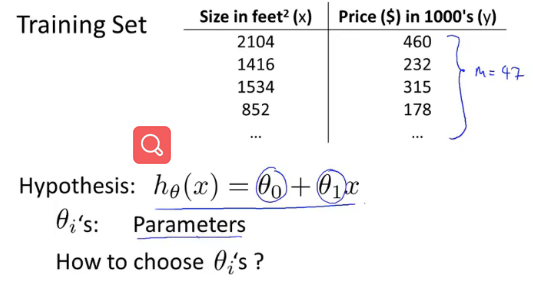
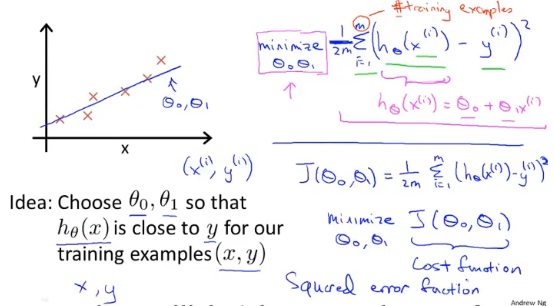
# 线性回归

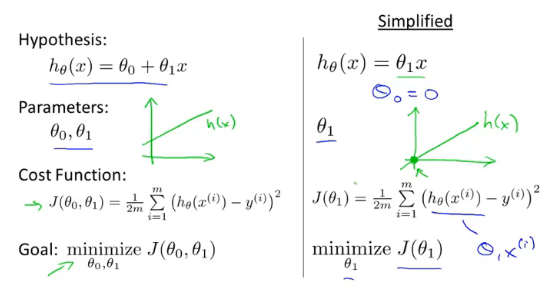


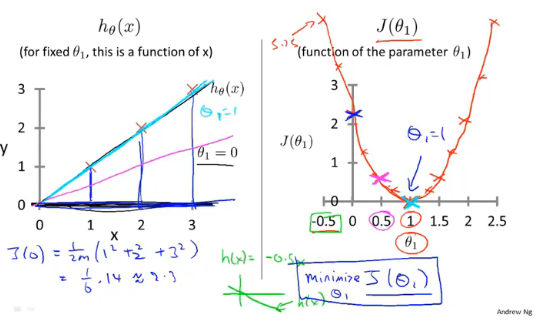
# 代价函数cost function

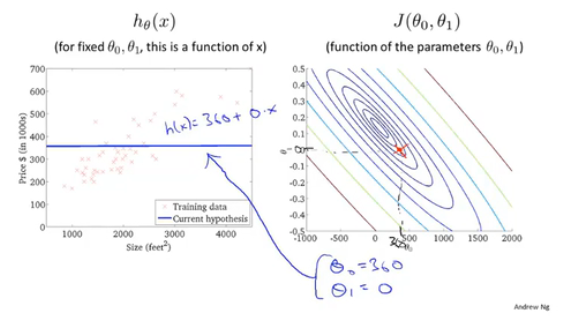
cost function也被称作squared error function



用的是mean squared error（均方误差）

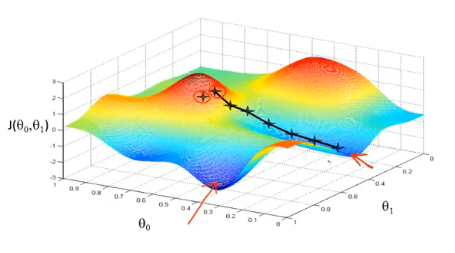




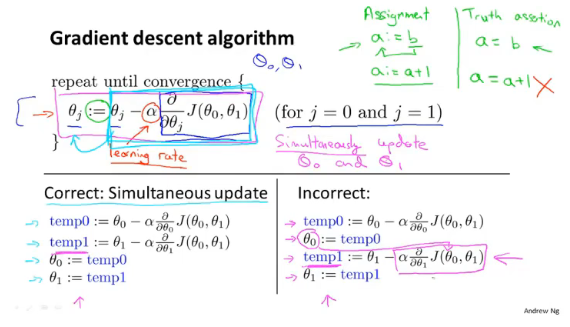


# 梯度下降Gredient descent

用梯度下降算法最小化任意函数J

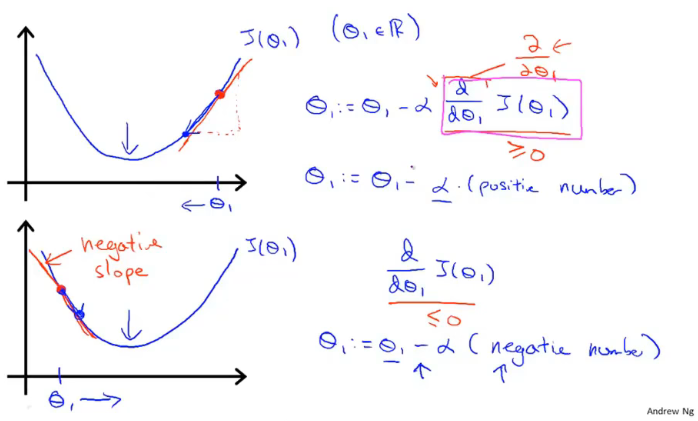


梯度下降算法起始点不同得到的局部最优解也是不同的



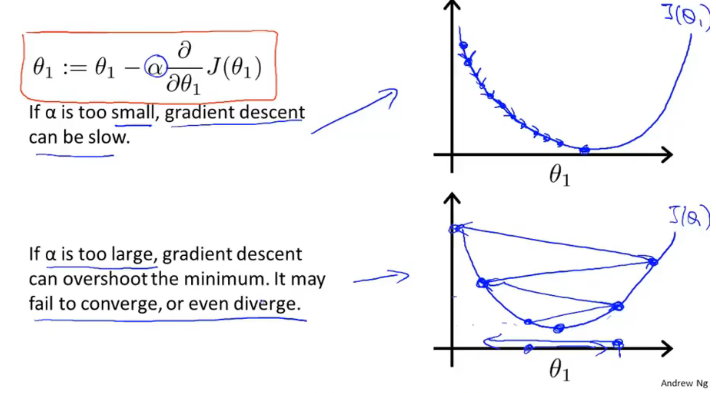
α越大下降的越快，它控制我们以多大幅度更新这个参数θj

θ0和θ1必须同步更新

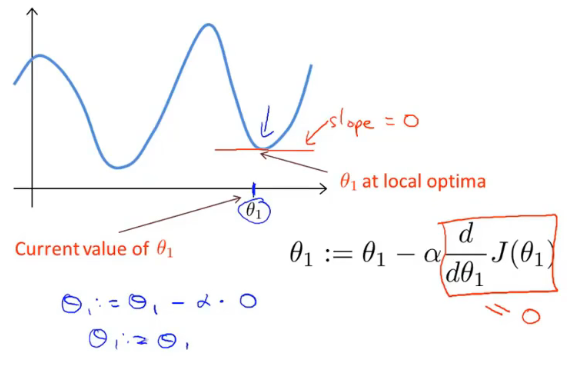


α后面的偏导数相当于一个斜率，减到“谷底”导数为0

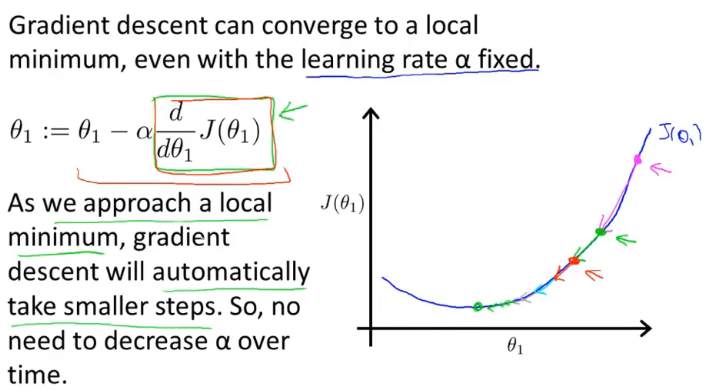
右边是正斜率，向左移动。左边是负斜率，向右移动。



如果α太小，速度会很慢。如果α太大，会直接越过最低点。

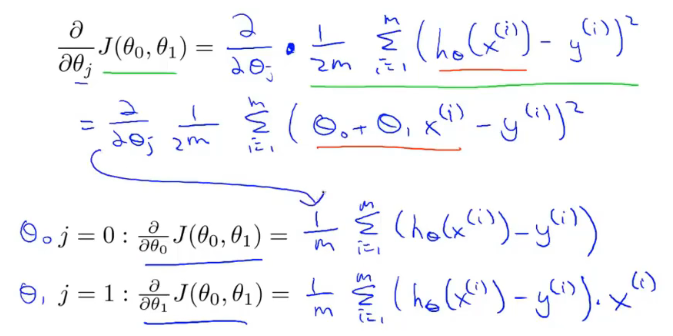


已经到达最低点，斜率为0，θ的值不再变化

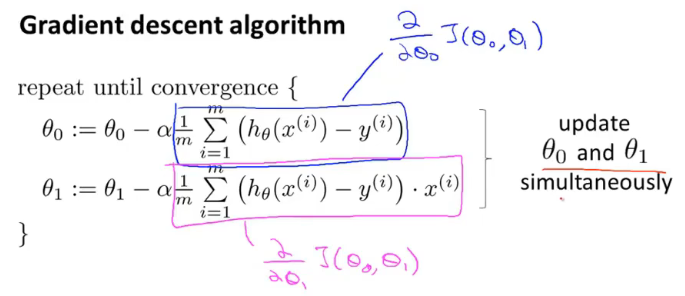


斜率会越变越小，所以跨度会越来越小

# 代价函数和梯度下降的化简



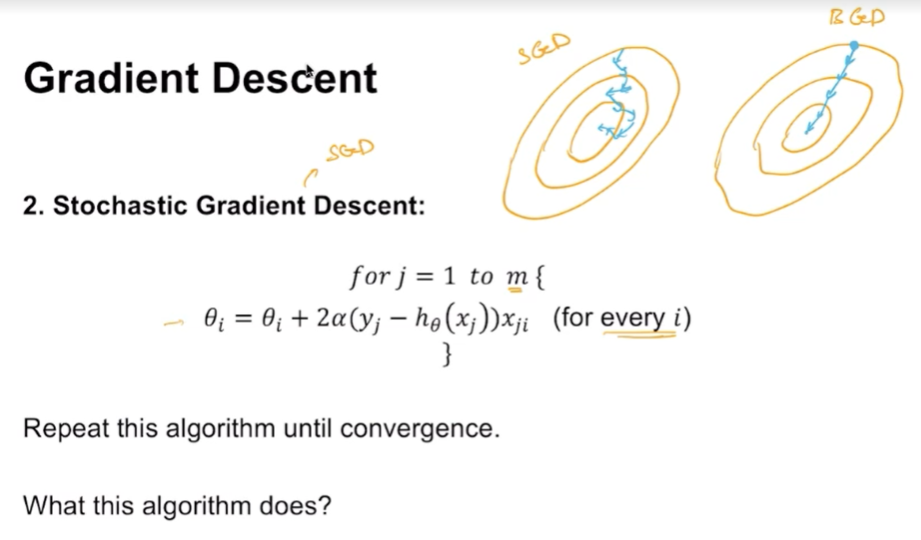
将代价函数带入求导



将导数化简之后

相比较高等数学的方程组解法，梯度下降法更适用于更大的数据集

# 补充



在随机梯度下降法中，分别在任意样本上更新�该算法比批处理梯度下降算法开销小得多，但它可能永远不会收敛到最小值。