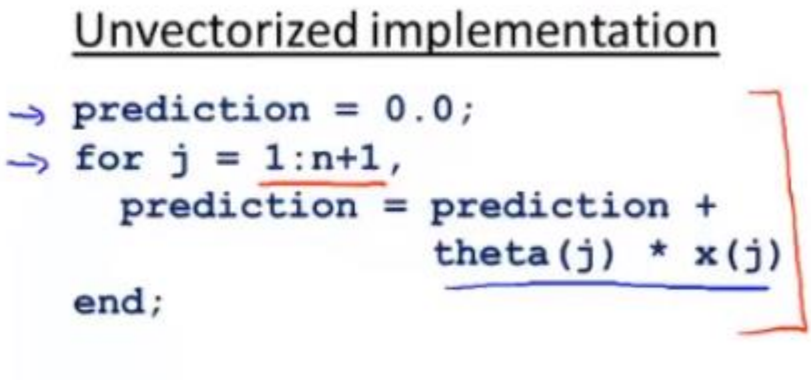
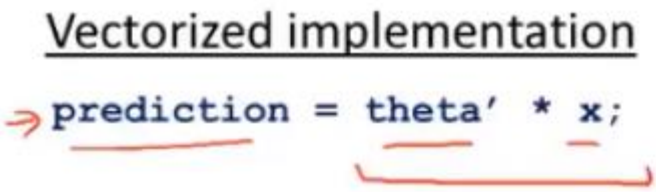
1、一般思路

当我们在求预测值的时候，往往输入部分含有多个特征值，比如$x_0x_1x_2x$ …，而参数值也有多个$\theta_0\theta_1\theta_2\theta$ …，我们利用$h_\theta(x)=\theta_0x_0+\theta_1x_1+\theta_2x_2+\cdots$可能会选择利用 for 获知 while 循环，然而 Octave 提供了向量化的运算思想，可以利用$X^T*\theta$快速得到结果，这比我使用 for 循环不仅省时，而且由于科学计算器件对矩阵运算的强大支持，使得其运算速度比单纯的 for 循环快上好几倍！以下两种计算过程展现出了它们的区别。



for 运算



向量化运算

2、梯度下降算法中矩阵运算的应用

A、计算代价差

代价差计算简化公式，cost = $\frac{1}{2m}sum((X\cdot\theta-y).\hat{}2)$

B、参数值的更新

以往我们同步更新参数值，可能得像这样

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_1^{(i)}$$

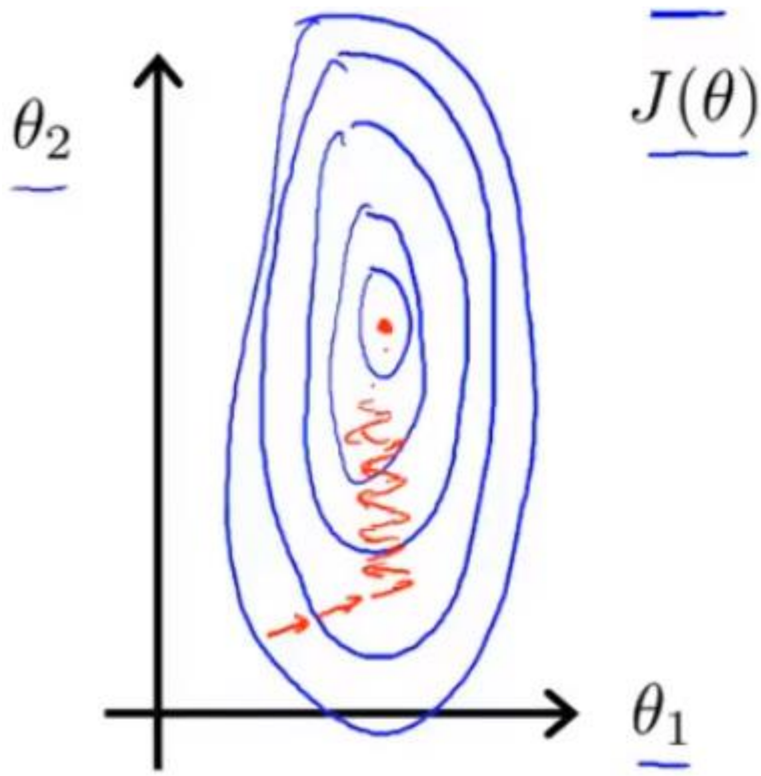
$$\theta_2 := \theta_2 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_2^{(i)}$$

运用向量化的方法，我们可以使得上述过程简化为如下式子$\theta = \theta - \alpha\delta$，

$\delta = \frac{1}{m}\sum_{i=1}^{m}(\hbar_\theta(x^{(i)}) - y^{(i)})x^{(i)}$，这样一来，我们便可使用这样一个简单的式子迅速更新所有的参数值了。观察式子，$\theta = \theta - \alpha\frac{1}{m}X' * (y_{pred} - y)$成为了我们最终想要的式子，与原来相比，新的式子更加整洁，但过于抽象，会使第一次看到它的人思考半天。

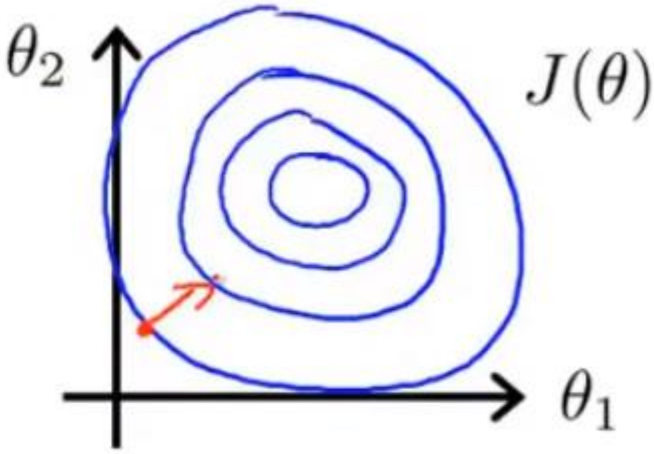
3、特征值缩放

举个例子，来看有两个参数值的预测函数，根据房屋面积、卧室数量预测房价，假如房屋面积为 1~2000 ㎡，房屋数量为 1~5，那么此时画出来的代价等值线可能是这样的，图像随θ_1变化而剧烈变化，它的一点改动，就可以使 J 值变化巨大，因此在梯度下降算法中，θ_1的值变化也将最剧烈，因为它有更大的偏导数，所以整个图像收敛过



程看起来就像图上这样，十分缓慢。

因此，引入了特征值缩放这一概念，上例中，房屋面积为 1~2000 ㎡，房屋数量为 1~5，我们将它分别除以 2000 和 5，这样一来，图像可能会变成这样，不过，不用担



心我们对特征值的改变，它不会影响将来对待测数据的预测，只是将整个拟合过程变得快速合理许多。

4、均值归一化

对一特征值，若$\bar{x} = 1000$， $\max(x) - \min(x) = 500$，则令$X_i = \frac{x_i - 1000}{500}\,(i = 1{:}m)$，分母部分可以换成标准差或是方差，不一定是极大值与极小值的差。