to perform digit recognition. Example for digit '9'.

The overall implemented process is schematized in figure 3.2 and discussed in the following.

The idea is to exploit the system memory to compress the information from a row into a single value, which depends on time correlations. The advantages that such system could bring with respect to five discrete memristor reservoir [69] are:

- the network is tolerant to broken memristive junctions, providing alternative conductive paths. With discrete devices, just one broken memristor would result in inoperative reservoir
- hetero-synaptic plasticity may highlight spatial correlations, differently from independent devices which do not communicate with each other

### 3.2.1 Input and pulse stream

The adopted training dataset is made up of 10 unperturbed digits (figure 3.3a), while three testing datasets have been build up: 1-bit noise and 2-bit noise on the training one, literature [69] noise test data (in order to provide a benchmark). They are depicted in figures 3.3b, 3.3c and 3.3d respectively.

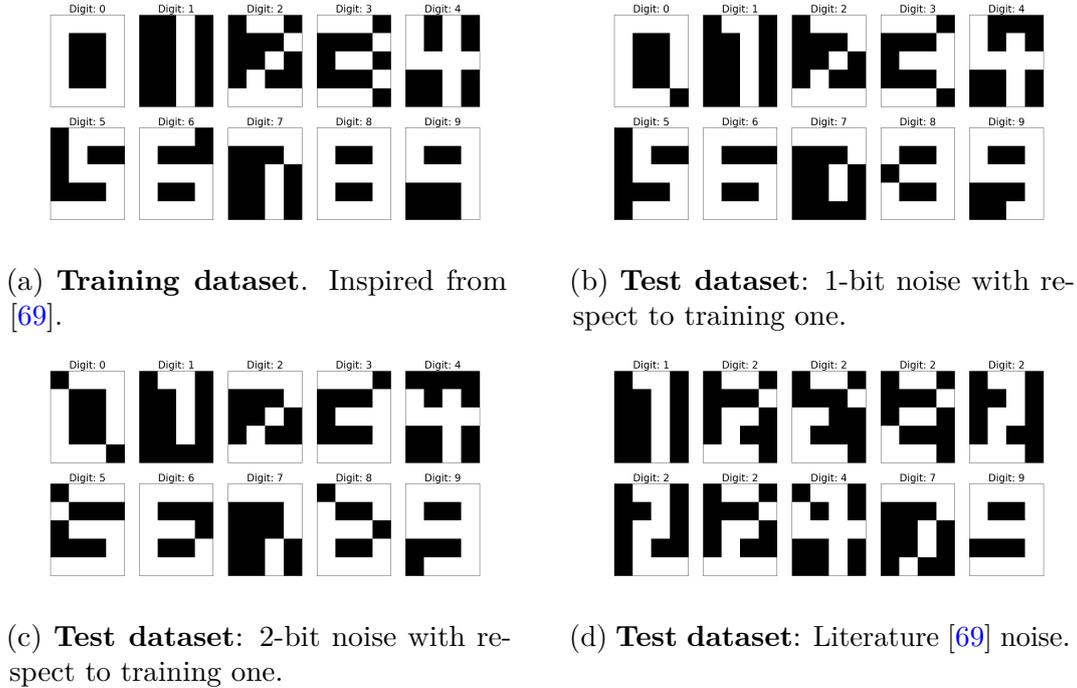


Figure 3.3: Training and test datasets.

It is here remarked that the literature dataset may not provide a good key performance indicator since it is composed of six repeated digit '2', where sometimes more than 2-bit of noise are considered. Simulation results will be, however, reported even if their significance is of minor interest.

The adopted system for the latter, as mentioned, is a reservoir composed of 5 discrete memristors, through which it can be possible to reach an accuracy of 80% [69].

Each digit is translated into a collection of five independent voltage signals (one each row) with a pulse every time the column bit is a logic '1', otherwise the signal is zero.

Pulse patterns have been constructed with amplitude and timing customized on the network parameters (which will be discussed in the following), as shown in figure 3.4.

Different timing have been analyzed, where the proposed one is the more performing according to the chosen network parameters.

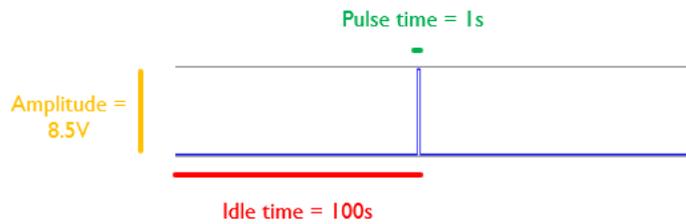


Figure 3.4: Pulse structure corresponding to each logic '1' bit of each digit map.

### 3.2.2 NW network reservoir

The reservoir is represented by the previous described network grid model, adopting the random diagonal configuration with a grid dimension of 21x21 in order to correctly map five electrodes for each side of the network.

Given the model, the parameters have been defined according to the fitted data from *Set 2* in hetero-synaptic analysis in order to provide a plausible dynamics of the network.

For each digit, the five inputs are sent to the network by exploiting the pad configuration depicted in figure 3.5, where each source electrode on the left is associated to the correspondent ground node on the right. In this first approach, when the signal is low the input is disconnected from the network and kept floating, in order to avoid parasitic potentiation among vertical electrodes.

The reservoir, after the temporal sequence of pulses characteristic of each digit map, exhibits different potentiation patterns. Some of them are reported in figure 3.6 for digit '2', '5' and '7', respectively.

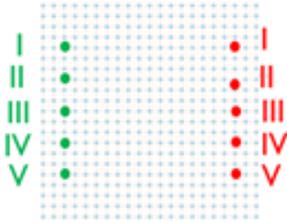


Figure 3.5: Five sources and respective five grounds configuration adopted for network stimulation.

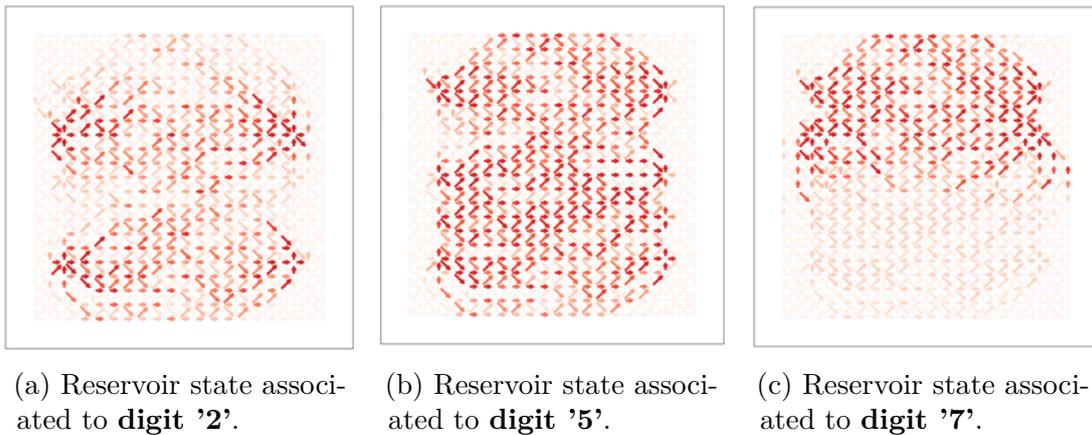


Figure 3.6: Reservoir states shown in terms of network edge conductances after potentiation.

What is interesting here is that the network final states visibly exhibit a differentiation for different input pulse streams. In particular the network exhibits higher spatial potentiation according to where the digit map is denser in white pixels. This process maps the input (20 binary pixel) on a much higher space (1240 analog conductance edges).

In order to quantify the reservoir state and extrapolate from it relevant features to classify, five conductance values are read from the five pairs of electrodes adopted during stimulation. Following this process, each digit will have its own fingerprint considering the 5-values reservoir state, shown in figure 3.7. As it can be noticed, some histograms are similar despite the different digit, as in the case of digit '0', '1', '8' and '9'.

This non perfect *input separation* is linked to the fact that the memory effect is more evident on the last column of the digit map. In case of cited similar digits, the last non-zero column is full of white pixels, resulting in a similar output pattern.

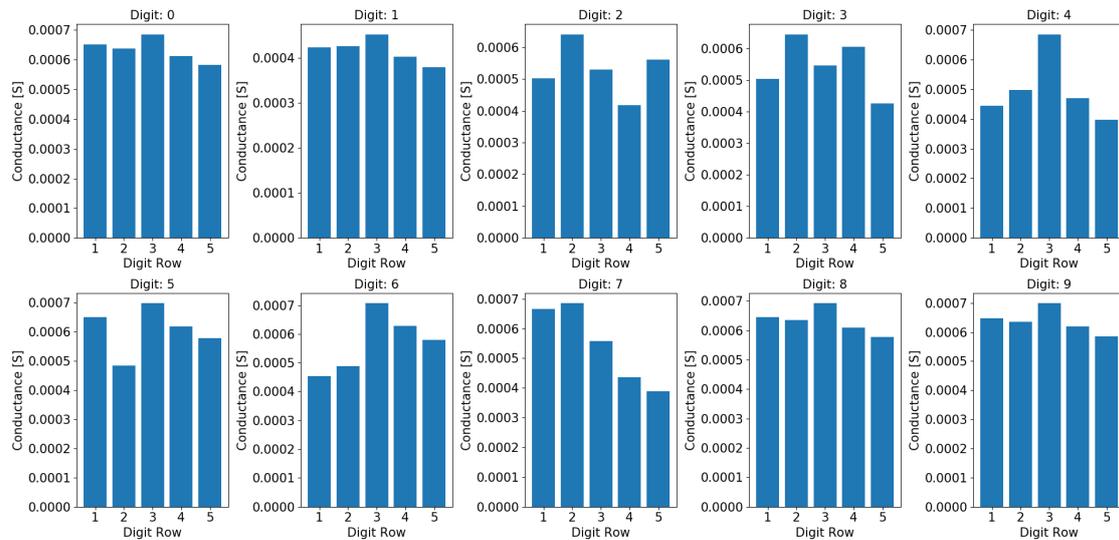


Figure 3.7: Readout input for each train digit.

### 3.2.3 Readout function

Once obtained the fingerprint of each input, the five conductance values become the inputs of a linear transformation which associates the five element vector to the actual digit class to recognize. The classes are identified by the 10 different digits to recognize.

As mentioned before, this is the unique element of the whole system which needs a training process.

In this case a supervised learning has been implemented through a 5x10 neural network, developed in Python environment exploiting the library *tensorflow.keras*. Here it is reported a code excerpt used for readout implementation:

```
[...]
sc = StandardScaler()
training_inputs = sc.fit_transform(training_inputs)

ohe = OneHotEncoder()
training_outputs = ohe.fit_transform(training_outputs).toarray()

model = Sequential()
model.add(Dense(10, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam',
              metrics=['accuracy'])

history = model.fit(training_inputs, training_outputs, epochs=1500,
                   batch_size=64)

[...]
test_inputs = sc.fit_transform(test_inputs)
y_pred = model.predict(test_inputs)
```

[...]

As it can be seen, the output classes (both for train and test) have been encoded with the one-hot representation, while inputs (both for train and test) have been scaled through the function *StandardScaler()* in order to avoid accuracy oscillations during training. The basic working principle of the NN is a matrix-vector multiplication between the weights (which define a 2D 10x5 matrix) and the input vector (5x1). The output (10x1) is a ten-values vector interpreted as probabilities, where the highest one defines the predicted class.

### 3.2.4 Training

Due to the extreme simplicity of the readout function, its training process will result in very low computational cost, hence time.

The training process deals with a backpropagation algorithm which minimizes a certain loss function associated to the classification correctness. Its working principle is to initialize the neural network with a random distribution of weights, then inputs from figure 3.7 are sent to the network and it is checked whether the classification is correct (knowing the desired output, from here it is a supervised learning). A loss function is calculated on the basis of guessed classes and the NN weights are adjusted to minimize it: this process is re-iterated up to the defined *epochs* of iteration. As the loss is minimized, the training accuracy increases. A plot of training history with respect to epochs is shown in figure 3.8.

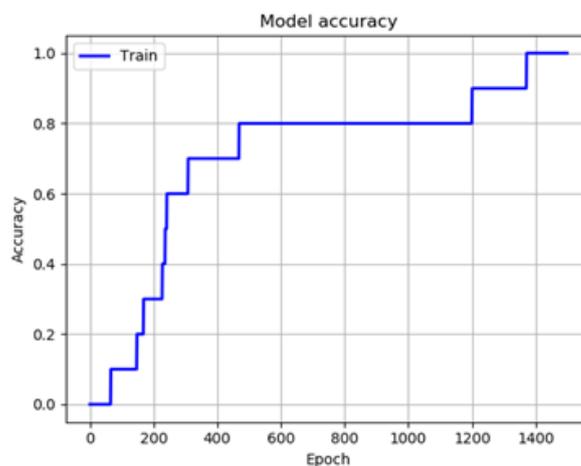


Figure 3.8: Model accuracy on training data over 1500 epochs.

## 3.3 Results

### 3.3.1 Testing

Once trained the readout function, the NN weights configuration is fixed. The NN is ready to receive new inputs and classify them.

Testing on the three defined noisy datasets has been performed to analyze the recognition capability.

The outcome accuracy is:

- 70% for 1-bit noise data
- 50% for 2-bit noise data
- 40% for literature data

These results suggest that the overall system performs well, even if some improvements are necessary. It is important to stress that 1-bit and 2-bit noise applied on 5x4 bit map may lead to a discrimination between two different digits of just 1 or 2 cells, providing lower accuracy of the system (linked to a lower separability of reservoir states). Results also depend on network model parameters, which lead to different potentiation patterns at fixed input signal.

### 3.3.2 Optimization of electrodes configuration

A possible way to improve the classification accuracy is to play with the spatial arrangement of the electrodes. Moreover, it can also be used a single ground electrode as a common reference for sources or, in addition, replicate some inputs on different location of the network. Some of the analyzed configurations are presented in figure 3.9. Among all, the best scenario is the *Configuration 1* in figure 3.9a which brings the literature data accuracy from 40% to 60%. However, the benefit on the other two test datasets is absent.

This demonstrates how the spatial characteristic of inputs may influence the system performance.

The studied optimization is constrained to peripheral electrodes disposition, according to the physical studied device. Surely, an optimization operation over the whole device area could be performed, but this goes further the presented device potentiality. However, for the sake of completeness, the configuration in figure 3.10 has been tested, emphasizing a major advantage with respect to peripheral electrodes:

- 80% for 1-bit noise data
- 70% for 2-bit noise data
- 60% for literature data

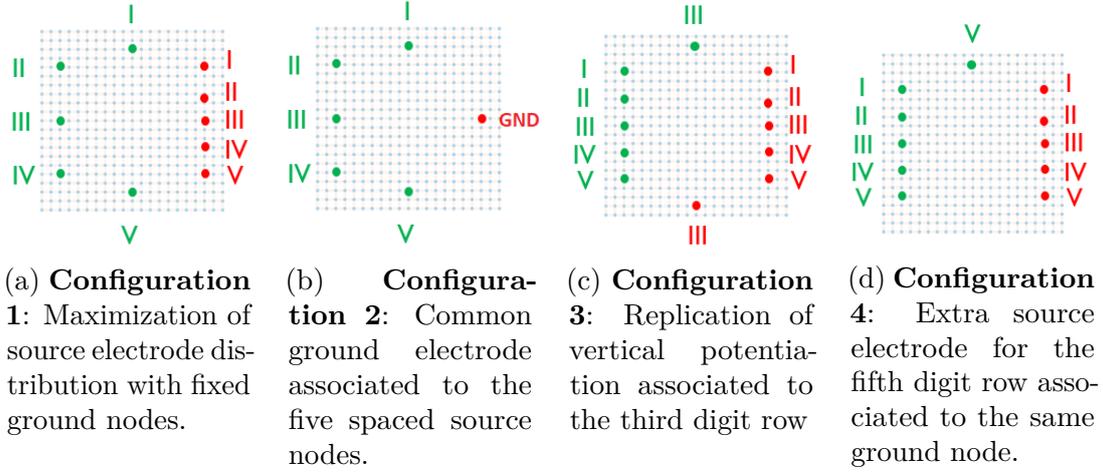


Figure 3.9: Proposed electrodes configurations. **Green:** Source, **Red:** Ground.

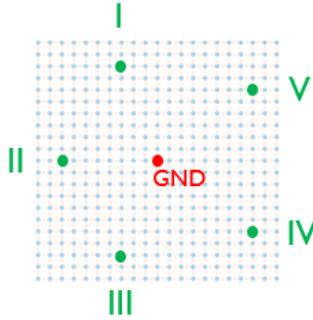


Figure 3.10: Spatial optimization of electrodes by exploiting the whole network area.

### 3.3.3 Effect of input processing

Stated the limitation in managing the spatial distribution of electrodes, an operation of input processing can be performed to build a new dataset in one-to-one relationship with the original one, such that the state separation would result more evident. The used electrode configuration is the original one (figure 3.5) in order to discard the effect of electrodes managing.

In this work it is proposed a two-steps processing operation schematized in figure 3.11:

- First, the digit map elements are shifted of one position on the right, going on the next row when at the right border (figure 3.11a)
- Second, considering the digit map as a matrix composed of '0' and '1', the new obtained is subtracted from original one element by element (figure 3.11b)

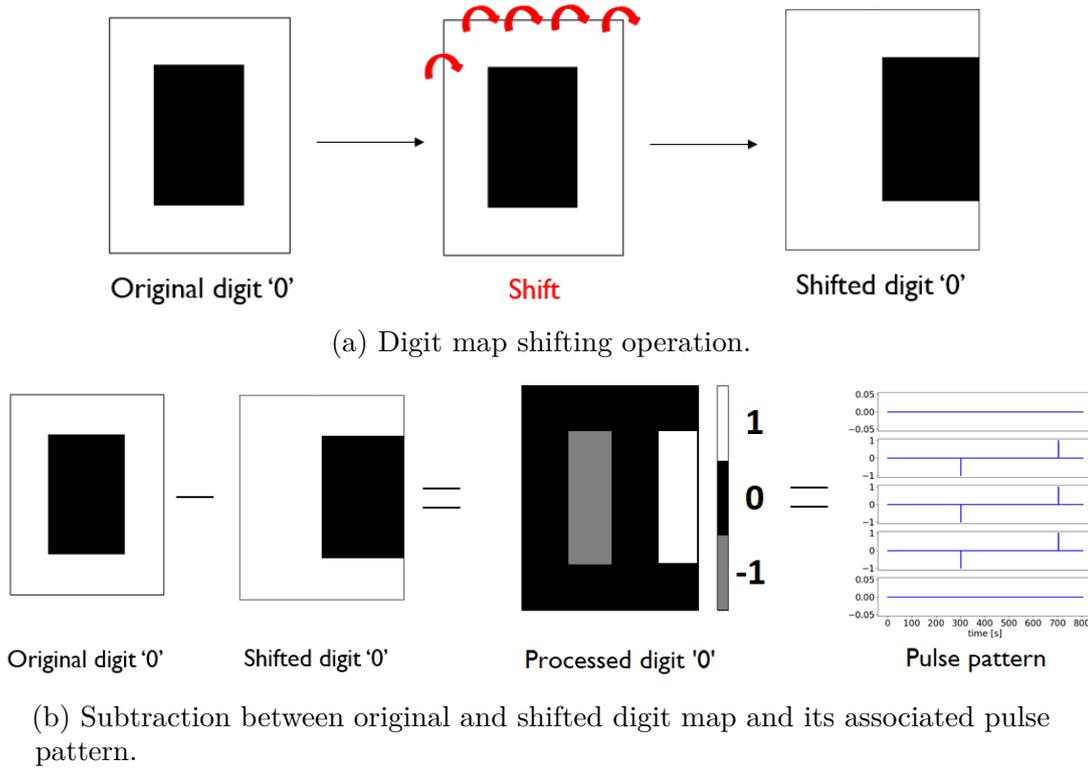


Figure 3.11: Input processing: example on digit '0'.

As it can be deduced, the new digit map is now composed of ternary information given by values -1, 0, 1. This brings to the use of negative pulses as well. From an experimental point of view, this operation should be performed carefully, since the maximum applied voltage amplitude is two times higher than previous case and the network may be stressed over its safe operating area.

However, the idea behind this operation is to emphasize the contours of the written digit. While other shifting procedures before subtraction operation have been investigated, such as 2 and 3 positions shift, the here proposed is the most performing one.

Accuracy obtained in this framework are:

- 90% for 1-bit noise data
- 70% for 2-bit noise data
- 40% for literature data

showing a substantial improvement of the classification capabilities for the first two scenarios.

Moreover, as an additional confirmation, the training history in figure 3.12 shows a

100% training process after about 500 epochs (against 1500 in the original scenario).

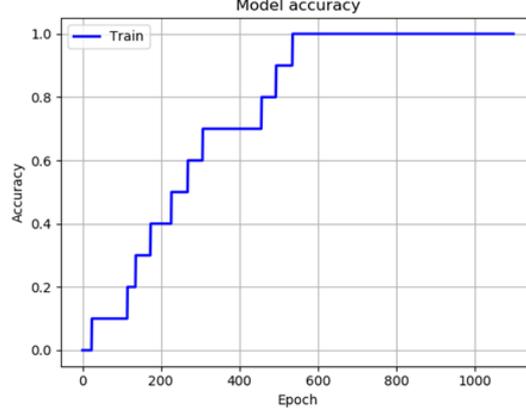


Figure 3.12: **Input processing**: training accuracy history

### 3.3.4 Managing of ground and floating nodes

Up to now, the low-level input signal has been equivalent to keep source and ground nodes floating and letting them evolve according to other stimulation inputs effect. From an experimental point of view this introduces some limitations on the measurement apparatus, which should be equipped with as many relays as the source and ground nodes are. For this reason, the use of always connected signals has been investigated, meaning that low signals correspond to ground level instead of floating node.

The analysis has been performed in all the relevant scenario presented up to now and summarized in table 3.1.

Test Dataset	Case 1	Case 2	Case 3
<i>1bit Noise</i>	40%	40%	50%
<i>2bit Noise</i>	50%	50%	60%
<i>Literature</i>	30%	40%	30%

Table 3.1: Accuracy using ground nodes instead of floating ones.

**Case 1:** Electrodes in fig. 3.5 without input processing

**Case 2:** Electrodes in fig. 3.9a without input processing

**Case 3:** Electrodes in fig. 3.5 with input processing

The obtained results are not really encouraging, even if electrodes optimization or input processing is performed. Parasitic potentiation among electrodes leads to a lower state separability due to hetero-synaptic effect.

### 3.3.5 Effect of different readout functions

To ensure that the one-layer neural network is the most suited readout function in this case, other linear transformation algorithms have been implemented and tested.

In view of building up a fully memristive system, *linear regression* classification has been implemented by the construction of the pseudo-inverse matrix of the problem, as shown in the following code lines:

```
[...]
pseudo_inv = np.linalg.pinv(training_inputs)
weights = np.matmul(pseudo_inv, training_outputs)
y_pred = np.matmul(test_inputs, weights)
[...]
```

As presented in [72], the pseudo-inverse matrix can be accelerated by a one-shot process by means of a memristive crossbar array.

By analyzing the accuracy, this approach is not able to classify at all the data, leading to a performance around 10%, equivalent to a random guess of the digit.

Another implemented algorithm is the *logistic regression* one, according to the following Python code lines:

```
[...]
sc = StandardScaler()
training_inputs = sc.fit_transform(training_inputs)
test_inputs = sc.fit_transform(test_inputs)
model = LogisticRegression(C=1.0, class_weight=None, dual=False,
                           fit_intercept=True, intercept_scaling=1, l1_ratio=None,
                           max_iter=100, multi_class='ovr', n_jobs=None,
                           penalty='l2', random_state=0, solver='liblinear',
                           tol=0.0001, verbose=0, warm_start=False)
model.fit(training_inputs, training_outputs)
y_pred = model.predict_proba(test_inputs)
[...]
```

Results are more performing than linear regression, even if slightly worse than one-layer neural network, confirming that the initial choice was the best.

### 3.4 Discussion

The presented simulations have outlined how the self-assembly Ag nanowire network can be involved in reservoir computing. This dissertation has highlighted some ideas to improve system efficiency by acting on different process configurations to stress spatio-temporal correlations.

Moreover, even if the obtained results seem to be encouraging, it is important to remember that the problem dimension is low (5x4 digit map). Identical simulations have been performed on a bigger network considering a small subset of the *Mnist* dataset (35 train digits, 10 test digits) which defines a higher dimension problem (28x28 digit map). Results show an accuracy falling down to 20%-30%.

The optimization, however, is still opened: the system is still far from a competitive advantage with respect to other existing classification paradigms.

Other memristive RC paradigm, however, shows a high potentiality of this technology.

Wei et al. [69], in fact, have used a reservoir with 88 discrete memristors to classify the Mnist dataset. By training over 14.000 samples and testing over 2000 ones, accuracy of 88.1% is experimentally achieved. Further investigation have been performed to analyze the ideal accuracy by simulations, obtaining 91.1% recognition (simulations eliminate cycle-to-cycle variations).

Wu et al. [70], still adopting a parallel memristors RC reservoir, have been able to classify the MNIST dataset (18.000 train, 2.000 test samples) with an experimental accuracy of 97.6% (close to software simulations of 98.0%). This high percentage has been obtained through a proper designed input processing.

Considering the proposed scenario, NW networks have been exploited to solve similar problems which well suit other memristive reservoir structures. Actually, the potentialities of this self-assembled system need to be further explored in case of different datasets, which may be better processed by the network according to its characteristics.

Since these networks are characterized by millisecond scale dynamics, the competitive advantage, with respect to other computing paradigms, should refer to energy efficiency, low training time and complex task performances, rather than speeding up the single operation.

A proper problem to solve, going in this direction, may be *speech recognition*, for which the millisecond scale is not a constraint. The network, responding to time correlations, may be suited for this task.

Keeping the same dataset, on the other hand, an optimization over pads can be investigated by depositing electrodes not only on the periphery but also on the area of the device (as it has been already marginally discussed).

A further step can be done on network topology by designing different regions with

variable memristive junctions density and weakly connected among them. In this way, the different dynamics for the same input may help to extract more relevant features and better separate states.

Finally, multiple reservoirs may be designed and perform a different input processing before each of them: this case should be, however, carefully designed to guarantee technology advantages despite the selected number of reservoirs to use.



## Chapter 4

# Fully-Memristive Classification

Reservoir computing, as discussed, introduces new conceptual and practical advantages in efficient computing.

Memristor based architectures have shown to be energy-efficient with the possibility to perform complex operations one-shot [72; 38], as already mentioned. The idea to build up a fully memristive classification system, thus, can bring non negligible benefits to the whole process performances.

The reservoir is already a memristive device, as deeply studied previously, so, in order to demonstrate the possibility to build up a fully memristive system, the implementation of the readout function on memristive cross-bar array has been investigated, both concerning training and testing.

Moreover, a benchmark will be provided concerning two fully memristive systems for classification:

- Reservoir (nanowire network) + one-layer neural network (crossbar array)
- Two-layers neural network (crossbar array)

### 4.1 Memristive Cross-Bar Array as a Hardware Neural Network

The basic operations a memristive cross-bar can perform are matrix-vector multiplication and the rank one outer product [73]. It is important to remark that the crossbar array, differently from self-assembled nanowire networks, is a *non-volatile* device: once the conductance state of each cell is written, no spontaneous relaxation occurs in case of zero applied signal. This is of fundamental importance, since once the weight (i.e. conductance of each cell) are trained, they should remain unchanged

to guarantee a proper classification.

These two operations are at the base of the working of a neural network, where the former allows to transform the vector values of a layer to the vector values of the successive layer, while the latter is fundamental for weight update during training. In order to train a neural network, a back-propagation algorithm needs to be implemented, whose working principle has been already discussed in section 3.2.4. An open source Python API (*CrossSim*) modeling resistive memory crossbar developed by *Sandia National Laboratories* has been exploited and tailored to this study specific purposes [74; 76].

#### 4.1.1 Cross-bar array architecture for *on-chip* training

The hardware modeled in *CrossSim* is presented in figure 4.1.

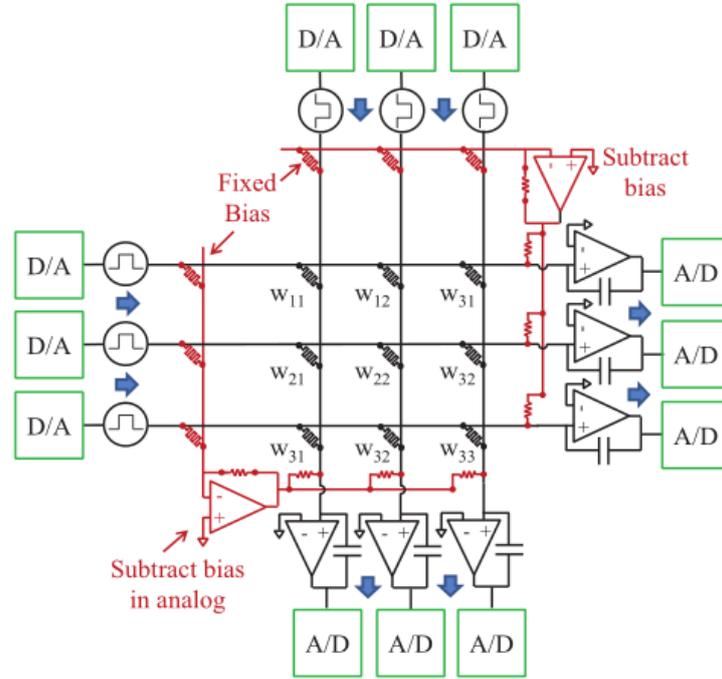


Figure 4.1: Neural core implemented in *CrossSim*. Reprinted from [74].

In addition to the memristor matrix (whose conductance weight are labeled as  $w_{ij}$ ), an extra row and column are implemented: their role is to provide an analog bias subtraction in order to map negative weights on positive conductance values. Moreover, digital inputs must be first converted into analog data by a digital-analog converter (DAC) in order to exploit the analog nature of memristors. Dually, the output must undergo the reverse operation of analog-digital conversion (ADC). The use of this hardware device works by mapping the weights of a software neural

network on physical conductances of the crossbar. The latter is an information stored in look-up tables (LUT) defined for the particular technology to use (from experimental data).

In order to implement the back-propagation, there is the need of a neural core for each layer, which in the case of a one-layer NN is just one. Moreover, a digital core is necessary to compute the sigmoid derivative before the weight update, as schematized in figure 4.2.

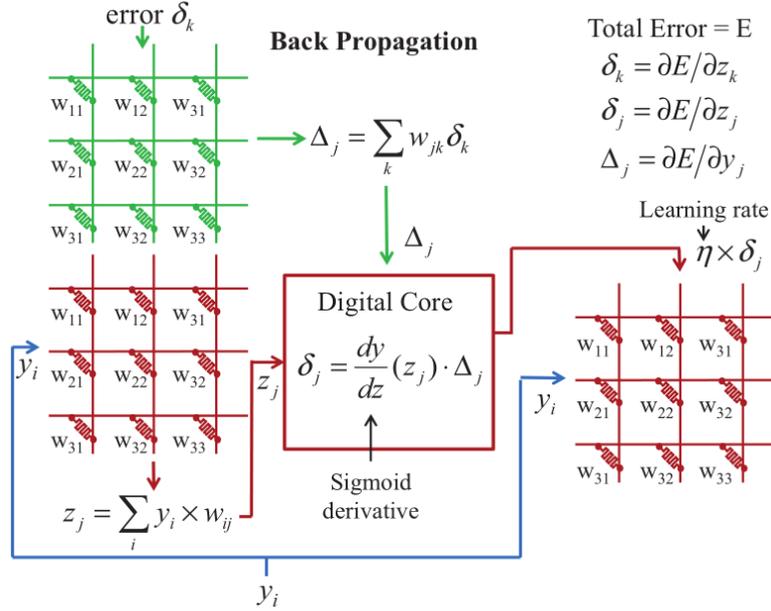


Figure 4.2: Back-propagation algorithm hardware scheme. Reprinted from [74].

The steps for each input data required for a backpropagation algorithm, after random initialization of cell conductances  $w_{ij}$ , on memristive crossbar array are [75]:

- Apply the input data ( $y_i$ ) on the crossbar row to perform MVM and evaluate output neurons values:

$$z_j = \sum_i y_i w_{ij}$$

- Calculate the associated error between the output neuron and the desired target ( $t_j$ ):

$$\Delta_j = t_j - z_j$$

- Back-propagate the error to exploiting, again, MVM operation:

$$\Delta_j = \sum_k \delta_k w_{kj}$$

- Compute the weight update:

$$\Delta w = \eta \frac{dy}{dz}(z_j) \Delta_j y_i$$

- Update conductances through the outer product by sending on rows information of  $y_i$  and on columns information about  $\frac{\Delta w}{y_i}$ .

This process is performed up to error convergency.

In order to ensure a well operating hardware, different parameters of the neural core need to be defined:

- Clipping, the adopted software weights range
- Row/Column input/output range and bit resolution
- $G_{min}$  and  $G_{max}$ , corresponding to the actual relative conductance range used from experimental LUT
- Learning rate, which is the step size to approach the minimum of the loss function
- Epochs, that are the number of iterations performed during training
- $G_{on}/G_{off}$ , a finite number to express the ratio between the maximum and minimum device conductance
- Dataset scaling, to avoid training accuracy oscillating

### 4.1.2 Device Simulations

Despite the use of look-up tables which store the real experimental behavior of the selected device, what makes the simulations close to reality is the implementation of non-idealities in *CrossSim*: read noise, write noise and write non-linearity [74]. *Reading noise* is attributed to three main effects: thermal noise, pink noise and random telegraph noise (RTN). The latter is the dominant effect, which can be shown to be modeled as a Gaussian noise in this situation (due to the central limit theorem applied on the sum of many read noise contributions). Moreover the Gaussian distribution is also able to approximate the effect of the other two noise effects.

*Write noise* is typically higher than *reading* one. It is assumed to be both Gaussian (for the same considerations of before) and dependent on the conductance variation generated in write process, while independent on initial conductance state.

*Non-linearity noise* arises from piloting the device with same sized pulses, even if the conductance variation depends on the starting state. For perfect linear responses,

this error is zero. In general, a model [78] to compute the actual conductance variation dependent on number of pulses and asymmetry can be used to get the associated error.

In this work, the readout function has been implemented and simulated by considering different technologies, LISTA and ENODE, as discussed in the following. The neural core parameters to optimize the accuracy should be defined depending on the particular technology. However, some parameters have been set in the same way for all analyzed cases:

- Epochs: 1500
- Learning rate: 0.01
- $G_{on}/G_{off} = 10$
- Dataset scaling: through function *StandardScaler()*, as described in section 3.2.3
- Row/Column input/output range and bit resolution: default settings from Python API

The analyzed technologies refer to a *Lithium Ion Synaptic Transistor* [76] operating with input current pulses (*LISTA\_current*) and voltage pulses (*LISTA\_voltage*) and a *Electrochemical Neuromorphic Organic Device* (ENODE) [77]. A cell structure for the two technologies is presented in figure 4.3.

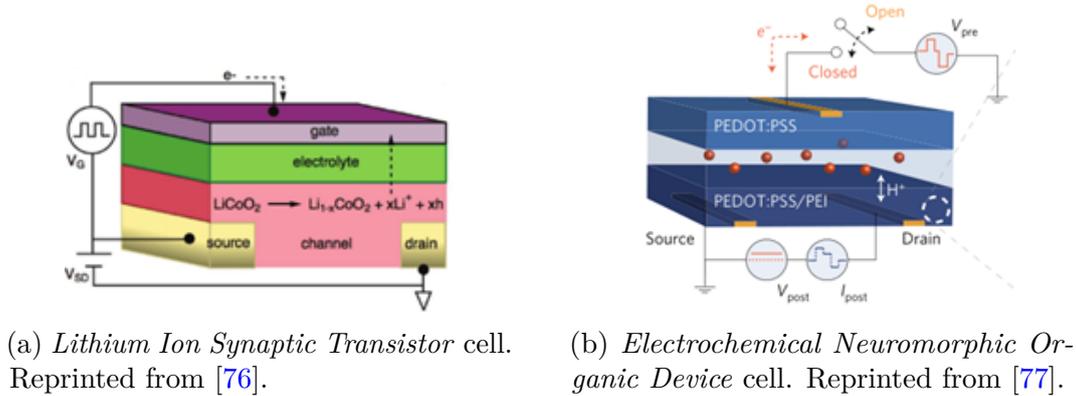


Figure 4.3

**LISTA**[76] The working principle of lithium synaptic transistor, as the name suggests, is based on the intercalation of lithium ions thanks to a solid electrolyte. The latter allows the removal of lithium ions from the  $LiCoO_2$ -based channel with

a subsequent generation of positive charged polarons: transition from insulating to conductive behavior is obtained, reaching almost six order of magnitude difference in conductivity. The process is reversible upon voltage sign inversion.

The ion intercalation, differently from other resistive switching devices, does not produce structure modifications, generating a longer device lifetime.

Moreover, low voltages are necessary to switch the device:  $10\text{ mV}$  are sufficient, with a projection of  $< 5\text{ mV}$  for sub-micrometer devices. The associated write energy is of the order of units of  $aJ$  for sub-micrometer devices (projection), while the read energy sets the limit around  $1\text{ fJ}$ .

However, the absence of parasitic leakage current and the strongly linear write behavior let this device a good choice for neuromorphic solutions.

**ENODE**[\[77\]](#) The ENODE technology works by decoupling the write and read operations. During write operation, a positive voltage is applied to the pre-synaptic (PEDOT:PSS) electrode, so that a flux of ions through the electrolyte reaches the post-synaptic (PEDOT:PSS/PEI) electrode. This let the PEI (poly-ethylenimine) to be protonated, causing a conductivity variation. During the reading operation, instead, the cell is disconnected and the conductance state remains unaltered thanks to the electron-blocking behavior of the electrolyte.

The potentialities of this technologies are linked to the existence of more than 500 distinct and non-volatile states within  $\sim 1V$ . Moreover, the switching energy has been projected to around  $35\text{ aJ}$  for sub-micrometer devices. Since it is based on an organic structure, it is a good candidate for technology-biology interfaces. Not less important is the flexible mechanical property of these devices, which opens possibility to 3D integration to reach the efficient connectivity of the human brain.

## 4.2 Results

### 4.2.1 One-layer NN classification

Two frameworks have been analyzed in the following simulations: both with the pads configuration as in figure [3.5](#), one without input processing, the other performing the already discussed shift and subtraction operation on input data.

The clipping ranges defined for the three cases are:

- LISTA\_current:  $[-1, +1]$
- LISTA\_voltage:  $[-1.5, +1.5]$
- ENODE:  $[-1, +1]$

The effect of this parameter is to limit the weight distribution and optimize the conductance dynamic range of memristive cells. The effect is an accumulation of weights at the extremes of the range, as demonstrated in figure [4.4](#).

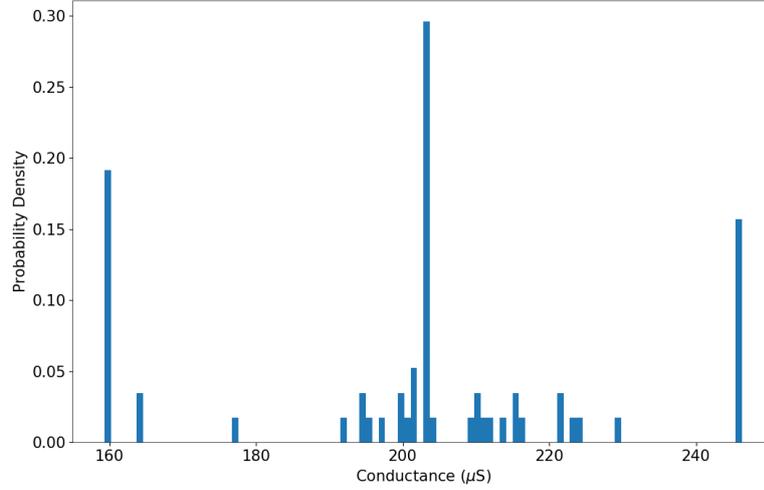


Figure 4.4: **LISTA\_current**: conductance weight distributions after training, showing accumulation at range extremes due to clipping parameter.

The defined  $G_{min}$  and  $G_{max}$ , on the other hand, have been defined as:

- LISTA\_current: [0.25, 0.75]
- LISTA\_voltage: [0.25, 0.75]
- ENODE: [0.10, 0.90]

The range selects the desired relative conductance dynamic from LUT, where an example of it is shown in figure 4.5.

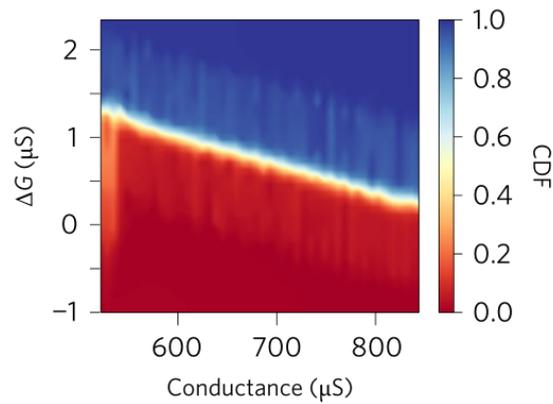


Figure 4.5: **ENODE**: Look-up table for increasing conductance writing. Conductance increase versus initial conductance state and cumulative distribution function. Reprinted from [77].

Simulation results are summarized in table 4.1 and 4.2. Beside the already stated better performance of framework 2, the highest classification accuracies are obtained with LISTA device driven with current pulses. Those percentages are really close to previous investigated software ones. ENODE also performs well, even if slightly worse than LISTA\_current.

Device	Train	1bit	2bit	Literature
<i>LISTA_current</i>	80%	70%	60%	50%
<i>LISTA_voltage</i>	80%	70%	50%	40%
<i>ENODE</i>	80%	70%	60%	40%
<i>Software</i>	100%	70%	50%	50%

Table 4.1: **Framework 1:** Electrodes configuration as in 3.5, without input processing.

Device	Train	1bit	2bit	Literature
<i>LISTA_current</i>	100%	80%	80%	40%
<i>LISTA_voltage</i>	80%	60%	50%	20%
<i>ENODE</i>	90%	80%	60%	40%
<i>Software</i>	100%	90%	70%	40%

Table 4.2: **Framework 2:** Electrodes configuration as in 3.5, with input processing.

### 4.2.2 Two-layers NN classification

In order to understand the advantage introduced by the reservoir, simulation using a two-layers neural network have been performed. In other words, the reservoir has been removed and the whole classification is entrusted to a 20x15x10 neural network (figure 4.6) implemented on two memristive crossbar arrays in cascade. Parameters of the two cores have been set identical among each other, given a certain technology.

Results are provided in table 4.3. Besides train accuracy after 1500 epochs, it is significant to observe the associated history. Figure 4.7b shows the need of just 400 epochs to reach 100% training accuracy.

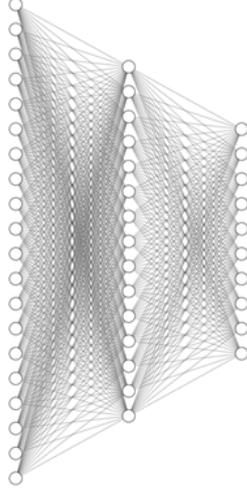


Figure 4.6: 20x15x10 Neural network scheme.

Device	Train	1bit	2bit	Literature
<i>LISTA_current</i>	100%	100%	80%	50%
<i>LISTA_voltage</i>	100%	90%	80%	50%
<i>ENODE</i>	100%	100%	90%	70%
<i>Software</i>	100%	100%	90%	50%

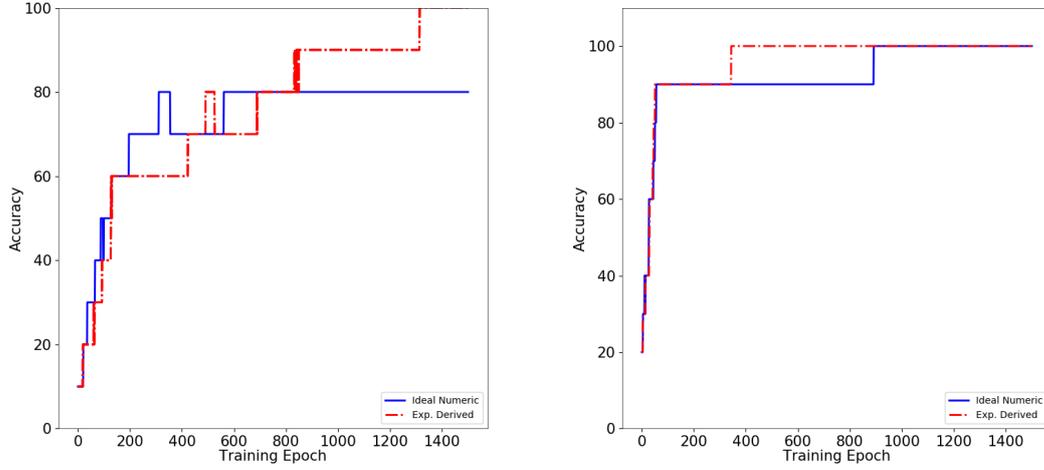
Table 4.3: **20x15x10 Neural Network**: Classification accuracy.

### 4.3 Discussion

By looking at results from reservoir computing with one-layer NN, it is clear how it is not limiting accuracies with respect to previous software simulation, confirming the possibility to build up a fully memristive classification.

In addition, previous results leave room to a discussion about the better paradigm (among the two proposed) to choose. Different considerations come into play and allow the assessment of the system:

- Training and testing accuracy
- Training time
- Chip area
- Power consumption



(a) **LISTA\_current, 1-layer NN:** Training history for reservoir + 5x10 neural network classification system.

(b) **ENODE, 2-layers NN:** Training history for 20x15x10 neural network classification system mapped over two crossbars in cascade.

Figure 4.7: Training history comparison between reservoir + one-layer NN system and two-layers NN one.

By merely looking at accuracies, it is evident that the two-layer NN is able to better classify data in general. Moreover, looking to accuracy histories in figure 4.7, it is clear that the the 2-layers NN is trained over a lower number of epochs with respect to 1-layer NN (400 against 1300, respectively).

However, introducing two bigger neural cores results in the need to train 450 weights, against the 50 of one-layer NN. The difference is of about one order of magnitude, leading to a much slower process (at fixed epochs). From simulations, the difference in training computational time is about 2 times. This factor is associated to the need of train two neural cores instead of one: one-shot operations provided by the crossbar let the speed of each single crossbar independent on its dimension.

Referring to the chip area, at the moment it is difficult to provide a benchmark, since nanowire network is not a mature technology yet and it is difficult to provide scaling information.

However, from an energy point of view, the use of two neural cores means to double the number of DAC/ADC, devices with a higher power consumption. Also, even if the speed of the device is independent on its dimension, it is not the

same for energy consumed, where both write and MVM energy should be considered. Moreover, to provide a reliable benchmark, energy consumption from reservoir has been analyzed.

It is important to stress that the aim is to provide an order of magnitude of consumed energy for the whole on-chip training. Thus, for simplicity, first approximations models and considerations have been adopted. The following analysis is devoted to estimate the energy consumption for on-chip training, neglecting the operational one since it is of minor significance.

**ADC/DAC energy** ADC and DAC converter, as mentioned, are high power-consumption devices. It has been demonstrated [80] the possibility to build a ADC working at  $f = 200 \text{ kHz}$  at  $0.85 \text{ fJ/level}$  (the level is equivalent to the conversion step). Considering, for example, 7 bit converters, each of them uses around  $6 \text{ fJ}$  for each conversion. For simplicity, this energy value is extended also to DAC.

**Write energy** Parallel write on a crossbar is guaranteed by an outer product operation, according to which:

$$w'_{ij} = w_{ij} + x_i y_j$$

where  $x_i$  are the row inputs and  $y_j$  the column ones. The row values are encoded on the voltage amplitude of pulses, while column values are mapped on different pulse duration. In this way the update follows the desired outer product to update cells conductance [74]. Moreover, current levels are generally fixed since memristive cells require a programmed amount of current to be driven.

According to this, the energy required to update (write operation) the cells of a  $N \times M$  crossbar is [79]:

$$E_{write} = N \cdot M \cdot I_{write} \cdot V_{write} \cdot \tau_{write}$$

with  $I_{write}$  the program current,  $V_{write}$  the operating voltage and  $\tau_{write}$  the pulse duration.

By adopting values from an high-performance resistive RAM [81], for which  $I_{write} = 0.1 \mu\text{A}$ ,  $V_{write} = 3 \text{ V}$  and  $\tau_{write} = 20 \text{ ns}$ , the energy required for each write step is:

$$E_{write} = 6 \cdot N \cdot M \text{ [fJ]}$$

Each outer product operation also includes an expense for converters, since there is an activation of  $(N + M)$  ADC.

**MVM energy** Besides write process, energy is consumed to perform MVM operation.

The way to perform it, always in a parallel fashion, is to charge the rows (according to input vector) and read the column lines, for which [79]:

$$E_{MVM} = N \cdot E_{charge,row} + M \cdot E_{charge,col}$$

The charge energy depends on the line capacitance, thus on the length of the line itself. Since the length of the line is also proportional to the number of cells along it, it holds:

$$E_{charge,row} = N \cdot C_{cell} V_{read}^2$$

$$E_{charge,col} = M \cdot C_{cell} V_{read}^2$$

with  $C_{cell}$  the capacitance of each cell of the crossbar.

By considering a cell capacitance of about  $50 \text{ aF}$  [82] and  $V_{read} = 0.2 \text{ V}$  [81]:

$$E_{MVM} = 2(N^2 + M^2) [aJ]$$

Each MVM operation also includes an expense for converters, since there is an activation of  $2(N + M)$  ADC.

**Reservoir energy** In order to estimate the energy consumption of the reservoir for each digit map, integral over time of the product of voltage and current has been performed exploiting previous simulations:

$$E_{reservoir} = \sum_{src} \int_0^{t_{max}} I_{src}(t) V_{src}(t) dt$$

The consumed energy to potentiate the reservoir will surely depend on the number of white cells in the written digit map. To provide a complete analysis, the energy behavior with respect to number of white cells is plotted in figure 4.8. As it can be seen, it shows approximately a linear increasing with white cells, which in the worst case it requires an energy consumption of  $0.65 \text{ J}$ .

This simulation relies on stimulation configuration discussed previously (figure 3.4), where the pulse duration is of  $t = 1, \text{ s}$ . For real devices, millisecond values for potentiation are reasonable. In this scenario, the reservoir energy for a single digit reduces to  $0.5 \text{ mJ}$  in the worst white pixels case.

**Benchmark** Overall, according to previous described backpropagation algorithm and energy estimations, the type and the number of operations to perform on the two scenarios are:

$$\text{Scenario 1 : } n_{data} \cdot R_{pot} + (n_{data} \cdot MVM + OP) \cdot epochs_1$$

$$\text{Scenario 2 : } [n_{data} \cdot (MVM_1 + 2MVM_2 + OP_1 + OP_2)] \cdot epochs_2$$

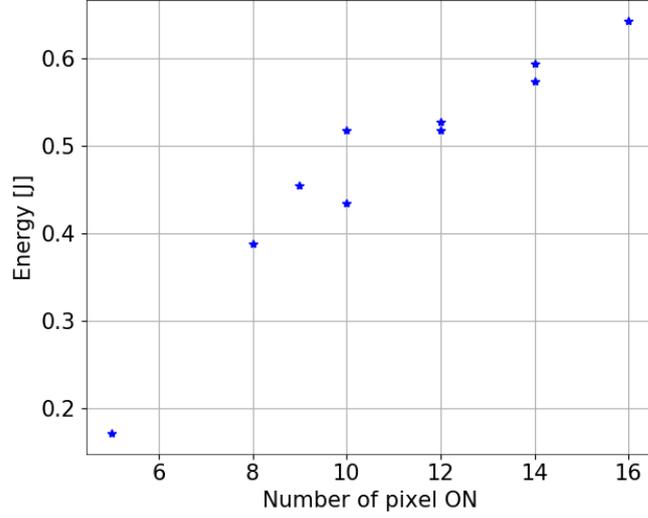


Figure 4.8: Reservoir energy consumption to be potentiated with respect to the number of white cells in digit map.

where  $R_{pot}$ ,  $OP$ ,  $n_{data}$  stay for reservoir potentiation, outer product and number of train data inputs, respectively. By computing the overall computational cost exploiting previous considerations, it emerges:

$$\text{Scenario 1 : } E_{train,1} = R_{pot} + 2.83 \text{ nJ} \sim 5 \text{ mJ}$$

$$\text{Scenario 2 : } E_{train,2} = 16.28 \text{ nJ}$$

According to this analysis, it emerges that the reservoir energy consumption is orders of magnitude higher than the fully crossbar solution.

However, at least two improvements on the reservoir can be performed to achieve lower energies:

- The stimulation time can be reduced accordingly to the idle time to guarantee a proper potentiation-relaxation pattern, but with a lower potentiation energy
- The nanowires conductivity can be decreased by properly engineering the core-shell structure of NWs or by properly reducing the nanowire density during deposition, in this way lower currents, thus energies, come into play

The energy optimization, thus, is a still opened task to perform on the device design. Moreover, as already mentioned, a single reservoir can be equipped with multiple readout functions to create a multi-task system, while NN solutions need to be customized for each problem.



## Chapter 5

# Experimental Reservoir Computing

Having demonstrated, in a simulation framework, the possibility to perform reservoir computing by exploiting Ag nanowire networks, experimental reservoir computing has been addressed and discussed in the following. It is here remarked that the presented results belong to an ongoing experimental work, where in this chapter the first relevant and promising results are highlighted, filling the gap between simulation analysis and experimental feasibility.

### 5.1 Experimental setup

Since the 5x4 written digit recognition, as it has been described, would require a complex experimental setup to measure all the relevant data, a simpler configuration has been considered.

The implemented process is schematized in figure 5.1, which is conceptually identical to the previously discussed in figure 3.2.

Five different patterns (figure 5.2) on 4x4 map have been selected for the recognition task. Then, input patterns are converted into voltage pulseshapes, as already discussed in section 3.2.1, and sent to their associated electrodes. In order to minimize the number of electrodes and reduce the complexity of the experimental measurement apparatus, a slightly different approach has been implemented with respect to simulations in chapter 3: only four electrodes, as depicted in figure 5.3, have been designed, which act either as source and reference nodes. In particular, the row pulses are sent respectively to labeled nodes  $N$ ,  $E$ ,  $S$ ,  $W$  (figure 5.1).

Since oscilloscope only reads voltage levels, the reading stage is possible by placing a resistance  $R$  between each electrode and ground, in order to measure a voltage as a consequence of a flowing current.

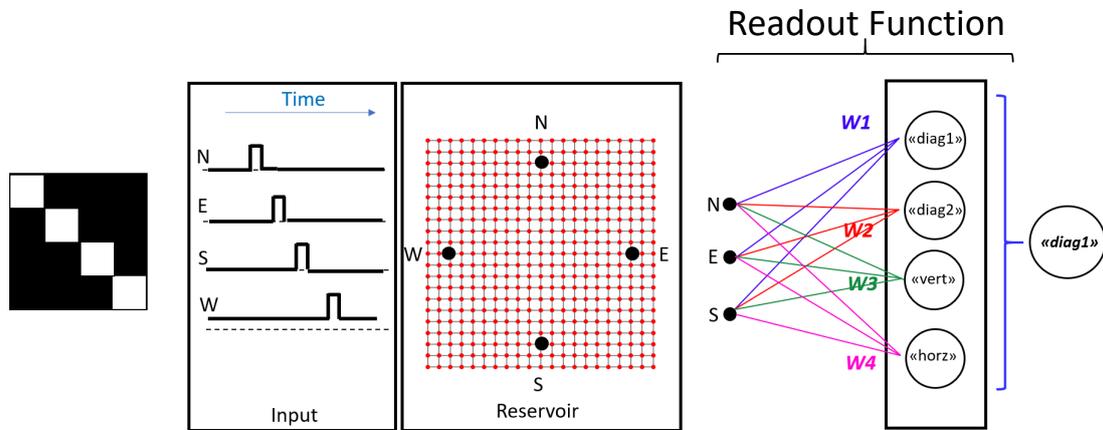


Figure 5.1: Experimental process for pattern recognition.

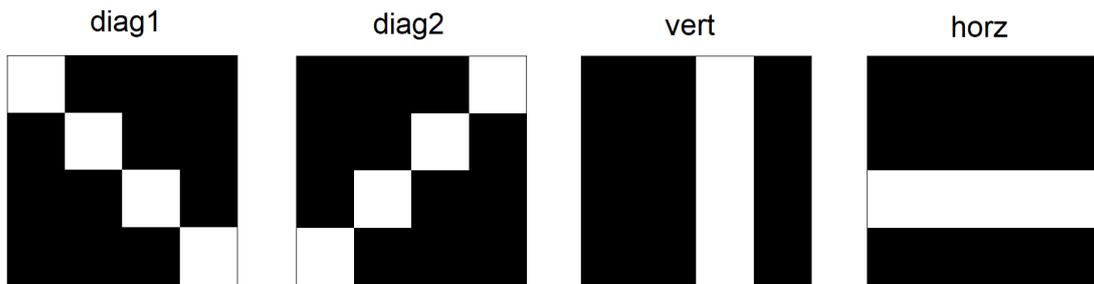


Figure 5.2: Patterns (4x4) for experimental reservoir computing classification.

It is important to remark that while electrodes labeled as  $N$ ,  $E$  and  $S$  receive an input which is *ground* or  $V_{max}$ , the electrode  $W$  always presents a small bias ( $V_{read}$ ) which allows the reading of voltages on the other three nodes. In this way, electrode  $W$  receives as input  $V_{read}$  or  $V_{max} + V_{read}$ . In this sense,  $N$ ,  $E$  and  $S$  electrodes are used either as source and reference: they behave as source electrodes during potentiation and as reference ones during reading.

After potentiated the network, the 3 voltage levels read in correspondence of nodes  $N$ ,  $E$  and  $S$  are used as the input for the readout function.

It is remarked that the information on output node  $W$  would not add any extra information since, according to Kirchoff current law, its voltage level turns out to be the negative sum of the other three output voltages.

The readout function is now a 3x4 neural network, having 3 inputs (also called neurons in the following) and 4 patterns to classify.

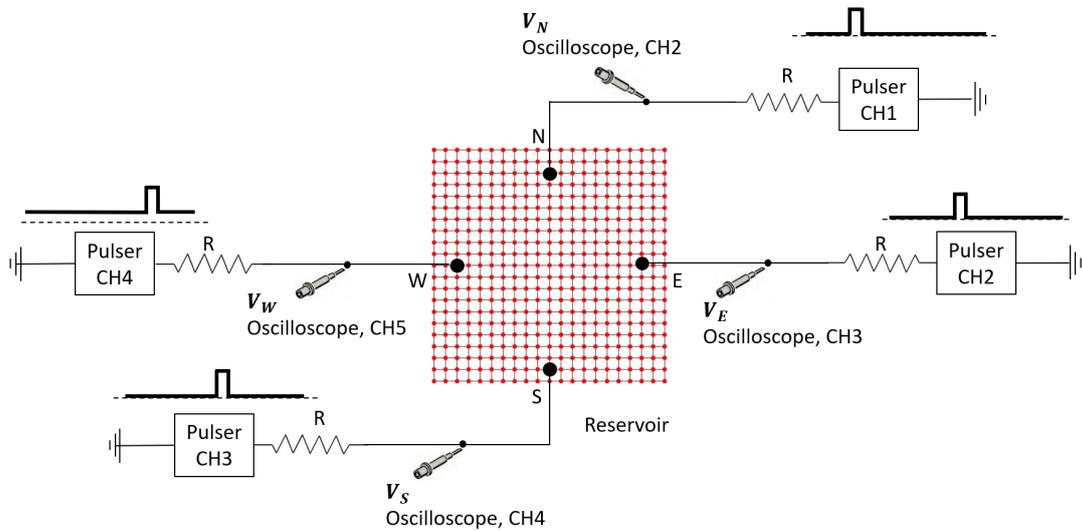


Figure 5.3: Experimental setup for pattern recognition. Example input pulses from *diag1* pattern.

## 5.2 Model simulation

Before experimental measurements, simulation investigation has been performed by exploiting the random diagonal model already discussed in section 2.3. Differently than before, however, 4 extra edges at fixed conductance have been designed in order to reproduce the 4 resistances  $R$  of the experimental setup (figure 5.3) to read voltage levels instead of conductances.

By keeping the same network parameters and pulse timing as in chapter 3, the evolution of the reservoir internal state is presented in figure 5.4 for each of the four patterns. As it can be seen, a clear separation of states can be obtained.

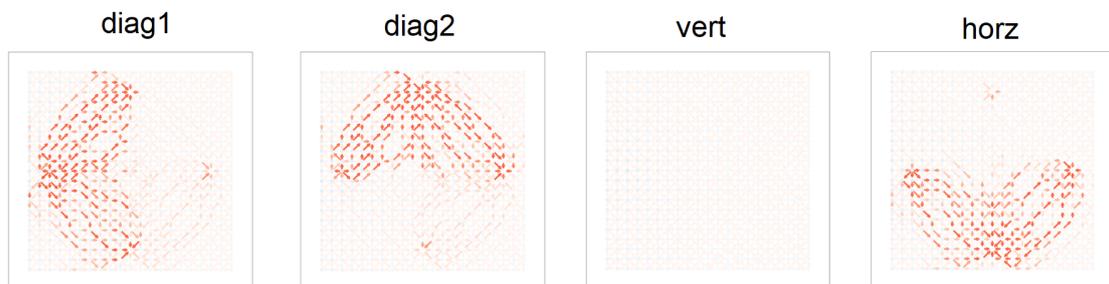


Figure 5.4: Simulated reservoir state after stimulation with patterns from figure 5.2.

It is worth noting that a direct consequence of the built setup is that the vertical pattern does not produce any potentiation: since all the electrodes are stimulated

at the same time, no voltage drop falls on the network. Actually,  $W$  is at a slightly higher voltage (due to  $V_{read}$ ) but it is too small to generate a perturbation. The reading of reservoir internal state, moreover, is depicted in figure 5.5.

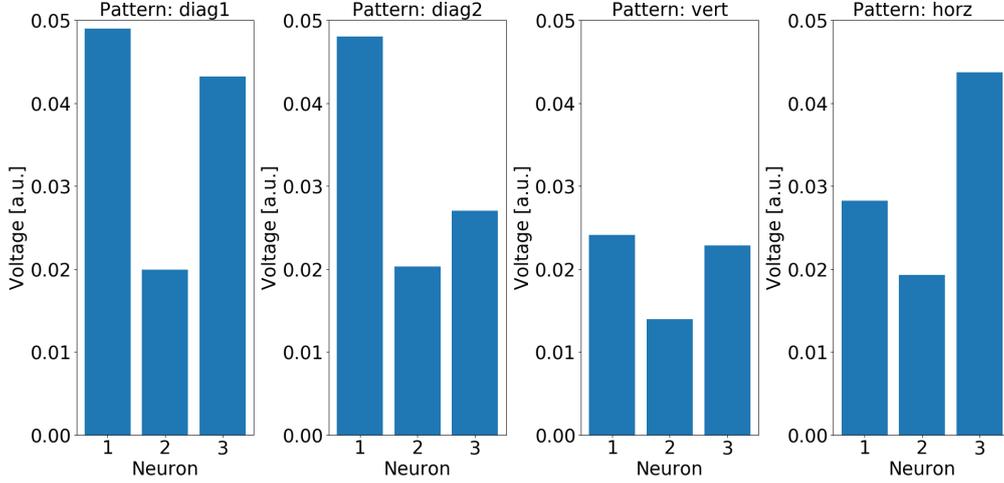


Figure 5.5: Reading of simulated reservoir state after stimulation with patterns from figure 5.2.

The reservoir state reading produces, again, clearly differentiated outputs. Voltages are expressed in arbitrary units in order to be free from the particular resistance value choice and from the network model parameters, highlighting the phenomenological states separability.

### 5.3 Experimental data analysis

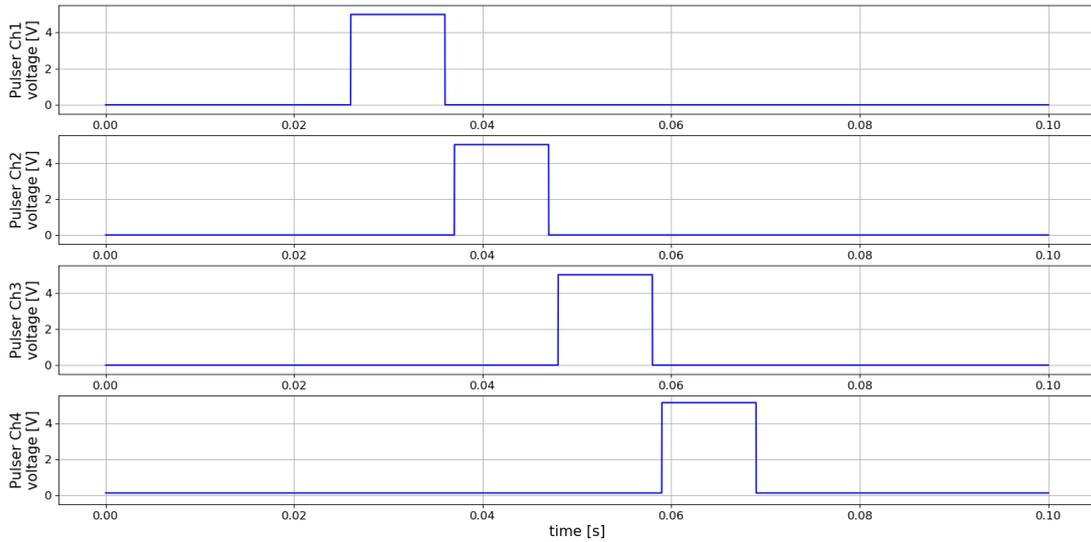
Experimental measurements have been performed by sending input voltages through 4 pulsers and reading voltage levels through 4 different oscilloscope channels (electrodes  $N$ ,  $E$ ,  $S$ ,  $W$  correspond to oscilloscope  $Ch2$ ,  $Ch3$ ,  $Ch4$ ,  $Ch5$ , respectively) as depicted in figure 5.3. The input pulseshape is characterized by a  $V_{max} = 5V$  of  $10ms$  duration and  $5ms$  between successive pulses, and a  $V_{read} = 125mV$ .

Moreover, the chosen resistance value is  $82\Omega$ , which is the result of a trade-off: it should be sufficiently high to guarantee a clear voltage reading value, but also adequately low to let the potentiation voltage to fall mostly on the network.

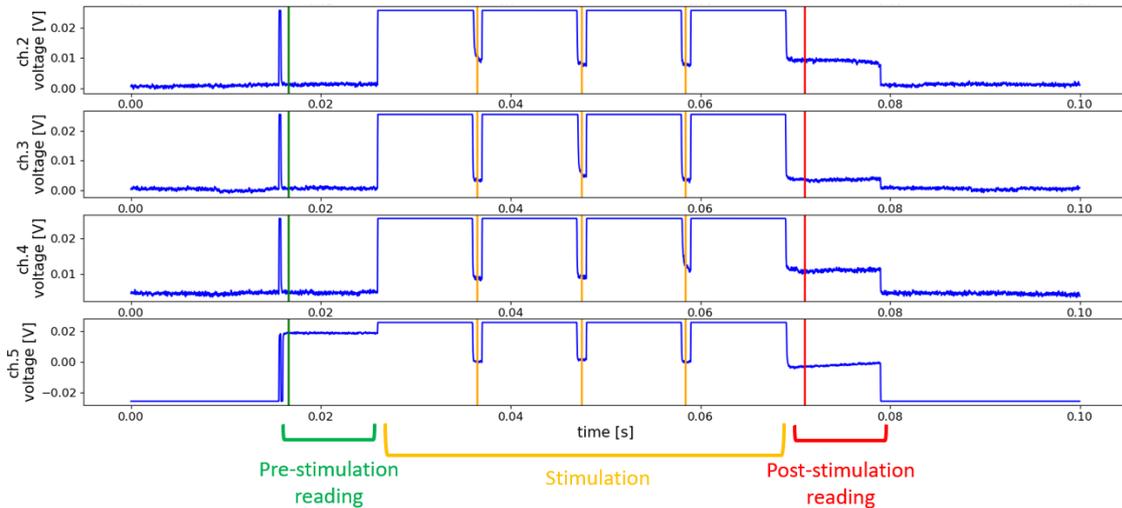
The applied methodology consists in:

- measuring the initial state of the network by biasing only the electrode  $W$
- stimulating the network according to the described voltage inputs
- reading the final state, again, by biasing only the electrode  $W$

A single pattern measurement example representing these three steps, as a response to pattern *diag1* whose input signals are depicted in figure 5.6a, is presented in figure 5.6b. In particular, the vertical lines are the chosen points for voltage reading. As it can be noticed, not only pre- and post-stimulation reading has been investigated, but also the one between each pulse, thanks to the mentioned constant reading bias on electrode  $W$ .



(a) Example of pulser channel inputs associated to *diag1* pattern.



(b) Example of oscilloscope channels reading during network stimulation with *diag1* pattern.

Figure 5.6: Pulse and read waveforms associated to *diag1* pattern.

In this experimental framework, each of the four patterns has been sent to the

network ten times, obtaining ten related oscilloscope measurements. It is remarked that, due to memory effect of the device, a sufficient waiting time (about 3 minutes) separates each measurement, in order to allow the device to spontaneously relax back to its pristine state.

By analyzing the time evolution of the three output voltages (on nodes N, E, S) from pre-stimulation region to post-stimulation one, it is possible to understand the changing of the reservoir state. This analysis is presented in figure 5.7, where each point is equipped with a certain error bar arising from the experimental variability of the ten multiple measurements for each pattern.

Remarkably, the four patterns produce different network evolution, as expected from simulations. In particular, looking at post-stimulation voltages, it is evident the different pattern and magnitude of the output voltage values. Moreover, since from an application point of view what is relevant is the final voltage read from the unbiased nodes (time instant  $t_4$  in figure 5.7) rather than any other combination of outputs in time, only the post-stimulation reading are considered for classification, containing to some extent all the network evolution history associated to a certain pattern. To better visualize the state separation and to confirm that the device starts from its pristine state, histograms related to the pre- and post- stimulation neurons values are reported in figure 5.8. As it can be seen in figure 5.8c, no potentiation characterizes the *Vert* pattern, as previously predicted from the model simulation. Moreover, the pristine state, besides experimental noise, turns out to be mostly the same for all measurements.

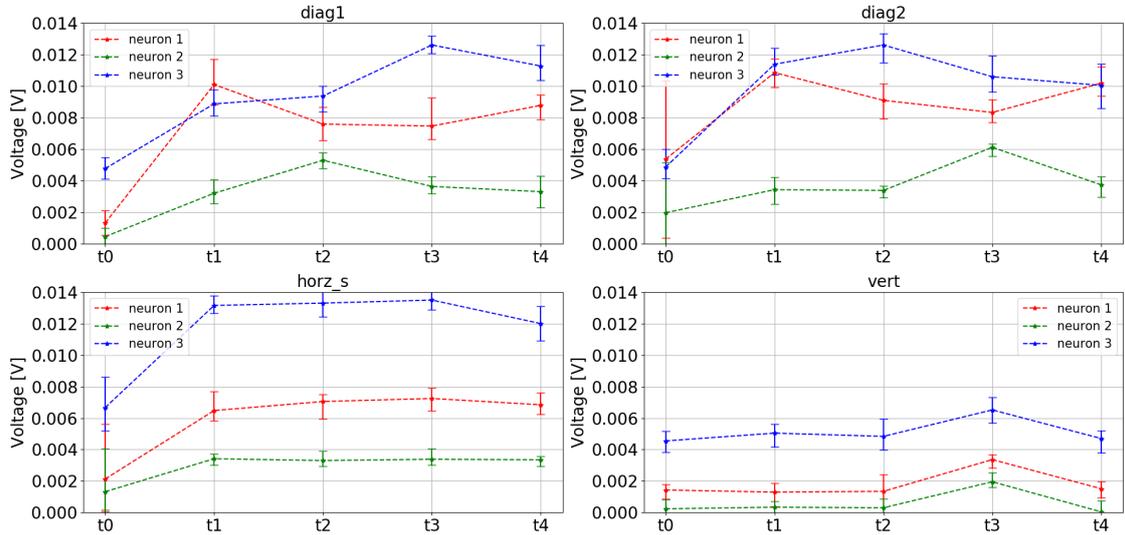
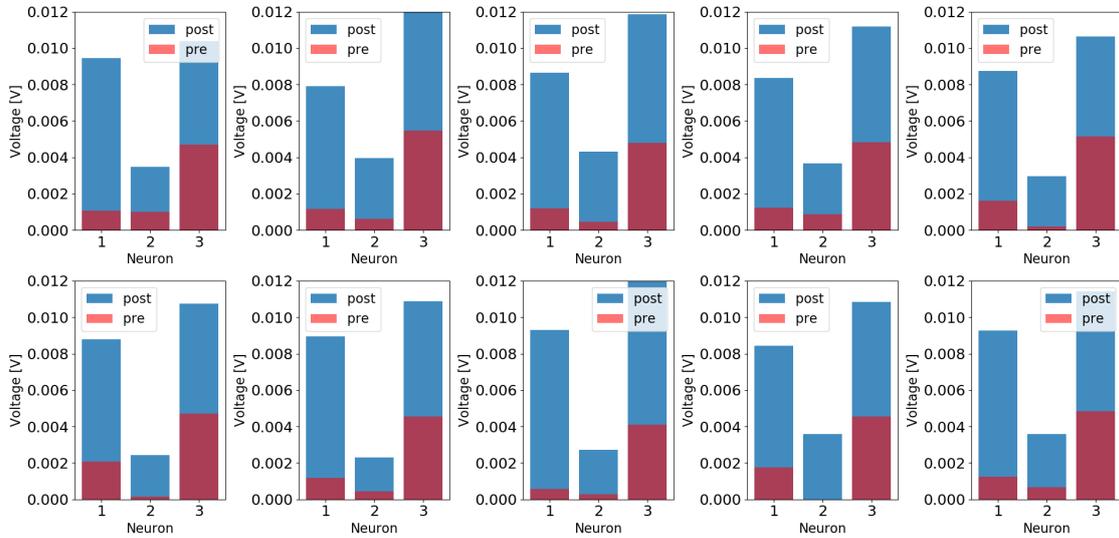


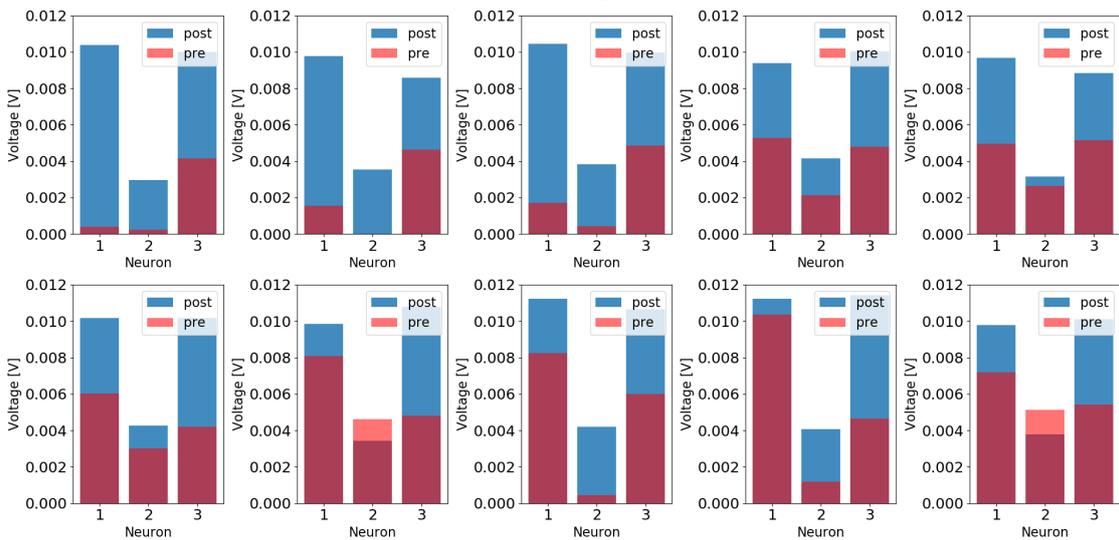
Figure 5.7: Output voltages (electrodes N, E, S labeled as neuron 1, 2 and 3 respectively) evolution for each pattern for pre-stimulation region ( $t_0$ ), stimulation region ( $t_1, t_2, t_3$ ) and post-stimulation region ( $t_4$ ).

### Diag1



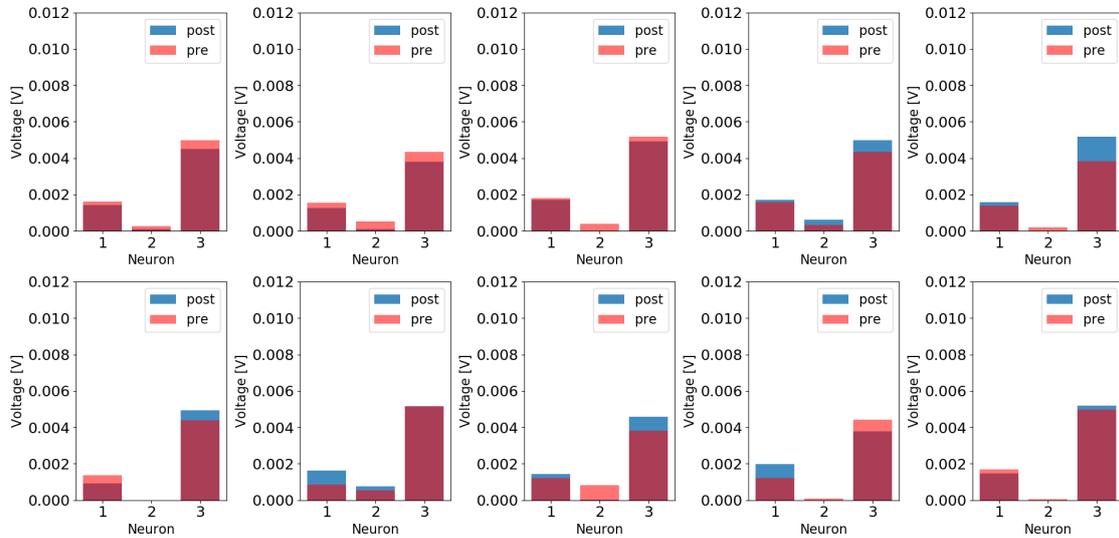
(a) Pre- and post- stimulus experimental histograms of output neurons voltages associated to *diag1* pattern.

### Diag2



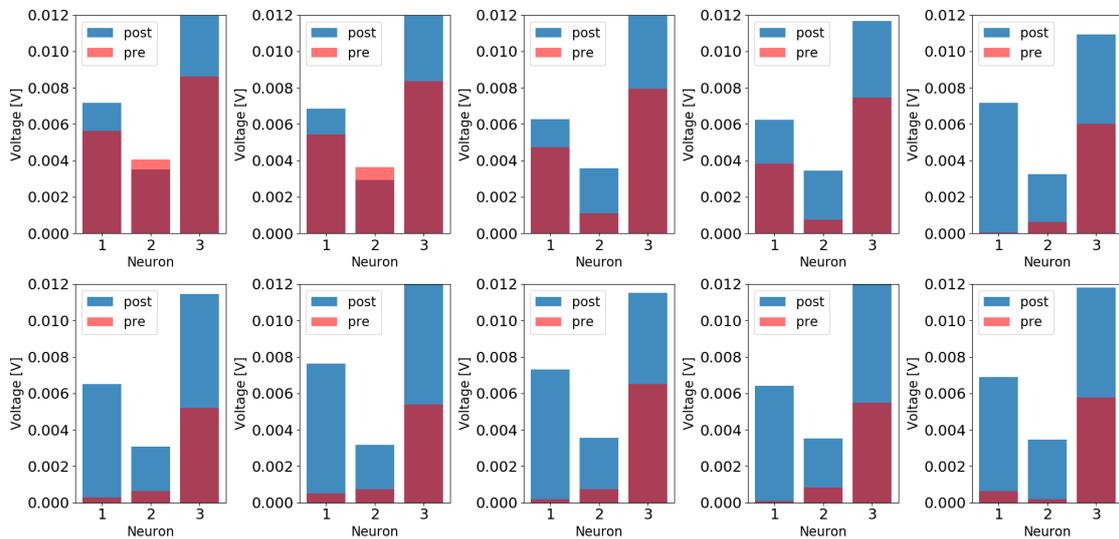
(b) Pre- and post- stimulus experimental histograms of output neurons voltages associated to *diag2* pattern.

### Vert



(c) Pre- and post- stimulus experimental histograms of output neurons voltages associated to *vert* pattern.

### Horz



(d) Pre- and post- stimulus experimental histograms of output neurons voltages associated to *horz* pattern.

Figure 5.8: Pre- and post- stimulation experimental histograms of output neurons voltages.

By taking the experimental post-stimulation output it is possible to train the readout function to understand the classification capability by considering these data.

The adopted methodology consists in:

- defining the dataset as a collection of 40 inputs (10 repetitions each pattern) over three values (output electrodes)
- random shuffling these data
- dividing the dataset in train and test ones, two third and one third of the whole data collection, respectively
- training a 3x4 neural network with the train dataset
- estimating accuracy performance on the test dataset

By considering a training over 1500 epochs (figure 5.9), the classification is able to reach 92.3% of accuracy.

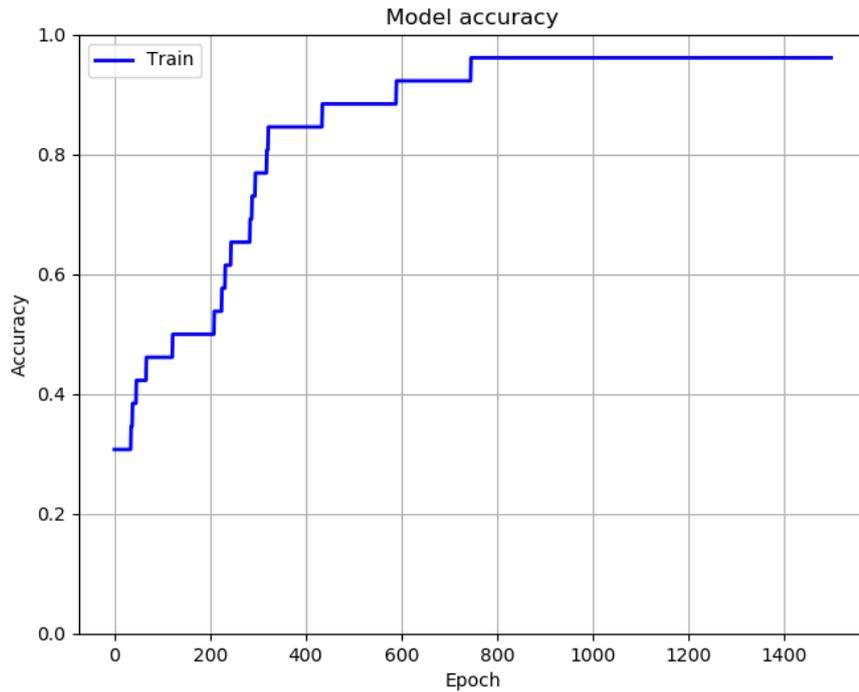


Figure 5.9: Training history of a 3x4 NN over 1500 epochs.

This results is an highly encouraging performance indicator, demonstrating that Ag nanowire networks exhibit good properties to implement reservoir computing.

## 5.4 Discussion

This brief analysis confirms the possibility to perform experimental reservoir computing on Ag nanowire networks, achieving high accuracy performance despite the experimental noise. Moreover, involving the same electrodes both as input and output nodes allows an extreme reduction on the total number of electrodes, generating a competitive advantage in the reduction of control electronics. While most of literature works [69; 70] demonstrating experimental reservoir computing on memristive devices deal with a collection of discrete memristors, the great step forward of these results relies on the adoption of a single device as a reservoir, interpreting this computing paradigm in its full principles. Even if the results belong to an early stage work activity, they behave as a proof of concept which is fundamental for further investigations. In this scenario, the modeling has provided interesting insights to predict and optimize experimental measurements. A step forward may be the fitting of this new experimental device curves to extrapolate the model parameters. In this way, more complex classification can be investigated by the model first and experimentally second.

## Chapter 6

# Conclusions and future perspectives

This work has demonstrated the possibility to develop a simple compact model by exploiting a balanced-rate equation and a grid structure, to describe the relevant experimental properties of self-organizing memristive nanowire networks. In particular the homo- and hetero-synaptic plasticity has been shown to be correctly reproduced from the model, along with paired-pulse facilitation and short-term plasticity. Moreover, spatial and morphological information on conductive path formation and spontaneous relaxation has been possible thanks to the implemented grid structure to map synaptic connections of the device. In particular, a good link between model and experimental data has been provided through fitted model parameters.

The adoption of a compact model, resulting in low computational cost, has made possible the investigation of reservoir computing (RC) process, exploiting the nanowire network as a reservoir and a one-layer neural network as a readout function.

By considering 5x4 written digit maps, accuracy of 90% can be reached (in simulation) on a single-bit noisy dataset and 70% for two-bits noisy dataset. These results demonstrate high potentiality for this system to be exploited as a physical reservoir. In the direction of energy-efficiency and high operational frequency, a fully-memristive system has been demonstrated by implementing the NN readout on a memristive crossbar array. The latter has been involved in *on-chip* training simulations, showing good accuracies as software implementation even considering crossbar read and write operational noise. Proposing a simplified analysis, a reservoir equipped with a one-layer readout on crossbar array turns out to be more energy expensive with respect to a fully memristive solution implemented on a two-layer NN (on two crossbar arrays).

However, while the crossbar array is almost saturated in terms of improvements, the presented nanowire network is still opened to optimizations. Moreover, a single

reservoir may be equipped, in principle, with a high number of readout functions to create a general-purpose chip to solve problems of different nature. This is, in perspective, one of the highest competitive advantage for energy efficiency with respect to fully-crossbar solutions, which should be differentiated on the base of the particular problem to solve.

Finally, experimental reservoir computing has been demonstrated with great success: accuracy of 92.3% has been reached in the recognition of 4 different patterns mapped on a 4x4 matrix. This results confirm the initial simulation investigation, providing a proof of concept for more complex experimental measurements.

Future perspectives concern the optimization of electrodes configuration, input processing and reservoir design to deal with on larger dimension problems, such as MNIST dataset recognition.

Further developments may refer to the identification of different datasets which may be better classified by this device and this computing paradigm. Examples are speech and human activity recognition tasks, where the relatively slow dynamics is not a constraint for applications.

The idea is to identify a set of operations easily performed by the network, exploiting them to define a proper input managing to achieve a high state separation.

Moreover, a great investigation needs to be done on the reservoir properties as mentioned: electrode configuration, density of the nanowires, topology of the deposited 2D structure are some of the possible improvements.

A limit is linked to the low connectivity of 2D deposited nanowire with respect to the 3D brain high connectivity.

Even if still in its early stage, these devices involved in reservoir computing approach demonstrate really high potentialities due their strong similarity to human brain both in physical structure and its learning processes.

# Bibliography

- [1] Chua, L. Memristor-The missing circuit element. *IEEE Trans. Circuit Theory* 18, 507–519. issn: 0018-9324 (1971).
- [2] Chua, L. & Sung Mo Kang. Memristive devices and systems. *Proc. IEEE* 64, 209–223. issn: 0018-9219 (1976).
- [3] Yesil, Abdullah & Gül, Fatih & Babacan, Yunus. (2018). Emulator Circuits and Resistive Switching Parameters of Memristor. 10.5772/intechopen.71903.
- [4] Vongehr, S. & Meng, X. The Missing Memristor has Not been Found. *Sci. Rep.* 5, 11657. issn: 2045-2322 (Dec. 2015)
- [5] Abraham, I. The case for rejecting the memristor as a fundamental circuit element. *Sci. Rep.* 8, 10972. issn: 2045-2322 (Dec. 2018).
- [6] Strukov, D. B., Snider, G. S., Stewart, D. R. & Williams, R. S. The missing memristor found. *Nature* 453, 80–83. issn: 0028-0836 (May 2008).
- [7] Waser, R. & Aono, M. Nanoionics-based resistive switching memories. *Nat. Mater.* 6, 833–840. issn: 1476-1122 (Nov. 2007).
- [8] Chua, L. Resistance switching memories are memristors. *Appl. Phys. A* 102, 765–783. issn: 0947-8396 (Mar. 2011).
- [9] Waser, R. in *Memristive Phenom. - From Fundam. Phys. to Neuromorphic Comput.* 47th IFF Spring Sch. 2016 - Lect. Notes 15–37 (2016). isbn: 978-3-95806-091-3.
- [10] Waser R. *Nanoelectronics and information technology: advanced electronic materials and novel devices*, 3rd ed. Weinheim, Germany, Wiley, 2012.
- [11] Valov I., Tsuruoka T., Effects of moisture and redox reactions in VCM and ECM resistive switching memories, *Journal of Physics D: Applied Physics*, Vol. 51, n. 41 (2018)

- [12] Zhang, Xumeng & Liu, Sen & Zhao, Xiaolong & Wu, Facai & Wu, Quantan & Wang, Wei & Cao, Rongrong & Fang, Yilin & Lv, Hangbing & Long, Shibing & Liu, Qi & Liu, Ming. (2017). Emulating Short-term and Long-term Plasticity of Bio-synapse based on Cu/a-Si/Pt Memristor. *IEEE Electron Device Letters*. PP. 1-1. 10.1109/LED.2017.2722463.
- [13] L. F. Abbott and W. G. Regehr, "Synaptic computation," *Nature*, vol. 431, pp. 796–803, Oct. 2004, doi: 10.1038/nature03010
- [14] Rosenbaum R., Rubin J., Doiron B. Short term synaptic depression imposes a frequency dependent filter on synaptic information transfer *PLoS Computational Biology*, 8 (6) (2012), p. 18
- [15] Maass W., Markram H. Synapses as dynamic memory buffers *Neural Networks*, 15 (2) (2002), pp. 155-161
- [16] Leng L., Martel R., Breitwieser O., Bytschok I., Senn W., Schemmel J., et al. Spiking neurons with short-term synaptic plasticity form superior generative networks *Scientific Reports*, 8 (1) (2018), Article 10651
- [17] Zeng, Guanxiong & Huang, Xuhui & Jiang, Tianzi & Yu, Shan. (2019). Short-term synaptic plasticity expands the operational range of long-term synaptic changes in neural networks. *Neural Networks*. 118. 10.1016/j.neunet.2019.06.002.
- [18] Lanza, Mario & Wong, H.-S. Philip & Pop, Eric & Ielmini, D. & Strukov, Dimitri & Regan, Brian & Larcher, L. & Villena, Marco & Yang, Jianhua Joshua & Goux, L. & Belmonte, Attilio & Yang, Yuchao & Puglisi, Francesco & Kang, Jinfeng & Magyari-Kope, Blanka & Yalon, Eilam & Kenyon, Anthony & Buckwell, Mark & Mehonic, Adnan & Shi, Yuanyuan. (2018). Recommended Methods to Study Resistive Switching Devices. *Advanced Electronic Materials*. 5. 1800143. 10.1002/aelm.201800143.
- [19] Wang Q, Sun HJ, Zhang JJ, Xu XH, And Miao XS. Electrode materials for Ge<sub>2</sub>Sb<sub>2</sub>Te<sub>5</sub>-based memristors. *Journal of Electronic Materials*. 2012;41(12):3417-3422. DOI: 10.1007/s11664-012-2256-6
- [20] Prodromakis T, Peh BP, Papavassiliou C, Toumazou C. A versatile memristor model with non-linear dopant kinetics. *IEEE Transactions on Electron Devices*. 2011;58(9):3099-3105. DOI: 10.1109/TED.2011.2158004
- [21] Yang JJ, Pickett MD, Li X, Ohlberg DAA, Stewart DR, Williams RS. Memristive switching mechanism for metal/oxide/metal nanodevices. *Nature Nanotechnology*. 2008;3(7):429-433. DOI: 10.1038/nnano.2008.160
- [22] S. Wu, K. Michael Wong, M. Tsodyks, Eds., Neural information processing with dynamical synapses, *Frontiers in Computational Neuroscience*, vol 7, 2013. DOI: 10.3389/978-2-88919-383-7

- [23] Joglekar YN, Wolf SJ. The elusive memristor: Properties of basic electrical circuits. *European Journal of Physics*. 2009;30(4):661-675. DOI: 10.1088/0143-0807/30/4/001
- [24] Biolek Z, Biolek D, Biolkova V. SPICE model of memristor with nonlinear dopant drift. *Radioengineering*. 2009;18(2):210-214
- [25] Prodromakis T, Peh BP, Papavassiliou C, Toumazou C. A versatile memristor model with non-linear dopant kinetics. *IEEE Transactions on Electron Devices*. 2011;58(9):3099-3105. DOI: 10.1109/TED.2011.2158004
- [26] Zha J, Huang H, Liu Y. A novel window function for memristor model with application in programming analog circuits. *IEEE Transactions On Circuits and Systems—II: Express Briefs*. 2016;63(5):423-427. DOI:0.1109/TCSII.2015.2505959
- [27] Miranda, Enrique & Milano, Gianluca & Ricciardi, Carlo. (2020). Modeling of Short-Term Synaptic Plasticity Effects in ZnO Nanowire-Based Memristors Using a Potentiation-Depression Rate Balance Equation. *IEEE Transactions on Nanotechnology*. PP. 1-1. 10.1109/TNANO.2020.3009734.
- [28] S. Menzel, S. Tappertzhofen, R. Waser, I. Valov, “Switching kinetics of electrochemical metallization memory cells,” *Phys Chem Phys* 15, 6945 (2013). DOI: 10.1039/c3cp50738f
- [29] Z. Wang, A. Joshi, S. Savelev, H. Jiang, R. Midya, P. Lin, M. Hu, N. Ge, J. Strachan, Z. Li, Q. Wu, M. Barnell, G. Li, H. Xin, R. Stanley Williams, Q. Xia,, J Yang, “Memristors with diffusive dynamics as synaptic emulators for neuromorphic computing,” *Nat Mat* 16, pp. 101-108 (2017). DOI: 10.1038/nmat4756
- [30] A. Rodriguez-Fernandez, C. Cagli, J. Suñé, E. Miranda, “Switching voltage and time statistics of filamentary conductive paths in HfO<sub>2</sub>-based ReRAM devices,” *IEEE Electron Dev Lett* 39, 656-659, 2018. DOI: 10.1109/LED.2018.2822047
- [31] Oliver, Sean & Fairfield, Jessamyn & Bellew, Allen & Lee, Sunghun & Champlain, James & Ruppalt, Laura & Boland, John & Vora, Patrick. (2016). Quantum point contacts and resistive switching in Ni/NiO nanowire junctions. *Applied Physics Letters*. 109. 10.1063/1.4967502.
- [32] Mikheev, E., Hoskins, B. D., Strukov, D. B. & Stemmer, S. Resistive switching and its suppression in Pt/Nb:SrTiO<sub>3</sub> junctions. *Nat. Commun.* 5, 3990 (2014).
- [33] Choi, Sanghyeon & Jang, Seonghoon & Moon, Jung Hwan & Kim, Jong & Jeong, Hu Young & Jang, Peonghwa & Lee, Kyung-Jin & Wang, Gunuk. (2018). A self-rectifying TaO<sub>y</sub>/nanoporous TaO<sub>x</sub> memristor synaptic array for learning and energy-efficient neuromorphic systems. *NPG Asia Materials*. 10. 10.1038/s41427-018-0101-y.

- [34] Hu, M., Strachan, J. P., Li, Z. & Williams, S. R. Dot-product engine as computing memory to accelerate machine learning algorithms. In 17th Int. Symp. Quality Electronic Design (ISQED) 374–379 (IEEE, 2016).
- [35] Yao, P. et al. Face classification using electronic synapses. *Nat. Commun.* 8, 15199 (2017).
- [36] Sheridan, P. M. et al. Sparse coding with memristor networks. *Nat. Nanotechnol.* 12, 784–789 (2017).
- [37] Xia, Qiangfei & Yang, Jianhua Joshua. (2019). Memristive crossbar arrays for brain-inspired computing. *Nature Materials*. 18. 309-323. 10.1038/s41563-019-0291-x.
- [38] Ielmini, D. & Wong, H.-S. Philip. (2018). In-memory computing with resistive switching devices. *Nature Electronics*. 1. 10.1038/s41928-018-0092-2.
- [39] Sun, Zhong & Pedretti, Giacomo & Bricalli, Alessandro & Ielmini, D.. (2020). One-step regression and classification with crosspoint resistive memory arrays.
- [40] Milano, Gianluca & Pedretti, Giacomo & Fretto, M. & Boarino, Luca & Benfenati, Fabio & Ielmini, D. & Valov, Ilia & Ricciardi, Carlo. (2019). Self-organizing memristive nanowire networks with structural plasticity emulate biological neuronal circuits.
- [41] Diaz-Alvarez, Adrian & Higuchi, Rintaro & Sanz-Leon, Paula & Marcus, Ido & Shingaya, Yoshitaka & Stieg, Adam & Gimzewski, James & Kuncic, Zdenka & Nakayama, Tomonobu. (2019). Emergent dynamics of neuromorphic nanowire networks. *Scientific Reports*. 9. 10.1038/s41598-019-51330-6.
- [42] Pike, Matthew & Bose, Saurabh & Mallinson, Joshua & Acharya, Susant & Shirai, Shota & Galli, Edoardo & Weddell, Stephen & Bones, P.J. & Arnold, Matthew & Brown, Simon. (2020). Atomic Scale Dynamics Drive Brain-like Avalanches in Percolating Nanostructured Networks. *Nano Letters*. XXXX. 10.1021/acs.nanolett.0c01096.
- [43] Gomes da Rocha, Claudia & Ferreira, Mauro & Boland, John & Manning, Hugh & Biswas, Subhjit. (2018). Emergence of winner-takes-all connectivity paths in random nanowire networks. *Nature Communications*. 9. 3219. 10.1038/s41467-018-05517-6.
- [44] Chistiakova, Marina & Bannon, Nicholas & Bazhenov, Maxim & Volgushev, Maxim. (2014). Heterosynaptic Plasticity: Multiple Mechanisms and Multiple Roles. *The Neuroscientist : a review journal bringing neurobiology, neurology and psychiatry*. 20. 10.1177/1073858414529829.

- [45] Park, J., Lee, S. & Yong, K. Photo-stimulated resistive switching of ZnO nanorods. *Nanotechnology* 23, 385707. issn: 0957-4484 (Sept. 2012).
- [46] Younis, A. et al. High-Performance Nanocomposite Based Memristor with Controlled Quantum Dots as Charge Traps. *ACS Appl. Mater. Interfaces* 5, 2249–2254. issn: 1944-8244 (Mar. 2013).
- [47] Porro, S. et al. Multiple resistive switching in core–shell ZnO nanowires exhibiting tunable surface states. *J. Mater. Chem. C* 5, 10517–10523. issn: 2050-7526 (2017).
- [48] Russo, P., Xiao, M., Liang, R. & Zhou, N. Y. UV-Induced Multilevel Current Amplification Memory Effect in Zinc Oxide Rods Resistive Switching Devices. *Adv. Funct. Mater.* 28, 1706230. issn: 1616301X (Mar. 2018).
- [49] Wang, Wei & Wang, Ming & Ambrosi, Elia & Bricalli, Alessandro & Laudato, Mario & Sun, Zhong & Chen, Xiaodong & Ielmini, Daniele. (2019). Surface diffusion-limited lifetime of silver and copper nanofilaments in resistive switching devices. *Nature Communications*. 10. 81. 10.1038/s41467-018-07979-0.
- [50] Milano, G. et al. Self-limited single nanowire systems combining all-in-one memristive and neuromorphic functionalities. *Nat. Commun.* 9, 5151 (2018).
- [51] Wang, Wei & Covi, Erika & Lin, Yu-Hsuan & Ambrosi, Elia & Ielmini, D.. (2019). Modeling of switching speed and retention time in volatile resistive switching memory by ionic drift and diffusion. 32.3.1-32.3.4. 10.1109/IEDM19573.2019.8993625.
- [52] Tanaka, Gouhei & Yamane, Toshiyuki & Heroux, Jean & Nakane, Ryosho & Kanazawa, Naoki & Takeda, Seiji & Numata, Hidetoshi & Nakano, Daiju & Hirose, Akira. (2018). Recent Advances in Physical Reservoir Computing: A Review.
- [53] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). Learning internal representations by error propagation. Technical Report California Univ. San Diego La Jolla Inst. for Cognitive Science.
- [54] Werbos, P. J. (1990). Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78, 1550–1560.
- [55] Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1, 270–280.
- [56] Doya, K. (1998). Recurrent networks: supervised learning. In *The handbook of brain theory and neural networks* (pp. 796–800). MIT Press.

- [57] Jaeger, H. (2001). The “echo state” approach to analysing and training recurrent neural networks. Technical Report GMD Report 148, German National Research Center for Information Technology.
- [58] Maass, W., Natschläger, T., & Markram, H. (2002). Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14, 2531–2560.
- [59] Verstraeten, D., Schrauwen, B., Stroobandt, D., & Van Campenhout, J. (2005b). Isolated word recognition with the liquid state machine: a case study. *Information Processing Letters*, 95, 521–528.
- [60] Soh, H., & Demiris, Y. (2012). Iterative temporal learning and prediction with the sparse online echo state Gaussian process. In *Proc. International Joint Conference on Neural Networks (IJCNN)* (pp. 1–8). IEEE
- [61] Jalalvand, A., Van Wallendael, G., & Van de Walle, R. (2015). Real-time reservoir computing network-based systems for detection tasks on visual contents. In *Proc. 7th International Conference on Computational Intelligence, Communication Systems and Networks (CICSyN)* (pp. 146–151). IEEE.
- [62] Paquot, Y., Duport, F., Smerieri, A., Dambre, J., Schrauwen, B., Haelterman, M., & Massar, S. (2012). Optoelectronic reservoir computing. *Scientific Reports*, 2, 287.
- [63] Jaeger, H. (2002b). Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “echo state network” approach. Technical Report GMD Report 159, German National Research Center for Information Technology.
- [64] Hauser, H., Ijspeert, A. J., Füchslin, R. M., Pfeifer, R., & Maass, W. (2011). Towards a theoretical foundation for morphological computation with compliant bodies. *Biological Cybernetics*, 105, 355–370.
- [65] Appeltant, L., Soriano, M. C., Van der Sande, G., Danckaert, J., Massar, S., Dambre, J., Schrauwen, B., Mirasso, C. R., & Fischer, I. (2011). Information processing using a single dynamical node as complex system. *Nature Communications*, 2, 468.
- [66] Vandoorne, K., Dierckx, W., Schrauwen, B., Verstraeten, D., Baets, R., Bienstman, P., & Campenhout, J. V. (2008). Toward optical signal processing using photonic reservoir computing. *Optics Express*, 16, 11182–11192.
- [67] Torrejon, J., Riou, M., Araujo, F. A., Tsunegi, S., Khalsa, G., Querlioz, D., Bortolotti, P., Cros, V., Yakushiji, K., Fukushima, A. et al. (2017). Neuromorphic computing with nanoscale spintronic oscillators. *Nature*, 547, 428.

- [68] Buonomano, D. V., & Maass, W. (2009). State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10, 113–125.
- [69] Du, Chao & Cai, Fuxi & Zidan, Mohammed & Ma, Wen & Lee, Seung Hwan & Lu, Wei. (2017). Reservoir computing using dynamic memristors for temporal information processing. *Nature Communications*. 8. 10.1038/s41467-017-02337-y.
- [70] Zhong, Yanan & Tang, Jianshi & Li, Xinyi & Gao, Bin & Qian, He & Wu, Huaqiang. (2020). Dynamic Memristor-based Reservoir Computing for High-Efficiency Spatiotemporal Signal Processing. 10.21203/rs.3.rs-40717/v1.
- [71] Schrauwen, Benjamin & Verstraeten, David & Campenhout, Jan. (2007). An overview of reservoir computing: Theory, applications and implementations. *Proceedings of the 15th European Symposium on Artificial Neural Networks*. 471-482.
- [72] Sun, Zhong & Pedretti, Giacomo & Bricalli, Alessandro & Ielmini, D.. (2020). One-step regression and classification with crosspoint resistive memory arrays.
- [73] S. Agarwal, T.-T. Quach, O. Parekh, et al., "Energy Scaling Advantages of Resistive Memory Crossbar Based Computation and its Application to Sparse Coding," *Frontiers in Neuroscience*, vol. 9, 2016.
- [74] Agarwal, Sapan & Plimpton, Steven & Hughart, David & Hsia, Alexander & Richter, Isaac & Cox, Jonathan & James, Conrad & Marinella, Matthew. (2016). Resistive memory device requirements for a neural algorithm accelerator. 929-938. 10.1109/IJCNN.2016.7727298.
- [75] Hasan, Raqibul & Taha, T.M.. (2014). Enabling back propagation training of memristor crossbar neuromorphic processors. *Proceedings of the International Joint Conference on Neural Networks*. 21-28. 10.1109/IJCNN.2014.6889893.
- [76] Fuller, Elliot & Gabaly, Farid & Léonard, François & Agarwal, Sapan & Plimpton, Steven & Jacobs-Gedrim, Robin & James, Conrad & Marinella, Matthew & Talin, A.. (2016). Li-Ion Synaptic Transistor for Low Power Analog Computing. *Advanced Materials*. 29. 10.1002/adma.201604310.
- [77] Van de Burgt, Yoeri & Lubberman, Ewout & Fuller, Elliot & Keene, Scott & Faria, Gregório & Agarwal, Sapan & Marinella, Matthew & Talin, A. & Salleo, Alberto. (2017). A non-volatile organic electrochemical device as a low-voltage artificial synapse for neuromorphic computing. *Nature Materials*. 16. 10.1038/NMAT4856.
- [78] P.-Y. Chen, B. Lin, I.-T. Wang, et al., "Mitigating Effects of Non-ideal Synaptic Device Characteristics for On-chip Learning," presented at the Proceedings of

- the IEEE/ACM International Conference on Computer-Aided Design, Austin, TX, USA, 2015.
- [79] Agarwal, Sapan & Quach, Tu-Thach & Parekh, Ojas & Hsia, Alexander & DeBenedictis, Erik & James, Conrad & Marinella, Matthew & Aimone, James. (2016). Energy Scaling Advantages of Resistive Memory Crossbar Based Computation and Its Application to Sparse Coding. *Frontiers in Neuroscience*. 9. 10.3389/fnins.2015.00484.
- [80] Tai, H. Y., Hu, Y.-S., Chen, H.-W., and Chen, H.-S. (2014). "A 0.85fJ/conversion-step 10b 200kS/s subranging SAR ADC in 40nm CMOS," *IEEE International Solid-State Circuits Conference (San Francisco, CA)*, 196–197.
- [81] Chun-Hu, Cheng & Tsai, Chia & Chin, Albert & Yeh, F.S.. (2011). High Performance Ultra-Low Energy RRAM with Good Retention and Endurance. *IEDM Tech. Dig.*. 19.4.1 - 19.4.4. 10.1109/IEDM.2010.5703392.
- [82] International Technology Roadmap for Semiconductors (ITRS) (2013).