

**Projet : Apprentissage de la fonction
OR avec un réseau de neurones
simple (Keras) et un arbre de
décision (scikit-learn)**

Nom : Thinhinane Ziane

Master : M1 Informatique RSA

Matière : Réseaux de Neurones

Date : 29 avril 2025

Table des matières

Introduction	2
Données et Méthodologie	3
Entraînement et Ajustements	4
Résultats et Graphique de Séparation	5
Bilan intermédiaire : Réseau de Neurones	6
Apprentissage de la fonction OR avec un Arbre de Décision	7
Comparaison entre le Réseau de Neurones et l'Arbre de Décision	8
Conclusion Générale	9

Introduction

Le but de ce projet est d'appliquer les concepts théoriques des méthodes d'apprentissage supervisé vues en cours, en réalisant l'apprentissage de la fonction logique OR par deux approches différentes.

L'objectif est de construire, entraîner, tester et sauvegarder un réseau de neurones simple utilisant Keras, ainsi que d'entraîner un arbre de décision avec scikit-learn, afin de comparer leurs performances respectives.

Ce projet a été réalisé en utilisant les bibliothèques Python TensorFlow/Keras pour le réseau de neurones et scikit-learn pour l'arbre de décision.

Données et Méthodologie

Les données utilisées représentent les entrées et sorties de la fonction logique OR :

x_1	x_2	OR(x_1, x_2)
0	0	0
0	1	1
1	0	1
1	1	1

Le réseau construit comporte :

- Deux neurones d'entrée (x_1 et x_2),
- Un seul neurone de sortie (y),
- Une fonction d'activation sigmoïde,
- Une fonction de perte : Binary Crossentropy (BCE),
- Un optimiseur : descente de gradient stochastique (SGD).

Un entraînement supervisé a été réalisé sur ce réseau.

Entraînement et Ajustements

Lors des premiers essais, l'entraînement sur 1000 époques n'était pas suffisant pour obtenir une bonne convergence : les prédictions n'étaient pas toutes correctes.

Après observation, le nombre d'époques a été progressivement augmenté jusqu'à 2500 époques pour permettre au réseau d'apprendre correctement la fonction OR.

Une fois le modèle satisfaisant, il a été sauvegardé dans un fichier `train_OR.h5`, ce qui permet de recharger directement le réseau sans avoir à le réentraîner.

En effet, sans cette sauvegarde, une nouvelle exécution réinitialiserait les poids de manière aléatoire, modifiant ainsi les prédictions.

Résultats et Graphique de Séparation

Après entraînement et test, les prédictions du réseau sur les 4 entrées sont :

Entrée (x_1, x_2)	Sortie attendue	Sortie prédite (arrondie)
(0,0)	0	0
(0,1)	1	1
(1,0)	1	1
(1,1)	1	1

Le réseau a donc appris correctement la fonction logique OR.

Graphique de séparation du réseau

Une représentation graphique a été réalisée pour visualiser la séparation des classes.

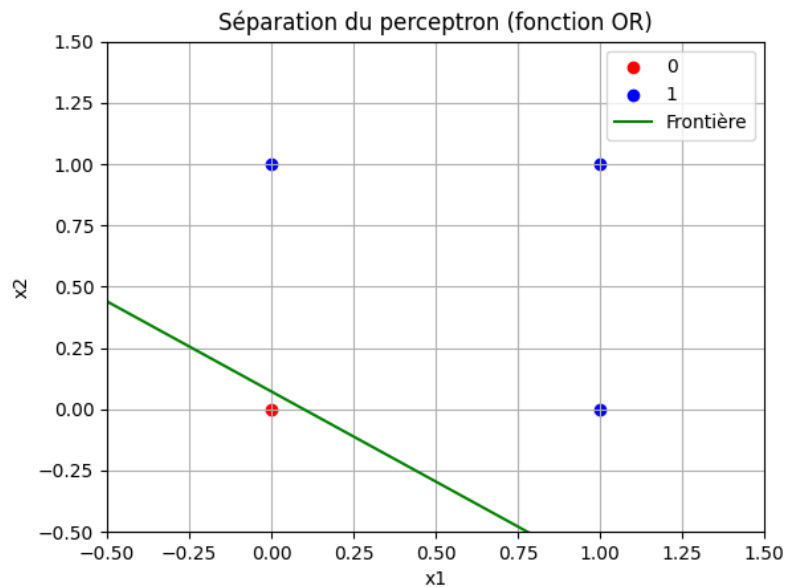


FIGURE 1 – Graphique de la séparation des classes par le réseau.

Bilan intermédiaire : Réseau de Neurones

La première approche mise en œuvre a consisté à entraîner un réseau de neurones simple pour apprendre la fonction OR.

Le réseau a appris efficacement la fonction après ajustement du nombre d'époques.

Le modèle a été sauvegardé (`train_OR.h5`) et peut être rechargé pour effectuer des prédictions sans nécessiter un nouvel entraînement.

Dans la section suivante, J'explore une autre méthode d'apprentissage supervisé : l'arbre de décision.

Apprentissage de la fonction OR avec un Arbre de Décision

Pour évaluer une approche différente, J'ai également utilisé un Arbre de Décision pour apprendre la fonction OR. L'entraînement est immédiat sur ces données simples et le modèle réussit parfaitement la tâche.

Entrée (x_1, x_2)	Sortie attendue	Sortie prédite par l'Arbre
(0,0)	0	0
(0,1)	1	1
(1,0)	1	1
(1,1)	1	1

L'arbre de décision parvient également à apprendre parfaitement la fonction OR.

Comparaison entre le Réseau de Neurones et l'Arbre de Décision

J'ai comparé les deux approches sur la tâche d'apprentissage de la fonction OR :

Critère	Réseau de Neurones	Arbre de Décision
Précision sur les données	100%	100%
Temps d'entraînement	1s (2500 époques)	Instantané
Complexité d'implémentation	Moyenne (choix d'époques)	Très simple
Interprétabilité	Moyenne	Excellente (arbre visible)

Les deux méthodes réussissent, mais l'arbre de décision est plus rapide et plus facile à interpréter pour ce type de données simples.

Conclusion Générale

Ce projet a permis de mettre en pratique l'apprentissage supervisé à travers deux approches différentes : un réseau de neurones simple et un arbre de décision.

Le réseau de neurones, après ajustement du nombre d'époques, a parfaitement appris la fonction OR, montrant l'importance du réglage des paramètres d'entraînement.

L'arbre de décision a également réussi à apprendre cette fonction, de manière plus rapide et avec une interprétation plus intuitive.

La comparaison des deux méthodes a mis en évidence leurs forces respectives : le réseau de neurones est plus général pour des tâches complexes, tandis que l'arbre de décision est très efficace pour des problèmes simples et bien structurés.

Ce travail facultatif m'a permis de consolider ma compréhension des modèles d'apprentissage automatique de base et d'expérimenter différentes approches sur un problème simple.