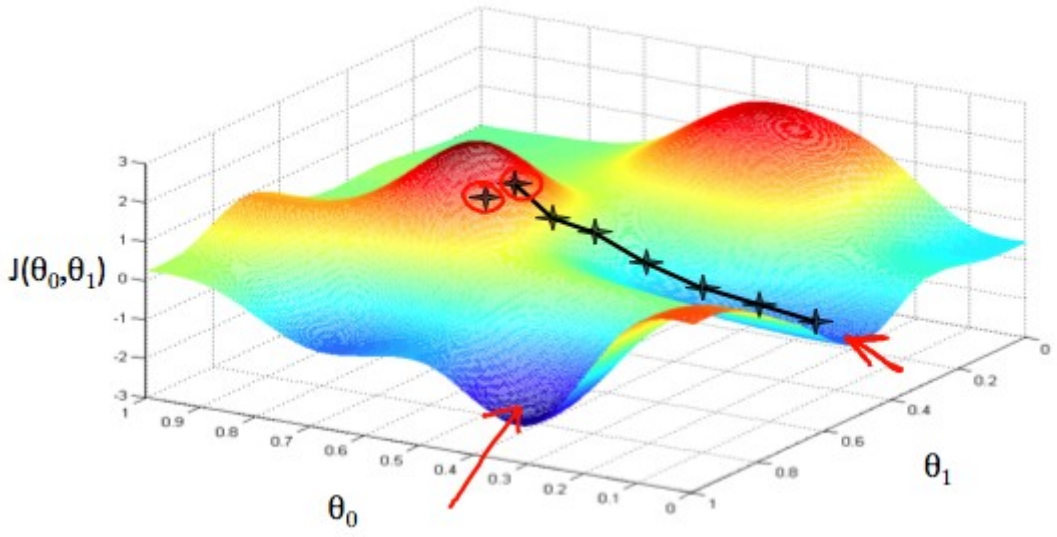
Descente en gradient

Nous avons donc notre fonction d'hypothèse et nous avons un moyen de mesurer son adéquation aux données. Maintenant, nous devons estimer les paramètres de la fonction d'hypothèse. C'est là que la descente de gradient entre en jeu.

Imaginez que nous graphions notre fonction d'hypothèse en fonction de ses champs θ_0 et θ_1 (en fait, nous graphions la fonction de coût en fonction des estimations des paramètres). Nous ne représentons pas x et y en soi, mais la plage de paramètres de notre fonction d'hypothèse et le coût résultant de la sélection d'un ensemble particulier de paramètres.

Nous plaçons \theta_0θ_0 sur l'axe x et \theta_1θ_1 sur l'axe y, avec la fonction de coût sur l'axe z vertical. Les points sur notre graphique seront le résultat de la fonction de coût utilisant notre hypothèse avec ces paramètres thêta spécifiques. Le graphique ci-dessous illustre une telle configuration.



Nous saurons que nous avons réussi lorsque notre fonction de coût se trouve tout en bas des creux de notre graphique, c'est-à-dire lorsque sa valeur est minimale. Les flèches rouges indiquent les points minimums du graphique.

Pour ce faire, nous prenons la dérivée (la ligne tangentielle à une fonction) de notre fonction de coût. La pente de la tangente est la dérivée à ce point et elle nous donne une direction vers laquelle aller. Nous descendons la fonction de coût dans la direction où la pente est la plus forte. La taille de chaque étape est déterminée par le paramètre α, qui est appelé le taux d'apprentissage.

Par exemple, la distance entre chaque "étoile" dans le graphique ci-dessus représente un pas déterminé par notre paramètre **α**. Un **α** plus petit entraînerait un pas plus petit et un α plus grand un pas plus grand. La direction dans laquelle le pas est effectué est déterminée par la dérivée partielle de $J(\theta_0,\theta_1)$. Selon l'endroit où l'on commence sur le graphique, on peut se retrouver à différents points. L'image ci-dessus nous montre deux points de départ différents qui aboutissent à deux endroits différents.

L'algorithme de descente de gradient est :

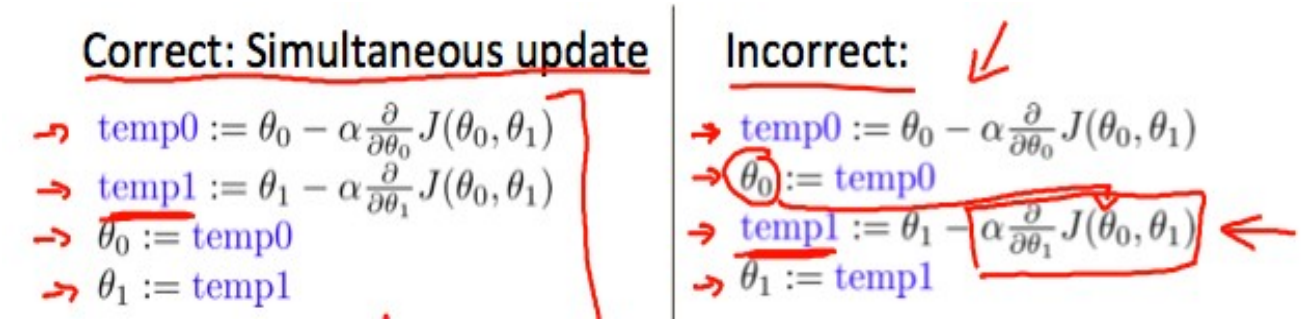
répétez jusqu'à la convergence :

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

où

$j=0,1$ représente le numéro d'index de la caractéristique.

À chaque itération j, il faut mettre à jour simultanément les paramètres $\theta_1,\theta_2,\ldots,\theta_n$. La mise à jour d'un paramètre spécifique avant le calcul d'un autre paramètre à la $j(th)$ itération conduirait à une implémentation erronée.



Intuition de la descente par gradient

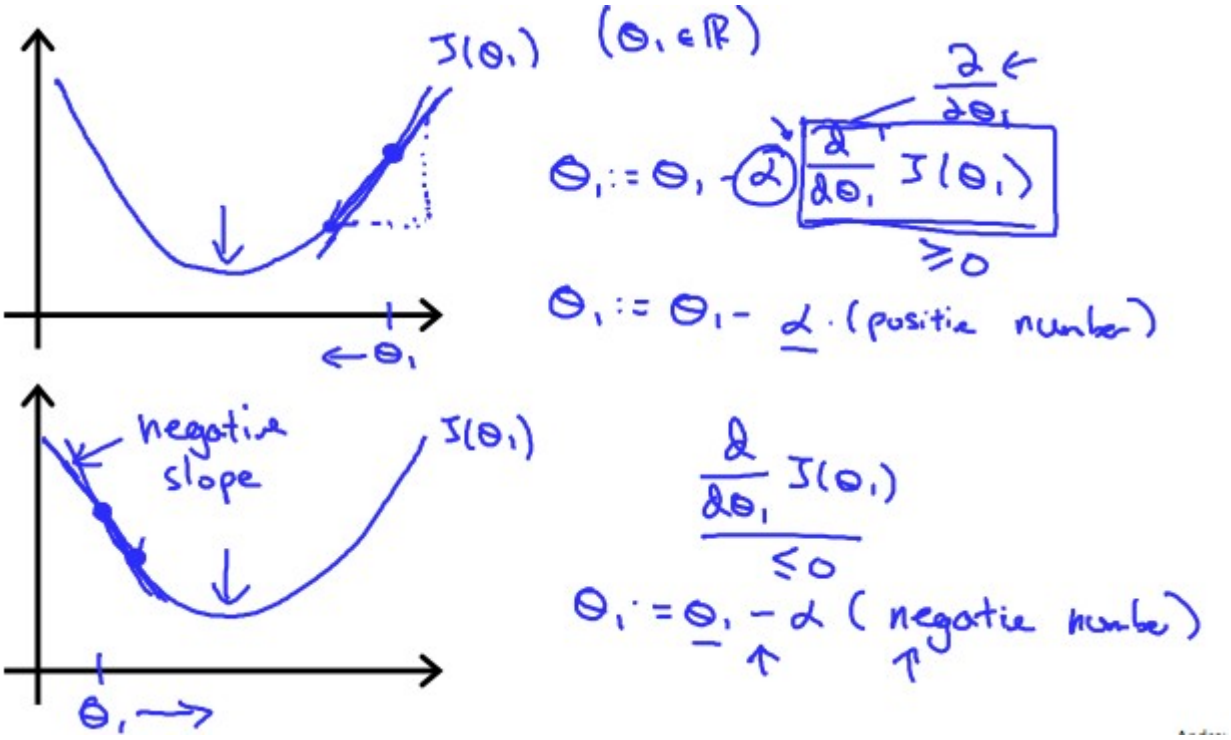
Dans cette vidéo, nous avons exploré le scénario où nous avons utilisé un seul paramètre θ_1 et tracé sa fonction de coût pour mettre en œuvre une descente de gradient. Notre formule pour un seul paramètre était :

Répétez jusqu'à la convergence :

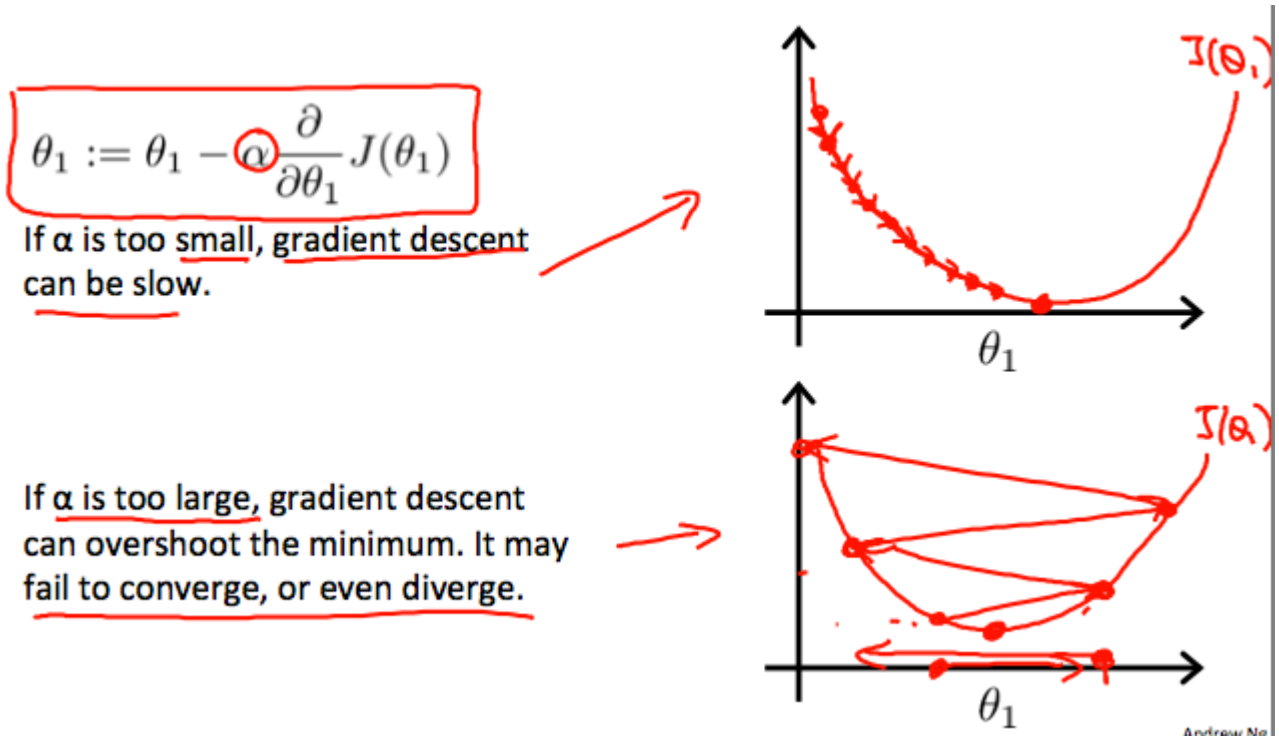
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

Quel que soit le signe de la pente pour $\frac{d}{d\theta_1} J(\theta_1), \theta_1$

finit par converger vers sa valeur minimale. Le graphique suivant montre que lorsque la pente est négative, la valeur de θ_1 augmente et que lorsqu'elle est positive, la valeur de θ_1 diminue.



En passant, nous devrions ajuster notre paramètre **α** pour nous assurer que l'algorithme de descente de gradient converge dans un temps raisonnable. L'absence de convergence ou un temps trop long pour obtenir la valeur minimale impliquent que notre taille de pas est erronée.



Comment la descente de gradient converge-t-elle avec une taille de pas fixe **α** ?

L'intuition derrière la convergence est $\frac{d}{d\theta_1} J(\theta_1)$;

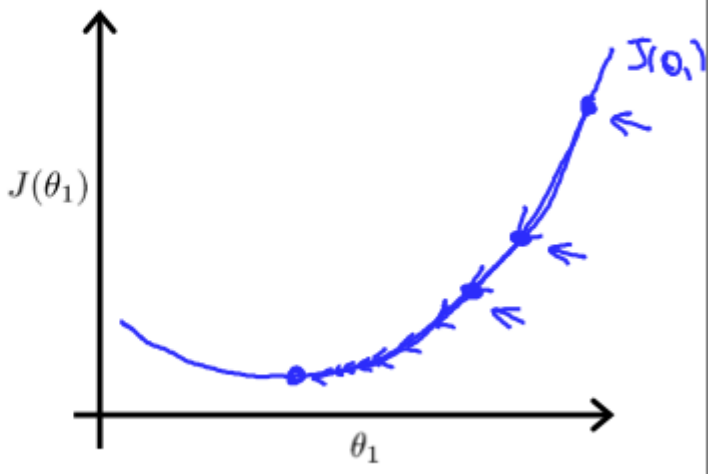
que se rapproche de 0 au fur et à mesure que l'on s'approche du bas de notre fonction convexe. Au minimum, la dérivée sera toujours 0 et donc on obtient :

$\Theta_1 := \theta_1 - \alpha * 0$

Gradient descent can converge to a local minimum, even with the learning rate α fixed.

$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

As we approach a local minimum, gradient descent will automatically take smaller steps. So, no need to decrease α over time.



Andrew Ng

Descente de gradient pour la régression linéaire

Note : *[A 6:15 "h(x) = -900 - 0.1x"* devrait être *"h(x) = 900 - 0.1x"]*

Lorsqu'on l'applique spécifiquement au cas de la régression linéaire, une nouvelle forme de l'équation de descente du gradient peut être dérivée. Nous pouvons substituer notre fonction de coût réelle et notre fonction d'hypothèse réelle et modifier l'équation en :

répéter jusqu'à convergence : {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x_i) - y_i) x_i$$

}

où m est la taille de l'ensemble d'apprentissage, θ_0 une constante qui sera modifiée simultanément avec θ_1 et x_i, y_i *sont les* valeurs de l'ensemble d'apprentissage donné (données).

Notez que nous avons séparé les deux cas pour θ_j en équations distinctes pour θ_0 et θ_1 ; et que pour θ_1 nous multiplions x_i à la fin à cause de la dérivée.

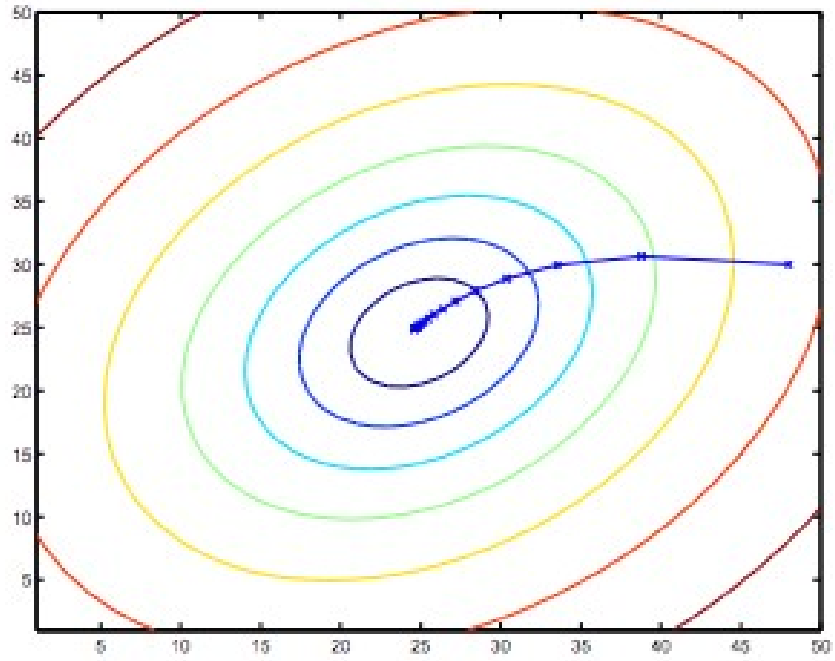
Ce qui suit est une dérivation de $\frac{\partial}{\partial \theta_j} J(\theta)$ pour

un seul exemple :

$$\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2} \left(h_\theta(x) - y \right)^2 \\
&= 2 \cdot \frac{1}{2} \left(h_\theta(x) - y \right) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y) \\
&= \left(h_\theta(x) - y \right) \cdot \frac{\partial}{\partial \theta_j} \left(\sum_{i=0}^{n} \theta_i x_i - y \right) \\
&= \left(h_\theta(x) - y \right) x_j
\end{aligned}$$

L'intérêt de tout cela est que si nous commençons par deviner notre hypothèse et que nous appliquons ensuite de manière répétée ces équations de descente de gradient, notre hypothèse deviendra de plus en plus précise.

Il s'agit donc simplement d'une descente de gradient sur la fonction de coût originale J. Cette méthode examine chaque exemple de l'ensemble d'apprentissage à chaque étape, et est appelée **descente de gradient par lots**. Notez que, bien que la descente de gradient puisse être sensible aux minima locaux en général, le problème d'optimisation que nous avons posé ici pour la régression linéaire n'a qu'un seul optima global, et aucun autre optima local ; ainsi la descente de gradient converge toujours (en supposant que le taux d'apprentissage α n'est pas trop grand) vers le minimum global. En effet, J est une fonction quadratique convexe. Voici un exemple de descente de gradient telle qu'elle est exécutée pour minimiser une fonction quadratique.



Les ellipses représentées ci-dessus sont les contours d'une fonction quadratique. On voit également la trajectoire prise par la descente par gradient, qui a été initialisée à $(48,30)$. Les x de la figure (reliés par des lignes droites) marquent les valeurs successives de θ par lesquelles la descente par gradient est passée pour converger vers son minimum.