**1、简介**

前面所用的对参数值的更新，无非梯度下降法和正规方程组法，然而，这只是最一般的计算的方法。事实证明，这样的算法受很多因素限制，譬如梯度下降法需要确定α学习率，正规方程法受特征值数量影响。因此，我们给出一个经优化后的梯度下降算法，本质上还是梯度下降法，不过将处理细节抽象了许多，一般编程人员直接调用即可。经检验，此方法在特征值很多时，仍能够快速而又精确的计算出值。

**2、调用方法**

该方法中，我们需提供的参数有初始值，迭代次数，返回计算代价函数和偏导数的方法，以及options配置参数，用于fminunc函数。

options = optimset('GradObj', 'on', 'MaxIter', 400);该式中，设置fminunc配置项，其中'GradObj'设置打开，表示需要计算出最终代价值，'MaxIter'为迭代次数，至于α学习率为何不用提供，这隐藏在了fminunc内部，事实表明，每次迭代的学习率，它本身自己会提供一个最理想的α，每次迭代α均不同，这就解释了它为何能够算的又快又准确。

[theta, cost] = fminunc(@(t)(costFunction(t, X, y)), initial\_theta, options)

这样一个式子就自动计算出了最终的值，关于costFunction的定义，如下：

function [J, grad] = costFunction(theta, X, y)

m = length(y);

J = 0;

grad = zeros(size(theta));

y\_pred = sigmoid(X\*theta);

J = -1.0 / m \* sum( y.\*log(y\_pred) + (1-y).\*log(1-y\_pred) );

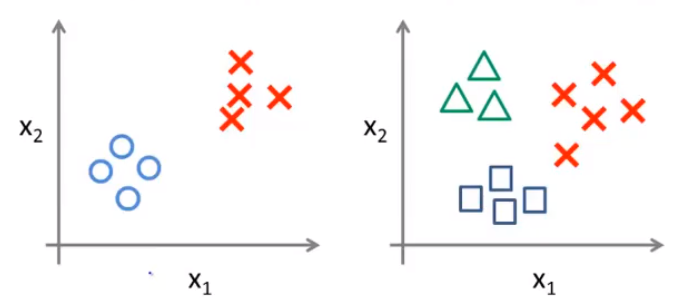
grad = 1.0 / m .\* (X' \* (y\_pred - y));

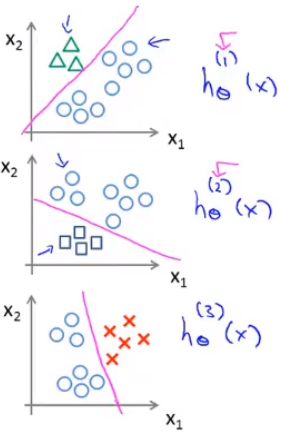
end

其中，J返回代价值，grad返回当前偏导向量，函数前的@表示函数指针。

**3、多分类问题**

加入你要根据花的特征，预测它是玫瑰花、喇叭花还是牵牛花，这是一个多分类问题，与之前不同的是，这次我们的类别看起来是这样的：

一对二、一对多问题

 根据之前所学的知识，我们无法同时预测出多于两个分组的分类问题。不过可以用逻辑分类的思想，既然无法同时预测3个，那么可不可以依次预测两类，每次只关心一个，由此引入了一对多逻辑分类算法。

这是算法过程，本例中，共有3个组别，每次将一个作为逻辑1，另外2个作为逻辑0，这样由之前的知识也能将输入分为两个组。

最后，将得到三个值，如果要预测某一个分类，需要算3次，仅有一个大于0.5，此即为它所对应的类别。