|  |
| --- |
| ENSAH |
| Rapport de TP Perceptron |
| GI 2 2017/2018 |

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

Table de matières

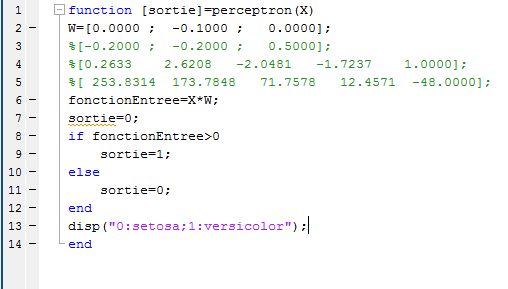
* Introduction
* Solution
* conclusion

Introduction

Dans l’apprentissage statique plusieurs méthodes peuvent utiliser pour réaliser un classificateur de données .dans ce TP nous allons utiliser un Perceptron pour réaliser un classificateur en utilisant Iris Data comme données de comparaison et données de test.

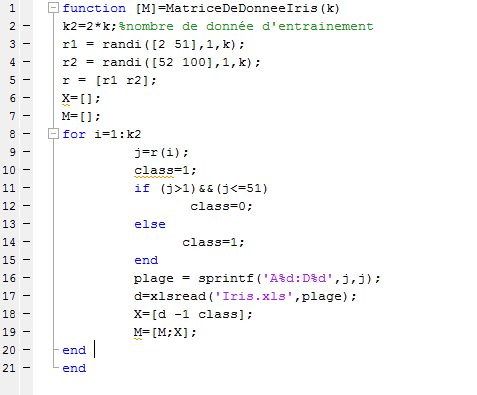
Solution

* 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

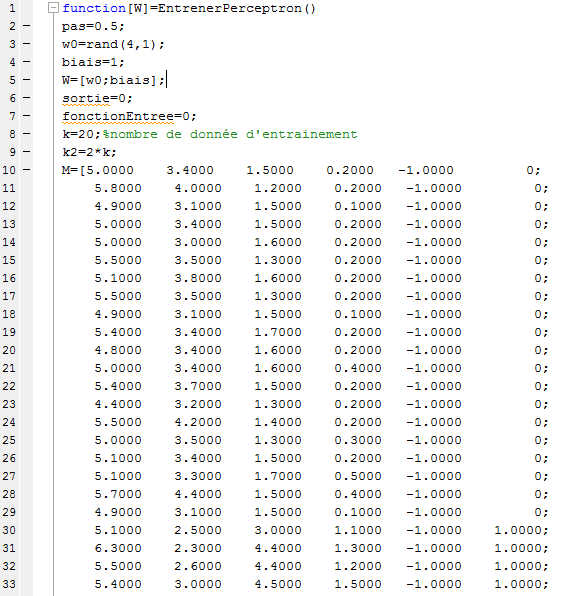


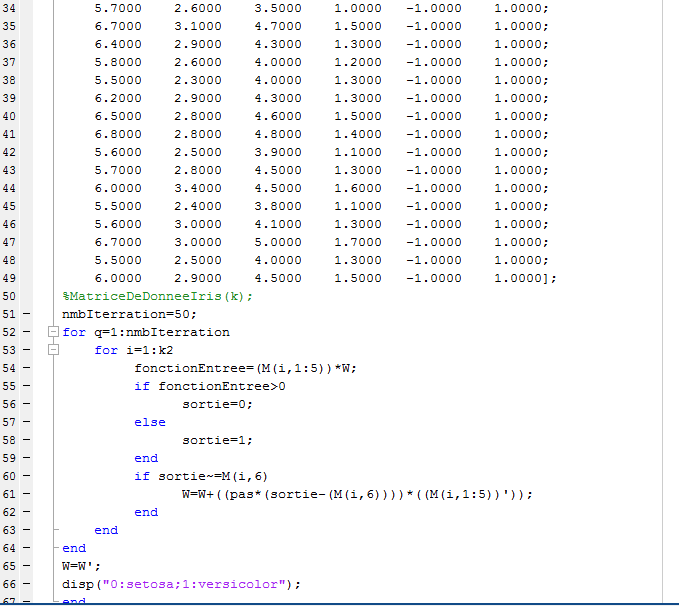
* La fonction [M]=MatriceDeDonneeIris(k)
* K : nombre de données d’entrainement de chaque type
* M : une matrice contenant nos données d’entrainement

Cette fonction va nous permettre de récupérer les données d’entrainement une fois pour tout ce qui démunira le temps d’exécution de notre programme.



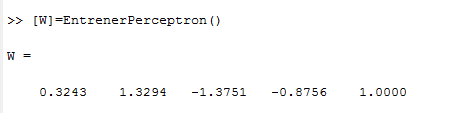
* Puis la fonction [W]=EntrenerPerceptron()
* W : un vecteur contenant les poids de connexion de notre perceptron après l’entrainement.



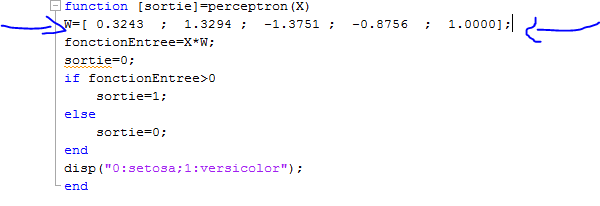


Exemples

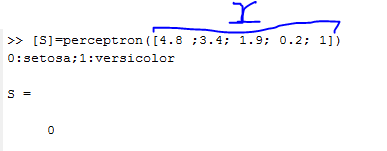
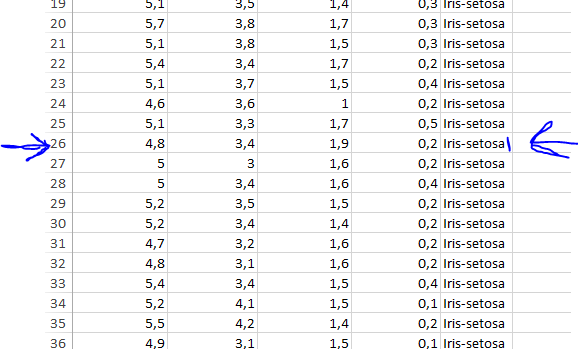
-nous commençons par entrainer notre perceptron sur la classification des données Iris



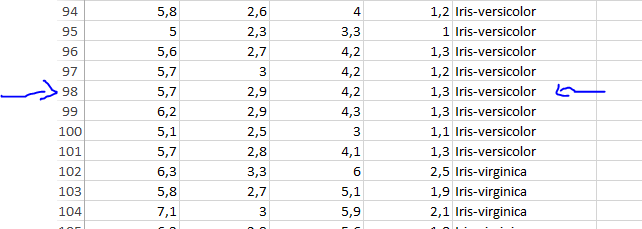
-puis on affecte les poids donnés aux connexions de nôtre perceptron pour qu’il classifie les données.

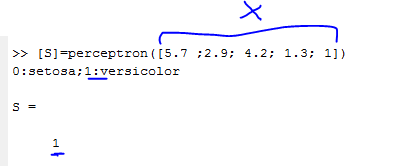


-puis on teste sur les données :



Le nombre 1 ajouter correspond au coefficient du biais.





Le nombre 1 ajouter correspond au coefficient du biais.

Conclusion

Nous avons donc vue dans ce TP que le Perceptron peut résoudre les problèmes de classification avec une haute précision et qu’il est plus efficace que le modèle KNN.