|  |
| --- |
| ENSAH |
| Compte rendu des exercices |
| GI 2 2017/2018 |

|  |
| --- |
| Mohamed El Ghaouth  21/10/2018 |

Table de matières

* Introduction
* Solution
* Exercice 1 du TP2
* TP modèles linaires en ces paramètres
* conclusion

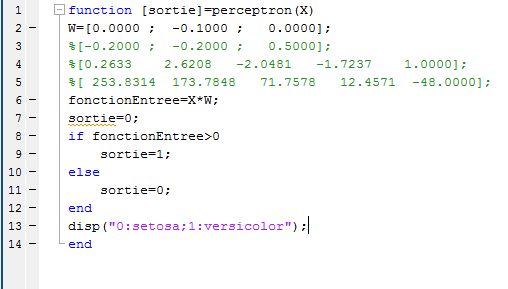
Introduction

Dans l’apprentissage statique plusieurs méthodes et concept peuvent utiliser pour la résolution de problèmes rencontrer .dans ce compte rendu nous allons présenter les solutions de l’exercice 1 du Td3 et le TP du chapitre 2.

Solution

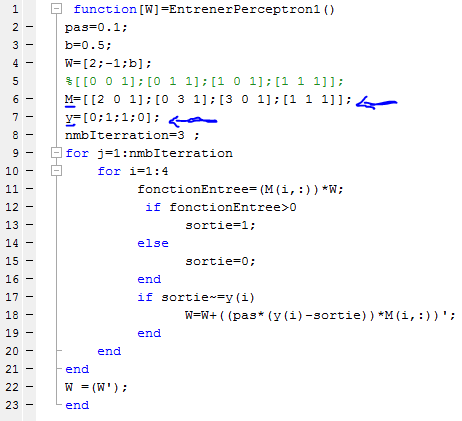
1. Exercice 1 du TP2

* nous avons commencé par une fonction [S]=perceptron(X), qui est baser sur une fonction d’activation de type seuil dure.
* X : la donnée a classifiée
* S: la sortie donnée par notre perceptron



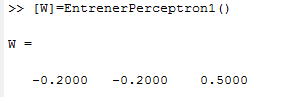
* Puis la fonction [W]=EntrenerPerceptron1()
* W : un vecteur contenant les poids de connexion de notre perceptron après l’entrainement.

Dans le corps de cette fonction nous utilisons une matrice M contenant les données d’entrainement et une matrice Y contenant le résultat entendu pour chaque donnée d’entrainement ; et on commence l‘entrainement par l’initialisation du vecteur W au valeur donné dans l’exercice.

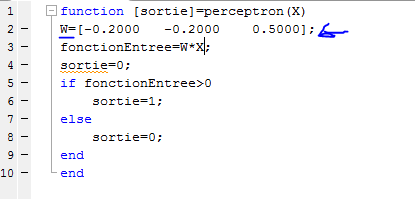


Exemple

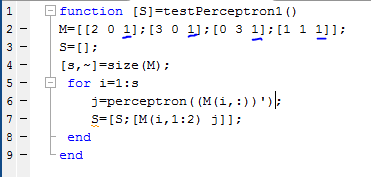
-nous commençons par entrainer notre perceptron sur la classification des données Iris



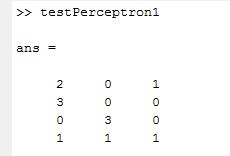
-puis on affecte les poids donnés aux connexions de nôtre perceptron pour qu’il classifie les données.



-puis on teste sur les données en utilisant une fonction qui retourne une matrice représentant le tableau par l’exercice.

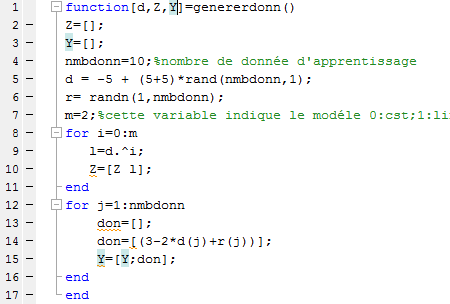


Le nombre 1 ajouter correspond au coefficient du biais.

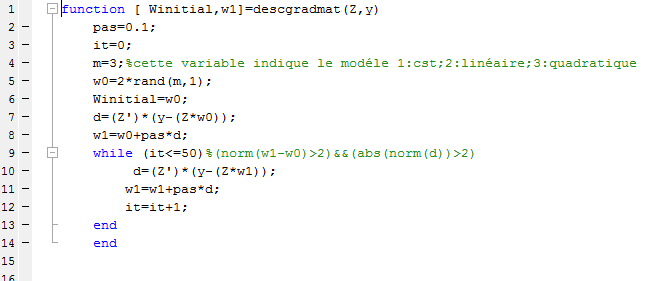


2. TP modèles linaires en ces paramètres

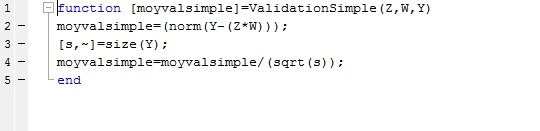
* nous avons commencé par une fonction [d,Z,Y]=genererdonn(), qui génère les données.
* d : les données générer aléatoirement
* Z: la matrice d’observation
* Y :les sorties entendus pour ces données



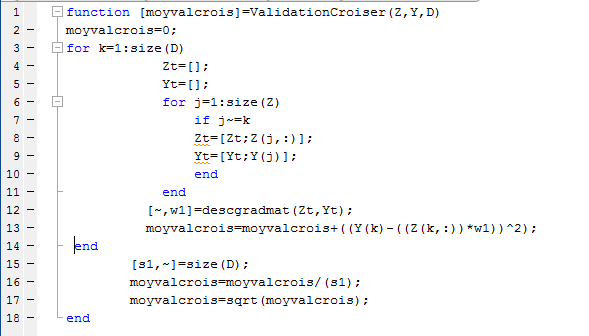
* Puis la fonction [Winitial,W]=descgradmat(Z,Y)
* W : un vecteur contenant les paramètres convenable.
* Winitial :contenant les avec lesquelles on a commencé.
* Z:la matrice d’observation
* Y : les sorties entendus pour ces données



* Puis la fonction [moyvalsimple]=ValidationSimple(Z,W,Y)
* W : un vecteur contenant les paramètres convenable
* Z:la matrice d’observation
* Y : les sorties entendus pour ces données
* moyvalsimple : la moyenne de la validation simple.

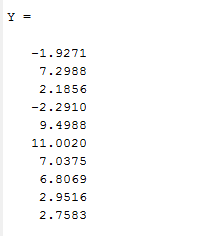
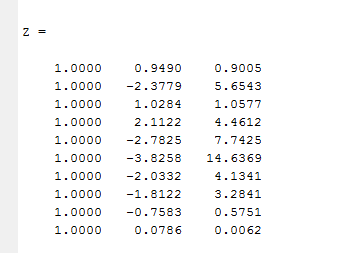
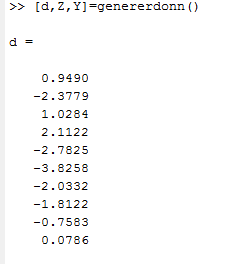


* Puis la fonction [moyvalcrois]=ValidationSimple(Z,Y,D)
* D : un vecteur contenant les données générer aléatoirement.
* Z:la matrice d’observation
* Y : les sorties entendus pour ces données
* moyvalcrois : la moyenne de la validation simple.

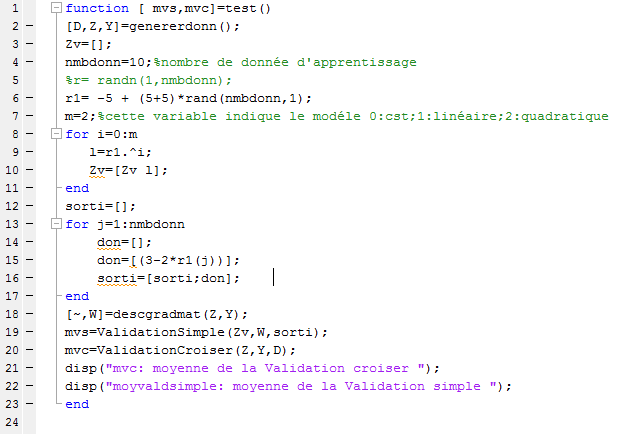


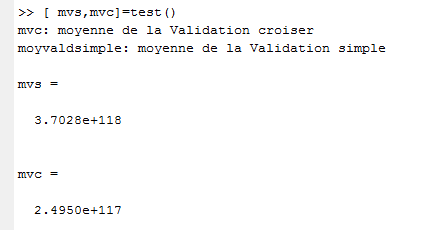
Exemple

-nous commençons par générer les données



-puis on teste tous nos fonction aves une fonction qui utilisera les fonctions précédentes et retournera la moyenne de la validation simple et la moyenne de la validation croiser.





Conclusion

Nous avons donc vue dans ce TP que le Perceptron peut résoudre les problèmes de classification avec une haute précision ; et que en utilisant la méthode descente de gradient et les méthodes de validations nous pouvons reprocher le comportement de n’importe quelle modèle.