

---

# MI201 - Apprentissage Automatique

---

Résumé Théorique - Neural Network

12 septembre 2024

---

Guilherme Nunes Trofino  
2022-2024

# Table des matières

---

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Apprentissage</b>	<b>3</b>
2.1	Algorithme . . . . .	3
2.1.1	Avantages . . . . .	3
2.1.2	Incovenients . . . . .	3
2.1.3	Applications . . . . .	4

# 1. Introduction

---

**Repository** Hello! My name is Guilherme Nunes Trofino and this is my LaTeX notebook of MI201 - Apprentissage Automatique that can be found in my GitHub repository : [https://github.com/tr0fin0/classes\\_ensta](https://github.com/tr0fin0/classes_ensta).

**Disclaimer** This notebook is made so it may help others in this subject and is not intend to be used to cheat on tests so use it by your on risk.

**Suggestions** If you may find something on this document that does not seam correct please reach me by e-mail : [guitrofino@gmail.com](mailto:guitrofino@gmail.com).

## 2. Apprentissage

### 2.1. Algorithme

On considère que l'algorithme peut être définie par la définition suivante :

**Définition 2.1.** .

La choix de  $n$  déterminera la qualité de la prédiction. Si on réduire  $n$  la qualité de la prediction améliore et le sur-apprentissage diminue. Généralement on aura le comportement suivant :

$$\begin{array}{lll} n \uparrow & \text{biais} \downarrow & \text{variance} \uparrow \\ n \downarrow & \text{biais} \uparrow & \text{variance} \downarrow \end{array}$$

TABLE 2.1 : Comportement Neural Network

Comme fonction d'activation on considère souvent la fonction **relu** pour sa simplicité et capacité de reproduire d'autres fonctions :

**Définition 2.2.** On considère que la **relu** sera donne pour :

$$\boxed{relu(x) = \max(x, 0)} \quad (2.1)$$

C'est possible d'avoir plusieurs fonctions à partir de la **relu** comme

$$|x| = relu(x) + relu(-x) \quad (2.2)$$

$$\|\mathbf{x}\|_1 = |x_1| + |x_2| = relu((1, 0) \cdot \mathbf{x}) + relu((-1, 0) \cdot \mathbf{x}) + relu((0, 1) \cdot \mathbf{x}) + relu((0, -1) \cdot \mathbf{x}) \quad (2.3)$$

$$\max(x, y) = relu(x - y) \frac{x}{x - y} + relu(y - x) \frac{y}{y - x} \quad (2.4)$$

$$\min(x, y) = relu(x - y) \frac{y}{x - y} + relu(y - x) \frac{x}{y - x} \quad (2.5)$$

Comme les réseaux de neurones sont beaucoup utilisés on peut faire une modification de structure pour implémenter le **Deep Learning** :

**Définition 2.3.** On utilise des **Neurones Convolutif**...

**Remarque.** **Neurones Convolutifs** sont une variation de construction de réseaux de neurones où...

#### 2.1.1. Avantages

Dans ce cas, cet algorithme

#### 2.1.2. Inconvénients

On peut citer :

1. pour apprendre par coeur il suffit d'avoir  $2nd + n + 1$  neurones ;
  - (a)  $2nd$  : numéro d'entrées ;
  - (b)  $n$  : numéro de données ;
  - (c) 1 : sorti ;

### 2.1.3. Applications

Cet algorithme est souvent utilisé