

Master en architecture des systèmes informatiques (MI11)
Master en sciences industrielles – orientation informatique (M18)
Traitement du signal 1D et 2D – Introduction aux réseaux de neurones
Année académique 2024-2025
Professeur : Jean-Marc Wagner

Projet de laboratoire
Etude et implémentation du Perceptron et de sa mise en réseau

Ce projet a pour but de vous familiariser avec les réseaux de neurones artificiels et leur apprentissage :

- Perceptron simple
- Perceptron utilisant la technique d'apprentissage de la descente du gradient
- Perceptron utilisant la technique d'apprentissage ADALINE
- Perceptron mono-couche (il s'agit d'un premier réseau de neurones travaillant en parallèle mais n'ayant aucune influence les uns sur les autres).
- Perceptron multi-couches (algorithme de rétro-propagation de l'erreur)

Ce projet consiste donc à implémenter ces différentes techniques. Il vous est loisible de choisir le langage et le paradigme de programmation de votre choix. Il vous est néanmoins interdit d'utiliser une librairie dédiée aux réseaux de neurones. Vous devez donc tout implémenter « from scratch ». Vous pouvez parcontre utiliser une librairie pour afficher les graphiques de vos résultats.

Les données d'apprentissage devront être lues dans un fichier texte du type csv. Vous devez donc également réaliser l'implémentation de cette partie.

Mise au point

En vue de sa mise au point, vous pourrez entrainer votre Perceptron sur des données simples (comme la porte logique ET).

Validation

Une fois mis au point, on vous demande de tester votre implémentation sur les exemples du cours théorique :

Perceptron :

- Opérateur logique ET (table 2.1 ou table 2.3) → mise au point
- Classification de données linéaires séparables (table 2.9)
- Classification de données non linéairement séparables (table 2.10)
- Régression linéaire (table 2.11)

Perceptron monocouche :

- Classification à 3 classes (table 3.1)
- Classification à 4 classes (table 3.5)

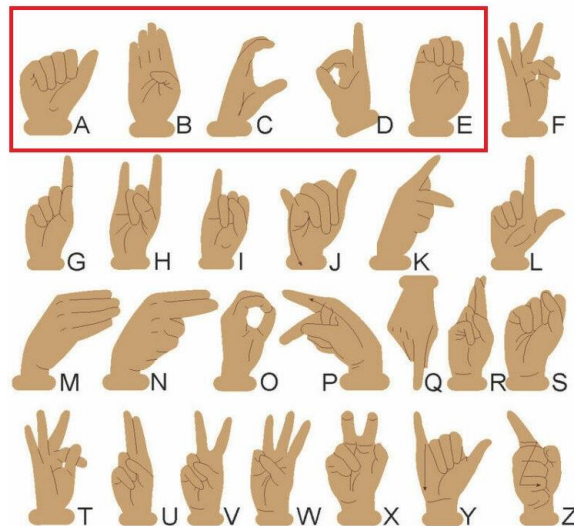
Perceptron multicouche :

- Opérateur logique XOR (table 4.3)
- Classification à 2 classes non linéairement séparables (table 4.12)
- Classification à 3 classes non linéairement séparables (table 4.14)
- Régression non-linéaire (table 4.17)

A chaque fois, on vous demande, lorsque c'est possible, d'afficher graphiquement les résultats de régression et de classification.

Un DataSet réel à traiter : « le langage des signes »

Un peu de culture générale :



Dans le répertoire « LangageDesSignes » fournis, vous trouverez :

- Un répertoire « **pictures** » contenant 300 photos de lettres (A, B, C, D, E) réalisées avec le langage des signes, 60 photos différentes par signes.
- Un fichier « **data.csv** » contenant 300 lignes, une pour chaque photo : la 1^{ère} colonne correspond à la classe (1 pour A, 2 pour B, ..., 5 pour E), suivie de 42 colonnes de nombres compris entre -1 et +1. Il s'agit des coordonnées normalisées de 21 points de coordonnées (x,y) obtenus par une librairie permettant d'extraire ces coordonnées au départ des images fournies → donc le pré-traitement des données « image → 21 points (x,y) » a déjà été réalisé pour vous → les images ne vous sont donc fournies que pour information
- Un fichier « **data_formatted.csv** » dans lequel les données du fichier précédent ont été formatées pour être adaptées à l'entrée de votre réseau de neurones.

On vous demande donc de concevoir un réseau de neurones capable de faire un apprentissage correct de ces données (classification à 5 classes). Pour ce faire :

- créez un « **learning dataset** » composé de 250 images (50 images par signe)
- créez un « **validation dataset** » composé de 50 images (10 images par signe)
- réalisez l'apprentissage sur le learning dataset
- validez cet apprentissage sur le validation dataset.

Consignes du projet

Ce projet est à réaliser par équipe de 3 étudiants.

Pour ce projet, vous devez me rendre votre projet complet (sous forme d'une archive) que vous m'enverrez par e-mail (jean-marc.wagner@hepl.be) à la date convenue.

Une évaluation orale, avec présentation de vos résultats, sera alors réalisée. Vous devrez être capable d'expliquer vos implémentations et les comportements/résultats obtenus.