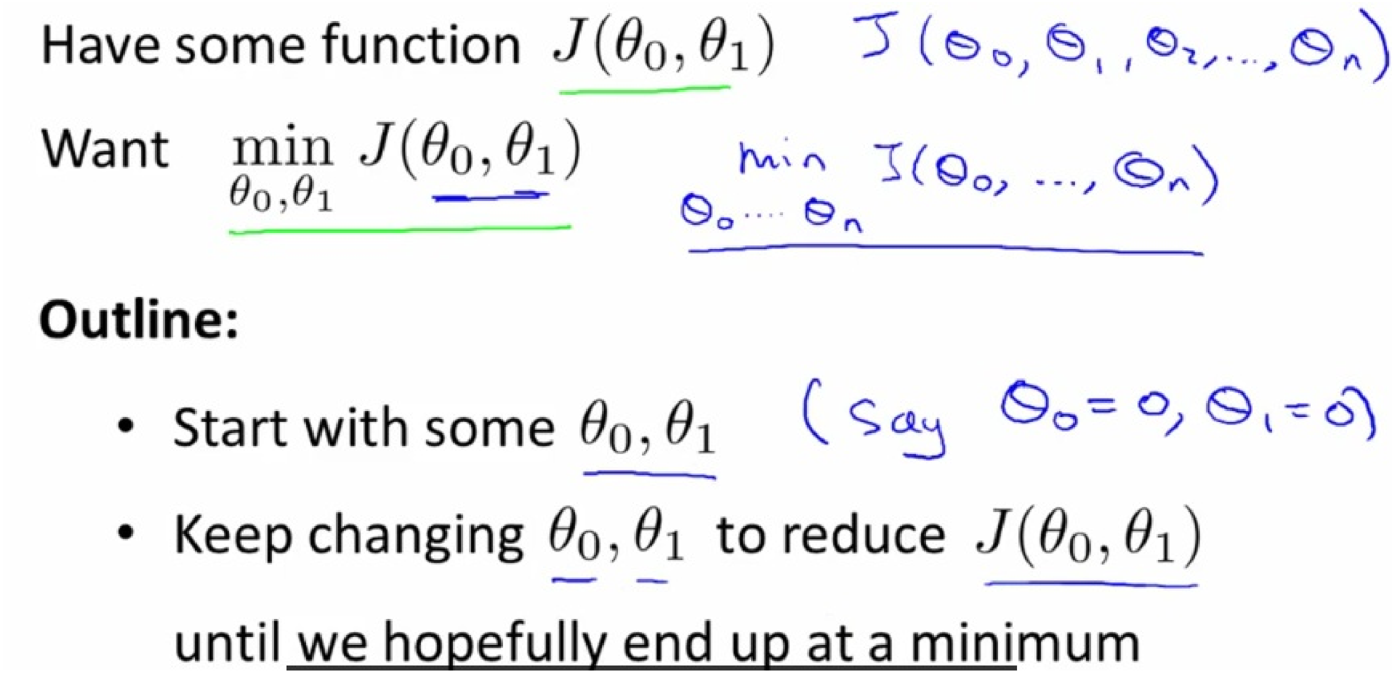
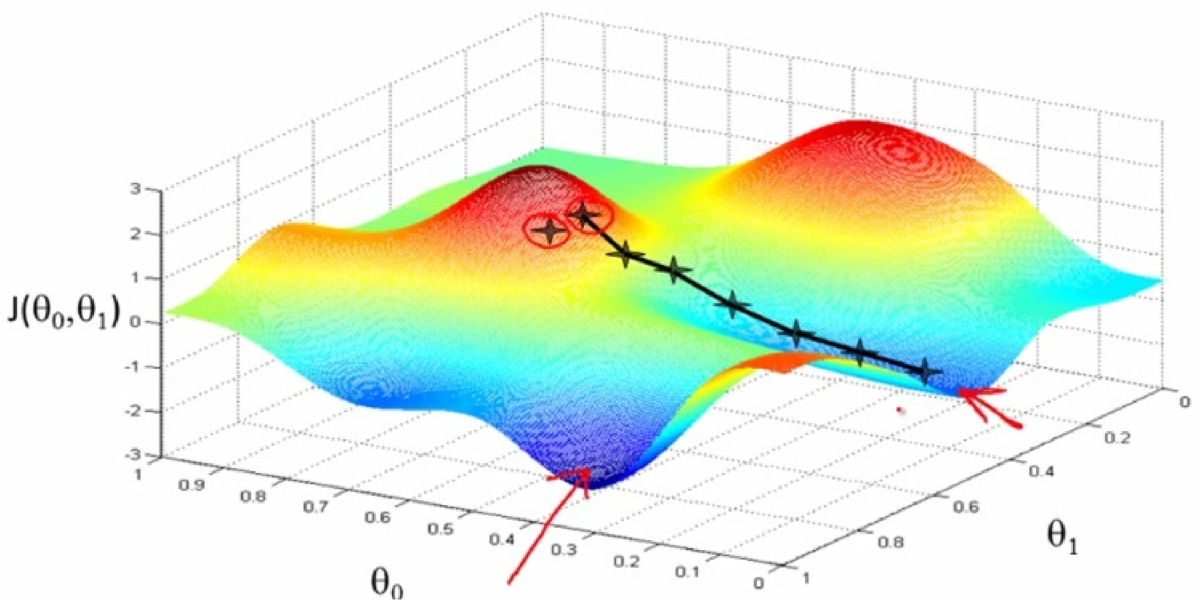
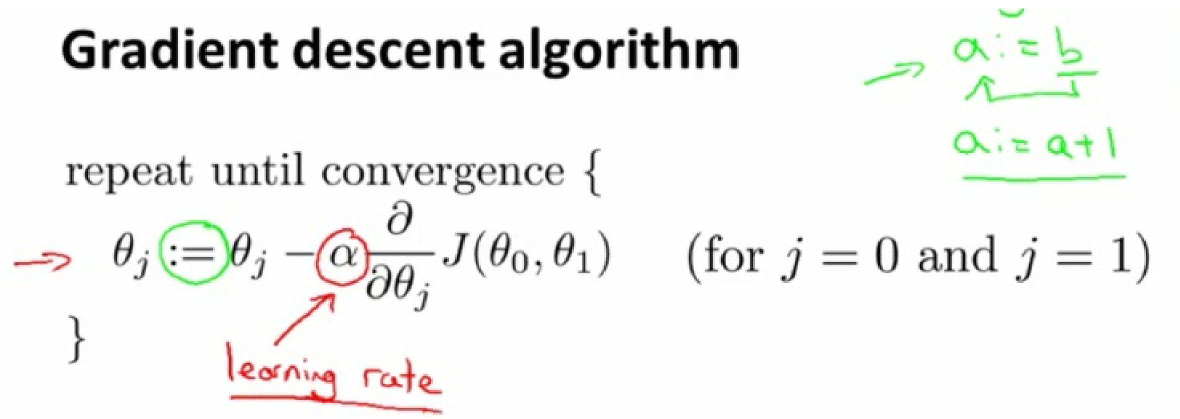
1. 使用梯度下降法寻找最小化代价函数的参数。



梯度下降算法的思想是：下山每一步都选择坡度最大的方向，特点是：最终结果取决于初始值的选择，有可能会陷入局部最优。

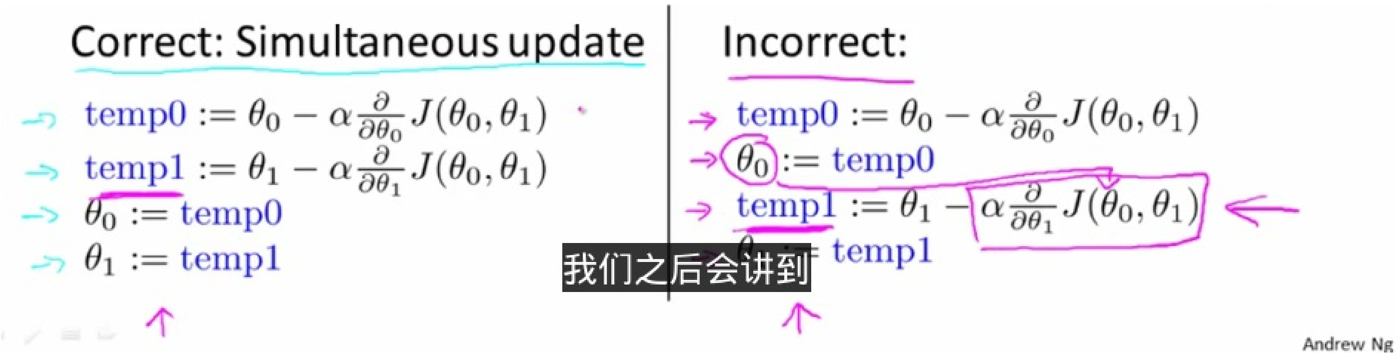


算法为：



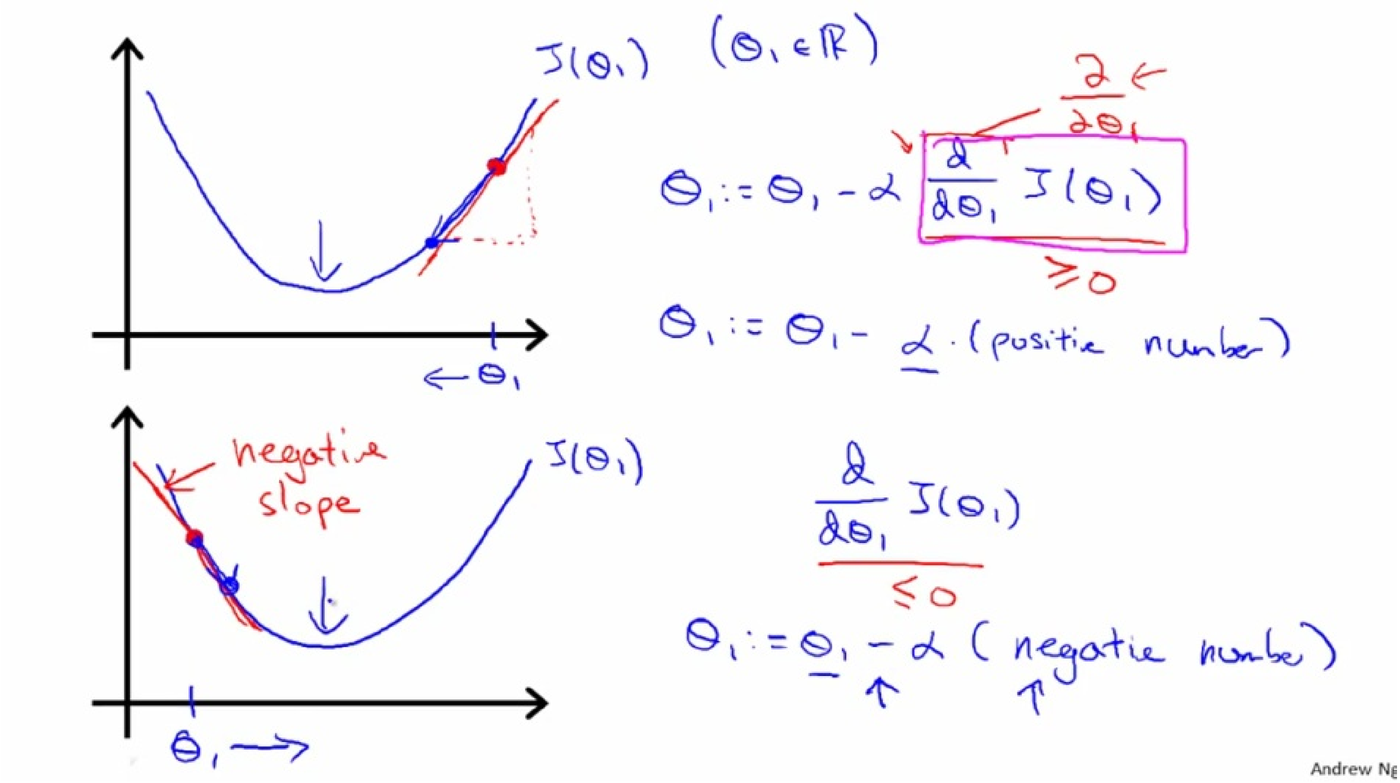
迭代直到收敛。其中alpha是学习速率，其值越大，更新地越快。

算法需要同步更新theta0和theta1:

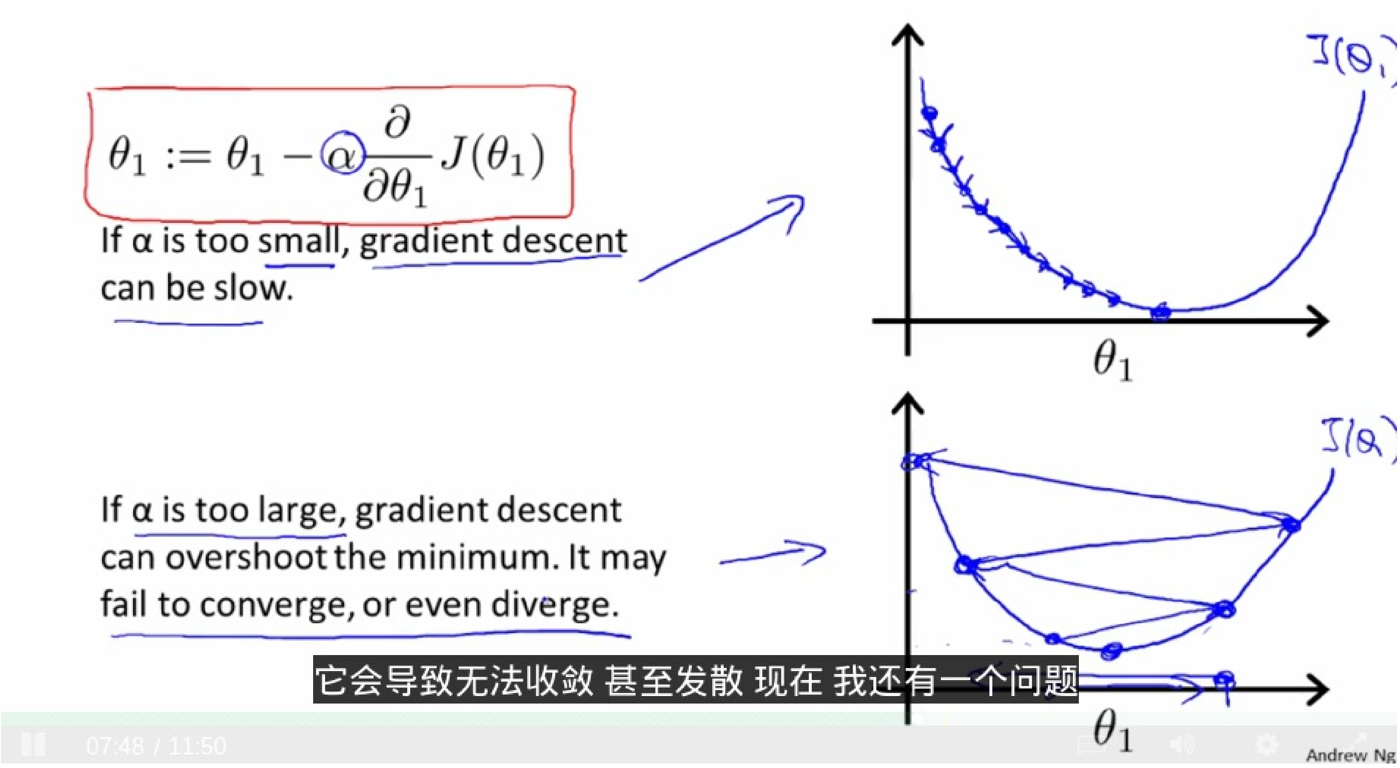


2. 理解梯度算法中的偏导数，首先考虑简单情况，即theta0 = 0:

很显然，沿着导数的方向，函数下降地最快，并且在最小值左边，由于导数为负，更新theta1会增加theta1的值，同理在右边会减小theta1的值，这样便使得theta1不断靠近最小值点：

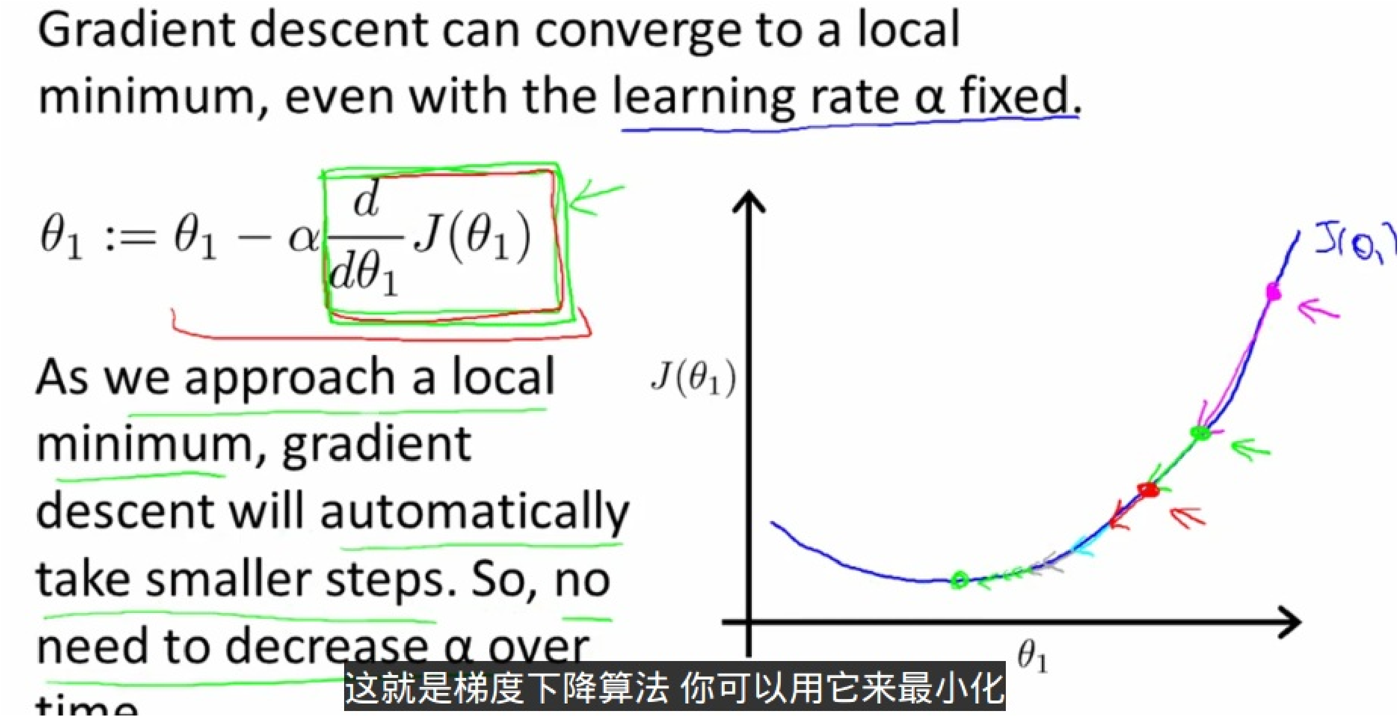


学习速率过小，收敛速度慢; 过大，可能造成不收敛：

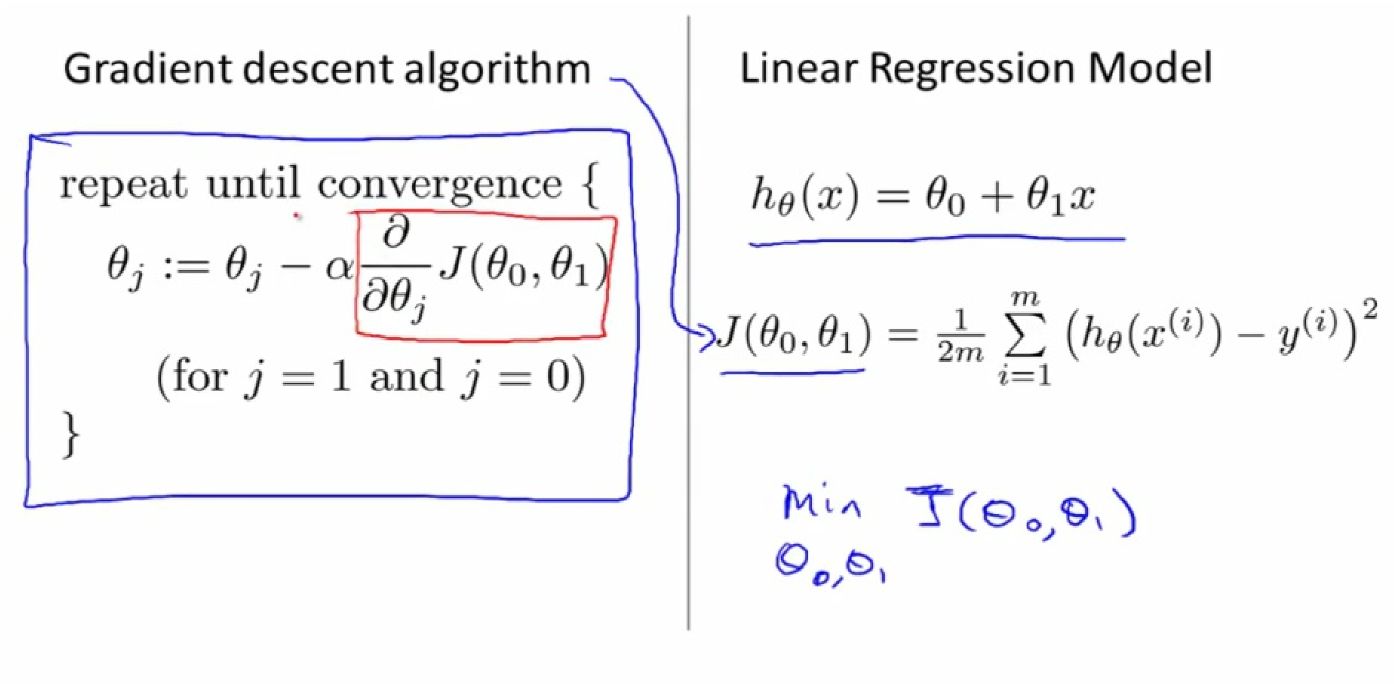


如果已经到达一个局部最低点，因为导数为0, 梯度下降法不会更新参数。如果靠近局部最低点（因为学习速率的选择无法到达局部最低点，将会在局部最小值左右摇摆）

另外，靠近最小值时，导数会变小，因此更新速度会下降，所以不用更改学习速率，梯度下降算法在快要收敛时会自动下降更新速度。



3. 将梯度下降算法应用到线性回归中。



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

$$= \frac{\partial}{\partial\theta_j}\frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0+\theta_1x^{(i)}-y^{(i)}\right)^2$$

$$\theta_0\quad j=0:\ \frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)$$

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

梯度下降算法：

Gradient descent algorithm

$$\frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1)$$

repeat until convergence {

$$\theta_0 := \theta_0 - \alpha\boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)}$$

$$\theta_1 := \theta_1 - \alpha\boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)})-y^{(i)}\right)\cdot x^{(i)}}$$

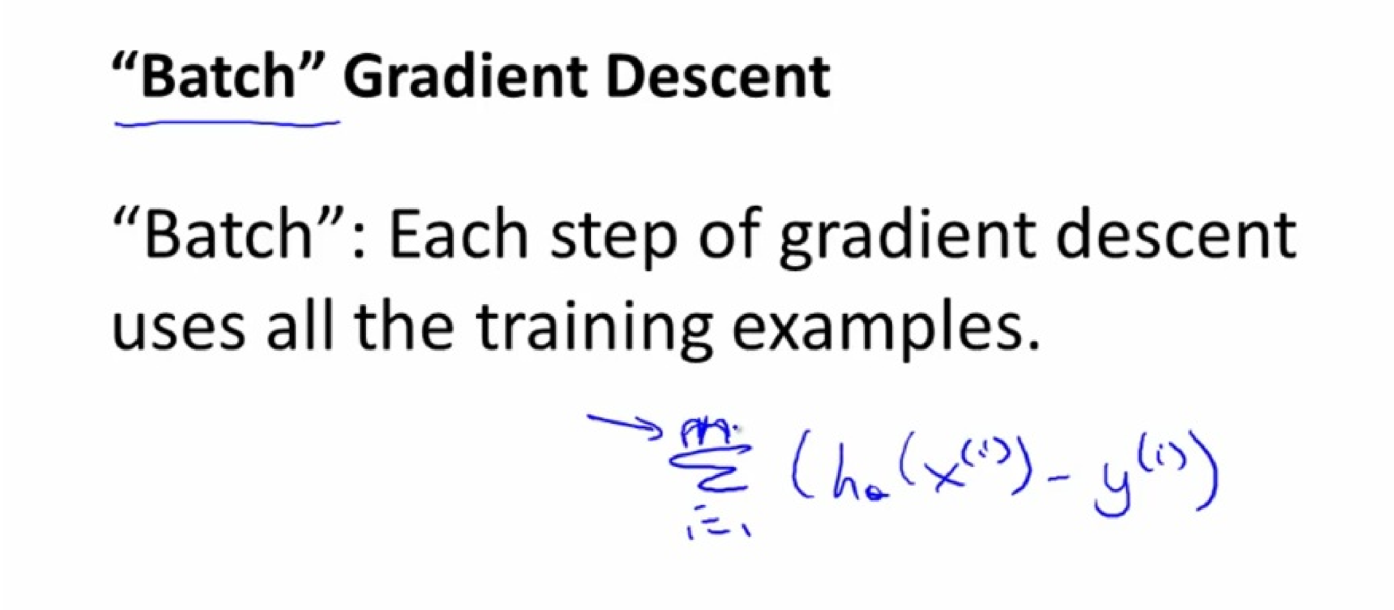
}

update θ_0 and θ_1 simultaneously

$$\frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1)$$

执行梯度下降时

批量梯度下降：每一步用到所有的训练集：



线性代数中，可以不用梯度下降，使用名为正规方程（normal equations）的数值解法。