

TD4 – Réseaux de neurones pour la régression et la classification

L'objectif de ce TD est d'implémenter en *Python* un modèle de réseau de neurones. Deux versions seront développées, l'une pour un cas de régression et l'autre de classification. Des expérimentations seront ensuite menées pour évaluer la performance des modèles sur des jeux d'essai.

1. Présentation des éléments fournis

Un fichier Python (**rn_regression.py**) est fourni, contenant des fonctions à compléter pour rendre le programme opérationnel dans le cas de la régression.

Pour tester les fonctions développées, quatre fichiers de données sont fournis :

- **food_truck.txt** : contient un jeu de données caractérisé par une seule variable prédictive (la taille de la population des villes, dans la première colonne). La variable cible (deuxième colonne) correspond au bénéfice d'un food truck dans cette ville. Le problème associé à ce jeu de données est de prédire le bénéfice qu'un food truck pourrait réaliser en fonction de la taille de la ville.
- **houses.txt** : contient un jeu de données caractérisé par deux variables prédictives (la superficie d'une maison et le nombre de pièces, dans les deux premières colonnes). La variable cible (troisième colonne) correspond au prix de la maison. Le problème associé à ce jeu de données est de prédire le prix d'une maison en fonction de sa superficie et de son nombre de pièces.
- **scores.txt** : contient un jeu de données caractérisé par deux variables prédictives (les scores des étudiants à deux examens, dans les deux premières colonnes). La variable cible (troisième colonne) indique si l'étudiant est admis (1) ou non (0) à l'université. Le problème associé à ce jeu de données est de prédire si un étudiant sera admis à l'université en fonction des scores obtenus aux deux examens.
- **iris.txt** : contient un célèbre jeu de données constitué de 150 échantillons correspondant à des plantes du type "Iris". Ces échantillons sont caractérisés par 4 variables prédictives (longueur et largeur des sépales, longueur et largeur des pétales). Chaque échantillon appartient à l'une des trois familles d'Iris : Setosa (code 0), Versicolour (code 1), Virginica (code 2).

2. Réseau de neurones pour la régression

Complétez les fonctions dans le fichier **rn_regression.py** pour rendre le programme opérationnel. Il est recommandé de suivre l'ordre des fonctions indiqué dans la section précédente et de les tester à chaque étape.

Les fichiers **food_truck.txt**, **houses.txt** et **scores.txt** peuvent être utilisés pour tester le programme.

3. Réseau de neurones pour la classification

En vous inspirant du programme dans le fichier **rn_regression.py**, écrivez un nouveau programme dans un fichier nommé **rn_classification.py** pour implémenter un réseau de neurones pour la classification. Comme vu dans le cours, une solution typique pour la classification consiste à utiliser la fonction d'activation *softmax* pour la couche finale du réseau et une fonction de perte d'entropie croisée (*cross-entropy*).

Pour tester le programme, les fichiers **scores.txt** et **iris.txt** peuvent être utilisés.

4. Expérimentations supplémentaires

Les deux modèles de réseaux de neurones pour la régression et la classification doivent être testés sur divers jeux de données de votre choix (par exemple, provenant de : <https://archive.ics.uci.edu/>) et avec différentes configurations (nombre de couches cachées, nombre de neurones par couche cachée, types de fonctions d'activation, etc.) afin d'étudier de manière approfondie le comportement des réseaux.