



**STATISTIQUE
SCIENCE DES DONNÉES**
UNIVERSITÉ DE MONTPELLIER

Apprentissage Statistique

TP n°5 : Perceptron multicouche

Rédigé par

CÔME OLIVIER

PRALON Nicolas

SENE Assane

IMAG

INSTITUT MONTPELLIERAIN
ALEXANDER GROTHENDIECK

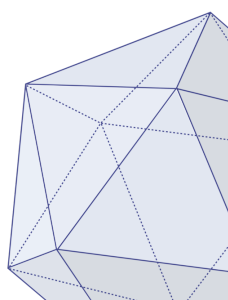


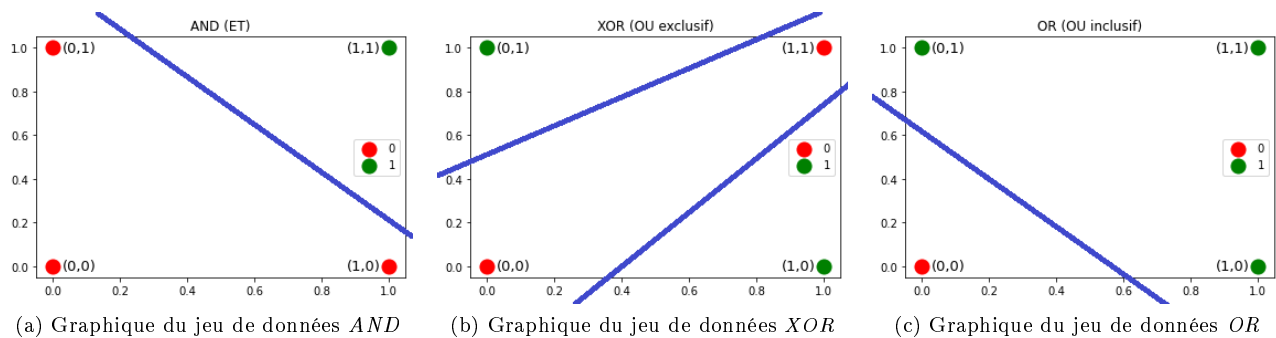
Table des matières

Introduction	2
Question 1	2
Question 2	2
Question 3	2
Question 3	3
0.0.1 a)	3
Question 4	3
0.0.2 b)	3
0.0.3 c)	3
Question 5	3
Question 6	3
Bibliographie	4

Introduction

Question 1

Présentons très synthétiquement le jeu de données pour le *AND*, *XOR* et *OR* :



Les difficultés de classification résident dans le fait que les données du *XOR* ne sont pas linéairement séparables. Sur le graphique (b) on voit qu'il y a deux droites de séparations et non une seule. De ce fait, un seul neurone ne peut réussir à classer les données.

Question 2

Définissons un classifieur *MLP* pour apprendre l'opérateur *AND* :

```
Score obtenu sur les données de test pour l'apprentissage de l'opérateur AND: 1.0
```

FIGURE 1 – Score obtenu par le classifieur *MLP* sur les données de test de l'opérateur *AND*

Dans le cas où l'on souhaite apprendre l'opérateur *AND*, le classifieur *MLP* ne comprenant aucune couche cachée fournit de très bons résultats en terme de prédiction. Comme on peut le voir sur la figure 1, le classifieur ne fait aucune erreur de prédiction (le score vaut 1). Pour calculer ce score nous avons utilisé la fonction `score(x_test, y_test)` de *sklearn*.

Question 3

Définissons un classifieur *MLP* pour apprendre l'opérateur *OR* :

```
Score obtenu sur les données de test pour l'apprentissage de l'opérateur OR: 1.0
```

FIGURE 2 – Score obtenu par le classifieur *MLP* sur les données de test de l'opérateur *OR*

Dans le cas où l'on souhaite apprendre l'opérateur *OR*, le classifieur *MLP* ne comprenant aucune couche cachée fournit de très bons résultats en terme de prédiction. Comme on peut le voir sur la figure 2, le classifieur ne fait aucune erreur de prédiction (le score vaut 1). Pour calculer ce score nous avons utilisé la fonction `score(x_test, y_test)` de *sklearn*.

Question 4

0.0.1 a)

Définissons un classifieur *MLP* sans couches cachées pour apprendre l'opérateur *XOR* :

```
Score obtenu sur les données de test pour l'apprentissage de l'opérateur XOR sans couche cachée: 0.5
```

FIGURE 3 – Score obtenu par le classifieur *MLP* sur les données de test de l'opérateur *XOR* sans utiliser de couches cachées

Dans le cas où l'on souhaite apprendre l'opérateur *XOR*, le classifieur *MLP* ne comprenant aucune couche cachée fournit de mauvais résultats en terme de prédiction. Comme on peut le voir sur la figure 3, la précision de prédiction du classifieur est de 50% (le score vaut 0.5), ce qui est assez bas.

Cela s'explique par le fait qu'un perceptron ne peut apprendre que sur des données linéairement séparables or ce n'est ici pas le cas.

0.0.2 b)

Définissons un classifieur *MLP* avec deux couches cachées pour apprendre l'opérateur *XOR* :

```
Score obtenu sur les données de test pour l'apprentissage de l'opérateur XOR avec les 2 couches cachées: 0.75
```

FIGURE 4 – Score obtenu par le classifieur *MLP* sur les données de test de l'opérateur *XOR* en utilisant 2 couches cachées

Le classifieur *MLP* comprenant deux couches cachées (4 neurones sur la première couche et 2 neurones sur la deuxième) fournit de meilleurs résultats (en terme de prédiction) que le classifieur ne comprenant aucune couche cachée. En effet, comme on peut le voir sur la figure 4, la précision de prédiction de ce nouveau classifieur est de 75% contre 50% pour le précédent.

Cela s'explique par le fait que l'ajout de couches cachées dans un perceptron permet de transformer un problème non linéairement séparable en un problème linéairement séparable. Ce nouveau classifieur (perceptron multi-couche) sera donc en mesure d'apprendre l'opérateur *XOR* contrairement au précédent.

0.0.3 c)

Classifieur *MLP* avec deux couches cachées et des fonctions d'activation hyperbolique pour apprendre l'opérateur *XOR* :

```
Score obtenu avec la fonction d'activation tanh: 1.0
```

FIGURE 5 – Score obtenu par le classifieur *MLP* sur les données de test de l'opérateur *XOR* en utilisant 2 couches cachées et des fonctions d'activation *tanh*

Les résultats obtenus ici sont encore mieux que ceux obtenus dans les deux cas précédents. En effet, comme on peut le voir sur la figure 5, la précision de prédiction de ce nouveau classifieur est de 100% contre 75% et 50% pour les deux précédents. Les fonctions d'activation hyperboliques (*tanh*) cherchent des séparateurs non linéaires or nos données sont ici non linéairement séparables. Par conséquent, ces fonctions d'activation seront bien adaptées à notre situation. Cela explique pourquoi les résultats obtenus ici sont mieux que ceux obtenus précédemment.

Question 5

blabla

Question 6

blabla

Bibliographie

- [1] WikiStat. An introduction to network inference and mining, *Article*
http://www.nathalievialaneix.eu/doc/pdf/wikistat-network_compiled.pdf
- [2] PNAS. Modularity and community structure in networks (2015), *Article*
<https://www.pnas.org/doi/10.1073/pnas.0601602103#abstract>
- [3] Wikipédia (2022). Méthode de Louvain, *Article*
https://fr.wikipedia.org/wiki/Méthode_de_Louvain
- [4] igraph, *Documentation*
<https://igraph.org/python/versions/latest/>
- [5] igraph, *Documentation*
https://igraph.org/python/versions/latest/tutorials/visualize_communities/visualize_communities.html
- [6] igraph, *Tutoriel*
https://igraph.org/python/api/latest/igraph._igraph.GraphBase.html#Erdos_Renyi