

Le TP se fait sous Octave, dont la syntaxe est similaire à celle de Matlab.

1 Classification binaire

1. Téléchargez les données d'apprentissage et de test (`optdigits.tra` et `optdigits.tes`) depuis l'url suivante :
<http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits>
2. A l'aide de la fonction `imagesc`, représentez le premier chiffre manuscrit de l'ensemble d'apprentissage.
3. Ecrire une fonction `function [w]=perceptron_learn(X, Y, T)`, qui apprend un classifieur binaire w grâce à l'algorithme du Perceptron vu en cours. Cette fonction prend comme paramètres
 - une matrice X qui contient les exemples d'apprentissage (1 par ligne de la matrice)
 - les étiquettes binaires Y correspondantes
 - et un nombre T d'itérations.
4. Ecrire une fonction `function [labels]=perceptron_predict(X, w)` qui calcule les étiquettes prédites par le classifieur w pour les données des vecteurs d'exemples X .
5. Ecrivez un programme capable de distinguer les chiffres 0,1,2,3,4 des chiffres 5,6,7,8 9.
6. Évaluez les performances des classifieurs appris sur les données de test pour différentes valeurs de T utilisées lors de l'apprentissage.

2 Classification multiclasse

1. Ecrivez un programme capable de distinguer chaque chiffre manuscrit en vous aidant de l'un des algorithmes de Perceptron multiclasse vus en cours.
2. Évaluez les performances des classifieurs appris sur les données de test pour différentes valeurs de T utilisées lors de l'apprentissage.