# 机器学习笔记

## 前言：

吴恩达的机器学习及深度学习两门课，github上有人做了详尽的笔记

<https://github.com/fengdu78/Coursera-ML-AndrewNg-Notes>

因为有了完整的笔记，这里笔者就不打算记录完整的笔记，而是记录一些对于笔者而言重要的笔记，所以对大家而言就没有什么参考价值了。

## 一、引言

无监督学习

无监督学习中的数据没有任何标签，我们仅有一个数据集，不知如何处理。无监督学习能判断数据中的聚集簇，从而将数据分类，这称之为聚类算法（cluster）。

## 二、单变量线性回归

如果以y = ax + b这样的线性方程来拟合数据，并且以mse（均方误差）为代价函数来计算，那么一下是推导过程。

方程



代价函数

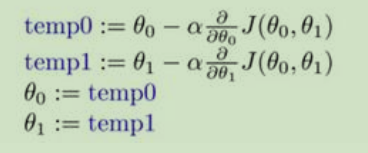


目标：代价函数最小

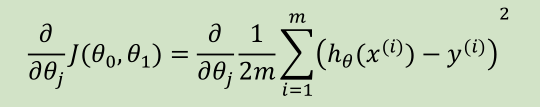
方法：通过更改theta1和theta2的值，使得代价函数降低。而最快的方法则是沿着梯度的方向改变值。Alpha是学习率，那么theta的更新函数如下所示

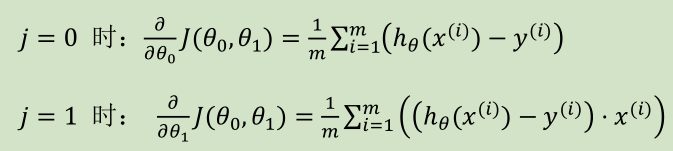


更具体一点，计算过程应该如下：

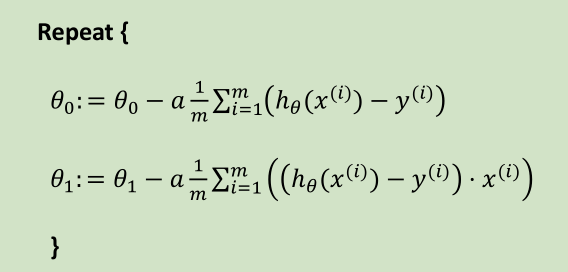


关于偏导数的公式推导：





那算法改写如下：



## 三、线性代数回顾

### 规律

矩阵乘法不满足交换律



矩阵乘法满足结合律



单位矩阵，除对角线元素为1，其余元素都为0的矩阵。一般用I或者E表示。对于单位矩阵，有

AI = IA = A

### 逆矩阵

只有方阵才有逆矩阵。

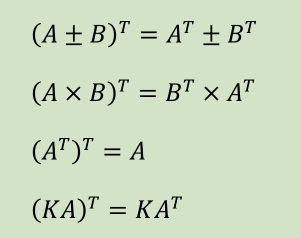
如果有逆矩阵，则

没有逆矩阵的矩阵成为奇异矩阵(Singular)或者退化矩阵(degenerate)

Numpy中求矩阵的额逆的方法

np.linalg.inv()

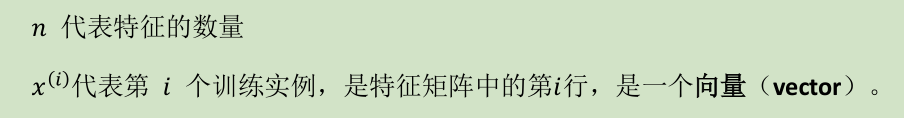
### 矩阵转置

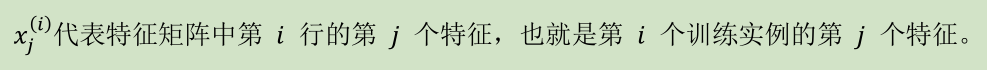


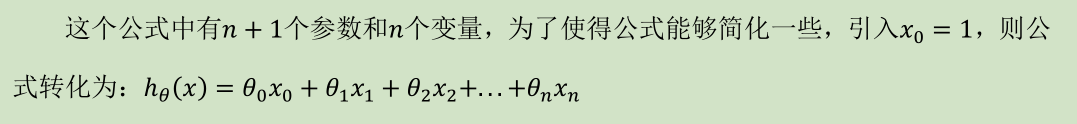
## 四、多变量线性回归

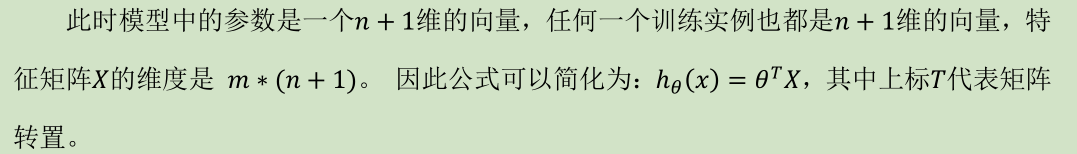
### 符号含义

有了多维特征后，公式推导的符号含义：









### 梯度下降

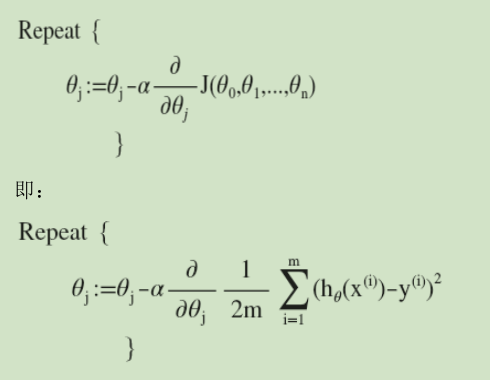
#### 公式



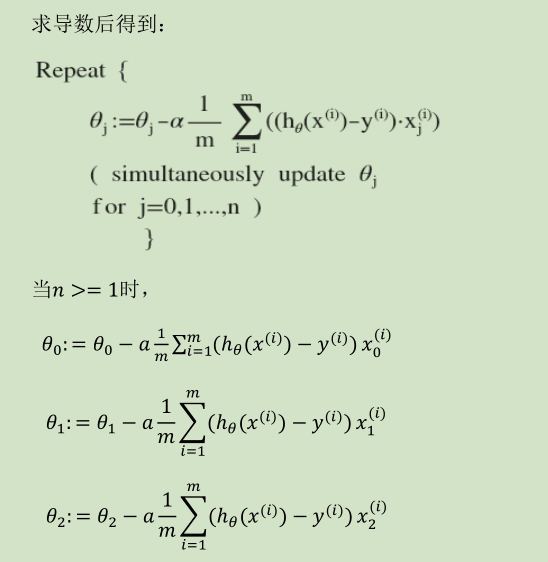
#### 代价函数：



梯度下降算法为：



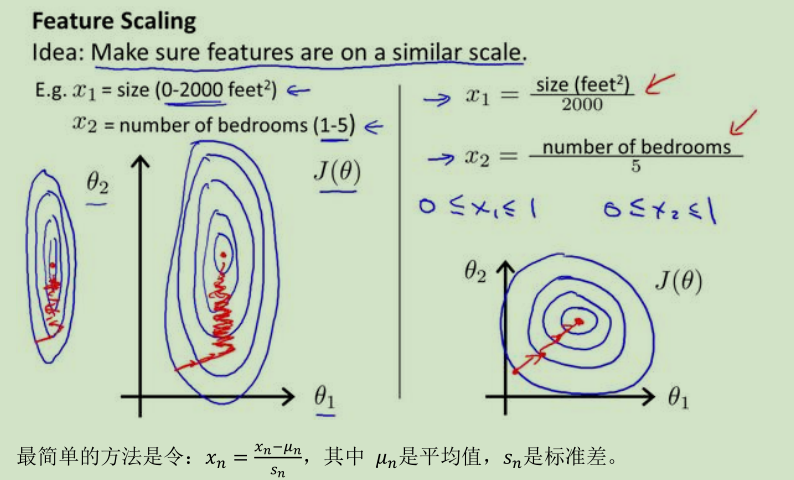
计算后得到



### 特征缩放

在我们面对多维特征问题的时候，我们要保证这些特征都具有相近的尺度，这将帮助梯度下降算法更快地收敛。

以房价问题为例，假设我们使用两个特征，房屋的尺寸和房间的数量，尺寸的值为 0-2000 平方英尺，而房间数量的值则是 0-5，以两个参数分别为横纵坐标，绘制代价函数的等高线图能，看出图像会显得很扁，梯度下降算法需要非常多次的迭代才能收敛。

解决的方法是尝试将所有特征的尺度都缩放到一个大致相等的区间之间。如图：

### 学习率

学习率也考虑多测试几个值，Andrew推荐的序列是以3倍递增：

0.001，0.003，0.01，0.03，0.1……

### 多项式回归模型

线性模型有时候并不能满足我们的需要，这时候可以考虑别的曲线来拟合数据，比如二次方模型、三次方模型v





或者：





多项式回归模型的话，特征缩放非常有必要

### 正规方程

梯度下降算法的原理是找到损失函数的最低点；从数学上讲，这个点就是该点梯度为0的地方，而正规方程就是一步到位，求出损失函数梯度为0 的点。

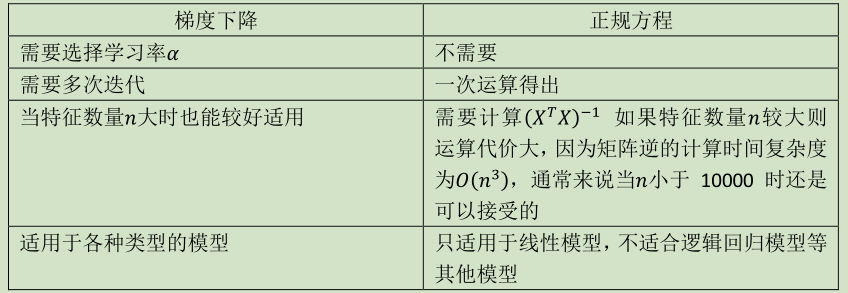
求解公式为：

θ = (XTX)-1XTy

矩阵不可逆通常有2种情况引起：

* 使用了线性相关的特征，比如同时使用m和feet
* 使用了太多特征，比如想用10个数据算出100个特征。

#### 比较



#### 结论

只要特征变量的数目并不大，标准方程是一个很好的计算参数的替代方法。

具体地说，只要特征变量数量小于一万，我通常使用标准方程法，而不使用梯度下降法。

随着我们要讲的学习算法越来越复杂，例如，当我们讲到分类算法，像逻辑回归算法只要特征变量的数目并不大，标准方程是一个很好的计算参数𝜄的替代方法。对于那些更复杂的学习算法，

我们将不得不仍然使用梯度下降法。

正规方程的 python 实现：

import numpy as np

def normalEqn(X, y):

theta = np.linalg.inv([X.T@X)@X.T@y](mailto:X.T@X)@X.T@y) #X.T@X 等价于 X.T.dot(X)

return theta