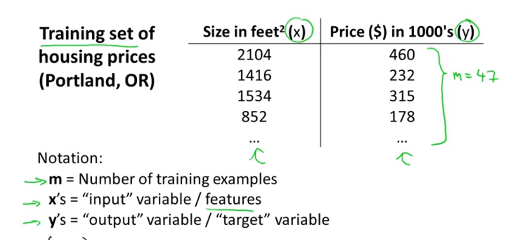
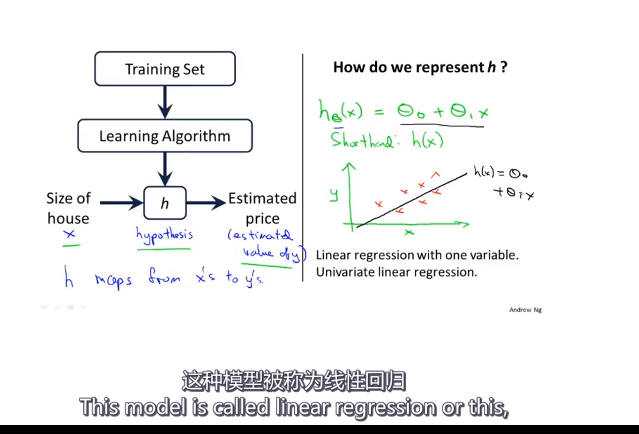
第一个章节，主要是介绍了，那个监督学习和非监督学习的区别，所谓监督学习，它是有一个标签用来判断这是不是这个东西的，比如我们用它来预测房价，我们用它来识别这个是不是癌症，用它来判断是不是猫，而无监督学习通常是分为几个蔟，还可以对混合声音的处理，使之分离出来

A regression problem 回归指的是我们预测一个具体的数值的输出，也就是价格

另一个最常见的监督学习的问题，就是分类问题



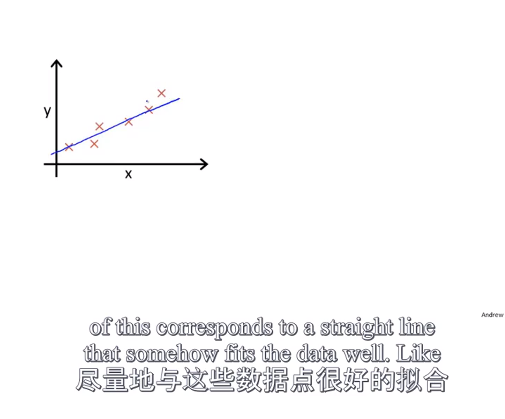
我们用一个（x，y）来表示训练样本，表格中的一行，代表一个训练样本



这种模型也被称为线性回归

2.7 代价函数

目的是最有可能的将我们的数据与直线 像拟合

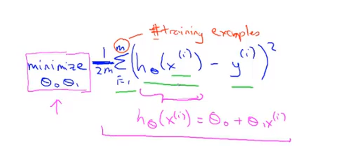


我们要做的就是构造函数，使它能够更好地和它相拟合

------》也就是我们所预测的{h（x0）-y}²要尽可能地小才行，

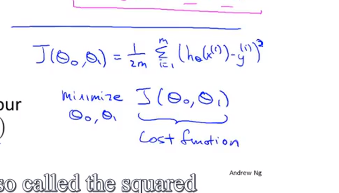
所以就是将1/m Σ1~m{h（x0）-y}²

我们要做的就是找到01,02



值最小

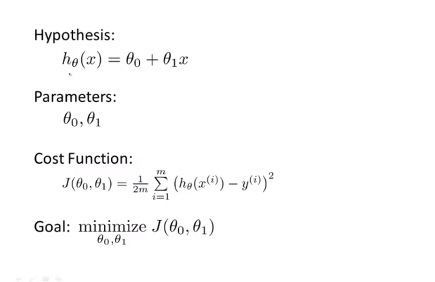
所以我们定义的代价函数就是这个啦哈哈哈



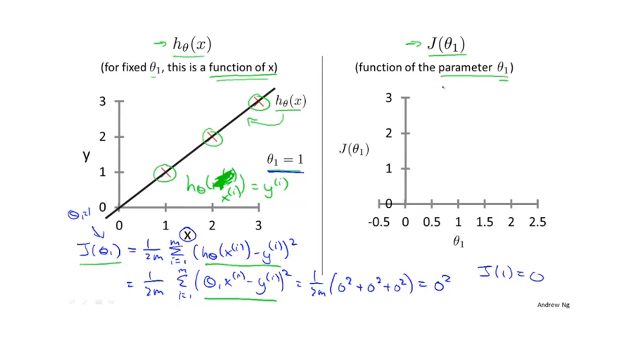
代价函数也被称为平方误差函数

So 平方误差代价函数可能是解决回归问题最常用的手段了

2.8代价函数是做什么的



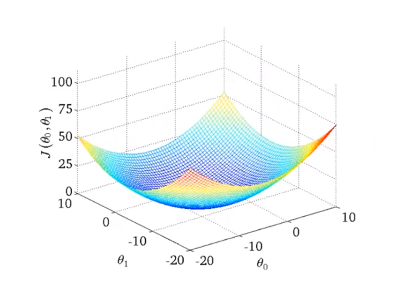
为了拟合一条直线，我们构造了一条假设



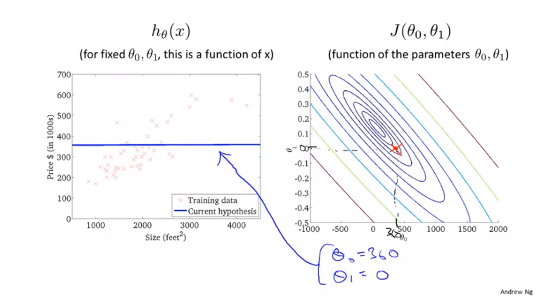
学习算法的目的在于获得一个值01,使得代价函数j取最小值

当我们有两个参数的时候，我们画出的图像是一个碗状的图像

00和01两个参数



3d的代价函数的图像



等高线图如图所示

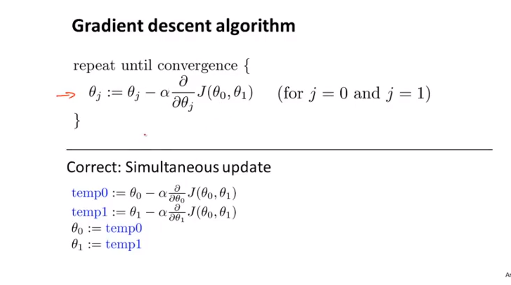
我们正真想要的是一个高效的算法，来自动寻找代价函数j的最小值

2.10 梯度下降是很常用的算法

可以将代价函数j最小化的梯度下降法

当然的梯度下降不仅仅是只用在线性回归中，而是用到很多的函数中

梯度下降法的大概思路



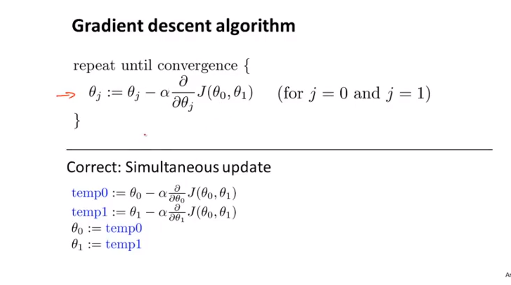
这里的α被称作为学习率这个东西

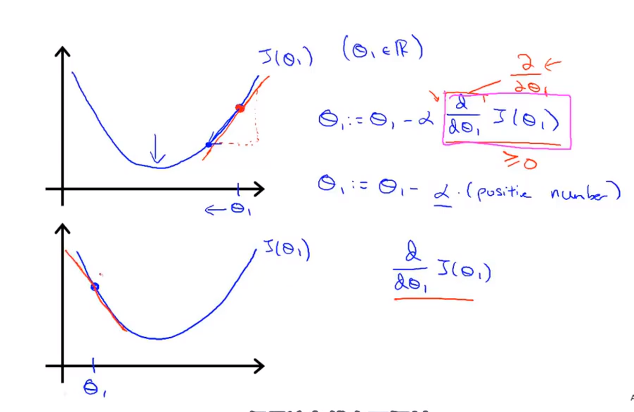
阿尔法用来控制我们梯度下降时，迈出的步子有多大，如果学习率越大，我们下降得就越快

在梯度下降中，我们需要不断更新0的值

正如正确的表达式中的式子

2.11 梯度下降的知识点总结

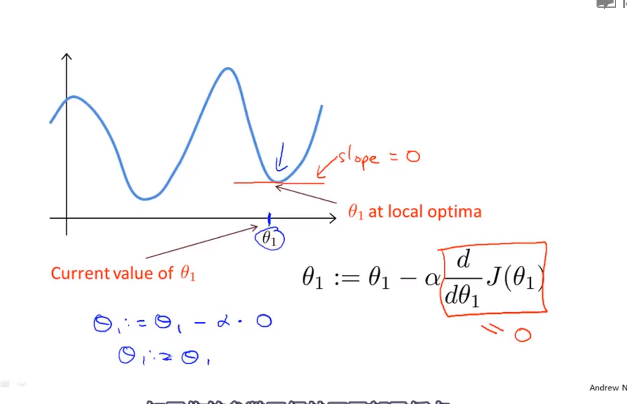




这个东西就是斜率，所以才会朝着中点的值，不断靠近

所以关于学习率一定要合适，才能更好地到达中间呐

如果学习率过大的话，可能造成无法收敛或者无法发散

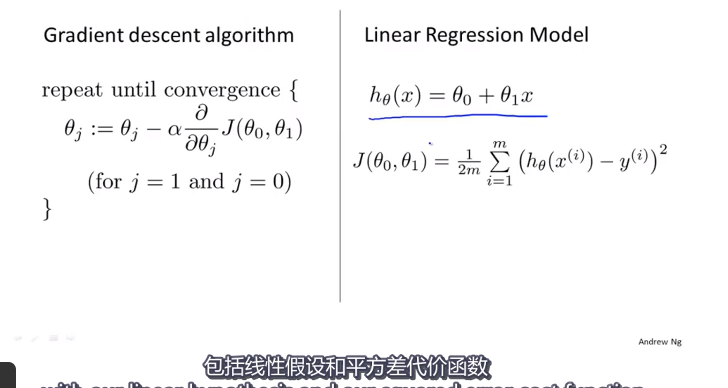


如果到达局部最低点的意思

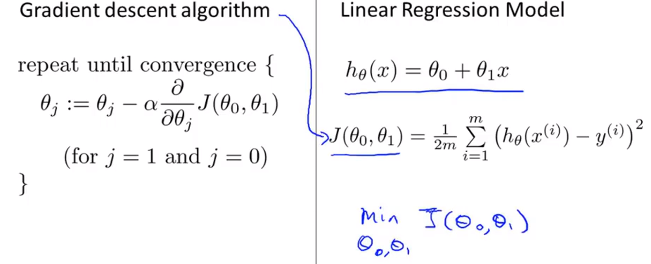
当我们发现我们的值越来越小，减不动的时候，我们就已经发现了，收敛到了局部最小值

2.12 在之前的内容，我们讲解了梯度下降算法线性回归模型，以及平方差代价函数

我们将梯度函数和代价函数结合，得到线性回归算法，他可以用来直线拟合数据

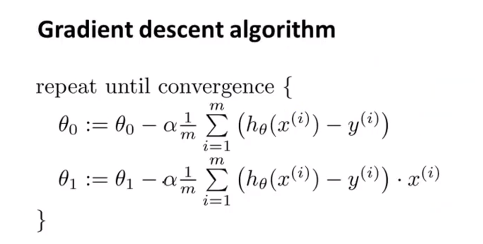


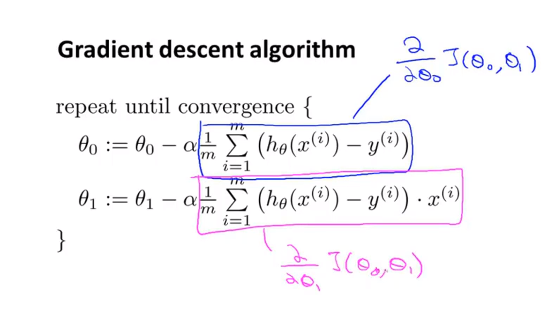
我们要做的就是这个，如下图所示



为了写好代码，我们学习的最关键的就是这个导数项

计算出偏导数数后，带回去就可以更新参数了



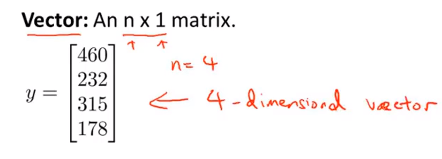


刚才的算法，也叫batch梯度下降法

引子：求解代价函数j最小值，而不需要使用梯度下降的迭代算法，它被成为正规方程组方法

3.14 矩阵和向量

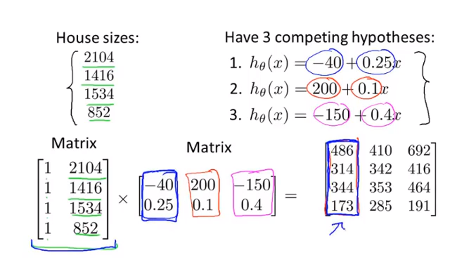
向量是一种特殊的矩阵，只有一列的矩阵



N的话可以代表是几维的向量，这里的话指的是4维向量

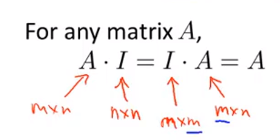
通常的情况下，我们用大写字母代表矩阵，小写字母代表向量

3.17



3.18矩阵乘法特征

矩阵乘法不符合交换律，但是符合结合律

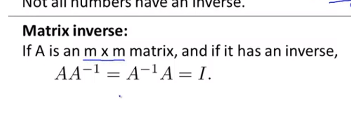


这个技巧还是要好好掌握

3.19特殊矩阵运算

从自然数开始，我们发现每个数都有一个倒数

比如3有一个倒数，与之相乘后得1



矩阵的逆运算

矩阵的转置运算