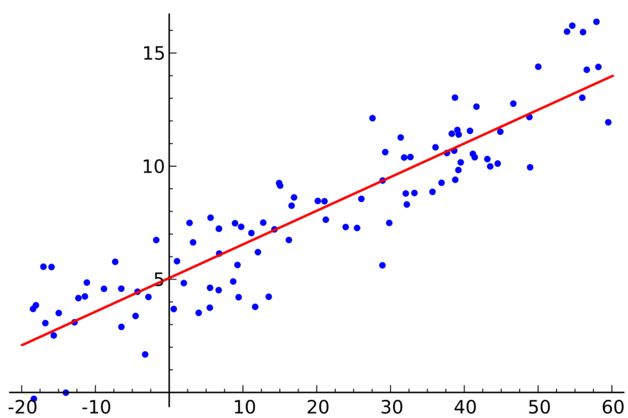
Introduction à l'IA - Descente de gradient

Christophe Rodrigues

29 avril 2019

1 Etude de la descente de gradient sur un cas particulier : la régression linéaire simple

En apprentissage supervisé, la régression linéaire simple consiste à trouver une droite se rapprochant le plus possible des données. la droite est un cas particulier où y la variable de sortie est exprimée en fonction d'une unique variable x. Il s'agit d'une droite car il n'y a qu'une variable en entrée. Dans le cas de la régression linéaire multiple, avec 2 variables en entrée on cherchera un plan, et dans le cas général un hyperplan. Dans ce qui suit on se concentrera sur le cas simple (néanmoins tout est transposable au cas multiple).



Cette droite représente un modèle des données et pour chaque valeur de x, ce modèle peut associer (prédire) une valeur y. Il peut d'ailleurs prédire des valeurs pour des valeurs de x qu'il n'a jamais rencontré (et c'est là que réside le principal intérêt de l'apprentissage: apprendre sur des exemples afin d'être capable de traiter de nouveaux cas jamais rencontrés mais proches des premiers).

Comme nous pouvons le voir sur la figure, notre modèle peut commettre des erreurs. pour une valeur de x donnée, la distance entre la valeur prédite (la droite) et sa valeur réelle (point bleu) représente l'erreur du modèle sur cette valeur. On peut calculer l'erreur totale du modèle en calculant la moyenne des erreurs sur chacun des exemples disponibles. Plus cette erreur est petite, meilleur est le modèle.

Comme déjà abordé nous pouvons utiliser le MSE (Mean Square Error) afin de calculer l'erreur d'un modèle en régression. Si Y_i est la valeur observée pour l'exemple i et $\hat{Y}_i$ la valeur prédite pour ce même exemple, alors pour n exemples l'erreur MSE s'écrit:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Puisqu'une droite s'écrit sous la forme : $y = mx + b$, obtenir le meilleur modèle possible revient alors à trouver les paramètres m et b pour lesquelles l'erreur est minimale. Pour le faire, nous allons utiliser la méthode de la descente de gradient.

1.1 Brute Force

Afin de trouver la meilleure droite, on se propose de calculer un grand ensemble de combinaisons possibles pour les paramètres m et b. Supposons que :

- $-50 < m \leq 50$
- $-10 < b \leq 10$
- précision pour m et b de 10^{-2}

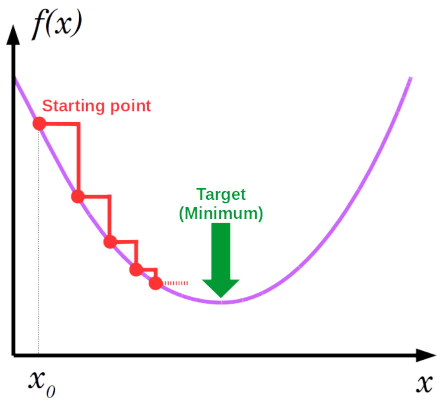
Question : Combien de modèles différents sont possibles?

Question : Ecrire une fonction permettant de calculer l'erreur d'un modèle. Cette fonction doit prendre en paramètre m, b et un ensemble d'exemples(X,Y) et retourner l'erreur MSE.

Question : A partir de brightspace, WP6, charger en mémoire le fichier de données data.csv (il est composé de deux colonnes (la variable X suivie de Y). Faire une boucle permettant de trouver le meilleur modèle.

Question : Quelles sont les limites de cette approche?

1.2 Descente de gradient



La descente de gradient permet de trouver le minimum global d'une fonction convexe. Conceptuellement, cela consiste à regarder le sens de la pente et à la suivre de proche en proche jusqu'à ne plus pouvoir (pente nulle), on a alors atteint le minimum.

Dans notre cas, le minimum que nous cherchons est celui de l'erreur du modèle. La méthode consiste alors à calculer les dérivés des paramètres m et b et de suivre leur pente respective vers le bas.

Question : Calculer les dérivés partielles de l'erreur selon m et puis b.

Question : Ecrire une fonction permettant de mettre à jour les paramètres m et b. La fonction prendra en entrée des valeurs nulles pour m et b. Cette fonction retournera les nouveaux paramètres m-maj et b-maj trouvés.

Question : Ecrire une boucle permettant de mettre à jour les paramètres m et b à chaque itération. (Tout l'intérêt étant de ne pas repartir de paramètres aléatoires à partir de la dernière mise à jour)