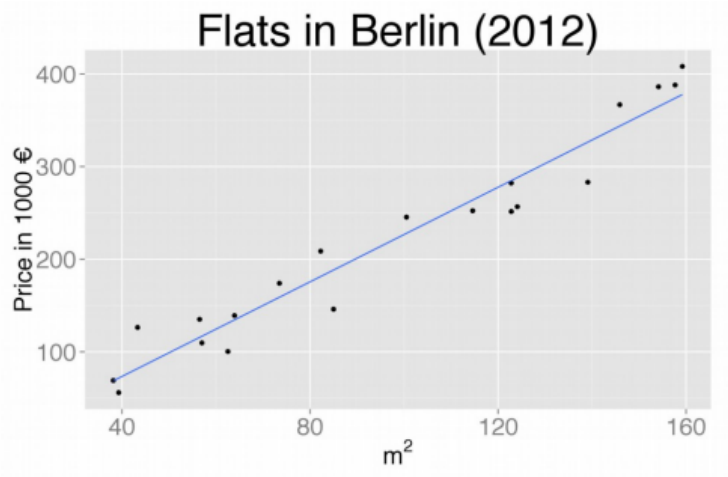
Funcionamento do algoritmo gradient descent

O gradient decent é um algoritmo muito usado em problemas de regressão, pois permite atualizar os parâmetros de uma função a fim de minimizar a diferença entre uma hipótese e um conjunto de treinamento.

Um bom exemplo de uso desse algoritmo é o problema de prever o preço de uma casa com base na área dela. Supondo que queremos queiramos uma curva linear para prever o preço das casas, teremos uma hipótese do tipo $\boldsymbol{y} = b + \boldsymbol{x}w + \boldsymbol{e}$ onde w e b são os parâmetros que queremos prever e **x** é o conjunto de tamanho das casas. O objetivo é encontrar os valores de b e w que nos permita uma curva semelhante a da figura abaixo:

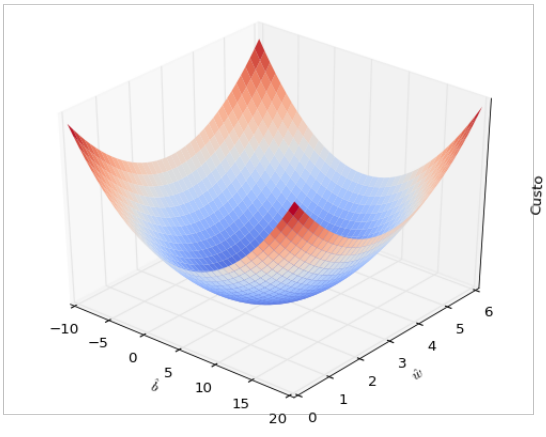


Queremos achar os valores de b e w que minimizam o quadrado da norma do vetor **e**, ou seja, minimiza a soma dos quadrados dos resíduos. Dessa forma garantimos que nossa função custo tenha apenas um mínimo e evitamos de cair em um mínimo local e não global.

Encontrar o menor valor dessa função significa encontrar valores de b e w que nos permita ter uma hipótese mais confiável. Vamos usar o gradient descent para fazer isso.

Entendendo e visualizando o gradient descent

Para melhor entendimento do algoritmo, é bom visualizar nossa função custo quando plotada nas duas dimensões dos parâmetros b e w que queremos aprender:



O gradiente dessa função é simplesmente um vetor de derivadas parciais, que dão a inclinação em cada ponto e em cada direção:

$$\nabla(L) = \left[\frac{\partial L}{\partial b}, \frac{\partial L}{\partial w}\right]$$

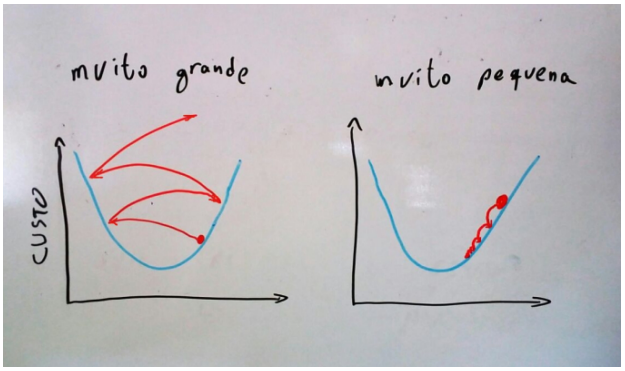
Se nos seguirmos na direção oposta do gradiente, então chegaremos ao ponto de mínimo.

A cada iteração os parâmetros b e w dão um passo em direção ao mínimo, o tamanho desse passo será o valor do gradiente naquele ponto multiplicado pela constante α. Quando mais próximos estamos do ponto mínimo, menor é a inclinação do custo, menor é o gradiente e consequentemente menor é o passo em direção ao mínimo.

Hiper-parâmetros

Diferente dos parâmetros w, que são aprendidos durante o treinamento de uma regressão linear, os hiper-parâmetros não são aprendidos pela máquina durante o treinamento e devem ser ajustados manualmente. Temos três hiper-parâmetros: **A taxa de aprendizado, O número de iterações de treino e os valores iniciais de w.** Como a função custo é convexa, os valores iniciais de w não são muito importantes desde que os outros parâmetros estejam ajustados corretamente.

A **taxa de aprendizado** define o tamanho dos passos que daremos em direção ao mínimo em cada iteração. Se os passos forem pequenos, é muito provável que chegaremos bem próximo do mínimo, porém mais iterações serão necessárias e consequentemente o algoritmo ficará mais lento. Porém, por outro lado, se os passos forem muito grandes corre o risco de nos afastarmos do mínimo, já que do ponto inicial, podemos chegar em um ponto custo mais alto, nesse ponto o gradiente será ainda maior fazendo com que o passo seja ainda maior, dessa forma afastando mais ainda do mínimo.



Com uma boa taxa de aprendizado, é fácil selecionar o **número de iterações de treino.** Pode ocorrer de haver tantas iterações que o custo cai, como iterações que o custo sobe. Se a função custa flutua muito a cada iteração, recomenda-de diminuir a taxa de aprendizado. Se a função custo desce suavemente, mas muito devagar, recomenda-se aumentar a taxa de aprendizado.

Referências

- Slides da aula
- https://matheusfacure.github.io/2017/02/20/MQO-Gradiente-Descendente/
- https://oliveiratiano.wordpress.com/2016/07/31/aprendizado-de-maquina-na-pratica-1-entendendo-o-algoritmo-mestre/