线性回归算法是一种监督学习的算法，其算法的思路是利用一条直线拟合多元的数据，使得所有数据与该直线的偏差和最小。

数据的数量：m

数据的维度：n

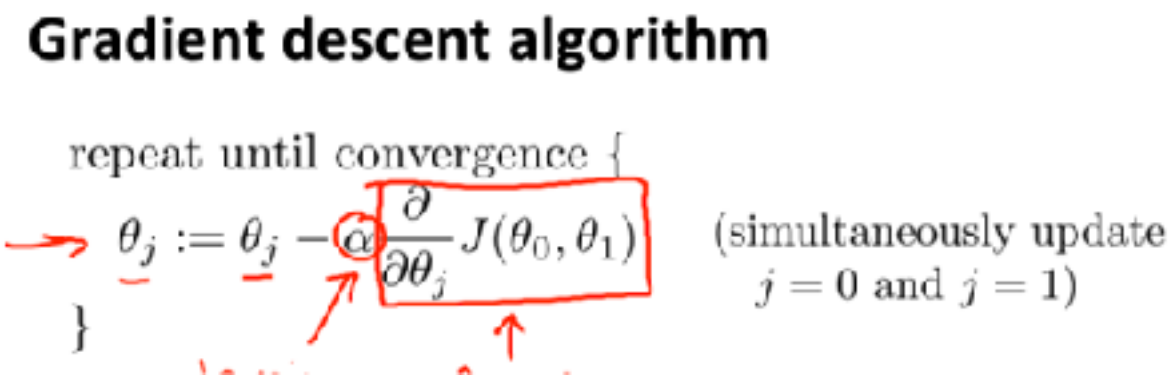
估计函数：$h_\theta(x) = \theta_0 + \theta_1 * x$

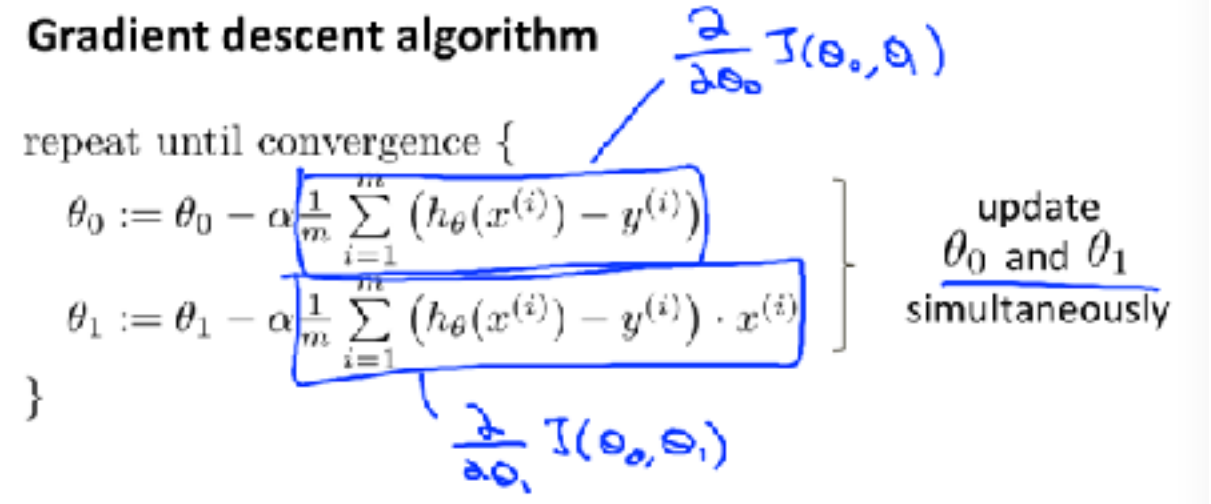
数据已知的结果：y

损失函数设计（以二元为例子）：

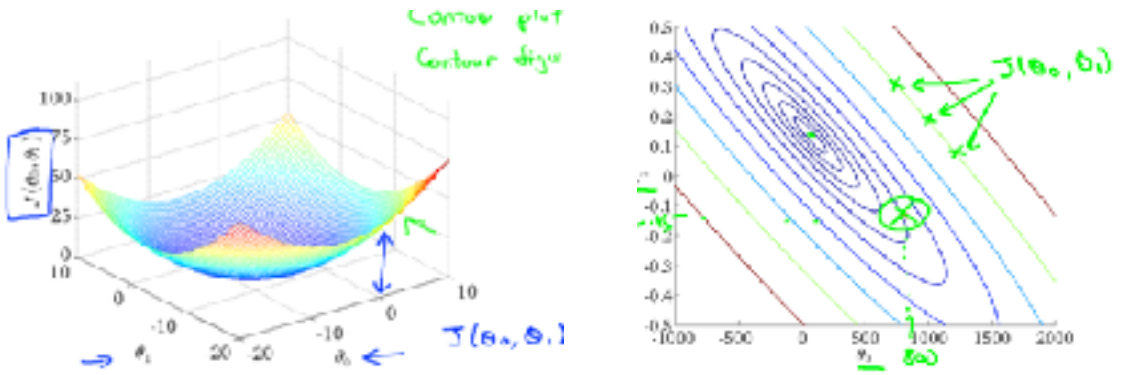
$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

采用梯度下降法求解 θ_0 θ_1参数：



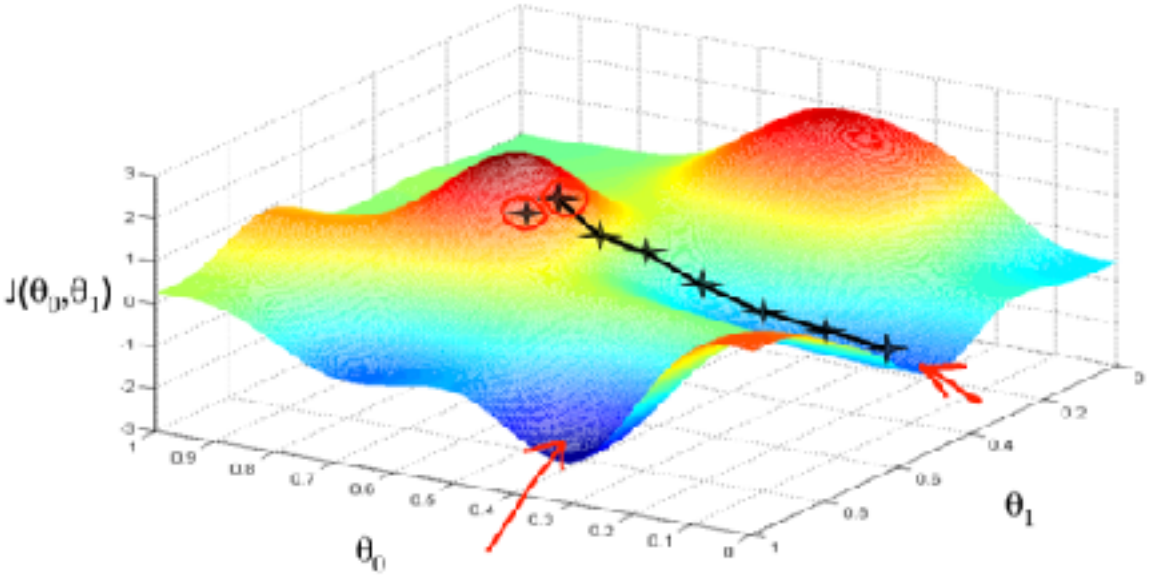


过程展示和总结：



逐次的迭代会调整参数直至损失函数逐渐减小，当整个函数的形状是convex(凸性)的时候，保证函数最终可以到达一个最小值。左图为全貌图，右图为contour plot，contour plot可以看出每一次迭代数据点在损失函数上的升降。

然而，当函数不是convex时，得到的结果未必是全局的最小值，而可能达到局部的极小值。因此，梯度下降法有点贪心算法的味道，每一次都选取损失函数下降最快的方向进行迭代，从而失去了全局最优解。



衍生出来的问题：学习率的设定，学习率如果大，那么收敛速度快，但是收敛到的极限不够准确；学习率如果小，那么收敛速度慢，但是收敛到的极限准确。综上所述，最好学习率先大后小，因此常用指数下降的学习率。