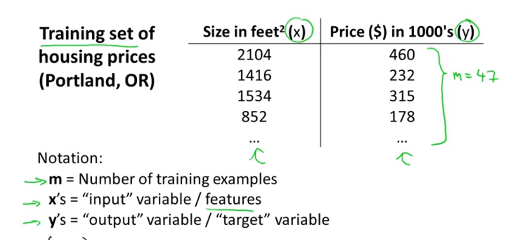
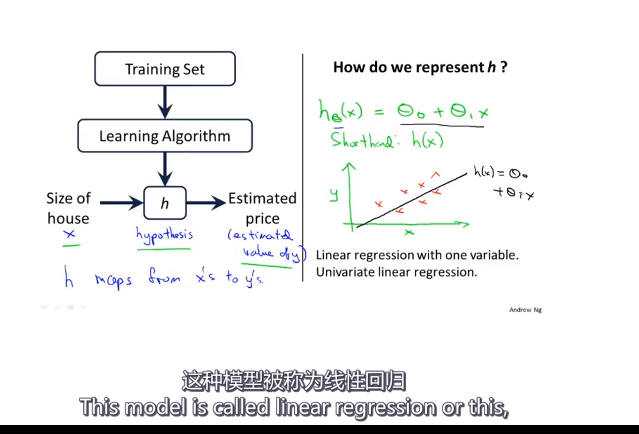
第一个章节，主要是介绍了，那个监督学习和非监督学习的区别，所谓监督学习，它是有一个标签用来判断这是不是这个东西的，比如我们用它来预测房价，我们用它来识别这个是不是癌症，用它来判断是不是猫，而无监督学习通常是分为几个蔟，还可以对混合声音的处理，使之分离出来

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



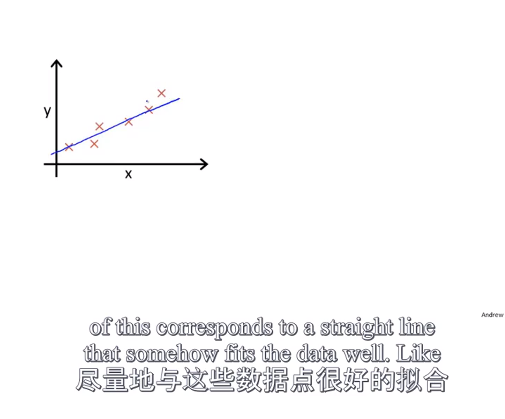
我们用一个（x，y）来表示训练样本，表格中的一行，代表一个训练样本



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

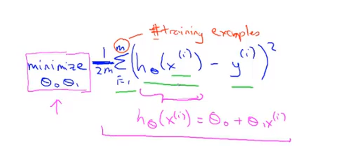


我们要做的就是构造函数，使它能够更好地和它相拟合

------》也就是我们所预测的{h（x0）-y}²要尽可能地小才行，

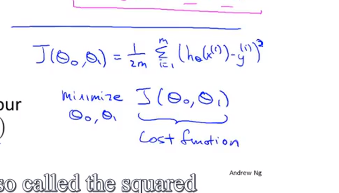
所以就是将1/m Σ1~m{h（x0）-y}²

我们要做的就是找到01,02



值最小

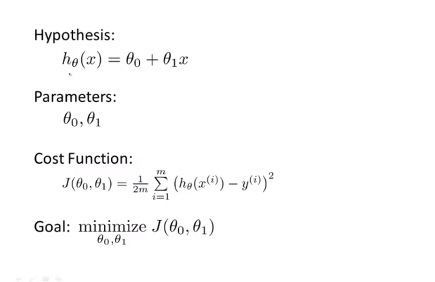
所以我们定义的代价函数就是这个啦哈哈哈



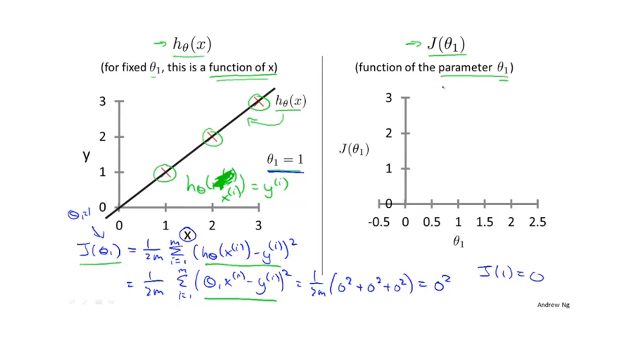
代价函数也被称为平方误差函数

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



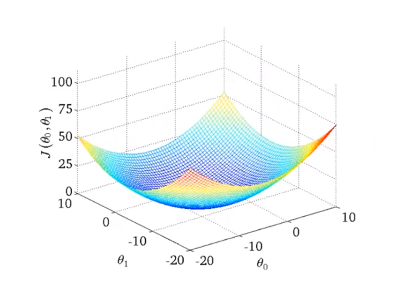
为了拟合一条直线，我们构造了一条假设



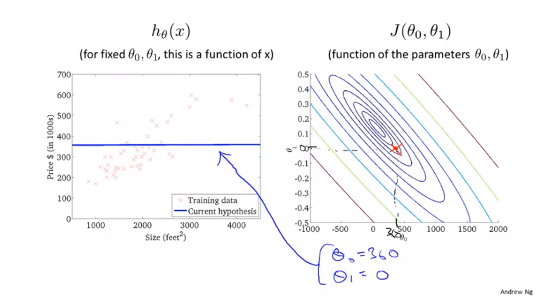
学习算法的目的在于获得一个值01,使得代价函数j取最小值

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

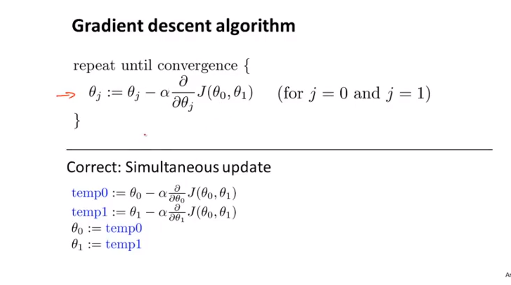
我们正真想要的是一个高效的算法，来自动寻找代价函数j的最小值

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



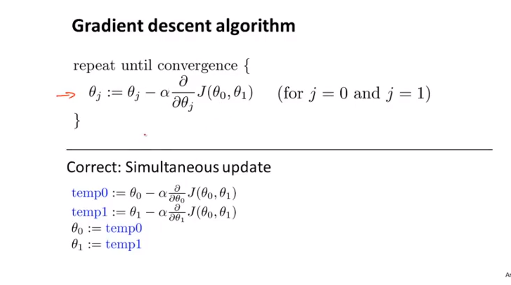
这里的α被称作为学习率这个东西

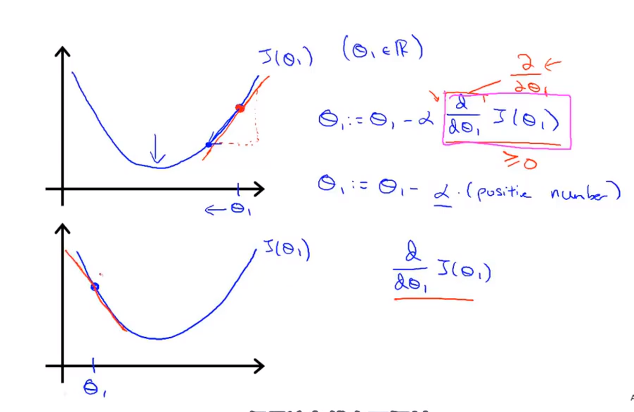
阿尔法用来控制我们梯度下降时，迈出的步子有多大，如果学习率越大，我们下降得就越快

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

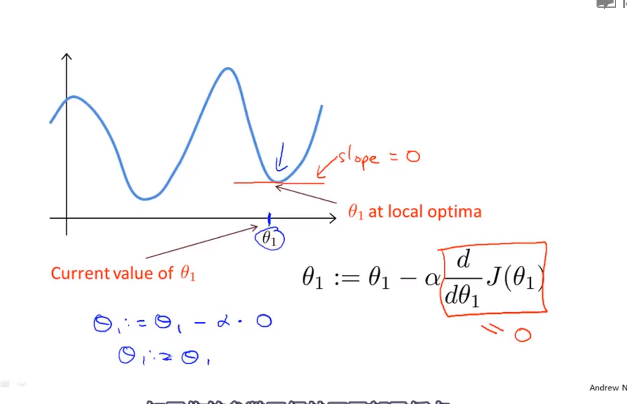




这个东西就是斜率，所以才会朝着中点的值，不断靠近

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

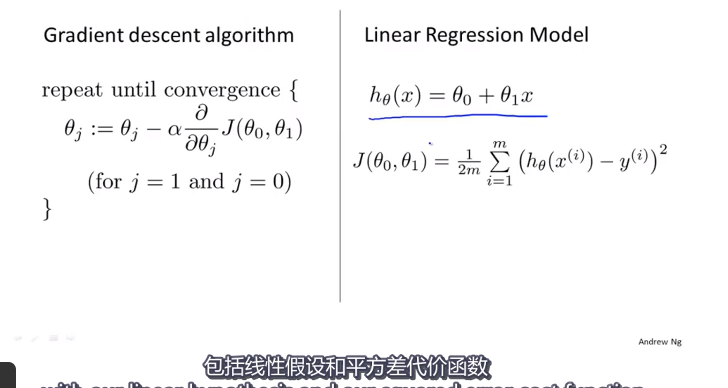


如果到达局部最低点的意思

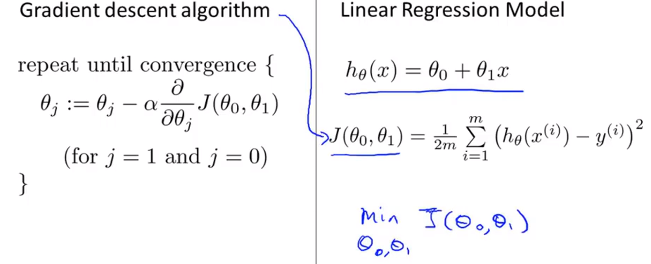
当我们发现我们的值越来越小，减不动的时候，我们就已经发现了，收敛到了局部最小值

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

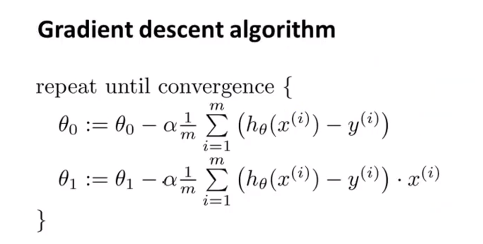


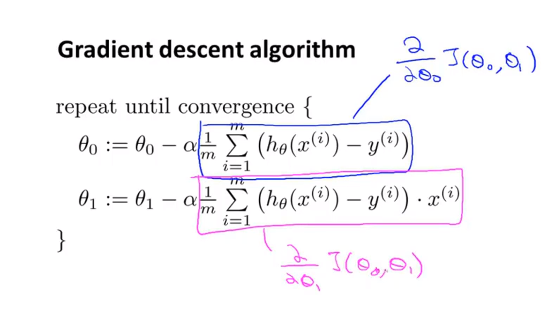
我们要做的就是这个，如下图所示



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



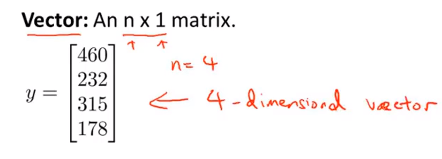


刚才的算法，也叫batch梯度下降法

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

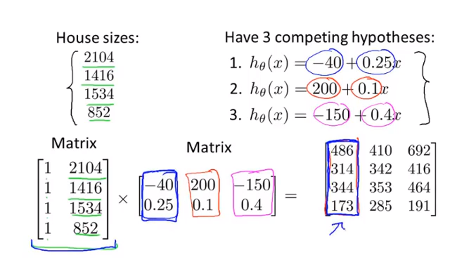
向量是一种特殊的矩阵，只有一列的矩阵



N的话可以代表是几维的向量，这里的话指的是4维向量

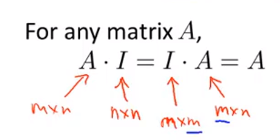
通常的情况下，我们用大写字母代表矩阵，小写字母代表向量

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

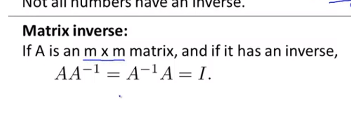


这个技巧还是要好好掌握

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



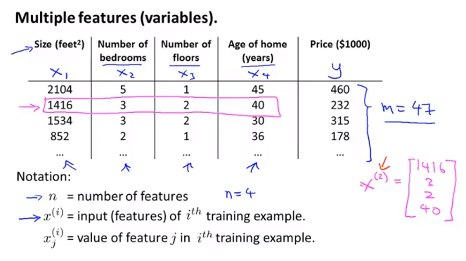
矩阵的逆运算

矩阵的转置运算

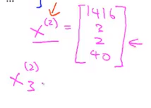
Day two

今天我们讨论线性回归的另一个版本适用于多个变量，或者多特征量

N表示特征量，m表示的是行数

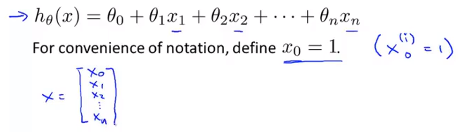


这个2就是一个训练集的索引

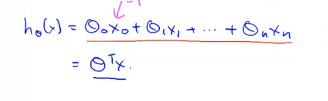


我们的模拟预测函数也会随之下降

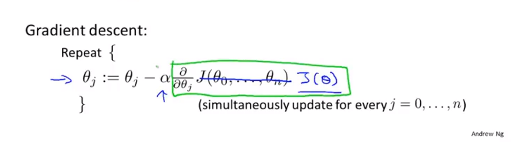
所以我们的特征向量也随之改变了



所以这个目标函数就可以写成转置的形式



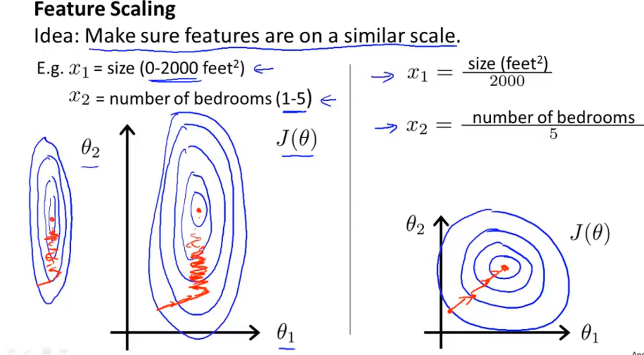
经过向量参数的封装后，代价函数和梯度下降就变成了这个样子



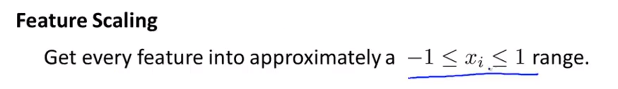
多元梯度下降法演练——特征缩放

就是单位的换算之后呢，梯度就没有那么抖了，就可以非常的好画了

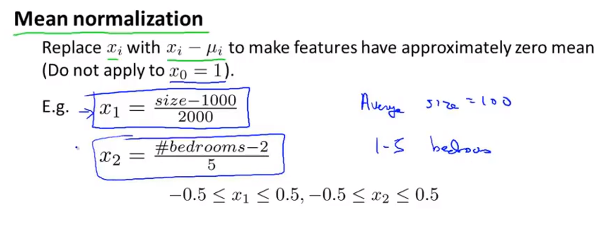
如图所示



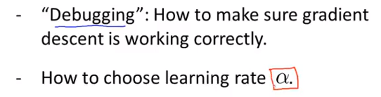
我们通常做的事情呢是将特征值的取值，取到



当然也要用到均值归一化，来初始你给的参数



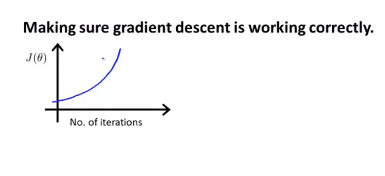
31 多元梯度下降法的使用技巧，我们将讨论学习率α



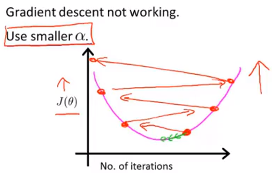
我们实战中所用的技巧

另外也可以进行一些自动的收敛测试---也就是采用一种算法来判断是不是已经收敛的

当然图像还可以方便你看bug，如果看到图像这个样子

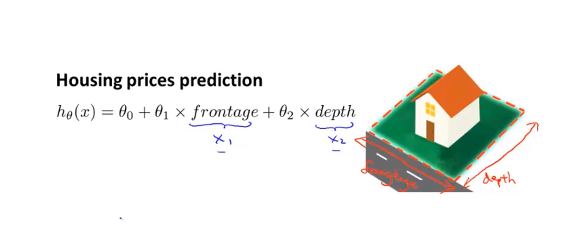


改变学习率，因为他很可能是梯度下降这样形成的



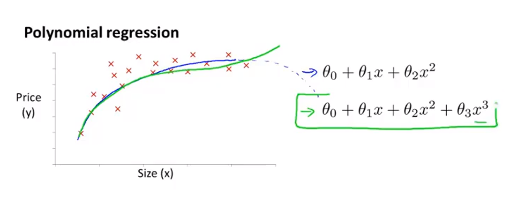
32 特征和多项式

这节课将会告诉我，一些可供选择的特征，以及如何得到不同学习的算法，当选择合适的特征,这些算法往往都是有效的，讲一讲多项式回归它使得你能够用线性回归来拟合非常复杂的函数，甚至是非线性函数



比如给到房子的深度和宽度，你可以自己创造一个变量面积来表示特征

假如我们的数据看着不像线性回归，就要用其他的模型去拟合

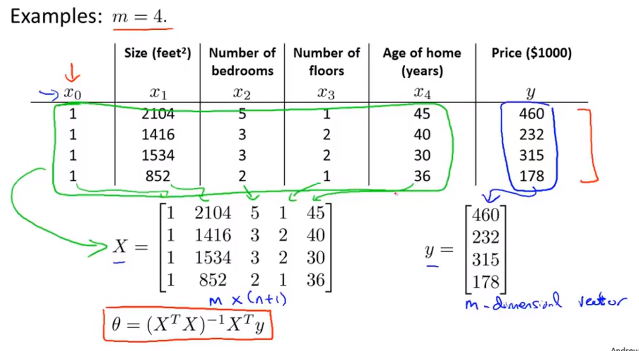


**通过把特征的三次方，可以变为x1，x2，x3刚好又回到了以前**

**构造的线性模型（多元多项式）**

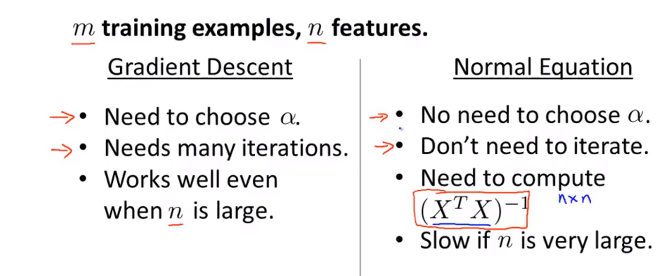
如果你选择这样构建模型的话，那么将数据变小就显得异常重要了

33正规方程（区别迭代法的直接解法）



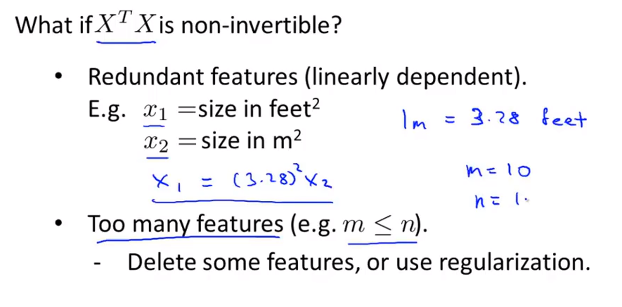
这样就能求的使函数最小化代价函数0的矩阵了

这里就是梯度下降和正规方程的优缺点



34 正规方程以及不 可逆性

可能出现的情况



Pvin叫做伪逆

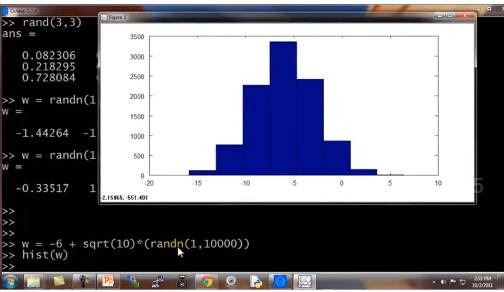
38.octave教学

机器学习最常用的语言



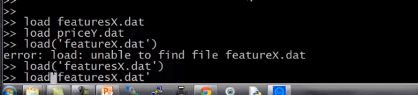
A =[2 1; 2 3; 7 8]

;号表示换行



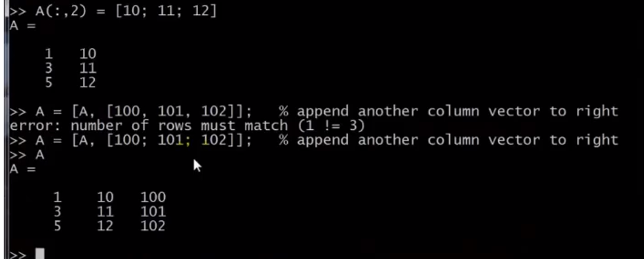
39移动数据

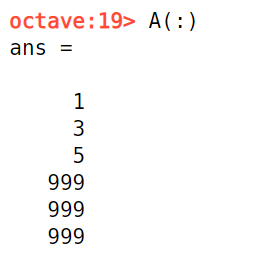
如果你有机器学习得到数据，怎样加载进去，怎样运算，怎样操作他们



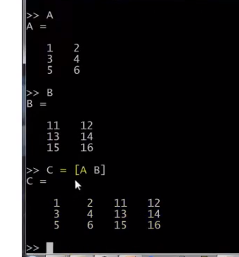
：表示该行，该列的所有元素

加数据就是这个





将所有的A放到一个新的向量里面去



结合起来

40计算数据

A .\*B的就是A里的所有的元素与B相乘

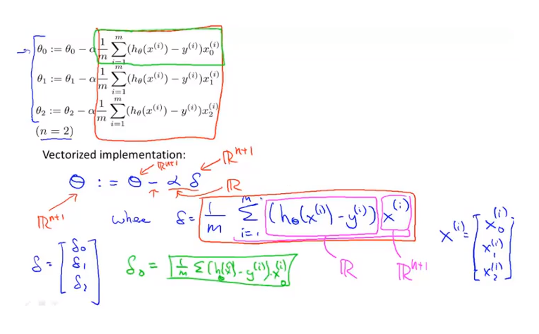
倒数的话为

1 . /A

41 数据的可视化

43 适量

适量的运用就是方便减少不必要的代码量的



即就是这个五颜六色的东西含那个是个实数

Day3

46 logistic regression classification

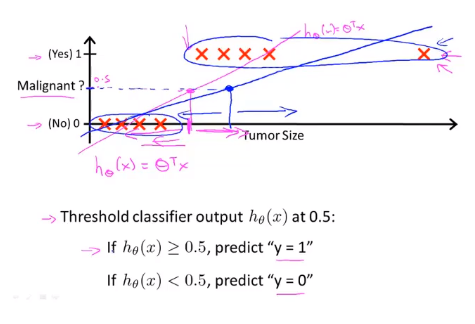
我们要预测的变量y是一个离散值情况下的分类问题

之前我们学的是预测，这次我们学的就是分类算法

通常这类算法的值，有两个，第一个0，叫做负类，第二个1叫做正类，而

0~1之间叫做接近程度，以我们的损失函数一样，这里也有损失函数

如何开发一个分类算法就是我们这个章节要学习的东西呐



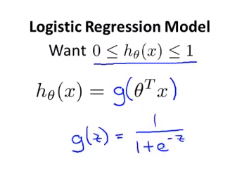
如果我们还是用线性问题的眼光去看他们的话，就会发现不成立了

Logistics regression是一个分类算法用在标签离散值是0或1 的情况下

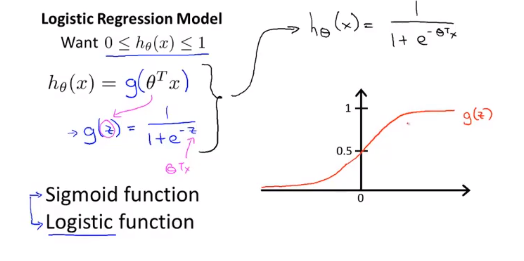
47-怎样表述我们的预测值方程

之前我们知道了我们希望我们的分类器输出的值在0到1之间，也就是是与不是

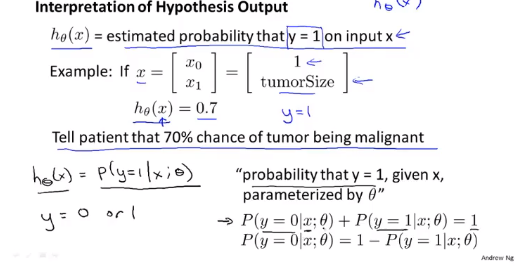
这里的话我们只需要做一个小小的改变



通过这个小小的改变就可以将



从sigmoid function可以看出他的预测值永远都在0-1之间浮动

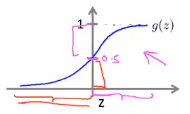


这里的话，是给定参数Θ，对具有某个特征X的人估计

48 决策界限

具体来说这个目标函数是在给定的x，θ的情况下预测y=1的概率

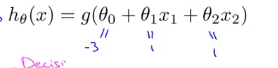
在什么时候，我们预测1或者0，由图可知



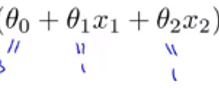
在Z＞0的时候，也就是θtX》0的时候就是我们这样预测的

首先，表示出目标模型函数

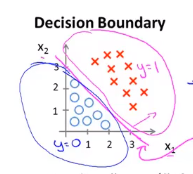
我们先假设出我们的参数为这样一个向量Θ



通过我们刚才学的那个sigmoid函数，要想最后预测的结果为1就要让z>0也就是

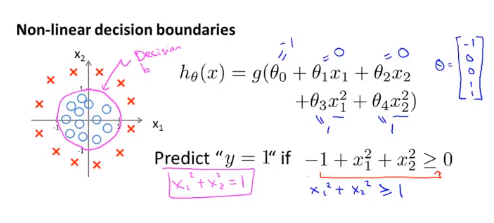
-3+x1+x2大于0

，而我们画出这一条直线，奇迹的发现这条直线就是决策边界，就是0，1的分界线



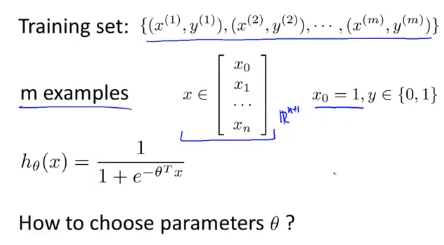
澄清一点的是决策边界是决策函数的一个假设，也就是说归属的

当然多项式决策的例子也是这个道理

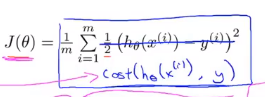


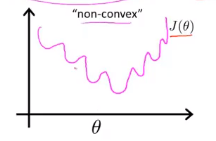
49 代价函数

自动选择一些参数，是我们在给定数据给定的情况下，自动拟合参数

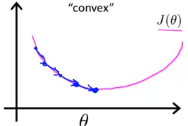


因为我们定义了sigmoid函数来限制y的预测值，所以当我们将代价韩式定义为原来线性回归的方式的时候



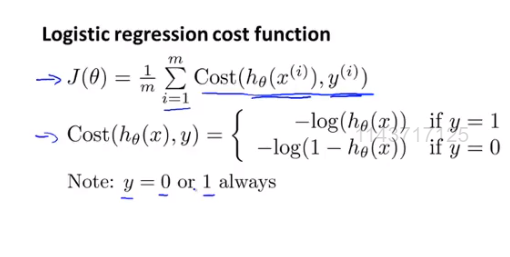
我们发现了这个hθ很明显画出的代价函数不会是以前那样只有一个拐点的，而是有很多个凹凸点不能保证局部收敛最小就是全局收敛最小

而 凸函数就是这样的

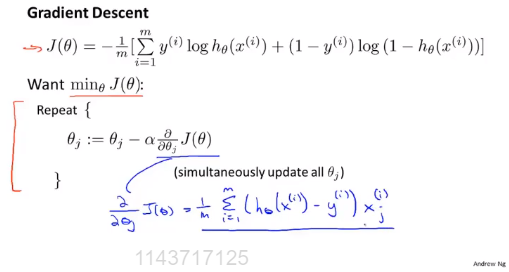


50.简化代价函数和梯度下降

这里的话是我们的整体代价函数







因为假设目标函数不一样所以求得函数也就不一样，也就是两个不同的东西