**代价函数-均方误差代数函数小结**

在机器学习中，线性回归、逻辑回归问题总是绕不开代价函数。本文将从代价函数的个人理解，均方误差代数函数作用原理以及在线性回归问题中为什么选择均方误差代数函数阐述。

1. 代价函数的理解：

真实的值与预测的值之间的偏差，由于偏差有可能正有可能负，因此使用均方差来表示。代价函数的目的即用于找到最优解。

**损失函数Loss Function**：定义在单个样本上，算的是一个样本的误差。

**代价函数Cost Function**：定义在整个训练集上，算的是所有样本的误差，也就是损失函数的平均。

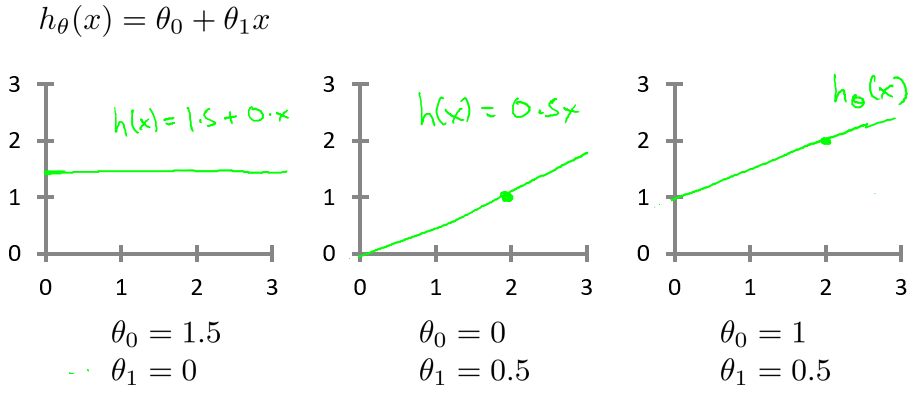
**目标函数Object Function**：定义为最终需要优化的函数。

Object Function = Cost Function（经验风险） + 正则化项（结构风险）

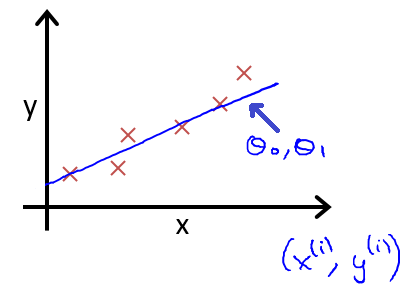
1. 均方误差代数函数作用原理：

对于回归问题，代价函数常选择均方误差函数。下面阐述其作用原理：

对于假设函数： 【h\_{\theta}(x)=\theta\_{0}+\theta\_{1} x】

参数和两个参数的变化会导致假设函数的变化：

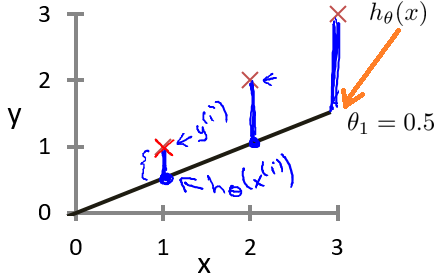
实际的例子中，数据会以很多点的形式给出，回归问题就是将这些点拟合成一条曲线，即找到最优的参数和使得拟合的曲线更能代表所有的数据。



如何找到最优参数和呢？使用代价函数！以均方误差代数函数为例：

从最简单的单一参数来看，假设函数为：

**均方误差代数函数**的主要思想就是将实际数据给出的值与拟合出的线的对应的值做差，求出拟合出的直线与实际的差距。

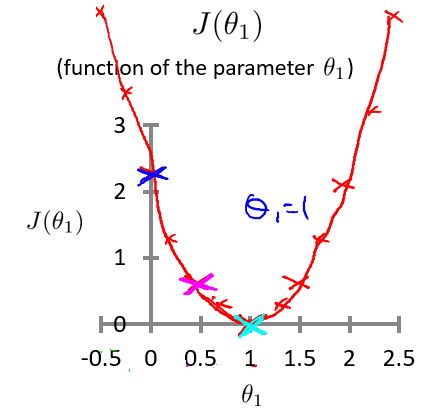
****

为了使这个值不受个别极端数据的影响而产生巨大波动，采用类似方差再取二分之一的方式来减小个别数据影响。

这样，就产生了代价函数：

J\left(\theta\_{0},\theta\_{1}\right)=\frac{1}{2m}\sum\_{i=1}^{m}\left(h\_{\theta}\left(x^{(i)}\right)-y^{(i)}\right)^{2}

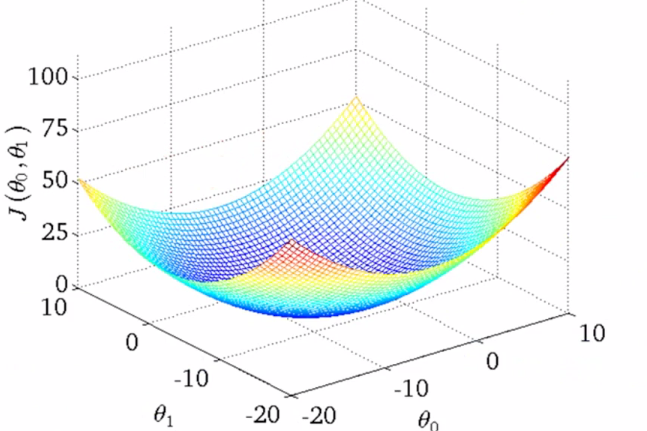
而最优解即为代价函数的最小值，根据以上公式多次计算可得到代价函数的图像：



可以轻易知道该代价函数的最小值，这时可以解出对应的

如果更多参数的话，图像就会更为复杂，两个参数的时候就已经是三维图像：

如何求解最小值？梯度下降算法！



高度即为均方误差代价函数的值，若参数再多则无法进行可视化，但其原理是相似的。

因此，我们可以终结得到：对于回归问题的代价函数，其最小值为：

3、在回归问题中为什么选择均方误差函数：

**从代价函数的定义考虑**：代价是预测值和实际值之间的差距（两点之差），对于多个样本来说，就是差距之和。

将每一个样本的预测值和实际值之差相加，这不就是代价？但是注意到就每一个样本而言，代价有正有负，所有样本的代价加起来有可能出现正负抵消的情况，所以这并不是一个合适的代价函数（指将每一个样本的预测值和实际值之差相加）。

**解决有正负的问题：**

使用绝对值函数来表示代价，为了方便计算最小代价（计算最小代价可能用到最小二乘法），直接使用平方来衡量代价，即使用绝对值的平方来表示单个样本的代价，那么一个数据集的代价为：

**是不是使用平方之和就没有什么问题了？**

肯定不是！代价函数应该与样本的数量有关，否则一个样本和m个样本的差距平方和之间的比较也没有多少意义，所以将 m 个样本的代价之和乘以 1/2m，即代价函数为：

为什么分母取2m而非m，是为了计算方便。