Università degli Studi di Catania

Dipartimento di Ingegneria Elettrica, Elettronica e Informatica
Dottorato in Ingegneria dei Sistemi, Energetica, Informatica e delle Telecomunicazioni
XXXIII Ciclo

Neoteric Approaches to Complexity, Modelling of Dynamical Systems, and Recurrent Networks through Formal and Semi-Empirical Frameworks

Angelo Giuseppe Spinosa

Supervisor: Prof. P. Arena

Thesis Doctoralis

"Il saccente è qualcuno che non parla dopo aver riflettuto, ma spaccia con sicumera una scienza che non ha, o che ha soltanto orecchiato."

– *Luca Serianni*

"All over the place, from the popular culture to the propaganda system, there is constant pressure to make people feel that they are helpless, that the only role they can have is to ratify decisions and to consume."

– *Avram Noam Chomsky*

„Ich wollte nämlich schreiben, mein Werk besteht aus zwei Teilen: aus dem, der hier vorliegt, und aus alledem, was ich nicht geschrieben habe. Und gerade dieser zweite Teil ist der Wichtige.“

– *Ludwig Josef Johann Wittgenstein*

Contents

List of Figures

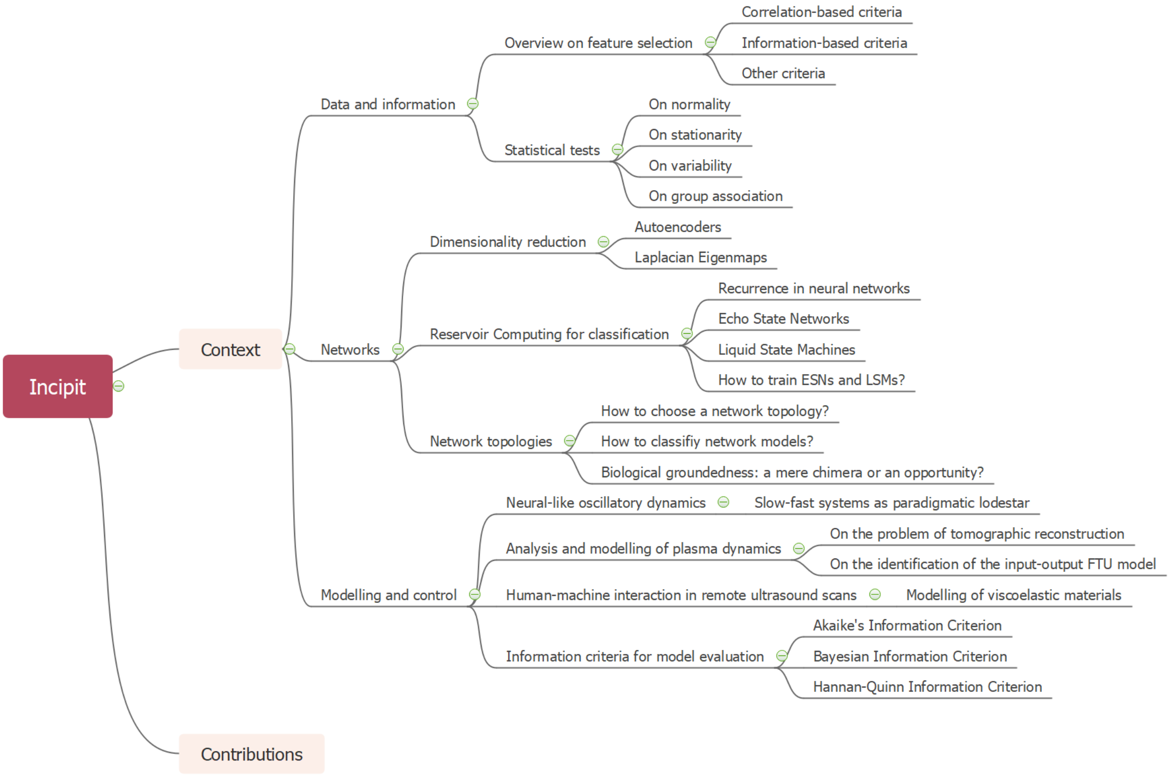
List of Tables

Incipit*- Versione italiana

Questa tesi costituisce un resoconto delle attività e dei temi di ricerca affrontati nel corso del Dottorato di Ricerca. Data la varietà degli stessi, ho strutturato la tesi dimodoché il lettore/la lettrice possa seguire percorsi tematici preferenziali indipendenti. Infatti, il documento è stato strutturato in modo che le sue sezioni siano potenzialmente slegate le une dalle altre, in modo che il lettore/la lettrice possa consultare ciò che ritiene più interessante.

Il titolo di questa tesi si riferisce ad argomenti diversi. L'intera attività di ricerca iniziò incentrandosi su alcune problematiche di alto livello legate alla biorobotica, in particolare quelle legate alla classificazione bioispirata. Quest'argomento giocò un ruolo decisivo specialmente agli inizi dell'attività, quando la mia prima area di ricerca fu quella delle reti neurali ricorrenti per la classificazione e la riduzione di dimensionalità tramite automappe del Laplaciano. In seguito, fui convolto in attività differenti da quella testé menzionata concernenti la modellazione di oscillatori non lineari attraverso una nuova strategia per il loro controllo. Ciò consentì lo sviluppo di un filone di ricerca collaterale, inizialmente non legato al tema delle reti neurali di cui sopra. La possibilità di poter legare questi ambiti si concretizzò in particolar modo quando potei lavorare sulla fisica del plasma: in un simile contesto, essi trovarono un terreno comune in cui poter crescere parallelamente. Inoltre, l'aver affrontato il problema della modellazione di materiali viscoelastici in sistemi medicali, nonché del loro controllo, rafforzò ulteriormente questo legame. Ecco perché il testo in questione è stato realizzato in modo che la struttura ricordi questo percorso che ho appena descritto.

Per mostrare meglio l'organizzazione del documento in maniera grafica, ho riportato una mappa concettuale che descrive i possibili percorsi tematici.


Incipit
Context
Data and information
Overview on feature selection
Correlation-based criteria
Information-based criteria
Other criteria
Statistical tests
On normality
On stationarity
On variability
On group association
Networks
Dimensionality reduction
Autoencoders
Laplacian Eigenmaps
Reservoir Computing for classification
Recurrence in neural networks
Echo State Networks
Liquid State Machines
How to train ESNs and LSMs?
Network topologies
How to choose a network topology?
How to classifiy network models?
Biological groundedness: a mere chimera or an opportunity?
Modelling and control
Neural-like oscillatory dynamics
Slow-fast systems as paradigmatic lodestar
Analysis and modelling of plasma dynamics
On the problem of tomographic reconstruction
On the identification of the input-output FTU model
Human-machine interaction in remote ultrasound scans
Modelling of viscoelastic materials
Information criteria for model evaluation
Akaike's Information Criterion
Bayesian Information Criterion
Hannan-Quinn Information Criterion
Contributions

* Dal latino medievale, "qui inizia".

Più nel dettaglio, vorrei evidenziare i punti cruciali di ciascun capitolo:

1. Parte I: *Context (Contesto)* - Il titolo della prima parte è sufficientemente esaustivo, poiché è stata introdotta per fornire le conoscenze fondamentali riguardanti tutti i lavori svolti nel corso del Dottorato di Ricerca. La sua realizzazione, però, è avvenuta procedendo in modo che non si trattasse di una pedissequa descrizione di tematiche prelevate da articoli, manuali o altro; piuttosto, l'idea di base è stata quella di fornire una visione fluida e discorsiva, eppure coerente e coesa, degli strumenti necessari alla comprensione dei lavori da me realizzati.
    a) Capitolo *Data and information (Dati e informazione)* - Questo capitolo mira a fornire quelle conoscenze fondamentali degli algoritmi e delle strategie per il processamento dei dati. Queste sono state organizzate in modo che potessero corrispondere al meglio con la natura dei lavori realizzati, escludendo quelle informazioni non strettamente necessarie al contesto in esame e privilegiando dunque le nozioni propedeutiche alla consultazione dei capitoli successivi.
    b) Capitolo *Networks (Reti)* - Questo capitolo tratta più nello specifico il tema delle reti neurali artificiali, mostrandone alcune applicazioni peculiari con un interesse particolare per tutto ciò che concerne la classificazione e la riduzione di dimensionalità, la seconda delle quali condivide alcuni aspetti con la selezione degli attributi più significativi che, invece, è stata trattata nel capitolo precedente.
    c) Capitolo *Modelling and control (Modellazione e controllo)* - A differenza del capitolo precedente, quest'ultimo si concentra sulla modellazione e sul controllo di sistemi dinamici, un tema, questo, che ha costituito una parte non indifferente della mia attività di ricerca sviluppantesi collateralmente al tema delle reti neurali artificiali. La trattazione in questione si è concentrata sui casi di studio di interesse, ovvero le dinamiche non lineari lente-veloci, la fisica del plasma ed il controllo di sistemi di natura viscoelastica per applicazioni medicali. Per concludere il capitolo, poi, s'è deciso di inserire un inserto sulla validazione dei modelli tramite alcuni criteri informativi.
2. Parte II: *Contributions (Contributi)* - In quest'ultima parte, ho incluso sia le introduzioni che delle brevi descrizioni di tutti i lavori da me personalmente realizzati o nei quali sono stato coinvolto. Anche stavolta, la struttura di questa sezione di documento riflette l'organizzazione principale del testo che segue l'ordine in cui i temi, e quindi i capitoli della Parte I, sono stati presentati, con la finalità di poter raggruppare questi lavori in maniera tematica. Si osservi che il Capitolo 1 della Parte I non ha effettivamente una sezione dedicata nella Parte II, ma i temi in esso trattati hanno comunque costituito degli elementi fondamentali per lo sviluppo dei Capitoli 1 e 2 della Parte II.

Vorrei precisare che, al momento della realizzazione di questa tesi, alcuni degli argomenti presentati nel Capitolo 3 della Parte I non sono stati pubblicati oppure sono in corso di revisione. Per avere più informazioni sempre aggiornate e dettagliate sulle attività in cui sono stato/sono coinvolto, è possibile consultare il mio profilo Linkedin scansionando il seguente codice QR:

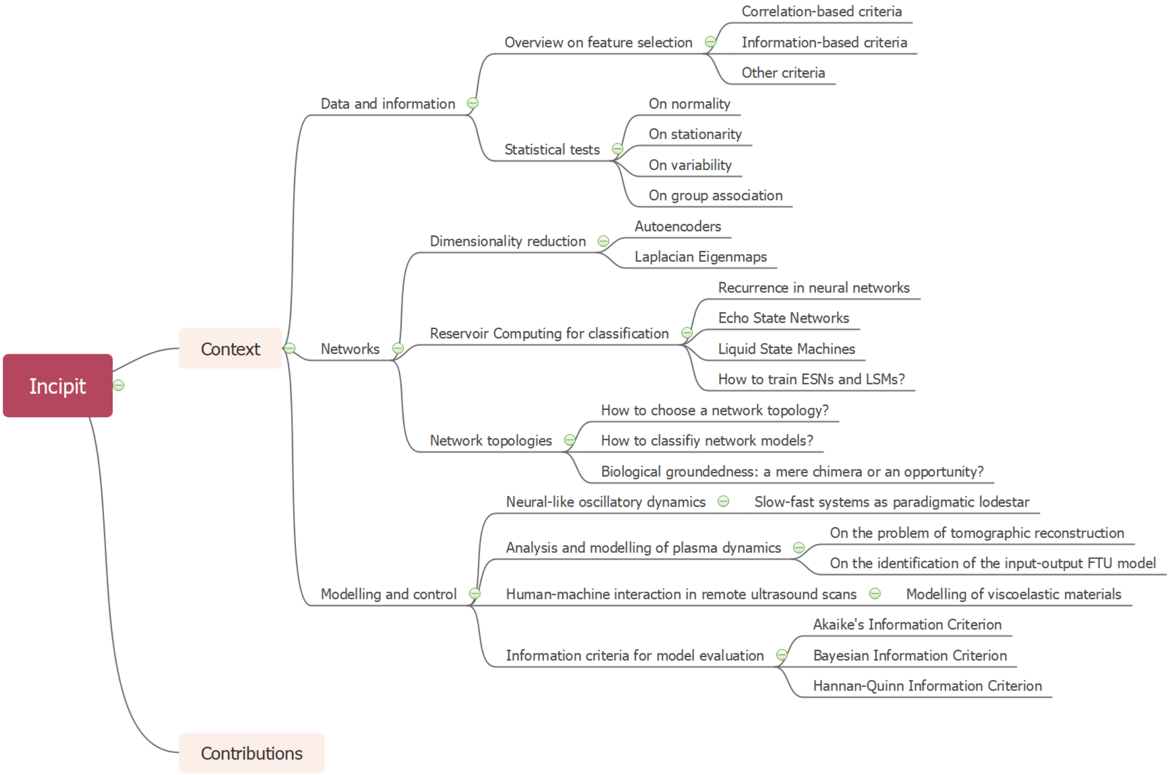
SCAN ME

Incipit[†]- English version

This thesis is a comprehensive report of the research themes and activities I worked on as Ph.D. student. Because of the diversity of these themes, I have structured the text so that the reader can follow various, preferential pathways without neglecting anything. In fact, the document has been structured in such a way each of its sections is potentially independent from each other, in order to allow the reader to consult what may sound more interesting.

The title of this thesis refers to different things. The whole research activity started as mainly focused on high-level issues in biorobotics, especially those regarding bio-inspired classification. This topic had played an important role, especially at the very beginning, when my first area of intervention had been recurrent networks for classification and dimensionality reduction through Laplacian Eigenmaps. In a next moment, I was involved in different activities regarding the problem of modelling non-linear oscillators by means of a novel control strategy. That had been the beginning of a new branch of my research, initially detached from the first one on neural networks. The opportunity of binding them came when I had been working on plasma dynamics. In that context, both data mining algorithms and systems theory had found a common ground where to grow. This bond was further extended when I had dealt with the problem of modelling visco-elastic materials for the development of control strategies in medical applications. That is why I have thereby structured the whole text in order to recall this kind of journey henceforth.

To show how the document is organised in a graphical way, I have reported a map below to describe the possible pathways.


Incipit
Context
Contributions
Data and information
Overview on feature selection
Correlation-based criteria
Information-based criteria
Other criteria
Statistical tests
On normality
On stationarity
On variability
On group association
Networks
Dimensionality reduction
Autoencoders
Laplacian Eigenmaps
Reservoir Computing for classification
Recurrence in neural networks
Echo State Networks
Liquid State Machines
How to train ESNs and LSMs?
Network topologies
How to choose a network topology?
How to classifiy network models?
Biological groundedness: a mere chimera or an opportunity?
Modelling and control
Neural-like oscillatory dynamics
Slow-fast systems as paradigmatic lodestar
Analysis and modelling of plasma dynamics
On the problem of tomographic reconstruction
On the identification of the input-output FTU model
Human-machine interaction in remote ultrasound scans
Modelling of viscoelastic materials
Information criteria for model evaluation
Akaike's Information Criterion
Bayesian Information Criterion
Hannan-Quinn Information Criterion

[†] From medieval Latin, "here begins".

In particular, I would point out what each chapter aims at:

1. Part I: *Context* - The title of this first part is quite self-explanatory: it aims at showing and discussing the fundamental ideas behind each contribution produced in this work. When writing this part, it has been decided not to slavishly report details from other sources (papers, manuals, and so forth), rather to include them in a more cohesive and coherent manner to provide a verbose description of the problems of interest.
   a) Chapter *Data and information* - This chapter provides a very fundamental glimpse into data mining and tools for information retrieval. I have revised some of the most known methods in this field to give the reader the opportunity to understand both the notation and the basic definitions for the topics I have reported in the next chapters.
   b) Chapter *Networks* - This chapter deals with neural networks and some of their capabilities. In particular, my main interest has concerned classification and dimensionality reduction, which is a similar topic to feature selection that has been outlined in the previous chapter.
   c) Chapter *Modelling and control* - Contrarily to the previous chapter, this one deals with slightly different themes. Modelling and control of dynamical systems have constituted another branch of my research activity and this chapter includes the topics I have worked on, ranging from non-linear dynamics in slow-fast systems to plasma physics and modelling of visco-elastic materials for medical applications and their control. To conclude the chapter, further information about model validation by means of some information criteria have been reported for the sake of completeness.
2. Part II: *Contributions* - This last part includes all the papers I have personally written or have been involved in. Again, to match these papers with the structure of the previous part of the document, I have organised everything so that the papers are grouped together thematically. Observe that Part I - Chapter 1 has not any explicit counterpart in Part II, but the themes I have presented there have been somehow implicit and widely employed in both Part II - Chapter 1 and Part II - Chapter 2.

I would remind the reader that some of the themes in Part I - Chapter 3 have been either not totally published or partially revised at the moment of writing. For more information about my work and activities I was/am involved in, please visit my personal Linkedin profile by scanning the following QR code:

SCAN ME

Context

Data and information | 1

The essential difference between data and information is given by *interpretation*. This has always been a so engaging topic in semiotics and science owing to the implications it carries, that even knowledge is difficult to define in a formal way [1]. When gathering data from various sources, information can be extracted once two conditions are met:

1. because of the perspective data can be observed through, users ought to decide what kind of property mining algorithms have to set off. This step is particularly crucial and strictly dependent on the purpose, which is not easily generalisable;
2. once the algorithm is chosen, data must undergo an appropriate treatment that makes them suitable for the algorithm itself. In fact, not all the mining algorithms work similarly and they may require different operating hypotheses[1] .

On one hand, the first point provides *variety* to the problem of extracting meaningful information from raw data, since the absence of a general criterion forces designers to accurately choose an appropriate algorithm. In fact, algorithms may elicit internal dependencies, as well as predictive capabilities or other properties. On the other hand, the second point provides *admissibility* for all the tasks and operations that follow and depend on the previously extracted information. For example, parametric variations on empirically identified models have to be statistically evaluated in order to understand whether they are valid independently from changes in their parameters. For example, this can be primarily assessed with confidence intervals: if they do not overlap, then model differences due to parametric variations are statistically relevant.

1.1 Overview on feature selection

Redundancy can be either an advantage or a drawback depending on the context. For example, it is a fundamental characteristic in language learning thanks to which meaning can be expressed in many ways [3], as well as in engineering or ICT applications [4]. However, the other side of the coin concerns the mere aspect of information and how much amount of it data sets can make explicit. In fact, redundancy can be thought of as an index that should be as low as possible in order to:

- find the smallest subset(s) of features that carry the most informative content;
- simplify both modelling and simulation phases thanks to a reduced number of degrees of freedom

and many other things. However, redundancy is just one of the different characteristics an attribute may have and therefore feature selection algorithms can detect various forms of relevance. That is why these

[1] In some cases these conditions are met just in a wide sense and this may lead to some approximations. An example is given by the FSV algorithm [2], which is based on concave minimisation.

algorithms try to emphasise specific aspects of the given attributes and carry out a cautious selection according to a specific criterion. I have already pointed out this aspect after all: in a wider sense, the mosaic of feature selection algorithms include myriads of solutions depending on what the designer aims at doing. To give a preliminary idea, Figure 1.1 is a schematic representation of the various forms of feature selection algorithms. In this figure there is not any specific indication on particular algorithms, instead it proposes to clearly show how algorithms are conceptually organised and therefore how they differ to each other.

But why should feature selection play such a role? To answer this question, remember that feature selection

- reduces the training efforts by limiting the number of input attributes the algorithm has to process;
- reduces the overall complexity of the model, because the latter will depend on a reduced number of characteristics;
- prevents overfitting

among other things. Thereby, it is not surprising if feature selection may accomplish good results in cost-benefit analyses. To give an idea, suppose that a company wants to produce a new smart device endowed with multiple sensors for continuous data acquisition and stream classification. A proper cost-benefit analysis ought to highlight how needless the use of over-dimensioned systems is if equivalent devices, with less sensors, perform equivalently or even better. That would result in a great, financial advantage too.

Correlation-based criteria

Let $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]^{\mathrm{T}}$ be a collection of n D-dimensional vectors $\mathbf{x}_i = [x_{i,1}, \cdots, x_{i,D}] \in \mathbb{R}^D$, $i \in \{1, \cdots, n\}$. A correlation function maps a pair of vectors $(\mathbf{x}_i, \mathbf{x}_j)$ to a single real value within $[-1, 1] \subset \mathbb{R}$ and therefore it is possible to calculate the whole correlation matrix as:

$$\mathrm{P}(\mathbf{X}) = \mathrm{P}(\mathbf{X})^{\mathrm{T}} = \begin{bmatrix} \rho(\mathbf{x}_1,\mathbf{x}_1) & \rho(\mathbf{x}_1,\mathbf{x}_2) & \cdots & \rho(\mathbf{x}_1,\mathbf{x}_n) \\ \rho(\mathbf{x}_2,\mathbf{x}_1) & \rho(\mathbf{x}_2,\mathbf{x}_2) & \cdots & \rho(\mathbf{x}_2,\mathbf{x}_n) \\ \rho(\mathbf{x}_3,\mathbf{x}_1) & \rho(\mathbf{x}_3,\mathbf{x}_2) & \cdots & \rho(\mathbf{x}_3,\mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ \rho(\mathbf{x}_n,\mathbf{x}_1) & \rho(\mathbf{x}_n,\mathbf{x}_2) & \cdots & \rho(\mathbf{x}_n,\mathbf{x}_n) \end{bmatrix} \tag{1.1}$$

In statistics there are (at least) three main types of correlation. In order to show how they differ to each other, let us suppose two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ only for the sake of simplicity. Additionally, I will refer to $\mathrm{cov}(\mathbf{x}_i, \mathbf{x}_j)$ and $\sigma(\mathbf{x}_i)$ in the following as the covariance between $\mathbf{x}_i$ and $\mathbf{x}_j$ and the standard deviation of $\mathbf{x}_i$, respectively. In any case, an usual and generally accepted way to interpret the correlation between two variables states that the association is

- weak if $0 \leq |\rho(\mathbf{x}_i, \mathbf{x}_j)| < 0.3$;
- medium (or moderate) if $0.3 \leq |\rho(\mathbf{x}_i, \mathbf{x}_j)| < 0.7$;
- strong if $0.7 \leq |\rho(\mathbf{x}_i, \mathbf{x}_j)| \leq 1$

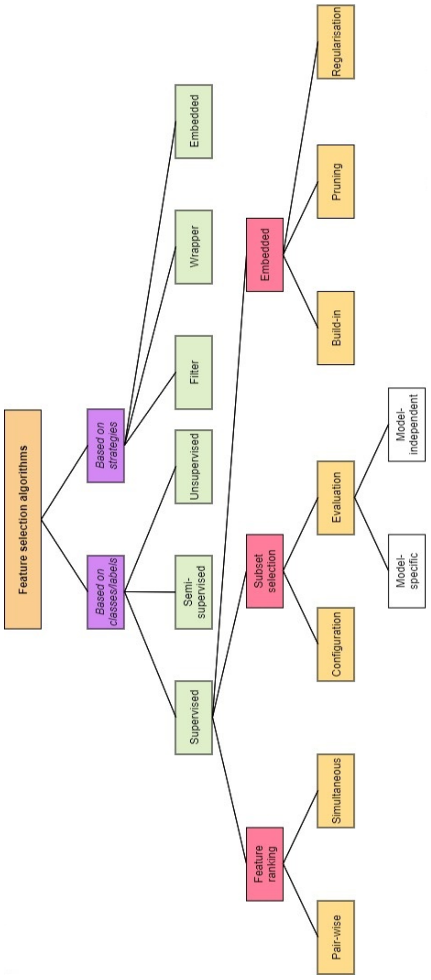


Figure 1.1: General taxonomy of feature selection algorithms. The first branch expresses how the algorithms can be categorised, depending either on the information given by (possible) labels or the adopted strategy [5]. In the former case, three solutions are possible and that reflects how the information given by the membership to a specific class, if it exists, is handled. Thereby, there are supervised, semi-supervised or even unsupervised algorithms (in this last case, no classes are available and data are completely unlabelled). In [6] a further categorisation of the supervised methods is reported: here, the author detected three types of algorithms (ranking algorithms, subset selection algorithms and embedded algorithms), each having its own peculiarities. Interestingly, but not surprisingly, some algorithms can belong to both the major categories, such as the embedded methods.

independently from the correlation function. The biggest limitation of a correlation-based measure regards its capability of capturing linear relationships only: for example, let $X \sim \text{Uniform}[-1,1]$ and $Y = X^2$ be two generic random variables (note that the expected value[2] of X is $\mathbb{E}\{X\} = 0$). By calculating the correlation coefficient between X and Y, the result is given by:

$$\mathbb{E}\{(X - \mathbb{E}\{X\})(Y - \mathbb{E}\{Y\})\} = \mathbb{E}\left\{X^3\right\} - \mathbb{E}\left\{X^2\right\}\mathbb{E}\{X\} = 0$$

Pearson's correlation

Surely it is the most known form of correlation that can be calculated whenever I want to check whether two series of data are mutually and linearly related. Its expression is given by:

$$\rho^{\text{Pearson}}(\mathbf{x}_i, \mathbf{x}_j) \doteq \frac{\text{cov}(\mathbf{x}_i, \mathbf{x}_j)}{\sigma(\mathbf{x}_i)\sigma(\mathbf{x}_j)} \tag{1.2}$$

One the most interesting properties of this coefficient is its invariance to affine transformations, meaning that

$$\rho^{\text{Pearson}}(a\mathbf{x}_i + \mathbf{b}, c\mathbf{x}_j + \mathbf{d}) = \rho^{\text{Pearson}}(\mathbf{x}_i, \mathbf{x}_j) \tag{1.3}$$

[2] The expected value of a random variable, either continuous or discrete, is the probability-weighted average of all its possible values. In its most general formulation, given a random variable X its expected value is $\mathbb{E}\{X\} = \int_{\mathbb{R}} x f(x) dx$. For more details, see the proper section about random variables.

independently from $a, c \in \mathbb{R}$ and $\mathbf{b}, \mathbf{d} \in \mathbb{R}^D$.

A right use of this correlation function requires some preliminary hypotheses that ought to be somehow assessed, otherwise it might lead to unreliable outcomes in terms of both modelling and interpretation:

- when evaluating the Pearson's correlation coefficient, data ought to be normally distributed. There are different tools that may be exploited for this purpose; in the following, I will provide some prefatory information about the Shapiro-Wilk and the Kolmogorov-Smirnov tests;
- data ought to manifest homoscedasticity[3] and linearity. The former can be assessed thanks to suitable statistical tests (as already required by the aforementioned normality assumption), such as the Bartlett's test. Although the former is a formal way to identify homoscedasticity within data, graphical approaches work as well and more immediately; a simple solution in this sense is given by scatter plots for regression analyses, because whenever data are homoscedastic they distribute in scatter plots in a rectangular pattern [7].

Spearman's correlation

Spearman's rank coefficient for correlation checks whether there is an association, and how strong it is, between two variables that could be described through a monotonic function. In order to run this (non-parametric) test, the concept of rank[4] of a given random variable is required; eventually, the expression of the correlation coefficient is given by:

$$\rho^{\text{Spearman}}(\mathbf{x}_i, \mathbf{x}_j) \doteq \frac{\text{cov}(\text{rank}(\mathbf{x}_i), \text{rank}(\mathbf{x}_j))}{\sigma(\text{rank}(\mathbf{x}_i))\sigma(\text{rank}(\mathbf{x}_j))} \tag{1.4}$$

It immediately follows that $\rho^{\text{Spearman}}(\mathbf{x}_i, \mathbf{x}_j) \equiv \rho^{\text{Pearson}}(\text{rank}(\mathbf{x}_i), \text{rank}(\mathbf{x}_j))$. ρ^{Spearman} behaves differently compared to ρ^{Pearson} and Figure 1.2 gives some hints, despite some similarities. For example, both indicate either a positive or negative association between the variables when the sign is positive or negative, but ρ^{Spearman} is less sensitive to outliers than ρ^{Pearson}, resulting in a more robust indicator for particularly noisy data. Additionally, ρ^{Spearman} does not require the hypothesis of normally distributed data to work correctly, because it is a non-parametric test, as already written before. This allows ρ^{Spearman} to relate two variables of interest through *any* monotonic function, unlike ρ^{Pearson}.

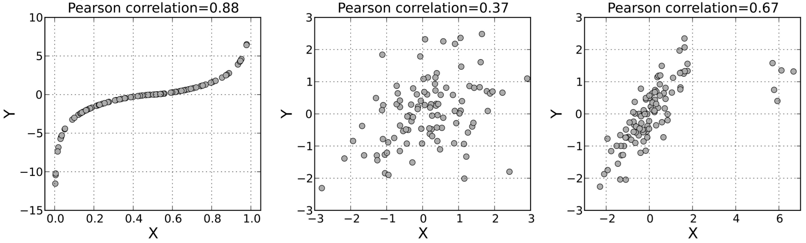


Figure 1.2: Comparison between ρ^{Pearson} and ρ^{Spearman}. When a monotonic, but not linear, function can relate two variables (leftmost image), then ρ^{Spearman} is higher than ρ^{Pearson}. Instead, both the coefficients give comparable results when data are roughly distributed without any criterion (middlemost image), but when data are affected by outliers ρ^{Pearson} is more affected by them than ρ^{Spearman} (rightmost image), resulting in a lower value.

[3] Homoscedasticity is the condition of having an equal error term across the independent variables. In other words, data are homoscedastic when they have the same variance.

[4] Ranks are defined as the positions of every value assumed by a random variable once sorted. For example, if $\mathbf{x} = [9, 3, 4, 7, 5]$, then $\text{rank}(\mathbf{x}) = [5, 1, 2, 4, 3]$.

Kendall's correlation

It is a non-parametric test that quantifies how strong the dependency between two variables is:

$$\rho^{\text{Kendall}}(\mathbf{x}_i, \mathbf{x}_j) \doteq \frac{\sum_{k=1}^{D}\sum_{n=1}^{D} \operatorname{sgn}(x_{i,k} - x_{i,n})\operatorname{sgn}(x_{j,k} - x_{j,n})}{D(D-1)} \tag{1.5}$$

where $\operatorname{sgn}(\cdot)$ is the usual sign function that is exploited to evaluate the discrepancy between concordant and discordant pairs. In particular, the product $\operatorname{sgn}(x_{ik} - x_{in})\operatorname{sgn}(x_{jk} - x_{jn})$ can be thought of a concordance indicator that is exactly equal to 1 (-1) for concordant (discordant) pairs of values. Unlike the Pearson's coefficient, what the Kendall's test for association does is to consider only the concordance/discordance amongst pairs regardless of their degrees.

Information-based criteria

To begin with, a more formal description of random variables is required. In its simplest formulation, a random variable is more properly a (real-valued) function that maps the events ω from a sample space Ω[5] to a measurable space $\mathbb{S}$:

$$\boldsymbol{\mathfrak{X}}(\omega) : \Omega \rightarrow \mathbb{S} \subseteq \mathbb{R}^D \tag{1.6}$$

Any real-valued random variable $\boldsymbol{\mathfrak{X}}$ is fully described by two functions only:

- *cumulative distribution function* (CDF): $\text{CDF}_{\boldsymbol{\mathfrak{X}}}(\mathbf{x}) \doteq \Pr\{\boldsymbol{\mathfrak{X}} \leq \mathbf{x}\}$
- *probability distribution function* (PDF): $\text{PDF}_{\boldsymbol{\mathfrak{X}}}(\mathbf{x}) \doteq \frac{d\text{CDF}_{\boldsymbol{\mathfrak{X}}}(\mathbf{x})}{d\mathbf{x}}$

An interesting, alternative description of random variables is the one founded on vector spaces [8]. These are possible if, given two stochastic variables $\boldsymbol{\mathfrak{X}}$ and $\boldsymbol{\mathfrak{Y}}$, the following operations are introduced:

- summation: $\boldsymbol{\mathfrak{X}} + \boldsymbol{\mathfrak{Y}} \in \mathbb{S}$
- scaling: $\lambda\boldsymbol{\mathfrak{X}} \in \mathbb{S}, \forall\lambda \in \mathbb{R}$
- k-norm: $\|\boldsymbol{\mathfrak{X}}\|_k \doteq [\mathbb{E}\left\{|\boldsymbol{\mathfrak{X}}|^k\right\}]^{-k} \in \mathbb{R}_0^+, \forall k \in \mathbb{R}$
- k-distance: $\|\boldsymbol{\mathfrak{X}} - \boldsymbol{\mathfrak{Y}}\|_k, \forall k \in \mathbb{R}$
- inner product: $\boldsymbol{\mathfrak{X}} \cdot \boldsymbol{\mathfrak{Y}} \doteq \mathbb{E}\{\boldsymbol{\mathfrak{X}}\boldsymbol{\mathfrak{Y}}\}$

If $\mathbb{E}\{\boldsymbol{\mathfrak{X}}\} = \eta_{\boldsymbol{\mathfrak{X}}}$ and $\mathbb{E}\{\boldsymbol{\mathfrak{Y}}\} = \eta_{\boldsymbol{\mathfrak{Y}}}$, then:

$$\begin{array}{rcl} \rho^{\text{Pearson}}(\boldsymbol{\mathfrak{X}}, \boldsymbol{\mathfrak{Y}}) & = & \boldsymbol{\mathfrak{X}} \cdot \boldsymbol{\mathfrak{Y}} \\ \operatorname{cov}(\boldsymbol{\mathfrak{X}}, \boldsymbol{\mathfrak{Y}}) & = & (\boldsymbol{\mathfrak{X}} - \eta_{\boldsymbol{\mathfrak{X}}}) \cdot (\boldsymbol{\mathfrak{Y}} - \eta_{\boldsymbol{\mathfrak{Y}}}) \end{array} \tag{1.7}$$

If not otherwise stated, in the following I will suppose $D = 1$ in Equation 1.6 so that $\mathfrak{X}$ is a real-valued, scalar, random variable.

[5] More in detail, $\Omega \neq \varnothing$ is one of the items belonging to a probability space (Ω, Λ, ν) where $\Lambda \subseteq 2^{\Omega}$ is a Borel set of events and $\nu : \Lambda \rightarrow [0, 1]$ is the probability measure that satisfies the Kolmogorov's axiomatic definition.

Entropy and Kullback-Leibler divergence

A fundamental quantity in information theory is given by the entropy[6] of a random variable:

$$\mathrm{H}(\mathcal{X}) \doteq -\int \alpha(x) \log \alpha(x) dx \tag{1.8}$$

where $\alpha(x) \equiv \mathrm{PDF}_{\mathcal{X}}(x)$. Intuitively, the entropy quantifies the average amount of information that a random variable contains and it can be adopted as a discriminating function for feature ranking. To do that, I introduce the Kullback-Leibler (KL) divergence [10] of the probability distribution β from the probability distribution α, or relative entropy, as the function:

$$\mathrm{KL}(\alpha \parallel \beta) \doteq -\int \alpha(x) \log \frac{\beta(x)}{\alpha(x)} dx \tag{1.9}$$

Observe that the KL function makes sense for those probability distributions $\alpha(x)$ and $\beta(x)$ such that $\beta(x) = 0 \Rightarrow \alpha(x) = 0$. Additionally, it can be proved that:

- $\mathrm{KL}(\alpha \parallel \beta) \neq \mathrm{KL}(\beta \parallel \alpha)$
- $\mathrm{KL}(\alpha \parallel \beta) \geq 0$

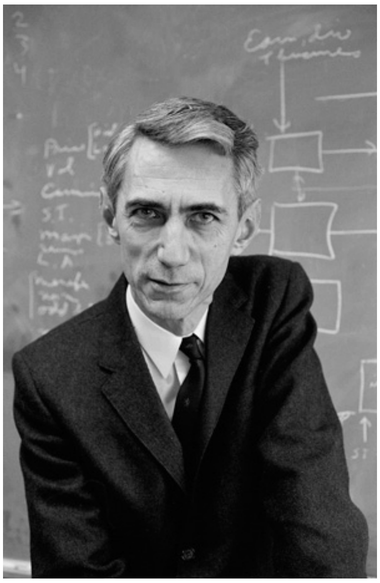


Figure 1.3: Claude Shannon (1916-2001), a pioneer of modern information theory.

Mutual information

From Equation 1.9 the mutual information between two random variables $\mathcal{X}$ and $\mathcal{Y}$, whose marginal distributions are $\alpha(x)$ and $\beta(y)$ respectively, is defined as follows:

$$\mathrm{MI}(\mathcal{X}, \mathcal{Y}) \doteq \mathrm{KL}(\Gamma(x, y) \parallel \alpha(x)\beta(y)) = -\int \int \Gamma(x, y) \log \frac{\alpha(x)\beta(y)}{\Gamma(x, y)} dx dy \tag{1.10}$$

where $\Gamma(x, y)$ is the joint probability distribution of $\mathcal{X}$ and $\mathcal{Y}$. The mutual information between two variables is characterised by the following properties:

- $\mathrm{MI}(\mathcal{X}, \mathcal{Y}) \geq 0$[7]
- $\mathrm{MI}(\mathcal{X}, \mathcal{Y}) = \mathrm{MI}(\mathcal{Y}, \mathcal{X})$
- if both $\mathcal{X}$ and $\mathcal{Y}$ are independent, then $\mathrm{MI}(\mathcal{X}, \mathcal{Y}) = 0$
- let $\mathrm{NMI}(\mathcal{X}, \mathcal{Y}) \doteq \frac{\mathrm{MI}(\mathcal{X}, \mathcal{Y})}{\mathrm{H}(\mathcal{Y})}$ be the normalised mutual information; if $\nu(\mathcal{Y}|\mathcal{X}) = \beta(y)$ then $\mathrm{NMI}(\mathcal{X}, \mathcal{Y}) = 0$, whilst $\mathrm{NMI}(\mathcal{X}, \mathcal{Y}) = 1$ when $\mathcal{X}$ completely identifies $\mathcal{Y}$.

Unlike correlation-based measures, MI can detect arbitrary, nonlinear relationships between two variables, whose probability distributions should be calculated in advance.

[6] Claude Shannon is credited with the introduction of this concept in 1948 [9]. It is said that Shannon named it following John von Neumann's recommendations and eventually after the Boltzmann's H theorem.

[7] This result is known in literature as *Jensen's inequality* [11].

Information gain

Since entropy is something related to how "surprising" information is, it could happen that a simple change of the reference frame may lead to a significant change of the entropy value. Alternatively, this variation could be used to evaluate how relevant a given feature is in determining a given output class. Essentially, information gain is the reduction in entropy or surprise by transforming a data set and is often used when training decision trees[8] . Thereby, it is calculated by comparing the entropy of the data set before and after a transformation. In other words, information gain quantifies how good a decision tree is at separating features to have distinct clusters belonging to the same class.

Let $\mathcal{X}$ and $\mathcal{Y}$ be two random variables and let $H(\mathcal{X})$ and $H(\mathcal{X},\mathcal{Y})$ be the marginal entropy of $\mathcal{X}$ and the joint entropy of both $\mathcal{X}$ and $\mathcal{Y}$, respectively. Thereby, the information gain can be easily calculated as:

$$IG(\mathcal{X}) = H(\mathcal{X}) - H(\mathcal{X},\mathcal{Y}) \tag{1.11}$$

Observe that $H(\mathcal{X}|\mathcal{Y}) = H(\mathcal{X},\mathcal{Y}) - H(\mathcal{Y})$, therefore $H(\mathcal{X}|\mathcal{Y}) = 0 \Rightarrow H(\mathcal{X},\mathcal{Y}) = H(\mathcal{Y}) \Rightarrow IG(\mathcal{X}) = \Delta H \equiv H(\mathcal{X}) - H(\mathcal{Y})$.

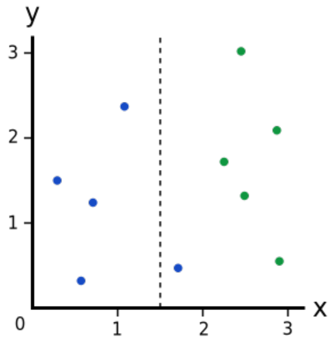


Figure 1.4: Example of imperfect separation when training decision trees and calculating the information gain. The best decision tree, when trained with the information gain, is the one that maximises it, because that is perfectly equivalent to minimising the entropy.

FCBF

FCBF, which stands for Fast Correlation Based Filter, was introduced in [13] and makes use of the concept of *predominant correlation*, meaning that this method selects features so that they have high correlation with the target variable, but little correlation with each other. Notably, the correlation used here is known as Symmetric Uncertainty (SU) [14], which is an information theory-based form of correlation described as follows (note that features have been meant as random variables and therefore they have inherited the appropriate notation I have already shown before):

$$SU(\mathcal{X},\mathcal{Y}) = \frac{2MI(\mathcal{X},\mathcal{Y})}{H(\mathcal{X}) + H(\mathcal{Y})} \tag{1.12}$$

At the beginning, the algorithm selects features whose SU with the class variable is greater than a given threshold, then it detects predominant correlations of features with the class. The definition is that, for a predominant feature $\mathcal{X}$, no other feature is more correlated to $\mathcal{X}$ than $\mathcal{X}$ itself to the class. The features more correlated with $\mathcal{X}$ than with the class are then tested, and either $\mathcal{X}$ or any other feature from this correlation group emerges as the predominant correlation feature.

Morisita estimator of the Intrinsic Dimension

Data spaces have their own dimension, which might be even fractal[9] . But suppose a set $\mathbb{K} \subseteq \mathbb{R}^n$ at first: it has topological dimension 0 if for every point $\mathbf{x} \in \mathbb{K}$ there is an open ball $\mathbb{B}_\delta(\mathbf{x})$ in $\mathbb{R}^n$ having arbitrarily small radius, whose boundary does not intersect $\mathbb{K}$. Conversely, $\mathbb{K}$ has topological dimension $k \in \mathbb{Z}^+$ if every point $\mathbf{x} \in \mathbb{K}$ is surrounded by an

[8] Training data sets with the information gain is one of the most known application of this measure, as shown by the ID3 algorithm [12].

[9] To put it simply, the fractal dimension, *aka* the Hausdorff-Bezicovič dimension, quantifies how "complicated" a self-similar figure is. In this sense, self-similarity is a crucial property that refers to an infinite nesting of structure on all scales and strictly speaking refers to a characteristic of a form exhibited when a substructure resembles a superstructure in the same form.

open ball $\mathbb{B}_\delta(\mathbf{x})$ having arbitrarily small radius whose boundary intersects $\mathbb{K}$ in a set of topological dimension $k-1$, and k is the least positive integer with this property.

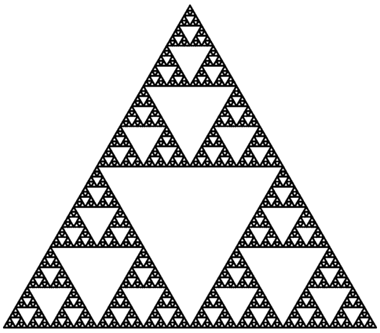


Figure 1.5: An example of fractal set: the Sierpiński triangle, whose fractal dimension is $d = \log_2 3$.

In order to introduce a more formal description of the Hausdorff-Bezicovič dimension, the concept of *box* is needed. A box in $\mathbb{R}^n$ is a set of the form:

$$\mathbb{B}_{\mathbb{R}^n}(\mathbf{a}, \mathbf{b}) \doteq \{\mathbf{x} \in \mathbb{R}^n : a_i \leq x_i \leq b_i, i = 1, \ldots, n\} \tag{1.13}$$

Now, if $\mathbb{K} \subseteq \mathbb{R}^n$ then I can introduce $N_\delta(\mathbb{K})$ as the smallest number of boxes, whose sides are equal to δ, needed to fully cover $\mathbb{K}$. It is quite obvious to deduce that as long as δ decreases, $N_\delta(\mathbb{K})$ increases following a power law function $N_\delta(\mathbb{K}) \sim \delta^{-d}$, where d is the *box counting dimension* defined as:

$$d \doteq -\lim_{\delta \to 0} \frac{\log N_\delta(\mathbb{K})}{\log \delta} \tag{1.14}$$

if it exists. For example, if $\mathbb{K} = \{x \in [0, 1]\}$, then we can cover it with one interval of length $\delta = 1$, two intervals of equal $\delta = 1/2$ and so on. Thereby $N_\delta(\mathbb{K}) = 2^n$ as long as $\delta = 2^{-n}$, then $d = 1$ according to Equation 1.14 and as obviously expected. Conversely, the well-known Cantor set has $d = \frac{\log 2}{\log 3}$ because it can be covered with 2^n intervals of length $\delta = 3^{-n}$.

What has fractality to do with dynamical systems? In many research fields this concept has proved to be particularly appealing for the implications on the phenomena of interest. For example, in [15] it was pointed out how fractality plays an interesting role in many biological phenomena, whereas in [16] the author showed how fractality could be an emergent property even in econometric systems. Now, the minimum number of features that can minimally describe a data set is called the *Intrinsic Dimension* (ID) and the so-called *Morisita estimator* of the ID does exactly what it says. This algorithm proposes to solve the well-known *curse of dimensionality*[10] problem trying to find out which of the given input features are the less redundant, thereby implying that the higher the redundancy is, the needless keeping both is. It is interesting to note that the idea behind the ID is similar to the Lyapunov's Dimension (LD) in non-linear dynamics, where the main interest regards the evaluation of the size of attractors [18–20], such as the Lorenz's system or the Hénon's map. In this case, an alternative approach is given by the well-known Kaplan-Yorke conjecture [21], which is based on the calculation of Lyapunov's exponents. Formally speaking, let $\lambda_1 \geq \lambda_2 \geq \ldots \lambda_n$ be the n Lyapunov's exponents once ordered so that $\exists j$ such that $\sum_{i=1}^{j} \lambda_i \geq 0$ and $\sum_{i=1}^{j+1} \lambda_i \leq 0$. Thereby, the LD value is calculated as:

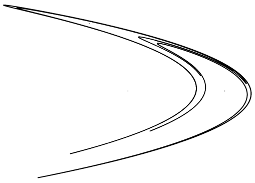


Figure 1.6: Hénon's map and calculation of its Lyapunov's dimension. When $a = 1.4$ and $b = 0.3$, its Lyapunov's exponents are $\lambda_1 = 0.603 > \lambda_2 = -2.34$, thus $j = 1$ and therefore LD = 1.26.

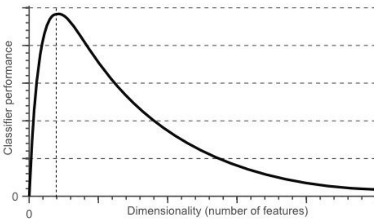


Figure 1.7: Classification performance and number of features. When the curse of dimensionality arises, the more the features for classification are, the poorer the overall goodness of classification becomes. This is due to the increasingly "specialisation" of the classifier, which learns the exceptions instead of generalising what comes from the external environment (overfitting).

$$\mathrm{LD} = j + \frac{\sum_{i=1}^{j} \lambda_i}{|\lambda_{j+1}|} \tag{1.15}$$

Regarding the calculation of the ID, to begin with let N be the number of data points and Q be the number of cell grids, whose diagonal size is

[10] Historically speaking, this expression was coined by Richard E. Bellman [17] and refers to a situation where all or some of the most relevant attributes in a data set are not discriminating enough because they are somehow concentrated near their median or mean.

set to δ, a D-dimensional space is made up of. I introduce the index $I_{m,\delta}$, namely the multi-point Morisita indicator [22], defined as:

$$I_{m,\delta} = Q^{m-1} \frac{\Sigma_{i=1}^{Q} \left[\prod_{k=0}^{m-1}(n_i - k)\right]}{\prod_{k=0}^{m-1}(n_i - k)} \tag{1.16}$$

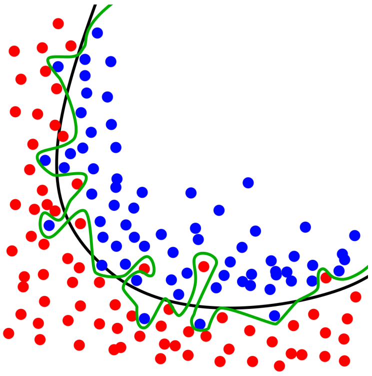


Figure 1.8: Example of overfitting. When the classifier learns perfectly how to separate two classes, then there is no room for further generalisation: in this case, a more straightforward separation, given by the black parabolic-like curve, is preferable.

which measures how many times it is more likely that $m \geq 2$ randomly selected data points come from the same cell grid than it would be if the N points were distributed accordingly to a Poisson process. Thereby, n_i in Equation 1.16 is the number of data points inside the i-th grid cell and generally $I_{m,\delta}$ is calculated R times for different values of δ. Figure 1.9 shows some examples of application of $I_{m,\delta}$ and how data partitioning changes as long as Q increases.

$I_{m,\delta}$ can be used to determine whether two features, namely F_1 and F_2 in the following for the sake of simplicity, are redundant or not. According to [23], where all the details about the feature selection algorithm are reported and the reader is reminded to refer to for further information, it can be established the following statement:

$$\text{ID}(F_1, F_2) \approx \begin{cases} 1 & F_1 \text{ and } F_2 \text{ are either linearly or non-linearly redundant} \\ 2 & F_1 \text{ and } F_2 \text{ are noy redundant} \end{cases} \tag{1.17}$$

The idea of the feature selection algorithm based on the Morisita estimator regards the discovery of those features that mostly contribute to the ID of the whole data set. Therefore, when incrementally adding new features, similarly to other more common techniques like the Sequential Feature Selection, there will be a moment when further features will not add more information together with the previously included attributes. Selection takes place when adding new features becomes useless because of the small contribution they may give to the computation of the final ID value.

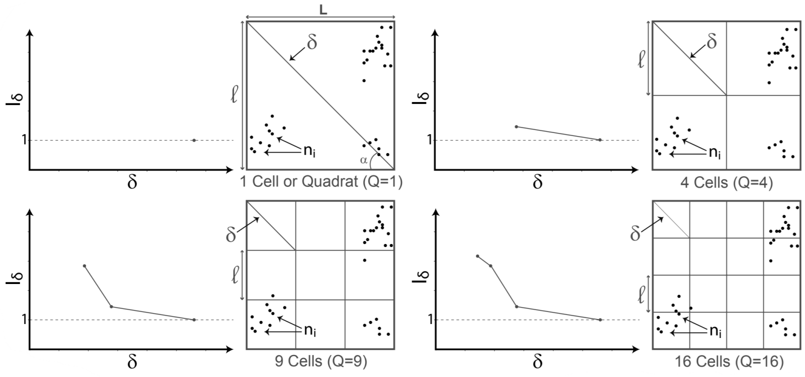


Figure 1.9: $I_{m,\delta}$ and examples of data partitioning.

Other criteria

In the following, I have reported other methods for feature selection that do not strictly belong to the previous categories I have already presented. Some of these criteria rely on statistical tests used to assess specific properties, but others simply do not. In the former case, I would refer the reader to the appropriate section of this document focused entirely on statistical tests.

Gini's index

The Gini's index (GI) [24], named after the Italian mathematician and sociologist Corrado Gini, is a measure that establishes how discriminant a given feature is with respect to the classes it could be referred to. If n is the number of classes and c_i is the i-th class, then the index for the j-th feature $\mathbf{f}_j$ is calculated as follows:

$$\mathrm{GI}(\mathbf{f}_j) = \sum_{i=1}^{n} \left[p(\mathbf{f}_j|c_i) p(c_i|\mathbf{f}_j) \right]^2 \qquad (1.18)$$

where $p(\mathbf{f}_j|c_i)$ gives the probability that the j-th feature belongs to the i-th class and $p(c_i|\mathbf{f}_j)$ gives the probability that the i-th class is drawn from the j-th feature. As well as the information gain, GI can be used to realise decision trees as splitting measure in classification problems, meaning that while building the decision tree those features with the smallest GI are preferable.

Figure 1.10: Corrado Gini (1884-1965), an Italian statistiscian and sociologist, massively contributed to different fields like economic statistics and demography with more than 800 publications.

Relief-F

The original idea behind the Relief algorithm came from Kira and Rendell [25], who introduced a statistic to quantify how good a feature at predicting the outcome is. These statistics are referred to as feature weights so that w_A is the weight of the feature A. Additionally, each feature weight can range from -1 (worst score) to $+1$ (best score). The algorithm for determining the best features is intrinsically iterative: after establishing the number of iterations m, for each iteration a training target instance R is selected without replacement and the feature weight is updated so that each update takes into account both the so-called nearest hit H and the nearest miss M, namely those instances with the same class and with the opposite class respectively whose distance to the selected target instance is minimum. Eventually, the feature weight is updated as follows:

$$w_X \leftarrow w_X + \frac{\mathrm{diff}(X, R, M) - \mathrm{diff}(X, R, H)}{m} \qquad (1.19)$$

where $\mathrm{diff}(X, I_1, I_2)$ may be defined differently depending on the type of feature (discrete or continuous) [26].

1.2 Statistical tests

Independently from the task I may be required to accomplish, data-driven results ought to be validated somehow and this can be done by running appropriate statistical tests. Additionally, these tools provide interesting information about the variables of interest and can certify whether a certain mathematical tool can be applied or not.

When dealing with statistical features to be assessed by means of a suitable test, a key concept is given by the power of that test, namely the probability of rejecting the null hypothesis and thereby making a type II error[11] . If the probability of committing a type II error is equal to β, then the power of a test is simply given by $1 - \beta$. In the following, I have reported some hints about the correct use of statistical tests with respect to their power, providing a systematic way to categorise them according to how good a formal assessment procedure for significant, statistical features like normality and stationarity may be.

[11] In statistics, when a true null hypothesis is wrongly rejected then the test commits a type I error, whilst the test makes a type II error when it accepts a false null hypothesis.

On normality

Let $\mathfrak{X}$ be a D-dimensional random variable defined as stated in Equation 1.6. If $\mathfrak{X}$ is normally distributed, *id est* $\mathfrak{X} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then:

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \tag{1.20}$$

where $\boldsymbol{\Sigma} \in \mathbb{R}^{D,D}$ is the covariance matrix and $\boldsymbol{\mu} \in \mathbb{R}^{D}$ is the mean vector. The quantity:

$$\boldsymbol{\Delta} \doteq \sqrt{(\mathbf{x}-\boldsymbol{\mu})^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})} \tag{1.21}$$

is the Mahalanobis distance from $\boldsymbol{\mu}$ to $\mathbf{x}$ and reduces to the Euclidean one when $\boldsymbol{\Sigma} = \mathbf{I}$. Furthermore, it is straightforward to check that $\boldsymbol{\Delta}^2$ is proportional to the log-likelihood function of the distribution:

$$\log p(\mathbf{x}) = -\frac{1}{2}\left[D \log 2\pi + \log|\boldsymbol{\Sigma}|\right] - \frac{1}{2}\boldsymbol{\Delta}^2 \tag{1.22}$$

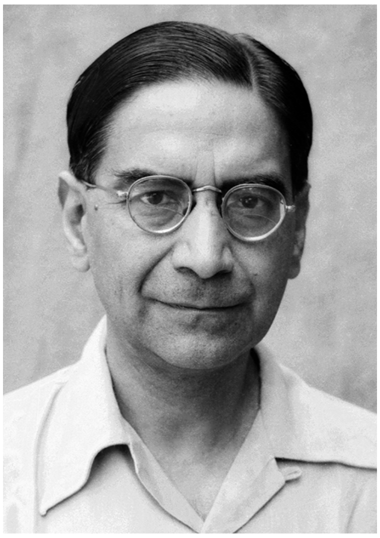


Figure 1.11: Prasanta Chandra Mahalanobis (1893-1972), who is unanimously credited with the foundation of modern statistics in India.

The importance of this kind of distribution is due to its pervasiveness in nature, where many phenomena follow, at least approximately, a normal distribution [27, 28]. Additionally, if a given phenomenon can be described as normally distributed, many more theoretically-grounded tools become available for its analysis. Since correlation implies causation if and only if the process is normal, the more Gaussian-like a phenomenon is, the stronger the implication is. Therefore, I may be interested in checking the normality of a data set out with proper tests. For the sake of this work, two main tests have been considered and whose details have been reported below, but before accounting for them I would refer the reader to [29] for further details about the power of normality tests and some hints for their mutual comparison. The aforecited article, then, states how outperforming the Shapiro-Wilk test is with respect to other possible candidates.

Shapiro-Wilk test

The most common way to determine whether some data are normally distributed or not consists in carrying out the Shapiro-Wilk test and the corresponding statistic W [30], whose null and alternative hypotheses are:

$$\begin{array}{lcc} H_0 & : & \text{population is normally distributed} \\ H_1 & : & \text{otherwise} \end{array} \tag{1.23}$$

I refer the reader to the original paper for further details about the expression of W; for the sake of simplicity, when W tends to be small then the data distribution departs from the normal one and, as a rule of thumb, I reject H_0 when the p-value is lower than the significance level (usually, $\alpha = 0.05$). To some extent, W can be deemed the squared correlation coefficient in Q-Q plots.

Kolmogorov-Smirnov test

Another solution for the problem of testing the normality hypothesis is given by the Kolmogorov-Smirnov test, which has very often been compared to other strategies like the aforementioned Shapiro-Wilk test in many papers and works [31] due to the differences in terms of their grounding principles. In fact, the Kolmogorov-Smirnov test is more general than the Shapiro-Wilk test, even though its power is lower. Additionally, the empirical CDF built when running the Kolmogorov-Smirnov test converges in probability to the real distribution as long as the number of i.i.d.[12] observations increases[13] . More in detail, let $F(x)$ and $G(x)$ be the empirical CDF and hypothesised CDF, respectively. Let:

$$D^* \doteq \max_x |F(x) - G(x)| \tag{1.24}$$

be the Kolmogorov-Smirnov statistic. Therefore:

$$\begin{array}{lcc} H_0 & : & D^* \text{ is sufficiently small} \\ H_1 & : & \text{otherwise} \end{array} \tag{1.25}$$

Straightforwardly the Kolmogorov-Smirnov test is exactly equivalent to the Shapiro-Wilk test when $G(x) \equiv \int_{-\infty}^{x} p(x)dx$ and $p(x)$ is given by the unidimensional version of Equation 1.20.

On stationarity

Before listing the most common ways of solving the problem of assessing whether some data are stationary or not, it is mandatory to provide the grounding elements to better understand what to be stationary implies.

As already done elsewhere in this document, random variables can be defined over a probability space (Ω, Λ, ν). Thereby, a stochastic process is simply a collection of random variables indexed thanks to a proper index set, usually referred to as $\mathbb{T}$ owing to the (usual) time-based nature

[12] i.i.d. stands for "independent and identically distributed" and applies for those random variables that are independent and whose distributions are equal.

[13] The theoretical premises are given by the Glivenko-Cantelli theorem [32], which establishes the asymptotic behaviour of an empirical distribution function like the one described by the Kolmogorov-Smirnov test.

of its items. On account of that, stochastic processes are functions in the form:

$$\mathcal{S}(t, \omega) : \mathbb{T} \times \Omega \rightarrow \mathbb{E} \subseteq \mathbb{R}^D \tag{1.26}$$

where $\mathcal{S}(\cdot, \omega)$ is called *realisation* or *sample function* [33]. Stationarity occurs when all the sample functions within the same process are identically distributed, but this may be too restrictive for practical use. In many cases, a more acceptable condition is a wide-sense stationarity that occurs when the given stochastic process has:

- a time-independent mean value;
- a covariance function that depends on the difference between the time instants only at which it is evaluated.

Of course, the stronger formulation of stationarity must imply the weaker one, but one must remind that the opposite implication is not always valid. In fact, the unique exception is given by Gaussian stochastic processes, for which both forms of stationarity are mutually inferable. Additionally, when moments of a stochastic process are desumable from a single realisation, then the process is further referable to as *ergodic*.

To assess stationarity given a time series, one may adopt various tools. Following the same path I have already shown before when discussing about normality, a common way consists in looking at some particular plots that may highlight interesting trends within the data or running appropriate statistical tests. To begin with the former type of assessment procedure, it is worth mentioning an interesting graphical tool for time series called correlogram (*aka* Auto Correlation Function plot or even ACF plot for the sake of brevity), namely a plot that shows autocorrelation in time-varying data like time series. In other words, the ACF plot tells how the autocorrelation changes when lagged. To give an idea of how the ACF plot works, I have reported an example with real stock trends of Amazon drawn from Yahoo Finance Stock Market, starting from 1st January 2015 to 31st December 2019 (Figure 1.12). By looking at the corresponding correlogram (Figure 1.13), it is easy to see that if the signal had been stationary, then the autocorrelation function would have progressively died down without those numerous fluctuations.

The interest in stationarity is thereby surely motivated, because it may play a striking role in data analysis especially considering that it suffices to determine whether statistical moments will converge *in probability* to their real values. A cornerstone in statistical tests for stationarity is the Augmented Dickey-Fuller (ADF) test [34]. As well as other similar tests, this kind of inspection evaluates the presence of a unit root in an autoregressive model, thereby stating that whenever H_0 cannot be accepted the process shows a stationary trend. In other words, the ADF test checks the following hypotheses out:

$$\begin{array}{rcc} H_0 & : & \text{unit root detected} \\ H_1 & : & \text{stationary trend detected} \end{array} \tag{1.27}$$

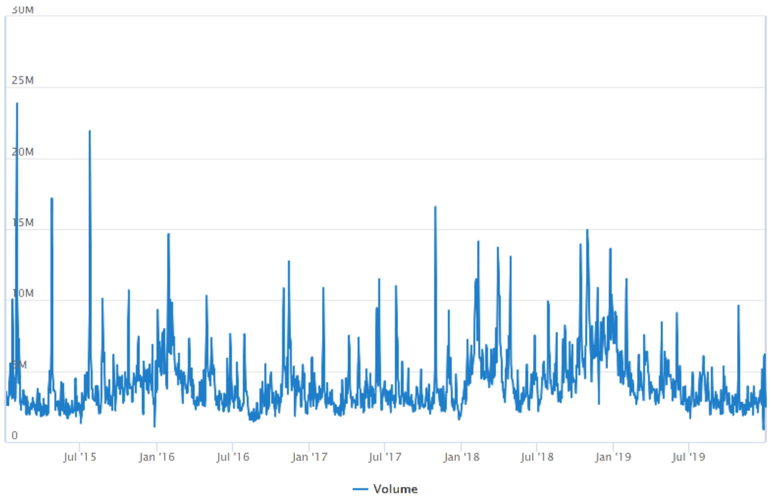


Figure 1.12: Amazon stocks by Dollar volume from 1-1-2015 to 31-12-2019.

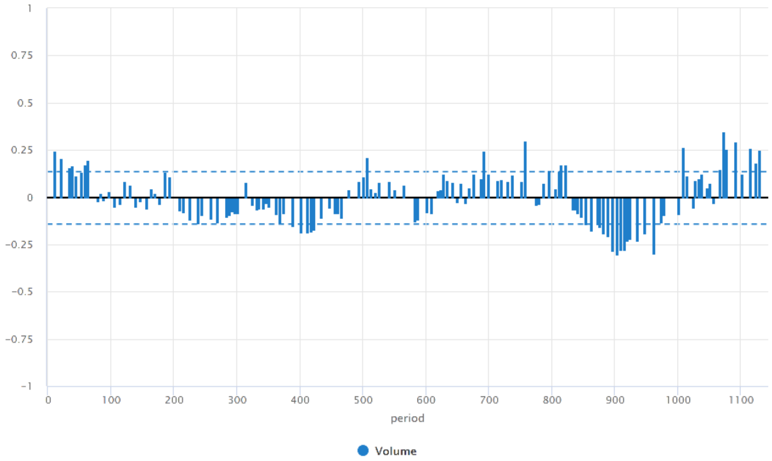


Figure 1.13: Correlogram calculated from the Amazon stocks by Dollar volume shown in Figure 1.12. Dashed lines denote the 95% significance interval.

On variability

One of the most interesting, yet straightforward, concepts in statistics is variance. Thanks to it, it is possible to determine the degree of dispersion and therefore evaluate how good some measurements are with respect to an estimated mean value. The analysis of variance, *aka* ANOVA, is therefore particularly enthralling for data processing, because by selecting those features whose variance is significantly high it impacts the performance of the classifier (the lower the variance of a given feature is, the less impactful it is with respect to the final outcome). The ANOVA test assumes the following hypotheses:

$$\begin{array}{lcc} H_0 & : & \text{Mean values of all groups are equal} \\ H_1 & : & \text{There is (at least) one mean that differs from the others} \end{array} \tag{1.28}$$

It is crucial to highlight that ANOVA works properly if data are normally distributed; when this were false, then non-parametric tests ought to be adopted, such as the Kruskal-Wallis test or the Friedman's test. Therefore, assume that two groups of data to analyse are normally distributed: if so, their reciprocal difference is significant as long as their distributions do not overlap, meaning that their mean values are very distant from each other. To quantify how different two means are, the F-ratio[14] is generally adopted and then employed for ranking features.

On group association

One may want to test how strong a possible association between two groups of data is. Although other metrics are possible, such those I have already presented before like correlation and mutual information, the χ^2 test has been designed exactly for this purpose. This test can be used to determine the association between categorical variables for n observations and is founded on the difference between the expected frequencies, denoted here as $e_{i,j}$, and the observed frequencies, denoted here as $n_{i,j}$ in one or more categories in the frequency table. If $e_{i,j} = \frac{n_i n_j}{n}$, then the χ^2 statistic is calculated as follows:

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{i,j} - e_{i,j})^2}{e_{i,j}} \tag{1.29}$$

Practically speaking, this distribution returns a probability for the computed χ^2 and the degrees of freedom $\mathrm{df} = (r-1)(c-1)$. When this probability equals 0, then there is a complete dependency between two categorical variables, whilst a probability equal to 1 means that two categorical variables are completely independent. Very often, to quantify how dependent two features are the Tschuprow's Contingency Coefficient [35] (TCC, for short) is used:

$$\mathrm{TCC} = \sqrt{\frac{\chi^2}{n\sqrt{\mathrm{df}}}} \tag{1.30}$$

where TCC = 0 denotes the absence of any association.

[14] Given a random variable $\mathcal{X}$ with parameters d_1 and d_2, if $\mathcal{X} \sim \mathrm{F}(d_1, d_2)$ then the PDF of $\mathcal{X}$ is $\frac{\sqrt{\frac{(d_1 x)^{d_1} d_2^{d_2}}{(d_1 x + d_2)^{d_1+d_2}}}}{x\mathrm{B}(0.5d_1, 0.5d_2)}$, where $x > 0$ and $\mathrm{B}(x, y) = \int_0^1 t^{x-1}(1-t)^{y-1} dt$ is the Beta function.

Networks | 2

Data classification constitutes a very well-known problem in artificial intelligence. Technically speaking, data are classified when they are provided with their own classes of membership *a priori*, meaning that attributing a label to them is essentially supervised. To some extent, assigning a label to some data is exactly equal to teaching: the student learns what the teacher says and the teacher indicates the right direction to follow. In other words: the outcomes of the process itself are already known and what the classifier ought to do is replicating them without "learning by heart", otherwise it would fall into the trap of *data overfitting*.

On the other hand, unsupervised learning is more complicated because of the lack of any direction to follow in order to guide learning towards knowledge. In other words, the learner must learn by its own without any help. Generally speaking, unsupervised learning tries to find hidden structures or meaningful properties according to a pre-defined *objective*, that can greatly vary depending on the purpose the architecture aims at fulfilling. For example, one may want to highlight how data are spatially mapped in a "topographic" fashion: Self-Organising Maps (SOMs)[1] and their variants have been designed for that. And again, one may want to detect naturally emerging groups within a data set according to some distance metrics: *k*-means and its variants do exactly that.

Independently from the myriads of algorithms one can adopt for learning, in this thesis a key role has been played by neural networks mostly. As the name itself suggests, neural networks are structures of mutually and variably connected items that resemble the structure of a brain and therefore they were introduced as a computational paradigm for emulating it. Interestingly, neural networks can be considered both as supervised and unsupervised instruments (of course, it depends on the specific type of network, because not all the neural networks have this sort of versatility) and therefore have a great range of applications. But most importantly, neural networks work very well when realising computational models of biological phenomena, owing to their intrinsic tendency to biological resemblance. In contexts like biorobotics, where one may want to introduce biologically plausible mechanisms in robots, this sounds undeniably compelling.

[1] Introduced by Teuvo Kohonen in 1982 [36], self-organising maps perform a sort of dimensionality reduction by mapping input data onto dimensionally reduced representations made up of highly clustered groups. They belong to the category of unsupervised neural networks because they do not require any prior information about which class data belong to.

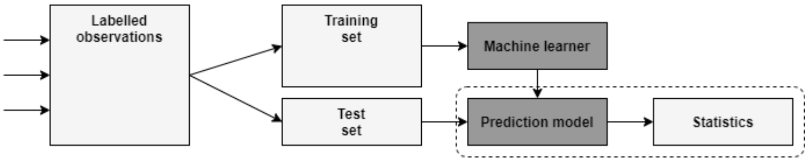


Figure 2.1: Graphical explanation of supervised learning. A prediction model requires a suitable machine learner trained over some data that constitute the training set and some test data to evaluate its performance through some statistics. Accuracy will definitely depend on how many data are correctly assigned to the expected class. Data partitioning may assume different forms: the most trivial consists in selecting a fixed amount of data from the whole data set and assigning it to the learner, but smarter solutions, like the stratified sampling or cross-validation, are allowed.

2.1 Dimensionality reduction and networks

One may argue that there is not any real difference between feature selection and dimensionality reduction. Actually, these are two distinct concepts that share some similarities. When talking about feature selection, as I have already done before, one refers to those methods for selecting the most relevant features and ruling the useless ones out because of their poor contribution to a specific objective. There is not any transformation so that the feature space turns into something else. On the contrary, dimensionality reduction changes a space by means of suitable transformations that are designed to highlight specific properties. Literature offers myriads of techniques that can be exploited for dimensionality reduction [38], but unfortunately there is not a solution that can solve every possible scenario and it all depends on what kind of data the system is requested to process.

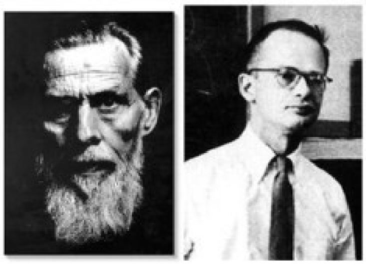


Figure 2.2: Warren McCulloch (left) (1898-1969) and Walter Pitts (right) (1923-1969) were a neurophysiologist and a mathematician, respectively. In 1943 they introduced the very first computational model of artificial neuron [37], trying to resemble some biological and physiological properties like the firing effect through a simple thresholding mechanism.

Autoencoders

Autoencoders (AEs) have gained an increasing popularity in the last few years, despite being a relatively old form of neural network which was proposed as a modular architecture for pre-training further learners in the first place, as stated in [39][2] .

The simplest form of AE is given by a network made up of three layers, where the number of hidden nodes is (strictly) lower than the number of the input nodes which in turn must equal the number of the output nodes. These constraints may sound curious, but the real motivation regards the objective this kind of network proposes to reach: what an AE outputs is a set of signals whose *reconstruction error* due to the compressed representation it stores in its innermost section is as low as possible. Thereby, let $\mathbb{U}_D \doteq \left\{\mathbf{u} \in \mathbb{R}^D\right\}$ ($\mathbb{Y}_D \doteq \left\{\mathbf{y} \in \mathbb{R}^D\right\}$) be the set of all the D-dimensional input (output) vectors; what an AE does is to learn how to map properly these two sets thanks to an *encoding function* $\phi : \mathbb{U} \longrightarrow \mathbb{F}$ and a *decoding function* $\gamma : \mathbb{F} \longrightarrow \mathbb{Y}$ so that the reconstruction error $\|\mathbf{u} - \mathbf{y}\|_2^2$ is minimum. According to this notation, $\mathbb{F} = \left\{\mathbf{f} \in \mathbb{R}^d\right\}$ is the set of the compressed features, where $d < D$. It turns out that $\mathbb{Y} \equiv (\gamma \circ \phi)(\mathbb{U})$ and therefore the problem of training an AE consists in finding both the encoding and the decoding functions:

$$\exists \bar{\phi}, \bar{\gamma} \quad : \quad \bar{\phi}, \bar{\gamma} = \underset{\phi,\gamma}{\arg\min} \left\|\mathbf{u} - (\gamma \circ \phi)(\mathbf{u})\right\|_2^2 \tag{2.1}$$

[2] In this paper, it is not clearly stated that the type of network the author dealt with was a proper AE, however. Historical ambiguities may be due to the different terminologies that various authors had used, even though they might have referred to the same concepts.

There are different reasons why AEs are particularly interesting. Because of their intrinsic structure, which resembles symmetric layering, and the sizes of the hidden layers, AEs automatically perform a dimensionality reduction process by which input data are mapped onto a shrunk representation stored inside the innermost layer within the whole structure. The problem stated in 2.1 can be rewritten differently by introducing a proper loss function that could consider various aspects of learning, such as sparsity (referred to as a function $\mathcal{S}$) or regularisation (referred to as a function $\mathcal{W}$):

$$\exists \bar{\phi}, \bar{\gamma} \quad : \quad \bar{\phi}, \bar{\gamma} = \underset{\phi, \gamma}{\arg\min} \left(\mathcal{L} + \beta \mathcal{S} + \lambda \mathcal{W} \right) \tag{2.2}$$

where $\beta, \lambda \in \mathbb{R}$ are just weighting coefficients. In order to show how each term in 2.2 is formed, suppose that all quantities related to the encoding (decoding) part of the AE are labelled with subscript E (D), so that output features are given by $\mathbf{y} = \sigma_{\mathrm{D}} \left(\mathbf{W}_{\mathrm{D}} \mathbf{f} + \mathbf{b}_{\mathrm{D}} \right)$, whilst $\mathbf{f} = \sigma_{\mathrm{E}} \left(\mathbf{W}_{\mathrm{E}} \mathbf{u} + \mathbf{b}_{\mathrm{E}} \right)$[3] . Thereby:

$$\mathcal{L} \equiv \left\| \mathbf{u} - \sigma_{\mathrm{D}} \left(\mathbf{W}_{\mathrm{D}} \sigma_{\mathrm{E}} \left(\mathbf{W}_{\mathrm{E}} \mathbf{u} + \mathbf{b}_{\mathrm{E}} \right) + \mathbf{b}_{\mathrm{D}} \right) \right\|_2^2 \tag{2.3}$$

Regarding $\mathcal{S}$, let N and $\hat{\boldsymbol{\rho}} = \left[\hat{\rho}_1, ..., \hat{\rho}_d \right]$ be the total number of input patterns to feed into the AE and the distribution of the average activation values of every hidden node, respectively. If the features produced by inputting the j-th pattern $\mathbf{u}^{(j)}$ is $\mathbf{f}^{(j)} = \sigma_{\mathrm{E}}(\mathbf{W}_{\mathrm{E}} \mathbf{u}^{(j)} + \mathbf{b}_{\mathrm{E}})$, then:

$$\hat{\boldsymbol{\rho}} = \frac{\sum_{j=1}^{N} \mathbf{f}^{(j)}}{N} \tag{2.4}$$

If $\hat{\boldsymbol{\rho}}$ is requested to be as close as possible to a desired reference value ρ, $\mathcal{S}$ can be expressed by means of the KL of $\hat{\boldsymbol{\rho}}$ and ρ:

$$\mathcal{S} \equiv \sum_{j=1}^{d} \mathrm{KL}(\rho \parallel \hat{\rho}_j) = \sum_{j=1}^{d} \left[\rho \log \left(\frac{\rho}{\hat{\rho}_j} \right) + (1 - \rho) \log \left(\frac{1 - \rho}{1 - \hat{\rho}_j} \right) \right] \tag{2.5}$$

Consequently, $\mathrm{KL}(\rho \parallel \hat{\rho}_j) \iff \rho = \hat{\rho}_j$ and as long as these values are different their mutual KL divergence increases. One of the collateral effects of sparsification concerns the small value of the sparsity regulariser itself due to smaller $\mathbf{W}_{\mathrm{E}}$ and higher $\mathbf{x}$. This kind of effect can be regulated by introducing a so-called L_2 weight regularisation $\mathcal{W}$:

$$\mathcal{W} \equiv \frac{1}{2} \sum_{i}^{N_h} \sum_{j}^{N} \sum_{k}^{N_v} w_{j,k}^{(i)} \tag{2.6}$$

where $w_{j,k}^{(i)}$ is the (j, k)-th element in $\mathbf{W}_{\mathrm{E}}$ for the i-th input pattern. Here, N_h and N_v denote the total number of hidden layers and the number

[3] In this context, $\sigma(\cdot)$ denotes the multivariate activation function of a single node.

of variables in the training data, respectively. Eventually, I can write the final form of the optimisation problem to be solved by the AE:

$$\exists \bar{\epsilon}, \bar{\delta} : \bar{\epsilon}, \bar{\delta} = \arg\min_{\epsilon,\delta} \left(\mathscr{L}(\mathbf{u}, \mathbf{y}) + \gamma \mathscr{S} + \lambda \mathscr{W}\right) \tag{2.7}$$

where λ weighs the contribution of the weight regulariser and thereby it behaves similarly to γ.

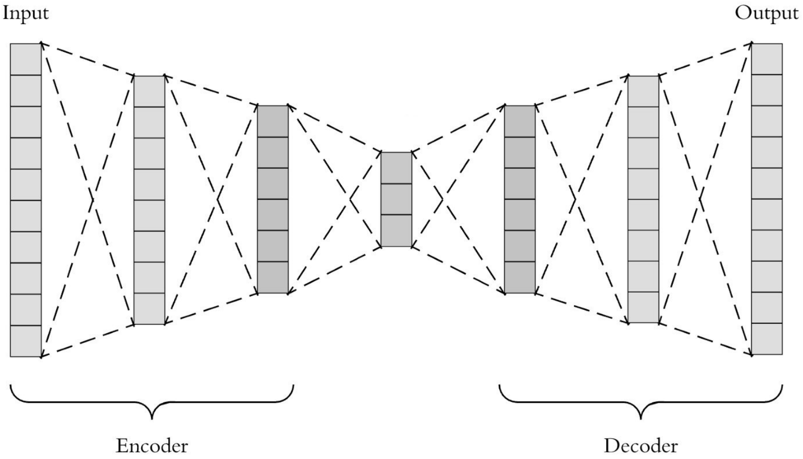


Figure 2.3: Basic scheme of an AE. Owing to the many hidden layers the AE is made up of, it should be said that this AE is deep because of the deepness through which data pass. In any case, AEs reproduce symmetric structures where the main core constitutes the compressed representation of all the input data.

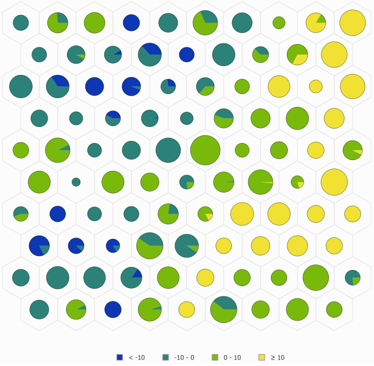


Figure 2.4: 10-by-10 hexagonal SOM of the Swiss Roll data set.

Laplacian Eigenmaps

To distinguish dimensionality reduction methods, one can rely on the type of space given by the input patterns; in this sense, one can distinguish linear sub-spaces from non-linear ones. In the first case, classical techniques like Principal Component Analysis (PCA) are sufficient to find a low-dimensional representation of the original data, but sometimes this cannot be possible. To better explain this concept, I would introduce to the well-known Swiss Roll data set, which is widely used in this context in order to show how good a manifold reduction process can be [40].

Laplacian Eigenmaps (LEs) constitute a non-linear dimensionality reduction algorithm, which employs an algebraic transformation that turns the original data space into a smaller approximating manifold. More generally, non-linear techniques like LEs, diffusion maps or Hessian Local Linear Embedding (LLE), are processing techniques and because of this they are similar to regularisation methods. However, there are some differences between these two groups of approaches [42]. The algorithm for computing LEs consists of several steps. For the sake of clarity, I have reported them below in a ordered way to highlight the natural processing flow the algorithm is based on:

1. suppose n D-dimensional points $\mathbf{x}_1, \dots, \mathbf{x}_n$. The first step consists in creating a directed graph $\mathscr{G} = (V, E)$, where $|V| = n$ is the

Type of embedding algorithm	Power
MDS	0.49
Joint Isomaps	1
Laplacian Eigenmaps	0.94

Table 2.1: Power of various manifold matching algorithms over the Swiss Roll Data set.

set of nodes and $|E| = 0$ is an initially empty set of edges. To populate the graph, one may adopt various solutions to give the sense of "closeness" amongst nodes. For example, one can use a ϵ-neighbourhood so that nodes i and j are connected if $\left\|\mathbf{x}_i - \mathbf{x}_j\right\| \leq \epsilon$[4] or the k-NN algorithm[5] ;

2. once established whether two nodes are connected or not, one has to determine how strong the connection is. To begin with, let $w_{i,j}$ be the weight of the link between nodes i and j. The simplest formulation thinks of the graph as an undirected object with just $0/1$ entries within the adjacency matrix, so that $w_{i,j} = 0$ if the nodes are not connected, $w_{i,j} = 1$ otherwise. Another solution consists in calculating the weights by means of the so-called *heat kernel*:

$$w_{i,j} = \begin{cases} e^{-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{t}} & \left\|\mathbf{x}_i - \mathbf{x}_j\right\| \leq \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{2.8}$$

   which leads to a weighted adjacency matrix whose entries belong to the real interval $[0, 1]$;

3. if $\mathcal{G}$ is has multiple connected components, this last step must be repeated for each component. Therefore, suppose a connected $\mathcal{G}$ whose (symmetric) adjacency matrix is $\mathbf{W}$. If the (i, j)-th element of $\mathbf{W}$ is $w_{i,j}$, then the matrix $\mathbf{D}$, whose (i, j)-th element is $d_{i,j} = \sum_j w_{i,j}$, is diagonal. Thereby, the Laplacian matrix of $\mathcal{G}$ is simply given by $\mathbf{L} = \mathbf{D} - \mathbf{W}$.

4. once the matrices $\mathbf{D}$ and $\mathbf{L}$ are known, the algorithm solves the generalised eigenvector problem:

$$\mathbf{Lf} = \lambda \mathbf{Df} \tag{2.9}$$

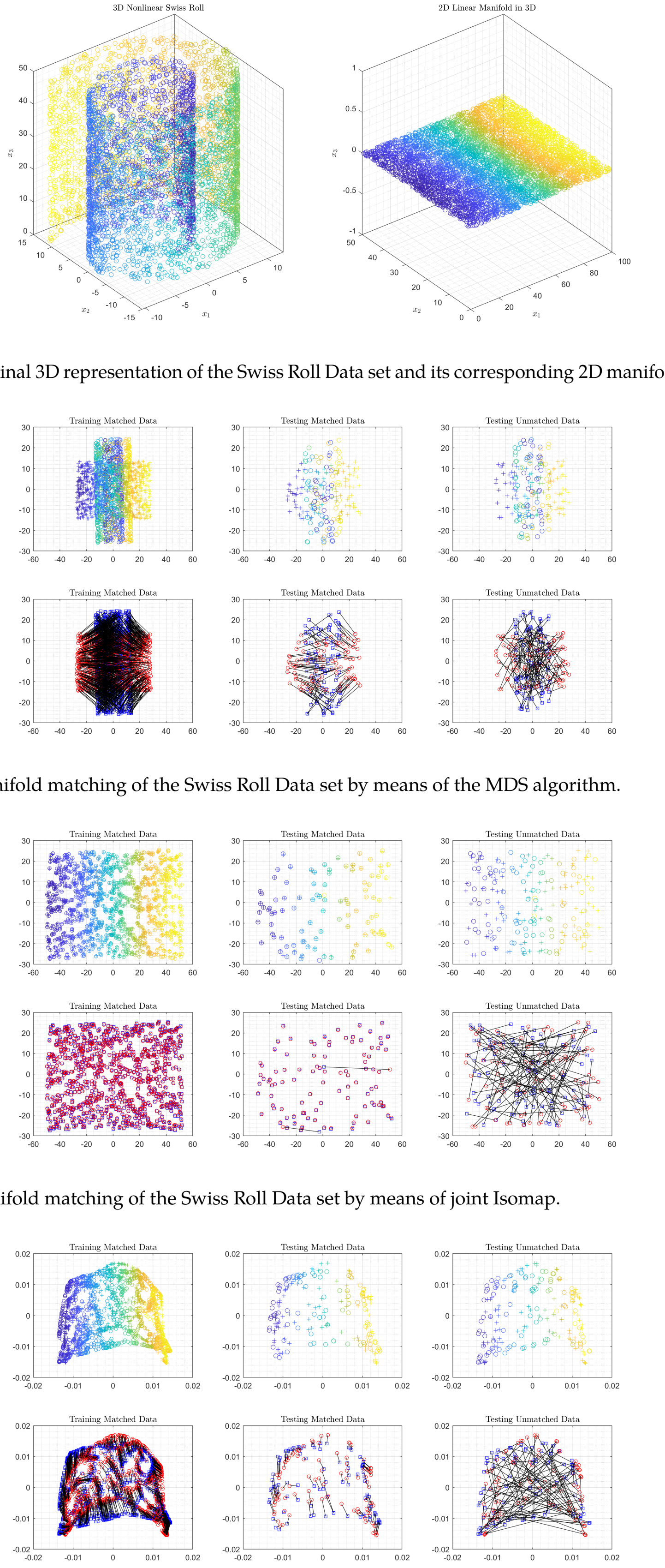
   Equation 2.9 admits n solutions $\mathbf{f}_0, \ldots, \mathbf{f}_{n-1}$, where $\mathbf{f}_0$ is the eigenvector associated to the null eigenvalue.

Further details about LEs algorithm are listed in [43], whereas Donoho and Grimes compared LEs with LLE reporting their own results in [44]. Here the manifold to be reduced was meant as locally isometric to an open and connected Euclidean space with lower dimension. Further considerations about manifold reduction and why its application in real data sets may prove very useful are reported in [45]: here, the authors have pointed out that the LEs algorithm belongs to the class of the so-called sparse spectral techniques, a sub-category of more general convex techniques for which the objective function to be optimised has the form of a generalised Rayleigh quotient and whose optimisation passes through the solution of a generalised eigenproblem. In [46] the robustness of LEs to noise and outliers has been discussed and the authors have highlighted how non-linear techniques try to preserve some notion of local geometry in the final embedding. To some extent, embedding projection relies on theoretical results that guarantee some kind of nearly preservation of mutual distances among some given points, just like the Johnson-Lindenstrauss lemma[6] [47] claims.

[4] This approach is particularly simple to interpret even from a geometrical perspective, but one must pay attention to multiple connected components within the graphs that may arise.

[5] k-NN does not tend to create multiple connected components. In spite of this, the geometrical interpretation becomes weaker.

[6] Given $0 < \varepsilon < 1$, a set $\mathbb{X} \subseteq \mathbb{R}^N$ of m points and a number $n > \frac{8 \log m}{\varepsilon^2}$, the lemma states that $\exists f(\cdot) : \mathbb{X} \to \mathbb{R}^n$ such that $\frac{\|f(u)-f(v)\|^2}{1+\varepsilon} \leq \|u - v\|^2 \leq \frac{\|f(u)-f(v)\|^2}{1-\varepsilon}$.



(a) Original 3D representation of the Swiss Roll Data set and its corresponding 2D manifold.

(b) Manifold matching of the Swiss Roll Data set by means of the MDS algorithm.

(c) Manifold matching of the Swiss Roll Data set by means of joint Isomap.

(d) Manifold matching of the Swiss Roll Data set by means of Laplacian Eigenmaps.

Figure 2.5: Manifold matching and approximated embeddings: the Swiss Roll data set example. This kind of analysis has been drawn from [41], where the idea of defining a non-linear manifold matching algorithm based on both shortest-path distance and joint neighbourhood selection is exploited to evaluate how close two embeddings are (the whole algorithm has been called Manifold Matching using Shortest-Path Distance and Joint Neighbourhood Selection on purpose, or MMSJ for short). By running a Procruster analysis either with or without non-linear embedding, one may determine how well-performing a matching algorithm is by calculating the degree of matching amongst every couple of data (x_{im}, x_{jm}), where m denotes the m-th modality of observation, so that when $x_{im} \sim x_{jm}$ then these two data are somehow matched. This analysis has been done with both training and testing data to understand the amount of "matchedness" grasped by each possible matching algorithm. Apart from the first figure on the top, every figure is made up of two rows, each having three different pictures. Every second row shows the degree of matching in a graphical way: to put it simply, a matching algorithm performs well if the matching ratio, namely the amount of corrected matched data, is very high. The MMSJ algorithm is configured so that data are split into three groups: training matched data pairs, testing matched data pairs and testing unmatched data pairs. Essentially, it learns how to map data into a dimensionally smaller embedding and then test the matching over testing data. The null hypothesis H_0 states that there is matching between two observations, therefore the power of the test $1 - \beta$ contributes to the calculation of the goodness of the matching algorithm, which has been summarised in Table 2.1. If the matching algorithm is powerful enough, then mapped points are close to each other (meaning that the modalities overlap), otherwise their mutual distance is much more significant. When the latter situation occurs, the resulting net of links connecting data points becomes clearly entangled and this is what happens with the MDS algorithm in particular, where data belonging to the two modalities are not close enough to determine short, connecting edges within the graph (that is why its second series of pictures shows intricate connections). If the MDS algorithm had performed better, then the amount of long-distance connections amongst data point would have been less significant, leading to sparse connections as happened with the Isomap algorithm, where only two pairs have shown a meaningful discrepancy. Similar concepts hold for the Laplacian Eigenmaps, which is at the halfway in terms of performance.

2.2 Reservoir Computing for classification

Reservoir Computing (RC) relies on networks where learning takes place only at the output stage, *id est* the one that concerns connections from the (last) intermediate layer to the output layer. This mechanism allows trainable readout maps, usually through regression, that make the whole network capable of replicating some pre-defined target signals (if learning is supervised). Additionally, intermediate layers are characterised by recurrent connections amongst their nodes, leading to both excitatory and inhibitory effects within them that contribute to the overall evolution.

Recurrence in neural networks

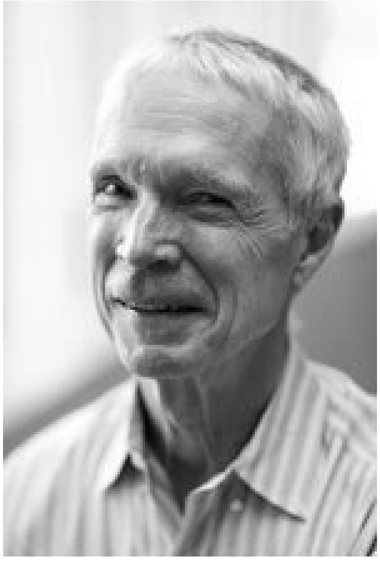


Figure 2.6: John Hopfield (1933-), born in the United States of America, is the scientist credited with the introduction of the first associative neural network.

More formally, reservoir-based networks are Recurrent Neural Networks (RNNs), whose first details were introduced in 1986 [48] even though the very first model of RNN was introduced by Hopfield[7] in 1982 [49].

What deeply characterise RNNs is the capability of processing sequential data and that constitutes a striking difference with more classical neural networks. In fact, recurrence shows to be more reliable when both input and output data are not independent from each other, as happens in many real contexts where data are intrinsically sequenced (for example, when processing audio/video streams [50] or text sequences [51]). To put it differently, RNNs are endowed with memory capabilities which allow them to be particularly useful for many applications, ranging from time series processing [52, 53] to classification [54, 55]. The simplest way to describe how a RNN works is shown in Figure 2.7, where by unfolding a single recurrence it is possible to explain the intrinsic sequential nature of the whole processing. Essentially, this property consists in turning a RNN into a feed-forward network where each cell deals with information at a specific time instant only, implying that a 10-folded RNN equals a feed-forward network with 10 intermediate layers. Additionally, it is worth noticing how both self-loops and connections from other nodes are allowed when computing the total incoming signal to be processed.

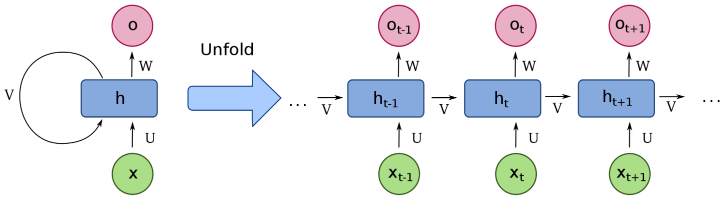


Figure 2.7: Scheme of an unfolded basic RNN (Image provided by fdeloche - Own work, CC BY-SA 4.0). The leftmost diagram is equivalent to the unrolled sequence of multiple diagrams where input and output data are considered at different time instants. Usually, the activation function $h(\cdot)$ is a sigmoidal function, but other possibilities are allowed.

Echo State Networks

Echo State Networks (ESNs) are a widely employed kind of RNN in RC computing, named after the echo property they are endowed with [56, 57]. In fact, ESNs provide a (typically supervised) learning architecture that aims to reproduce some input signals by training the weights between an output layer, which determines a combination of the whole set of incoming data, and a fixed, usually not adaptive, reservoir-like

[7] 7: Hopfield networks were introduced as associative memories with binary threshold units, meaning that the status of a node can assume two values only depending on whether it overcomes a given threshold or not.

pool of units. Training takes place once a collective reservoir activity is determined by the continuously applied input signals, which ought to be properly defined to stimulate the internal nodes of the ESN, and algebraically manipulated, typically involving the Least Mean Square (LMS) optimisation procedure. The term "echo" is due to the tunable capability of echoing past states throughout the whole time evolution, thereby leading to an intrinsic recurrence that makes ESNs particularly suitable for sequential data processing.

Suppose a neural network made up of three layers only, even though there could be versions with more than one intermediate layer [58]. This kind of network can be fully described by:

- weight matrices for both intra- and inter-layer connections. In particular, let n_i, n_r and n_o be the number of input, reservoir and output neurons, respectively. Thereby, $\mathbf{W}_{\text{in}\to\text{res}} \in \mathbb{W}_1$, $\mathbf{W}_{\text{res}} \in \mathbb{W}_2$, $\mathbf{W}_{\text{res}\to\text{out}} \in \mathbb{W}_3$ and $\mathbf{W}_{\text{out}\to\text{res}} \in \mathbb{W}_4$ are the input-to-reservoir weights, the reservoir internal weights, the reservoir-to-output weights and the feedback weights respectively, where $\mathbb{W}_1 \subseteq \mathbb{R}^{n_i,n_r}$, $\mathbb{W}_2 \subseteq \mathbb{R}^{n_r,n_r}$, $\mathbb{W}_3 \subseteq \mathbb{R}^{n_r,n_o}$ and $\mathbb{W}_4 \subseteq \mathbb{R}^{n_o,n_r}$ are normalising sets. According to the RC paradigm, the only matrix that undergoes learning is $\mathbf{W}_{\text{res}\to\text{out}}$ and this can be accomplished thanks to different approaches that will be discussed later;
- a time-valued state vector $\mathbf{x} \equiv \mathbf{x}(t) = [x_1(t), ..., x_{n_r}(t)]$, where n_r is the number of reservoir neurons. Generally speaking, it can be either $t \in \mathbb{T} \subseteq \mathbb{R}_0^+$ or $t \in \mathbb{T} \subseteq \mathbb{N}_0$ depending on whether time is continuous or discrete.

On account of the previous notation, equations that govern a typical ESN in discrete time domain are:

$$\begin{cases} \mathbf{feedback}[t] & = & \mathbf{W}_{\text{out}\to\text{res}}\mathbf{y}[t] \\ \mathbf{old}[t] & = & \mathbf{W}_{\text{res}}\mathbf{x}[t] \\ \mathbf{input}[t+1] & = & \mathbf{W}_{\text{in}\to\text{res}}\mathbf{u}[t+1] \\ \mathbf{update}[t+1] & = & \mathbf{h}\left(\mathbf{feedback}[t] + \mathbf{old}[t] + \mathbf{input}[t+1]\right) \\ \mathbf{x}[t+1] & = & \alpha\mathbf{update}[t+1] + (1-\alpha)\mathbf{x}[t] \\ \mathbf{y}[t] & = & \mathbf{g}\left(\mathbf{W}_{\text{res}\to\text{out}}\mathbf{x}[t]\right) \end{cases} \tag{2.10}$$

where $\alpha \in [0, 1]$ is the damping coefficient that weighs the contribution of the past state with respect to the newer one. The main idea of ESNs regards the way they are trained, because their training is aimed to avoid the so-called *vanishing gradient problem*, which mostly occurs in those training algorithms based on gradient descent and back-propagation. To put it simply, when a network is trained thanks to the gradient descent and back-propagation algorithms, it may occur that weights do not change sensitively due to low values of the partial derivative of the error function. In the worst case, when a network is affected by the vanishing gradient problem it cannot be updated anymore and its status is totally frozen. This effect is even more impactful if the number of layers within the network is high, because when differentiating the error function the chain rule applies and therefore the gradient becomes smaller and smaller. Although there are very simple ways of making this effect less harmful, such as by changing the activation function of the nodes (for example, ReLU[8] suffers less from it [59]), a totally different approach concerns how networks are trained, especially when dealing with long data dependencies like in time series processing, where the vanishing gradient problem might be even worse. In fact, when dealing with ESNs a fundamental requirement is the *echo state property* which assures that the network can actually forget its input signals after a while. This property is pragmatically very easy to assess, because it is sufficient to guarantee that the spectral radius[9] of the underlying adjacency matrix that describes how reservoir neurons are connected is not greater than 1.

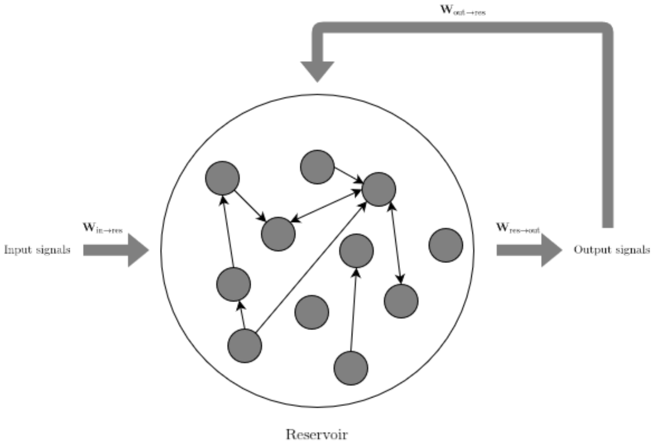

$\mathbf{W}_{\text{out}\to\text{res}}$
$\mathbf{W}_{\text{in}\to\text{res}}$
Input signals
$\mathbf{W}_{\text{res}\to\text{out}}$
Output signals
Reservoir

Figure 2.8: Basic scheme of an ESN. In this picture, I have assumed randomly fixed connections amongst reservoir neurons depicted as thinner arrows within the circle, whereas thicker arrows in grey show all the other matrices required by a general ESN.

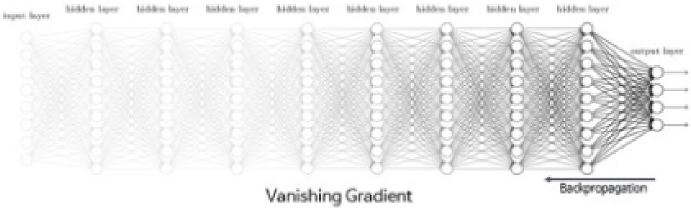


Figure 2.9: Effects of the vanishing gradient problem. The higher the number of middle layers within the network is, the smaller and smaller the gradient becomes and thereby the less likely the weights change. If a network undergoes the effects of an increasingly vanishing gradient, then it will not be able to reach the optimal weights it may require to fulfil its task, reaching instead a local optimal state.

Liquid State Machines

In principle, Liquid State Machines (LSMs) follow the same scheme shown in Figure 2.8, but there is a crucial difference between them and ESNs. Equations 2.10 describe how an ESN works, but no particular assumptions are made on the state of each node. The simplest approach consists in thinking of the state function as one of the sigmoid functions, such as the hyperbolic tangent or the logistic function, but the degree of complexity arises sensitively when the nodes are replaced with dynamical systems. When doing that, what it is generally obtained is a more complex architecture also referred to as LSMs, where the nodes behave as spiking neurons with a certain degree of biological plausibility [61, 62]. In other words, those Spiking Neural Networks (SNNs) belonging to the RC paradigm are LSMs. Biological groundedness could be obtained not only by implementing spiking nodes, but even by connecting the nodes differently and more appropriately. In fact, in spite of the original formulation it had been given to LSMs I would point out that connections amongst reservoir neurons may not be randomly fixed necessarily. Accordingly to the purpose of this kind of architecture, where spiking neurons aims at emulating real neural behaviours, it could be interesting

[8] ReLU = Rectified Linear Unit, whose expression is given by $\text{ReLU}(x) = \max(0, x)$.

[9] Let $\mathbf{A}$ be generic square matrix. Then, its spectral radius is $\rho(\mathbf{A}) = \max_{\lambda \in \text{Spec}(\mathbf{A})} |\lambda|$. Thanks to the Gelfand's formula [60], it can be rewritten differently as $\rho(\mathbf{A}) = \lim_{k \to +\infty} \sqrt[k]{\|\mathbf{A}^k\|}$, where $\|\cdot\|$ is a generic matrix norm.

to analyse how the whole system behaves when internal connections resemble, or at least try to, likely neuronal arrangements. This topic will be further discussed in the following, where additional details about network topologies have been reported for the sake of completeness.

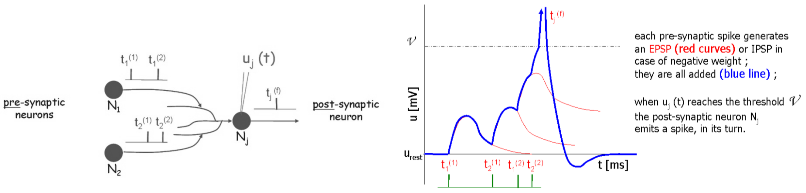

pre-synaptic neurons
$t_1^{(1)}$
$t_1^{(2)}$
N_1
$t_2^{(1)}$
$t_2^{(2)}$
N_2
u_j (t)
$t_j^{(f)}$
N_j
post-synaptic neuron
$\mathcal{V}$
u [mV]
u_{rest}
$t_j^{(f)}$
$t_1^{(1)}$
$t_2^{(1)}$
$t_1^{(2)}$
$t_2^{(2)}$
t [ms]
each pre-synaptic spike generates an EPSP (red curves) or IPSP in case of negative weight ; they are all added (blue line) ;
when u_j (t) reaches the threshold $\mathcal{V}$ the post-synaptic neuron N_j emits a spike, in its turn.

Figure 2.10: Modelling of a spiking neuron. The emission of spike from the neuron N_j is possible whenever the weighted summation of the excitatory post-synaptic potentials (EPSPs) generated by pre-synaptic neurons is high enough to overcome a threshold. This *all-or-nothing* mechanism allows the post-synaptic neuron to fire instead of decaying in time (usually, exponentially). Whenever the threshold is overcome, thereby, the membrane potential "jumps".

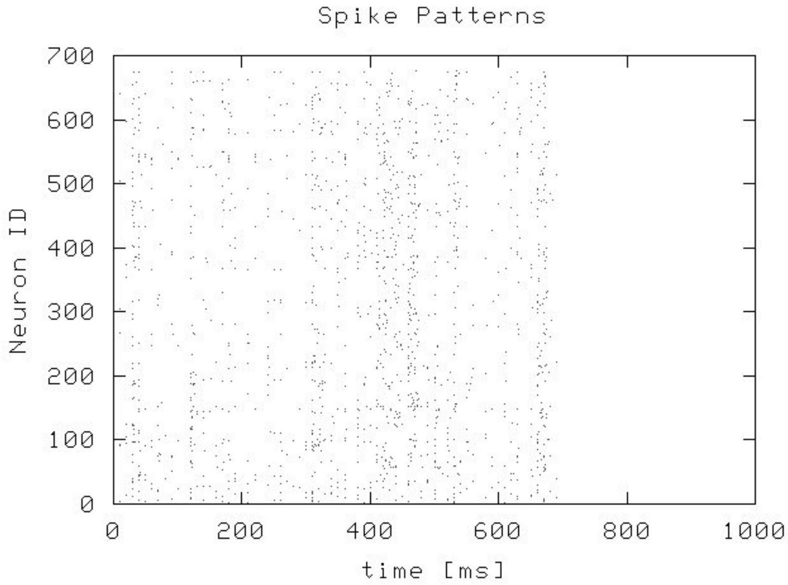


Figure 2.11: Spike trains in LSMs. This example has been realised thanks to the open-source project Amygdala, whose main purpose is to create realistic SNN-based models for AI applications. In particular, this network is made up of 680 neurons and 6360 synapses, each firing at different time instants. Pictures like this, also known as *raster plots*, show graphically series of dots whenever the neurons emit spikes.

Back in the day, LSMs had been introduced as a computational framework for neuromodelling issues, with a particular attention to cortical circuits modelling [63], proving to be quite successful. Actually, the majority of the information I have already reported about ESNs is still valid for LSMs as well, but LSMs aim to deal with *spatio-temporal information* encoded as spikes produced by reactive neurons that emit action potentials when properly stimulated. Being a specific type of SNN, it is reasonable to think of them as computational models for bio-inspired architectures: in this thesis I have worked on modelling tiny components of the *Drosophila melanogaster*'s brain in order to emulate some interesting capabilities, such as stimuli classification. These functionalities are not entirely concentrated over reduced portions of the brain, but they are supposedly spread all across it. However, a crucial information processing centre is given by the so-called Mushroom Bodies, where sensory signals coming from the external world are processed (Figure 2.12), which has been shown to be devoted to olfactory learning and memory [64, 65].

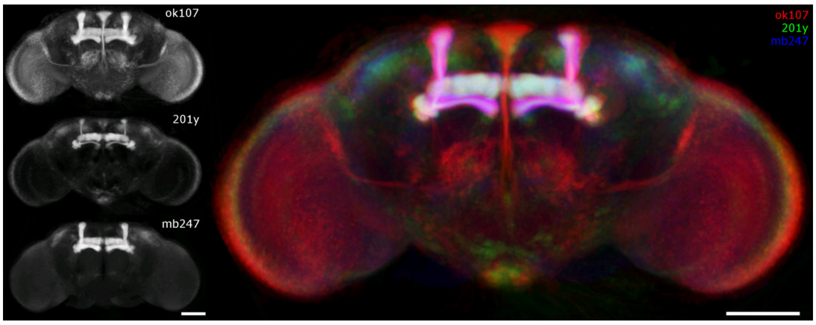


Figure 2.12: *Drosophila melanogaster* and its Mushroom Bodies [66]. Structurally speaking, they constitute neuropils of great interest for many researchers, since their involvement in learning and memory, as well as sensory information fusion from both the internal state of the body and the olfactory input to determine innate behaviour, constitutes a relevant lodestar for artificial networks and models.

How to train ESNs and LSMs?

As already mentioned elsewhere, what really makes the difference when talking about RNNs belonging to the RC paradigm is how they are trained, but so far I have not introduced these algorithms yet. Since this kind of RNNs have been introduced in order to overcome the vanishing gradient problem, it is quite expected to have networks that do not follow the back-propagation paradigm, despite probably being the most known way of training a network (actually, it is more a computational approach than other). Allegedly, there could be a neural-like form of back-propagation but that is not the same concept most of users may expect, because its current formulation does not allow any possibility of deeming it as something plausible in neural systems. But as already said, there is a specific research field entirely devoted to the understanding of whether a biological pool of neurons may reflect the essence of back-propagation and how [67–69]. In any case, what these studies show is how pressing finding an alternative description of neural phenomena is. Since LSMs, and more generally all those models that deal with biological-like dynamics, both handle and create information encoded as spike trains, an interesting paradigm that may replace the back-propagation algorithm is the Spike Timing-Dependent Plasticity[10] [70, 71] (STDP) model, which deeply binds to the spiking nature of the activity that arises in these systems.

LMS optimisation

Let $\mathbf{Ax} = \mathbf{b}$ be a system of linear equations, where $\mathbf{A} \in \mathbb{C}^{n,m}$ and $\mathbf{b} \in \mathbb{C}^{n,1}$ are known, fixed quantities. Additionally, let $\mathbf{A}^{\dagger} \equiv \left(\mathbf{A}^{\mathrm{H}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{H}}$ be the *pseudo-inverse matrix*, *aka* the Moore-Penrose matrix, of $\mathbf{A}$. The non-trivial solution to the problem is given by:

$$\hat{\mathbf{x}} = \mathbf{A}^{\dagger}\mathbf{b} \quad \text{if } \det(\mathbf{A}^{\mathrm{H}}\mathbf{A}) \neq 0 \tag{2.11}$$

In this context, the most appealing property of pseudo-inverse matrices regards how they are related to optimisation. In fact, it can be proved [72] that:

$$\forall \mathbf{x} \in \mathbb{C}^{m,1} \quad \|\mathbf{Ax} - \mathbf{b}\|_2 \geq \|\mathbf{A}\hat{\mathbf{x}} - \mathbf{b}\|_2 \tag{2.12}$$

which means that pseudo-inverse matrices solve the LMS problem effectively. Unfortunately the condition number[11] $\kappa(\mathbf{A}^{\mathrm{H}}\mathbf{A})$ is likely to be

[10] Essentially, if two neurons are connected through a synapse with weight w, then the connection is strengthened or weakened depending on the mutual spiking interaction. More precisely, if the pre-synaptic (post-synaptic) neuron emits a spike at time t_{pre} (t_{post}), then $\Delta w = f(\Delta t)$, where $\Delta t \equiv t_{\text{pre}} - t_{\text{post}}$ and f is chosen so that $\Delta t \lessgtr 0 \Rightarrow \Delta w \uparrow\downarrow$.

[11] Given the singular values $\sigma_1 \geq \ldots \geq \sigma_n$ of a n-th order matrix $\mathbf{M}$, the condition number of $\mathbf{M}$ is $\kappa(\mathbf{M}) = \frac{\sigma_1}{\sigma_n} \geq 1$.

very high in real scenarios and this may cause numerical instabilities when calculating the inverse matrix. Thereby, a simple correcting procedure, known as *Tikhonov's regularisation*, consists in adding a small perturbation in the form of a m-th order matrix $\beta\mathbf{I}$ as follows:

$$\hat{\mathbf{x}} \equiv \hat{\mathbf{x}}(\beta) = \left(\mathbf{A}^{\mathrm{H}}\mathbf{A} + \beta\mathbf{I}\right)^{-1}\mathbf{A}^{\mathrm{H}}\mathbf{b} \tag{2.13}$$

To some extent, β gives a quantitative idea on how close spectra of both $\mathbf{M}_1 \equiv \mathbf{A}^{\mathrm{H}}\mathbf{A}$ and $\mathbf{M}_2 \equiv \mathbf{M}_1 + \beta\mathbf{I}$ are and thereby this can be shown by analysing how their eigenvalues are related to this parameter. It is known that the eigenvalues of two matrices added together is not always the sum of their respective eigenvalues. However, there is a very particular condition that guarantees some more detailed results and it happens when the matrices commute[12] . This is exactly what happens for $\mathbf{M}_1$ and $\beta\mathbf{I}$. When two matrices commute, eigenvalues are simply given by the summation of the respective eigenvalues and therefore:

$$\mathrm{Spec}(\mathbf{M}_2) = \left\{\lambda + \beta : \lambda \in \mathrm{Spec}(\mathbf{M}_1)\right\} \tag{2.14}$$

In other words, Tikhonov regularisation shifts a matrix spectrum according to the tunable parameter β in order to move it away from singularity[13] and therefore disallow non-invertibility.

LMS optimisation can be used in neural networks as well for determining the optimal $\mathbf{W}_{\mathrm{res}\to\mathrm{out}}$ capable of producing output signals very close to some pre-defined targets. Supposing a linear read-out strategy like the following:

$$\mathbf{Z}\mathbf{W}_{\mathrm{res}\to\mathrm{out}} = \mathbf{Y} \tag{2.15}$$

where $\mathbf{Z} \in \mathbb{R}^{n_t,n_r}$ is the whole reservoir activity stored in a suitable matrix structure when the network undergoes some input stimuli of length n_t and $\mathbf{Y} \in \mathbb{R}^{n_t,n_o}$ is the corresponding set of time-varying output functions, if $\mathbf{T} \in \mathbb{R}^{n_t,n_o}$ is the set of target signals to track then the LMS algorithm allows to compute the optimal output weights as:

$$\hat{\mathbf{W}}_{\mathrm{res}\to\mathrm{out}} = \mathbf{Z}^{\dagger}\mathbf{T} \tag{2.16}$$

In this way, the algorithm guarantees the minimum $||\mathbf{Y} - \mathbf{T}||_2$. Again, when necessary a regularising coefficient β can be employed to better pose the computation from a numerical standpoint. I would bring up the *batch* nature of this form of learning, which means that the ongoing evolution of the whole system does not take part in the final stage of the training process. Instead, what really matters is the final "snapshot" that gives the total, collective, emerging activity stored inside the reservoir matrix.

[12] Two matrices $\mathbf{A}$ and $\mathbf{B}$ commute if $\mathbf{AB} = \mathbf{BA}$.

[13] This fact can be deemed a consequence of the Gershgorin's theorem [73].

FORCE algorithm

Although LMS training is essentially batch, meaning that learning does not occur step-by-step but just once, there are online alternatives that aim at changing the output weights at each simulation step.

Generally speaking, when dealing with incremental learning $\mathbf{W}_{\text{res}\rightarrow\text{out}} \equiv \mathbf{W}_{\text{res}\rightarrow\text{out}}[t]$[14] , meaning that each of its entries is adapted so that an error measure is minimised. For the sake of clarity, I refer to the weights connecting all the reservoir nodes to the i-th output node simply as $\mathbf{w}^{(i)} \equiv \mathbf{w}^{(i)}[t]$, therefore $\mathbf{W}_{\text{res}\rightarrow\text{out}} = \left[\mathbf{w}^{(i)}, ..., \mathbf{w}^{(n_o)}\right]$ and output weights are modified so that $\mathbf{w}^{(i)} \longleftarrow \mathbf{w}^{(i)} + \Delta\mathbf{w}^{(i)}$, where $\Delta\mathbf{w}^{(i)}$ is the i-th weight difference that is somehow function of an error measure[15] . Depending on the incremental learning algorithm, $\Delta\mathbf{w}^{(i)}$ can be calculated in different ways, but before accounting for it let $\mathbf{z}[t]$ be the state activity at time t of the whole reservoir, *id est* the row of $\mathbf{Z}$ at time t (thereby, its i-th entry indicates the activity of node i at time t), and $\mathbf{d} \in \mathbb{R}^{1,n_o} \equiv \mathbf{d}[t]$ be the vector of the n_o target signals the network ought to track (thereby, its i-th entry indicates the target that the i-th output node ought to track). According to the FORCE algorithm [74], $\Delta\mathbf{w}^{(i)}$ is related to a matrix $\mathbf{P} \equiv \mathbf{P}[t]$ whose entries $p_{i,j} \equiv p_{i,j}[t]$ decay in time in order to allow the algorithm to converge to a solution that minimises a given error function. The equations this algorithm relies on are shown below:

$$
\begin{array}{rcc}
\mathbf{P}[t] & = & \begin{cases} \frac{\mathbf{I}}{\alpha} & t = 0 \\ \mathbf{P}[t-1] - \frac{\mathbf{P}^{\mathrm{T}}[t-1]\mathbf{z}[t]^{\mathrm{T}}\mathbf{z}[t]\mathbf{P}^{\mathrm{T}}[t-1]}{1+\mathbf{z}[t]\mathbf{P}[t-1]\mathbf{z}[t]^{\mathrm{T}}} & \text{otherwise} \end{cases} \\
\mathbf{e}[t] & = & \mathbf{d}[t] - \mathbf{z}[t]\mathbf{W}_{\text{res}\rightarrow\text{out}}[t] \\
\Delta\mathbf{w}^{(i)}[t] & = & -\mathbf{e}_i[t]\mathbf{P}[t]\mathbf{z}^{\mathrm{T}}[t] \\
\mathbf{w}^{(i)}[t] & = & \mathbf{w}^{(i)}[t-1] + \Delta\mathbf{w}^{(i)}[t]
\end{array}
\tag{2.17}
$$

where $\alpha \in \mathbb{R}$ determines the diagonal of $\mathbf{P}[0]$ and therefore the performance of the whole algorithm.

2.3 Network topologies

Speaking of reservoir-based networks, an engaging aspect I could consider regards how middle neurons can be connected to each other. There is not a preferred criterion for addressing this kind of issue, because network topologies may be determined according to a specific property I may be interested in. Although this deep characterisation is essentially computational, there are some clues for selecting a topology instead of another and this section aims to briefly report the essential properties of both well-known schemes and less common configurations.

How to choose a network topology?

Generally speaking, many indices and properties can be considered in order to analyse how well-performing a network is. In the majority of cases performance is determined by how nodes are topologically arranged and

[14] In the following notation, square brackets denote discrete time-valued functions and subscripts denote entries of a vector or matrix.

[15] It is trivial to see that $\mathbf{w}^{(i)}, \Delta\mathbf{w}^{(i)} \in \mathbb{R}^{n_r,1}$.

therefore their properties follow from that. The fundamental properties are thereby:

- *degrees*[16] within the network provide a first insightful indicator of how connections are organised. It is a straightforward, yet preliminary, way of showing how the whole network is generally arranged and it helps to determine what kind of structure the system assumes (scale-free configuration, preferential attachment...);
- *paths* constitute an engaging challenge when designing networks because one may require that information travels as fast as possible from one node to another;
- tendency to *transitivity*, *id est* the tendency of creating at least triangles. The higher the level of transitivity is, the more probable clusters of nodes within the networks are;
- *centralities* are less trivial measures of importance that take into account multiple, and often more hidden, aspects of the network. There are many forms of centrality each describing a specific behaviour, therefore I refer the reader to the works and papers available in literature for a wider comprehension [75, 76].

There might be different reasons for comparing two networks, as well as tools and metrics. For example, graph spectra capture well the information about the network as a whole, as already happens in dynamical systems where eigenvalues determine their evolution. Additionally, algebraic connectivities give more details on how the network is topologically shaped [77]. Generally speaking, if $\mathcal{G}_1$ and $\mathcal{G}_2$ are two distinct graphs with both N nodes and with adjacency matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ respectively, then it may be more convenient to distinguish two types of criteria for their comparison:

- *structural distances* reflect local changes within a network. To this category belong well-known measures like the Hamming's distance[17] or the Jaccard's distance[18] , which are particularly short-sighted despite being relatively simple. In fact, what these measures do is to analyse the neighbourhood of every node within the network and therefore they neglect what is happening on a greater scale, but this is quite comprehensible because of the intrinsic nature of the distances themselves. It is also remarkable to state that whenever a distance metric evaluated over two graphs is 0, it does not mean that the graphs are equal necessarily. That is why it is necessary to make use of multiple measures, as shown in [78] where it is shown how the nullity of a distance does not imply topological equivalence;
- *spectral distances*, instead, consider how the network evolves thanks to changes in its Laplacian matrix. To this category belong the aforementioned criteria based on algebraic connectivities, as well as the spanning tree-based similarity[19] or the Hamming-Ipsen-Mikhailov distance [79, 80];
- although these distances try to overcome the limitations of the structural distances, eigenvalues must be calculated in any case[20] and sometimes this could be quite burdensome from various perspectives, not only from a computational one but also because of the sensitivity to the properties of the graph. That is why other solutions have been proposed to give a further description from a mesoscale standpoint. To this type of solutions belongs the so-called

[16] The degree is the number of connections directed to (in-degree)/from (out-degree) a given node. There are no differences in the graph if it is undirected, because in-degrees and out-degrees are exactly equivalent.

[17] $\mathrm{Hamming}(\mathcal{G}_1, \mathcal{G}_2) = \frac{\|\mathbf{A}_1 - \mathbf{A}_2\|_{1,1}}{N(N-1)}$.

[18] $\mathrm{Jaccard}(\mathcal{G}_1, \mathcal{G}_2) = \frac{\|\mathbf{A}_1 - \mathbf{A}_2\|_{1,1}}{\|\mathbf{A}_1 + \mathbf{A}_2\|_*}$.

[19] Such a similarity exploits spanning trees that could arise or be destroyed when trying to match a graph with another. That is because spanning trees could be thought of as a level of interconnectedness between the given graphs and therefore if the number of spanning trees of graph $\mathcal{G}_i$ is $\mathcal{T}_i$, then spanning tree-based similarity is given by $\mathrm{Spanning}(\mathcal{G}_1, \mathcal{G}_2) = \left|\log(\mathcal{T}_1) - \log(\mathcal{T}_2)\right|$.

[20] Computationally speaking, this calculation has complexity $O(N^3)$ for a graph with N nodes.

polynomial approach, which consists in dealing with consecutive powers of the adjacency matrices.

How to classify network models?

According to the previous list, collective behaviours arising from networks may differ a lot. An interesting classification in this sense is available in [81] and I will refer to it in the following. That having been said, there are some macro-groups of networks that can be described:

- regular graphs and trees are those networks whose level of heterogeneity is the lowest because of their intrinsic deterministic nature. Thereby the whole network tends to be highly dense, implying that clustering is very likely. However, statistically speaking these networks have the longest average paths;
- highly random networks generated by iteratively connecting couples of nodes with uniform probability are the Erdős–Rényi networks [82]. Despite their low heterogeneity, degree distribution is (approximately) normal and average paths are generally short;
- both the previous scenarios are extreme and rare situations. That is why real contexts are characterised by networks whose degree distributions vary a lot as well as the other properties I have already listed. Scale-free networks [83] constitute an important example in this sense: they are generally characterised by high modularity and degree distributions that follow a power-law[21] , at least asymptotically. A crucial difference between the normal and power-law distribution is that the number of nodes with really high numbers of edges is much higher in the power-law distribution than in the normal distribution. But generally, well-connected nodes are more common in a normal distribution. This means that in networks I often find a small number of very highly connected nodes, which have a number of connections that would not occur if the distribution were normal.

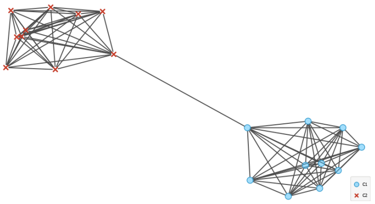


Figure 2.13: A 10-Barbell graph as example of highly regular and hierarchical network made up of two distinct modules. The idea behind this kind of graph consists in replicating a complete graph of n vertices and connecting them by means of a single link [84].

To summarise this brief introduction, I would refer the reader to Figure 2.14 where the various types of network are spatially arranged according to three classification criteria only: randomness, heterogeneity and modularity. Interestingly enough, networks with a biological meaning tend to be highly random or with a scale-free configuration [85, 86], probably owing to the need of rapidly sending information through the underlying graph.

Biological groundedness: a mere chimera or an opportunity?

The problem of establishing whether a biological network is random or not is far from being trivial. In fact, randomness in such networks might be the reflection of an internal organisational tendency that could be hidden by it. In principle, these questions are just apparently contradictory. What seems to be common in many biological networks is the preferential attachment behaviour that typically arises in scale-free networks [87, 88]. Because of the way of connecting nodes, it turns out that bigger hubs[22]

[21] A power-law distribution occurs when the fraction of nodes having degree k is distributed exponentially as $P(k) \sim k^{-\gamma}$, where $2 < \gamma < 3$ typically.

[22] I refer to those nodes with the highest degrees as *hubs*. Even though their importance is due to their huge amount of connections, it depends on the domain of application and therefore it is drawn from the context.

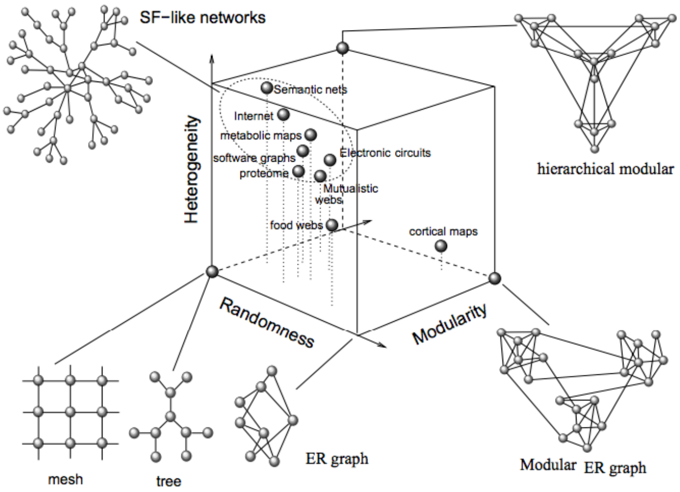


Figure 2.14: Types of network and their classification based on how random/heterogeneous/modular the network is [81].

are close to smaller ones and this leads to an intrinsic robustness to failure, even in biological networks [89]. On the other hand, there are other cases where connections are not as random as before. For example, in [90] it was shown how axonal connections in the mushroom bodies of locusts are spatially arranged in honeycomb-like structures. In this case, the underlying graph belongs to the class of the so-called lattice graphs and these structures do not make any exception[23] . There could be many reasons why a certain phenomenon should produce highly patterned and regular structures, such as the well-known reaction-diffusion process [92], but establishing the reasons why regular graphs in neural networks should appear is still not clear.

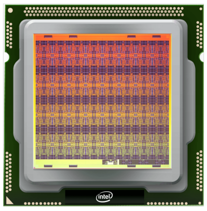


Figure 2.15: Intel Loihi, an example of 128-neuromorphic cores IC fabricated on 14 nm process by Intel for asynchronous SNN programming. It comprises a total of 130.000 neurons 130 million synapses.

In a more general perspective, when speaking about the future models of neural networks, the so-called 3rd generation of networks, some people claim that SNNs will gain more and more popularity till reaching the new *de facto* standard for neural networks. It is true that the gap between SNNs and more artificial ones is increasingly vanishing [93] but there are still some problems regarding both implementability (unless designers have neuromorphic platforms at their disposal, simulating a SNN usually takes more computational resources than its 2nd generation counterpart) and goodness of the outcomes, thereby one may argue that whenever it comes to defining intelligent machines the need of adopting biologically plausible models is meaningless if not totally useless. Not surprisingly, SNNs are mostly used in more theoretical scenarios, where the essential crux of the matter is given by the investigation into the neural mechanisms *per se*.

Figure 2.16: Intel Pohoiki Springs is currently the latest Intel Loihi-based system for neuromorphic computation. With its 768 Intel Loihi chips, this board can reproduce 100 million neurons but consumes just 500 W of power. Currently, it is the closest programmable device to a (small) mammal brain.

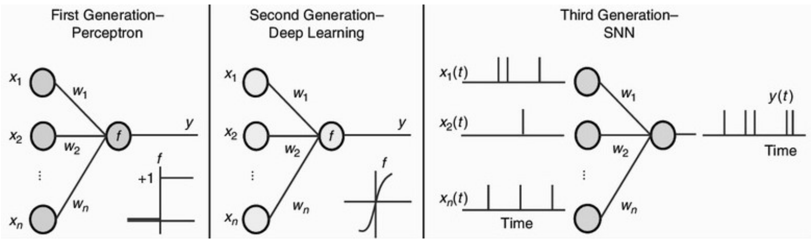


Figure 2.17: Generational evolution of neural networks. One of the most striking arguments against the 3rd generation of neural networks regards performance: so far, 2nd generation outperforms the 3rd one almost always in many applications and asking what even a 4th generation will be like is practically a pie in the sky.

[23] In this specific case, space is partitioned so that identical hexagons regularly occupy it. According to [91] there are at least three irregular hexagonal tilings.

Modelling and control 3

Modelling of natural phenomena constitutes a striking example of how challenging both the analysis and the control of complexity might be. In fact, nature has proved to be a great source of inspiration for engineering applications but a burdensome cradle of complexity as well for which common modelling techniques may not be appropriate.

3.1 Neural-like oscillatory dynamics

To be interested in oscillatory behaviours is not a waste of time, since many natural systems behave like that [94]. However, they arise from non-linear dynamical systems most of the time and can be reproduced thanks to some interesting properties drawn from their state space topology. To better understand this fact, one ought to have a clear idea of what manifolds are and what their role in modelling complex dynamics is. These prefatory details are necessary to grasp the way the nullcline-based algorithm for customisable slow-fast dynamics works.

Slow-fast systems as paradigmatic lodestar

A very common set of methods and mathematical details that proposes to clarify the behaviour of such systems, at least near their fixed points, is the *centre manifold theory* [95]. To begin with, *manifolds* are mathematical objects and as such they require a proper formalism. Intuitively, a manifold is a space that is locally Euclidean, but globally it might be more complicated, for example resulting in a torus or a sphere. In other words, given a point on a manifold its neighbourhood is constituted by all those points that are *homeomorphically*[1] related to an Euclidean space of the same dimension of the manifold. More formally, if $N \in \mathbb{N}$ then a N-manifold is a Hausdorff space where each point has a homeomorphic neighbourhood to $\mathring{\mathbb{D}} \doteq \left\{\mathbf{x} \in \mathbb{R}^N : \|\mathbf{x}\| < 1\right\}$, which is a N-dimensional, real-valued open ring [96].

Suppose the system:

$$\Sigma : \begin{cases} \dot{\mathbf{x}} & = & \mathbf{A}\mathbf{x} + \mathbf{F}(\mathbf{x}, \mathbf{y}) \\ \dot{\mathbf{y}} & = & \mathbf{B}\mathbf{y} + \mathbf{G}(\mathbf{x}, \mathbf{y}) \end{cases} \tag{3.1}$$

where $(\mathbf{x}, \mathbf{y}) \equiv (\mathbf{x}(t), \mathbf{y}(t)) \in \mathbb{R}^n \times \mathbb{R}^m$ for some $n, m \in \mathbb{N}$, $\forall t$ and such that $\mathbf{F}(\mathbf{0}, \mathbf{0}) = \mathbf{G}(\mathbf{0}, \mathbf{0}) = \mathbf{J}_{\mathbf{F}}(\mathbf{0}, \mathbf{0}) = \mathbf{J}_{\mathbf{G}}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$. When $\mathbf{F}(\mathbf{x}, \mathbf{y}) = \mathbf{G}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, $\forall(\mathbf{x}, \mathbf{y})$, then $\mathbf{x} = \mathbf{y} = 0$ are two trivial manifolds. If $\Re\{\lambda\} = 0$, $\forall \lambda \in \mathrm{Spec}(\mathbf{A})$ and $\Re\{\lambda\} < 0$, $\forall \lambda \in \mathrm{Spec}(\mathbf{B})$, then on the $\mathbf{x}$-manifold all solutions decay to zero exponentially fast. In particular, $\mathbf{x} = \mathbf{0}$ is called *stable manifold*, whereas $\mathbf{y} = \mathbf{0}$ is called *centre manifold*. More generally, a centre manifold $\mathbf{y} = \mathbf{H}(\mathbf{x})$, with $\mathbf{H}(\mathbf{0}) = \mathbf{J}_{\mathbf{H}}(\mathbf{0}) = \mathbf{0}$, is defined for small

[1] If two distinct sets $\mathbb{A}$ and $\mathbb{B}$ can be related through a bijective function $f \in C^0 : f^{-1} \in C^0$, then $\mathbb{A}$ and $\mathbb{B}$ are homeomorphic and f (or f^{-1}) is a homeomorphism.

$\|\mathbf{x}\|$ so that the $\mathbf{y}$-dynamics follow the $\mathbf{x}$-dynamics. This statement can be rewritten alternatively as follows:

Theorem 3.1.1 *Let Σ be a system governed by a set of equations as those shown in Equations 3.1, where $\mathbf{F}, \mathbf{G} \in C^2$, such that $\Re\{\lambda\} = 0, \forall\lambda \in \mathrm{Spec}(\mathbf{A})$ and $\Re\{\lambda\} < 0, \forall\lambda \in \mathrm{Spec}(\mathbf{B})$. Thereby, $\exists\mathbf{H}(\mathbf{x}) \in C^1$, with $\|\mathbf{x}\| < \delta$ where $\delta > 0$, such that $\mathbf{H}(\mathbf{x})$ is a centre manifold.*

Proof. Refer to [97]. □

Moreover, a manifold $\mathbf{H}(\mathbf{x}) = \mathbf{0}$ is said *invariant* when the following condition holds:

$$\mathbf{H}(\mathbf{x}(0)) = 0 \Rightarrow \mathbf{H}(\mathbf{x}(t)) = 0 \quad \forall t \geq 0 \tag{3.2}$$

When $\mathbf{y}(0) = \mathbf{H}(\mathbf{x}(0))$, *id est* the initial conditions are located inside the centre manifold itself, the centre manifold is invariant and thereby Σ can be reformulated as:

$$\Sigma : \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{F}(\mathbf{x}, \mathbf{H}(\mathbf{x})) \tag{3.3}$$

that is intrinsically dimensionally smaller than the system described by Equations 3.1. Additionally, the reduced system shown in Equations 3.3 plays a key role when evaluating stability of a non-linear system, especially when the so-called *Lyapunov's indirect method* for assessing stability fails. This theorem states:

Theorem 3.1.2 (Lyapunov's indirect method) *Let $\mathbf{x} = \mathbf{F}(\mathbf{x})$ be an autonomous, non-linear system whose state function is defined over a n-dimensional space $\mathbb{X}$ so that $\mathbf{F} : \mathbb{X} \longrightarrow \mathbb{X}$. Additionally, suppose that $\mathbf{F} \in C^1$ and $\bar{\mathbf{x}}$ is a fixed point. Thereby:*

- *$\bar{\mathbf{x}}$ is asymptotically stable if all the eigenvalues of $\mathbf{J}(\bar{\mathbf{x}})$ have strictly negative real parts*
- *$\bar{\mathbf{x}}$ is unstable if there is at least one eigenvalue of $\mathbf{J}(\bar{\mathbf{x}})$ whose real part is strictly positive*

Proof. Refer to [97]. □

As already written before, when the Lyapunov's indirect method is not applicable, for example when there is at least one eigenvalue of the Jacobian matrix whose real part is 0, the centre manifold theory I have introduced before can overcome this limitation:

Theorem 3.1.3 *Supposing that the hypotheses of Theorem 3.1.1 are verified, if the state $\mathbf{x} = \mathbf{0}$ is an asymptotically stable (unstable) fixed point for the reduced system in Equations 3.3, then so it is for the full system in Equations 3.1 as well.*

Proof. Refer to [97]. □

Singularly perturbed systems

In order to introduce how slow-fast dynamics work, it is necessary to give more detail about a specific class of dynamical systems of great interest in this field, whose generic expression is given by:

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{F}(t, \mathbf{x}, \mathbf{y}, \varepsilon) \\ \varepsilon\dot{\mathbf{y}} &= \mathbf{G}(t, \mathbf{x}, \mathbf{y}, \varepsilon) \end{cases} \tag{3.4}$$

where $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$. Here, $\varepsilon \longrightarrow 0^+$ is a very small positive parameter that tunes how abrupt changes in time scale are between $\mathbf{x}$ and $\mathbf{y}$. Equations 3.4 describe a $(n+m)$-order system, but if $\varepsilon = 0$ then its order reduces to n only because of the either algebraic or transcendental constraint given by:

$$\mathbf{0} = \mathbf{G}(t, \mathbf{x}, \mathbf{y}, 0) \tag{3.5}$$

Equations 3.4 constitute the broadest version of the so-called *singularly perturbed models*, whose main feature is given by multiple time scales occurring when feeding external signals into the system. This variety of temporal behaviours is manifested through the emergence of both slow and fast responses within the same system. Of course, if Equations 3.4 are time-invariant and do not depend on ε, they can be rewritten in a simpler form as

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{F}(\mathbf{x}, \mathbf{y}) \\ \varepsilon\dot{\mathbf{y}} &= \mathbf{G}(\mathbf{x}, \mathbf{y}) \end{cases} \tag{3.6}$$

and Equation 3.5 becomes:

$$\mathbf{0} = \mathbf{G}(\mathbf{x}, \mathbf{y}) \tag{3.7}$$

Slow and fast manifolds

To introduce this crucial topic, I would recall an example drawn from [98] to better explain it. Let

$$\begin{cases} \dot{x} &= 1 \\ \varepsilon\dot{y} &= -y + \varepsilon f(x) \end{cases} \tag{3.8}$$

with $x(0) = y(0) = 1$ and $f(x) \in C^\infty$ is a scalar function. Straightforwardly $\varepsilon = 0 \Rightarrow y = 0$, meaning that the resulting solution is given by:

$$\begin{cases} x(t) &= t+1 \\ y(t) &= 0 \end{cases} \tag{3.9}$$

but it is clear that $y(t)$ does not satisfy its initial condition. What the theory of singularly perturbed systems says is that $y(t)$ must change suddenly when it is close to a neighbourhood of $t = 0$, after passing a kind of threshold that is called *boundary layer*. Theoretically speaking, the time spent when such trajectories trespass a boundary layer is 0, or at least infinitesimal.

A preliminary result for introducing the concept of slow and fast manifolds is given by the *Tikhonov's theorem*:

Theorem 3.1.4 (Tikhonov's theorem) *Suppose a dynamical system as follows:*

$$\begin{cases} \dot{\mathbf{x}} & = & \mathbf{F}(\mathbf{x}, \mathbf{y}, t) \\ \varepsilon\dot{\mathbf{y}} & = & \mathbf{G}(\mathbf{x}, \mathbf{y}, t) \end{cases} \tag{3.10}$$

Let $\mathbf{y} = \boldsymbol{\phi}(\mathbf{x}, t)$ *be the solution of the constraint* $\mathbf{0} = \mathbf{G}(\mathbf{x}, \mathbf{y}, t)$ *when* $\varepsilon = 0$. *As long as* $\varepsilon \longrightarrow 0^+$ *the solution of the given system approaches to the solution of the reduced system* $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \boldsymbol{\phi}(\mathbf{x}, t), t)$ *if* $\mathbf{y} = \boldsymbol{\phi}(\mathbf{x}, t)$ *is a stable root of* $\dot{\mathbf{y}} = \mathbf{G}(\mathbf{x}, \mathbf{y}, t)$.

Proof. Refer to [99]. □

The Tikhonov's theorem implies the existence of two types of manifolds: $\mathbf{x}(t)$ is the equation of a slow manifold, whilst $\mathbf{y}(t)$ is the fast counterpart.

Figure 3.1: Andrey Nikolayevich Tikhonov (1906-1993), a Soviet and Russian mathematician and geophysicist known for his contributions in different areas of mathematics, such as topology, ill-posed problems and functional analysis.

Phase plane analysis

In two-dimensional dynamical systems, it is particularly interesting to analyse how they behave in time by considering their evolution on the phase plane, which is a common and suitable representation of the state flow as parameterised curves.

An important, dynamical system for the nullcline-based algorithm I have already mentioned is the FitzHugh-Nagumo model, fully described by the following set of equations [100]:

$$\begin{cases} \dot{u} & = & c\left(w + u - \frac{1}{3}u^3 + z\right) \\ \dot{w} & = & -\frac{1}{c}(u - a + bw) \end{cases} \tag{3.11}$$

where u is the membrane potential-like variable, w is a recovery variable, z is an external input and a, b, c are fixed parameters so that:

$$\begin{cases} 1 - \frac{2}{3}b & < & a & < & 1 \\ 0 & < & b & < & 1 \\ & & b & < & c^2 \end{cases} \tag{3.12}$$

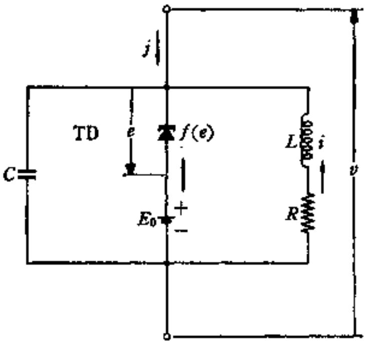


Figure 3.2: Circuit scheme of a FitzHugh-Nagumo neuron [102].

This model is one of the most known reductions aimed at describing the properties of the original Hodgkin-Huxley model [101] and since its publication it was acclaimed for its computational simplicity and capability of showing non-trivial outcomes. To some extent, it is possible to think of it as a slow-fast system where $\varepsilon \equiv \frac{1}{c}$, thereby as long as $c \longrightarrow +\infty$ the relative speed between the two nullclines changes drastically so that $\left|\frac{\dot{u}}{\dot{w}}\right|$ is weighted by a constant speed ratio of $\frac{1}{\varepsilon^2}$. No surprise, then, if the u-nullcline (w-nullcline) is the fast (slow) manifold of the system.

One of the greatest advantages of the FitzHugh-Nagumo model regards its easiness of interpretability, since neuronal dynamics can be viewed on the phase plane directly as the result of proper stimulation. Like the

Hodgkin-Huxley model, the FitzHugh-Nagumo model does not have a fixed firing threshold because both models lack in terms of all-or-none responses[2] . Additionally, a preeminent aspect of such systems regards the existence of limit cycles, as shown and deeply described by the *Poincaré-Bendixson theorem* [103] and shown in Figure 3.4.

[2] Mathematically speaking, this corresponds to have no saddle equilibria and thereby the "illusion" of a firing threshold is given by the emergence of a canard explosion that arises when the input z is sufficiently big to let the trajectory overcome the resting basin shown in Figure 3.3.

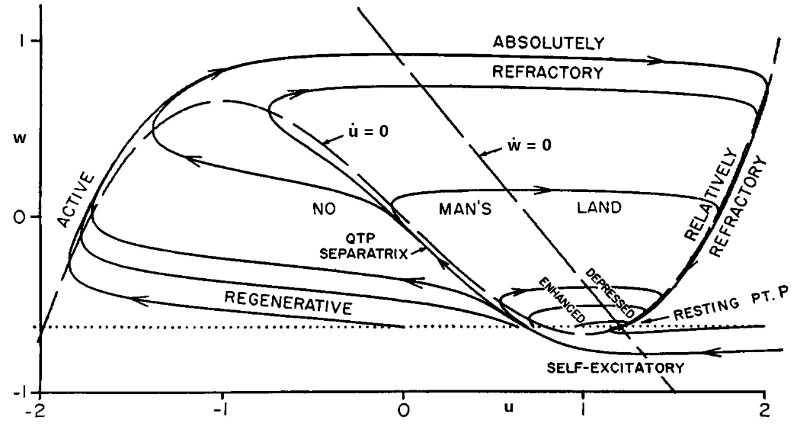


Figure 3.3: Phase plane of the FitzHugh-Nagumo model ($a = 0.7$, $b = 0.8$, $c = 3$ and $z = 0$).

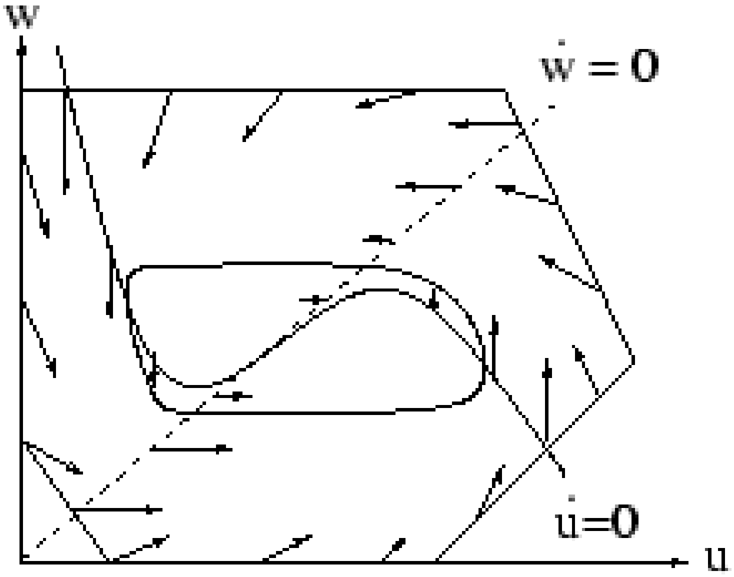


Figure 3.4: Bounding surface for the existence of limit cycles in the FitzHugh-Nagumo model. According to the Poincaré-Bendixson theorem, a limit cycle can emerge if a boundary surface that surrounds a repulsive fixed point can be constructed so that the state flow "points" towards the interior.

Nullcline-based timing

It is clear that nullclines are strictly related to oscillations, because whenever the system 3.11 evolves along the trajectories defined by its limit cycle and makes a complete turn, a whole period is completed. However, when dealing with slow-fast systems the calculation of a period can be simplified a lot.

Let T be the period of a single oscillation. Owing to the presence of two distinct branches, namely the slow and the fast manifolds within the system, one can assume that $T = T_{\text{fast}} + T_{\text{slow}}$ if T_{fast} (T_{slow}) is the time spent by the system when it travels along the fast (slow) manifold. Now, suppose a cycle like the one depicted in Figure 3.5, where $G_{-}(w)$ and $G_{+}(w)$ are the descending and ascending portions of the whole slow manifold, respectively. The fastest manifold is given by the union of the two (either exactly or almost, depending on ε) horizontal segments, where $T_{\text{fast}} = 0$ (or $T_{\text{fast}} \to 0$ at least), thereby:

$$\lim_{\varepsilon \to 0} T = T_{\text{slow}} \tag{3.13}$$

Thus, to calculate the period one should only evaluate how the system behaves within the slow manifold. Getting the results reported in [104] back, this procedure is conceptually easy, because one can evaluate T_{slow} by means of the following notation:

$$\begin{aligned} \mathbb{S}_- &\doteq \left\{ w \equiv w(t) \in \mathbb{R} | \frac{\partial f(u,w)}{\partial u} < 0 \right\} \\ \mathbb{S}_+ &\doteq \left\{ w \equiv w(t) \in \mathbb{R} | \frac{\partial f(u,w)}{\partial u} > 0 \right\} \end{aligned} \tag{3.14}$$

so that:

$$\begin{cases} G_-(w) = \dot{w} & \text{with } w \in \mathbb{S}_- \\ G_+(w) = \dot{w} & \text{with } w \in \mathbb{S}_+ \end{cases} \tag{3.15}$$

Therefore:

$$T_{\text{slow}} = \int_{\mathbb{S}_-} \frac{dw}{G_-(w)} + \int_{\mathbb{S}_+} \frac{dw}{G_+(w)} \simeq T \tag{3.16}$$

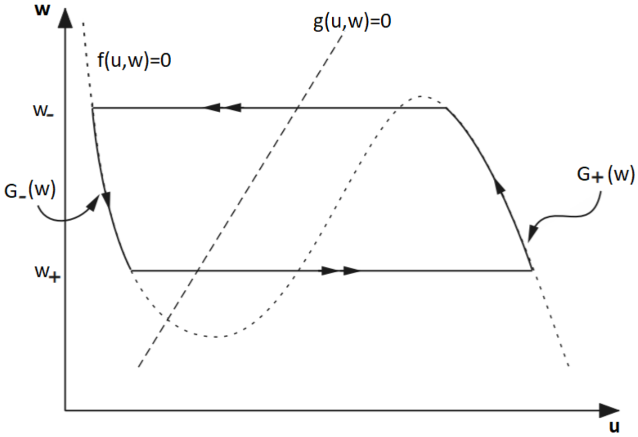


Figure 3.5: A generic slow-fast system with a cubic-shaped nullcline, such as the FitzHugh-Nagumo model, evolves along two main branches. Horizontal segments denote those "jumps" that occur when the system passes from the fastest manifold to the slowest ones and the time spent here is practically irrelevant because of the small ε.

Equation 3.16 indicates a way to calculate the period of oscillation given the nullclines of the system. However, the calculation of each integral may be tricky and not so immediate, therefore one may think to simplify them by replacing the nullclines with something else, easier but topologically equivalent at the same time. In this context, a suitable approximation that can be tuned as wished in order to fit the same topological properties of an original nullcline is a piecewise function[3] .

To begin with, let us introduce a function $\Pi(x)$ to approximate a nonlinear expression as the union of multiple segments each one having its own slope. If the number of segments is equal to $n+1$, then the number of break-points is n; furthermore, the i-th slope is referred to as $m_i \in \mathbb{R}$, with $i \in \{0, ..., n\}$ being m_0 the slope of the leftmost segment and m_n the

[3] 3: Two dynamical systems are topologically equivalent if their fixed points are qualitatively similar, despite their own analytical formulations. This implies that if a fixed point is asymptotically stable for the first system, then it is asymptotically stable for the second system too.

slope of the rightmost one [105]. Thus, the most generic form of $\Pi(x)$ is:

$$\Pi(x) = a_0 + a_1 x + \sum_{j=1}^{n} b_j \left|x - e_j\right| \tag{3.17}$$

where $\Pi(0) \in \mathbb{R}$ and $a_0, a_1, b_j, e_j \in \mathbb{R}$, with $j \in \{1, \ldots, n\}$, depend on the slopes of the segments:

$$\begin{cases} a_0 = & \Pi(0) - \sum_{j=1}^{n} b_j \left|e_j\right| \\ a_1 = & 0.5(m_0 + m_n) \\ b_j = & 0.5(m_j - m_{j-1}) \end{cases} \tag{3.18}$$

Thereby, $\Pi(x)$ is a piecewise-linear function (PWL) with tunable parameters which does not depend on the original function to be approximated. In fact, $\Pi(x)$ can be calculated in an iterative way by simply partitioning the domain of a given function $f(x)$ to approximate, where $x \in [x_{\min}, x_{\max}]$, so that $f(k\Delta x + x_{\min}) \doteq y[k]$ is the $(k+1)$-th value obtained by sampling the function with a sampling interval equal to Δx and $k \in \left\{0, \ldots, \frac{x_{\max} - x_{\min}}{\Delta x}\right\}$. Therefore, it immediately follows that $m_k \equiv \frac{y[k+1] - y[k]}{\Delta x}$ and $e_{k+1} \equiv (k+1)\Delta x + x_{\min}$.

Thanks to PWL functions, one may replace an entire nullcline with something easier to handle from a computational standpoint, especially to solve Equation 3.16 since it may contain non-linear functions within each integral it is made up of. In fact, the essential idea is to replace both $G_{-}(w)$ and $G_{+}(w)$ with affine functions, whose integration is certainly more immediate. For the sake of the next results, I will suppose a slow-fast system in the form of:

$$\begin{cases} \dot{u} = f_1(u, w) \equiv & \frac{1}{\varepsilon}\left[\Pi(u) - w\right] \\ \dot{w} = f_2(u, w) \equiv & au + bw + c \end{cases} \tag{3.19}$$

The first nullcline gives a PWL function $w = \Pi(u)$ which can be thought of as a series of segments in the form of affine functions $w = m_* u + q_*$ for some $m_*, q_* \in \mathbb{R}$ with $m_* \neq 0$ that change depending on the portion of interest of the whole cycle (in other words, $* \equiv -$ for the leftmost branch, $* \equiv +$ otherwise). If $w = m_* u + q_*$, then $u = \frac{1}{m_*} w - \frac{1}{m_*} q_*$ and therefore $f_2(\frac{1}{m_*} w - \frac{1}{m_*} q_*, w) \equiv G_*(w) = \left(b + \frac{a}{m_*}\right) w - \left(c + \frac{a}{q_*}\right)$, that can be rewritten in a more compact form as $G_*(w) = M_* w + Q_*$ if $M_* \equiv (b + \frac{a}{m_*})$ and $Q_* \equiv -\left(c + \frac{a}{q_*}\right)$. By means of this reduction, each integral in Equation 3.16 is easily solvable:

$$\begin{cases} T_{\text{slow}}^{\text{left}} & \equiv \int_{\mathbb{S}_-} \frac{dw}{G_-(w)} & = \int_{w_-}^{w_+} \frac{dw}{M_- w + Q_-} & = \frac{1}{M_-} \log\left(\frac{M_- w_+ + Q_-}{M_- w_- + Q_-}\right) \\ T_{\text{slow}}^{\text{right}} & \equiv \int_{\mathbb{S}_+} \frac{dw}{G_+(w)} & = \int_{w_+}^{w_-} \frac{dw}{M_+ w + Q_+} & = \frac{1}{M_+} \log\left(\frac{M_+ w_- + Q_+}{M_+ w_+ + Q_+}\right) \end{cases} \tag{3.20}$$

By means of Equations 3.20 it is possible to determine the whole period of oscillation with a simple and suitable PWL approximation given by

Equation 3.17, whose main idea consists in partitioning a non-linear dynamic into multiple linear sub-parts, each describable thanks to 1-st order polynomials. Observe that $T_{\text{slow}}^{\text{left}} \equiv T_{\text{slow}}^{\text{left}}(m_{-})$ and $T_{\text{slow}}^{\text{right}} \equiv T_{\text{slow}}^{\text{right}}(m_{+})$.

Equations 3.20 are particularly useful when it comes to determining the period of a single oscillation and therefore to implementing more advanced operations, such as synchronisation. For example, one may implement them to create dynamical controllers that allow a slave system to follow a pre-defined timing. Owing to their relationship with the external slopes that model both the left and the right branches, it is possible to alter the period of oscillation by making both the slopes (or just one of them) steeper or flatter.

3.2 Analysis and modelling of plasma dynamics

Plasma arises spontaneously as state of matter in those fluids that reach the thermo-dynamical equilibrium and whose temperature is sufficiently high. However, the transition from gas to plasma by means of over-heating is not as discontinuous as in other state transitions[4] , meaning that this state is reachable whenever the ratio between thermal collisions and electro-statical forces is increased. To put it simply, plasma is an ionised gas characterised by ubiquitous electric fields that allow the particles to collide, as well as the natural tendency of particles to collision owing to temperature. This kind of state of matter exists in vacuum only, otherwise air would cool the plasma and thereby particles would turn into neutral atoms again. Observe that not all ionised gases can be thought of as plasma: to be more precise, plasma is therefore a quasi-neutral gas made of charged and neutral particles capable of showing a collective behaviour, where "quasi-neutral" means that ions and electrons are compelled to move without any sensible charge separation (at least, macroscopically speaking) and "collective behaviour" refers to that situation where particles generate collateral effects due to the concentration of positive/negative charge, which in turn produces magnetic fields that thereby affect motion [106].

[4] This is a well-known concept in thermodynamics, where more common state transitions occur by means of heat exchanges. Technically speaking, the amount of exchanged heat is called *latent heat*.

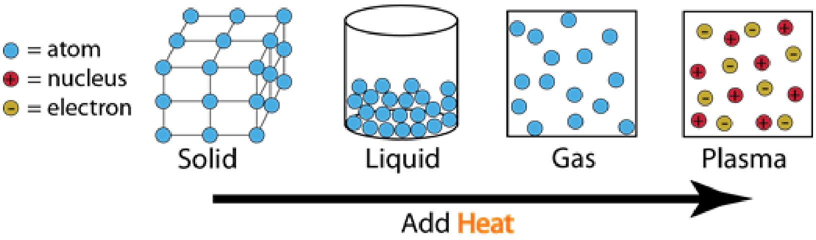


Figure 3.6: Plasma reactions and how they happen. To produce energy by means of plasma, fusion reactions must be both exothermic and occur amongst atoms whose atomic numbers ought to be as low as possible. Two possible candidates are deuterium and tritium, but they must be reduced to highly energetic plasma (about 10 KeV) to produce energy. More in detail, in order to create a significant amount of energy the Lawson's criterion must hold [107].

On the problem of tomographic reconstruction

Amongst the variety of issues designers and engineers may have to deal with, tomographic reconstruction constitutes an interesting example of well-known, but ill-posed problem. Introduced by Radon [108], who is also credited with the integral transform named after him, and despite being relatively familiar in plenty of real contexts, such as computed tomographies [109], its conceptual core remains particularly burdensome in terms of numerical realisation [110, 111]. Generally speaking, a tomographic reconstruction consists in getting a 2D or 3D profile of a given physical quantity of interest thanks to a finite number of projections over straight lines. Intuitively, the higher the number of projections is, the more detailed the resulting profile is owing to the overall contribution of the more numerous line integrals. As long as the integration operates over a very high number of lines, numerical posedness becomes increasingly better and this is perfectly coherent with the Radon's statement claiming that his transform is well-posed for an infinite number of data. That having been said, not surprisingly there is a clear intention of handling the problem in order to make it more computational affordable and significantly more reliable, especially when there is an intrinsic lack of data the reconstruction algorithm has to work on. A relevant scenario in this sense is given by nuclear fusion and what happens in Tokamak machines [112–114]. To put it simply, these devices produce a toroidal magnetic field for plasma confinement and subsequent energy production. One of the main problems of plasma fusion is the high temperature that ions and electrons reach, thereby leading to large velocities. In order to maintain the fusion process, particles from the hot plasma must be confined in the central region by means of the Lorentz's force, otherwise the plasma will rapidly cool.

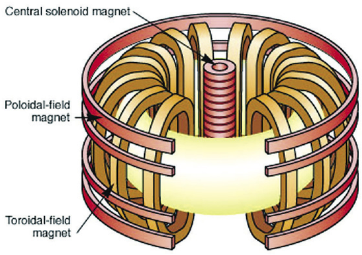


Figure 3.7: Scheme of a Tokamak machine. This term is a transliteration of the Russian words "toroidal'naya kamera s magnitnymi katushkami", or "toroidal chamber with magnetic coils" in English and refers to a system invented in the 1950s by Soviet physicists Igor Yevgenyevich Tamm and Andrei Sakharov (who were in turn inspired by an original idea of Oleg Lavrentyev).

Solution to the inverse problem

From a purely mathematical perspective, the problem of tomographically reconstructing a spatial function from one-dimensional projections may be severely ill-posed, as already written before. The conceptual key to understanding of how it could be solved lies on some fundamental integral transforms that relate both the resulting, spatial function and its projections, namely the Abel's transform (Equation 3.21), the Fourier's transform (Equation 3.22) and the Hankel's transform (Equation 3.23):

$$\mathcal{A}\left\{\alpha(r)\right\}(x) \doteq 2\int_{x}^{+\infty} \frac{r\alpha(r)dr}{\sqrt{r^2 - x^2}} \tag{3.21}$$

$$\mathcal{F}\left\{\zeta(x)\right\}(\rho) \doteq \int_{-\infty}^{+\infty} \zeta(x)e^{-2\pi i\rho x}dx \tag{3.22}$$

$$\mathcal{H}_\nu\left\{\eta(\rho)\right\}(k) \doteq \int_{0}^{+\infty} \eta(\rho)J_\nu(k\rho)\rho d\rho \tag{3.23}$$

where J_ν is the first type Bessel's function[5] of order $\nu \geq -\frac{1}{2}$. Together, they constitute the so-called FHA cycle, which is the basis of the projection-slice theorem [115]. This theorem states that the 1D Fourier transform over a 1D projection of a 2D function exactly equals the slicing (*id est*, the act of extracting a 1D central slice from a 2D function) of the 2D Fourier transform of the same function. This is particularly helpful because the inverse problem, namely the determination of $\alpha(r)$ from its Abel's transform, is difficult to solve, since the analytical expression of the inverse Abel's transform, namely:

$$\alpha(r) = -\frac{1}{\pi}\int_{r}^{+\infty} \frac{(d\mathcal{A}\left\{\alpha(r)\right\}(x))/dx}{\sqrt{x^2 - r^2}}dx \tag{3.24}$$

has not any real, practical relevance. In fact, most of the time $\mathcal{A}\left\{\alpha(r)\right\}(x)$ is available either at some points only or not at all and it may be corrupted by noise. However, when the function to be processed is circularly symmetric it can be proved [116] that:

$$\mathcal{F}\left\{\mathcal{A}\left\{\alpha(r)\right\}\right\} = \mathcal{H}_\nu\left\{\alpha(r)\right\} \tag{3.25}$$

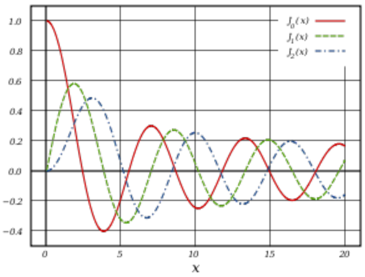


Figure 3.8: Bessel functions for $n = 0, 1, 2$.

[5] The Bessel's functions of the first type $J_\nu(x)$, also known as cylindrical harmonics, are the non-trivial solutions of the differential equation $x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + (x^2 - \nu^2)y = 0$.

and thereby implying that:

$$\mathscr{A}^{-1}\left\{\alpha(r)\right\} = \mathscr{H}_{\nu}^{-1}\left\{\mathscr{F}\left\{\alpha(r)\right\}\right\} \tag{3.26}$$

Observe that the inverse Hankel's transform of a generic function $F_\nu(k)$ is given by:

$$\mathscr{H}_{\nu}^{-1}\left\{F_\nu(k)\right\} = \int_0^{+\infty} F_\nu(k) J_\nu(kr) k dk \tag{3.27}$$

which is practically equivalent to Equation 3.23 except for $F_\nu(k)$ in place of $\eta(\rho)$.

On the identification of the input-output FTU model

Instead of being interested in reconstructing a plasma profile, one would determine a (computational) model of a Tokamak machine when working at steady, regular conditions. This may lead to (among other things):

- a better understanding of the underlying phenomena that regulate plasma production;
- a further tool for modelling and analysis of complex dynamics.

It is worth noticing that modelling plasma dynamics is not a novelty. In [117] authors attempted to model a Tokamak device by means of the H_∞ identification procedure[6] , whilst in [118] a linearised, yet non-rigid MHD[7] consistent displacement model was introduced to fit the study of the vertical stability of a plasma in either an air-core or iron-core Tokamak.

Prediction Error Method for grey-box models

Unlike other forms of model, grey-box ones are partially known: what it is known is given by their internal structure, thereby this implies that grey-box identification aims to find the best parameter space according to both the given structure and input-output relationships. In other words, with grey-box models theoretical aspects and data-grounded issues are combined together to provide a more complete depiction of a phenomenon. The problem of determining the parameters of such models could be particularly onerous, but fortunately there are different strategies to solve it. In this paragraph, I will introduce the very fundamental ideas of the Prediction Error Method (PEM) [119].

Unlike other approaches like those based on likelihood maximisation, where the modelling accuracy is given by the mismatch between the estimated model and the expected one (which is essentially statistical), PEM is probably the most system-oriented because it determines the overall modelling goodness in terms of how close the outcomes produced by the estimated model to the observations are. Providing a model:

$$y = g(x, \theta_0) + \epsilon \tag{3.28}$$

[6] H_∞ identification leads to a linear representation of an unknown system and relies on the calculation of the homonymous norm. Generally it is employed to create even approximated models that abide by the Hankel's approximation, meaning that if $G(s)$ is a continuous transfer function whose singular values are $\sigma_1 \geq \ldots \geq \sigma_n$and $\tilde{G}(s)$ is an approximated version of it, then a r-th order model ($r < n$) leads to the disequalities $\sigma_{r+1} \leq \left\|G(s) - \tilde{G}(s)\right\|_\infty \leq \sum_{i=r+1}^{n} \sigma_i$.

[7] MHD stands for magnetohydrodynamics, namely the study of the magnetic properties and behaviour of electrically conducting fluids, such as plasma or electrolytes.

where $x \equiv x[t]$, $\epsilon \equiv \epsilon[t]$ and thus $y \equiv y[t]$, the problem I am interested in is calculating the parameter θ. To better point out the peculiarities of the PEM algorithm, I would compare it with other common strategies, namely the Least Squares (LS) and the Maximum Likelihood (ML) methods, as follows:

$$\begin{array}{lccc} \text{LS} & : & \hat{\theta} & \doteq & \underset{\theta}{\arg\min} \frac{1}{2} \sum_{t=1}^{n} \left[y[t] - g(x[t], \theta) \right]^2 \\ \text{ML} & : & \hat{\theta} & \doteq & \underset{\theta}{\arg\max} \log \text{Likelihood}(Y_1, \dots, Y_n, \theta) \\ \text{PEM} & : & \hat{\theta} & \doteq & \underset{\theta}{\arg\min} \frac{1}{2} \sum_{t=1}^{n-1} \left[y[t+1] - \hat{y}[t+1](y[1], \dots, y[t], \theta) \right]^2 \end{array} \tag{3.29}$$

The LS method is quite self-explanatory. On the other hand, ML aims to maximise the likelihood function of a stochastic model, namely a n-variate PDF, whilst PEM determines the best parameter so that the error between the output and its predicted value $\hat{y}$, which is supposed to be dependent on its previous values, is minimum. It has been shown that PEM can outperform other identification techniques, such as the Subspace Identification Methods (SIMs) proposed by Viberg in 1994 [120, 121]. For the sake of next considerations, let $J_n(\theta) \doteq \frac{1}{2} \sum_{t=1}^{n-1} \left[y[t+1] - \hat{y}[t+1](y[1], \dots, y[t], \theta) \right]$; the optimisation problem that PEM aims to solve is not generally solvable analytically, but it requires an iterative approach which consists in evaluating the function $J_n(\theta)$ for several $\theta \in \{\theta_1, \dots, \theta_m\}$ such that:

$$J_n(\theta_i) \geq J_n(\theta_{i+1}) \tag{3.30}$$

for each $i \in \{1, \dots, m\}$ until $J_n(\theta_m) \approx \hat{\theta}$ for some $m \in \{1, \dots, m\}$. Starting from an initial value θ_1, the parameter is updated as follows:

$$\theta_i = \theta_{i-1} + \Delta \tag{3.31}$$

where Δ is the incremental update that can be calculated in different ways:

$$\begin{array}{lccc} \text{Steepest descent} & : & \Delta & \equiv & -\gamma_i \nabla J_n(\theta_i) \\ \text{Newton-Raphson} & : & \Delta & \equiv & -\gamma_i \left[\nabla^2 J_n(\theta_i) \right]^{-1} \nabla J_n(\theta_i) \\ \text{Gauss-Newton} & : & \Delta & \equiv & -\gamma_i \mathbb{E} \left\{ \left[\nabla^2 J_n(\theta_i) \right]^{-1} \right\} \nabla J_n(\theta_i) \end{array} \tag{3.32}$$

Wavelet networks as non-linear estimators

What about the form of the predictor $\hat{y}$? Actually, there is not a specific criterion for selecting an estimator and thereby one has multiple choices. An interesting estimator I would discuss here is given by Wavelet Neural Networks (WNNs), which have gained an increasingly significance in the last years because of both their structure and concept. Additionally, they fit well the general idea of this thesis of using neural-like tools for solving various problems.

This approach consists in superimposing dilated and translated copies of a single function, namely the *mother wavelet*, localised in both the space and frequency domains in order to reproduce non-linearities. The combination of a sufficiently high number of these modified replica can actually lead to the approximation of any given input function, which makes the whole network a universal approximator. WNNs embed wavelets as activation functions, allowing them to be more easily trainable than classical multi-layered networks because of the spatial localisation of the wavelets [122]. A wavelet $\psi(t)$ is therefore adapted to a given signal by means of proper translations and dilations, which are given by tunable, real-valued parameters:

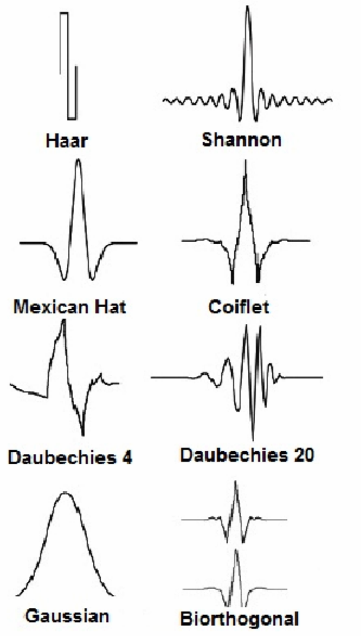


Figure 3.9: Most typical wavelets [123].

$$\psi_{a,b}(x) \doteq \frac{1}{\sqrt{a}}\psi\left(\frac{x-a}{b}\right) \tag{3.33}$$

where a (b) is the shifting (scaling) term. Each $\psi_{a,b}(t)$ is (not surprisingly) a *child wavelet* whose parameters are usually calculated by means of the back-propagation algorithm, although more exotic solutions, such as genetic programming [124] or hierarchical evolutionary algorithms [125], are allowed:

$$\left\{\begin{array}{ccc} \mu(\mathbf{z}) & = & e^{-\frac{1}{2}\mathbf{z}\mathbf{z}^{\mathrm{T}}} \\ \psi(\mathbf{z}) & = & (m-\mathbf{z}\mathbf{z}^{\mathrm{T}})\mu(\mathbf{z}) \\ \hat{y}(\mathbf{x},\theta) & = & (\mathbf{x}-\mathbf{r})\mathbf{P}\mathbf{L}+\sum_{i=1}^{n_\mu} a_i^{(\mu)}\mu\left(b_i^{(\mu)}\left[(\mathbf{x}-\mathbf{r})\mathbf{Q}-c_i^{(\mu)}\right]\right)+ \\ & + & \sum_{i=1}^{n_\psi} a_i^{(\psi)}\psi\left(b_i^{(\psi)}\left[(\mathbf{x}-\mathbf{r})\mathbf{Q}-c_i^{(\psi)}\right]\right)+d \end{array}\right. \tag{3.34}$$

where $\mu(\mathbf{z})$ is the scaling function whose argument is $\mathbf{z} \in \mathbb{R}^{1,q}$, θ is the parameter that ought to be estimated and $\mathbf{P}$ and $\mathbf{Q}$ are projection matrices drawn from the PCA of the estimation data[8] . Additionally, $\mathbf{r} \in \mathbb{R}^{1,m}$ is the mean of the estimation data. There are other parameters in Equations 3.34 with either the upperscript (μ) ($\mathbf{a}^{(\mu)} = \left[a_1^{(\mu)},\dots,a_{n_\mu}^{(\mu)}\right]$, $\mathbf{b}^{(\mu)} = \left[b_1^{(\mu)},\dots,b_{n_\mu}^{(\mu)}\right]$ and $\mathbf{c}^{(\mu)} = \left[c_1^{(\mu)},\dots,c_{n_\mu}^{(\mu)}\right]$), namely the shifting parameters, or the upperscript (ψ) ($\mathbf{a}^{(\psi)} = \left[a_1^{(\psi)},\dots,a_{n_\psi}^{(\psi)}\right]$, $\mathbf{b}^{(\psi)} = \left[b_1^{(\psi)},\dots,b_{n_\psi}^{(\psi)}\right]$ and $\mathbf{c}^{(\psi)} = \left[c_1^{(\psi)},\dots,c_{n_\psi}^{(\psi)}\right]$), that refer to the wavelet parameters. Eventually, d simply defines an offset. Observe that both $\psi(\cdot)$ and $\mu(\cdot)$ are radial functions[9] .

Hammerstein-Wiener models

A typical Hammerstein's model consists of a static, non-linear modeller followed by a linear dynamic element, whilst a Wiener's model behaves reversely so that the linear element precedes the static, non-linear characteristic. When these two forms are "sandwiched" so that the linear modeller is placed amid two non-linear functions, the resulting model, whose general structure is shown Figure 3.10, is a so-called Hammerstein-Wiener (HW) model [126, 127].

In HW models, input signals are processed throughout multiple elements each pursuing the known outcomes. Non-linearities within the model

[8] $\mathbf{P} \in \mathbb{R}^{m,p}$ and $\mathbf{Q} \in \mathbb{R}^{m,q}$ respectively, where $p \leq m$ (equality does not hold if there are linearly dependent components within the data). Furthermore, q denotes the number of components used in both μ and ψ.

[9] A function $f(\mathbf{x})$ defined on $\mathbb{R}^n$ is said radial if its value at each point depends only on the distance between that point and the origin. In other words, $f(\mathbf{x})$ is radial if $\exists \bar{f}(r) : f(\mathbf{x}) = \bar{f}(r)$ with $r = \|\mathbf{x}\|_2$.

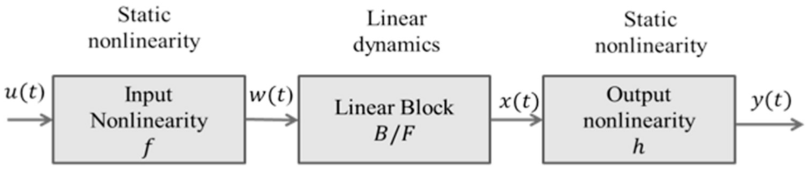


Figure 3.10: Structure of Hammerstein-Wiener models. non-linearities are given by pre-defined estimators, as well as the linear block in the form of a transfer function. It immediately follows that $w(t) = f(u(t))$ and $y(t) = h(x(t))$, while both $B(\cdot)$ and $F(\cdot)$ are polynomial functions whose orders are pre-defined.

process both the input and the output of the unique linear sub-system, which could easily be expressed as a collection of transfer functions:

$$\mathrm{TF}(p) \doteq \begin{bmatrix} \frac{\Sigma_{i=0}^{on} a_i^{(1,1)} p^i}{\Sigma_{i=0}^{od} b_i^{(1,1)} p^i} & \cdots & \frac{\Sigma_{i=0}^{on} a_i^{(1,nu)} p^i}{\Sigma_{i=0}^{od} b_i^{(1,nu)} p^i} \\ \vdots & \ddots & \vdots \\ \frac{\Sigma_{i=0}^{on} a_i^{(ny,1)} p^i}{\Sigma_{i=0}^{od} b_i^{(ny,1)} p^i} & \cdots & \frac{\Sigma_{i=0}^{on} a_i^{(ny,nu)} p^i}{\Sigma_{i=0}^{od} b_i^{(ny,nu)} p^i} \end{bmatrix} \tag{3.35}$$

where p coincides either with the usual Laplace complex variable s for continuous systems or with the complex variable z of the Z-transform otherwise. In other words, if Transform$(\cdot)$ denotes either the Laplace transform or the Z-transform, then $x(t) = \text{Transform}(w(t))$. In the previous notation, n_u and n_y refer to the number of input and output signals respectively, whilst o_n and o_d denote the orders of both the numerator and the denominator (generally, $o_d \geq o_n$). Observe that $\mathrm{TF}(p) \equiv \mathrm{TF}(p, \mathbf{A}, \mathbf{B})$, where

$$\mathbf{A} \doteq \begin{bmatrix} \mathbf{a}^{(1,1)} & \cdots & \mathbf{a}^{(1,n_u)} \\ \vdots & \ddots & \vdots \\ \mathbf{a}^{(n_y,1)} & \cdots & \mathbf{a}^{(n_y,n_u)} \end{bmatrix} \in \mathbb{R}^{n_y, n_u o_n} \tag{3.36}$$

$$\mathbf{B} \doteq \begin{bmatrix} \mathbf{b}^{(1,1)} & \cdots & \mathbf{b}^{(1,n_u)} \\ \vdots & \ddots & \vdots \\ \mathbf{b}^{(n_y,1)} & \cdots & \mathbf{b}^{(n_y,n_u)} \end{bmatrix} \in \mathbb{R}^{n_y, n_u o_d} \tag{3.37}$$

where $\mathbf{a}^{(j,k)} = \left[a_1^{(j,k)}, \ldots, a_{o_n}^{(j,k)}\right]$ and $\mathbf{b}^{(j,k)} = \left[b_1^{(j,k)}, \ldots, b_{o_d}^{(j,k)}\right]$ are the parameters estimated during the identification phase. A fundamental requirement for all the functions in Equation 3.35 is stability, meaning that if

$$\mathrm{Spec}^{(j,k)} \doteq \left\{ p \in \mathbb{C} : \sum_{i=0}^{o_d} b_i^{(j,k)} p^i = 0 \right\} \tag{3.38}$$

then the identification procedure must determine the matrices $\mathbf{A}$ and $\mathbf{B}$ so that $\forall j, k$ it follows:

$$\mathrm{Spec}^{(j,k)} \subseteq \begin{cases} \{s \in \mathbb{C} : \Re(s) \leq 0\} & p \equiv s \\ \{z \in \mathbb{C} : |z| \leq 1\} & p \equiv z \end{cases} \tag{3.39}$$

Regarding the non-linear estimators, one can adopt the preferred estimator depending on the purpose, meaning that even WNNs may work well. If so, it follows that both $w(t)$ and $y(t)$ are computed by implementing the system shown in Equation 3.34, thereby implying that both of them depend on some additional parameters that I have referred to as θ to be identified.

To sum up, a generic HW model is a combination of three operators in the form $\Sigma \doteq (f \circ \text{Transform} \circ h)$; if $\mathbb{S} \doteq \{\Sigma\}$ and $\text{Fitness}(y_{\text{ref}}(t), y(t))$ is a fitness function evaluated between the real output signal and the known outcome y_{ref}, then the problem of finding the best Σ corresponds to solving the following optimisation problem:

$$\exists\bar{\Sigma} : \bar{\Sigma} = \arg\max_{\Sigma \in \mathbb{S}} \left(\text{Fitness}(y_{\text{ref}}(t), y(t))\right) \tag{3.40}$$

3.3 Human-machine interaction in remote ultrasound scans

To put it simply and from a purely physical perspective, ultrasound is perfectly equal to the normal audible sound, except that humans cannot hear it. As normally happens in medical examinations, frequencies of ultrasound scans change according to the part of the body to scan (see Table 3.1), but how they are executed does not.

The sonographer usually holds a transducer which is placed on the patient's skin. Ultrasound travels through soft tissue and fluids, but it bounces back off denser surfaces. This allows the creation of an image that is analysed and interpreted by radiologists, cardiologists or other specialists. Higher frequencies provide better quality images but are more readily absorbed by the skin and other tissue, so they cannot penetrate as deeply as lower frequencies. On the contrary, lower frequencies penetrate deeper, but the image quality is inferior. In fact, it is important to remember that mechanical waves cannot travel without attenuation[10] and each body they enter in contact with behaves differently in this sense (actually, attenuation will occur not only in the beam of sound produced by the transducer as it propagates through tissue, but also in the returning echoes as they travel back to the transducer).

Many researchers [128] have worked on the field of remote sensing with ultrasound probes and equipment, but some of them have also claimed an intrinsic difficulty using these devices especially for a long time [129, 130]. An important point of more recent advances in medical technologies is remote diagnosis; on one hand this implies the possibility of operating without an effective physical contact and therefore allowing the physician to help people in spite of long distances and logistic difficulties, such as those concerning rural areas, on the other hand it implies several problems, such as delays, instability and implementative issues. Various researches have been proposed to account for the intrinsically higher complexity of a remote sensing-based ultrasound equipment [131, 132].

Modelling of visco-elastic materials

It is quite interesting that the most common models for visco-elastic materials are linear [133]. Generally speaking, there are different possible

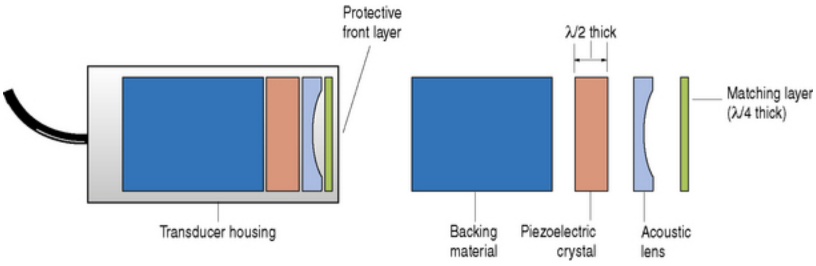


Figure 3.11: Cross-sectional view of an ultrasound transducer that shows how it is structured. The transducer housing is the wrapping material inside which all those elements required for the scan are placed. In other words, it provides the necessary support and protection for them. The back-bouncing effect is due to the presence of a piezoelectric crystal (usually PZT) that allows the device to work both as transmitter and as receiver. Since the operator is interested in capturing only those vibrations that come off the front face of the transducer, the backing material, usually realised in tungsten powder and plastic or epoxy resin, acts as a damper.

[10] Attenuation can occur in different ways. For example, it happens whenever the energy associated to the travelling mechanical waves is converted into heat (absorption) or when the waves encounter some irregular tissues whose size is much smaller than the wavelength (scattering).

Frequency	Applications
2.5 MHz	deep abdomen, obstetric and gynaecological imaging
3.5 MHz	general abdomen, obstetric and gynaecological imaging
5 MHz	vascular, breast, pelvic imaging
7.5 MHz	breast, thyroid
10 MHz	breast, thyroid, superficial veins, superficial masses, musculoskeletal imaging
15 MHz	superficial structures, musculoskeletal imaging

Table 3.1: Examples of application involving ultrasound scan with their operating frequencies.

solutions to describe the behaviour of a visco-elastic material through a linear framework:

- *Maxwell's model* (Figure 3.12): the combined action of both the viscous and the elastic behaviour is modelled through a spring and a damper attached together so that the stress of the former equals the stress of the latter, whilst the total strain is given by the sum of the two strain functions. The equations that govern the model are reported below:

$$\begin{cases} \sigma_{\text{spring}} = k\varepsilon_{\text{spring}} \\ \sigma_{\text{damper}} = \eta\dot{\varepsilon}_{\text{spring}} \\ \sigma_{\text{total}} = \sigma_{\text{spring}} = \sigma_{\text{damper}} \\ \varepsilon_{\text{total}} = \varepsilon_{\text{spring}} + \varepsilon_{\text{damper}} \end{cases} \tag{3.41}$$

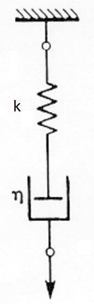


Figure 3.12: Maxwell's model of visco-elastic materials.

  where k is the stiffness of the spring and η is the viscosity. All the previous equations are perfectly equivalent to the single differential equation:

$$\dot{\sigma}_{\text{total}} + \frac{1}{\tau}\dot{\sigma}_{\text{total}} = k\dot{\varepsilon}_{\text{total}} \tag{3.42}$$

  where $\tau = \frac{\eta}{k}$ is the time constant. The model is usually applied to the case of small deformations, because when these are too marked one should include some non-linear effects that depend on the geometry of the material. An important disadvantage of this model concerns its incapability of describing creep or recovery.

- *Kelvin-Voigt model* (Figure 3.13): this model differs from the Maxwell's one because of the presence of a deferred elasticity: instead of placing the damper in series to the spring, these items are positioned in parallel so that the underlying equations that describe how the system evolves become:

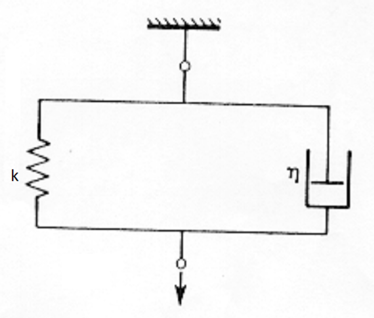


Figure 3.13: Kelvin-Voigt model of visco-elastic materials.

$$\begin{cases} \sigma_{\text{spring}} = k\varepsilon_{\text{spring}} \\ \sigma_{\text{damper}} = \eta\dot{\varepsilon}_{\text{spring}} \\ \sigma_{\text{total}} = \sigma_{\text{spring}} + \sigma_{\text{damper}} \\ \varepsilon_{\text{total}} = \varepsilon_{\text{spring}} = \varepsilon_{\text{damper}} \end{cases} \tag{3.43}$$

  Again, these equations can lead to a single ODE as follows:

$$\dot{\varepsilon}_{\text{total}} + \frac{1}{\tau}\dot{\varepsilon}_{\text{total}} = \frac{1}{k\tau}\dot{\sigma}_{\text{total}} \tag{3.44}$$

  where $\tau = \frac{\eta}{k}$ again. One of the drawbacks of this model regards the impossibility of describing stress relaxation.

- *Zener's model*: there are two possible ways to present this model, namely the Maxwell's (Figure 3.14) and the Kelvin's representation (Figure 3.15). However, the underlying ODE is formally equivalent for both the variants:

$$a\ddot{\sigma}_{\text{total}} + \dot{\sigma}_{\text{total}} + b\ddot{\varepsilon}_{\text{total}} + d\dot{\varepsilon}_{\text{total}} \tag{3.45}$$

but

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{cases} \begin{bmatrix} \tau \\ \tau(k_1 + k_2) \\ k_1 \end{bmatrix} & \text{Maxwell's representation} \\ \begin{bmatrix} \frac{\eta}{k_1+k_2} \\ \frac{k_1\eta}{k_1+k_2} \\ \frac{k_1k_2}{k_1+k_2} \end{bmatrix} & \text{Kelvin's representation} \end{cases} \tag{3.46}$$

where $\tau = \frac{\eta}{k_2}$.

- *Burgers' model* (Figure 3.16): this model is realised by placing two Maxwell's models in parallel and therefore it is governed by the following equation:

$$\tau_1\tau_2\ddot{\sigma}_{\text{total}} + (\tau_1 + \tau_2)\dot{\sigma}_{\text{total}} + \sigma_{\text{total}} = \tau_1\tau_2(k_1 + k_2)\ddot{\varepsilon}_{\text{total}} + (\eta_1 + \eta_2)\dot{\varepsilon}_{\text{total}} \tag{3.47}$$

  where $\tau_i = \frac{\eta_i}{k_i}$ are two distinct time constants that determine how fast/slow the modes are.
- *Maxwell-Wiechert model* (Figure 3.17): this is the most general linear model for visco-elastic materials. Given $(N+1)$ stiffness coefficients k_i and viscosities η_i so that $\tau_i = \frac{\eta_i}{k_i}$ is the i-th relaxation time, the model is governed by the following differential equation:

$$\begin{gathered} \sigma_{\text{total}} + \textstyle\sum_{n=1}^{N} \left(\sum_{i_1=1}^{N-n+1} \cdots \left(\sum_{i_a=i_{a-1}+1}^{N-(n-a)+1} \cdots \left(\sum_{i_n=i_{n-1}+1}^{N} \left(\prod_{j\in\{i_1,\ldots,i_n\}} \tau_j\right)\right)\right)\right) \frac{\partial^n \sigma_{\text{total}}}{\partial t^n} \\ = \\ k_e\varepsilon + \left(\textstyle\sum_{i_1=1}^{N-n+1} \cdots \left(\sum_{i_a=i_{a-1}+1}^{N-(n-a)+1} \cdots \left(\sum_{i_n=i_{n-1}+1}^{N} \left(k_e + \sum_{j\in\{i_1,\ldots,i_n\}} k_j\right) \left(\prod_{l\in\{i_1,\ldots,i_n\}} \tau_l\right)\right)\right)\right) \frac{\partial^n \varepsilon_{\text{total}}}{\partial t^n} \end{gathered} \tag{3.48}$$

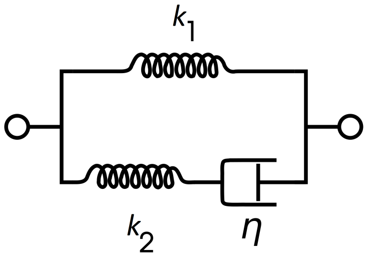


Figure 3.14: Zener's model of visco-elastic materials according to the Maxwell's representation.

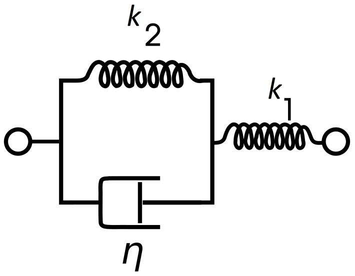


Figure 3.15: Zener's model of visco-elastic materials according to the Kelvin's representation.

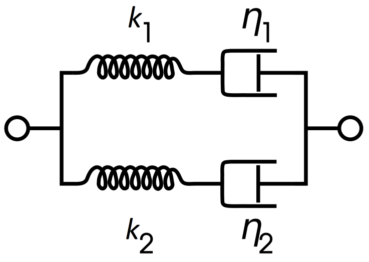


Figure 3.16: Burgers' model of visco-elastic materials.

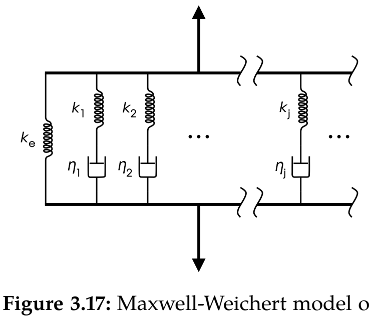


Figure 3.17: Maxwell-Weichert model of visco-elastic materials.

3.4 Information criteria for model validation

A system can be modelled in many ways. That is why it is crucial to set a common set of tools to compare them properly. Statistically speaking, these tools are usually referred to as information criteria and most of the time they consider two main aspects:

- predictive/explanatory power: the model must be as exhaustive as possible with respect to the data used to create the model itself, albeit this could imply severe drawbacks in terms of its own realisation;
- structural complexity: this is the other side of the coin, because whenever the model tends to be very precise at predicting an outcome, its internal complexity becomes significant. It is not preferable to have such models, because the number of degrees of freedom increase too and this could make things more difficult to handle.

To some extent, model selection embodies the Ockham's razor[11] : the simplest model is most likely to be the best choice. In the following, I have covered just some methods because of their implications in some of my papers.

Akaike's Information Criterion

The Akaike's Information Criterion (AIC) [134] is an information theory-based method for quantifying how good a model is with respect to various candidates:

$$\mathrm{AIC} = 2n_p - 2\log\left(\mathscr{L}_{\max}\right) \tag{3.49}$$

where n_p is the number of parameters and $\mathscr{L}_{\max}$ is the maximum of the likelihood function for the given model. As already written before, the AIC is based on information theory, meaning that if I represented an unknown process f by means of two possible candidates g_1 and g_2 and I had to select the best one, I could calculate $\mathrm{KL}(f \parallel g_1)$ and $\mathrm{KL}(f \parallel g_2)$ to evaluate the information lost from using g_1 and g_2 to represent f. Therefore, the best model would be the one that minimises this loss, but unfortunately this decision cannot be made without uncertainty and that is why the AIC was introduced as an asymptotic estimation, which may fail if the number of observations is particularly small. In this last scenario, if the number of observations is N then the AIC indicator ought to be modified as follows:

$$\mathrm{AICc} = \mathrm{AIC} + \frac{2n_p(n_p+1)}{N - n_p - 1} \tag{3.50}$$

and straightforwardly $\lim_{N\to+\infty} \mathrm{AICc} = \mathrm{AIC}$.

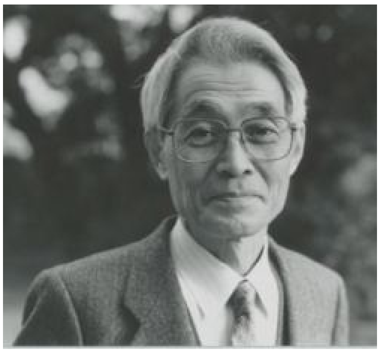


Figure 3.18: Hirotugu Akaike (1927-2009), credited with the introduction of the homonymous information criterion. For this important advancement in statistics, he received the Kyoto Prize in 2006.

Independently from the version of the AIC one wants to use, from a practical standpoint if r candidate models have AIC values equal to $\mathrm{AIC}_1, \ldots, \mathrm{AIC}_r$, then $\exists \bar{r} : \mathrm{AIC}_{\bar{r}} = \min_i \mathrm{AIC}_i$ so that $e^{\frac{\mathrm{AIC}_{\bar{r}} - \mathrm{AIC}_i}{2}} \propto$ probability

[11] The idea of the English Franciscan friar William of Ockham is an example of abductive reasoning and the preference for simpler solutions agrees with the falsifiability principle introduced by Karl Popper in his *Logik der Forschung* (1934).

that the i-th model minimises the estimated information loss. Thereby, the $\bar{r}$-th model is the best amongst the others.

Bayesian Information Criterion

The Bayesian Information Criterion (BIC, for short) was introduced by G. Schwarz in 1978 [135] and is very similar to the AIC, except for some slight changes as shown below:

$$\text{BIC} = n_p \log(N) - 2\log\left(\mathscr{L}_{\text{max}}\right) \tag{3.51}$$

where the symbols have the same meaning as reported in the previous paragraph. A first difference between the AIC and the BIC concerns how each criterion penalises the number of parameters, since the latter penalises model complexity more than the former. Additionally, what the BIC aims at doing is different: if the AIC proposes to evaluate the best model amongst various candidates that may explain an unknown system or collection of data, the BIC aims at finding the true model amongst these possibilities. However, in terms of practicality both the AIC and the BIC tend to agree, implying that when a model is characterised by a low AIC, then its BIC ought to be low as well. Mutual discrepancies arise because of the way these criteria were designed; as already mentioned, the difference in penalty may provide slightly different outcomes in terms of numerical values and this explains why the BIC behaves better than the AIC when the number is relatively small [136]. Unfortunately, the BIC has some drawbacks too:

- the underlying approximation of the BIC is valid as long as $N \gg n_p$;
- when it comes to evaluating models in highly dimensional spaces, the BIC may not perform as well as other information criteria [137].

Hannan-Quinn Information Criterion

An alternative to both the AIC and the BIC is given by the Hannan-Quinn Criterion (or HQC, for short), which was introduced by E. J. Hannan and B. G. Quinn in 1979 [138]. Like the BIC, but unlike the AIC, the HQC is not an estimator of KL and is not asymptotically efficient; however, it misses the optimal estimation rate by a very small $\log\big(\log(N)\big)$ factor, which allows the HQC to be much more consistent than the AIC and the BIC because of the law of the iterated logarithm[12] . The expression of the HQC is given below:

$$\text{HQC} = 2n_p \log\big(\log(N)\big) - 2\log\left(\mathscr{L}_{\text{max}}\right) \tag{3.52}$$

Again, the notation is the same as the one used before.

[12] This statement was proposed at first by the Soviet mathematician A. Y. Khinchin in 1924 [139] and states that if $\mathscr{X}_1, \ldots, \mathscr{X}_n$ are n random variables, each having mean equal to 0 and variance equal to 1, and $\mathscr{S}_n \doteq \sum_i^n \mathscr{X}_i$, then $\limsup_{n \leftarrow +\infty} \frac{\pm\mathscr{S}_n}{\sqrt{2n \log\log n}} = 1$ almost surely.

Contributions

Conspectus | 4

Before introducing a complete list of my publications, I have reported a simple, yet effective, graphical summary of the works I did during my Ph.D. activity.

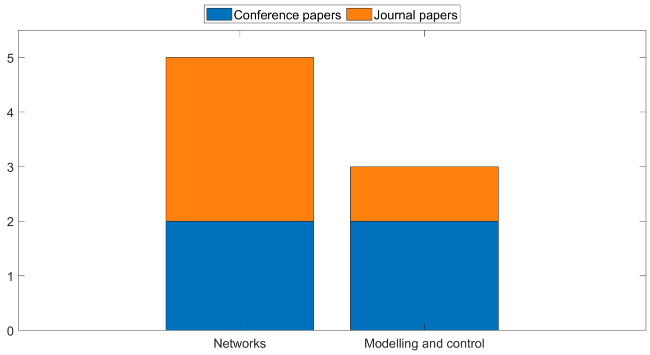


Figure 4.1: Count of the publications and their categorisation into research themes.

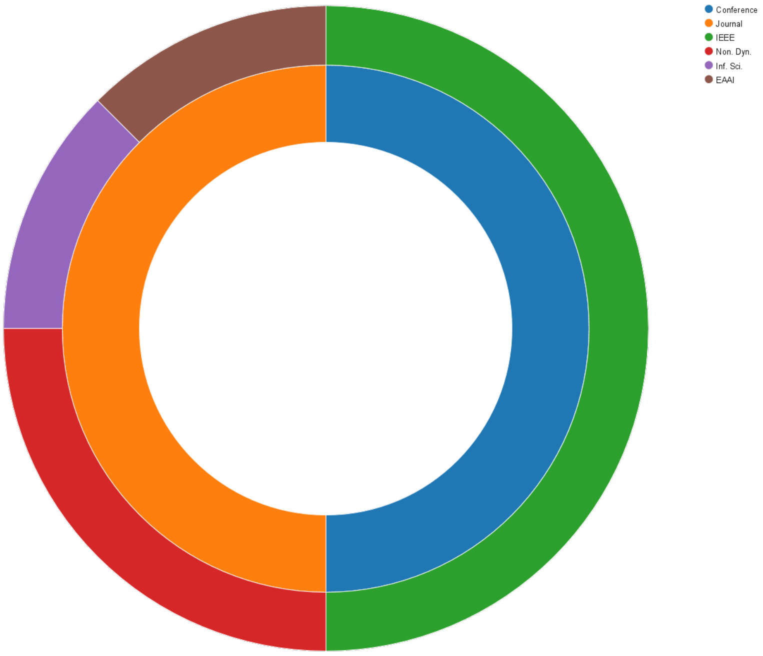


Figure 4.2: Categorisation of every publication into type (either journal or conference) and destination.

Networks 5

5.1 Insect-inspired spatial-temporal cellular processing for feature-action learning

Abstract - In this paper, an insect brain-inspired neural processing architecture was developed to be applied on board of a bio-robot requested to solve feature-to-action association tasks. Relying on visual features, the system could solve classification problems by using a spatial-temporal approach that is typical of bio-inspired neural architectures. Taking inspiration from the mushroom bodies of the fruit fly, the proposed neural structure was employed to emulate some capabilities that had been discovered in various experiments, especially those focused on non-elemental learning strategies. An important peculiarity of the hidden processing layer of the proposed multi-layer architecture was the local connectivity amongst spiking neurons, that had resembled the Cellular Non-linear Network[1] paradigm.

Description - My first conference paper concerned the development of a LSM for classification by means of the parallel computing paradigm. The original idea regarded the understanding of the decision making processes occurring in bio-inspired neural networks implemented in simulated, legged robots that had resembled the common fruit fly. In particular, the LSM was trained to make sequential decisions in a Markovian-like fashion, meaning that the robot had been requested to make two conceptually disjointed, but environmentally related, decisions where the latter had depended on the former. More in detail, once placed the virtual robot within a Y-shaped labyrinth the decisions regarded which motor action the robot needed to perform depending on environmental stimuli (for the sake of simplicity, simple, visual indicators were used to encode the information). In this particular case study, these indicators

[1] The so-called Cellular Non-linear (or Neural) Network paradigm (CNN, for short) was introduced by Leon Chua and Lin Yang in 1988 [140] to describe a way of connecting several nodes within a regular lattice. Each node is characterised by a neighbourhood that affects nodal communication, meaning that a node can communicate with its neighbours only.

were employed to provide a pragmatical representation of the so-called Positive Patterning Discrimination [141] (or PPD, for short) paradigm [141], which claims that if two singularly acting stimuli A and B are somehow reinforced, then their combined action, namely the stimulus AB, is not[2] . The stimuli were encoded as two distinct pairs of actions: turn right-turn left and walk-climb. The former pair was expressed with the colour of a landmark (a red or green object had encoded the "turn right" decision, whereas a yellow object (red + green) had encoded the "turn left" decision), whilst the latter was expressed by means of its shape (both horizontally and vertically distributed objects had denoted the presence of an obstacle and thereby had implied the need of climbing, whereas a perfectly squared object had denoted the absence of obstacles). According to these premises, then, the whole architecture was tested in order to deal with Markovian-like decision making processes affected by "contradicting" outcomes. Both the networks and the simulations within the dynamical environment where the legged *Drosophila*-like robot was simulated were carried out by means of the open-source simulators GeNN [142] and V-REP [143], respectively. The results I obtained were particularly indicative of the encumbrance imputable to the sequential nature of the whole computational activity, thereby justifying the use of RNNs because of their intrinsic capability of dealing with temporal information. Despite the elaborateness, a simple LSM endowed with Class I Izhikevich neurons, arranged in a 8-by-8 grid with local connections, and trained by means of the LMS algorithm and a Winner-Takes-All layer (or WTA, for short) for label assignment could perform pretty well (the average probabilities of correctly classifying each possible pair of motor action were equal to 87.05% for the learning phase and 86.91% for the testing phase).

More details can be found in [144].

5.2 A CNN-based neuromorphic model for classification and decision control

Abstract - In this paper, an insect brain-inspired computational structure was developed. The peculiarity of the core processing layer was the local connectivity amongst spiking neurons in a CNN-like fashion. Moreover, the processing layer worked as a LSM with fixed connections and trainable, output weights. Learning was accomplished by adopting a simple supervised, batch approach based on the calculation of the Moore–Penrose matrix. Then, the architecture was evaluated and compared with other standard and bio-inspired solutions available in literature, with a view to three different scenarios.

Description - This work prosecuted what already shown in the previous paper by introducing new elements and insights and was dedicated to the improvement of LSMs with local connections and Class I Izhikevich neurons for classification, especially of more realistic and less abstract data. Additionally, I provided new information on how the size of the processing layer affected the resulting classification performance, in comparison with other common methods.

More details can be found in [145].

[2]: Of course, there is also its negative counterpart, namely the Negative Patterning Discrimination (or NPD, for short) paradigm. An example of that regards students: as result of poor performance on a midterm, they are prompted to study intensely for the final to raise their class grade. As result of poor performance on quizzes in another class, they are again prompted to study intensely for the final to raise their grade. However, if they perform poorly on the midterm and on quizzes in the same class, they will not be prompted to study at all for the final because a high grade on the final cannot raise their overall grade in the class. Therefore, they will be more likely to study for final exams when they have performed poorly on a midterm in one class and quizzes in another.

5.3 Data-based analysis of Laplacian Eigenmaps for manifold reduction in supervised Liquid State classifiers

Abstract - The manuscript introduced a data-driven technique founded on LEs for manifold reduction in bio-inspired LSMs. Starting from a preliminary introduction about the algorithm and the need of using manifold reduction methods for data representation, a statistical analysis of hyperparameters involved in the LEs technique was presented and the effects of quantisation over trained weights was discussed with a view to efficient implementation of multiple, parallel mappings in the digital domain.

Description - Together with the two previous papers, this work constituted a preeminent part of my publications because it extended the context of LSMs in bio-inspired computation further and introduced the problem of curse of dimensionality as well, together with a possible solution by means of LEs. This was the beginning of a new section of my work, mainly consisting in finding smaller, easier data representations by means of feature selection and/or algorithms for dimensionality reduction. The use of LEs in LSMs allowed me to create classifiers whose noise rejection had improved significantly; in particular, this work dealt with various forms of noise and results showed that LEs made the network more robust against parametric disturbances over the learnt weights. Moreover, the whole strategy was deeply specialised, meaning that selection of hyperparameters was made semi-automatic thanks to the statistical properties of the data at disposal, namely how each percentile had distributed. According to that, it was possible to set the heat kernel and thereby realising a semi-automatic dimensionality reduction procedure, without any prior knowledge about the signals to process. Model selection was performed thanks to the HQC index, which had allowed me to select the optimal n as stated by Equation 2.9.

More details can be found in [146].

5.4 Structural and input reduction in an ESN for robotic navigation tasks

Abstract - This manuscript aimed at showing the effects of feature selection and manifold reduction methods when dealing with the wall-following problem in mobile robotics, a well-known, non-linearly separable classification problem in which sensor recordings are associated to controlled motor responses. The capabilities of state manifold reduction in ESNs through LEs were described in terms of noise rejection over the trained weights. Furthermore, various machine learning-based and data mining-based methodologies were applied to show the advantages of using the most informative contents drawn from the original sensor readings.

Description - This paper aimed to show the effects of both feature selection and LEs over ESNs when dealing with real recordings drawn from experiments concerning mobile robotics. The idea was to provide less complex, yet reliable, ESNs capable of dealing with multiple decisions with a reduced number of discriminant, input features. In fact, what emerged was the interesting property of manifesting good classification performance despite the lack of some input attributes, which had been less determinant in encapsulating the core information regarding the decisions to make. The evaluation of the most significant features was carried out thanks to most of the methods I have reported in Part I, such as FCBF, IG, GI, Relief-F and both the ANOVA and χ^2 tests. As expected, the effects of feature selection were totally positive because it had helped the ESNs (and other classifiers used for the sake of comparison) to perform better. In order to finely tune the dimensionality reduction algorithm, all the information criteria I have mentioned before were used and compared.

More details can be found in [147].

5.5 Robust modelling of binary decisions in Laplacian Eigenmaps-based Echo State Networks

Abstract - This paper aimed to present a framework for supervised, binary classification of n-Boolean functions through ESNs endowed with LEs for dimensionality reduction. The proposed method was applied both to improve the classification performance when the learnt weights are quantised in view of a digital implementation and as a computational demonstration of the neural reuse theory [148] when parallel outputs are allowed. My analysis focused on the effect of various forms of noise (*id est*, normal noise, uniform noise and quantisation noise) over all the possible Boolean functions of n input bits. These disturbances were applied both over the learnt weights and the input features so that we could analyse how resilient the whole architecture was. Results presented here showed that dimensionality reduction allowed by the LEs improved robustness to these different sources of noise, leading to reduced memory storage requirements while maintaining high classification performance. Our results were compared to those derived from other more common classification techniques in terms of learning performance and computational complexity.

Description - The concepts of neural reuse and parallel mappings, which had been introduced in the first paper reported in this section, were further discussed here. Generally speaking, a complex decision may be thought of as the combination of multiple, elementary decisions. Once established how many sub-decisions can determine a more complicated outcome, the essential idea of this work consisted in developing ESNs capable of learning every possible Boolean function determined by the single sub-decisions. In this context, a binary encoding established the occurrence or the absence of a given sub-decision and thereby each macro-decision was meant to be a collection of 0 and 1 only. The problem, then, resulted in determining the best synaptic weights in order to explore, in presence of parametric uncertainty, a complete n-dimensional hypercube[3] whose vertices had corresponded to distinct binary strings/macro-decisions. Observe that the number of possible macro-decisions increases double-exponentially as 2^{2^n}, therefore the

[3] The fancy term "hypercube" is used to refer to the generalisation of a cube in a multidimensional space. From a computationally perspective, however, this does not sound scary at all, since hypercubes are collection of vectors of homogeneous data of smaller dimension. The problem of encoding each of their vertices is freely solvable, in the sense that there is not a strict requirement and two consecutive vertices may have a distance (like the Hamming's distance) greater than 1 bit.

problem of storing the hypercube required some efforts in terms of memory consumption. That is why I proposed to quantise the learnt weights after training the network, in order to reduce the number of bits for each possible macro-decision. Again, as already confirmed in literature I showed that LEs made the network more resistant to noise and therefore, when endowed with them, they could make the macro-decision correctly in spite of flipping the bits within each binary string. The idea of treating Boolean functions is not as abstract as it seems, because in many factory applications the activation/deactivation of switches and/or sensors determines how machines or devices work. LEs were tuned by means of the usual information criteria I have reported before (AIC, BIC and especially HQC).

More details can be found in [149].

6 Modelling and control

6.1 A nullcline-based control strategy for PWL-shaped oscillators

Abstract - Starting from the PWL framework applied to networked FitzHugh-Nagumo oscillators, this paper aimed at presenting a control strategy relying on phase portrait reshaping through the manipulation of their nullclines, in order to fulfil both phase and time requirements. This was achieved by relating the slopes of piece-wise, linearly approximated nullclines to the oscillation period of the model. Additionally, the targeted issue was addressed by combining the former framework with an event-driven control strategy aimed at reducing the control effects to specific time instants instead of continuously applying them, which would be much more computationally expensive. The strategy was therefore motivated by its simplicity and supported by key applications for bio-inspired locomotion control in legged robots, suggesting how a dynamics-preserving approximation of the phase portrait combined with a sampled control action could produce pre-defined phase topologies in directed, non-diffusive tree graphs.

Description - Most of the time, phase-locking and therefore synchronisation of oscillators occur in a diffusive way, meaning that there is a sort of physical connection amongst the units that carries the "feedback error" employed for corrections in terms of frequency and/or phase. What I aimed at in this work was something different: by coupling non-linear oscillators through a more computational-oriented framework and by exploiting a simple, yet topologically equivalent, approximation of their nullclines, it was possible to control their behaviour so that an undefined number of slave nodes had been capable of tracking a single master once fixed their reciprocal phase displacements. I proved the goodness of the piece-wise approximation for a series of complex scenarios, ranging from the aforementioned phase-locking phenomenon to canards explosions. Additionally, everything was presented with an exhaustive, yet simple-to-follow, appendix with the mathematical details to clarify the most crucial aspects. Furthermore, the paper presented some examples of gait modelling because of the importance that CPGs[1] have in biorobotics; thanks to its intrinsic adaptability, the algorithm allowed to reproduce bipedal, quadrupedal and even dodecapodal gaits that had resembled animal locomotion. Additionally, instead of implementing a continuously operating algorithm, the whole strategy was set as an event-driven method that had allowed lower control efforts. This was meant to be a further advantage in view of microcontroller-based applications.

More details can be found in [151].

6.2 High-Level Analysis of Flux Measurements in Tokamak Machines for Clustering and Unsupervised Feature Selection

Abstract - Plasma physics is an example of research field where many measurements carried out at very specific working conditions need to be collected and processed. By looking at the properties of these data, it can be possible to explore their hidden features in order to solve challenging problems that usually require high computational efforts, such as the tomographic reconstruction. In this paper, preliminary but non-trivial analyses of flux measurements produced in a Tokamak machine were shown and discussed, with the aim of introducing an application of some algorithms for feature selection to detect hidden, relevant relationships within given sets of channels. All the statistical details, and therefore the feature selection procedure itself, were introduced in view of further deepenings, such as the aforementioned problem of tomographically reconstructing plasma profiles from flux measurements or modelling the system in terms of its input-output behaviour.

Description - This paper was thought of as an introductory, exploratory analysis of some time series produced by the FTU machine in Frascati (Rome), a common Tokamak used in plasma physics. Starting from preliminary considerations about the machine itself, this article reports both common and less perfunctory considerations about feature selection for physical signals. The analyses were of different types, from correlation analysis (Spearman's coefficient was the most suitable choice because both the normality assumption and the homoscedasticity hypothesis had noy been met and due to the spiky nature of the signals of interest) to non-linear analyses through MI and ID. Classical statistical tests, such as the Kolmogorov-Smirnov test and ADF test, were used as well. The results were quite interesting: apart from the mere, gnoseological impact, they showed that these kind of signals had tended to cluster

[1] A Central Pattern Generator, or CPG, is allegedly the main, nervous centre responsible for locomotion and production of rhythmic, patterned movements, not necessarily involved in motion (for example, respiration, heartbeat or even swallowing) [150].

autonomously owing to some internal similarities, that could be exploited for the definition of reduced order models.

At the moment of writing, further details were not available because the paper was accepted, but not published yet.

6.3 Human-Machine Models for Remote Control of Ultrasound Scan Equipment

Abstract - In this contribution, a new aspect in robotic applications was approached. The problem regarded the human-machine modelling for remote ultrasound scan equipment. Even though robotic systems for ultrasound scan applications with remote operations had widely been studied, in this research the remote force feedback control was investigated. Accordingly, the human operator receives the correct force perception as input that is transmitted by the remote ultrasound scan equipment to analyse the patient's body. Two principal aspects were investigated, namely the introduction of an artificial body model that would receive the control signals from the remote equipment and the study of suitable feedback control laws that would compensate both the uncertainty between the artificial body and the patient's body by also taking into account the transmission delay. Therefore, the task was to give to the operator the real perception, considering the force effect as well, in order to make a quite real platform capable of working in remote condition for ultrasound scan equipment.

Description - The content of this paper was realised in accordance with the participation of the university team to the Innovation Award 2020: Medical Robotics Challenge sponsored by KUKA, whose main aim had been the development of new solutions for medical and surgical robotics. Essentially, this work focused on developing effective, yet structurally simple, models for remote communication and actuation by means of a microcontroller-based platform that should have simulated haptic feedback on proper human tissue-like materials, in order to develop appropriate control laws for shaping the force response to apply. This was due to the essential fact that remote sensing had required both the modelling of the communication line and the deployment of a suitable controller capable of mixing both the outputted, sensor signal and the force applied by an operator. A crucial role in this work was played by the modelling of visco-elastic materials, whose properties have been shown to be perfectly fittable to the characteristics of most of the human tissues, especially those of our interest in this paper which regards ultrasound equipment. Further considerations about the use of these models were reported to show how resilient the whole setup was in terms of stability and promptness against transmission delays, which had constituted the main source of non-linearities.

More details can be found in [152].

Explicit[*]

Una formica si muove su una corda di gomma lunga 1 m ad una velocità di 1 cm/s verso una delle due estremità e partendo dal centro della stessa. Allo stesso tempo, la corda si estende uniformemente in entrambe le direzioni ad una velocità di 1 m/s. Riuscirà la formica a raggiungere la sua destinazione?[†]

Ho deciso di iniziare questo documento con tre differenti citazioni di tre differenti autori. Chi non mi conosce potrebbe falsamente ritenere che si tratti semplicemente di un modo altezzoso e pomposo di scrivere un documento, ignorando invero che queste tre citazioni sono la migliore espressione di ciò che questo dottorato di ricerca è stato per me. Se dovessi individuare delle persone a cui rivolgere i miei ringraziamenti per questo lavoro finale, allora mi rivolgerei a me stesso. Tutto è derivato dalla mia volontà di proseguire quest'inutilmente pernicioso percorso per la mia salute nonostante validi motivi per non farlo. *Ringrazio nessun altro se non me stesso per essere riuscito a raggiungere la fine della corda*. Voglio solo fare qualche menzione *sic vos non vobis* perché, nonostante tutto, queste persone hanno determinato i miei pensieri mentre il *Torschlusspanik* si diffondeva.

La mia famiglia ha avuto un ruolo non costante ma non per questo meno rilevante. La sua influenza è stata comunque determinante per il raggiungimento di questo traguardo, in un modo o nell'altro.

Andando a ritroso sino agli anni della mia infanzia, forse tra i più genuini a mia disposizione, non posso non ricordare Giuseppe e Danilo. Nonostante l'attuale lontananza reciproca, forse mai più colmabile, sono stati loro i fautori della forma più pura di amicizia che io abbia mai esperito.

Gli anni delle scuole superiori sono stati di gran lunga più incisivi e determinanti di quelli universitari. In quel periodo di alti e bassi ebbi la fortuna di incontrare docenti che hanno saputo veicolare l'interesse e la passione verso le materie tecniche: se so qualcosa di informatica od elettronica, è decisamente merito di professori come i professori Biuso, Cutugno, Dilettoso, Mita e Riscica, le cui personalità resteranno qualcosa che un qualsiasi docente universitario da me incontrato può solo lontanamente sperare di acquisire.

Durante gli anni dell'università diverse sono state le figure incontrate, molti i colleghi ma poche le persone degne di stima. Per questo motivo, mi vengono in mente Dario P. e Dario S., Francesco ed Antonio, Andrea V. e Andrea C., Luca ed Alberto, così come Carmelo e Serena, due sostegni che hanno desiderato ardentemente che io combattessi il buio che è in me, finendo con l'essere ripagati ingiustamente con malumore, disinteresse, indifferenza.

Non c'è altro da aggiungere.

A.G.S.,
la formica sulla corda di gomma

[*] Dal latino medievale, "qui finisce".
[†] Possibile formulazione del paradosso della formica sulla corda di gomma.

Bibliography

[1] F. Nake. 'Data, Information, and Knowledge'. In: *Organizational Semiotics*. 2001 (cited on page 6).

[2] P.S. Bradley and O.L. Mangasarian. 'Feature Selection via Concave Minimization and Support Vector Machines'. In: *Proceedings of the Fifteenth International Conference on Machine Learning*. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 82–90 (cited on page 6).

[3] S. Darian. 'The role of redundancy in language and language teaching'. In: *System* 07.01 (1979), pp. 47–59 (cited on page 6).

[4] P.E. Ammann. 'Data Redundancy for the Detection and Tolerance of Software Faults'. In: *Computing Science and Statistics*. Ed. by Connie Page and Raoul LePage. New York, NY: Springer New York, 1992, pp. 43–52 (cited on page 6).

[5] J. Miao and L. Niu. 'A Survey on Feature Selection'. In: *Procedia Computer Science* 91 (2016), pp. 919–926 (cited on page 8).

[6] S. Huang. 'Supervised feature selection: A tutorial'. In: *Artificial Intelligence Research* 4 (Mar. 2015) (cited on page 8).

[7] B.G. Tabachnick and L.S. Fidell. *Using Multivariate Statistics (5th Edition)*. USA: Allyn & Bacon, Inc., 2006 (cited on page 9).

[8] B.K. Sriperumbudur et al. 'Hilbert Space Embeddings and Metrics on Probability Measures'. In: *J. Mach. Learn. Res.* 11 (Aug. 2010), pp. 1517–1561 (cited on page 10).

[9] C.E. Shannon. 'A Mathematical Theory of Communication'. In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423 (cited on page 11).

[10] S. Kullback and R.A. Leibler. 'On Information and Sufficiency'. In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86 (cited on page 11).

[11] J.V. Jensen. 'Sur les fonctions convexes et les inégalités entre les valeurs moyennes'. In: *Acta Mathematica* 30 (), pp. 175–193 (cited on page 11).

[12] J.R. Quinlan. 'Induction of decision trees'. In: *Machine Learning* 1.1 (Mar. 1986), pp. 81–106 (cited on page 12).

[13] L. Yu and H. Liu. 'Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution'. In: vol. 2. Jan. 2003, pp. 856–863 (cited on page 12).

[14] S. Bakhshandeh, R. Azmi, and M. Teshnehlab. 'Symmetric uncertainty class-feature association map for feature selection in microarray dataset'. In: *International Journal of Machine Learning and Cybernetics* 11.1 (Jan. 2020), pp. 15–32 (cited on page 12).

[15] S. Havlin et al. 'Fractals in biology and medicine'. In: *Chaos, Solitons & Fractals* 6 (1995). Complex Systems in Computational Physics, pp. 171–201 (cited on page 13).

[16] A.S. Soliman. 'Fractals in nonlinear economic dynamic systems'. In: *Chaos, Solitons & Fractals* 7.2 (1996), pp. 247–256 (cited on page 13).

[17] R.E. Bellman. *Dynamic Programming*. USA: Dover Publications, Inc., 2003 (cited on page 13).

[18] P. Grassberger and I. Procaccia. 'Measuring the strangeness of strange attractors'. In: *Physica D: Nonlinear Phenomena* 9.1 (1983), pp. 189–208 (cited on page 13).

[19] F. Takens. 'On the numerical determination of the dimension of an attractor'. In: *Dynamical Systems and Bifurcations*. Ed. by Boele L. J. Braaksma, Hendrik W. Broer, and Floris Takens. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 99–106 (cited on page 13).

[20] C.D. Cutler. 'A Review of the theory and estimation of fractal dimension'. In: *Dimension Estimation and Models*, pp. 1–107 (cited on page 13).

[21] J. Kaplan and J. Yorke. 'Chaotic behavior of multidimensional difference equations'. In: vol. 730. Nov. 2006, pp. 204–227 (cited on page 13).

[22] J. Golay and M. Kanevski. 'A New Estimator of Intrinsic Dimension Based on the Multipoint Morisita Index'. In: *Pattern Recognition* 48 (Dec. 2015), pp. 4070–4081 (cited on page 14).

[23] J. Golay and M. Kanevski. 'Unsupervised feature selection based on the Morisita estimator of intrinsic dimension'. In: *Knowledge-Based Systems* 135 (2017), pp. 125–134 (cited on page 14).

[24] M. Wright Muelas et al. 'The role and robustness of the Gini coefficient as an unbiased tool for the selection of Gini genes for normalising expression profiling data'. In: *Scientific Reports* 9.1 (Nov. 2019), p. 17960 (cited on page 15).

[25] K. Kira and L.A. Rendell. 'The Feature Selection Problem: Traditional Methods and a New Algorithm'. In: *Proceedings of the Tenth National Conference on Artificial Intelligence*. AAAI'92. San Jose, California: AAAI Press, 1992, pp. 129–134 (cited on page 15).

[26] R.J. Urbanowicz et al. 'Relief-based feature selection: Introduction and review'. In: *Journal of Biomedical Informatics* 85 (2018), pp. 189–203 (cited on page 15).

[27] J.S. Huxley. 'Problems of Relative Growth'. In: *The Quarterly Review of Biology* 51 (1976), pp. 94–94 (cited on page 16).

[28] S.A. Frank. 'The common patterns of nature'. In: *Journal of Evolutionary Biology* 22.8 (2009), pp. 1563–1585 (cited on page 16).

[29] N. Mohd Razali and B. Yap. 'Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling Tests'. In: *J. Stat. Model. Analytics* 2 (Jan. 2011) (cited on page 16).

[30] S.S. Shapiro and M.B. Wilk. 'An analysis of variance test for normality (complete samples)'. In: *Biometrika* 52.3-4 (Dec. 1965), pp. 591–611 (cited on page 17).

[31] M.A. Stephens. 'EDF Statistics for Goodness of Fit and Some Comparisons'. In: *Journal of the American Statistical Association* 69.347 (1974), pp. 730–737 (cited on page 17).

[32] H.G. Tucker. 'A Generalization of the Glivenko-Cantelli Theorem'. In: *The Annals of Mathematical Statistics* 30.3 (Sept. 1959), pp. 828–830 (cited on page 17).

[33] A.O. Pittenger. 'Stochastic Processes: A Survey of the Mathematical Theory (John Lamperti)'. In: *SIAM Review* 21.3 (1979), pp. 421–422 (cited on page 18).

[34] W.A. Fuller. *Introduction to statistical time series*. A Wiley publication in applied statistics. Wiley, 1976 (cited on page 18).

[35] A. R. Crathorne. 'Review: A. A. Tschuprow, Principles of the Mathematical Theory of Correlation'. In: *Bull. Amer. Math. Soc.* 46.5 (May 1940), p. 389 (cited on page 20).

[36] T. Kohonen. 'Self-organized formation of topologically correct feature maps'. In: *Biological Cybernetics* 43.1 (Jan. 1982), pp. 59–69 (cited on page 21).

[37] W. S. McCulloch and W. Pitts. 'A logical calculus of the ideas immanent in nervous activity'. In: *Bulletin of Mathematical Biophysics* 5 (1943), pp. 115–133 (cited on page 22).

[38] Z. Yujin et al. 'GMFLLM: A general manifold framework unifying three classic models for dimensionality reduction'. In: *Engineering Applications of Artificial Intelligence* 65 (2017), pp. 421–432 (cited on page 22).

[39] D. H. Ballard. 'Modular Learning in Neural Networks'. In: *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*. AAAI'87. Seattle, Washington: AAAI Press, 1987, pp. 279–284 (cited on page 22).

[40] Z. Han et al. 'Incremental Alignment Manifold Learning'. In: *J. Comput. Sci. Technol.* 26 (Jan. 2011), pp. 153–165 (cited on page 24).

[41] C. Shen and C. Priebe. 'Manifold Matching using Shortest-Path Distance and Joint Neighborhood Selection'. In: *Pattern Recognition Letters* 92 (Dec. 2014) (cited on page 26).

[42] Jiang Y. and P. Guo. 'Regularization Versus Dimension Reduction, Which Is Better?' In: *Advances in Neural Networks – ISNN 2007*. Ed. by Derong Liu et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 474–482 (cited on page 24).

[43] M. Belkin and P. Niyogi. 'Laplacian Eigenmaps for Dimensionality Reduction and Data Representation'. In: *Neural Comput.* 15.6 (June 2003), pp. 1373–1396 (cited on page 25).

[44] D.L. Donoho and C. Grimes. 'Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data'. In: *Proceedings of the National Academy of Sciences* 100.10 (2003), pp. 5591–5596 (cited on page 25).

[45] L.J.P. van der Maaten, E.O. Postma, and H.J. van den Herik. 'Dimensionality Reduction: A Comparative Review'. In: *Journal of Machine Learning Research* 10 (2008), pp. 66–71 (cited on page 25).

[46] K. Levin and V. Lyzinski. 'Laplacian Eigenmaps From Sparse, Noisy Similarity Measurements'. In: *IEEE Transactions on Signal Processing* 65.8 (Apr. 2017), pp. 1988–2003 (cited on page 25).

[47] W. Johnson and J. Lindenstrauss. 'Extensions of Lipschitz maps into a Hilbert space'. In: *Contemporary Mathematics* 26 (Jan. 1984), pp. 189–206 (cited on page 25).

[48] D.E. Rumelhart, G.E. Hinton, and R.J. Williams. 'Learning Representations by Back-Propagating Errors'. In: *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988, pp. 696–699 (cited on page 27).

[49] J.J. Hopfield. 'Neural networks and physical systems with emergent collective computational abilities'. In: *Proceedings of the National Academy of Sciences* 79.8 (1982), pp. 2554–2558 (cited on page 27).

[50] T. Makino et al. *Recurrent Neural Network Transducer for Audio-Visual Speech Recognition*. 2019 (cited on page 27).

[51] I. Sutskever, J. Martens, and G. Hinton. 'Generating Text with Recurrent Neural Networks'. In: Jan. 2011, pp. 1017–1024 (cited on page 27).

[52] G. Petneházi. *Recurrent Neural Networks for Time Series Forecasting*. 2019 (cited on page 27).

[53] Z. Che et al. 'Recurrent Neural Networks for Multivariate Time Series with Missing Values'. In: *Scientific Reports* 8 (June 2016) (cited on page 27).

[54] CK Wang et al. 'Using a Recurrent Neural Network Model for Classification of Tweets Conveyed Influenza-related Information'. In: *Proceedings of the International Workshop on Digital Disease Detection using Social Media 2017*. Taipei, Taiwan: Association for Computational Linguistics, Nov. 2017, pp. 33–38 (cited on page 27).

[55] J. Hassan and U. Shoaib. 'Multi-class Review Rating Classification using Deep Recurrent Neural Network'. In: *Neural Processing Letters* (Oct. 2019) (cited on page 27).

[56] H. Jaeger. *The "echo state" approach to analysing and training recurrent neural networks*. Tech. rep. 148. GMD - German National Research Institute for Computer Science, 2001 (cited on page 27).

[57] H. Jaeger. *Short term memory in echo state networks*. Tech. rep. 152. GMD - German National Research Institute for Computer Science, 2002 (cited on page 27).

[58] Z.K. Malik, A. Hussain, and Q.J. Wu. 'Multilayered Echo State Machine: A Novel Architecture and Algorithm'. In: *IEEE Transactions on Cybernetics* 47.4 (Apr. 2017), pp. 946–959 (cited on page 28).

[59] X. Glorot, A. Bordes, and Y. Bengio. 'Deep Sparse Rectifier Neural Networks.' In: *AISTATS*. Ed. by G.J. Gordon, D.B. Dunson, and M. Dudík. Vol. 15. JMLR Proceedings. JMLR.org, 2011, pp. 315–323 (cited on page 29).

[60] V. Kozyakin. 'On accuracy of approximation of the spectral radius by the Gelfand formula'. In: *Linear Algebra and its Applications* 431.11 (2009), pp. 2134–2141 (cited on page 29).

[61] W. Maass and H. Markram. 'On the computational power of circuits of spiking neurons'. In: *Journal of Computer and System Sciences* 69.4 (2004), pp. 593–616 (cited on page 29).

[62] W. Maass, T. Natschläger, and H. Markram. 'Computational Models for Generic Cortical Microcircuits'. In: *Computational Neuroscience: A Comprehensive Approach* (Jan. 2004) (cited on page 29).

[63] W. Maass and H. Markram. 'Theory of the computational function of microcircuit dynamics'. In: *The interface between neurons and global brain function*. MIT Press, 2006, pp. 371–390 (cited on page 30).

[64] L.M. Masuda-Nakagawa et al. 'Localized olfactory representation in mushroom bodies of Drosophila larvae'. In: *Proceedings of the National Academy of Sciences* 106.25 (2009), pp. 10314–10319 (cited on page 30).

[65] DB. Akalal et al. 'Roles of Drosophila mushroom body neurons in olfactory learning and memory'. In: *Learning & memory (Cold Spring Harbor, N.Y.)* 13 (Sept. 2006), pp. 659–68 (cited on page 30).

[66] A. Jenett, J. Schindelin, and M. Heisenberg. 'The Virtual Insect Brain protocol: Creating and comparing standardized neuroanatomy'. In: *BMC bioinformatics* 7 (Feb. 2006), p. 544 (cited on page 31).

[67] J. Whittington and R. Bogacz. 'Theories of Error Back-Propagation in the Brain'. In: *Trends in Cognitive Sciences* 23 (Mar. 2019) (cited on page 31).

[68] B. Anderson and S. Donaldson. 'The backpropagation algorithm: Implications for the biological bases of individual differences in intelligence'. In: *Intelligence* 21.3 (1995), pp. 327–345 (cited on page 31).

[69] Y. Bengio et al. *Towards Biologically Plausible Deep Learning*. 2015 (cited on page 31).

[70] M. Taylor. 'The Problem of Stimulus Structure in the Behavioural Theory of Perception'. In: *South African journal of psychology = Suid-Afrikaanse tydskrif vir sielkunde* 3 (Jan. 1973), pp. 23–45 (cited on page 31).

[71] H. Markram et al. 'Regulation of Synaptic Efficacy by Coincidence of Postsynaptic APs and EPSPs'. In: *Science* 275.5297 (1997), pp. 213–215 (cited on page 31).

[72] R. Penrose. 'On best approximate solutions of linear matrix equations'. In: *Mathematical Proceedings of the Cambridge Philosophical Society* 52.1 (1956), pp. 17–19 (cited on page 31).

[73] S. Gerschgorin. 'Über die Abgrenzung der Eigenwerte einer Matrix'. In: *Izvestija Akademii Nauk SSSR, Serija Matematika* 7.3 (1931), pp. 749–754 (cited on page 32).

[74] B. DePasquale et al. 'full-FORCE: A target-based method for training recurrent networks'. In: *PLOS ONE* 13.2 (Feb. 2018), pp. 1–18 (cited on page 33).

[75] F.A. Rodrigues. *Network centrality: an introduction*. 2019 (cited on page 34).

[76] D. Kousik, S. Sovan, and P. Madhumangal. 'Study on centrality measures in social networks: a survey'. In: *Social Network Analysis and Mining* 8 (2018), pp. 1–11 (cited on page 34).

[77] P. Frankl and W: Rodl. 'Forbidden Intersections'. In: *Transactions of the American Mathematical Society* 300.1 (Mar. 1987) (cited on page 34).

[78] T.A. Schieber et al. 'Quantification of network structural dissimilarities'. In: *Nature Communications* 8.1 (2017), p. 13928 (cited on page 34).

[79] G. Jurman et al. *The HIM glocal metric and kernel for network comparison and classification*. 2012 (cited on page 34).

[80] G. Jurman et al. *Biological network comparison via Ipsen-Mikhailov distance*. 2011 (cited on page 34).

[81] R. Sole and S. Valverde. 'Information Theory of Complex Networks: On Evolution and Architectural Constraints'. In: 650 (Jan. 2004) (cited on pages 35, 36).

[82] P. Erdös and A. Rényi. 'On Random Graphs I'. In: *Publicationes Mathematicae Debrecen* 6 (1959), p. 290 (cited on page 35).

[83] AL Barabási and R. Albert. 'Emergence of Scaling in Random Networks'. In: *Science* 286.5439 (1999), pp. 509–512 (cited on page 35).

[84] A.a Ghosh, S. Boyd, and A. Saberi. 'Minimizing Effective Resistance of a Graph'. In: *SIAM Rev.* 50.1 (Feb. 2008), pp. 37–66 (cited on page 35).

[85] A. Réka. 'Scale-free networks in cell biology'. In: *Journal of Cell Science* 118.21 (2005), pp. 4947–4957 (cited on page 35).

[86] S. Bansal, S. Khandelwal, and L. Meyers. 'Exploring biological network structure with clustered random networks'. In: *BMC bioinformatics* 10 (Dec. 2009), p. 405 (cited on page 35).

[87] E. Ravasz et al. 'Hierarchical Organization of Modularity in Metabolic Networks'. In: *Science* 297.5586 (2002), pp. 1551–1555 (cited on page 35).

[88] B. Vogelstein, D. Lane, and A.J. Levine. 'Surfing the P53 network'. In: *Nature* 408 (Dec. 2000), pp. 307–10 (cited on page 35).

[89] H. Jeong et al. 'Lethality and Centrality in Protein Networks'. In: *Nature* 411 (June 2001), pp. 41–2 (cited on page 36).

[90] B. Leitch and G. Laurent. 'GABAergic synapses in the antennal lobe and mushroom body of the locust olfactory system'. In: *Journal of Comparative Neurology* 372.4 (1996), pp. 487–514 (cited on page 36).

[91] Gardner M. 'Tilings with Convex Polygons'. In: *Time Travel and Other Mathematical Bewilderments*. Ed. by W.H. Freeman. New York: Oxford University Press, 1988. Chap. 13, pp. 162–176 (cited on page 36).

[92] S. Kondo and T. Miura. 'Reaction-Diffusion Model as a Framework for Understanding Biological Pattern Formation'. In: *Science* 329.5999 (2010), pp. 1616–1620 (cited on page 36).

[93] A. Tavanaei et al. 'Deep learning in spiking neural networks'. In: *Neural Networks* 111 (2019), pp. 47–63 (cited on page 36).

[94] J. M. Orem. 'Rhythms of Life: The Biological Clocks that Control the Daily Lives of Every Living Thing. By Russell G Foster and Leon Kreitzman.' In: *The Quarterly Review of Biology* 80.2 (2005), pp. 266–267 (cited on page 37).

[95] J. Carr. 'Applications of centre manifold theory'. In: Springer-Verlag, 1981 (cited on page 37).

[96] C. Kosniowski. *A First Course in Algebraic Topology*. Cambridge University Press, 1980 (cited on page 37).

[97] H.K. Khalil. *Nonlinear systems; 3rd ed.* Upper Saddle River, NJ: Prentice-Hall, 2002 (cited on page 38).

[98] F. Verhulst and T. Bakri. 'The dynamics of slow manifold'. In: *J. Indones. Math. Soc.* 13 (Jan. 2012) (cited on page 39).

[99] A.N. Tikhonov. 'Systems of differential equations containing a small parameter multiplying the derivative'. In: *Mat. Sb.* 31 (1952), pp. 575–586 (cited on page 40).

[100] R. FitzHugh. 'Impulses and Physiological States in Theoretical Models of Nerve Membrane'. In: *Biophysical Journal* 1.6 (1961), pp. 445–466 (cited on page 40).

[101] A.L. Hodgkin and A.F. Huxley. 'A quantitative description of membrane current and its application to conduction and excitation in nerve'. In: *The Journal of Physiology* 117.4 (1952), pp. 500–544 (cited on page 40).

[102] J. Nagumo, S. Arimoto, and S. Yoshizawa. 'An active pulse transmission line simulating nerve axon'. In: *Proceeding IRE* 50 (1962), pp. 2061–2070 (cited on page 40).

[103] K. Ciesielski. 'The Poincaré-Bendixson Theorem: From Poincaré to the XXIst century'. In: *Central European Journal of Mathematics* 10 (July 2001) (cited on page 41).

[104] J. Keener and J. Sneyd. *Mathematical Physiology. II: Systems Physiology*. Springer, 2009 (cited on page 42).

[105] G. Manganaro, L. Fortuna, and P. Arena. *Cellular Neural Networks*. 1st. Berlin, Heidelberg: Springer-Verlag, 1999 (cited on page 43).

[106] F.F. Chen. *Introduction to Plasma Physics and Controlled Fusion*. Springer, 2016 (cited on page 45).

[107] J.D. Lawson. 'Some Criteria for a Power Producing Thermonuclear Reactor'. In: *Proceedings of the Physical Society. Section B* 70.1 (Jan. 1957), pp. 6–10 (cited on page 45).

[108] J. Radon. 'Über die Bestimmung von Funktionen durch ihre Integralwerte längs gewisser Mannigfaltigkeiten'. In: *Akad. Wiss.* 69 (1917), pp. 262–277 (cited on page 45).

[109] H.P. Hiriyannaiah. 'X-ray computed tomography for medical imaging'. In: *IEEE Signal Processing Magazine* 14.2 (Mar. 1997), pp. 42–59 (cited on page 45).

[110] M Bertero, C De Mol, and E R Pike. 'Linear inverse problems with discrete data. I. General formulation and singular system analysis'. In: *Inverse Problems* 1.4 (Nov. 1985), pp. 301–330 (cited on page 45).

[111] M. Bertero, T. A. Poggio, and V. Torre. 'Ill-posed problems in early vision'. In: *Proceedings of the IEEE* 76.8 (Aug. 1988), pp. 869–889 (cited on page 45).

[112] Jordan Cavalier et al. 'Tomographic reconstruction of tokamak edge turbulence from single visible camera data and automatic turbulence structure tracking'. In: *Nuclear Fusion* 59.5 (Apr. 2019), p. 056025 (cited on page 46).

[113] T. Odstrčil et al. 'Optimized tomography methods for plasma emissivity reconstruction at the ASDEX Upgrade tokamak'. In: *Review of Scientific Instruments* 87.12 (2016), p. 123505 (cited on page 46).

[114] R. Nguyen van yen et al. 'Tomographic reconstruction of tokamak plasma light emission from single image using wavelet-vaguelette decomposition'. In: *Nuclear Fusion* 52.1 (Nov. 2011) (cited on page 46).

[115] R. N. Bracewell. 'Numerical Transforms'. In: *Science* 248.4956 (1990), pp. 697–704 (cited on page 46).

[116] L. Montgomery Smith, Dennis R. Keefer, and S.I. Sudharsanan. 'Abel inversion using transform techniques'. In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 39.5 (1988), pp. 367–373 (cited on page 46).

[117] A. Coutlis et al. 'Frequency response identification of the dynamics of a Tokamak plasma'. In: *IEEE Transactions on Control Systems Technology* 8.4 (2000), pp. 646–659 (cited on page 47).

[118] R. Albanese, E. Coccorese, and G. Rubinacci. 'Plasma modelling for the control of vertical instabilities in tokamaks'. In: *Nuclear Fusion* 29.6 (June 1989), pp. 1013–1023 (cited on page 47).

[119] L. Ljung. 'Prediction error estimation methods'. In: *Circuits, Systems, and Signal Processing* 21 (Jan. 2002), pp. 11–21 (cited on page 47).

[120] M. Viberg. 'Subspace Methods in System Identification'. In: *IFAC Proceedings Volumes* 27.8 (1994). IFAC Symposium on System Identification (SYSID'94), Copenhagen, Denmark, 4-6 July, pp. 1–12 (cited on page 48).

[121] S. Joe Qin. 'An overview of subspace identification'. In: *Computers & Chemical Engineering* 30.10 (2006), pp. 1502–1513 (cited on page 48).

[122] Q. Zhang and A. Benveniste. 'Wavelet networks'. In: *IEEE Trans Neural Netw* 3.6 (1992), pp. 889–898 (cited on page 49).

[123] C. Harlişca and L. Szabo. 'Wavelet analysis and Park's Vector based condition monitoring of induction machines'. In: *Journal of Computer Science and Control Systems* 4 (Jan. 2011), pp. 35–38 (cited on page 49).

[124] P. Dan Cristea, R. Tuduce, and A. Cristea. 'Time series prediction with wavelet neural networks'. In: Feb. 2000, pp. 5–10 (cited on page 49).

[125] Y. He, F. Chu, and B. Zhong. 'A Hierarchical Evolutionary Algorithm for Constructing and Training Wavelet Networks'. In: *Neural Computing and Applications* 10 (Apr. 2002), pp. 357–366 (cited on page 49).

[126] A. Wills et al. 'Identification of Hammerstein–Wiener Models'. In: *Automatica* 49 (Jan. 2013), pp. 70–81 (cited on page 49).

[127] M. Schoukens et al. 'Parametric identification of parallel Wiener–Hammerstein systems'. In: *Automatica* 51 (Jan. 2015), pp. 111–122 (cited on page 49).

[128] A.M. Priester, S. Natarajan, and M.O. Culjat. 'Robotic ultrasound systems in medicine'. In: *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control* 60.3 (2013), pp. 507–523 (cited on page 52).

[129] S.E. Salcudean et al. 'Robot-Assisted Diagnostic Ultrasound – Design and Feasibility Experiments'. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI'99*. Ed. by Chris Taylor and Alain Colchester. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 1062–1071 (cited on page 52).

[130] K. Mathiassen et al. 'An Ultrasound Robotic System Using the Commercial Robot UR5'. In: *Frontiers in Robotics and AI* 3 (2016), p. 1 (cited on page 52).

[131] A. Vilchis Gonzales et al. 'TER: A System for Robotic Tele-echography'. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001*. Ed. by W.J. Niessen and M.A. Viergever. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 326–334 (cited on page 52).

[132] P. Arbeille et al. 'Realtime Tele-Operated Abdominal and Fetal Echography in 4 Medical Centres, from one Expert Center, using a Robotic Arm ISDN or Satellite Link'. In: *2008 IEEE International Conference on Automation, Quality and Testing, Robotics*. Vol. 1. 2008, pp. 45–46 (cited on page 52).

[133] R. Skalak, S. Chien, and R.E. Mates. 'Handbook of Bioengineering'. In: *Journal of Biomechanical Engineering* 109.4 (Nov. 1987), pp. 357–357 (cited on page 52).

[134] H. Akaike. 'Information Theory and an Extension of the Maximum Likelihood Principle'. In: *Selected Papers of Hirotugu Akaike*. Ed. by Emanuel Parzen, Kunio Tanabe, and Genshiro Kitagawa. New York, NY: Springer New York, 1998, pp. 199–213 (cited on page 55).

[135] G. Schwarz. 'Estimating the Dimension of a Model'. In: *The Annals of Statistics* 6.2 (1978), pp. 461–464 (cited on page 56).

[136] H. Lütkepohl. 'Comparison of criteria for estimating the order of a vector autoregressive process'. In: *Journal of Time Series Analysis* 6.1 (1985), pp. 35–52 (cited on page 56).

[137] C. Giraud. *Introduction to High-Dimensional Statistics*. CBC Press, 2014 (cited on page 56).

[138] E.J. Hannan and B.G. Quinn. 'The Determination of the Order of an Autoregression'. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 41.2 (1979), pp. 190–195 (cited on page 56).

[139] A. Khinchin. 'Über einen Satz der Wahrscheinlichkeitsrechnung'. In: *Fundamenta Mathematicae* 6.1 (1924), pp. 9–20 (cited on page 56).

[140] L. Chua and L. Yang. 'Cellular neural networks: Theory'. In: *Circuits and Systems, IEEE Transactions on* 35 (Nov. 1988), pp. 1257–1272 (cited on page 59).

[141] M.E. Young et al. 'Positive and negative patterning in human causal learning'. In: *The Quarterly Journal of Experimental Psychology Section B* 53.2b (2000), pp. 121–138 (cited on page 60).

[142] E. Yavuz, J. Turner, and T. Nowotny. 'GeNN: a code generation framework for accelerated brain simulations'. In: *Scientific Reports* 6.1 (Jan. 2016), p. 18854 (cited on page 60).

[143] E. E. Rohmer, S.P.N. Singh, and M. Freese. 'V-REP: A versatile and scalable robot simulation framework'. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2013, pp. 1321–1326 (cited on page 60).

[144] P. Arena, L. Patane, and A.G. Spinosa. 'Insect inspired spatial-temporal cellular processing for feature-action learning'. In: *2017 European Conference on Circuit Theory and Design (ECCTD)*. Sept. 2017, pp. 1–4 (cited on page 60).

[145] P. Arena et al. 'A CNN-based neuromorphic model for classification and decision control'. In: *Nonlinear Dynamics* 95 (Dec. 2018) (cited on page 60).

[146] P. Arena, L. Patanè, and A.G. Spinosa. 'Data-based analysis of Laplacian Eigenmaps for manifold reduction in supervised Liquid State classifiers'. In: *Information Sciences* 478 (2019), pp. 28–39 (cited on page 61).

[147] P. Arena, L. Patané, and A.G. Spinosa. 'Structural and input reduction in a ESN for robotic navigation tasks'. In: *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. Oct. 2019, pp. 3531–3536 (cited on page 62).

[148] M. L. Anderson. 'Neural reuse: A fundamental organizational principle of the brain'. In: *Behavioral and Brain Sciences* 33.4 (2010), pp. 245–266 (cited on page 62).

[149] P. Arena, L. Patanè, and A.G. Spinosa. 'Robust modelling of binary decisions in Laplacian Eigenmaps-based Echo State Networks'. In: *Engineering Applications of Artificial Intelligence* 95 (2020), p. 103828 (cited on page 63).

[150] S.L Hooper. 'Central Pattern Generators'. In: *eLS*. American Cancer Society, 2001 (cited on page 65).

[151] P. Arena, L. Patané, and A.G. Spinosa. 'A nullcline-based control strategy for PWL-shaped oscillators'. In: *Nonlinear Dynamics* (June 2019) (cited on page 65).

[152] M. Bucolo et al. 'Human Machine Models for Remote Control of Ultrasound Scan Equipment'. In: *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*. 2020, pp. 1–6 (cited on page 66).