

Mini projet (exercice noté)

Introduction au Deep Learning (Perceptron)

Nahid Emad (November 2023)

Ce mini projet est composé de deux parties. La première partie consiste à créer un *dataset* de m échantillons et leur m résultats/labels correspondants. La deuxième partie utilise le *dataset+labels* issus de la première partie pour entraîner un réseau d'un seul neurone.

Première partie (création de *dataset*) : Créez un *dataset* de m échantillons représentant m étudiants. Chacun des échantillons devra avoir au moins $n = 2$ caractéristiques/features (par exemple, les notes de mathématique et de français). Attribuez un résultat y à chacun des étudiants/échantillons (par exemple $y = 1$ pour *ADMIS* et $y = 0$ pour *NON – ADMIS*). La règle d'admission ou de non-admission d'un étudiant devra être définie par vous-même.

Afin de créer un échantillon de m caractéristiques, il suffit de générer m nombres aléatoires entre 0 et 20 représentant les m notes d'un étudiant. Cette opération peut être répétée m fois permettant la création d'un *dataset X* de m lignes et de m colonnes.

Concernant la création des labels, il suffit pour chacun des étudiants, de calculer la moyenne ou bien la somme pondérée des notes. Attention à faire un choix cohérent de pondération (dans une formation en informatique le coefficient des matières scientifiques est généralement plus grand que ceux des autres matières). Vous pourrez ensuite définir une fonction qui accepte en entrée cette moyenne et qui donne en sortie le résultat d'admission ou non-admission d'un étudiant (par exemple : si $Moyenne \geq 10$ alors l'étudiant est admis, sinon il n'est pas admis).

Une fois le *dataset X* et la liste de m labels/résultats définis, vous pourrez passer à la réalisation de la deuxième partie de ce projet.

Deuxième partie (apprentissage avec Perceptron) : Définir un réseau d'un seul neurone pour modéliser linéairement les données de la partie précédente. L'objectif est de définir une frontière de décision pour séparer linéairement (donc une droite) les deux groupes *ADMIS* et *NON – ADMIS* de cet ensemble d'étudiants. Pour cela utilisez la fonction de coût/erreur *LogLoss* et l'algorithme d'apprentissage *Descente de Gradient* (DG) afin de minimiser *LogLoss*. La réalisation de cette partie doit être faite en utilisant le calcul matriciel.

Évaluez les performances de votre modèle.

Vous devez préparer un rapport de 3 à 4 pages décrivant la création de votre *dataset*, les différentes étapes de l'entraînement supervisé que vous avez appliqué sur ce *dataset* à l'aide de Perceptron, l'algorithme de DG ainsi que les graphiques présentant la frontière de décision obtenue et l'évolution des erreurs au cours des itérations de l'algorithme DG.

Votre rendu doit comporter ce rapport ainsi que le code en Python (uniquement le fichier texte).