POLITECNICO DI TORINO

Master Degree in Nanotechnologies for ICTs

Master Thesis

Reservoir Computing in memristive nanowire networks

Politecnico
di Torino

Supervisor:
Prof. Carlo Ricciardi

Co-Supervisor:
Dr. Gianluca Milano

Candidate:
Matteo Agliuzza

October 2021

Abstract

In order to overcome the limitations given by transistor-based systems working in the Von Neumann architecture, it is necessary to develop new technologies and computing paradigms. In this framework, brain-inspired structures allow to perform spatio-temporal correlated operations, typical of neural circuits: in particular, the Memristor is one of the new analog devices which lets this approach possible. Defined as the fourth circuit element, the memristor is a passive element which exhibits non-linearities in its dynamics, thanks to the change in the internal resistance state due to the rearrangement of the atomic structure. Here, the computational capabilities of self-organized memristive nanowire (NW) networks are investigated by simulations: thanks to its resistive switching and fading memory properties, the system is capable of mimicking human brain's synapses basic functions in processing external signals. In particular, the complex nonlinear dynamics of the network allows it to be exploited as a Reservoir, in the so-called Reservoir Computing (RC). This approach can offer efficient temporal processing of recurrent neural networks with a very low training cost, which makes it perfect for temporal data classification. In particular, in Reservoir Computing, a dynamic system (reservoir) which offers short-term memory is capable of mapping non-linearly an input in a higher dimensional space to emphasize and extrapolate spatio-temporal input correlations, in order to simplify the data classification. The RC architecture developed in this work is composed by three main building blocks. Firstly, a pre-process is required in order to transform the data in electrical signals. The stimuli are then applied to the NW network, which evolves in conductive paths thanks to the memristive processes that happen at the cross-point junctions between the nanowires. Lastly, the reservoir states are collected and used as input for a supervised learning algorithm, implemented in a readout function (one-layer neural network, regression model). In order to reduce the physical dimensions of the device without affecting the overall performance, the concept of virtual nodes has been exploited as well: in particular, the data processed in the n-th virtual node is affected by the previous virtual node states, meaning that it carries out information of the near history of the dynamical reservoir. Thanks to this model, it is possible to perform temporal data analysis and solve complex tasks, such as speech recognition: in particular, the task requires the recognition of spoken digits in the form of audio samples. In order to maintain a bio-inspired structure, the audio signals are pre-processed with the Lyon's Auditory Model, which allows to transform sound waveforms in voltage spike trains, by simulating the human's auditory system. At last, the model is optimized by considering different degrees of freedom, such as electrodes configuration, number of virtual nodes, input pre-processing, and readout function parameters.

Acknowledgements

First and foremost, I would like to express my sincere gratitude towards my supervisor, Professor Carlo Ricciardi. His motivation, enthusiasm and deep knowledge has been a source of inspiration, and I am strongly grateful for all the help and support that he has given me within and outside the thesis.

A deep thank goes to my co-supervisor Dr. Gianluca Milano, for guiding me throughout the thesis work. Your dedication and professional rigour has been overwhelming, and a great occasion for my personal growth.

Special thank goes to my colleagues and friends Simone and Giulia: it has been a pleasure to face together the laboratories, exams and difficulties of the last 2 years. The helping hand we gave each other has been fundamental, and the start of a strong friendship.

I am strongly thankful to all my friends, and to my bandmates with whom I share the love for music: the time spent together helped me to face all the obstacles of my career. My gratitude to Andrea (Dre) especially, the true and only friend I can always trust and rely on.

A heartfelt thank to my family for the support in the difficult times, and all the sacrifices made for my studies. I will forever be grateful to my grandparents, who encouraged me more than anything since the beginning.

Contents

Chapter 1

Introduction

1.1 Background

The development of Information and Communication Technologies (ICTs), together with an ever increasing digitalization, has allowed in the last years great strides forward in many different fields such as Green Information Technology, Internet of things, globally-connected networks and Artificial Intelligence (AI) [1].
In this framework, digitalization requires both an exponentially growing amount of data to be processed and great energy efficiency, along with computational speed. Unfortunately, these conditions can not be fulfilled anymore by traditional digital computers based on Complementary Metal Oxide Semiconductor (CMOS) technology, due to the limited throughput of the von Neumann architecture: in fact, the constant flow of data between central processing unit (CPU) and memory unit limits strongly the processing speed (von Neumann bottleneck).
In order to overcome the limits imposed by the von Neumann architecture, it is needed to exploit new computing paradigms in which data processing and data storage units are spatially correlated. Nature, in fact, has already implemented such paradigm in the human brain: the high connectivity of neurons by means of synapses together with the merge of processing and memory units, make it possible for the brain to elaborate efficiently massive quantities of data at high speed and with low power consumption. This has inspired the development of the so-called 'In-Memory Computing' (IMC).
First theorized by L. Chua in 1971, the Memristor is the best promising candidate among all the beyond-CMOS proposed nanotechnologies for IMC. The Memristor, thanks to its resistive switching properties, is a device suited for resistance-based Random Access Memories (ReRAMs) in which the information is stored thanks to a change in the internal resistance state. Moreover, the capability of reproducing neural synapses activity (i.e. synaptic plasticity, short/long term plasticity, spike-time dependent plasticity etc..) causes the Memristor to be thoroughly studied for reproducing neurons in nano-electronic structures.

1.2 Memristor

1.2.1 Memristor as fourth circuit element

Considering the work of Leon Chua (1971) [2], the Memristor (memory-resistor) has been firstly theorized as the missing circuit element which relates the flux (or voltage momentum) ϕ and the charge (or current momentum) q, in a pure mathematical concept.

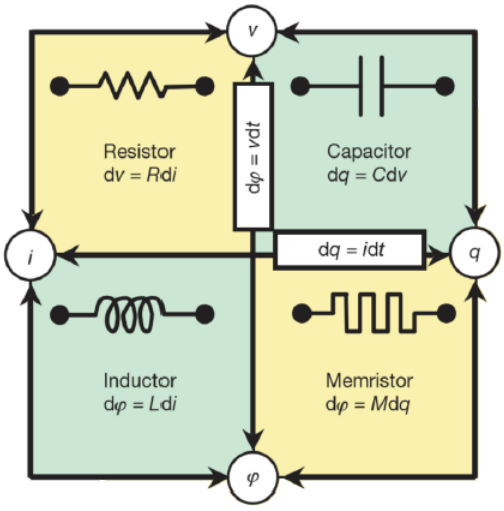


Figure 1.1: Fundamental electrical circuit elements

Thanks to the fundamental definitions of Current i and Voltage V, it can be expressed the flux and the charge by the constitutive relations:

$$\begin{cases} q = \int i dt \\ \varphi = \int V dt \end{cases} \Rightarrow \begin{cases} dq = i dt \\ d\varphi = V dt \end{cases} \tag{1.1}$$

Until 1971, the known fundamental circuit elements which relate i,V,φ and q were the resistance, the capacitor and the inductor:

- The resistor relates V with i through the Resistance R:

$$V = Ri \Rightarrow dV = Rdi$$

- The capacitor relates q with V through the Capacitance C:

$$q = CV \Rightarrow dq = Cdv$$

- The inductor relates φ with i through the inductance L:

$$\varphi = Li \Rightarrow d\varphi = Ldi$$

L. Chua's idea was to define theoretically a fourth circuital element which would connect φ and q, such that:

$$d\varphi = M(q)dq \tag{1.2}$$

It is important to note that $M(q)$ is not a constant, otherwise the new element would be just a normal resistor. By combining the constitutive relations (eq. 1.1) and eq. 1.2, in fact, it can be retrieved the classic Ohm's law:

$$V = M(q)i \tag{1.3}$$

The dependency of M on to the charge leads to a resistive system in which the internal state depends on the current/voltage history, meaning that there is memory of the past internal states.

1.2.2 Resistive switching in memristive devices

Due to the dependency of the memristance ($M(q)$) on the history, for the same applied voltage V_1 it is possible to retrieve different currents i_1, i_2 depending on which state the memristor had in the past. This behaviour is represented by a pinched hysteresis loop on the I-V curve, as shown in Fig. 1.2.

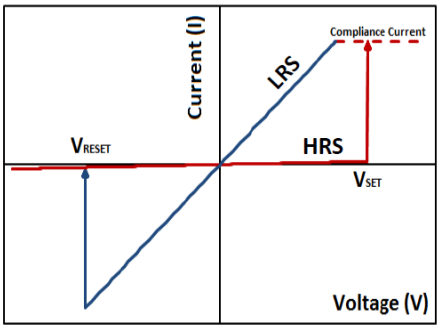


Figure 1.2: Pinched hysteresis loop, from R. Waser work [3]

The system switches between two different states: an high resistance state (HRS) and a low resistance state (LRS). The memristor has a threshold switching behaviour, meaning that a certain threshold (i.e. Voltage) must be reached in order for the switching mechanism to occur: it is needed therefore to apply a SET voltage (V_{SET}) to switch between HRS to LRS, and a RESET Voltage (V_{reset}) from LRS TO HRS. If the device is unipolar, V_{SET} and V_{RESET} have the same sign.

Until now, the memristor has been depicted as a pure mathematical concept, a theoretical circuit element with unique properties: a passive two-terminal element with pinched hysteresis loop in the I-V characteristic and non-volatile memory. The Memristor state should depend on a set of internal variables which are somehow related to the resistance of the system, without imposing any other conditions on the material and on the physics related to the resistive switching [4].
The first memristor was developed in HP labs by Williams et al. [5] in 2008: in this work,

it has been proposed the first experimental evidence after Chua's theoretical postulation in 1971. It showed that a system composed by a stacking of $Pt/TiO_2/Pt$ layers exhibits resistive switching behaviour, related to the drift of oxygen vacancies in the TiO_2 thin film layer with an external voltage applied (figure.1.3). The ON-OFF conditions in such

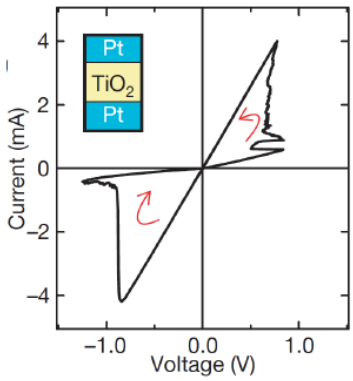


Figure 1.3: Resistive switching mechanism in $Pt/TiO_2/Pt$, reprinted from [5]

device are related to the creation of a narrow oxygen deficient region inside the metal oxide, changing the local conductivity of the material. Despite all the controversies and open debates on whether the ideal memristor is possible to be realized or not, William et al. work paved the way for new research fields in neuromorphic engineering and new non-volatile memories.

1.3 Physical mechanisms of Switching

The typical configuration for memristive devices is a stacking of $metal-insulator-metal$ layers, similar to a capacitor configuration. The physics behind the resistive switching depends strongly on the material of choose for the insulating layer and on the metal contacts; nevertheless, all the different possible configurations lead to a switching behaviour between an HRS (bit $'0'$) and a LRS (bit $'1'$).

The different natures of the resistive switching can be classified into 3 main groups: Magnetic, Electrostatic, and 'Atomic configuration' RAMs (figure 1.4).
The Magnetic RAMS (MRAMs) are based on changing the spin state of a material, and they are typically constituted by a sandwich of $ferromagnetic-insulator-ferromagnetic$ layers [6]. For example, in Spin-Torque transfer MRAMs (STT-MRAMs), the resistance of the system depends on the direction of the magnetization vectors of the two ferromagnetic layers thanks the *Magneto-Resistance* effect. When a current I is applied, the *minority* electrons (namely the electrons with spin polarization antiparallel to the magnetization vector of the first ferromagnet) are scattered back. If the second ferromagnetic layer has a magnetization antiparallel to the first layer, then the majority electrons are scattered back as well, leading to a high resistance (HRS). For the LRS, the two ferromagnetic layers have parallel magnetization vectors.
For what concerns the Electrostatic RAMS, one of the main solutions is the Ferroelectric

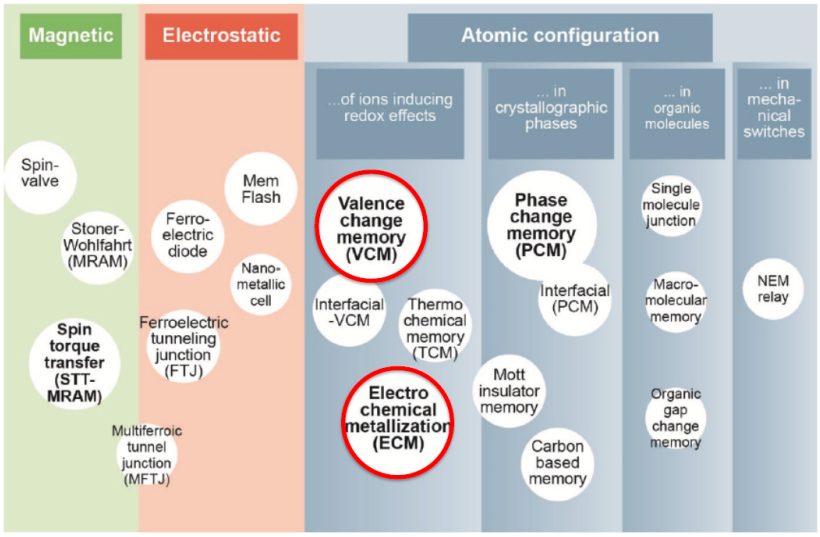


Figure 1.4: Resistive switching classification

RAM (FeRAM). The basic idea is to create a *1 Transistor- 1 Capacitor* memory cell, where the insulator in the capacitor is made of a ferroelectric material. Thanks to the ferroelectric properties, the capacitor possesses a natural polarization when short-circuited: in this fashion, the information is stored as voltage drop across the capacitor, which depends on the charge stored inside.
Lastly, the 'Atomic configuration' RAMs (RRAMs, or Memristors) are resistive memories in which the internal atomic structure of the material define the resistance of the device. It is composed by a sandwich of *metal-insulator-metal* layers, where the conductivity of the insulator layer changes according to different possible mechanisms: oxygen ion movement filament-based RRAM or metal ion movement filament-based RRAM.

1.3.1 Oxygen Ion Movement RRAMs

The simplest way to modify the conductivity of the device is to create a conductive filament inside the insulator such that electrons can flow easily between the two metallic electrodes. In Oxygen Ion Movement filament-based RRAMS, the conductive path is created thanks to the migration of oxygen ions: the creation of an oxygen deficient region leads to a change of the conductance due to the different local stoichiometry of the material. Typical materials used as insulating layers are metal oxides such as HfO_2,ZrO_2 or Ta_2O_5.

Valence Change Memory (VCM) Valence Change Memory RAMS (VCRAMs) are the most used Oxygen-based Memristors. These are bipolar devices characterized by an asymmetry in the structure: one electrode must be inert, while the other is reactive. The reactive electrode oxidizes at the interface reducing the oxide locally, so that the oxygen depleted filament can build up. In Fig. 1.5 it is depicted the resistive switching behaviour of VCRAMs in the case of ZrO_x.

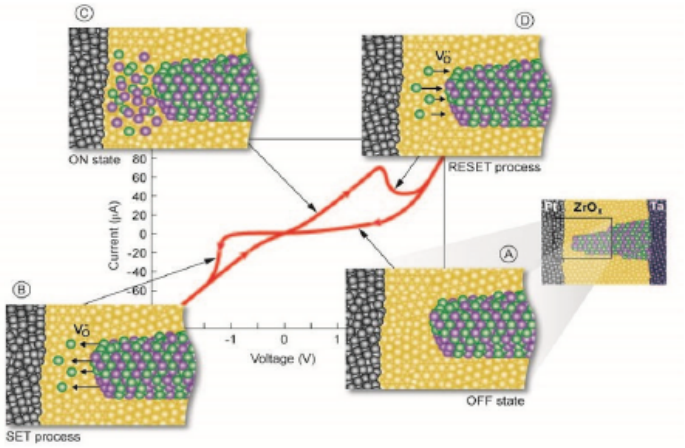


Figure 1.5: VCM resistive switching mechanism. Reprinted from [7]

When a SET Voltage V_{SET} is applied to the structure, oxygen ions move towards an electrode creating an oxygen-depleted region with increased conductivity. When a Voltage V_{RESET} with opposite polarity is applied, the oxygen ions move back interrupting the filament and resetting the system to an high resistance state. Let us follow the example of ZrO_x in Fig. 1.5. In this case, the stoichiometric rate reads:

$$ZrO_2 = Zr^{4+} + 2O^{2-} \tag{1.4}$$

On the other hand, the change in the valence of the metal is obtained by changing the stoichiometry of the metal-oxide:

$$ZrO = Zr^{2+} + O^{2-} \tag{1.5}$$

Since the change in stoichiometry leads to a transition between $2O^{2-}$ and O^{2-}, it is possible to define the oxygen vacancy V_o, which acts as a positive ion, such that when a electric field is applied the oxidation of the metal-oxide happens thus enabling the change in valence of the metal:

$$Zr^{4+} + 2O^{2-} = Zr^{2-} + V_o + \overbrace{O^{2-} + O^{2-}}^{O_2 gas} \tag{1.6}$$

An applied voltage on the electrodes causes the formation of O_2 gas and oxygen vacancies, and the migration of both O^{2-} and V_o. The evidence of gas production is reported in a work of Ilia Valov et al, where it has been observed the formation of water bubbles in a high relative humidity environment [8].

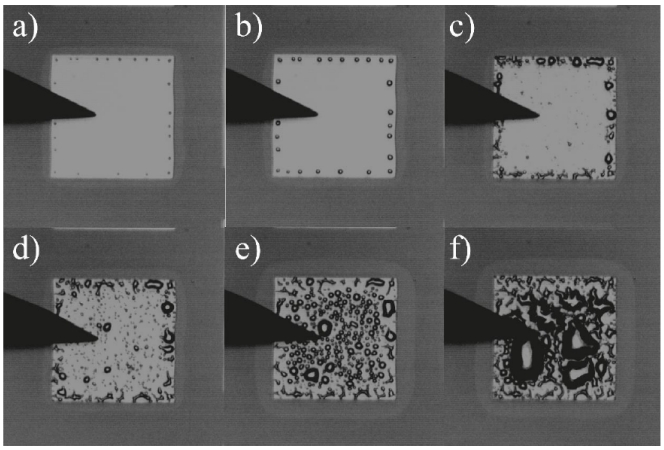


Figure 1.6: Formation and evolution over time (a-f) of liquid H_2O bubbles in $Ta/Ta_2O_5/Pt$ device at 80% relative humidity, highlighting the production of O_2 gas. Reprinted from [8]

Thermochemical memory (TCRam) TCRAMs are symmetric systems composed by a metal oxide (typically TiO_x or NiO_x) sandwiched between two inert electrodes.
The SET state of TCRAMs is similar to VCRAMs: in particular, in both systems the conductivity changes thanks to oxygen ion movement, but the main difference is that TCRAMs are unipolar (or non-polar) while VCRAMs are bipolar devices [9]. When a V_{SET} is applied, oxygen ions move away from the inner region creating a oxygen deficient filament, without necessarily migrating towards one preferred electrode. For the RESET state, a voltage V_{RESET} with the same polarity of V_{SET} is applied: a current starts to flow, heating the high conductive filament (due to Joule Effect) to the point of breaking it. TCRAMs have a very low endurance (1 to 100 cycles) and present a high reset current, making them incompatible for hard-drive disks or non-volatile RAMs; however, their unipolar nature could be suited for unidirectional diode-based crosspoint arrays [9].

1.3.2 Metal Ion Movement RRAMs

The Metal ion movement filament-based RRAMs rely on the process of Electrochemical Metallization (ECM). The ECM cell is composed by a $metal - insulator - metal$ system, where one electrode is made of an electrochemically active (EA) material such as Ag or Cu, and the other one is an inert electrode like W, Pt or Ir [10]. The creation of the filament is related to the migration of metal ions from the EA anode, as shown in Fig. 1.7.
Let us take as an example Ag as active metal electrode. For what concerns the SET state, a V_{SET} enables the oxidation of the anode Ag atoms:

$$Ag \rightarrow Ag^+ + e^- \tag{1.7}$$

Under the influence of the electric field, the positive Ag^+ ions diffuse inside the insulator and reach the cathode, where they get reduced:

$$Ag^+ + e^- \rightarrow Ag \tag{1.8}$$

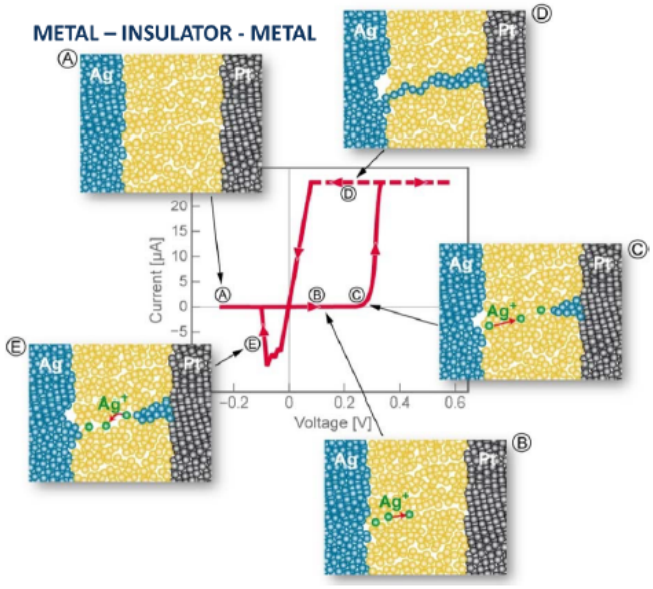


Figure 1.7: ECM resistive switching mechanism. Reprinted from [7]

After some time, the metal atoms start to form a filament leading to a low resistance of the device. In the RESET state, a negative voltage V_{RESET} causes the oxidation of metallic ions at the anode, and the subsequent rupture of the filament (i.e switch to high resistance state).

1.4 Neuromorphic Computing

First coined by Carver Mead [11], the term Neuromorphic Computing (or Neuromorphic engineering) determines the exploitation of analog electronic devices for brain-inspired structures which are capable of mimicking the architectures in the human nervous system. In particular, since neurons and synapses are the building block for creating artificial neural structures, it is important to understand how they are made and how they can process information and stimuli.

1.4.1 Neuron biological structure

Information in the brain is unidirectional, and is processed by neurons through particular signals called *active potentials*. The active potential is a voltage spike of $\approx 100mV$ peak-to-peak amplitude, in a Δt in the order of the milliseconds: the spikes are always the same, and the information is stored in the matching between different events. For what concerns the neurons, they consist in different parts: *soma*, *axon* and *dendrite*. A typical configuration is reported in Fig. 1.8.
The soma is the cell body of the neuron. The stimulus (action potential) is carried in the neuron through the dendrites, and is fired out in one single axon. Since there are

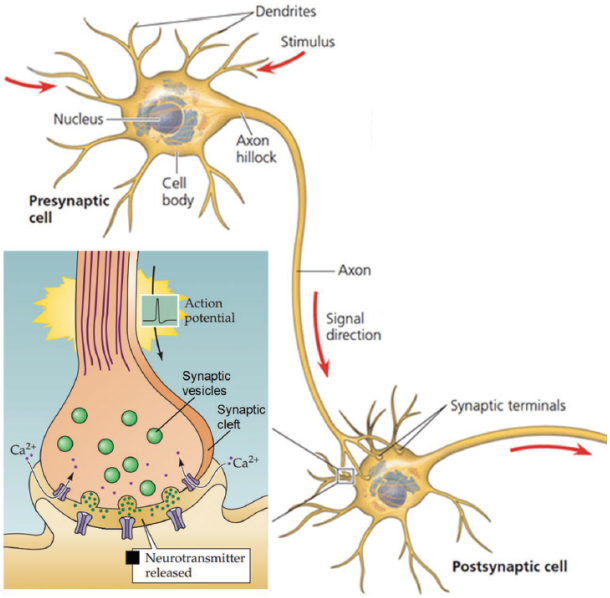


Figure 1.8: Representation of a neuron and synapses. Adapted and reprinted from [12]

different inputs but only one output (axon), the neuron is a voting system, meaning that the information is collapsed into one only output stimulus. This event follows the *integrate and fire* rule: each input pulse contributes to a short current pulse, and when the sum reaches a certain threshold than a voltage spike is fired in the output axon.
Lastly, the connecting structures between presynaptic cells and postsynaptic cells (fig. 1.8) are called *synapses*. They are responsible of transmitting the signal from the axon to the dendrite, thanks to neurotransmitters: the electric signal is converted into a chemical one in the presynaptic cell by releasing neurotransmitters through voltage-gated Ca^{2+} channels. The stimulus is switched back to a voltage pulse in the postsynaptic cell thanks to ligand-gated ion channels, which collect the neurotransmitters and propagate the information in the dendrite.

1.4.2 Synaptic plasticity and Memristive devices

Besides bridging different neurons together, synapses are capable of weakening (*depression*) or strengthening (*potentiation*) the connections. Also called *synaptic plasticity* [13], the modulation of the weights of the synapses is considered to be the critical point for peculiar brain functions such as memory and learning, and it must be studied in deep for artificial neural systems. There are two main effects: *long-term plasticity* (LTP) and *short-term plasticity* (STP).
LTP is defined as a potentiation/depression (of the synaptic weight) stable over time. An example is given by the phenomenon of *Spike-Time Dependent Plasticity* (STDP).: firstly suggested by Taylor M. in 1973 [14], it is a process for which the synaptic plasticity is re-adjusted depending on the temporal correlation between a pre-synaptic and post-synaptic signal (Fig. 1.9).

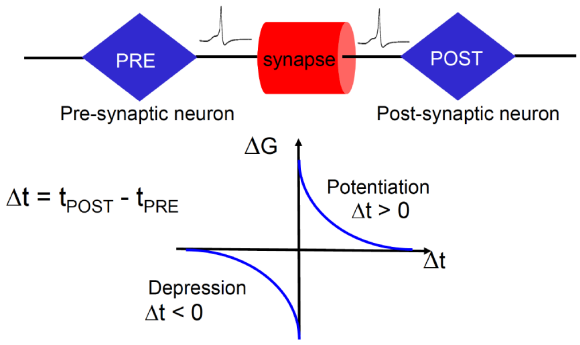


Figure 1.9: schematic overview of spike-time dependent plasticity. When the Δt between the pre and post synaptic signals is positive, long-term potentiation occurs. On the other hand, when $\Delta t < 0$, the synaptic plasticity is hindered leading to long-time depression (LTD).

STP, on the other hand, is the change in synaptic weight which is not stable over time (up to few minutes) [15]. A typical phenomenon of STP is the *Paired Pulse Facilitation* (PPF), in which the gradual potentiation of the synaptic weight is related to a number of input pulses with high spiking frequency. Right after the stimuli, spontaneous relaxation occurs.
Memristive devices are a promising technology for implementing artificial synapses, since they are capable of reproducing the changes in plasticity of the biologic neurons: in particular, the plasticity is linked to the change in conductivity, thanks to the resistive switching mechanisms which allow the device to have different conductance states. While the plasticity in the human brain is related to the ions concentration, in ReRAMs the change in conductance is related to the migration of ions (oxygen ions in VCM, metal ions in ECM) and the creation of conductive filaments between the electrodes (Fig. 1.10).

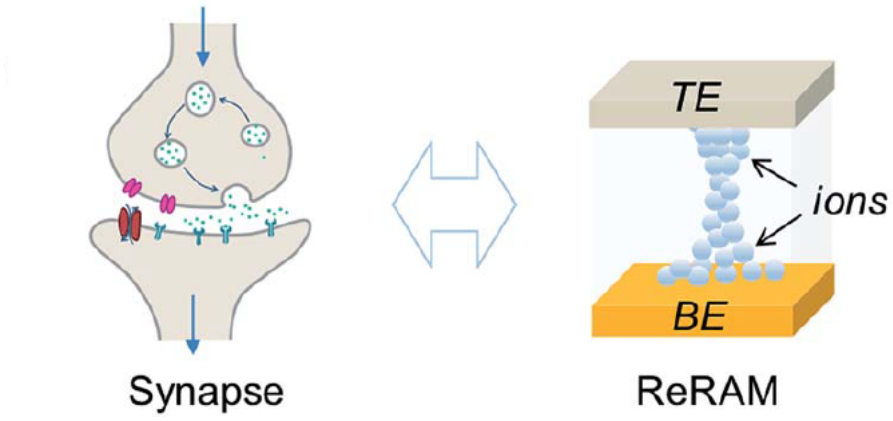


Figure 1.10: Parallelism between biological synapse (left) and artificial synapse (right) made of a ReRAM. Adapted and reprinted from [16]

There are two main memristive architectures which have been exploited for brain-inspired computing: memristive *cross-bar arrays* and *self-organized nanowire networks*. While they both rely on memristive processes for processing the information, they have different structures and characteristics, which will be discussed in the following.

1.4.3 Memristive crossbar arrays

Generally speaking, crossbar arrays (CBA) are structures with inputs and outputs which are arranged in a 2D intersection grid [17] (Fig. 1.11): at the cross-points between the row and column electrodes, memristive nodes represent the synapsis between pre-synaptic and post-synaptic signals.

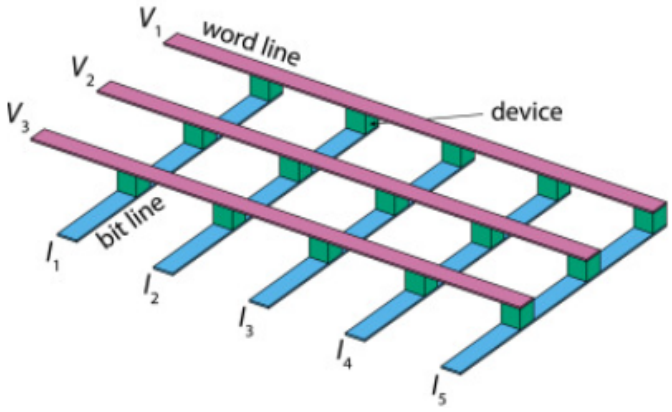


Figure 1.11: Crossbar array structure: inputs are applied to the electrode of the word line, while the outputs are obtained from the electrode of bit line. Reprinted from [17]

Thanks to their matrix-like arrangement, CBAs have been used to perform mathematical operations, such as matrix-vector multiplications (MVM), at low energy cost [18]. In MVM, for instance, G_{ij} coefficients are written in the memristor arrays as conductance states, and voltages V_j are applied to each row: the MVM can be retrieved by measuring the currents in the bit lines, such that $I_i = \sum_j G_{ij} V_j$. In this framework, CBAs offer *one-shot* solutions of matrix operations [19] just by exploiting Ohm's and Kirchhoff's laws for passive element circuits.
CBAs are mostly used as hardware implementation of Neural Network algorithms [20], in which the weights between the different neurons (see Chapter 3.4) can be stored offline (*ex situ*) as resistance states in the crossbar memristors. The tunable synaptic weights in each memristive cell make the CBAs suitable for a wide range of Artificial Intelligence applications, such as: object recognition [21], brain–machine interfaces [22], image processing [23], audio signal processing [24] and so on.

1.4.4 Self-organized networks

The highly organized networks, such as crossbar arrays and integrated circuits, lack of the complex interconnectivity which is typical of neural structures. For this reason, a lot of effort has been done in neuromorphic engineering in order to find new structures which

better reproduce the brain's architecture: in this framework, self-organized networks are promising technologies for fully-connected systems at low costs of fabrication and low power consumption.

In a S.A.Brown et al work [25], a self-organized network of conductive nanoparticles has been created on an insulating substrate by percolation. The particles form groups divided by tunnel gaps, in which synapse-like switching processes occur (Fig. 1.12).

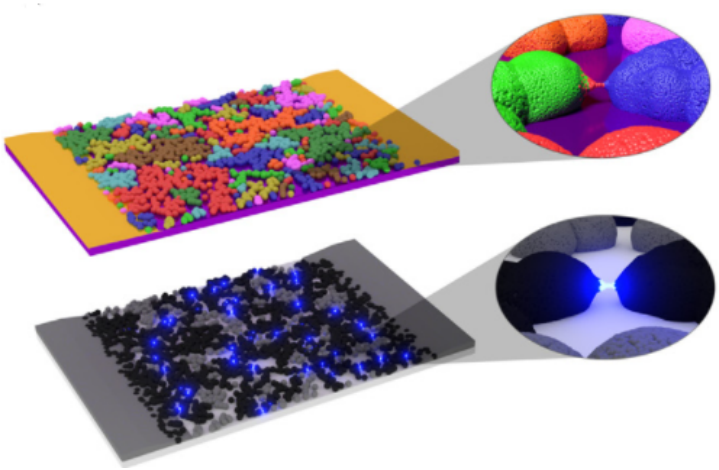


Figure 1.12: Example of percolating system of self organized nanoparticles. In the upper figure it is reported the groups of connected nanoparticles in different color, while in the lower figure it is represented the formation of conductive filaments in the tunnel gaps. Adapted and reprinted from [25]

.

Neuromorphic computation has been achieved as well in nanoelectronic systems composed by quantum dots self-assembled on a tunneling resonant diode [26], in which the complex spatial and temporal patterns given by the redistribution of the charges in the quantum dots out of equilibrium lead to neuromorphic computational effects.

Lastly, memristive nanowire (NW) networks are the most promising architectures for neuromorphic computing. Basically, these systems are composed by metallic nanowires randomly distributed in a highly interconnected network with self-assembly techniques. Thanks to their designless structure, extremely cheap fabrication and complex dynamics, self-organized NW networks are considered to be the best candidate for reproducing brain-inspired biological neuronal circuits. In addition, these systems can be grown by using bottom-up approaches, which help to go beyond the limits of scaling of the top-down lithographic approach [27].
In NW networks, the memristive processes which enable resistive switching behaviour in the device take place at the cross-point junctions between the different metal NWs. First observed by Cagli et al. in 2011 [28] in Ni–NiO core–shell NWs and then further elucidated by Ting et al. [29], the local morphological structure in Ni-NiO junctions were changed due to the creation of conductive filaments by oxygen ion movements (Fig. 1.13). By ex-

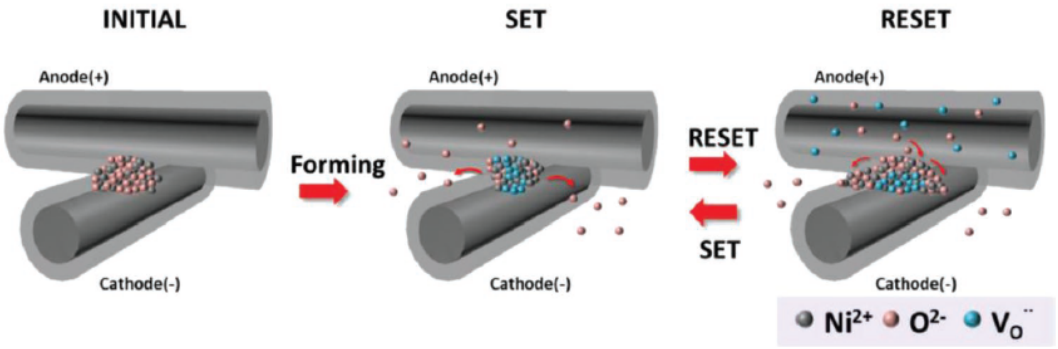


Figure 1.13: Resistive switching mechanism in a Ni/NiO junction due to VCM memristive effects. The creation of a conductive filament changes the conductance between the two NWs, leading to a preferred path for electrical conduction. Reprinted from [29].

tending the concept to highly interconnected NW networks, the global resistive switching behaviour of the system is related to the local creation/destruction of conductive filaments between the NWs, determining a time-evolving connectivity of the network [30][31].

Depending on the input stimuli applied on the network's electrodes, it has been first observed by Diaz-Alvarez et al. [32] peculiar properties, which attracted great attention in neuromorphic engineering:

- Collective memory response. Short-term memory is observed, related to non-linear temporal dynamics when the stimuli are switched off.
- Distinct global resistive switching between LRS and HRS.
- Synaptic plasticity and adaptive behaviour.
- Reconfiguration dynamics and self-healing processes

The overall dynamics are responsible for both homo- and heterosynaptic plasticity in the network. In particular, while homosynaptic plasticity occurs at synapses that are directly activated in the input stimulus and governed by Hebbian learning rules [33], heterosynaptic plasticity refers to the change in strength of neurons which are not directly stimulated [34] providing stable learning and enhancing synaptic competition [35]. In this framework, the inputs are mapped by the network in conductive paths: the system itself is capable of selecting the paths between the electrodes following energy efficiency principles (Fig. 1.14) [36].

In summary, Memristive NW networks are systems which mimic efficiently the way the brain processes the information, by exhibiting high connectivity and plasticity effects: moreover, these kind of networks are shown to be small–world architectures akin to the neural networks of *C. elegans* [37].

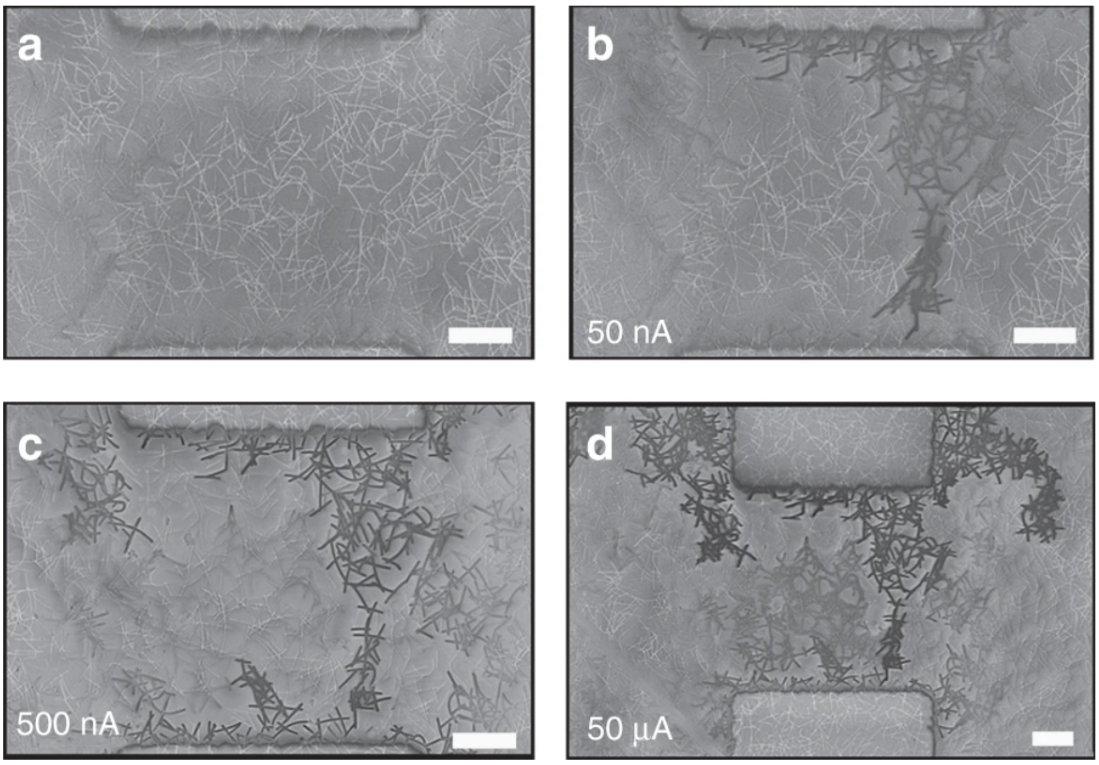


Figure 1.14: Winner-takes-all (WTA) conduction between two electrodes, with current compliance of 0 A (a), 50 nA (b), 500 nA (c), 50 μA (d) (scale bar= $2\mu m$). Reprinted from [36]

A lot of work has been done in order to create self-assembly systems capable of mimicking the brain's nature: in this work, memristive NW networks are going to be explored in depth, in order to prove their strength in neuromorphic computing.

Chapter 2

Memristive Nanowire Networks

The aim of this work is to develop a computational model for Reservoir Computing (RC), exploiting memristive nanowire networks for processing input data. For this purpose, it is important to understand how the device can be modelled and implemented in software, starting from its physical properties. The approach which will be discussed and exploited for the simulations consists in modelling the complex nanowire network as a simple graph model, where each edge will evolve in time in terms of conductance following a balanced rate equation for memristive processes.

2.1 Memristive devices based on nanowire networks

In this work, the network used as reference for the modelling is taken from a device made of Ag NWs of 115 nm diameter and length of 20-50 μm. In particular, the NWs are drop-casted in isopropyl alcohol suspension ($\approx$ 0.13%) on a SiO_2 1μm substrate, after spontaneous solvent evaporation. As a consequence of the polyol synthesis process (used as surfactant), a layer of polyvinylpyrrolidone (PVP) covers the surface of the Ag NWs: in this way, the PVP shell layer acts as dielectric in which Ag^+ ion migration occurs at the intersection of the NWs, forming the conductive filaments typical of memristive ECM effect [38].
At last, Au pads are deposited by sputtering for the electrical contacts: the sputter process is needed in order to etch the PVP layer around the Ag NWs in the proximity of the electrodes, thus allowing for a direct contact of the Au with the core of the nanowires. It is important to highlight the low cost of the fabrication, and the fact that it is not needed any lithographic process or cleanroom facilities.
In Fig. 2.1 it is reported both SEM and optical images of the memristive NW network: in this way, it is possible to better appreciate the high connectivity of the device, and the neural circuit-like structure that can be retrieved with simple self-assembly techniques.

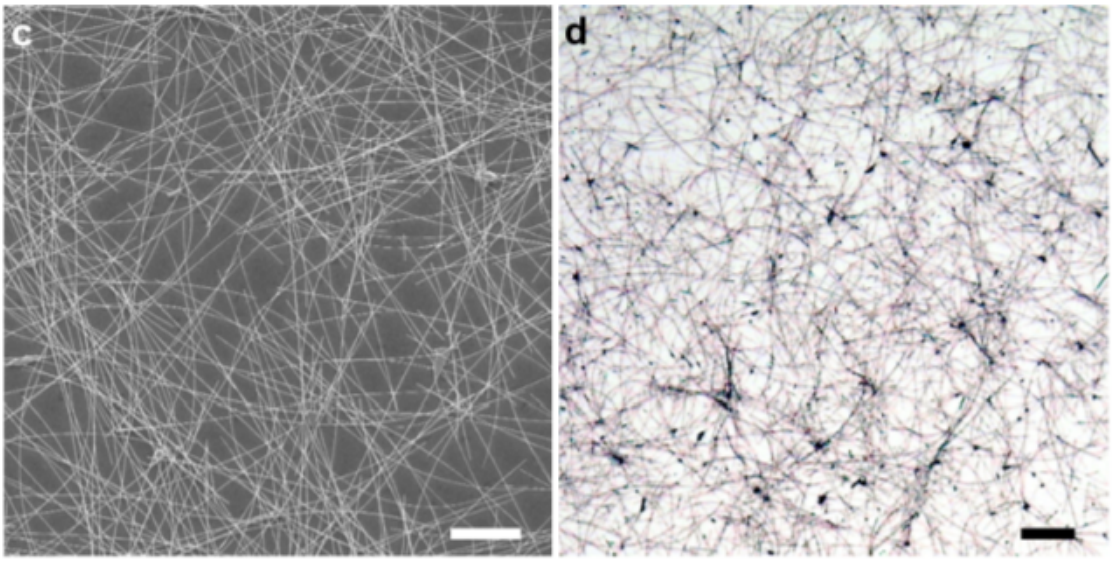


Figure 2.1: SEM image (left) and optical image (right) of NW network topology (scale bar= $50\mu m$). Adapted and reprinted from [39]

.

The two important phenomena which can be observed in the Ag NW network, and which contribute to the overall structural plasticity, are the *rewiring* and *reweighting* effects (Fig. 2.2 (a-b)).

- *Rewiring.* When high current densities flow in the single nanowire, electromigration occurs: due to the exchange of kinetic energy and momentum between electrons and atoms, and thanks to the Joule's heating process, ions start to get displaced by their equilibrium position in the crystal lattice and a needle-like gap starts to form in the NW. It is interesting to notice that the broken NW starts to behave as a new memristive cell, in which the electrical connection can be restored by the formation of a conductive filament in the nanogap, exhibiting the classic pinched hysteresis loop in the IV characteristics.

- *Reweighting.* A potential difference applied at two intersecting NWs induce ECM processes, in which anodic dissolution of Ag atoms and migration of Ag^+ positive ions in the PVP insulating shell create conductive filaments which bridge the cross-point junctions between NWs. The weight change of the memristive filaments lead to the formation of conductive paths in the network, according to the different stimulations.

The rewiring and reweighting phenomena of the NWs lead to the emerging global memristive properties of the network. In particular, by applying positive and negative voltage sweeps in the range [-0.8, +1]V in a two terminal configuration of the network, it has been observed the typical hysteresis of the IV plot with distinct HRS and LRS (Fig. 2.2 (c)).

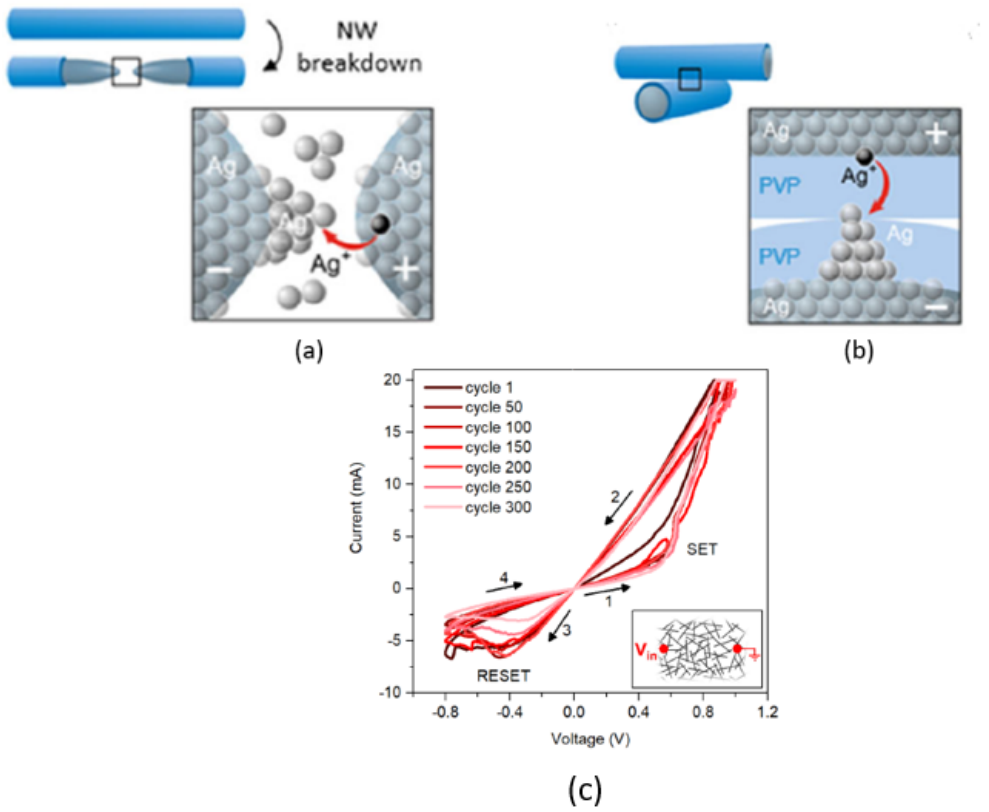


Figure 2.2: (a) NW breakdown (rewiring) and (b) conductive filament formation (reweighting) at the cross-point junction in the Ag memristive NW network. (c) Network's resistive switching behaviour in two terminal configuration. Adapted and reprinted from [38]

2.2 Modelling memristive nanowire networks

Let us now address the model structure, which will be implemented in *Python* environment.

2.2.1 Balanced Rate Equation

Described by Miranda et al. [40], a balanced voltage-controlled rate equation can be exploited to model STP effects in ZnO NWs between Ag-Pt electrodes, including potentiation, depression and relaxation effects due to external electrical stimuli.
The equations deal with the change in conductance of the device due to Ag^+ ions migrating on the crystal surface of the NW: despite being this particular memristive effect different from the effects shown in a memristive NW network, the proposed equations can be extended to all the possible structures which deal with the migration of Ag^+ ions.

In the simple analytical model, the conduction states after the filament formation are

described by the linear relationship:

$$I = [G_{min}(1-g) + G_{max}g]V \tag{2.1}$$

where g is the normalized conductance (0<g<1), and G_{min},G_{max} minimum and maximum values of the experimental conductance. The rate balance equation for the conductance dynamics as a function of the ions migration is expressed as:

$$\frac{dg}{dt} = k_P(V)(1-g) - k_D(V)g \tag{2.2}$$

The parameters k_P and k_D are the potentiation and depression coefficients, which for simplicity are given as a function of the applied voltage V only as follows:

$$k_{P,D} = k_{P0,D0}e^{\pm\eta_{P,D}V} \tag{2.3}$$

where $k_{P0,D0} > 0$ are fitting parameters and $\eta_{P,D} > 0$ transition rates. Due to the exponential dependency (typical of diffusive ionic process [41]), the different polarity of V give raise to several order of magnitudes of difference in $K_{P,D}$.
The final solution, assuming positive timesteps Δt, is a simple recursive equation in the form:

$$g_t = \frac{k_P}{k_P + k_D}\left\{1 - \left[1 - \left(1 + \frac{k_D}{k_P}\right)g_{t-1}\right]e^{-(k_P+k_D)\Delta t}\right\} \tag{2.4}$$

where the voltage dependency is hidden in the potentiation and depression coefficients. Eq. 2.4 will be exploited in order to model the dynamics of the system.

2.2.2 Network as a memristive grid

The real NW network device consists of millions of synapses per square millimeter: it is clear that such a complex system is impossible to be faithfully reproduced in software. It is important to highlight that the aim of the model is to both simulate the physics of the device and simplify the structure, in order to reduce the complexity of the system without losing any information. In this framework, a subset of the synapsis can be simulated by means of a square grid of nodes connected by random diagonals (Fig. 2.3): the nodes map the Ag NWs, while the edges map the PVP insulating layer.
A conductance state is associated to each edge, with dynamics driven by the balance rate equations (eq. 2.4): when a voltage is applied, the edges evolve in conductance leading to conductive paths which resemble the formation of filaments in the cross-point NW junctions. In the simulations, any grid's dimension is acceptable: practically speaking, it must be tuned according to the inputs to be processed and to the physical dimensions of the device to model.

For what concerns the evolution of the graph, it has been simulated thanks to the Modified Voltage Node Analysis (MVNA) algorithm: this method solves the circuit as a grid of passive elements, in which each memristor is defined with a certain resistance state depending on the evolution. By implementing the Kirchoff current law and placing voltage generators between stimulated nodes, it can be calculated the voltage drop at each node in every timestep of the simulation: in this fashion, the memristor conductances will be updated

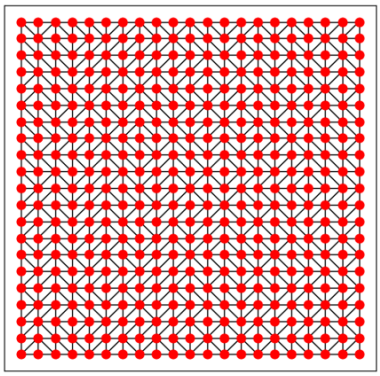


Figure 2.3: Graph reproducing the memristive NW network. The nodes (red dots) represent the NWs, while the edges (black lines) represent the PVP insulating layer and conductive paths.

following the balance rate equations, depending on the V on each edge, and each node is connected to an electrode.
The algorithm solves a linear system of the form:

$$Ax = z \tag{2.5}$$

For a system of n nodes and m independent voltage sources:

- A is a *(n+m)x(n+m)* matrix. In the upper left *nxn* part of the matrix, the elements conductances connected to ground appear on the diagonal while elements not connected to ground are on both diagonal and off-diagonals. The rest of the matrix contains 0,-1 or 1, corresponding to the presence of a source between two nodes.

- x is a *(n+m)x1* vector. The upper n elements are the node voltages, while the bottom m terms represent the currents through the m independent voltage sources.

- z is a *(n+m)x1* vector. The upper n elements are the sum of currents through each element connected to a certain node, while the bottom m terms represent the independent voltage sources.

Let us show an example of the network evolution in terms of conductance with a simple input. In particular, a voltage pulse of 9V in a time of 5ms is applied between one source node and one ground node, then the input is cleared for 15ms (as shown in Fig. 2.4).

The network states are reported in four different times in Fig. 2.5. In particular, it can be noticed the formation of a conductive path when the voltage pulse is on ($t = 1ms$ and $t = 4.6ms$): in this potentiation phase, conductive filaments are creating between the NWs leading to a overall low resistance state.
When the voltage is set to 0 ($t = 11.8ms$ and $t = 20ms$), it can be observed the spontaneous relaxation, in which the conductive filaments break down and the system returns autonomously to an high resistance state.

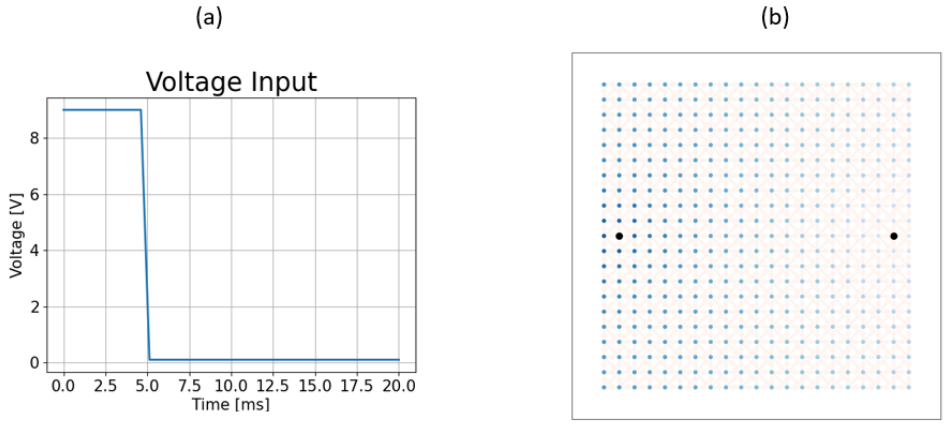


Figure 2.4: (a) Input voltage pulse. (b) 21x21 grid with source node (left black dot) and ground node(right black dot).

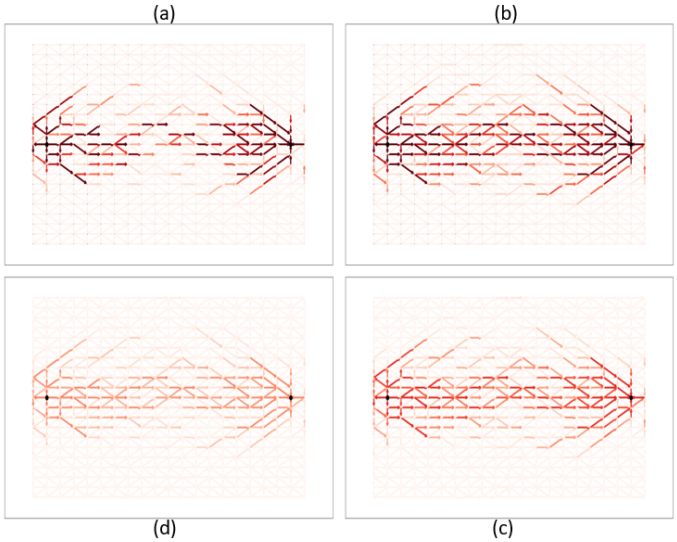


Figure 2.5: Evolution of the network conductance at: (a) $t = 1ms$ (b) $t = 4.6ms$ (c) $t = 11.8ms$ (d) $t = 20ms$. The darker is the edge, the higher is the conductance.

2.2.3 Setup for computing implementation

The scope of the model is to map the input electrical stimuli in conductance maps, thanks to the dynamics of the memristive grid. In order to be able to analyse and exploit the network states, it is needed to read the voltages on the nodes: in this framework, the inputs will be described by output voltage vectors, which depend on the conductive paths formed in the process. An insight of the writing and reading operations is now provided.

Setup The setup which has been exploited involves the connection of the grid of electrodes (i.e nodes of the graph in Fig. 2.3) on the NW network to an external ground (GND) by means of a resistance R, as shown in Fig. 2.6. Each resistance potential drop

is monitored in both set and read times, in order to perform writing (stimulation) and reading (output storage) operations.

Writing Operation In order to simulate the active potential in the synapses, the stimulation of the nodes on the grid happens thanks to voltage spikes applied on the corresponding electrodes. As depicted in Fig. 2.6, the input pulses are modelled as signal generators in series to a resistance connected to the different pads: when a node is stimulated, the potential drop with respect to the nearby nodes lead to a change in the edges conductances, following the balance rate equation dynamics. The writing operation serves to model the creation/dissolution of the memristive conductive filaments (edges) at the cross-point junction between the nanowires (nodes).

Reading Operation In order to be able to reproduce in hardware all the proposed simulations, it has been chosen to read the output voltages with a methodology which has been already tested experimentally. In particular, in a grid composed of M electrodes, 1 electrode serves as bias and the V_{out} is retrieved on the other M-1 nodes. On the bias node, a small bias DC voltage V_{read} (in the order of millivolts) is applied: in this way, it can be measured the potential drop across the resistances on the other electrodes. In Fig. 2.6, it is depicted a simplified scheme of the writing and reading operations on the memristive grid.

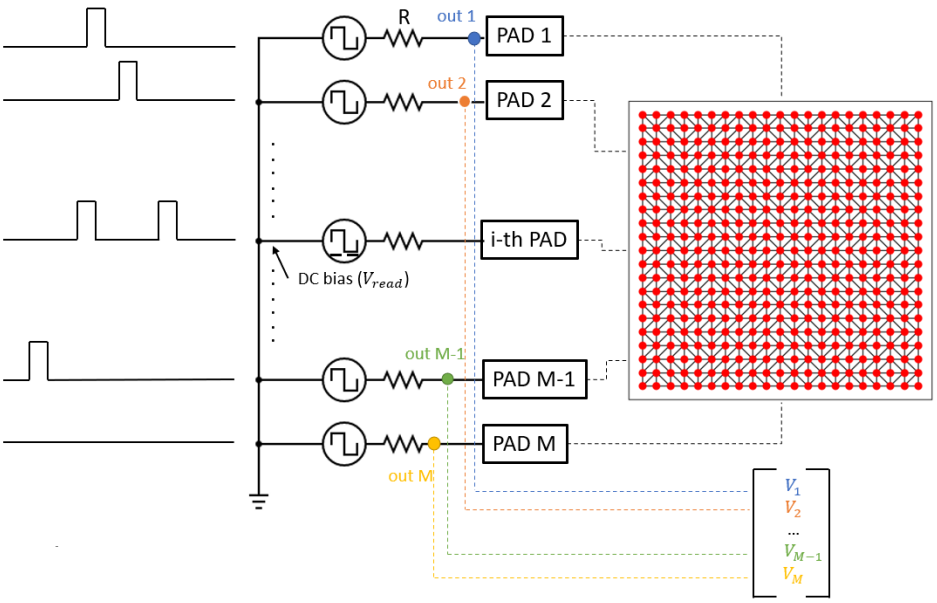


Figure 2.6: Schematic representation of an example of writing and reading operation on the network grid. The pads are connected to signal generators which provide the input voltage spikes (M PADS for M nodes of the grid) for the writing operation: one arbitrary PAD is biased with a small DC voltage V_{read} in addition to the input voltage stream, and the voltage drops on the terminal out_i of the series resistances R is stored in a vector. The final vector consists therefore in M-1 values of output voltages, which is a fingerprint of the network conductance state.

Although this present work will deal with speech recognition task and processing of audio

samples, let us take for now an example of a written digit as input of the network, in order to enforce the wide spectrum of possible inputs that can be processed in this framework. In Fig. 2.7 it is depicted the procedure to transform a written digit to electrical stimuli.

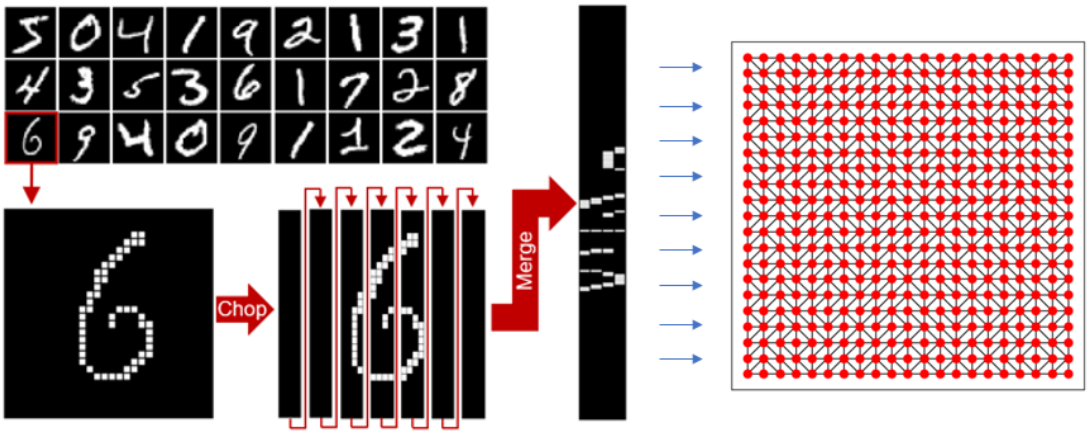


Figure 2.7: MNIST handwritten digit patterns. The original 28x28 pixel image is transformed in a 196x4 patter (196 pulse streams with 4 timesteps of simulation).

The original images, consisting in 28x28 grids representing the digitalized handwritten digits, are transformed in 0 (black) and 1 (white) pixels: the image is then chopped and merged in a 196x4 matrix. In this framework, the 4 columns represent the timesteps of the simulation, while the 196 rows are pulse streams which are sent to the nodes/electrodes of the memristive grid. In particular, when the pixel contains the BIT 1, it produces a voltage spike on the related electrode of $5.0V/10ms$; if the pixel has a BIT 0, it is not applied any stimuli and the node is relaxed. The spikes are induced by signal generators connected to the network pads through a resistance (Fig. 2.6): a similar procedure of handling the inputs will be adopted for speech recognition task (see Chapter 3), with differences only in the pre-processing technique. Once the voltage pulses are applied to the corresponding nodes, the memristive edges will evolve in conductance according to the balance rate equations, leading to the creation of conductive paths. In Fig. 2.8, it is shown the dynamics of the grid in the 4 different timesteps of simulation.
By reading the output voltages on the nodes in the last timestep, the handwritten '6' can be finally non-linearly transformed by the network in conductance state of the grid, which will be further used for classification.

2.2.4 Influence of model parameters

At last, let us inspect the conductance maps retrieved by the simulations, understanding how the parameters of the balance rate equations can influence the final result.
Let us now understand the influence of the model parameters on the conductance states of the NW network. In particular, two coefficients are analysed: η_p and η_d (from eq. 2.3). Generally, these coefficients are retrieved by fitting the experimental conductance graphs as a function of time: for example, in Fig. 2.8 it is used the fitting parameters coming from

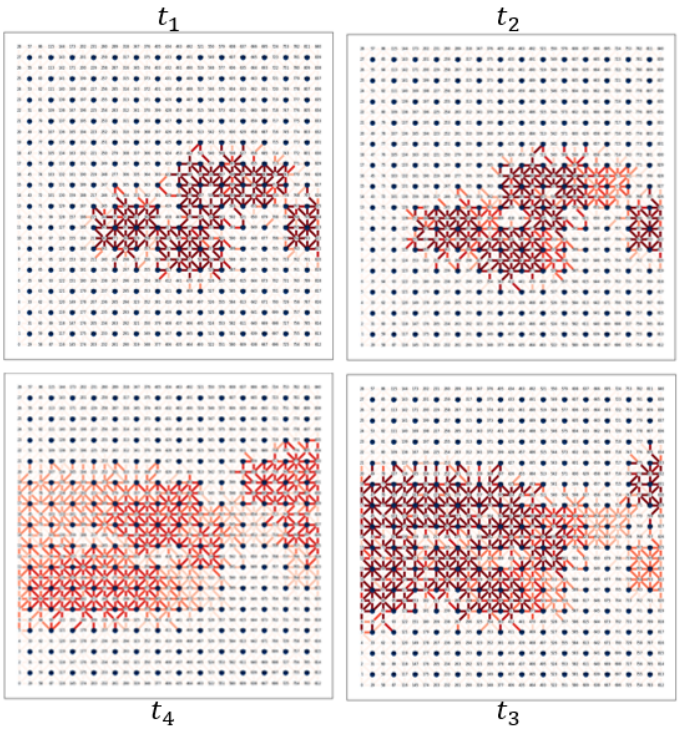


Figure 2.8: Evolution of conductive paths in the four timesteps of simulation, with handwritten digit '6' as input. The shade of red indicates the level of conductivity.

experiments on the hardware device. In the following, it is changed manually the values of the potentiation/depression coefficients as shown in the following table 2.1.

Table 2.1: Init. Config. refers to the initial configuration coming from fitting of the experiments.

	Init. Config.	*Config. #1*	*Config. #2*	*Config. #3*
η_p	$3.49215e+01$	$3.49215e+01$	$2.49215e+01$	$0.33002e+01$
η_d	$5.59060e+00$	$20.59060e+00$	$0.90060e+00$	$10.79060e+00$

In order to understand the parameter influences, it has been performed the simulation with the same input (handwritten digit '6'). The results are shown in Fig. 2.9.

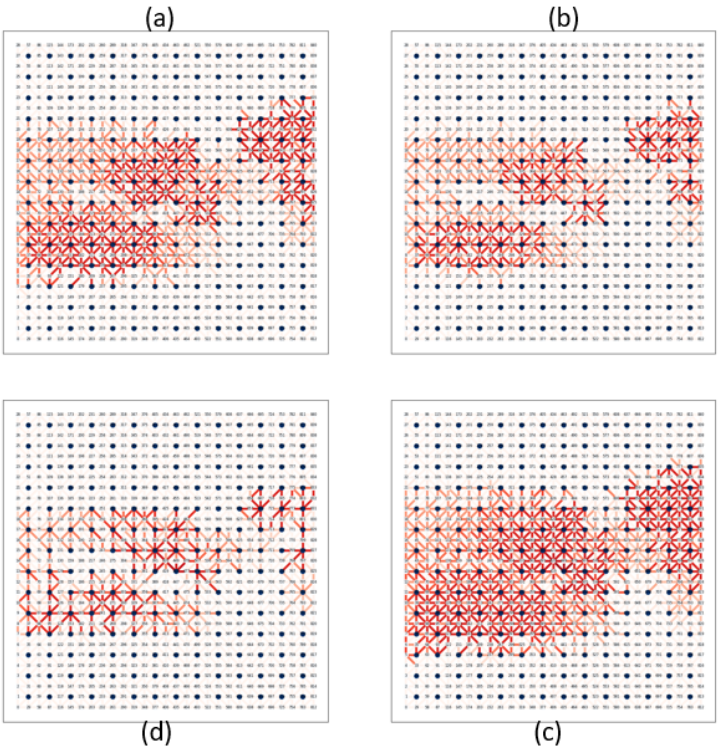


Figure 2.9: Conductive maps at the final timestep of the simulation, with (a) Init. Configuration, (b) Config. #1, (c) Config. #2 and (d) Config. #3.

In particular, it can be noticed that the maps have the same morphology in all the trials, while the degree of potentiation of the edges changes with changing the parameters. η_p and η_p must be therefore tuned in order to avoid saturation of the network, but they do not control the formation of the conductive paths.

Chapter 3

Reservoir Computing for Speech Recognition task

The scope of the developed model is to transform inputs (e.g audio signals) into electrical stimuli which will be processed by the network: thanks to the evolution of the memristive edges, the data can be transformed non-linearly in conductance maps, thus exploiting the high connectivity and short term memory of the network. The inputs are therefore brought in a high-dimensional space where the spatio-temporal correlations can be enhanced and further analysed for machine learning: in the Artificial Intelligence field, this paradigm is called Reservoir Computing, and the Memristive NW network serves as Reservoir for the computing algorithm.
In this chapter, it will be discussed the whole Reservoir Computing paradigm which has been developed in order to solve the Speech Recognition task. In particular, it is used audio samples in which different people are recorded pronouncing the digits from zero to nine: at the end of the process, the system must be capable of correctly recognize the digit which is spoken in the samples.

3.1 Introduction

Reservoir Computing (RC) is a computing paradigm which allows for processing of time-varying data [42], through a physical system called reservoir. RC's aim is to reduce the computational costs by fixing the dynamics of the reservoir [43], and in the same time exploit its complexity and non-linearity in order to process the inputs and extrapolate their spatio-temporal correlations. In Fig. 3.1 it is represented a simplified scheme of the RC paradigm.
Originally, RC is developed as a Recurrent Neural Network (RNN) framework [45], in which the internal state of the network can be used to process time-varying sequences of inputs of variable length.
RNNs are networks in which a feedback is introduced in the nodes, such that information of past inputs influence the processing of current inputs and outputs: in this fashion, when the network gets more and more complicated, it gets more challenging and computationally expensive to train the weights for classification. The introduction of a reservoir, on

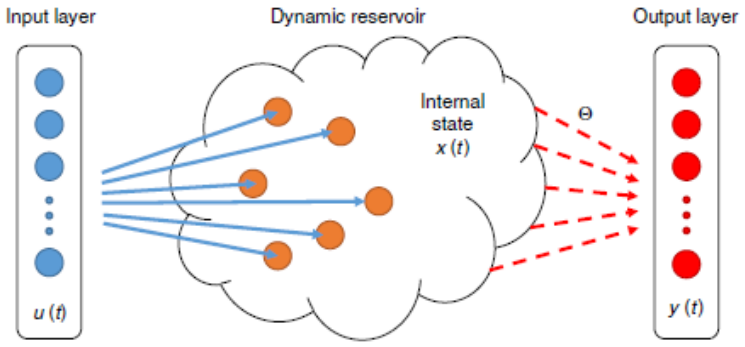


Figure 3.1: Schematic representation of an RC paradigm. The only part which needs to be trained are the weights Θ for the output layer. Reprinted from [44].

the other hand, implies the integration of a RNN-like system which does not require any training, and which could both be a physical or virtual device.

RC paradigm requires three main blocks: pre-process, reservoir, and readout. Although the algorithm structure is general, the pre-processing techniques and readout functions must be chosen according to the task to be solved. In fact, RC can be exploited in a wide spectrum of applications, such as: biomedicine (electrocardiogram classification [46], processing of EMG signals [47]), audio processing [48], image recognition, engineering, security etc. In Fig. 3.2, it is reported the RC scheme developed for solving the Speech Recognition Task.

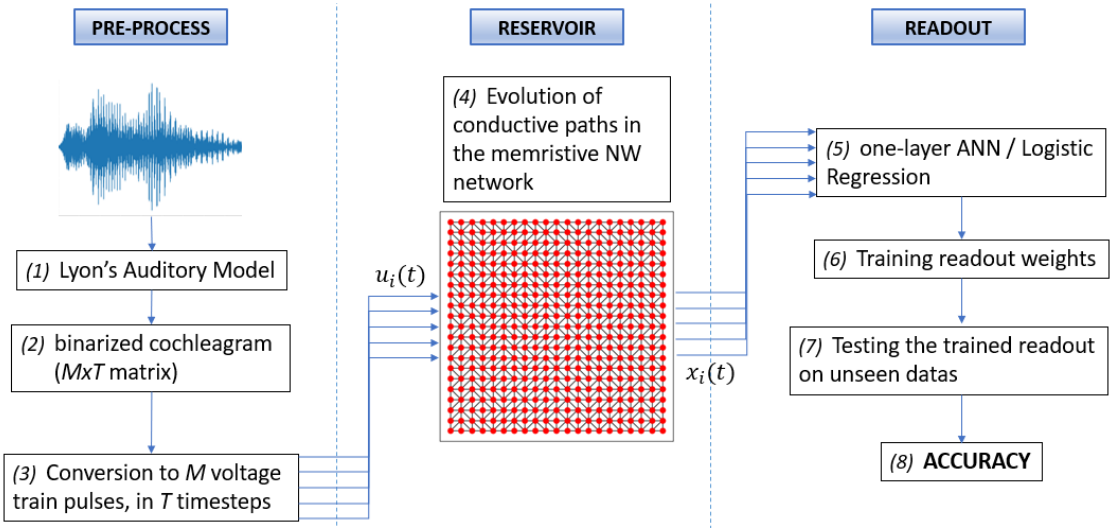


Figure 3.2: Schematic representation of the RC paradigm for speech recognition task. The audio samples are converted in $u_i(t)$ train pulses, and sent as input to the reservoir. The reservoir conductance states are recorded in $x_i(t)$ vectors (i.e output voltage values of the network's electrodes), and sent as input for the readout function.

3.1.1 RC paradigm overview

Let us provide a general framework on the RC paradigm scheme, which will be deeply discussed in the following sections.

Pre-Process Depending on the application, the input data can be of any nature. It is therefore mandatory to build a suitable pre-processor which transforms the data in signals which will be processed by the reservoir. For example, in this work audio signals are converted in voltage train spikes which are sent to electrodes on the reservoir (see Chapter 3.2.1).

Reservoir In order to be exploited as reservoir in RC, a system must satisfy four main conditions [49]:

- *Nonlinearity*, in order to differentiate inputs which are not linearly separable. In memristive devices, the nonlinearity is introduced thanks to the potentiation and relaxation of the conductive filaments, which lead to pinched hysteresis loops on the IV characteristics.
- *High dimensionality*. Inputs must be brought to high dimensional spaces in the reservoir, in order to enhance correlations which facilitate the classification tasks. In the NW network, the dimensionality is given by the grid size and the connections between the different nodes.
- *Short-term Memory*. Fading memory (also called echo state memory [50]) is an essential property: it is responsible for the influence of the recent-past inputs in the dynamics of the reservoir in sequential data.
- *Separation property*, to allow the separation of different inputs by the responses of the reservoir.

Apart from the memristive NW network used in this work, different systems fulfill the conditions required. For example, RC has been performed by exploiting a fluidic Reservoir composed by "water in a bucket" [51]: the inputs are transformed in mechanical stimuli in a bucket full of water, producing ripples on the surface of the liquid. The ripples are recorded with a video camera and processed by a readout function, and in this way spoken digit recognition task is demonstrated.
Electronic RC has been also implemented. For example, in a work of Appeltant et al. [52], a single node made of a nonlinear analog electronic circuit is exploited as a reservoir. Also Very Large-Scale Integrated circuits (VLSI) has ben exploited, along with Field Programmable Gate Arrays (FPGA) [53].
For what concerns the Photonic field, a lot of effort has been done to implement photonic reservoirs in machine learning algorithms. A delay-coupled photonic system is used as single physical node [54], composed by a laser source, a Mach-Zender modulator and a feedback system with gain. Other examples are given by a optical feedback in a semiconductor laser cavity [55], or by single-mode waveguides integrated on chip with Semiconductor optical amplifiers (SOAs) [56].
Spintronic [57], Mechanical and Biological [58] RCs have been proposed as well, demonstrating the wide spectrum of systems that may be used for reservoir computing.

Readout Function. The Readout is the only block in the RC paradigm which needs to be trained. It takes as input the reservoir states which map the original datas, and exploits them for solving the recognition tasks through supervised learning algorithms. Simple readout functions can be exploited, such as Regression models (Logistic, Ridge, Linear etc..), or more complicated ones such as FeedForward Neural Networks.
Depending on the readout system, the training can be performed either online (in real time) or offline, by previously simulating the network and then storing the weights in the conductance values of a cross-bar array RERAM.

3.2 Pre-Process

The pre-process is the first block in the RC paradigm: its role is to convert the inputs to be classified in electrical stimuli for the reservoir, such as voltage excitations applied on electrodes on the nanowire network. In this framework, the inputs to be processed are audio samples, which is to say sound waves captured by the microphone: it is therefore needed a model which converts these continuous signals into discretized train spikes for the reservoir.

Sound is a mechanical longitudinal wave, in which local pressure variations propagate as waves inside a transmitting medium. Humans have developed the ability to manipulate sound waves through the vocal cords, through which it is possible to replicate different arrays of sound [59] and speech. For what concerns the hearing process, sound waves are received by the human ears and transduced in nerve pulses into the brain, where they are processed and recognized [60]. In this work, in order to maintain the brain-inspired structure of the model, it has been chosen to exploit a pre-process technique which involves the modelling of the behaviour of the cochlea, which is the part of the ear responsible for the sound encoding: the Lyon's Auditory Model.

3.2.1 Lyon's Auditory Model

Proposed by R. Lyon in 1982 [61], the Lyon's model is a computational algorithm which expresses the most important cochlea's functions.
The human ear consists in three parts, outer, middle and inner ear. In the middle ear, the eardrum is responsible for transmitting the vibrations from the air in the outer ear to the fluids inside the cochlea in the inner part. The Basilar Membrane (BM) in the cochlea is the responsible for the reception of the sounds (Fig. 3.3): its local stiffness varies over the length of the cochlea [63], meaning that each part of the BM resonates with travelling waves of different frequencies (high frequencies near the base, low frequencies near the end). The mechanical energy is then converted in electrical signals by the hair cells on the BM, thanks to their *mechanotransduction* properties [64].
In order to model the BM's activity in the cochlea, the Lyon's auditory model is composed by different stages: filtering, detection and compression (Fig. 3.4).
Filtering stage The first stage consists in a series of notch filters (coupled with resonators), which separate the acoustic wave into its frequency components. This section models the BM's activity and the resonance of the travelling wave's frequency components

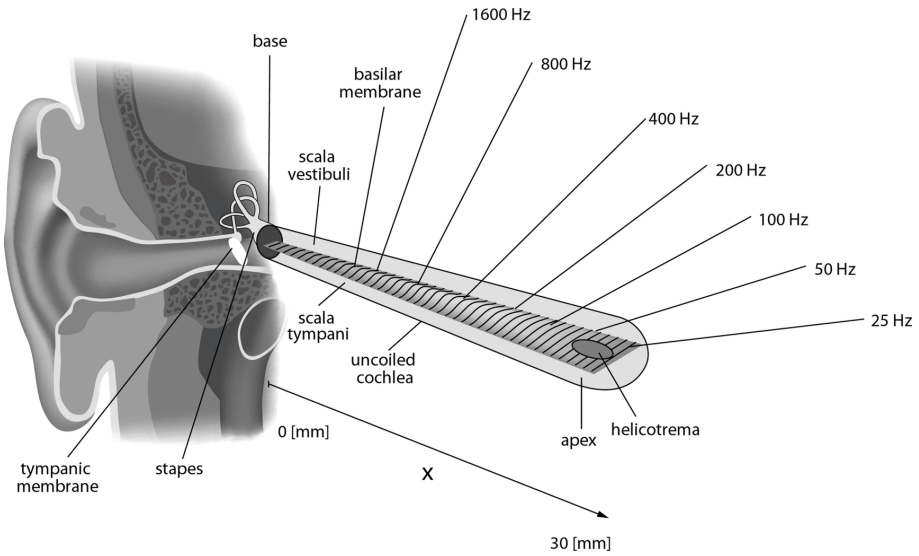


Figure 3.3: Representation of uncoiled cochlea and the frequency range as a function of the position x on the membrane. Reprinted from [62]

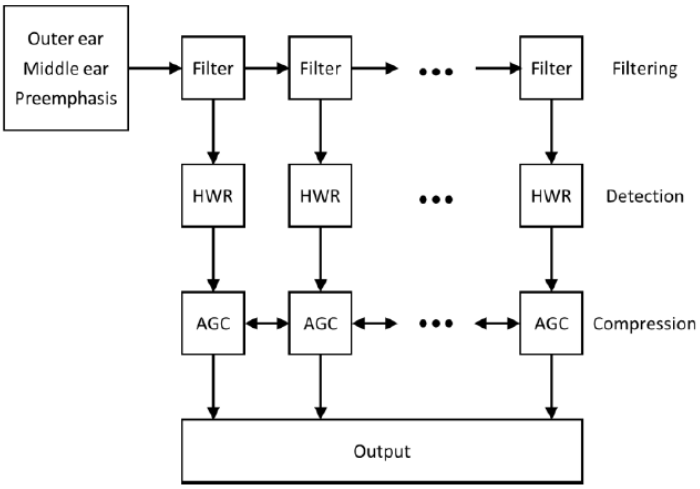


Figure 3.4: Block diagram representing the different stages of the Lyon's Auditory model.

with different parts of the cochlea.

Detection stage The outputs of the filtering stage are now processed by a series of Half-Wave Rectifiers (HWR), which simulate the hair cell detection of the BM's resonances. HWR detects the energy in the signal and cut out the negative half cycle of the input waveform.

Compression stage In the last stage, the output is compressed from a wide dynamic range signal to a more reasonable one, through Automatic Gain Control (AGC) closed-loop feedback. While the input acoustic wave's power could vary up to 12 orders of magnitude from threshold of hearing to threshold of pain, the output signals of the hair cells vary only of about 2 orders of magnitude, meaning that compression is needed in order to keep control of the output's dynamic range.

3.2.2 Cochleagram

The output of the Lyon's model paradigm can be graphically represented thanks to the so-called *Cochleagram*. Similarly to the spectrogram, the Cochleagram represents the intensity of the hair cells activity as it varies with time, which is strictly correlated to the frequency components of the audio signal. The Cochleagram is then exploited to transform the audio signals into voltage train spikes for the electrodes on the nanowire network: in this framework, each electrode acts as a neuron which could be either stimulated or not, depending on the hair cell activity.
In order to complete the processing of the audio inputs, the cochleagram must be interpolated, normalized, binarized, and then fed as input voltage spikes to the reservoir. In Fig. 3.5 it is depicted an example of a Cochleagram of a pronounced *nine*.

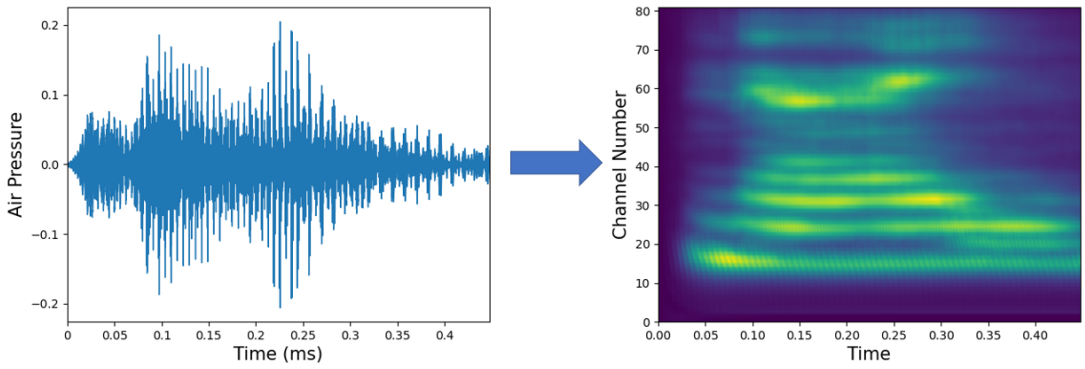


Figure 3.5: From sound wave to Cochleagram of a spoken digit *nine*. Each channel on the y-axis represent a narrow frequency window, covering the range of frequencies from 0 to 4 kHz (higher index channels refer to higher frequencies).

In this first step, the sound wave is fed to the Lyon's Model algorithm and depicted as a Cochleagram. As an example, it has been chosen a number of channels equal to 81, where each channel is related to an electrode on the reservoir, which therefore act as neurons receiving the hair cells signals.
Since the simulation takes place in a discrete number of timesteps, the Cochleagram must be discretized. Fig. 3.6 shows an interpolation of the diagram in 16 timesteps.
The one-dimensional piecewise linear interpolation simplifies the original spectrum, therefore choosing a high number of timesteps leads to a better and smoother graph. Nevertheless, using lower timesteps improve the computational cost and the time required for the simulation.

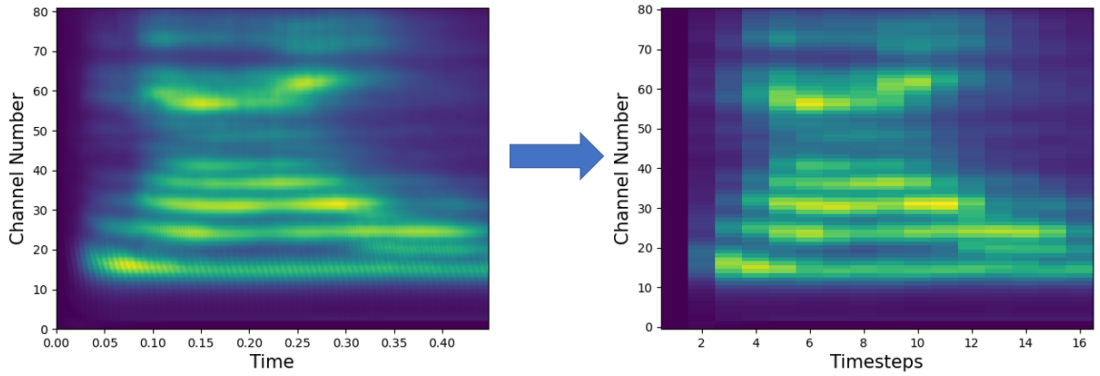


Figure 3.6: Cochleagram interpolation, 81 channels and 16 timesteps.

Lastly, the interpolated map is normalized and binarized, as shown in Fig. 3.7.

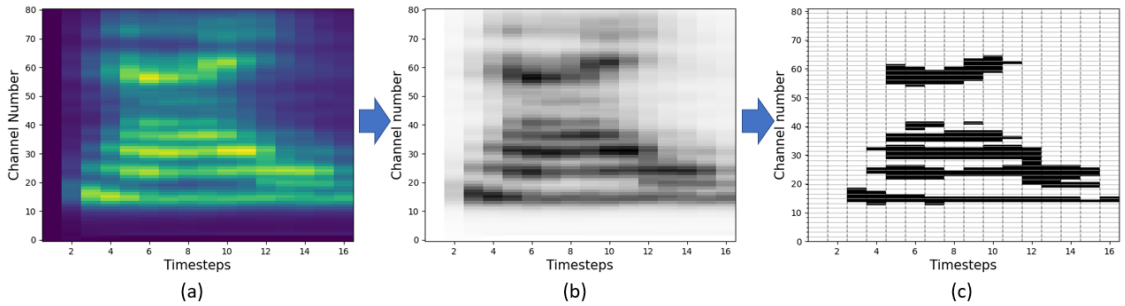


Figure 3.7: In the normalized map (b), the Cochleagram is scaled in the range $[0 \div 1]$. Depicted in grey scale, the darker is the pixel color the higher is its value. In the binarized map (c), black pixels equal to bit 1.

The binarization is retrieved by defining a threshold value between 0 and 1: if the pixel's value in the normalized map exceeds (is lower than) the threshold, it will be assigned a bit 1 (bit 0). In this work, it has been chosen a threshold equal to 0.6. Physically speaking, it means that when a hair cell receives a strong enough signal from the cochlea, it produces a electrical stimulus for the neuron (i.e electrode) which will be excited.

3.3 Reservoir

The pre-process output is fed as input for the Reservoir, which in this case is represented by the model of the Memristive nanowire network. Each electrode on the grid is assigned to a channel of the Cochleagram, meaning that there is a close correlation between the channels number and the reservoir size. The binarized map is exploited in order to define the voltage train spikes for the reservoir: at each timestep, if the pixel of a channel contains a bit 1, the corresponding electrode will be stimulated with a voltage spike. In the case of

a bit 0, no stimulation occurs leaving the electrode (neuron) relaxed. The input voltage spikes for stimulation are $5.0\,V/10ms$ pulses.

For what concerns the reservoir dynamics, it is of interest to show how the conductivity of the grid's memristive edges evolve with respect to the input map, as shown in Fig. 3.8. The evolution of the edge conductances non-linearly maps the inputs thanks to the dynamics of potentiation-depression of the memristive phenomena, allowing an enhancement of the spatio-temporal correlations between the inputs.

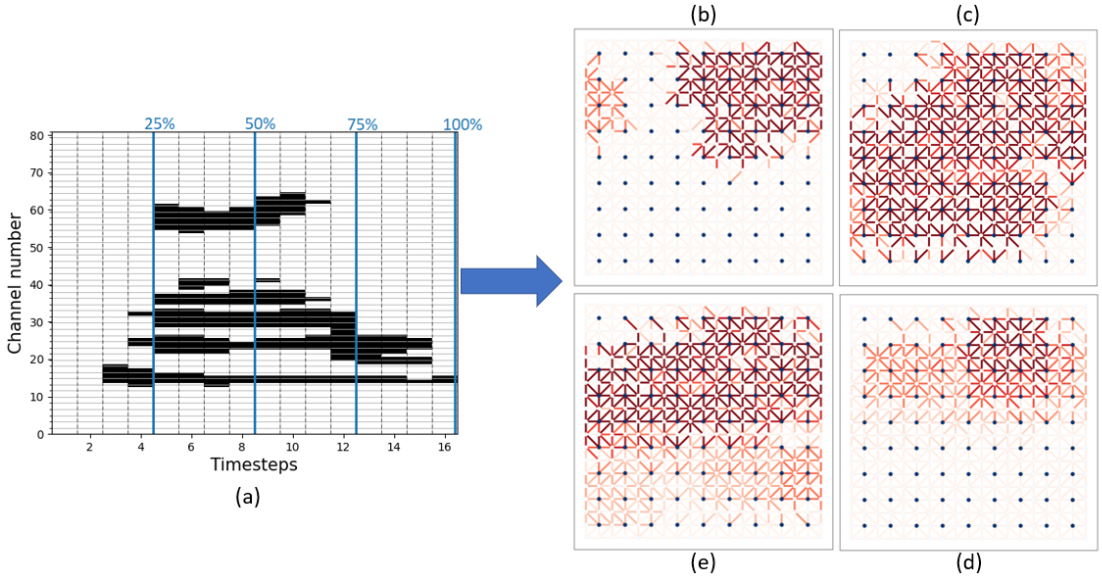


Figure 3.8: Evolution of the reservoir dynamics as a function of the input binarized map: example of spoken digit *nine* (a). The reservoir evolution is recorded at 25% (b), 50% (c), 75% (d), and 100% (e) of the input progess: the shade of red on the grid edges indicates the increase in conductance.

The reservoir states must now be recorded in such a way that it's possible to feed the output into a readout function for the classification task. For this purpose, it has been chosen to record the states in output voltage V_{out} vectors: in particular, it is applied a small DC voltage bias V_{read} (read pulse of $0.1\,V/5.5ms$) on one arbitrary electrode, and it is recorded the voltage drop V_{out} on a series resistance (R=82 Ω) in all the remaining electrodes of the grid. For this reason, for a number M of channels, it will be retrieved a vector of (M-1) output voltages. Following the example in Fig. 3.8, let us show the output voltage dynamics of six different channels, where the spoken digit *nine* colcheagram is used for the input voltage train spikes.

The difference in the evolution of the conductive paths lead to different V_{out} in each channel at the end of the simulation: in particular, it can be noticed that V_{out} changes in several orders of magnitude (Fig. 3.9, 3.10, 3.11), highlighting the complex dynamics in the reservoir. For these reason, the output is recorded in the 16th timestep, such that it can be reconstructed the input audio sample into a vector of (M-1) voltages.

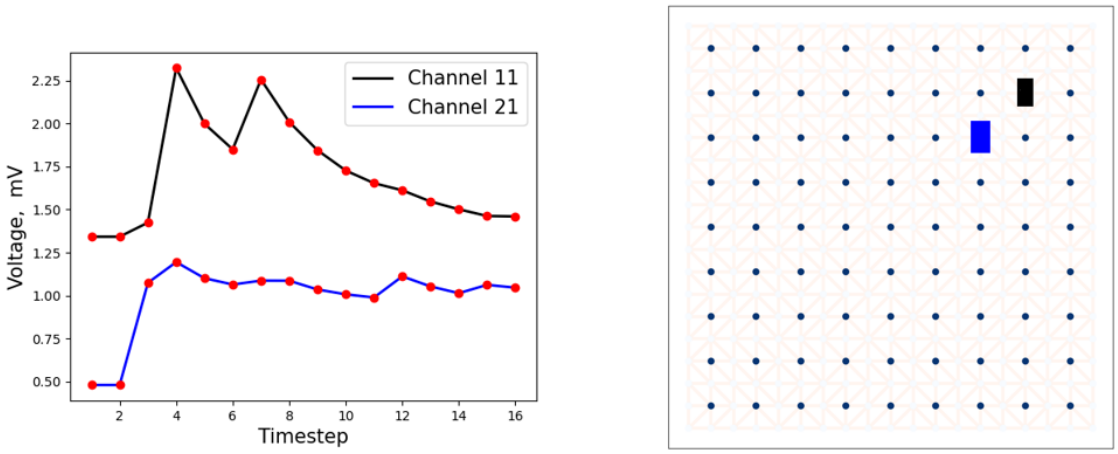


Figure 3.9: V_{out} dynamics of Channel 11 and Channel 21 with spoken digit *nine* as input binarized map

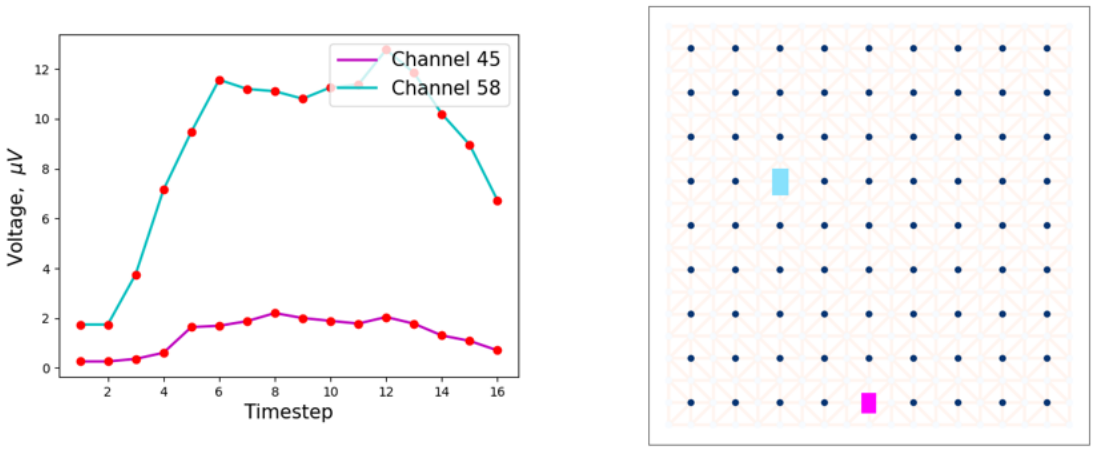


Figure 3.10: V_{out} dynamics of Channel 45 and Channel 58 with spoken digit *nine* as input binarized map

3.3.1 Virtual Nodes Processing

Memristive devices have a fading memory, related to the dissolution of the metal ion filaments when the electrical stimulus is cut off. Depending on the working conditions, on the experimental set-up and on the different materials used, the memory has a certain time decay constant: in the present work, in order to reproduce the experimental set-up, the model has been tuned such that the system loses the memory of the past after 4 timesteps, being a timestep equal to $250\mu s$. If the reservoir state is recorded only in one timestep, the far-history is lost: in order to address this problem, Virtual Nodes can be exploited. The virtual nodes concept has been first used for delayed systems [65][66], in which a delayed feedback is fed to a single non-linear node. The responses to the delayed signals

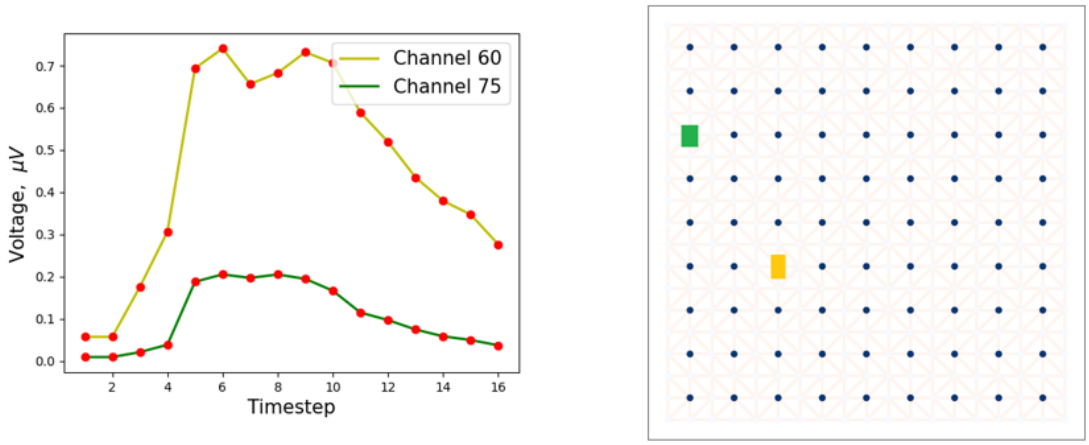


Figure 3.11: V_{out} dynamics of Channel 60 and Channel 75 with spoken digit *nine* as input binarized map

of the single physical node act as a chain of virtual nodes, which increase the size of the system without providing physical interconnections (Fig. 3.12).
This concept can be exploited with the memristive nanowire network as well: the reservoir state is recorded in different timesteps of the simulation, so that all the information can be recovered. For every virtual node, the size of the V_{out} vector increases of (M-1), which come from the evolution in time of the output voltages on the same physical electrodes: the reservoir size is therefore virtually increased, allowing for a better mapping of the input features (see section 3.5.2).

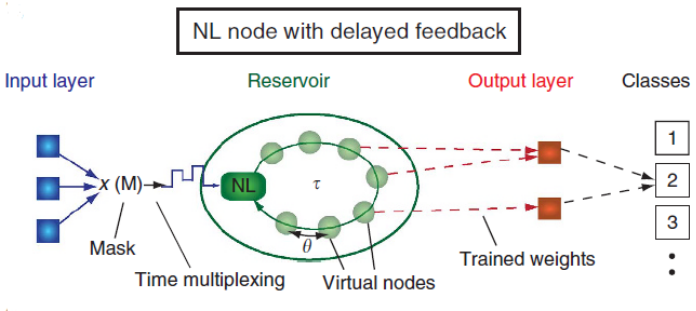


Figure 3.12: RC with nonlinear (NL) node and delayed feedback. A delay loop of duration τ is defined, and it is divided in N equidistant points named virtual nodes (distance equal to Θ in time), in which the reservoir states are recorded. Reprinted from [52]

3.4 Readout

The Readout Function is the last block for the RC paradigm, and the only part which needs to be trained: it is responsible for the final classification task, and in particular

it associates the V_{out} vectors of the reservoir to the actual digit classes. Thanks to the supervised machine learning algorithm, the readout can be trained on a portion of the dataset by defining the true labels of the inputs: after the training process, it is tested on the unseen part of the dataset and the final accuracy of classification is retrieved. Different readout functions and RC models are available: Echo State Networks (ECN) [67], Liquid State Machine (LSM) [68], Regression models, BackPropagation algorithms (BP) and so on. In this work, it has been studied in depth the Logistic Regression and a backpropagated Neural Network.

3.4.1 Artificial Neural Network

An Artificial Neural Network (ANN) is a computational algorithm which simulates an interconnected group of nodes, inspired by the connection of the neurons in the human brain. It's composed mainly of nodes and weighted connections, and it's structured in highly interconnected layers (Fig. 3.13)

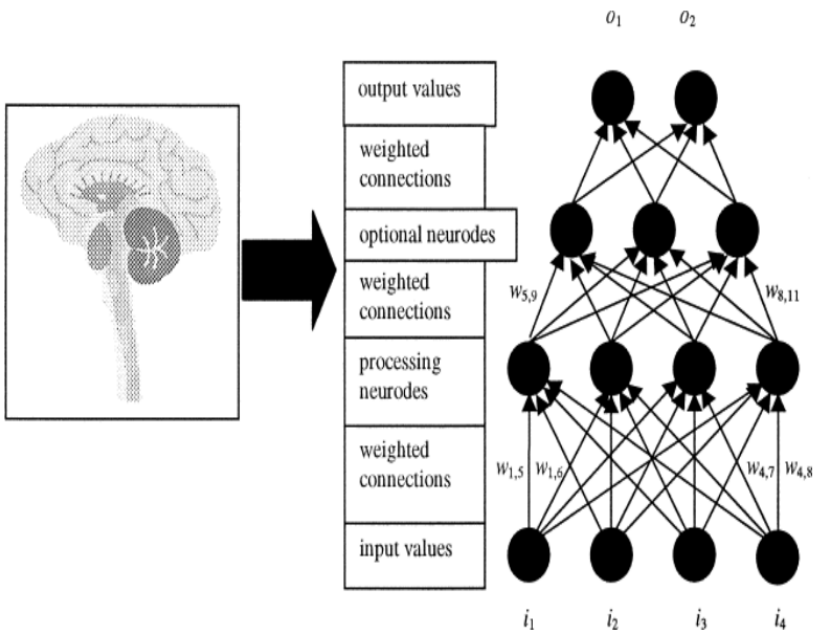


Figure 3.13: Structure of a typical ANN. Adapted and reprinted from [69]

The output V_{out} vector is stored inside the nodes values in the input layer. The activation of one layer brings the activation of the nodes in the following processing (or hidden) layers, thanks to the weighted connections: in particular, each node will store a number which is the weighted sum of all the nodes of the previous layer, whose weights are stored in the connections. Since the sum could retrieve any number, the activation of the neuron is restricted in the range from 0 to 1 thanks to the Activation Function. In this work, for example, it has been used a *sigmoid* function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

where σ is the node's activation number and x the weighted sum. Lastly, the output layer consists in 10 nodes defining the classes of the spoken digits (from digit *zero* to digit *nine*): the activated class will be the prediction of the input's class.
The learning process for the ANNs is related to the tuning of the weights, such that the inputs are correctly labelled in the output layer. A supervised learning algorithm is introduced through the Backpropagation [70]: in particular, it relies in minimizing the *categorical cross-entropy* loss function L(w)

$$L(w) = -\Sigma_{i=1}^{N} y_i(w) \cdot log(\hat{y}_i) \tag{3.2}$$

where: N is the output size, w the weights, $\hat{y}_i$ the model prediction and y_i the desired target. The sign minus assures that the loss function gets smaller when the target value is approached. The minimization of L(w) is done according to optimizer algorithm: in the following, let discuss the two main optimizer algorithms which are used in this work, *Adam* [71] and *RMSprop*.

Adam Optimizer The Adam algorithm is a stochastic gradient descent method which relies on adaptive tuning learning rates for each parameter to train:

$$w_t = w_{t-1} - \eta_t \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{3.3}$$

with η_t the learning rate coefficient, $\hat{m}$ and $\hat{v}$ bias-corrected first/second moment estimators, and ϵ a constant for numerical stability. A too small learning rate would require too many steps to reach the minimum point; on the contrary, a big learning rate coefficient would lead to divergency.

RMSprop Optimizer RMSpropr algorithm is an unpublished work first developed by Professor Geoff Hinton in his lecture notes at the University of Toronto. The weight update follows the following formula:

$$w_t = w_{t-1} - \frac{\eta}{\sqrt{E[g^2]_t}} \nabla L_i(w) \tag{3.4}$$

where $\sqrt{E[g^2]_t}$ is the moving average of squared gradients g. The normalization balances out the step size, decreasing (increasing) the step for large (small) gradients.

3.4.2 Logistic Regression

The Regression model is a statistical algorithm used for predictive analysis. Regression is a process for which it can be estimated the relationship between dependent and independent variables: for example, the most easy regression analysis is the linear regression, for which the data is fitted by a line.
Used primarily for dichotomous variables, the Logistic Regression (LR) can be extended for multiple classes as well [72], making it suitable for the speech recognition task. In LR, the data are fitted with a sigmoid function: similarly to ANNs, the weighted sum of the input features is calculated and then transformed by the sigmoid function in the range 0-1. The final output tells the probability of the input of belonging to particular class.

3.5 Results

Once the audio samples are correctly pre-processed, sent as input for the reservoir and mapped into reservoir states (i.e output voltages on the electrodes), it is important to investigate the performance of the system in terms of test accuracy: the speech recognition task, in fact, is a classification task, in which the readout is trained such that it can recognize the input and classify it. In this framework, the non-linear transformation of the reservoir should help extrapolating spatio-temporal correlations in the inputs, leading to a gain in the classification accuracy.
For the speech recognition task, it has been exploited a free-spoken digit dataset (Tensorflow) [73]. It consists in 3.000 audio samples in which 6 different speakers (George, Jackson, Lucas, Nicolas, Theo, Yweweler) pronounce 10 digits from 0 to 9. The speakers are recorded 50 different times for each digit, resulting in 300 audio samples for each digit and 3.000 overall files. The readout function is trained on 2.700 audio files, while it is tested in the remaining 300 audio samples (30 audio for each digit) through the k-fold cross validation technique, with k=10. In the following, it is reported some examples of binarized 81x16 maps with the Lyon Model approach on the TensorFlow dataset.

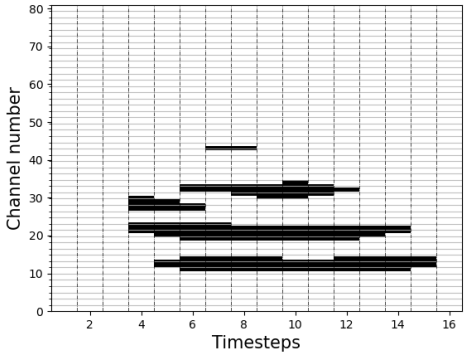


Figure 3.14: '4' spoken digit, Theo

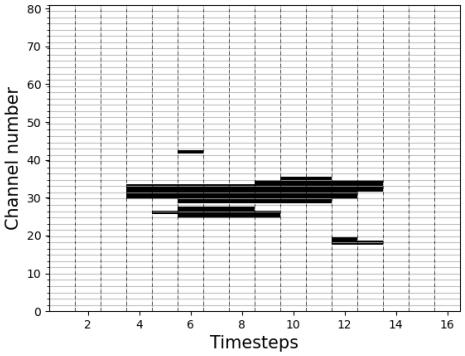


Figure 3.15: '4' spoken digit, Jackson

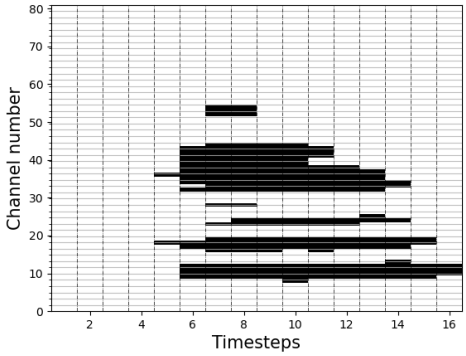


Figure 3.16: '9' spoken digit, Lucas

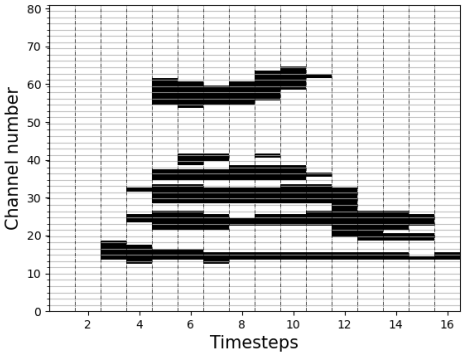


Figure 3.17: '9' spoken digit, George

3.5.1 No virtual nodes

At first, no virtual nodes are used during the recording of the reservoir states: it means that the electrodes output voltages are recorded only in the last timestep of the simulation, and used as input for the readout function. An example is given in Fig. 3.18.

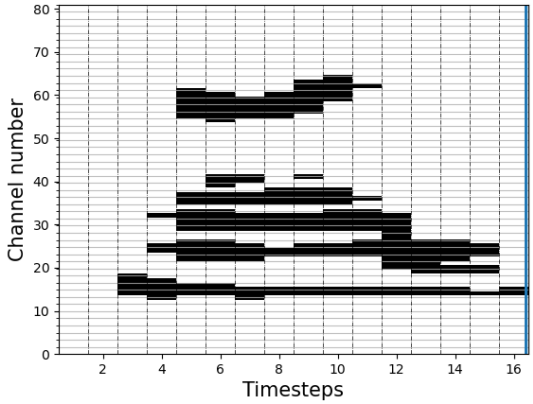


Figure 3.18: 81x16 map, spoken digit '9'. The blue vertical line represents the simulation timestep in which the reservoir state is recorded.

In the case of 81x16 binarized maps (81 channels/electrodes, 16 timesteps), the individual audio samples are transformed into 80x1 vectors which are fed to the readout function for training/testing. The readout layer is therefore a 80x10 network, where the 10 outputs are representing the different classes (i.e digits to be recognized). The reported accuracies represent the mean value of 10 different cross-fold validation set-ups of a logistic regression model as readout layer.
The first important result regards the mean test accuracy of the model with and without the reservoir (fig. 3.19).

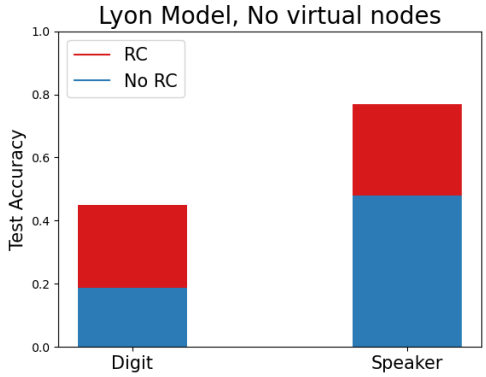


Figure 3.19: Test accuracy for both Digit and Speaker recognition, with and without reservoir (RC)

The histogram in Fig. 3.19 represents the obtained accuracy for both digit and speaker recognition. In particular, the digit recognition task requires the correct classification of the spoken digit in the audio file, while the speaker recognition task requires the correct classification of the person which is pronouncing the different digits. For what concerns the accuracy without the reservoir, it is retrieved by feeding the binarized maps directly to the readout function, without being processed by the reservoir itself. The results are comparable thanks to the same number of parameters which define the input.
It can be observed that there is a considerable gain in the accuracy with the reservoir in both recognition tasks (red bins). The following Table 3.1 summarizes the obtained values.

Table 3.1

	No RC	*RC*
Digit	20%	45%
Speaker	48.2%	77.2%

In the field of machine learning, it is important to report the confusion matrices as well: they allow for a visual representation of the performance of a supervised algorithm, and in particular it shows if there are some biases between the different classes. In Fig. 3.20 and Fig. 3.21 it is depicted respectively the confusion matrices of digit and speaker recognition with the reservoir. The vertical axis represents the true class, while the horizontal axis

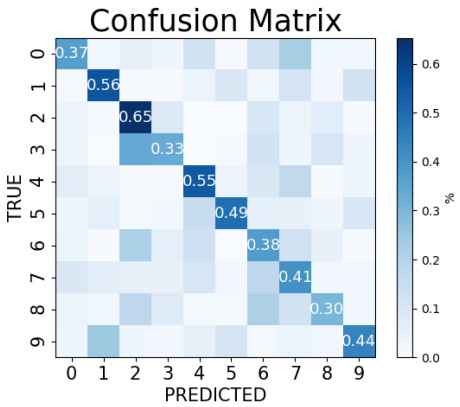


Figure 3.20: Confusion matrix, Digit recognition

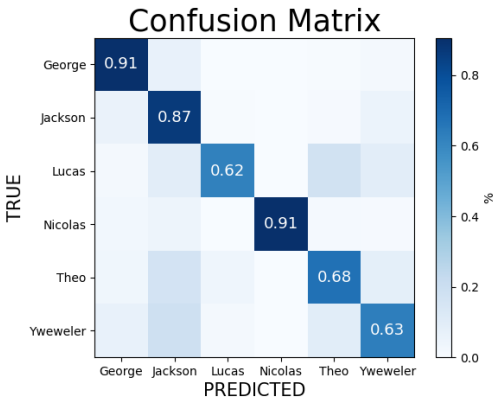
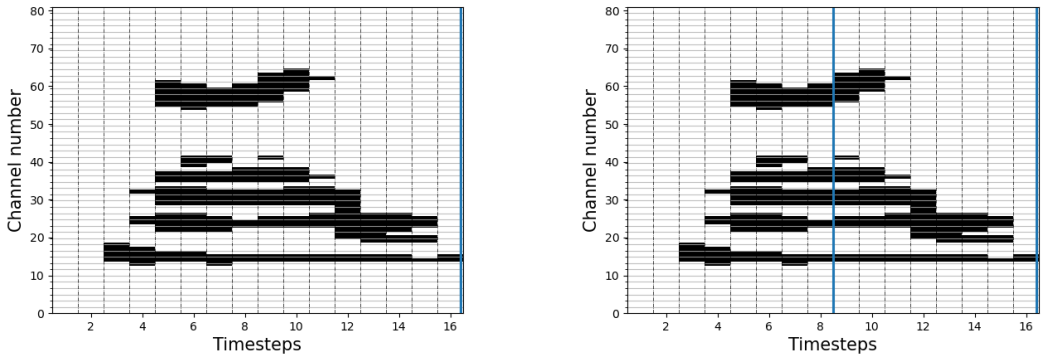


Figure 3.21: Confusion matrix, Speaker recognition

represents the class predicted by the algorithm, therefore the values in the main diagonal represent the percentage of correctly predicted input classes. For example, a value of 0.37 on the upper-left square of the matrix in Fig. 3.20 means that 37% of the audio files in which it is pronounced the digit 0 are correctly classified as 0.
It is possible to notice that some classes are more likely to be predicted, for example the spoken digit 2 and the speakers *George* and *Nicolas*.

3.5.2 Virtual nodes effect

Although the Reservoir is capable of increasing considerably the performance of the supervised learning process, the accuracies are still low (see Table 3.1), specially for digit recognition: the reason is related to the fact that the far-history (i.e first timesteps) of the simulation is lost, thus retrieving partial information of the audio inputs. The best way to reconstruct all the reservoir dynamics along all the timesteps of simulation and retrieve the lost information is to use the virtual nodes (Fig. 3.22).



(a) No Virtual nodes

(b) N=2 Virtual nodes

(c) N=4 Virtual nodes

(d) N=8 Virtual nodes

Figure 3.22: 81x16 map, spoken digit '9'. The virtual nodes (blue vertical lines) determine the timesteps in which the simulation is recorded. The number of timesteps between two virtual nodes is $\frac{16}{N}$.

The number of virtual nodes (N) determines how many times the reservoir state is recorded. It is therefore expected that the higher is N, the easier is the recognition and the higher the accuracy. However, it must be taken into account that for each virtual node it corresponds a number $(M-1)$ of voltage values, where M is the number of electrodes on the network (i.e channels in the Lyon Model, in this case M=81). In other words, the individual audio samples are now transformed into (80N)x1 vectors, leading to a (80N)x10 network as readout function. The trend of the test accuracy with respect to the number of virtual nodes

is reported in Fig. 3.23.

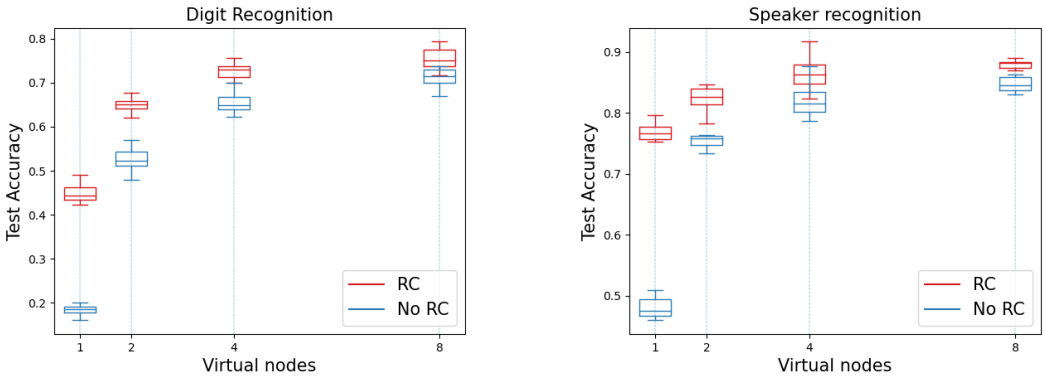


Figure 3.23: Accuracy vs Virtual node numbers for Digit recognition (a) and Speaker Recognition (b), in *boxplot* representation. The whiskers represent the minimum and the maximum values, the box lies in the Interquartile Range which is the distance between the first Quartile (25% of the distribution) and the third Quartile (75% of the distribution), and the middle line is the Median of the distribution.

Since the accuracies are collected in 10 different tests on the same dataset with the cross-validation technique, the results has been reported as *boxplots*: it is a statistical method to depict graphically the distribution of numerical data through their quartiles, such that one box represents 10 accuracies with 10 different test datasets. It can be observed that the accuracy increases with increasing virtual nodes, with a saturation for high N. Moreover, reproducing the results by means of histograms (Fig. 3.26), it is possible to better appreciate the fact that the gain of the Reservoir decreases with increasing N. Lastly, let us report the confusion matrices in the best case, with 8 Virtual nodes (Fig. 3.24, 3.25)

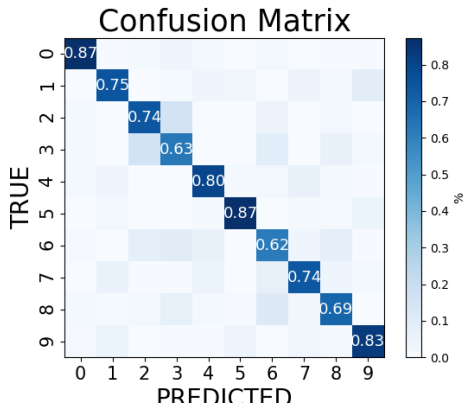


Figure 3.24: Digit recognition, N=8

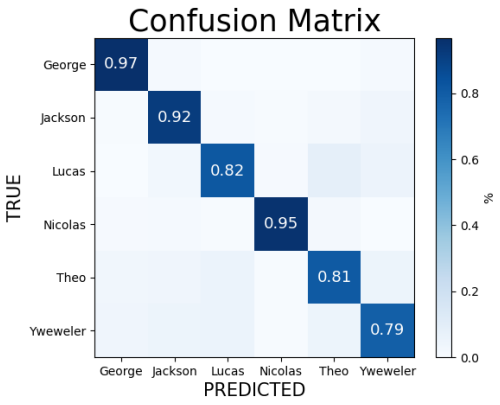
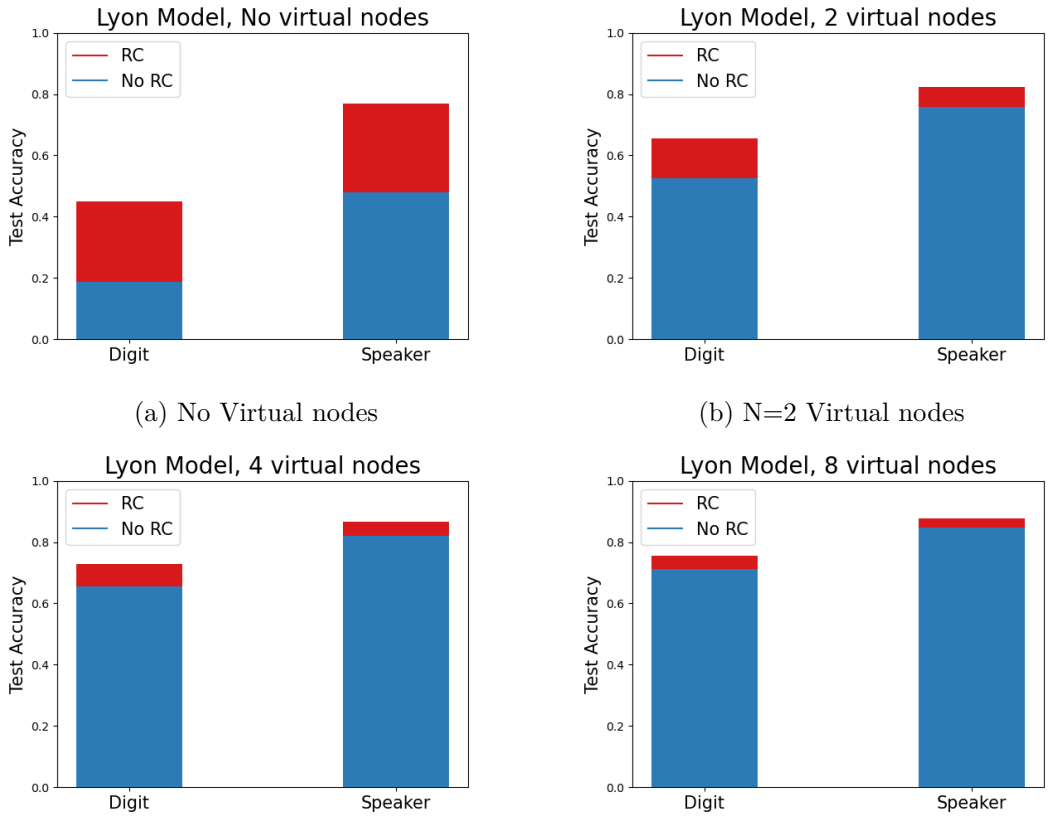


Figure 3.25: Speaker recognition, N=8



(a) No Virtual nodes

(b) N=2 Virtual nodes

(c) N=4 Virtual nodes

(d) N=8 Virtual nodes

Figure 3.26: Histograms representing the gain of the Reservoir Computing in terms of accuracy, as a function of the number of virtual nodes.

3.5.3 Effect of Network size

Until now, it has been taken into account a number of electrodes/channel (M) fixed at 81. In the simulation, it is possible to change M as much as we like, thus exploring much bigger number of electrodes and net size. Nevertheless, it is important to keep in mind that the model serves to understand how to set up the experiment in the best conditions and with the best performance: the lower is the size of the electrode grid, the easier would be the experimental setup, allowing for better reproducibility of the results. Most importantly, reducing the network size allows for easier and scaled down hardware devices, along with lower computational costs in terms of simulation. In this section, it is wanted to explore the influence of the grid size on the overall accuracy of the model.

The number of electrodes used not only influences the physical size of the network, but it also influences strongly the pre-processing. In the Lyon Model approach, in fact, every electrode is related to a single channel (i.e neuron) on the Cochleagram: the lower is M, the poorer would be the discretization, and the more difficult will be the recognition of the

spoken digits. The effect is clear in the following figures, where the same audio file (spoken digit ′9′) is interpolated using different numbers of electrodes.

Effect of electrode number on Cochleagrams

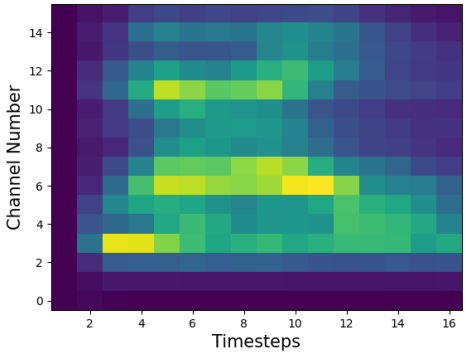


Figure 3.27: Interpolated map, 16x16

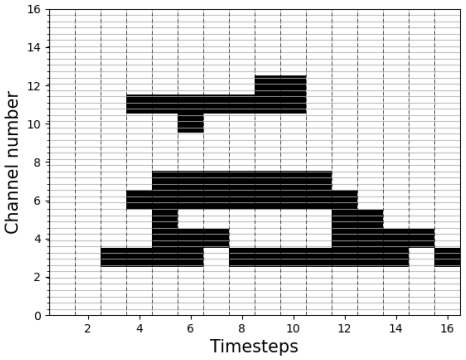


Figure 3.28: Binarized map, 16x16

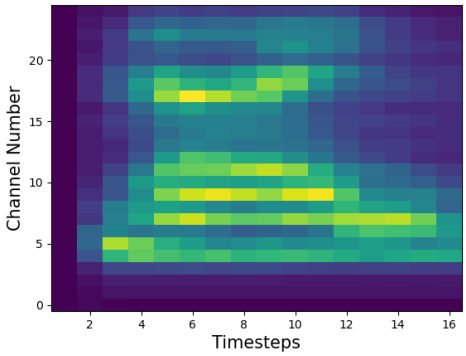


Figure 3.29: Interpolated map, 25x16

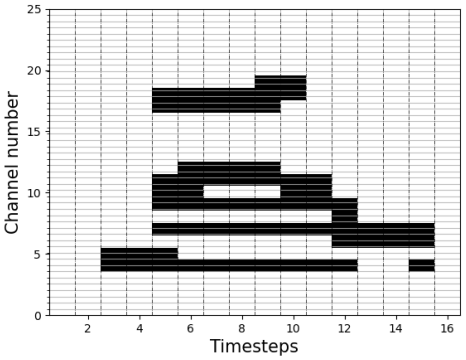


Figure 3.30: Binarized map, 25x16

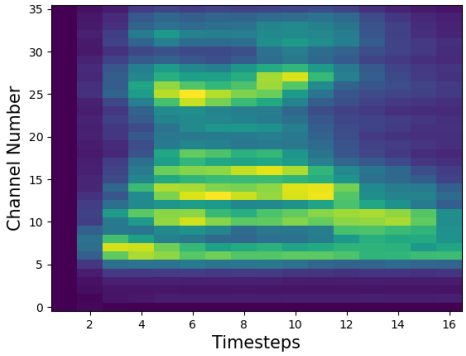


Figure 3.31: Interpolated map, 36x16

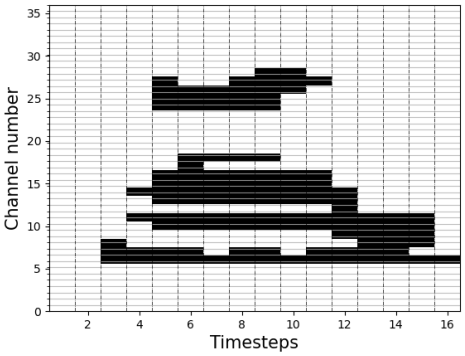


Figure 3.32: Binarized map, 36x16

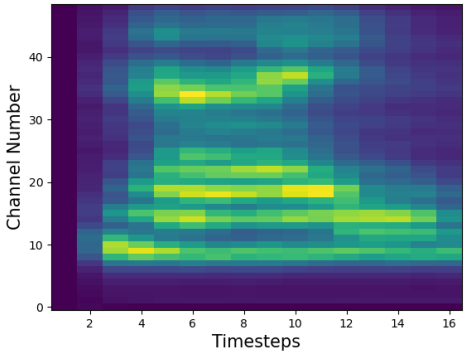


Figure 3.33: Interpolated map, 49x16

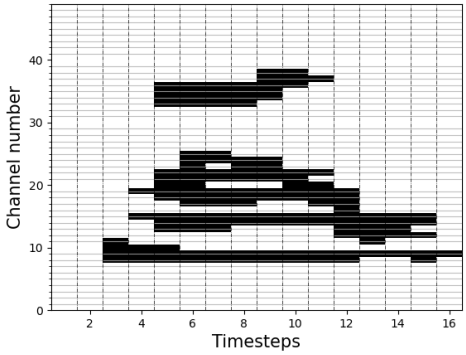


Figure 3.34: Binarized map, 49x16

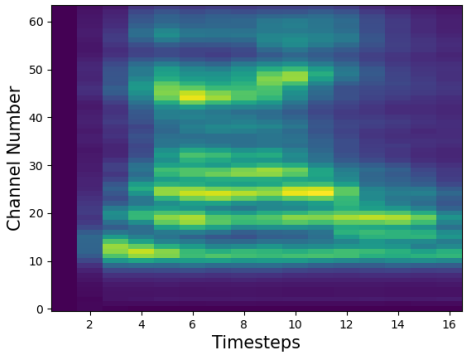


Figure 3.35: Interpolated map, 64x16

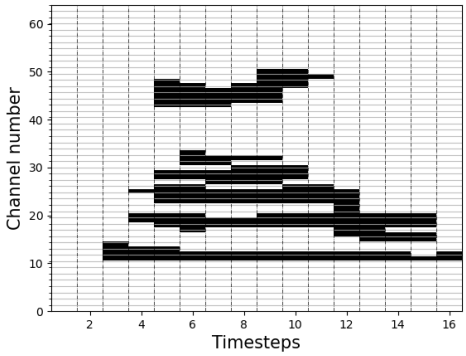


Figure 3.36: Binarized map, 64x16

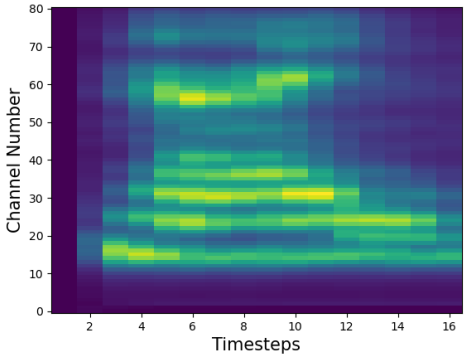


Figure 3.37: Interpolated map, 81x16

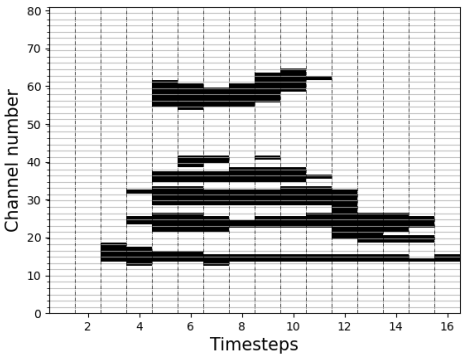


Figure 3.38: Binarized map, 81x16

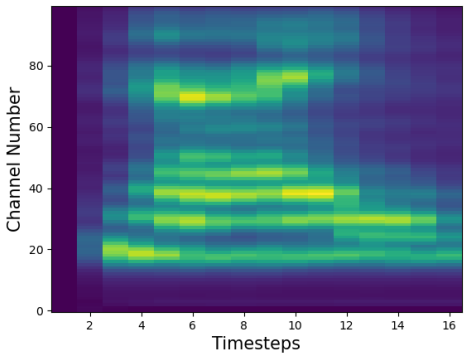


Figure 3.39: Interpolated map, 100x16

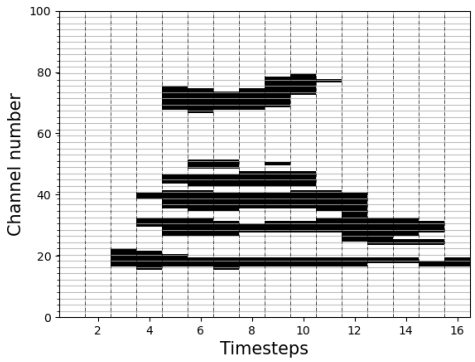


Figure 3.40: Binarized map, 100x16

It has been chosen values of M so that the electrode grids are square, with side equal to $\sqrt{M}$, in order to obtain more compact and symmetric structures.

Let us now study the performance of the model in terms of accuracy. In Fig. 3.41, 3.42 it is reported the trend of the accuracy as a function of the grid side at fixed virtual node N=8 (for further details, the box plots regarding the accuracy as a function of the virtual nodes for every value of M are reported in appendix A).

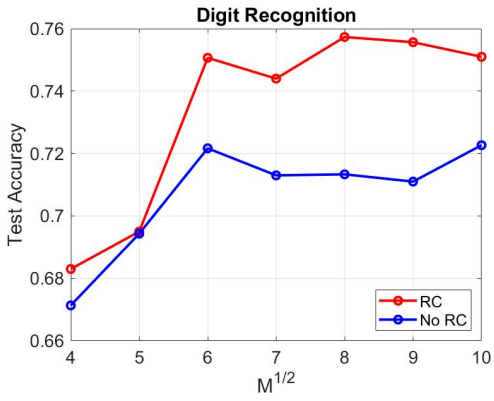


Figure 3.41: Mean Test accuracy as a function of the network size, Digit Recognition case

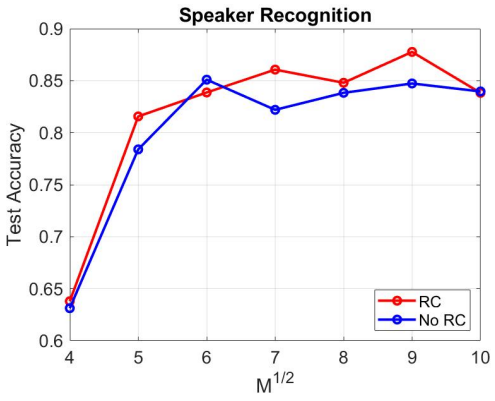


Figure 3.42: Mean Test accuracy as a function of the network size, Speaker Recognition case

The trend is similar in both Digit and Speaker recognition: the accuracy is lower for small values of M, then it progressively increases and saturates for higher values. It is also important to notice that in both RC and No RC cases there is a loss in performance at low sizes, meaning that the reduction of the network size influences the pre-processing, increasing significantly the loss of information.

3.5.4 Effect of the Readout

Let us now explore different readouts and their weight on the final result of these simulations. In particular, it has been used two different readout functions: an Artificial Neural Network (ANN) and a Logistic Regression Model.
The ANN used is one hidden layer of neurons fully connected (dense layer) with the input and output layers. The following code has been used for simulating the ANN, by exploiting the *tensorflow.keras* library.

```
[...]

model = Sequential()
model.add(Dense(10, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='RMSprop',
                                                metrics=['accuracy'])
history = model.fit(train_in, train_out, validation_data=(test_in,
                            test_out,),epochs=400, batch_size=250)

[...]
```

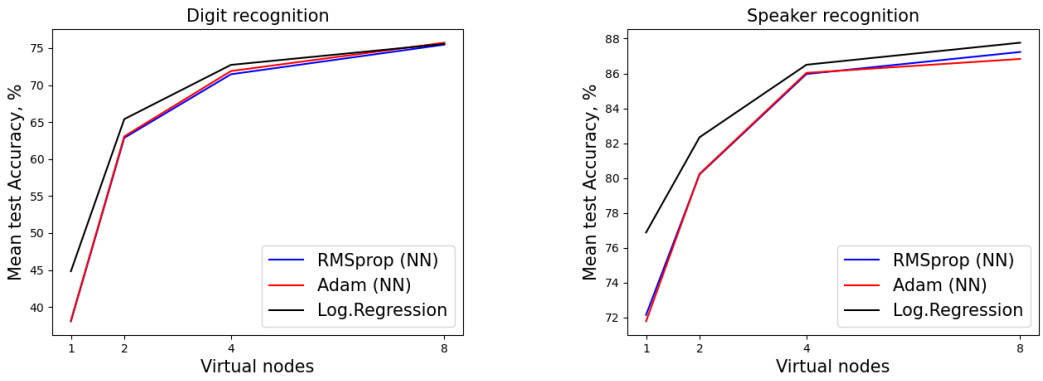
The Sequential model is used to stack a series of layer and build the readout function layer by layer: in this case, it has been added only one Dense layer. *Train_in* and *test_in* are the portion of the dataset used for train and test, while *train_out* and *test_out* are the classes of the train/test datasets. In the following, two different optimizers will be used: RMSprop and Adam.
For what concerns the Logistic Regression model, it has been used the *scikit-learn* library.

```
[...]

crossval = StratifiedKFold(n_splits=10)
for train_index, test_index in crossval.split(total_train_in, total_train_out):
   model=LogisticRegression(max_iter=600)
   x_train_fold, x_test_fold = total_train_in[train_index],
                                total_train_in[test_index]
   y_train_fold, y_test_fold = total_train_out[train_index],
                               total_train_out[test_index]
   model.fit(x_train_fold, y_train_fold)
   y_pred=model.predict(x_test_fold)

[...]
```

In this case, it is shown the way to implement the cross-validation technique as well, through the *StratifiedKFold* line. In Fig. 3.43 it is depicted the difference, in terms of test accuracy, in exploiting different readout functions.



(a) Digit Recognition

(b) Speaker Recognition

Figure 3.43: Test accuracy as a function of the virtual nodes, for different readout functions. Adam and RMSprop refer to a one-layer NN readout.

The Logistic regression model outperforms the one-layer NN in the speaker recognition case. For what concerns the digit recognition, Log. regr. is still better for small values of N, while for N=8 the readout functions converge to approximately equal accuracy values.

3.5.5 Real Time recognition

In all the previous results, the digit recognition was performed by taking into account the processing of the whole audio samples, meaning that the digit is classified after being spoken. In this last result's section, it is shown that it is possible to obtain a real-time recognition, in which the digit is recognized while it's being spoken. The way to proceed is similar to the virtual nodes case, as it is shown in Fig. 3.44.

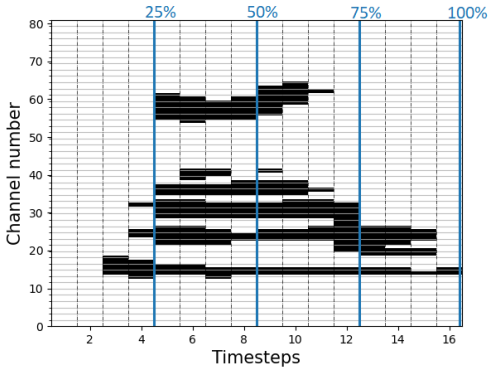


Figure 3.44: Input progression for real-time recognition

In this case, partial inputs are used for classification. The readout is trained and tested four different times, taking into account the progression of the simulation: for 25% of the input the reservoir state is saved at the fourth timestep, for 50% at the fourth and eight

timesteps and so on. The main difference is that, contrary to the virtual nodes case, it is not needed to wait for the whole audio sample to be processed because the reservoir states used for training the readout are progressively stored as the simulation proceeds. Fig. 3.45 shows the trend of the test accuracy in the real-time recognition case.

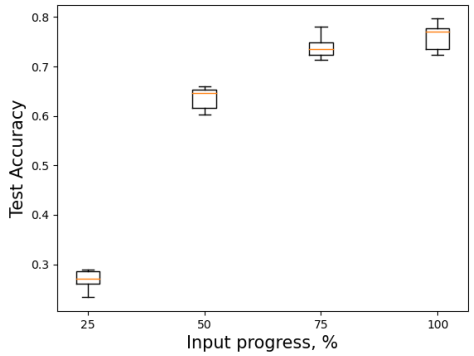


Figure 3.45: Test accuracy as a function of the input progress. The Median of the distributions are represented by the middle orange lines. The minimum and maximum values, and the Interquartile range, are reported as black whiskers and boxes

The accuracy increases as the input progresses. It can be noticed that the model achieves already great accuracies at 75% of the input progress: thinking about human brain, it's like starting to recognize a spoken *'nine'* when the word is not fully pronounced yet.

3.6 Discussions

The results which have been achieved in the simulations prove that the Reservoir Computing paradigm introduces a gain in the accuracy of recognition of spoken digits: in particular, by looking at the first result without virtual nodes, the accuracy is doubled. The reason behind this phenomena lies in the dynamics of the memristive reservoir: the capability of transforming non-linearly the inputs, along with a intrinsic non-volatile memory of the memristive processes, make so that the reservoir is suitable for extrapolating spatio-temporal correlations and allowing for a easier recognition of the audio samples.
Nevertheless, it is important to notice that, with virtual nodes, the gain of the reservoir decreases, as it is shown in Table 3.2.

Table 3.2: Word Success Rate gain, with M=81

Virtual nodes	*No RC*	*RC*	*Gain*
1	20%	45%	25%
2	52.6%	65.4%	12.8%
4	65.5%	72.7%	7.2%
8	71.1%	75.7%	4.4%

This trend suggests that, in the model, the most important part for the final accuracy is

the pre-process: the better is the pre-processing at distinguishing the different inputs, the lower would be the gain of the reservoir. Moreover, it is observed that the performance varies with respect to the different digits or speakers (a visual summary is provided in Fig. 3.46), meaning that the final accuracy depends strongly on the inputs (i.e on the specific dataset) as well. As a matter of fact, the absolute value of the final accuracy should not be a figure of merit of the reservoir by itself.

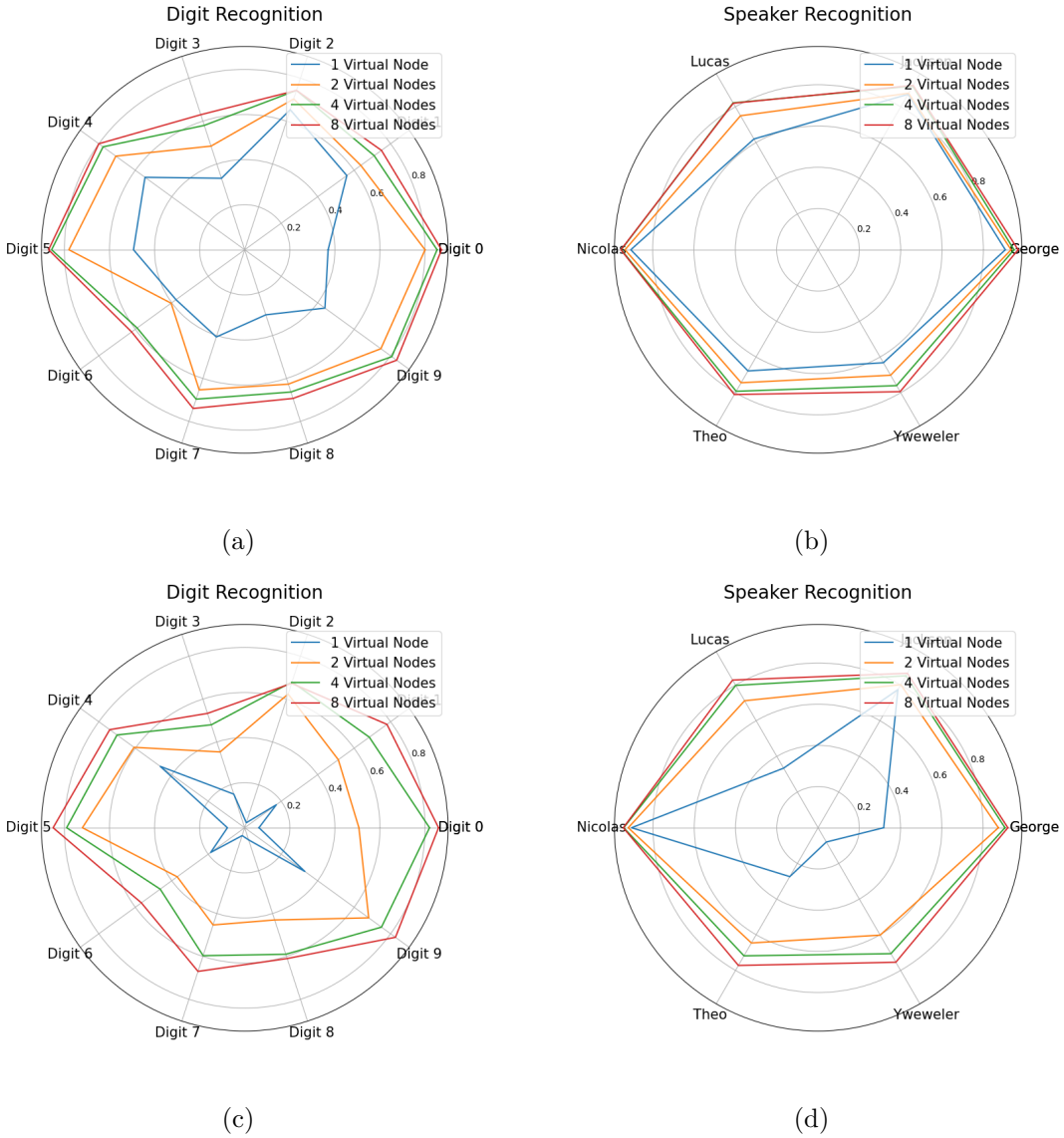
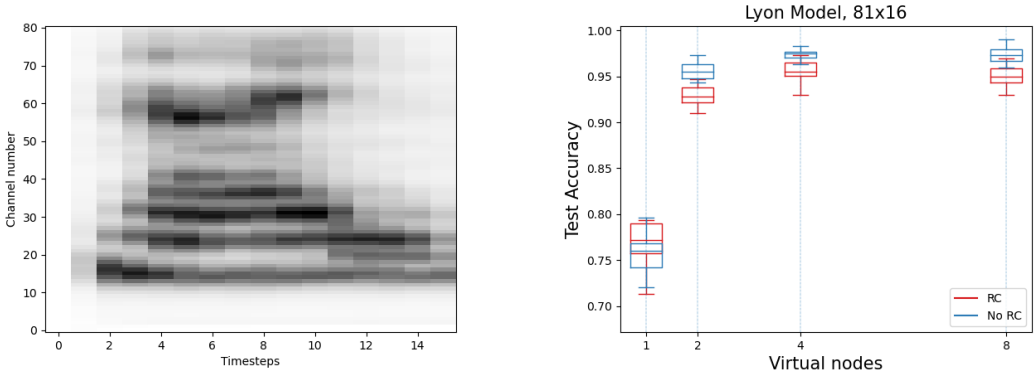


Figure 3.46: Charts representing the accuracy for Digit and Speaker recognition with (a,b) and without (c,d) RC, with different Virtual Nodes (blue: 1 VN, orange: 2 VN, green: 4 VN, red: 8 VN).

For what concerns the effect of the network size, the importance is related to the hardware implementation: the lower is the number of electrodes used on the reservoir, the easier is the experiment. A more complex and bigger structure would allow for better performances, but in the reported results it is shown that the accuracy saturates after reaching a certain value of the network size (Fig. 3.41), leading to a facilitated experimental set-up without any lack of performance.

In all the simulations which have been performed, it has been used binarized maps for the input train voltage spikes: by doing so, a lot of the information from the original Cochleagram is lost due to the introduced binarization through a certain threshold. In order to prevent the information loss, it is possible to use the normalized maps as inputs for the reservoir, but in this case the map's pixels do not represent the firing rate of the neurons: instead, since the map is composed of real numbers between 0 and 1, the voltage spikes are now modulated in amplitude in the range [0,5]V (i.e the voltage spike is multiplied by the pixel's value). In Fig. 3.47 it is shown the results in term of accuracy for the normalized maps case.



(a) (b)

Figure 3.47: Normalized Cochleagram of spoken digit '9' (a) and test accuracy as a function of the virtual nodes (b)

It is possible to notice that the accuracies are much higher in this case, but there is no gain using the RC. The reason behind this phenomenon is that there is no relaxation of the memristors inside the reservoir, since it is always potentiated with modulated voltage amplitudes, leading to a possible saturation of the conductances. This result suggests that the system acts efficiently only when the neurons are either stimulated or not (i.e using 0 or 1 as input), confirming its brain-inspired nature.

3.7 Literature comparisons

At last, it is reported a comparison between the literature on Reservoir Computing for speech recognition task. It is of interest to make a parallelism between different proposed self-assembly network structures for the reservoir, and to understand the performance in terms of accuracy on the recognition of spoken digits. Moreover, by looking at the literature, it is important to notice that the final accuracy depends strongly on the dataset which has been used.

(1) In a recent work of Gimzewski et al. [74], a Ag_2Se NW network has been used as reservoir with support-vector machines (SVM) as readout function [75], and Fast-Fourier Transform as pre-processing technique. The network, in this case, is referred as an Atomic Switch Network (ASN), in which the synaptic behaviour is induced by redox reactions occurring at the nanowire junctions during voltage-induced switching. The work showed similar gains with the RC paradigm with respect to our simulations, and it reached an accuracy of $\approx 89.5\%$ on the same Free Spoken Digit dataset (Tensorflow) when the recognition is performed with one speaker only out of all six speakers. In our present work it is reported the digit recognition task with all 6 speakers (75.7% accuracy), but simulations with one speaker lead to similar results ($\approx 89.2\%$) with respect to Gimzewski et al's paper.

(2) Moon et al [66] performed experimentally spoken digit recognition using as reservoir a 32×32 Cross-bar WO_x memristor array, exploiting the Lyon's Auditory model as pre-process and an ANN with RMSprop as a readout function; the input Cochleagrams are characterized by 50 channels, with 40 timesteps. In their work, it has been achieved an accuracy of 99.2% on digit recognition task, with a gain of 0.6% with respect to a Convolutional Neural Network (CNN) system without memristive reservoir. In this case, the dataset which has been used is the TI 46-Word from NIST.

(3) Another work based on the TI 46-Word dataset is from Grollier et al [76]. In this case, it has been developed a model based on a single non-linear magnetic oscillator as reservoir, in which non-linear transformations occur in time rather than in space thanks to time-multiplexing of the input sequences. As a readout instead, a simple Linear Regression model is used. The work deals with different pre-processing techniques, from simple spectrograms to Mel-frequency cepstral coefficients (MFCC) and Lyon's Auditory Model. For what concerns the results with the Lyon's pre-processing technique, 99.6% of accuracy is achieved with the NIST dataset. A more complicated dataset is used too (the Aurora-2), in which 63.24% accuracy is achieved.

(4) Lastly, rich dynamic and nonlinear electronic properties for Reservoir Computing has been demonstrated in a sulfonated polyaniline (SPAN) organic electrochemical network device (OEND) [77]. Polyaniline exhibits hopping process between localized conductive islands for charge transfer, along with ionic transport: the mix of ionic and electron transport give raise to strong non-linearities and hysteresis loops in the I-V characteristics making SPAN OEND suitable as Reservoirs. Speech recognition has been achieved on the Tensorflow dataset, with an accuracy of 66% on digit recognition with one speaker only out of six.

For simplicity, a table is provided showing the comparison between the present work and the literature, reporting both the accuracy in the digit recognition task (Digit Accuracy) and the accuracy in the speaker recognition task (Speaker Accuracy).

Work	Reservoir	Readout	Pre-Process	Dataset	Digit Accuracy	Speaker Accuracy
Present work	Ag NW network	Logistic Regression	Lyon Model	Tensorflow	75.7%	87.6%
				Tensorflow, 1 speaker	89.2%	
(1)	Ag2Se NW Network	support-vector machines (SVM)	Fast-Fourier Transform	Tensorflow		87.3%*
				Tensorflow, 1 speaker	89.5%*	
(2)	single non-linear magnetic oscillator	Linear regression	Lyon Model	Aurora-2	63.24%	
				TI-46	99.6%	
(3)	Cross-bar WO_x memristors	RMSprop (ANN)	Lyon Model	TI-46	99.2%*	
(4)	sulfonated polyaniline (SPAN) organic electrochemical network device (OEND)	Ridge Regression	Lyon Model	Tensorflow		60%*
				Tensorflow, 1 speaker	66%*	

Figure 3.48: Comparison between the present work and literature. The reported accuracies are the result of the RC paradigms with the reservoirs used to process the input data. The red asterisks refer to experimental results in hardware.

By looking at the results, it is clear that, in order to give a good comparison between the reservoir performance, the pre-process and (most of all) the dataset used must be coincident. Generally speaking, the final accuracies are similar for the same dataset with different reservoirs: it is important, on the other hand, that the RC paradigm brings an accuracy gain in the classification. A more detailed discussion is reported in section 3.6.

Chapter 4

Conclusions and Future Perspectives

With this present work, it has been demonstrated the possibility to exploit the memristive Ag nanowire network as reservoir in the Reservoir Computing framework: thanks to the reweighting and rewiring properties, the network is capable of reproducing homo- and hetero-synaptic plasticity, being therefore suitable for mapping spatio-temporal input spikes into a higher dimensional space similarly to a neural circuit.
In order to solve the complex Speech Recognition Task, a RC paradigm has been developed, composed by: a brain-inspired pre-process technique for the processing of the audio signals (Lyon's model), a computational model for the memristive reservoir dynamics and a simple readout function for the classification. Furthermore, by exploiting the concept of the virtual nodes, it has been possible to better reconstruct the input temporal sequences by retrieving the information of the far-history in the simulations, which would be lost otherwise due to the fading memory effect of the memristive processes.
By considering 81x16 input binarized Cochleagrams, and a network size of 81 electrodes, 75.7% of accuracy in the spoken digit recognition task has been achieved with the Tensorflow dataset with all six speakers, with a gain of 4.4% with respect to the system without RC. In the speaker recognition task, 87.6% accuracy has been reached in the same dataset, with a gain of 2.6% with respect to conventional one-layer neural networks.
Although the absolute value of the final accuracy is mostly determined by the pre-processing and the difficulty of the dataset (i.e noisy signals, unclear pronunciations etc..), the main result regards the gain in accuracy of the machine learning algorithm when the memristive reservoir is exploited for processing the input data: this phenomenon shows that the memristive NW network correctly transforms nonlinearly the inputs, and enhances the correlations for a more efficient classification.
Another important result concerns the gain in accuracy of the RC paradigm in hardware with lower number of nodes: it means that the architecture is capable of increasing the performance even in simpler and smaller devices, with fewer physical electrodes.

Future perspectives concern firstly the implementation in software of the proposed simulations on the digit recognition task: the performance may be different in hardware due

to the experimental complexity and the big number of electrodes involved in the task. Other tasks, such as time series-prediction and long-term future forecasting are also possible in upcoming studies.
Further progresses may lead to the combination of several networks in series, for a much more complex neural circuit-like structure. Nevertheless, the designless and self-assembly NW structure leads to unavoidable random device-to-device variations, which must be studied and controlled.
Moreover, the RC performance can be further boosted up by engineering properly the materials used for the network, optimizing the potentiation/relaxation dynamics for the specific task to be solved.

In conclusion, the memristive NW network offers great potentialities due to its brain inspired structure and synaptic behaviour, and their inclusion in the Reservoir Computing paradigm will lead to important steps forward in physical device for Artificial Intelligence.

Appendix A

Box plots: Accuracy as a function of the grid size

In section 3.5.3, it has been reported the mean test accuracy as a function of M, but for only one fixed value of virtual nodes (N=8). Here it is reported the box plots regarding the trend of the Test Accuracy for both digit and speaker recognition as a function of the virtual nodes, for every value of grid size (M).

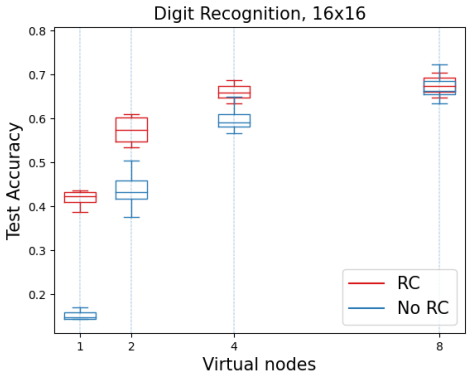


Figure A.1: Digit recognition, M=16

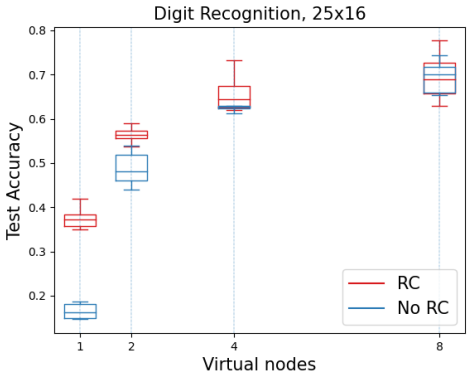


Figure A.2: Digit recognition, M=25

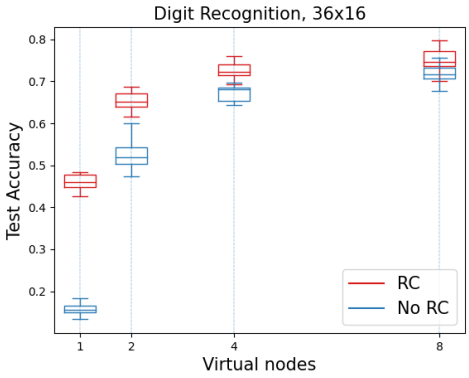


Figure A.3: Digit recognition, M=36

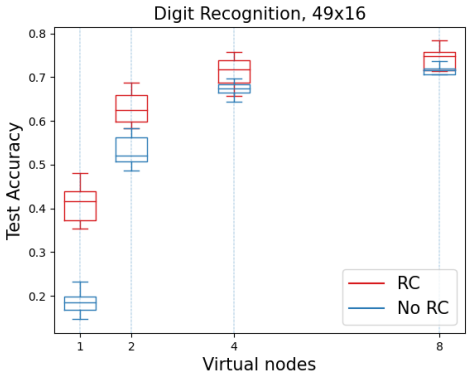


Figure A.4: Digit recognition, M=49

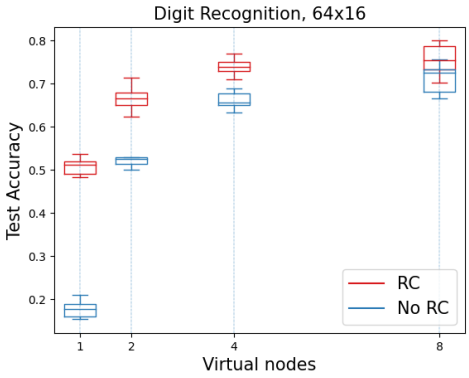


Figure A.5: Digit recognition, M=64

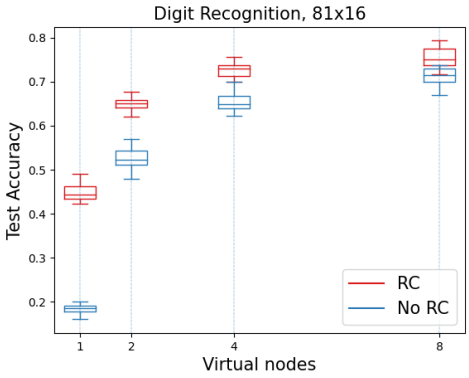


Figure A.6: Digit recognition, M=81

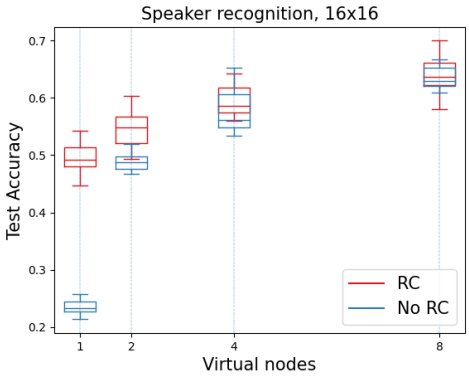


Figure A.7: Speaker recognition, M=16

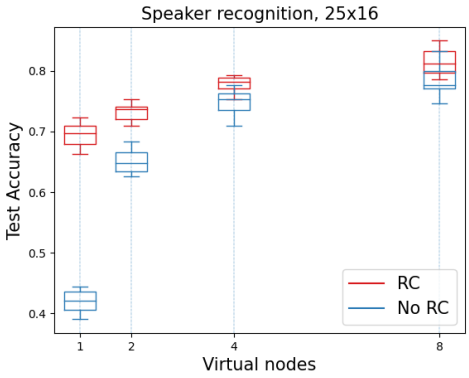


Figure A.8: Speaker recognition, M=25

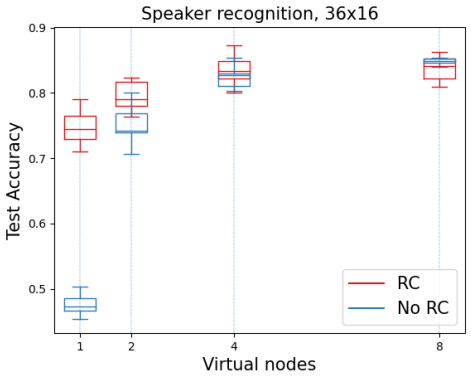


Figure A.9: Speaker recognition, M=36

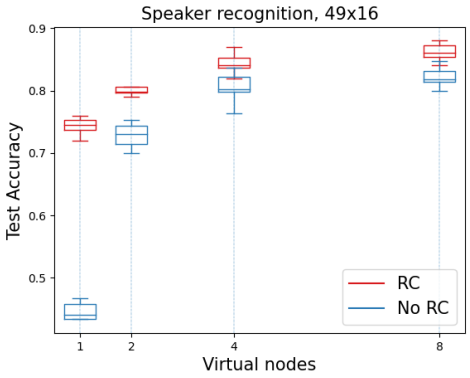


Figure A.10: Speaker recognition, M=49

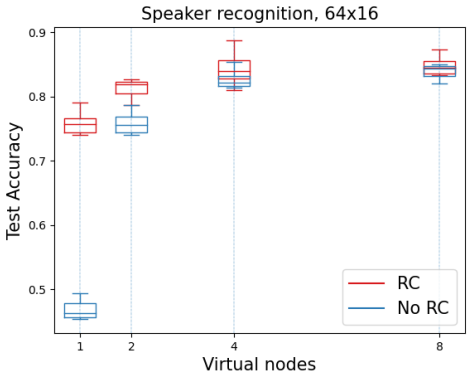


Figure A.11: Speaker recognition, M=64

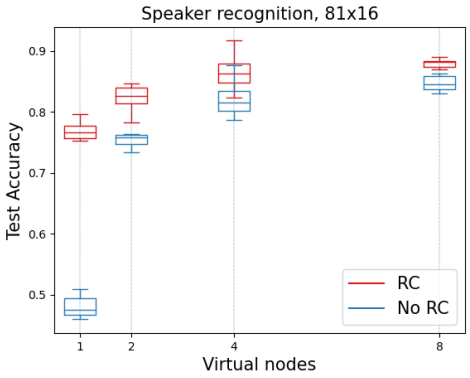


Figure A.12: Speaker recognition, M=81

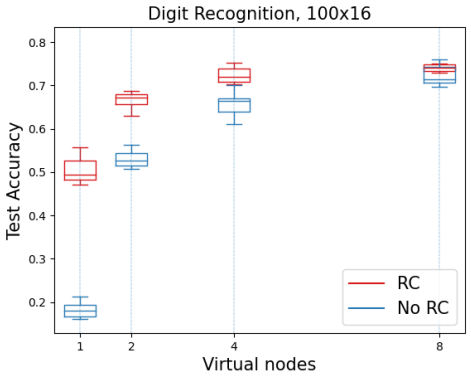


Figure A.13: Digit recognition, M=100

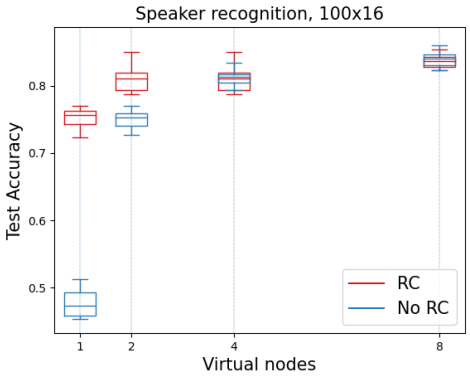


Figure A.14: Speaker recognition, M=100

Bibliography

[1] Milano, G., Luebben, M., Ma, Z. et al. *Self-limited single nanowire systems combining all in- one memristive and neuromorphic functionalities*, Nature Communications 9, 5151 (2018). DOI: 10.1038/s41467-018-07330-7

[2] L. O. Chua. *Memristor-The missing circuit element*, IEE Trans on circuit theory, vol. 18, no. 5, pp. 507-519 (1971). DOI: 10.1109/TCT.1971.1083337

[3] Waser, R., Aono, M. *Nanoionics-based resistive switching memories*, Nature Materials 6, 833-840 (2007). DOI: 10.1038/nmat2023

[4] L. Chua. *Resistance switching memories are memristors*, Appl. Phys. A 102, 765–783 (2011). DOI: 10.1007/s00339-011-6264-9

[5] Strukov, D., Snider, G., Stewart, D. et al. *The missing memristor found*, Nature 453, 80–83 (2008). DOI: 10.1038/nature06932

[6] Wang, Z., Wu, H., Burr, G.W. et al. *Resistive switching materials for information processing*, Nat Rev Mater 5, 173–195 (2020). DOI: 10.1038/s41578-019-0159-3

[7] Waser R. *Nanoelectronics and information technology: advanced electronic materials and novel devices*, 3rd ed. Weinheim, Germany, Wiley, ISBN: 978-3-527-40927-3 (2012).

[8] 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). DOI: 10.1088/1361-6463/aad581

[9] Yang, J.J., Inoue, I.H., Mikolajick, T. et al.*Metal oxide memories based on thermochemical and valence change mechanisms*. MRS Bulletin 37, 131–137 (2012). DOI: 10.1557/mrs.2011.356

[10] Valov I., Waser R., Jameson J.R., Kozicki M.N. *Electrochemical metallization memories—fundamentals, applications, prospects*. Nanotechnology, Vol. 22, n. 25 (2011). DOI: 10.1088/0957-4484/22/25/254003

[11] C. Mead, *Analog VLSI and Neural Systems*, Addison-Wesley Longman Publishing

Co., Inc., Boston, MA. ISBN:978-0-201-05992-2 (1989).

[12] Tang, J., Yuan, F., Shen, X., Wang, Z., Rao, M., He, Y., Sun, Y., Li, X., Zhang, W., Li, Y., Gao, B., Qian, H., Bi, G., Song, S., Yang, J. J., Wu, H., *Bridging Biological and Artificial Neural Networks with Emerging Neuromorphic Devices: Fundamentals, Progress, and Challenges*. Adv. Mater. 2019, 31, (2019). DOI: 10.1002/adma.201902761

[13] Abbott, L., Nelson, S. *Synaptic plasticity: taming the beast*. Nat Neurosci 3, 1178–1183 (2000). DOI: 10.1038/81453

[14] Taylor M. *The Problem of Stimulus Structure in the Behavioural Theory of Perception*. South African Journal of Psychology, vol. 3, pp. 23–45 (1973)

[15] Zucker, R., & Regehr, W. *Short-term synaptic plasticity. Annual review of physiology*. 64, 355-405 (2002). DOI: 10.1146/annurev.physiol.64.092501.114547

[16] Wang, W. and Pedretti, G. and Milo, V. and Carboni, R. and Calderoni, A. and Ramaswamy, N. and Spinelli, A. S. and Ielmini, D. *Computing of temporal information in spiking neural networks with ReRAM synapses*. Faraday Discuss., 2019,213, 453-469 (2019). DOI: 10.1039/C8FD00097B

[17] Dovydas Joksas, Adnan Mehonic, *badcrossbar: A Python tool for computing and plotting currents and voltages in passive crossbar arrays*, SoftwareX, Volume 12, 100617, ISSN 2352-7110 (2020). DOI: 10.1016/j.softx.2020.100617

[18] K. Steinbuch and U. A. W. Piske,*Learning matrices and their applications. in IEEE Transactions on Electronic Computers*, vol. EC-12, no. 6, pp. 846-862 (1963). DOI: 10.1109/PGEC.1963.263588

[19] Zhong Sun, Giacomo Pedretti, Elia Ambrosi, Alessandro Bricalli, Wei Wang, Daniele Ielmini. *Solving matrix equations in one step with cross-point resistive arrays*. Proceedings of the National Academy of Sciences, 116 (10) 4123-4128 (2019). DOI:10.1073/pnas.1815682116

[20] Xia, Q., Yang, J.J. *Memristive crossbar arrays for brain-inspired computing*. Nat. Mater. 18, 309–323 (2019). DOI: 10.1038/s41563-019-0291-x

[21] S. N. Truong, *Single Crossbar Array of Memristors With Bipolar Inputs for Neuromorphic Image Recognition*. in IEEE Access, vol. 8, pp. 69327-69332 (2020). DOI: 10.1109/ACCESS.2020.2986513

[22] Liu, Z., Tang, J., Gao, B. et al.*Neural signal analysis with memristor arrays towards high-efficiency brain–machine interfaces*. Nat Commun 11, 4234 (2020). DOI: 10.1038/s41467-020-18105-4

[23] Hu, X., Duan, S., Wang, L. et al. *Memristive crossbar array with applications in image processing*. Sci. China Inf. Sci. 55, 461–472 (2012). DOI: 10.1007/s11432-011-4410-9

[24] Duan, S., Hu, X., Wang, L. et al.*Analog memristive memory with applications in audio signal processing*. Sci. China Inf. Sci. 57, 1–15 (2014). DOI: 10.1007/s11432-013-4864-z

[25] Joel Hochstetter,Ruomin Zhu,Alon Loeffler,Adrian Diaz-Alvarez,Tomonobu Nakayama,Zdenka Kuncic,*Avalanches and edge-of-chaos learning in neuromorphic nanowire networks*, Nature Communications, 12, 1, (2021). DOI:10.1038/s41467-021-24260-z

[26] V. P. Roychowdhury, D. B. Janes, S. Bandyopadhyay and Xiaodong Wang, *Collective computational activity in self-assembled arrays of quantum dots: a novel neuromorphic architecture for nanoelectronics*, in IEEE Transactions on Electron Devices, vol. 43, no. 10, pp. 1688-1699, Oct. 1996. DOI: 10.1109/16.536815

[27] Milano, G., Porro, S., Valov, I., Ricciardi, C. *Recent Developments and Perspectives for Memristive Devices Based on Metal Oxide Nanowires*. Adv. Electron. Mater.,5, 1800909 (2019). DOI: 10.1002/aelm.201800909

[28] Cagli, C., Nardi, F., Harteneck, B., Tan, Z., Zhang, Y. and Ielmini, D. *Resistive-Switching Crossbar Memory Based on Ni–NiO Core–Shell Nanowires*. Small, 7: 2899-2905 (2011). DOI: 10.1002/smll.201101157

[29] Ting, Y.-H., Chen, J.-Y., Huang, C.-W., Huang, T.-K., Hsieh, C.-Y., Wu, W.-W., *Observation of Resistive Switching Behavior in Crossbar Core–Shell Ni/NiO Nanowires Memristor* Small, 14 (2018). DOI: 10.1002/smll.201703153

[30] P. N. Nirmalraj, A. T. Bellew, A. P. Bell, J. A. Fairfield, E. K. McCarthy, C. O'Kelly, L. F. C. Pereira, S. Sorel, D. Morosan, J. N. Coleman, M. S. Ferreira, J. J. Boland, Nano Lett. (2012), 12, 5966. DOI: 10.1021/nl303416h

[31] H. Du, T. Wan, B. Qu, F. Cao, Q. Lin, N. Chen, X. Lin, D. Chu, ACS Appl. Mater. Interfaces (2017), 9, 20762. DOI: 10.1021/acsami.7b04839

[32] Diaz-Alvarez, A., Higuchi, R., Sanz-Leon, P. et al. *Emergent dynamics of neuromorphic nanowire networks*. Sci Rep 9, 14920 (2019). DOI: 10.1038/s41598-019-51330-6

[33] Caporale, Natalia, and Yang Dan. *Spike timing–dependent plasticity: a Hebbian learning rule*. Annu. Rev. Neurosci. 31: 25-46. (2008). DOI: 10.1146/annurev.neuro.31.060407.125639

[34] LYNCH, G., DUNWIDDIE, T. & GRIBKOFF, V.*Heterosynaptic depression: a postsynaptic correlate of long-term potentiation*. Nature 266, 737–739 (1977). DOI:10.1038/266737a0

[35] Chistiakova, M., Bannon, N. M., Bazhenov, M., & Volgushev, M. *Heterosynaptic Plasticity: Multiple Mechanisms and Multiple Roles*. The Neuroscientist, 20(5), 483–498 (2014). DOI: 10.1177/1073858414529829

[36] Manning, H.G., Niosi, F., da Rocha, C.G. et al. *Emergence of winner-takes-all connectivity paths in random nanowire networks*. Nat Commun 9, 3219 (2018). DOI: 10.1038/s41467-018-05517-6

[37] Loeffler A, Zhu R, Hochstetter J, Li M, Fu K, Diaz-Alvarez A, Nakayama T, Shine JM and Kuncic Z. *Topological Properties of Neuromorphic Nanowire Networks*. Front. Neurosci. 14:184 (2020). DOI: 10.3389/fnins.2020.00184

[38] Milano, G., Pedretti, G., Fretto, M., Boarino, L., Benfenati, F., Ielmini, D., Valov, I. and Ricciardi, C. *Brain-Inspired Structural Plasticity through Reweighting and Rewiring in Multi-Terminal Self-Organizing Memristive Nanowire Networks*. Adv. Intell. Syst., 2: 2000096 (2020). DOI: 10.1002/aisy.202000096

[39] Milano, G., Pedretti, G., Fretto, M., Boarino, L., Benfenati, F., Ielmini, D., ... & Ricciardi, C. (2019). Self-organizing memristive nanowire networks with structural plasticity emulate biological neuronal circuits. arXiv preprint arXiv:1909.02438.

[40] E. Miranda, G. Milano and C. Ricciardi, *Modeling of Short-Term Synaptic Plasticity Effects in ZnO Nanowire-Based Memristors Using a Potentiation-Depression Rate Balance Equation*, in IEEE Transactions on Nanotechnology, vol. 19, pp. 609-612, (2020). DOI: 10.1109/TNANO.2020.3009734

[41] A. Rodriguez-Fernandez, C. Cagli, J. Suñe and E. Miranda, *Switching Voltage and Time Statistics of Filamentary Conductive Paths in HfO2-Based ReRAM Devices*, in IEEE Electron Device Letters, vol. 39, no. 5, pp. 656-659 (2018). DOI: 10.1109/LED.2018.2822047

[42] Verstraeten, D. et al. *An experimental unification of reservoir computing methods*. Neural Netw. 20, 391–403 (2007). DOI: 10.1016/j.neunet.2007.04.003

[43] Schrauwen, Benjamin, David Verstraeten, and Jan Van Campenhout. *An overview of reservoir computing: theory, applications and implementations*. Proceedings of the 15th european symposium on artificial neural networks. p. 471-482 (2007). DOI: 10.1.1.155.2814

[44] Du, C., Cai, F., Zidan, M.A. et al. *Reservoir computing using dynamic memristors for temporal information processing*. Nat Commun 8, 2204 (2017). DOI: 10.1038/s41467-017-02337-y

[45] Jaeger, H., & Haas, H.*Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication*. Science, 304, 78–80 (2004). DOI: 10.1126/science.1091277

[46] M. A. Escalona-Morán, M. C. Soriano, I. Fischer and C. R. Mirasso,*Electrocardiogram Classification Using Reservoir Computing With Logistic Regression*, in IEEE Journal of Biomedical and Health Informatics, vol. 19, no. 3, pp. 892-898, (2015). DOI: 10.1109/JBHI.2014.2332001

[47] E. Donati et al.,*Processing EMG signals using reservoir computing on an event-based neuromorphic system*, 2018 IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 1-4 (2018). DOI: 10.1109/BIOCAS.2018.8584674

[48] Keuninckx, L., Danckaert, J. & Van der Sande, G. *Real-time Audio Processing with a Cascade of Discrete-Time Delay Line-Based Reservoir Computers*. Cogn Comput 9, 315–326 (2017). DOI: 10.1007/s12559-017-9457-5

[49] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, Akira Hirose, *Recent advances in physical reservoir computing: A review*, Neural Networks, Vol. 115, P. 100-123, ISSN 0893-6080 (2019). DOI: 10.1016/j.neunet.2019.03.005

[50] S. Boyd and L. Chua,*Fading memory and the problem of approximating nonlinear operators with Volterra series*, in IEEE Transactions on Circuits and Systems, vol. 32, no. 11, pp. 1150-1161 (1985). DOI: 10.1109/TCS.1985.1085649

[51] Fernando C., Sojakka S. (2003) *Pattern Recognition in a Bucket*. In: Banzhaf W., Ziegler J., Christaller T., Dittrich P., Kim J.T. (eds) Advances in Artificial Life. ECAL 2003. Lecture Notes in Computer Science, vol 2801. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-540-39432-7_63

[52] Appeltant, L., Soriano, M., Van der Sande, G. et al. *Information processing using a single dynamical node as complex system*. Nat Commun 2, 468 (2011). DOI: 10.1038/ncomms1476

[53] Antonik, Piotr. *Application of FPGA to Real-Time Machine Learning: Hardware Reservoir Computers and Software Image Processing*. Springer, 2018. DOI: 10.1007/978-3-319-91053-6

[54] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer, *Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing*, Opt. Express 20, 3241-3249 (2012). DOI: 10.1364/OE.20.003241

[55] D. Brunner, M. C. Soriano and I. Fischer, *High-Speed Optical Vector and Matrix Operations Using a Semiconductor Laser*, in IEEE Photonics Technology Letters, vol. 25, no. 17, pp. 1680-1683, Sept.1, 2013. DOI: 10.1109/LPT.2013.2273373

[56] K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen and P. Bienstman, *Parallel*

Reservoir Computing Using Optical Amplifiers, in IEEE Transactions on Neural Networks, vol. 22, no. 9, pp. 1469-1481, Sept. 2011. DOI: 10.1109/TNN.2011.2161771

[57] Torrejon, J., Riou, M., Araujo, F. et al. *Neuromorphic computing with nanoscale spintronic oscillators*. Nature 547, 428–431 (2017). DOI: 10.1038/nature23011

[58] Hafizovic, Sadik, et al. *A CMOS-based microelectrode array for interaction with neuronal cultures*. Journal of neuroscience methods 164.1 (2007): 93-106. DOI: 10.1016/j.jneumeth.2007.04.006

[59] Ingo R. Titze & Daniel W. Martin , *Principles of Voice Production*, The Journal of the Acoustical Society of America 104, 1148-1148 (1998). DOI: 10.1121/1.424266

[60] Oxenham, Andrew J. *How We Hear: The Perception and Neural Coding of Sound*. Annual review of psychology, vol. 69, 27-50 (2018). DOI: 10.1146/annurev-psych-122216-011635

[61] R. Lyon,*A computational model of filtering, detection, and compression in the cochlea*. ICASSP '82. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1282-1285, (1982). DOI: 10.1109/ICASSP.1982.1171644

[62] Fitzpatrick D (2001) *The auditory system*. In: Purves D, Augustine GJ, Katz LC, LaMantia A-S, McNamara JO, Williams SM, eds (2001) Neuroscience, 2nd edition. Sunderland (Massachusetts): Sinauer Associates. pp 275–296. ISBN-10: 0-87893-742-0

[63] Günter Ehret. *Stiffness gradient along the basilar membrane as a basis for spatial frequency analysis within the cochlea*, The Journal of the Acoustical Society of America 64, 1723-1726 (1978). DOI: 10.1121/1.382153

[64] Meredith LeMasurier, Peter G. Gillespie. *Hair-Cell Mechanotransduction and Cochlear Amplification*, Neuron, Volume 48, Issue 3, 403-415 (2015). DOI: 10.1016/j.neuron.2005.10.017

[65] Appeltant, L., Van der Sande, G., Danckaert, J. et al. *Constructing optimized binary masks for reservoir computing with delay systems*. Sci Rep 4, 3629 (2014). DOI: 10.1038/srep03629

[66] Moon, J., Ma, W., Shin, J.H. et al. *Temporal data classification and forecasting using a memristor-based reservoir computing system*. Nat Electron 2, 480–487 (2019). DOI: 10.1038/s41928-019-0313-3

[67] Jaeger, Herbert.*Advances in Neural Information Processing Systems*. MIT Press, V. 15 (2003).

[68] D. Verstraeten, B. Schrauwen, D. Stroobandt, J. Van Campenhout. *Isolated word recognition with the Liquid State Machine: a case study, Information Processing Letters*. Volume 95, Issue 6, Pages 521-528, ISSN 0020-0190 (2005). DOI:

10.1016/j.ipl.2005.05.019

[69] Steven Walczak, Narciso Cerpa. *Artificial Neural Networks*. Editor(s): Robert A. Meyers, Encyclopedia of Physical Science and Technology (Third Edition), Academic Press, Pages 631-645, ISBN 9780122274107 (2003). DOI: 10.1016/B0-12-227410-5/00837-1

[70] Robert Hecht-Nielsen, III.3. *Theory of the Backpropagation Neural Network*. Editor(s): Harry Wechsler, Neural Networks for Perception, Academic Press, Pages 65-93, ISBN 9780127412528 (1992). DOI: 10.1016/B978-0-12-741252-8.50010-8

[71] Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization*. 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. 1–15 (2014).

[72] Wright, Raymond E. *Logistic regression*. American Psychological Association (1995).

[73] Jackson, Z. et al. *Free Spoken Digit dataset*. (2016). TensorFlow

[74] Takumi Kotooka, Sam Lilak, Adam Stieg et al. *Ag2Se Nanowire Network as an Effective In-Materio Reservoir Computing Device*, (2021). 10.21203/rs.3.rs-322405/v1

[75] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. *Support vector machines*. IEEE Intelligent Systems and their applications, vol. 13, no. 4, pp. 18-28 (1998). DOI: 10.1109/5254.708428

[76] Abreu Araujo, F., Riou, M., Torrejon, J. et al. *Role of non-linear data processing on speech recognition task in the framework of reservoir computing*. Sci Rep 10, 328 (2020). DOI: 10.1038/s41598-019-56991-x

[77] Usami, Y., van de, B., Mathew, D. G., Chen, T., Kotooka, T., Kawashima, Y., Tanaka, Y., Otsuka, Y., Ohoyama, H., Tamukoh, H., Tanaka, H., van der, W. G., Matsumoto, T., *In-Materio Reservoir Computing in a Sulfonated Polyaniline Network*. Adv. Mater. (2021). DOI: 10.1002/adma.202102688