

# IHDCM037 - Machine Learning

## Travaux pratiques

### Séance 5 : Multilayer Perceptrons

**Remarque :** Pour réaliser les travaux pratiques de ce cours, nous travaillerons avec le langage Python sur des Jupyter Notebook. Ceux-ci peuvent être utilisés en ouvrant la plateforme Anaconda Navigator et en lançant Jupyter Notebook.

## Première partie

La première partie de cette séance est dédiée à la compréhension et l'implémentation d'un perceptron à deux couches. Voici le pseudo-code d'un perceptron à deux couches :

```
Initialize all  $v_{ih}$  and  $w_{hj}$  to  $\text{rand}(-0.01, 0.01)$ 
Repeat
  For all  $(\mathbf{x}^t, r^t) \in \mathcal{X}$  in random order
    For  $h = 1, \dots, H$ 
       $z_h \leftarrow \text{sigmoid}(\mathbf{w}_h^T \mathbf{x}^t)$ 
    For  $i = 1, \dots, K$ 
       $y_i = \mathbf{v}_i^T \mathbf{z}$ 
    For  $i = 1, \dots, K$ 
       $\Delta \mathbf{v}_i = \eta(r_i^t - y_i^t) \mathbf{z}$ 
    For  $h = 1, \dots, H$ 
       $\Delta \mathbf{w}_h = \eta(\sum_i (r_i^t - y_i^t) v_{ih}) z_h (1 - z_h) \mathbf{x}^t$ 
    For  $i = 1, \dots, K$ 
       $\mathbf{v}_i \leftarrow \mathbf{v}_i + \Delta \mathbf{v}_i$ 
    For  $h = 1, \dots, H$ 
       $\mathbf{w}_h \leftarrow \mathbf{w}_h + \Delta \mathbf{w}_h$ 
Until convergence
```

Dans ce pseudo-code,

- $t = 1, \dots, T$  représente le nombre d'instances  $x_t$  et de targets  $r_t$  du dataset,
- $h = 1, \dots, H$  représente le nombre d'hidden units  $z_h$  dans la couche cachée du perceptron, et
- $k = 1, \dots, K$  représente le nombre d'outputs  $y_k$ .

Pour n'importe quel jeu de données manipulé lors des séances précédentes, implémentez un perceptron à deux couches, entraînez le sur des données d'entraînement et calculez l'erreur moyenne quadratique ou l'*accuracy* sur des données de test. Pour vous guider, un *template* du code est disponible sur Webcampus (vous n'êtes évidemment pas obligés de l'utiliser, il y a énormément de façons de coder le même algorithme).

## Deuxième partie

Lors de la première partie, vous avez choisi le nombre d'époques et le nombre d'unités cachées dans la couche cachée. Faites varier ces paramètres et vérifiez que vous obtenez bien les comportements vu dans le cours théorique pour l'erreur/l'*accuracy* évaluée sur les données d'entraînement et de test.