单变量线性回归_吴恩达_机器学习笔记

1、模型描述

定义： 预测数值是回归类问题，预测离散变量是分类问题

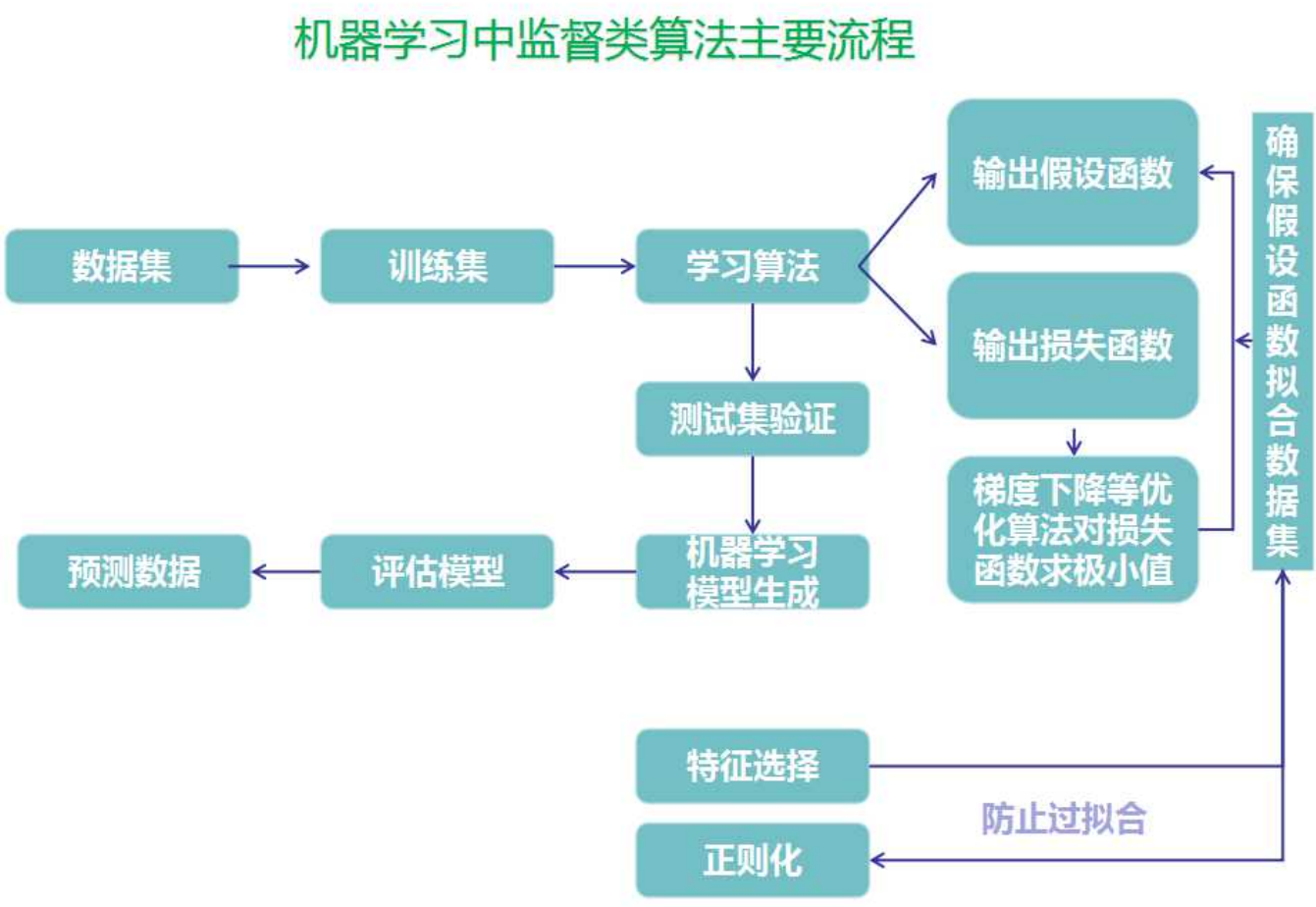
本视频中的常用符号定义

m : 样本数量，代表着每一行的数量

x : 输入变量

y : 输出变量

(x^i, y^i) 表示一个数据集的输入变量和输出变量，这是我晚上画的一个图，可以大概理解本章要说的主要内容。



- 在这个图中，我们也需要注意到，实际上，假设函数，就是我们通常所说的模型。 那么其实回归类问题也是有非常多的模型可以适用的，并不是只有单一的线性回归，对于非线性类问题，还可以通过支持向量机、决策树等等模型做回归类问题。这是初学者应该注意的一个问题。
- 而损失函数，也就是代价函数，都是同一个东西，那么除了梯度下降法，是否还有其他优化类算法呢？初学者建议先不急着去看凸优化类算法文章，除非非常有时间。建议先对梯度下降法这个最简单的算法进行彻底的掌握。实际上，吴恩达在阐述梯度下降法的时候，基本上也表明了，上到百万的数据样本，梯度下降法通常仍然能够很好的求解最优化问题。
- 如果你慢慢掌握了梯度下降，那么牛顿法和拟牛顿法，也是可以开始接触的，我建议看了李航老师的附录相关介绍以后，去相关的博客具体看，牛顿法和拟牛顿法的代码实现，这样，可能才能对这两个更加复杂的算法有一定的理解（其实发明这两个算法的人都是几百年的，唉！我居然没有进化脑子）！

一个简单的单变量线性回归的假设函数如下:

$$h_\Theta(x) = \Theta_0 + \Theta_1 x$$

单变量线性回归和多变量线性回归在业务解释上，均具有很直观的可解释意义。 比如假设函数的系数为正，那么意味着，该项特征对最终结果有正相关关系。例如，房价预测特征中，房屋面积特征的系数为正，表示房屋面积越大，房价越高；又如在房屋年龄上，房屋年龄特征的系数为负，表示房屋年龄越久，房价越便宜。

2、代价函数

- **平方误差函数**

$$minimize_{\Theta 0,\Theta 1} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\Theta(x^i) - y^i\right)^2$$

怎么来理解整个代价函数：

1、对数据的真实值和预测值计算误差平方的平均数的二分之一。

2、为何要取二分之一，原因只是为了数学上更加直观好看。

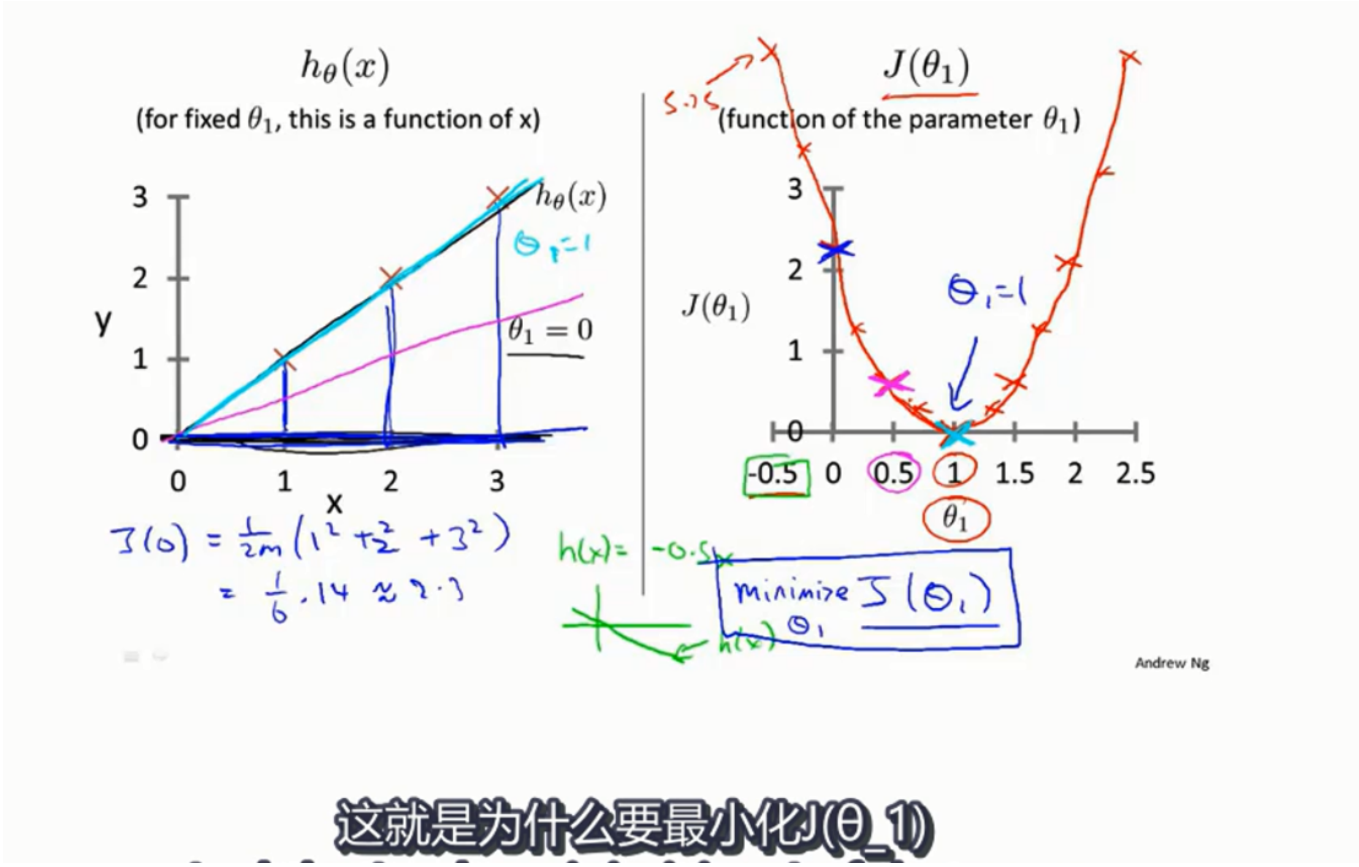
3、结合第一个简单的线性回归函数，那么此处的数据中的

$$h(x^i)$$

就是上面的假设函数的计算结果。 4、如何来求上述公式的最小值呢，我们都知道，当偏导数为0的时候，那么这个公式可以渠道极小值。实际上对上市的两个参数分别求偏导数，就可以求出数据的参数西塔1和西塔2。

2.1代价函数（一）

一张图让我们真正的了解代价函数和假设函数相互之间的关系，已经代价函数在假设函数最佳时，起到了什么样的关键作用？



1、这个图非常的直观和经典了，吴恩达为了让大家理解代价函数和假设函数的关系，用了最简单的参数德尔塔。左边是最简单的单变量的线性回归假设函数，右边是随之对应的代价函数。代价函数的图形，在本次讲解中，通过不同的参数取值来求对应的代价函数值，打点法来告诉我们，其实代价函数在这个地方是一个二次函数。

2、当德尔塔取1的时候，那么假设函数h(x)很好的拟合了数据集中的三个坐标，基本上重合了。

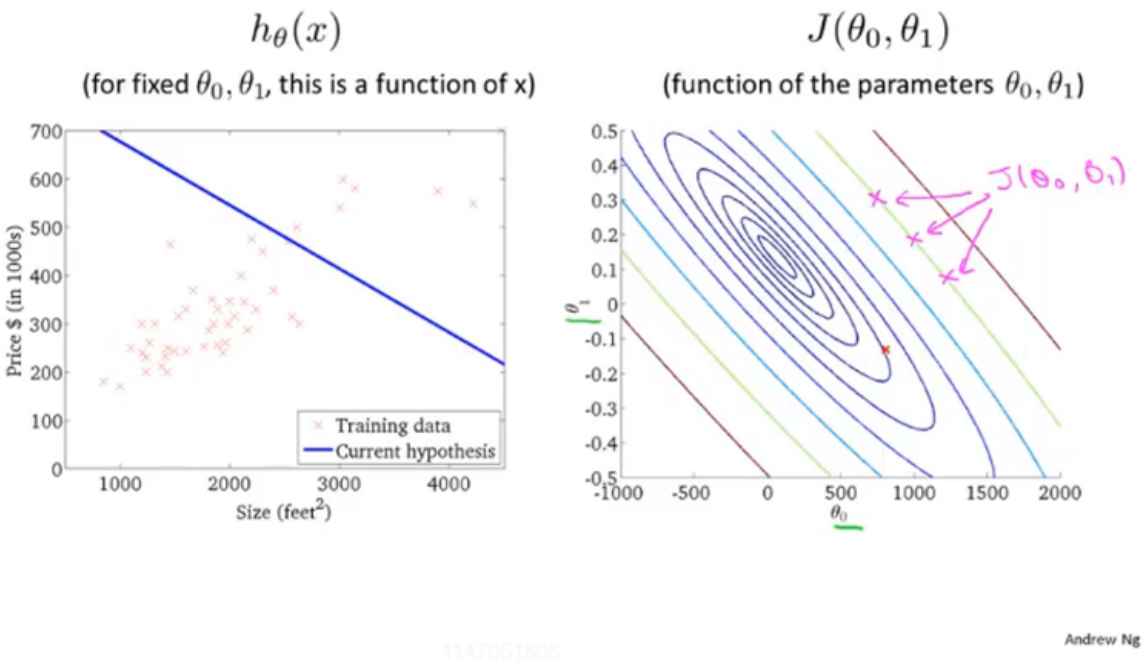
3、再看看，对应的代价函数当西塔为1的时候，取值为0，那么实际上也是在二次函数的最低点，极值点。

4、如果变换假设函数的不同取值，比如，取值为比1大的正数，或者比1小的负数，都会发现，此处对应的代价函数并不在极值点，而这个值会比极值点越来也大。

5、综上所述，当代价函数取到极值点的时候，那么我们的假设函数才能最完美的拟合真实数据。但是，是不是真的完美的拟合真实数据，就一定是我们想要的假设函数呢？并不是，因为还存在过拟合和欠拟合的问题。

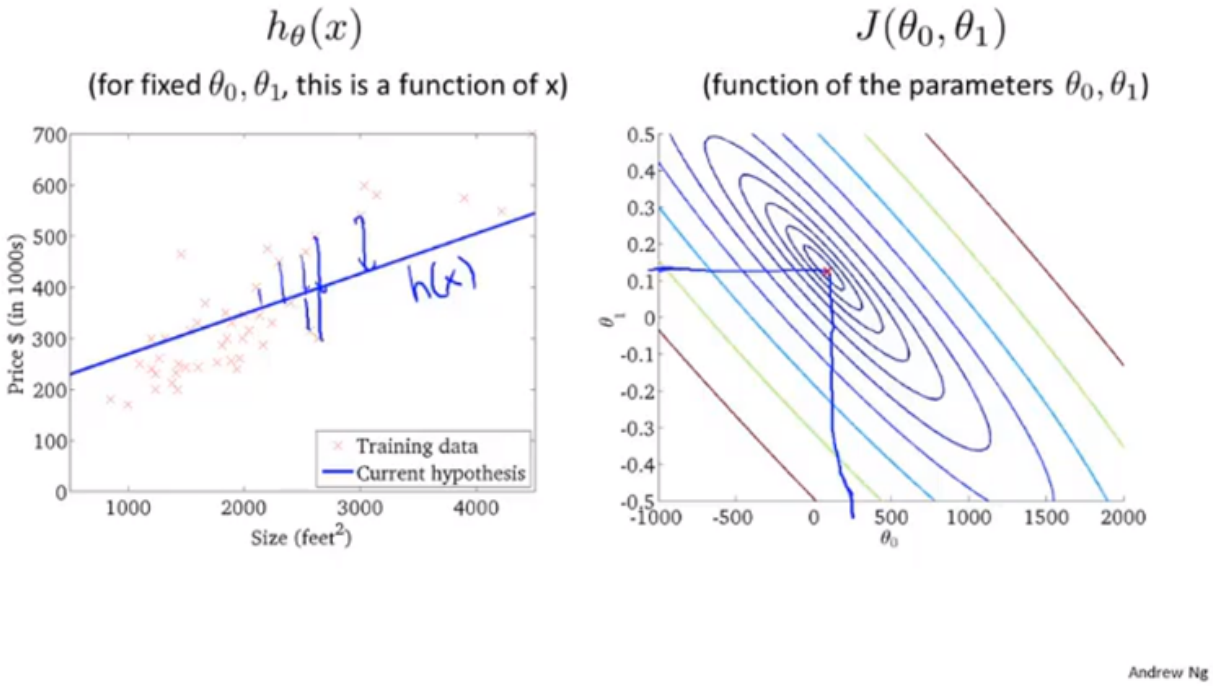
2.1代价函数（二）

当参数在增加是，假设函数包含了θ1和θ2的时候，那么整个代价函数将成为三维数据。想象等高线图中，在一个圆圈内的代价函数优化结果值都相等。具体如下：



但这三个点具有相同的J(θ_0,θ_1)值

那么，不同的参数所产生的假设函数，是否能够拟合数据，将取决于三维的凸函数锥形曲面是否能够下降到最低点，当代价函数在接近最低点的位置的时候，那么θ1和θ2构成的线性函数将能够最好的拟合原始数据。



但相当接近最小值

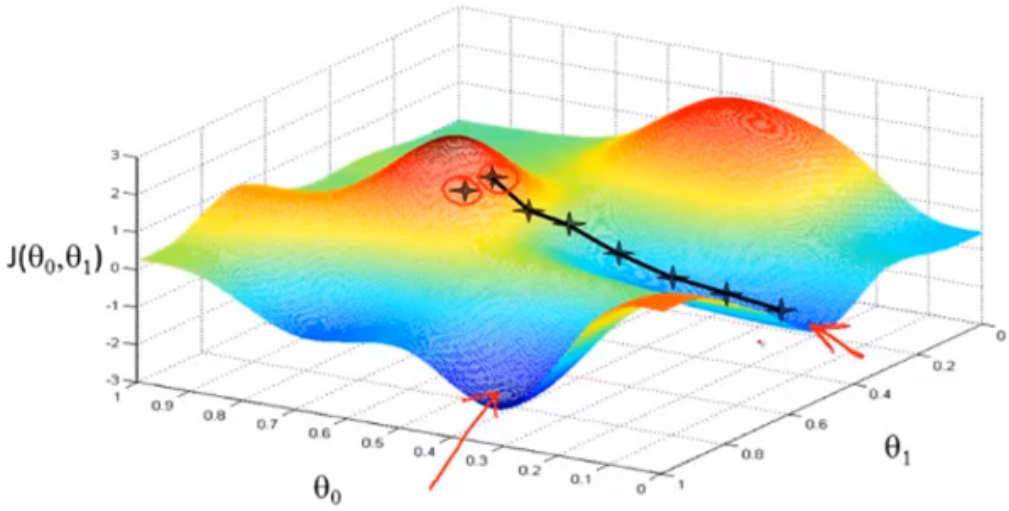
至此，我们基本上可以完整的理解代价函数和假设函数到底分别在从事什么样的工作了。

3、梯度下降

在本节中，将只使用θ1和θ0作为案例进行讲解，方便更加容易的理解。

梯度下降算法中，一般会将θ1和θ0的初始值均设置为0，然后再进行一步步的迭代计算出能够使代价函数下降到极小值的θ1和θ0。

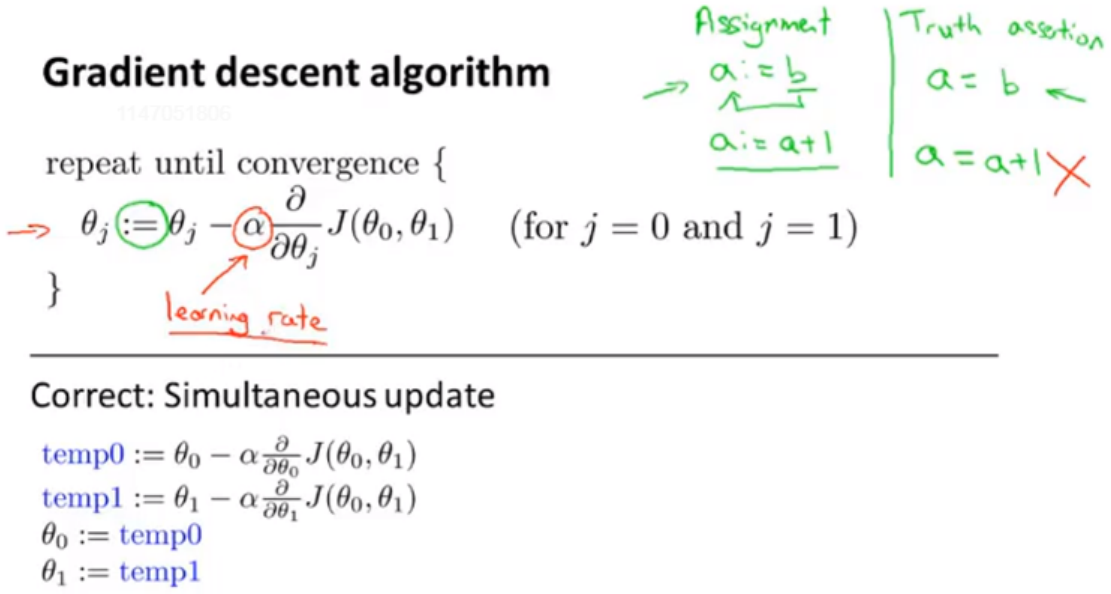
梯度下降算法的原理简单理解：就是一步一步选择一个比原来更低的点下山，那么这里涉及到两个重要变量进行考虑，第一，到底什么方向才是下山的方向，第二，每一步迈多大的步子，才能够保证最终收敛于一个相对的最低点，而不是在最低点的时候，因为步子迈太大而导致回到了相对的高点。



Andrew Ng

你会得到一个完全不同的局部最优解

上述问题涉及到到两个问题，那么可以理解，第一步，先对原始的代价函数求偏导数，这样新的参数就等于原来的参数值减去学习率与偏导数的积，如果这个结果在仍然变小，那么就表示在正确的下山；第二步是考虑学习率的问题，就是步子迈多大，才能保证到了最低点的时候，能够不会发生震荡。



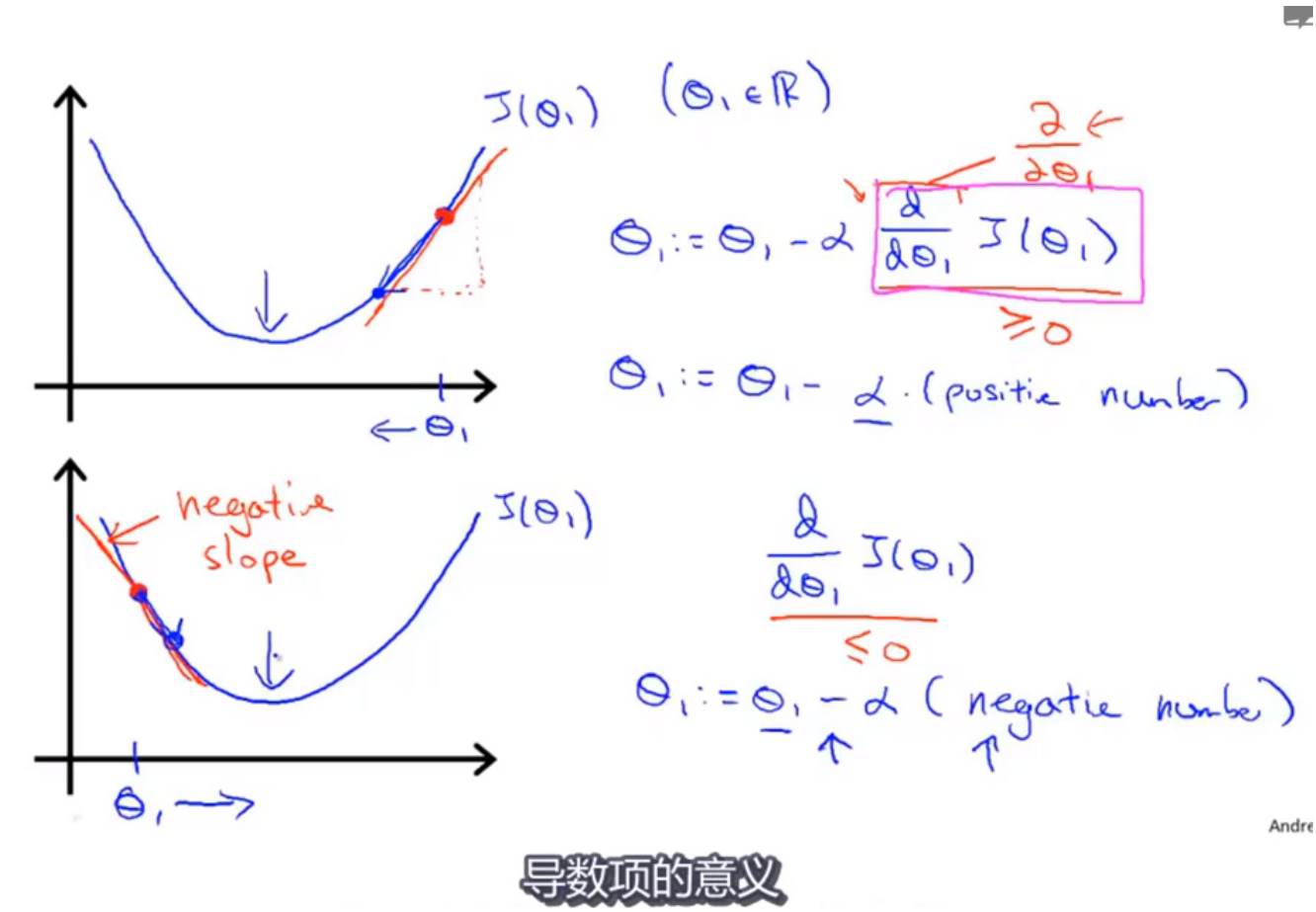
上图就是梯度下降的具体表达式了，为了理解这个表达式，需要重点观察这个表达式的以下几点：

- 这里的 ：= 表示的是赋值的功能，就是将旧的参数赋值到新的公式中进行梯度下降。
- 这里的学习率α如果很大，那么下降的速度就很快，当然，也可能会造成震荡。因为步子太大，可能直接越过了最低点，跑到了相对不那么低的地方去了。如果学习率比较小，对于比较大的高维数据集，可能时间开销会比较多，那么怎么设置学习率呢，一般的取值包含如下： 1 、 0.1 、 0.01、 0.001 或者 3、 0.3、 0.03、 0.003等，最好是有一些倍数变化来进行测试。
- 有时候，需要直观的观察梯度下降算法的代价函数结果和学习率之间的设置关系和变换趋势，以确保梯度下降算法在正常的工作。
- 这里的求偏导数步骤一定要注意，一定是先对所有的参数求完偏导数，然后再进行新一轮的赋值工作，否则就会导致梯度下降算法偏离原来的设定步骤。

3.1 梯度下降知识点总结

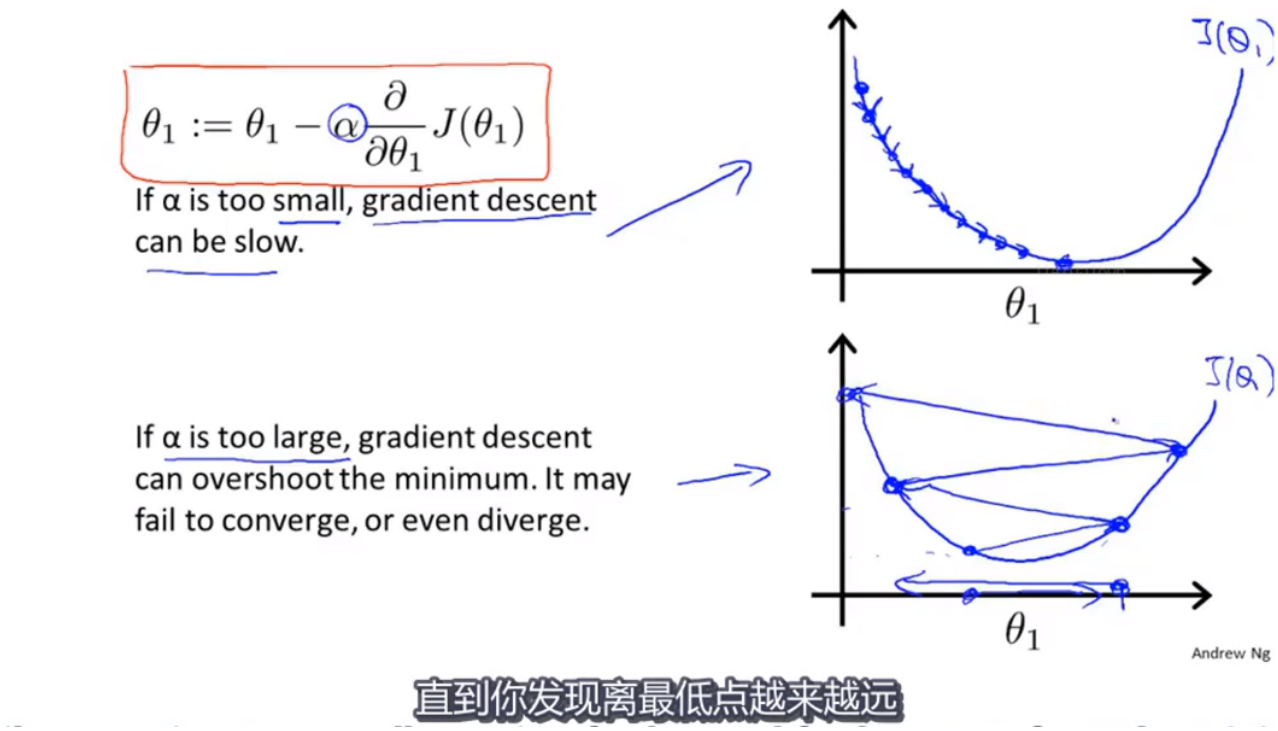
本节我们需要知道，为什么梯度下降算法是这样设置的，偏导数项在这里面产生了什么样的意义？学习率到底又起到了什么样的作用？

为了方便理解，这里只设置了代价函数只包含一个参数θ1。那么只包含一个参数的代价函数将成为一个二次凸函数，实际上，导数项在这里起到了，无论初始化下降开始点是在最低点的左边，那么导函数将保证梯度下降的值在不断增大，直到到达最低点；还是在最低点的右边，那么导函数保证梯度下降值在不断减小，直到到达最低点附近，具体如下图。

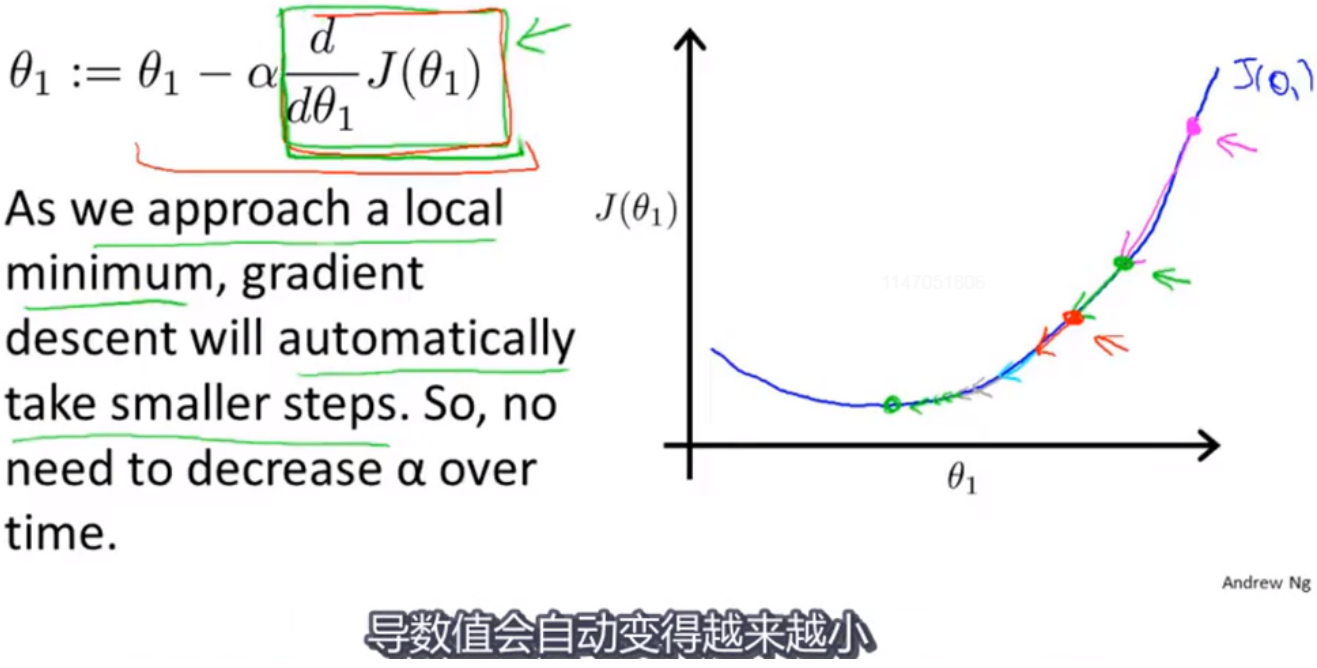


导数项的意义

学习率α在这里起到的作用，是保证能够正确的下降到最低点，而这个过程到底是大步下山，还是小碎步下山，还是说，下山过快导致冲过了最低点，然后代价函数的参数值θi在不断的增大，但是代价函数已经无法再取到最低值了。



再来看看导数项的变化将产生什么样的影响，上图中，实质上，斜率是在不断减少的一个过程，一开始斜率变化比较大，但是随着越来越接近代价函数的局部最低点，那么斜率会越来越小，这样，在同等学习率α的情况下，导数也会导致代价函数在越来越，越来越慢的靠近最小值点，这就是导数变化趋势的一个观察。



那么，这里还留下一个问题还没有解决，我们都知道，非凸函数可能存在极值点一个和多个局部极值点，那么梯度下降算法，怎么调整可以保证尽量的在最极值点进行收敛，而不是局部极值点进行收敛呢，吴恩达在本视频中虽然没有说，但是我需要去思考和考虑这个问题。当然，凸函数的话，梯度下降法是能够保证求到的极值点是收敛到最值点的。

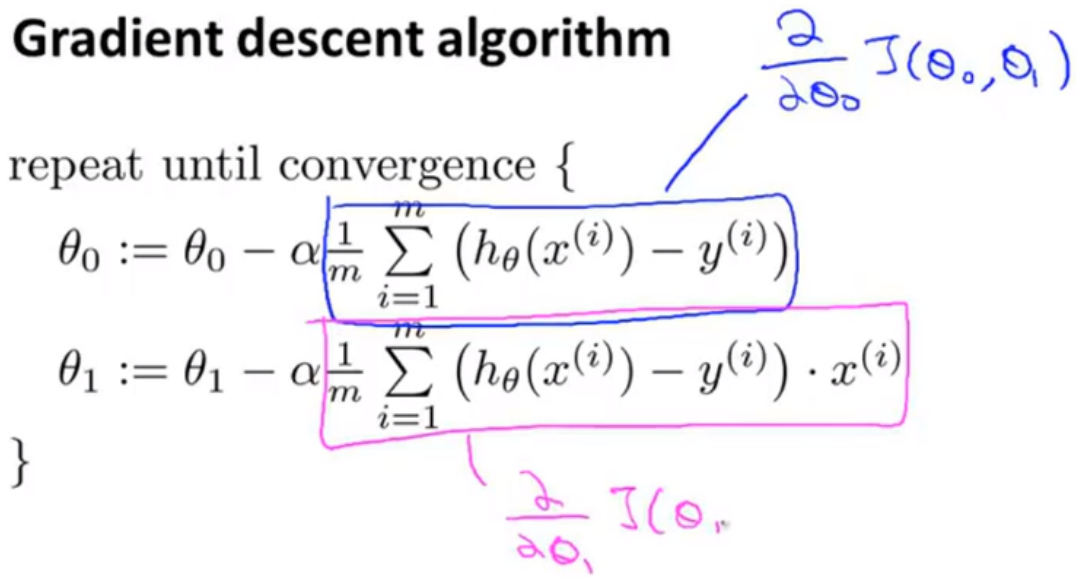
可以参考这里： https://ctmakro.github.io/site/on_learning/gd.html

4、线性回归的梯度下降

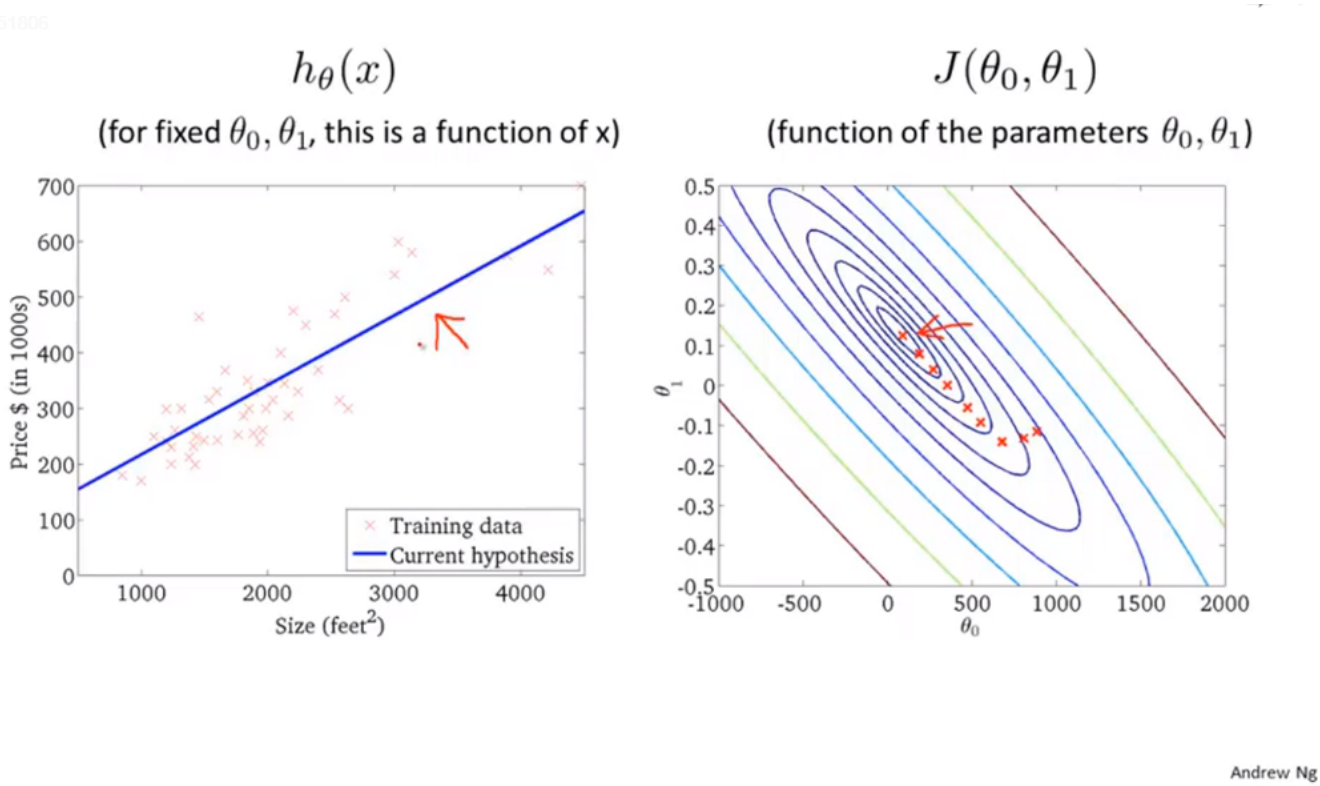
通过一个案例，我们仍然能够确定，只要是凸函数，那么只有梯度下降是可以保证能够求到唯一的收敛值的。

从下图中，我们可以理解，当一个线性假设函数，随着梯度下降的变化情况下，是如何一步步拟合数据的。此处只展示一张图，其他图自行想象。

先熟悉一下线性回归的代价函数梯度下降算法表达



下图是梯度下降到最终结果的展示图



这个全局最小对应的假设曲线

5、本章总结

至此，学习了整个梯度下降算法在线性回归中都干了什么，怎么实施的，结合最小二乘法，可以有更加深入的理解。

完整PPT看这里：

https://study.163.com/course/courseLearn.htm?courseId=1004570029#/learn/text?lessonId=1050362429&courseId=1004570029