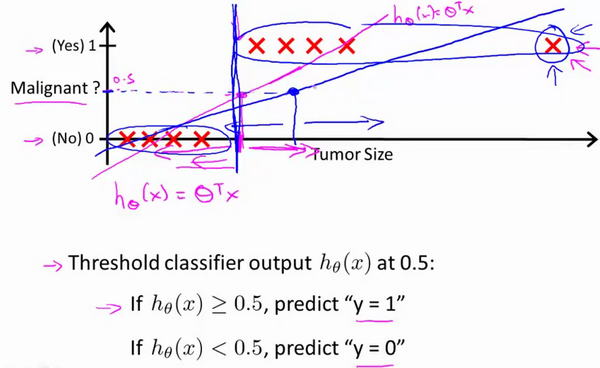
# 1.逻辑回归

## 1.1 分类问题

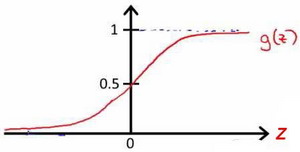
在本周，我们将要学习运用逻辑回归算法来处理分类问题。在分类问题中，我们要预测的变量y是离散的值。我们尝试预测结果属于哪一类，如判断一个邮件是否属于垃圾邮件，判断肿瘤是良性还是恶性。我们首先从二元分类的问题开始讨论。

如图所示，

我们将因变量可能归属的类别分为负向类和正向类，分别用0和1表示。如果我们想用线性回归处理该类问题，我们会发现，在一定情况下，函数的取值范围会不只是0和1.所以我们需要找到一个函数，取值范围永远是(0,1)。

## 1.2 假说表示

接下来我们开始寻找要用什么函数来表示我们的假设。由于线性函数的值域超过[0,1]，所以并不适合解决这类问题。由此，我们引入一个新模型，逻辑回归。逻辑回归模型的假设是： ，g代表逻辑函数，也叫做sigmoid函数。公式为： 。

如图所示即为sigmoid函数

的作用是，根据给定的样本参数x，计算出是正样本的概率。

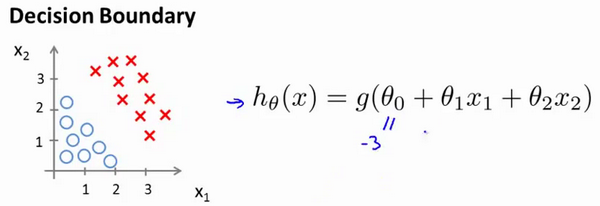
## 1.3 判定边界

现在我们讨论一下决策边界的概念，这个可以帮助我们明白假设函数在干些什么。



在逻辑回归中，我们预测,当g(z)>=0.5时，y=1,g(z)<0.5时，y=0

根据图像，我们可得，z=0时，g(z)=0.5,z<0时，g(z)<0.5,z>0时，g(z)>0.5

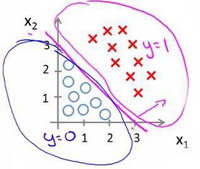
又,即可得

时，预测

时，预测

假设现在我们有一个如图所示的模型。

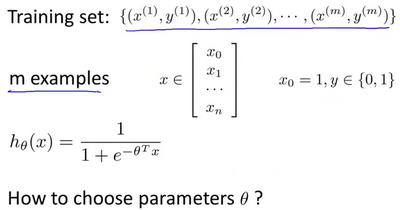
我们便有由此得到一条直线，将预测为1的区域和预测为0的区域隔开。



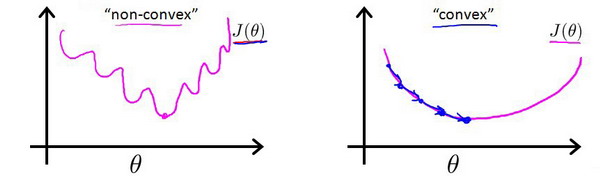
假设我们的样本呈现圆形状分布，我们也可以通过改变假设函数而成功进行分类。

## 1.4 代价函数

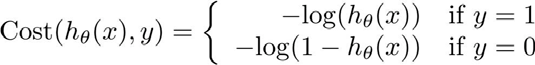
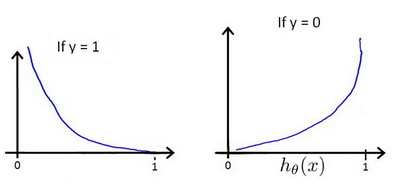
现在，我们的首要任务就是如何拟合参数θ，为此，我们需要引入代价函数。



首先，我们考虑使用线性回归时使用的平方误差函数，但事实上这是不行的，将假设函数带入，我们会发现所得到的图像有多个局部最优解，难以选择出最佳的参数θ。函数图像如图所示

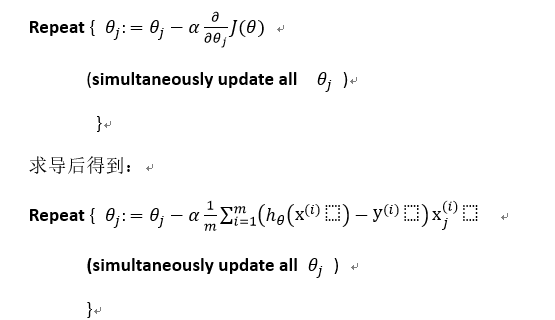


为此，我们便选择逻辑回归的代价函数为，

其中，由此我们可以绘制出对于单个样本的损失函数图像。

这样有个好处，当y等于1时，预测值也为1时，损失便为0，在y=0时也有类似结论。我们可以将这个分段函数合并为一个。即

得到这样的代价函数后，我们就可以使用梯度下降的算法寻找最优参数θ了。

算法为

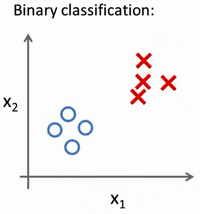
Ps:

除了梯度下降算法以外，我们也有一些其他算法也可以用来求最优的参数θ，这些算法更复杂也更优越，通常不需要人工选择学习率。这些算法有，共轭梯度，局部优化法，有限内存局部优化法。

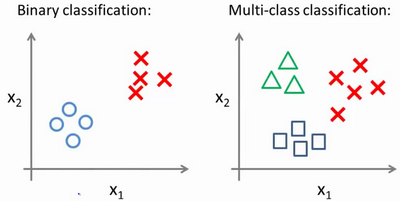
## 1.5 多类别分类 一对多

仍旧是以邮件分类问题为例，加入我们现在要设计一个算法来区分不同邮件的归属，比如将邮件分为四个类别，来自家人，朋友，同事，陌生人。

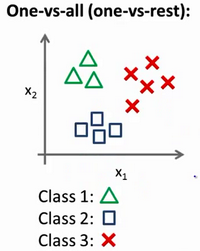
对于二分类问题，我们的数据看起来是这样的

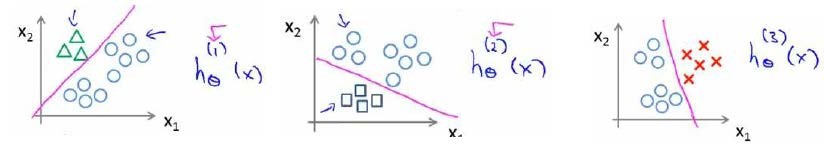


而对于多分类问题，我们的数据是这样



我们用三种符号来表示不同的类别，那么我们该如何区分这三个类别呢？我们将采用一个叫做(one vs rest)的方法



现在我们有如上图所示的数据集,我们要将其分为三类,我们实际上可以创建一个“伪”数据集，对于类别1，我们将2和3设为负类，而1为正类。对于类别二，我们将1和三设为负类，对于其他类别亦如此处理。同时，对于一个n>2的分类方案，我们要做出n个分类模型，如图所示

当我们有新的数据输入时，我们便能通过比较不同模型的输出值大小，输出值最大的那个模型的正类别，便是我们的新数据所归属的类。

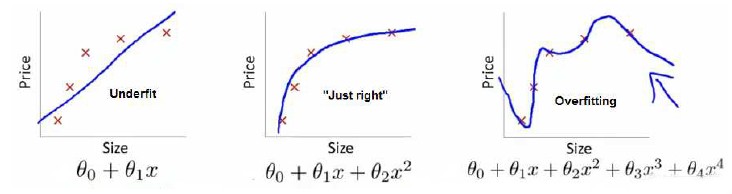
# 2.正则化

## 2.1过拟合问题

我们已经学习了线性回归算法和逻辑回归算法，它们能解决很多实际问题，但是遇到某些数据集时，却表现很差，这可能是遇到了过拟合问题。

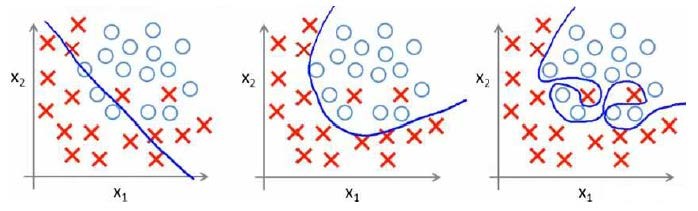
如果我们的数据集有非常多特征，那么我们通过学习到的假设函数可能能很好的拟合训练集，但是对于新的数据可能会有很差的表现。

下图是一个回归模型的例子



第一个模型是线性模型，欠拟合，第二个模型是二次函数，刚好能够拟合。而第四个模型是高次方程，过于强调拟合原始数据，而失去了算法的本质。若给出一个新值，则会给出误差很大的拟合结果。这个就是过拟合问题。

在分类问题中也有类似的情况



以多项式理解，就是次数越高，拟合的就越好，但是相应的预测能力就越差。

如果遇到了过拟合问题，我们该如何处理？两种办法，一是减少特征数量，而是使用正则化方法。

## 2.2 代价函数

上面的回归问题中，如果我们的假设函数是这样

我们从之前的案例中可以看出，正数高次项的存在导致了过拟合。如果我们能让这些高次项系数趋近于0的话，就能够很好的拟合了。

所以正则化在做的，就是在一定程度上减少θ的值。我们需要做的就是，在设定代价函数时，额外对参数θ增加惩罚。我们尝试修改后的代价函数是，

通过这样代价函数选择出的θ\_3和θ\_4，就会比最初的小很多，从而减少过拟合现象。

但是假如我们有非常多的特征，并不知道哪些特征导致了过拟合，我们此时就需要对所有参数进行正则化，得到这样的代价函数。

我们不对偏置项进行正则化。λ又称为正则化参数，λ过大，假设函数会变成一条直线。如图所示