IUT de Béziers	Travaux pratique	R420
Réseaux et Télécoms		Outils pour l'ingénierie
Université de Montpellier	Intelligence Artificielle	

1. Régression linéaire

On donne (sur moodle) le corps du programme qui traite et affiche les données du TD n°1.

a) Vous devez ajouter le calcul de l'estimateur des moindres carrés dans le cas d'une régression linéaire.
b) Mesurer et commenter l'efficacité de votre apprentissage pour plusieurs mélanges du dataset.

Il n'y a pas assez de points d'apprentissage pour que la mesure de l'efficacité soit vraiment intéressante.

c) Modifier le programme pour générer les données et vérifier une loi classique en IA : si on a peu de données d'apprentissage l'erreur d'apprentissage est faible et l'efficacité médiocre et vice-versa.

On pourra, par exemple fixer, a0 et a1 et générer des couples (age, salaire) en ajoutant une valeur aléatoire au salaire :

```
age=rng.integers(low=20, high=40, size=N)
salaire = 35000 + 250*age + rng.normal(0, 1000, size=N)
```

2. Régression polynomiale

a) Adapter le programme précédent pour faire une régression polynomiale. La librairie Python numpy comporte une fonction polyfit que vous utiliserez.
b) Vérifiez la propriété classique en IA : plus un modèle a de paramètres plus l'erreur d'apprentissage se réduit mais plus l'efficacité se réduit.

Le package Python Scikit-Learn, issu des laboratoires de recherche de l'INRIA, contient la plupart des fonctions utiles pour se former et prototyper en IA. Il comprend également des datasets d'exemples. La documentation est ici : https://scikit-learn.org/stable/user_guide.html

3. Classification par régression logistique

Cet exercice repart de l'exemple du cours sur la classification de l'ambiance ressentie en amphi par les étudiants. La modélisation est celle vue en cours.
Un programme de descente du gradient est fourni sur Moodle.

a) Vérifier la convergence de l'algorithme.
b) Vérifier la sensibilité de l'algorithme au taux d'apprentissage.

On veut estimer l'efficacité du prédicteur. Il faut donc scinder le dataset et définir un indicateur de performance. Dans Scikit-Learn la méthode sklearn.model_selection.train_test_split est faite pour ça.
En général on mesure la qualité d'un classifieur avec une matrice de confusion appelée aussi tableau de contingence. Il s'agit en fait de quatre indicateurs calculés sur les données de test. Pour chaque classe on calcule le nombre de points bien classés et le nombre points mal classés. La matrice de confusion peut être calculée automatiquement à l'aide de la méthode sklearn.metrics.confusion_matrix

c) Modifier le programme pour produire un dataset test et calculer la matrice de confusion sur ce dataset.
d) Vérifier que la qualité de la classification dépend de la convergence du gradient.
e) A partir de la matrice de confusion donner le taux des mesures classées inconfortables à tort.

4. Classification par réseau de neurones

Le package Tensorflow permet de manipuler des réseaux de neurones :
https://www.tensorflow.org/api_docs/python/tf

Le programme donné sur moodle permet de générer un modèle paramétrique à partir d'un réseau de neurones, de charger un dataset (plusieurs dataset sont à télécharger sur moodle), de préparer des données d'apprentissage et de test, d'entraîner le modèle, de mesurer l'efficacité du modèle et de prédire des valeurs.

a) Modifier le programme pour visualiser la segmentation du plan entre les deux classes.
b) Mesurer l'effet de la taille du dataset d'apprentissage sur la qualité de la segmentation
c) Mesurer l'effet de « l'époque » sur la qualité de l'apprentissage et de la prédiction
d) Rajouter des couches au modèle et mesurer l'effet de la complexité du modèle sur la qualité de l'apprentissage et de la prédiction