Rapport TP – Apprentissage de la fonction XOR avec un MLP

Nom : Smaoui
Prénom : Yassine
CNE : D110112916

Partie 1 : Objectif et configuration du réseau

Le but de ce TP est d'implémenter un réseau de neurones multicouche (MLP) capable d'apprendre la fonction logique XOR, qui est non linéairement séparable.

La fonction XOR est définie par :

Entrée A	Entrée B	Sortie XOR
0	0	0
0	1	1
1	0	1
1	1	0

Configuration du réseau :
— 2 entrées
— 1 couche cachée (3 neurones, activation ReLU)
— 1 neurone de sortie (activation sigmoïde)
— Apprentissage via rétropropagation
— Fonction de coût : MSE
— Taux d'apprentissage : 0.1, 5000 époques
Structure modulaire :
— Classe `Layer` : gère les poids, biais et activation
— Classe `Model` : orchestre les couches, l'apprentissage et les prédictions

Partie 2 : Fonctions d'activation et classe Layer

Deux fonctions d'activation ont été utilisées :
— ReLU : $\text{ReLU}(x) = \max(0, x)$
— Sigmoïde : $\sigma(x) = \frac{1}{1+e^{-x}}$

Classe `Layer` :
— Initialisation des poids W et biais b
— `forward(X)` : $Z = XW + b$, $A = activation(Z)$
— `backward(dA)` : calcule les gradients dW, db et met à jour les paramètres

Partie 3 : Structure de la classe Model

Classe `Model` :
— `add()` : ajoute une couche
— `forward()` : propagation avant
— `backward()` : rétropropagation
— `compute_loss()` : calcule la MSE/2
— `train()` : boucle d'entraînement
— `predict()` : prédiction binaire à partir d'un seuil (0.5)

Partie 4 : Expérimentation sur XOR

Données utilisées :

Entrée A	Entrée B	Sortie attendue
0	0	0
0	1	1
1	0	1
1	1	0

Paramètres :
— 2 entrées
— 1 couche cachée : 3 neurones (ReLU)
— 1 sortie : 1 neurone (sigmoïde)
— Taux d'apprentissage : 0.1
— 5000 époques
Extrait de loss :

```
Epoch 0, Loss: 0.1601
...
Epoch 4900, Loss: 0.0052
```

Prédictions finales :
0, 0 $\rightarrow$ 0
0, 1 $\rightarrow$ 1
1, 0 $\rightarrow$ 1
1, 1 $\rightarrow$ 0