

Série de Travaux Pratiques N° 3

Partie I :

Objectif de la partie I : Développer la Descente de Gradient sur un modèle de régression linéaire simple from scratch

On va travailler sur un modèle linéaire simple. Du coup, on va utiliser la fonction $f(x) = ax+b$.

1- Quelle la fonction du coût utilisé afin de calculer les erreurs pour le cas de $f(x) = ax+b$?

$$J(a,b) = \frac{1}{2m} \sum_{i=1}^m (ax^{(i)} + b - y^{(i)})^2$$

2- Définir une fonction nommée "compute_error" en python qui prend en argument a, b, X, Y et qui calcule J (a, b) la fonction du coût (ensemble des erreurs).

3- Définir une fonction nommée "calcul_gradient" en python, qui calcule les nouveaux a et b mises à jour en utilisant les dérivées partielles.

$$\frac{\partial J(a,b)}{\partial a} = \frac{1}{m} \sum_{i=1}^m (ax^{(i)} + b - y^{(i)}) \times x^{(i)}$$

$$\frac{\partial J(a,b)}{\partial b} = \frac{1}{m} \sum_{i=1}^m (ax^{(i)} + b - y^{(i)})$$

Répéter en boucle:

$$a = a - \alpha \frac{\partial J(a,b)}{\partial a}$$

$$b = b - \alpha \frac{\partial J(a,b)}{\partial b}$$

4- Définir une fonction nommée "GD_runner" en python, qui permet de calculer les nouveaux a et b pour n itérations. Pour chaque itération, veuillez afficher les éléments suivants :

A- La valeur de la fonction du coût

B- La valeur du coefficient a

C- La valeur du coefficient b

5- Visualiser l'évolution de la fonction du coût J (a, b) au des n itérations.

6- Testez votre algorithme avec d'autres valeurs du "learning rate" et du "nombre d'itérations".

Partie II :

Objectif de la partie II : Développer votre premier programme de Machine Learning.

- 1- Importer les librairies : numpy, matplotlib, sklearn.
- 2- Créer votre dataset en important la fonction `datasets.make_regression` et utilisez la pour générer un problème de régression aléatoire de 100 exemples avec une seule variable.
- 3- Visualiser vos données en utilisant la fonction `plt.scatter(x, y)`.
- 4- Définir votre modèle avec `SGDRegressor()` sur 100 itérations avec un Learning rate de 0.0001.
- 5- Entraîner votre modèle en utilisant la méthode `fit()`.
- 6- Calculer la précision de votre modèle en utilisant la fonction `score` qui calcule le coefficient de détermination entre le modèle et les valeurs `y` de votre Dataset.
- 7- Faire de nouvelles prédictions avec la fonction `predict()` et tracer les résultats avec la fonction `plt.plot()`.
- 8- Refaire le même travail en entraînant votre modèle sur 1000 itérations avec un Learning rate de 0.001.
- 9- Qu'est-ce que vous remarquez ?