

Master Informatique Spécialités EID2

Apprentissage Statistique

TP1 (Question bonus)

Descente de gradient

Descente de gradient pour la régression non-linéaire

Maintenant considérons le modèle de la régression polynomiale :

$$\hat{y}_i = a_0 + a_1x_i^1 + \dots + a_nx_i^n$$

1. Écrivez la fonction de coût $E(a_0..a_n)$ à minimiser.
2. Calculez les dérivées partielles de la fonction $E(a_0..a_n)$ selon les paramètres.
3. Implémentez l'algorithme DG sous Python pour la fonction $E(a_0..a_n)$. La fonction Python doit prendre en entrée n , le degré de la fonction polynomiale, ainsi que les données d'apprentissage X et Y .
4. Générez des données à partir d'une fonction polynomiale de degré et coefficient $(a_0..a_n)$ de votre choix, en ajoutant un bruit gaussien afin d'obtenir un nuage de points.
5. Appliquez l'algorithme implémenté au jeu de données généré. Affichez l'évolution de l'erreur $E(a_{0(min)}...a_{n(min)})$ au cours des itérations. Tester différentes valeurs de η et sélectionnez le meilleur résultat.
6. Les coefficients obtenus par descente de gradient sont-ils similaires à ceux que vous avez choisis ?
7. Essayez avec différentes fonctions de divers degrés n . La complexité de la fonction recherchée (le degré) influence-t-elle sur la qualité du résultat ?
8. Comparez avec les résultats obtenus en utilisant les méthodes de régression implémentées dans *scikit-learn*.