

Département	Informatique	Année	5A INFO
Matière	ML Big Data		
Enseignant	Khalid Benabdeslem		
Intitulé TP :	TP1 Réseaux de neurones et Bagging avec Python		
Contenu	<ul style="list-style-type: none"> • Réseaux de neurones • Bagging 		

0. Préambule

Dans ce TP, vous allez expérimenter les notions vues en cours sur l'apprentissage supervisé par les réseaux de neurones, avec le langage **Python**.

Il faudra rendre votre notebook python sur Moodle, sous le nom **TP1_NN_Nom1_Nom2.ipynb**.

Pour plus de détails concernant :

- le langage Python vous pouvez aller sur le site suivant : <http://www.python-course.eu/index.php>
- la librairie Scikit-learn vous pouvez aller sur le site suivant : <http://scikit-learn.org>

Pour lancer le notebook Python, il faut taper la commande **jupyter notebook** dans votre terminal. Une fenêtre va se lancer dans votre navigateur pour ouvrir l'application **Jupyter**. Créer un nouveau notebook Python et taper le code suivant dans une nouvelle cellule :

```
import numpy as np
np.set_printoptions(threshold=10000, suppress=True)
import pandas as pd
import warnings
import matplotlib
import matplotlib.pyplot as plt
warnings.filterwarnings('ignore')
```

1. Chargement de la base de données « Iris.txt »

Ce jeu de données comprend 50 échantillons de chacune des trois espèces d'iris (*Iris setosa*, *Iris virginica* et *Iris versicolor*). Quatre caractéristiques ont été mesurées à partir de chaque échantillon : la longueur et la largeur des sépales et des pétales, en centimètres.



Setosa

Virginica

Versicolor

→ Charger la base de données de telle sorte à mettre tous les individus avec les 4 premières variables dans **X** et la dernière colonne (qui correspond à la cible) dans **y**.

2. Découpage de la base en Apprentissage/Test

→ Écrire un programme pour découper votre base de données **X** avec un échantillonnage stratifié par rapport aux classes de **y** en deux sous-ensembles d'apprentissage **A** et de test **T** de tailles respectivement 2/3 et 1/3.

3. Implémentation d'un Perceptron Multi-classe

→ Écrire un programme implémentant un **Perceptron multi-classes** sur la base d'apprentissage **A**.

4. Évaluation des performances du modèle

→ Établir les prédictions sur les données de la base de test **T**.

→ Écrire un programme pour calculer la matrice de confusion, l'Accuracy globale, la précision et le rappel pour chaque classe.

5. Implémentation d'un Perceptron Multi-couches (MLP)

→ Appliquer la fonction **MLPClassifier de Scikit-learn** sur la base d'apprentissage **A** avec une couche cachée de 3 neurones (vous avez une liberté totale sur les paramètres du modèle à chercher à explorer dans la fonction).

6. Évaluation des performances du modèle MLP

→ Établir les prédictions sur les données de la base de test **T**.

→ Écrire un programme pour calculer la matrice de confusion, l'Accuracy globale, la précision et le rappel pour chaque classe.

→ Les réseaux de neurones apprennent mieux si les données sont préalablement **normalisées**, c'est à dire si on a pris soin que la variance des valeurs soit la même pour tous les descripteurs. Il faut bien sûr veiller à ce que cette même normalisation soit appliquée aux données de test.

→ Une comparaison entre normalisation/sans normalisation serait la bienvenue !

→ Ré-appliquer toute la chaîne de traitement sur les bases : (glass.txt, breast-cancer-wisconsin.txt, Lsun.txt et Wave) disponibles sur Moodle.

7. Bagging de réseaux de neurones

→ Écrire un nouveau programme permettant de :

- a) créer **K** échantillons bootstrap sur la base d'apprentissage **A** (**K** étant donné par l'utilisateur).
- b) créer un classifieur **MLP** sur chaque échantillon bootstrap.
- c) créer le modèle agrégé **H** à partir des **K** classificateurs appris dans b)
- d) évaluer les performances du modèle **H** sur la base de test **T**

Bon TP !