



Universitat Autònoma de Barcelona

Degree Thesis

---

# IMPROVEMENTS OF DETERMINISTIC PROCESSES THROUGH NEURAL NETWORKS

---

Author:  
Gerard Lahuerta Martín

Supervisor:  
Dr. Lluís Alsedà Soler

A THESIS SUBMITTED IN FULFILMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF COMPUTATIONAL  
MATHEMATICS AND DATA ANALYTICS IN THE

SCIENCE FACULTY

June 2024



## Declaration of Authorship

---

## Abstract

## Acknowledgements

I would like to express my gratitude to my family for their support throughout my thesis, degree, and my entire life.

I am especially grateful to my mother, Maria Montserrat, for her encouragement and guidance in facing the challenges of this thesis and my career.

Thanks to all the peoples that treat me as I was part of their family.

I also appreciate the encouragement and support of my colleagues.

Finally, I want to give a special mention to Dr. Lluís Alsedà for guiding me through the thesis and being an inspiration.

Thank you all for everything.

---

# Contents

## Preface

## **Introducción**



# 1 Neural Networks

A neural Network is made of individual and independent elements connected between them, passing and managing the information through the network formed.

In this thesis we will focus on one of the simplest networks, a multilayer perceptron, to test the different methods of optimization.

## 1.1 Multilayer Perceptron and Perceptron neuron

One of the simple Neural Networks to analyse is the Multilayer Perceptron<sup>1</sup>.

It was first proposed by Frank Rosenblat<sup>2</sup> in 1958 (nevertheless its approach did not learn either produce accurate results).

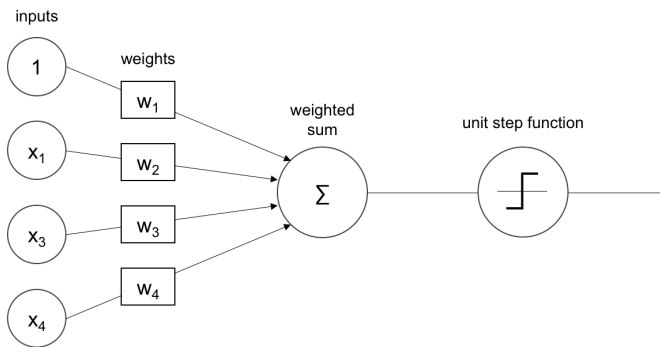


Figure 1: Schema of the Perceptron neuron

This Neural Network is formed by elements (the artificial neurons) called *Perceptrons*. This neuron is formed by input, weight and activation functions.

Nevertheless, the most important part of the neuron, and that determines significantly the capabilities of the neuron, is the activation function (which returns the output of the neuron).

The traditional activation function used in the Multilayer Perceptron is the Sigmoid:

$$f(x) = \frac{1}{1 + e^{-w \cdot x}}, \text{ where: } x, w \in \mathbb{R}^n$$

The Multilayer Perceptron Topology can be splitted into layers of three types:

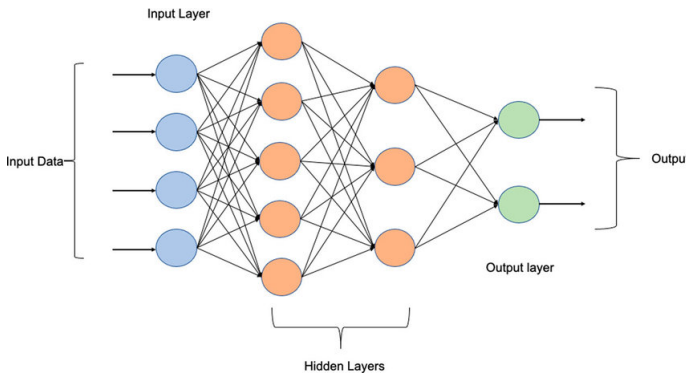


Figure 2: Schema of the Multilayer perceptron

- **Input Layer:** The initial set of neurons of the Multilayer Perceptron.
- **Output Layer:** The final set of neurons of the Multilayer Perceptron.
- **Hidden Layers:** The set of neurons (in layers) between the input and output layers.

<sup>1</sup>To obtain more information about Multilayer Perceptron functionality check ,

<sup>2</sup>Frank Rosenblat, psychologist and father of deep learning, check its [biography](#).

## 1.2 Why Neural Networks works

It is important to remark that what gives the capacity of model any system is because we can express every problem as a function (no matter if its a classification task, probability function, or prediction, regression function).

The association between a task and a function allow to apply the **Universal Approximation Theorem**.

- 
- **Definition:** Universal Approximation Theorem.

For any function  $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$  and a subset  $D \subset \mathbb{R}^n$  where  $f$  is continuous at all  $D$ ,  $\exists \{(w_i, b_i, c_i)\}_{i=0}^k$  that:

$$f(\vec{x}) - \lim_{k \rightarrow \infty} \sum_{i=0}^k c_i \sigma(w^T \cdot \vec{x} + b_i) = 0$$

Where  $w_i \in \mathbb{R}^n$ ,  $b_i \in \mathbb{R}$ ,  $c_i \in \mathbb{R}^m$ ,  $\vec{x} \in D$  and  $\sigma$  the sigmoid function.

---

## References

- [1] WIKIPEDIA, *Multilayer perceptron*,  
[https://en.wikipedia.org/wiki/Multilayer\\_perceptron](https://en.wikipedia.org/wiki/Multilayer_perceptron)
  - [2] WIKIPEDIA, *Perceptrón*,  
<https://es.wikipedia.org/wiki/Perceptr%C3%B3n>
  - [3] WIKIPEDIA, *Activation function*,  
[https://en.wikipedia.org/wiki/Activation\\_function](https://en.wikipedia.org/wiki/Activation_function)
  - [4] WIKIPEDIA, *Frank Rosenblatt*,  
[https://es.wikipedia.org/wiki/Frank\\_Rosenblatt](https://es.wikipedia.org/wiki/Frank_Rosenblatt)
  - [5] WIKIPEDIA, *Activating Function*,  
[https://en.wikipedia.org/wiki/Activating\\_function](https://en.wikipedia.org/wiki/Activating_function)
  - [6] WIKIPEDIA, *Artificial neural network*,  
[https://en.wikipedia.org/wiki/Artificial\\_neural\\_network](https://en.wikipedia.org/wiki/Artificial_neural_network)
  - [7] WIKIPEDIA, *Neural network*,  
[https://en.wikipedia.org/wiki/Neural\\_network](https://en.wikipedia.org/wiki/Neural_network)
  - [8] WIKIPEDIA, *Universal Approximation Theorem*,  
[https://en.wikipedia.org/wiki/Universal\\_approximation\\_theorem](https://en.wikipedia.org/wiki/Universal_approximation_theorem)
  - [9] WIKIPEDIA, *Mean Squared Error*,  
[https://en.wikipedia.org/wiki/Mean\\_squared\\_error](https://en.wikipedia.org/wiki/Mean_squared_error)
  - [10] WIKIPEDIA, *Mean Absolute Error*,  
[https://en.wikipedia.org/wiki/Mean\\_absolute\\_error](https://en.wikipedia.org/wiki/Mean_absolute_error)
  - [11] WIKIPEDIA, *Cross-Entropy*,  
<https://en.wikipedia.org/wiki/Cross-entropy>
-

- [12] WIKIPEDIA, *BackPropagation*,  
<https://en.wikipedia.org/wiki/Backpropagation>
- [13] DANTZIG, G.B. y P. WOLFE, «Decomposition principle for linear programs», *Operations Research*, **8**, págs. 101–111, 1960.