# 多变量线性回归

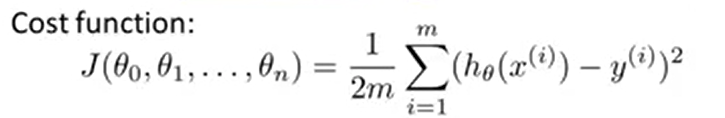
多变量线性回归算法思想与单变量线性回归算法思想基本一致，（1）先找到损失函数，（2）求损失函数最小化后的参数；本文还简单介绍了特征缩放，正则方程以及矩阵不可逆的情况

### 损失函数

多变量的线性回归的目标函数（考虑双变量）：



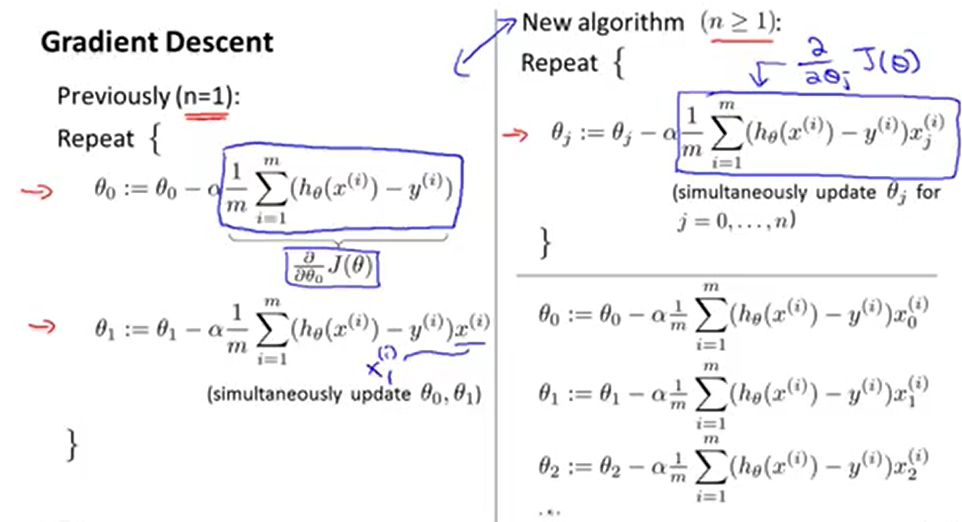
对应的损失函数：



### 损失函数的最小化算法

对训练样本构建模型也就是最小化损失函数，方法有前面提到的梯度下降算法和即将介绍的正则方程。

## 2.1 梯度下降算法



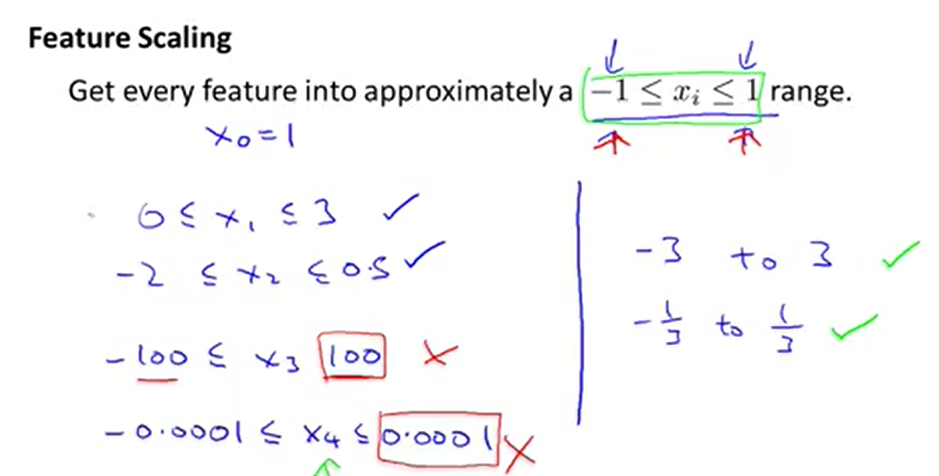
#### 2.1.1 特征缩放

如果特征都是在一个相近的范围，那么梯度下降算法能更快地收敛。如下图：

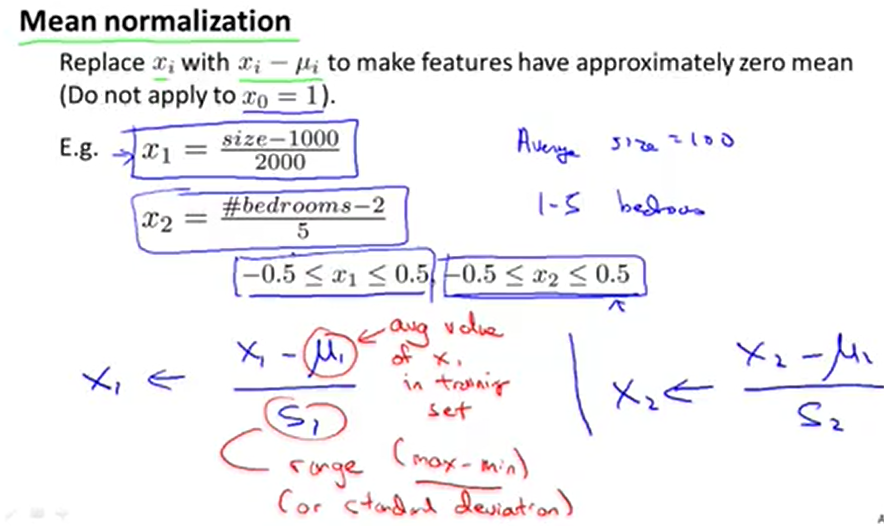
### 

当特征范围不是一个级数的时候，损失函数的曲线偏向于瘦长形，导致梯度下降算法收敛速度较慢，如果对每个特征都除以特征的范围，限制特征范围在0~1区间，则损失函数的曲线接近于圆，提高了收敛速度。

特征大致是【-1,1】的范围，若上下界范围与上下界相差不大，也是可行的。若范围非常接近，如【-0.0001,0.0001】，那么也需要经过特征缩放。

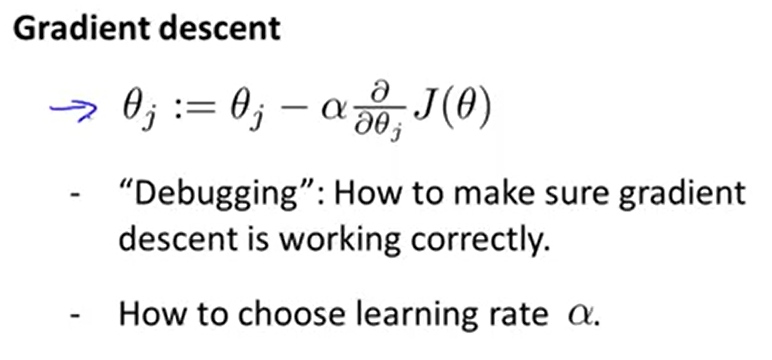


**均值归一化：**特征去均值的目的是使特征变量的均值为零（可参考概率论与数理统计一书），均值归一化等于特征去均值后再除以特征的范围，则特征的范围限制在【-0.5,0.5】附近，范围亦可用标准差来代替，如下图。

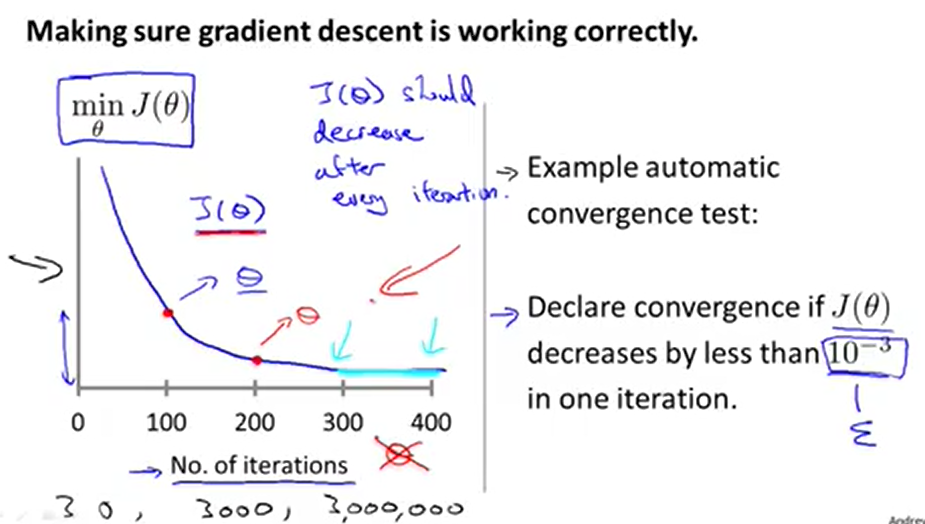


### 2.1.2 学习率α

本节内容包括如何确定梯度下降算法是正常工作的以及怎么选择学习速率α。

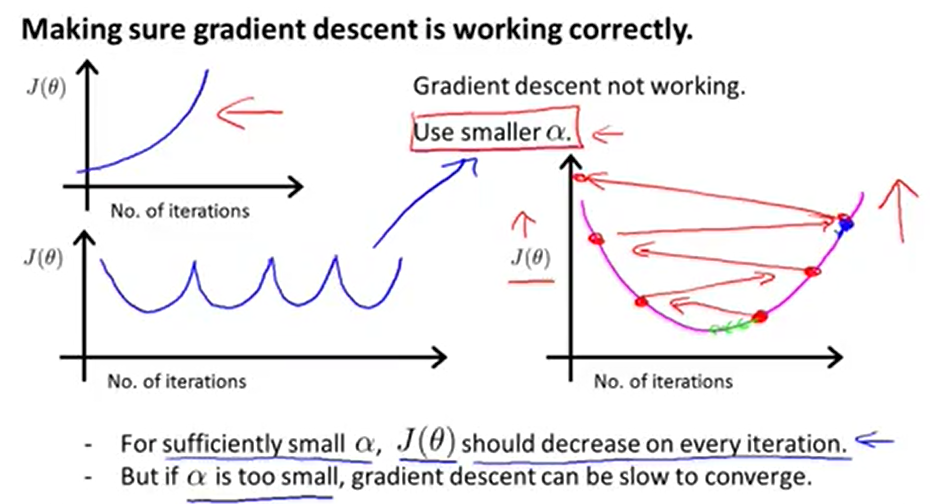


通过迭代次数来检查梯度下降算法是否正确，理论上损失函数会随着迭代次数的增加而减小，如下图，迭代次数300与400的损失函数下降不明显了，当迭代次数大于400时，可以看到损失函数近似于一条直线，那么对损失函数



只迭代400次，就满足损失函数最小化条件，迭代次数依实际情况而定，有些迭代次数小就能满足收敛条件，有些可能需要很大的迭代次数才能满足收敛条件。如果本次迭代损失函数下降的程度小于上次迭代损失函数的1e-3，那么就认为损失函数亦收敛，通常下降阈值不好给定，因此，最好的办法还是通过看迭代次数与损失函数曲线来判断。

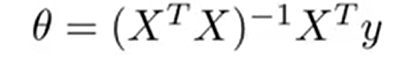
下图是梯度下降算法没有正常工作的情况，可以通过减小α的值来解决（个人认为最右边的图有一点点小错误，横坐标应该是参数，而不是迭代次数）



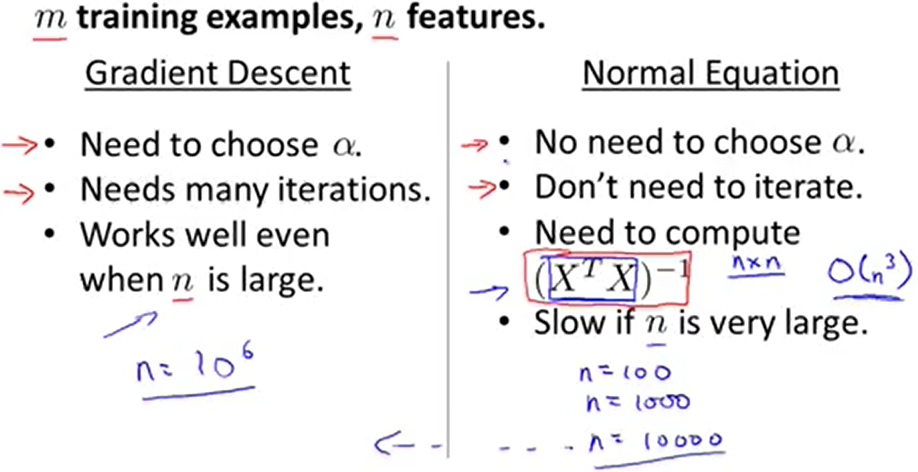
对于足够小的α，损失函数应该在每一次迭代的情况下都减小，但是如果α过小，那么梯度下降算法收敛速度会变慢。通过设置α的数值并对每一数值画出迭代次数与损失函数的曲线来确定学习率大小，Andrew老师设置了0.003，0.03，…，1，找到能满足收敛条件对应的最大α值。

## 2.2 正则方程

梯度下降算法需要经过多次的迭代才能收敛到最小值，正规方程提供了一种求θ的解析方法。其思想是损失函数对每个参数进行偏微分，并令偏微分为零，求得的θ即是损失函数最小化后的值。公式如下：



**正规方程与梯度下降算法的区别：**

****

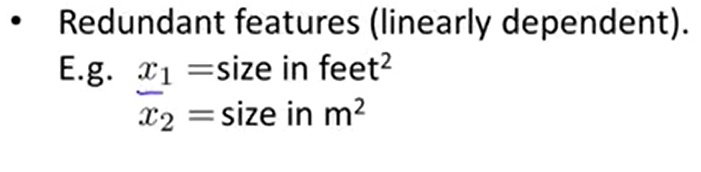
正规方程相比较于梯度下降算法不需要设置α和迭代，但是正规方程的求逆矩阵的计算量与特征的立方大致相当，当特征值特别大时，则选择梯度下降算法。后续课程的分类算法较复杂，求损失函数最小化过程中用到了梯度下降算法，正规方程已经不适合该情况，但是在特定的线性回归问题上，正规方程是更快的求解最优参数的算法。

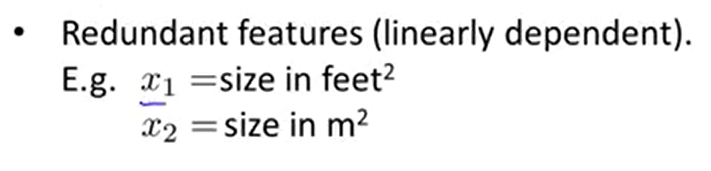
## 三、正规方程的不可逆性

正规方程包含了矩阵的求逆过程，若矩阵出现不可逆的情况，该怎么处理。Octave有两个求逆的函数：pinv和inv，其中pinv是求矩阵的准逆，当遇到不可逆的情况也能计算结果。

**矩阵出现不可逆情况：**

1. 特征线性相关，特征有可能描述的是同一种属性。





（二）、特征数太多，大于样本数。可以通过删除一些无关特征或使用正则化损失函数。

