# 赛题解析 baseline逐行精读

## 赛题解析

## 1.1 初赛

### 1.1.1数据集

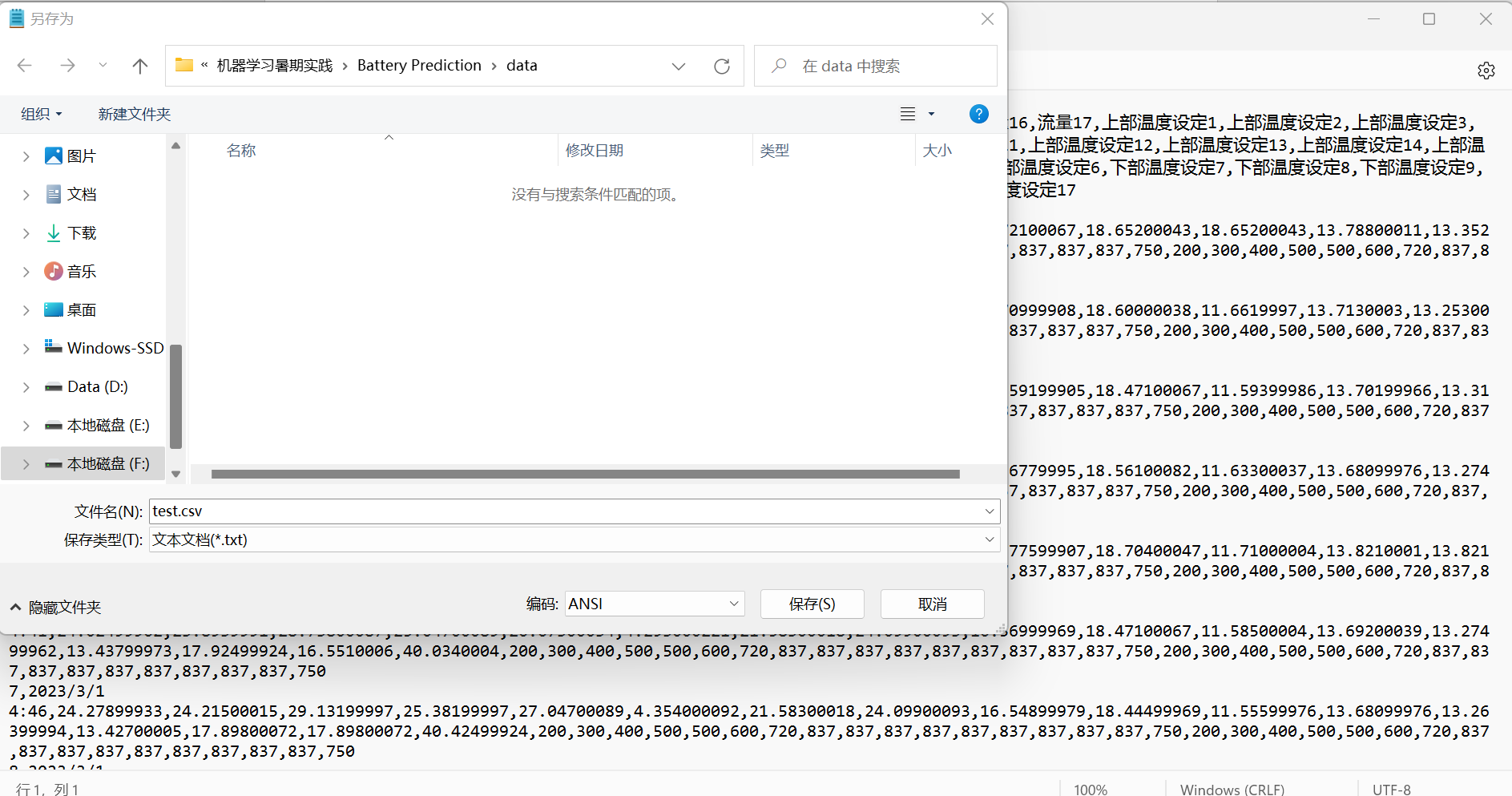
电炉上/下部各17组加热棒的设定温度

电炉底部17组进气口的设定进气流量

#### 1.1.2任务:

根据提供的数据样本构建模型，预测电路上下部空间17个测温点的测量温度值

#### 1.1.3excel查看csv数据中文乱码解决方案：

打开记事本发现数据集默认为utf-8编码，另存为ANSI编码的csv文件即可用excel正常打开

## 1.2 复赛

加入考虑因素：内部温度指标、测量值与稳定值之间的滞后的时间关系

## 机器学习理论知识

### 2.1 机器学习基础概念

#### 2.1.1什么是机器学习(Machine learning)

Arthur Samuel(1959):在没有明确设置下，使计算机具有学习能力

Tom Mitchell(1998):计算机从经验E学习，解决某一任务T，进行某一性能度量P，通过P测定在T上的表现因经验E而提高

监督学习: 教计算机学习

无监督学习：让计算机自己学习

#### 2.1.2 监督学习(Supervised learning)

对于数据集的每个样本（每个样本已经被明确告知什么是所谓的正确答案），想要算法预测并得出“正确答案”，根据预测值的分为以下两类:

Regression：预测一个连续的输出值

Classification:预测一个离散的输出

#### 2.1.3 无监督学习(Unsupervised learning)

对于数据集的每个样本（每个样本均未被告知所谓的正确答案），想要算法将不同的样本分簇聚合，典型的是聚类

### 2.2 简单线性回归及多元线性回归模型（监督学习）

#### 2.2.1 模型描述

对于训练集得到一个hypothesis假设函数（即模型）进行预测分析，最简单的是线性回归模型

其中表示参数Parameter

#### 2.2.2 代价函数

目标:

其中y代表真实值,m代表训练集样本容量,i代表第几个样本

令因此被称为平方误差代价函数

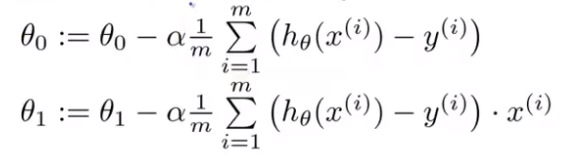
#### 2.2.3 梯度下降法(Gradient Descent)—目的是找到最小化代价函数的参数

初始化参数值

不断同步更新所有参数值直到找到J的最小值 这里“:=”表示赋值

 称为学习率(learning rate)，控制以多大幅度更新参数，过小会导致收敛慢，过大会导致无法收敛甚至发散，求导项(derivative term)决定了当我们接近局部最优点的时候，梯度下降法会自动采取更小的幅度（因此没有必要对学习率进行减小）

#### 2.2.4 线性回归的梯度下降-用于求解只有一个全局最优的凸函数最有效



Batch Gradient Descent:每一步梯度下降都用到全部的训练集样本

#### 2.2.5 线性代数基础知识回顾

（大写字母）表示4行3列的矩阵 表示具体某个元素

向量（小写字母）：仅有一列的矩阵（默认为列向量） 代表列向量 表示某个元素，小写字母也可用作表示标量

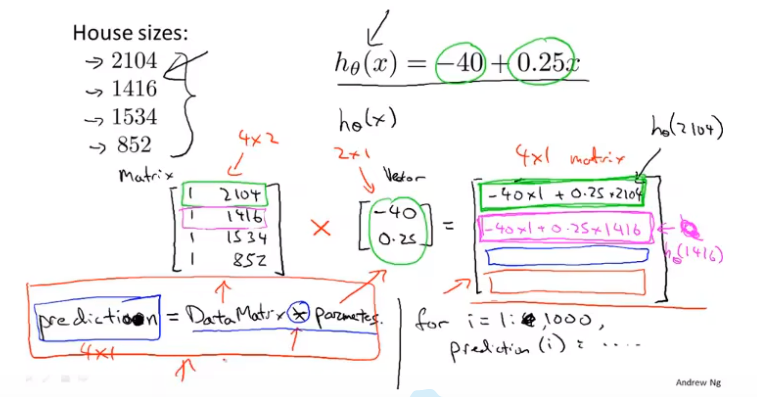
矩阵加法：对应元素相加

矩阵减法：对应元素相减

标量乘法：矩阵元素均与标量相乘（相除类似）

矩阵相乘：结果矩阵的元素是第一个矩阵的所有i行元素和第二个矩阵的所有j列元素对应相乘求和的结果

编程技巧：Prediction = DataMatrix \* Parameters 这种利用矩阵的求法求所有预测值要比循环节省时间



矩阵乘法的特征：不满足交换律，满足结合律

单位矩阵(Identity Matrix):

矩阵的逆(Matrix Inverse): 如果矩阵A是方阵(square matrix)并且它可以求逆，利用Matlab的pinv（可求伪逆矩阵）/inv可以很好的求逆矩阵，不存在逆矩阵的矩阵称为奇异矩阵(singular matrix)或退化矩阵(degenerate)

矩阵的转置(Matrix Transpose): 

#### 2.2.6 多元线性回归(Multivariate linear regression)

1. 表示训练集的第i个样本的第j个特征



1. 初始化参数值
2. 不断同步更新所有参数值直到找到J的最小值 这里“:=”表示赋值

 称为学习率(learning rate)，确保不同特征在一个相近的范围，这样梯度下降法才能更快收敛。通过看loss曲线图可以看出是否梯度下降正确收敛，进而确定学习率是否过大（loss曲线有上升就代表过大），一般从0.001,0.003,0.01，0.03,0.1,0.3,1慢慢确定学习率

1. 特征缩放 (Feature Scaling):将每个特征值约束到-1~+1(-3~+3,-1/3~+1/3都可以接受)
2. 均值归一化(Mean normalization): 为训练集特征的平均值，为训练集特征的最大值减去最小值也可以用特征的标准差
3. 自己根据问题构造特征，模型也可换用多项式回归模型,即对某些特征采用平方项、三次方项

#### 2.2.7 正规方程方法（与梯度下降法同作用求最小值）

**使用正规方程法不需要对特征进行缩放**

构造设计变量矩阵

给出最优参数值 Matlab编程:pinv(X’\*X)\*X’\*y

维度太大的时候不适用，1000以下个特征基本可用正规方程，10000考虑梯度下降

不可逆的原因：

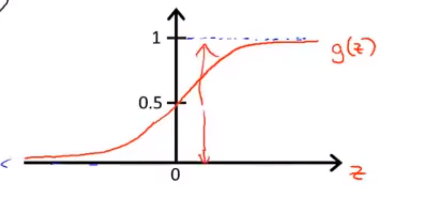
1.特征之间存在线性依赖导致矩阵的非满秩

2.样本数小于特征数（删除特征或者正则化）

### 2.3 Logistic回归模型（监督学习）

#### 2.3.1 二分类问题

1. 预测值y=0表示Negative class ;y=1表示Positive class
2. Logistic回归模型(分类算法)



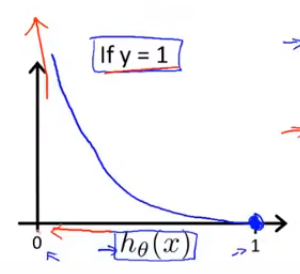
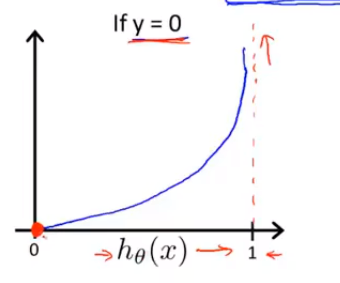
预计的是输入特征对应的预测值为1的概率



称为决策边界(decision boundary)

1. 传统的平方误差代价函数在分类问题中导致非凸函数（Non-convex Function）,因此构造新的代价函数





1. 参数更新的形式同多元线性回归一致，不同的只是代价函数的定义方式，并且也适用特征缩放
2. 高级优化求解参数最小值方法,不需要设计学习率并且收敛更快，Matlab fminunc可以调用

Conjugate gradient

BFGS

L-BFGS

#### 2.3.2多分类问题拓展

把多分类转换为多个一对余分类问题，根据类别数训练对应的模型

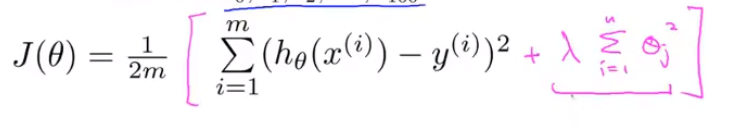
对每个种类i预测y=i的概率，作出最终预测时则对每个模型预测选出概率最大的值，确定为该类

#### 2.3.3 过拟合问题

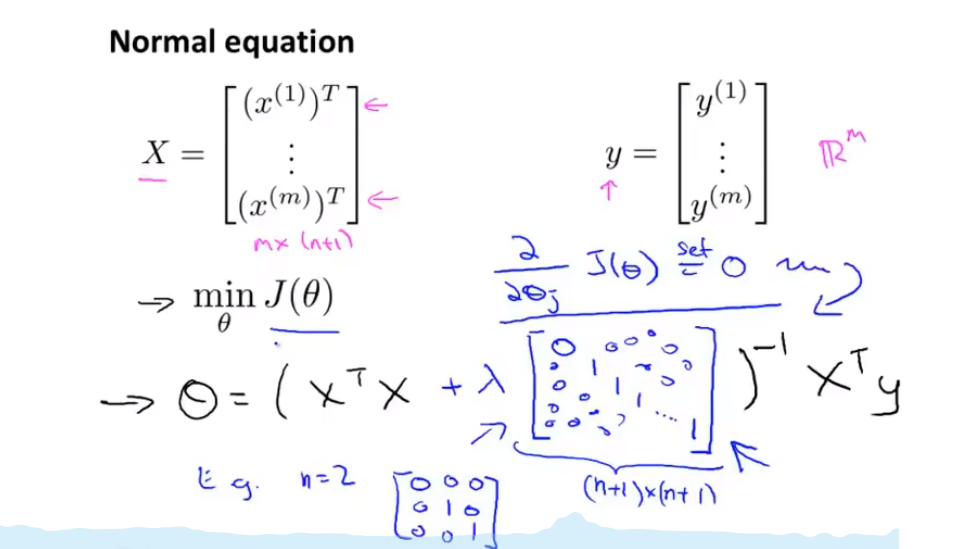
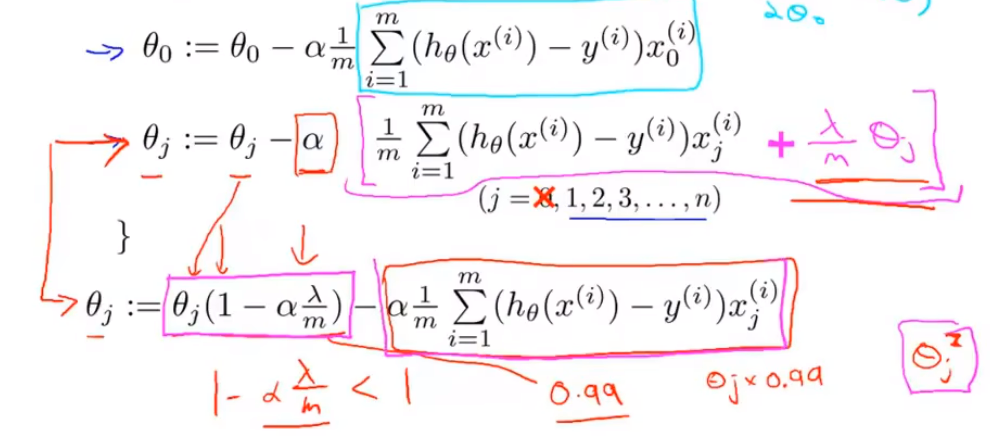
1. 模型预测值与实际值存在大偏差，称为欠拟合(underfit)
2. 模型预测值与实际值存在大方差，称为过拟合(overfit),有太多特征时容易出现，过度拟合了测试集的特征，缺少泛化性
3. 解决方案：

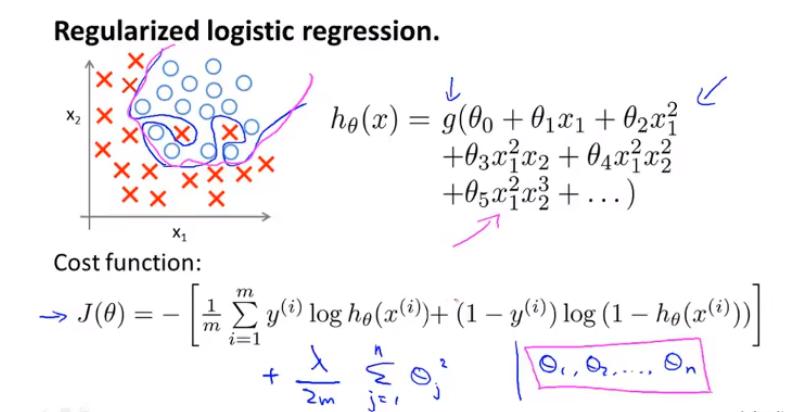
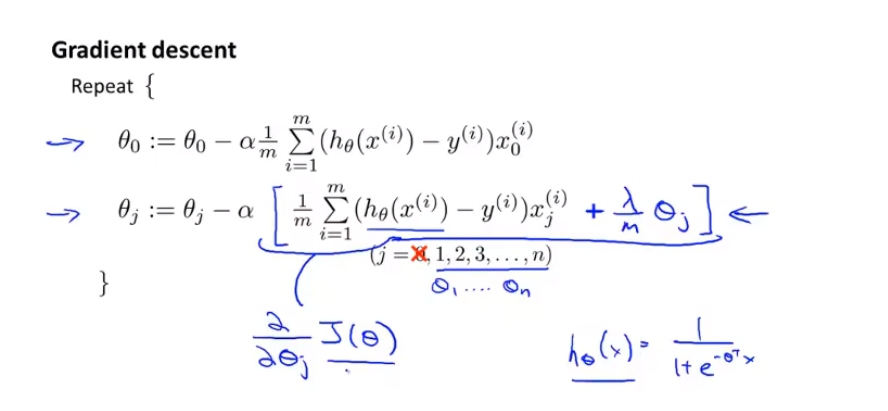
减少特征变量（手动选择、模型选择）

正则化(Regularization):在价值函数中增加（一般不含）



过大，参数乘法力度过大会导致欠拟合

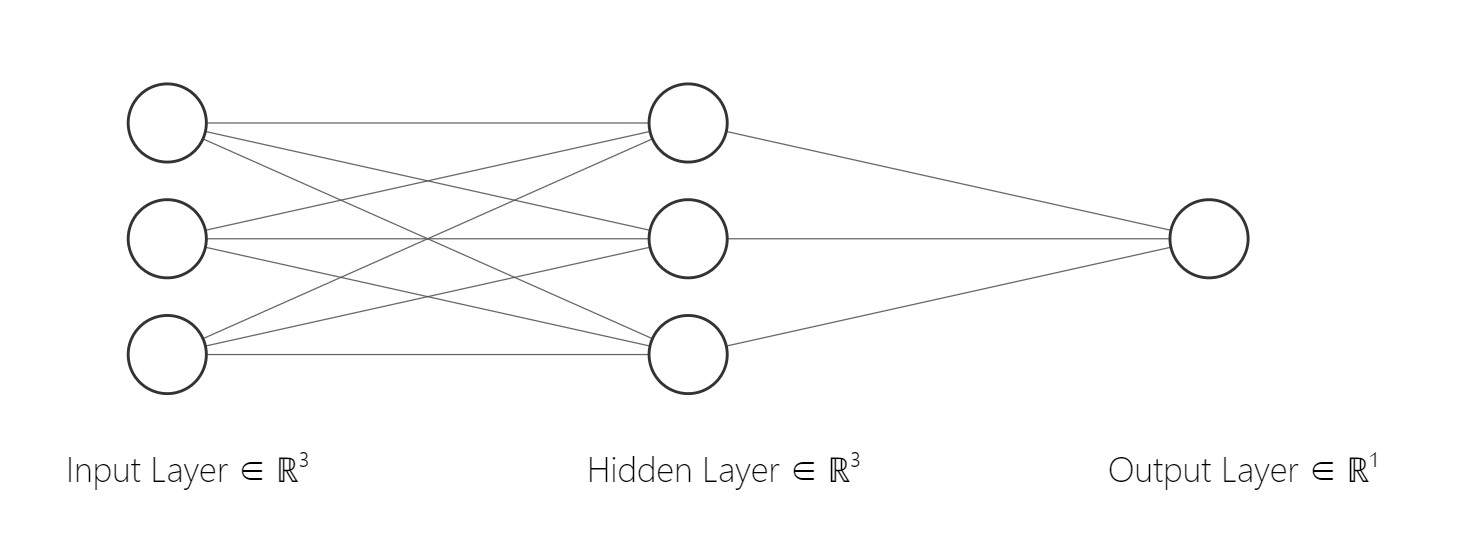
1. 线性回归的正则化梯度下降有如下改变：对于梯度下降相当于是缩小了参数更新的初始值，对于正规方程保证了矩阵的可逆性
2. 逻辑回归的正则化



### 2.4 神经网络（Neural Network）（监督学习）

#### 2.4.1神经网络的模型

1. 当特征个数过多时传统的线性回归和logistics回归都不再能拟合更好的模型（图像处理像素极多），此时需要非线性模型
2. 神经元：即一个logistic单元，输入为特征，输出为预测值，网络中第一层为输入层(input layer)，第三层为输出层(output layer)，第二层被称为隐藏层(hidden layer),x0=1



hθ(x)

X3

X2

X1

a3(2)

a2(2)

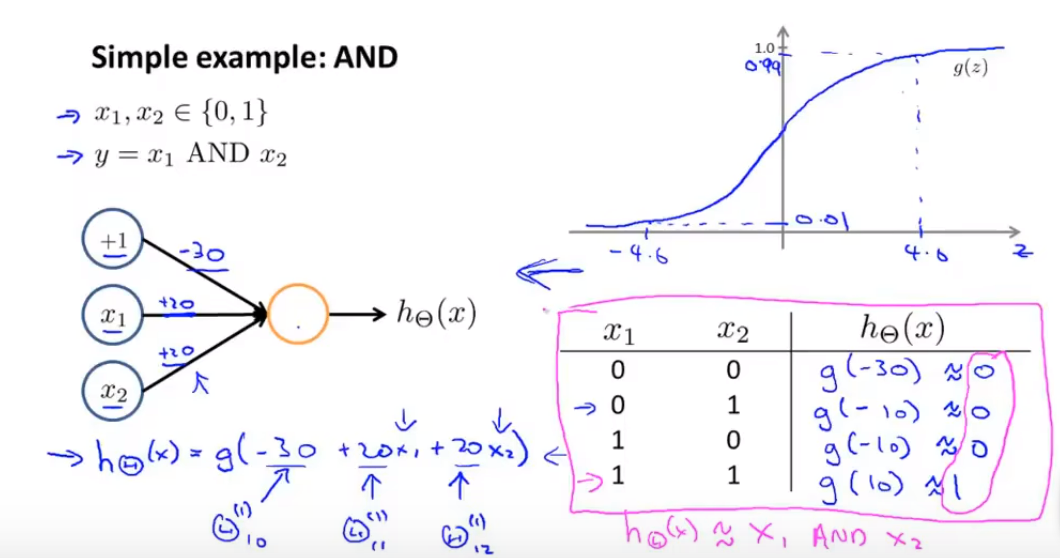
a1(2)

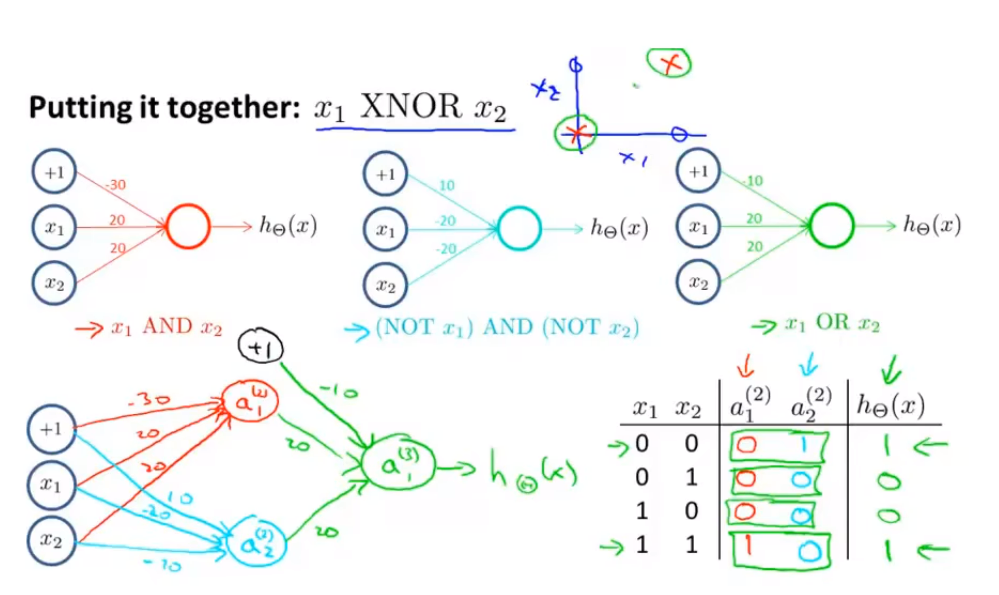
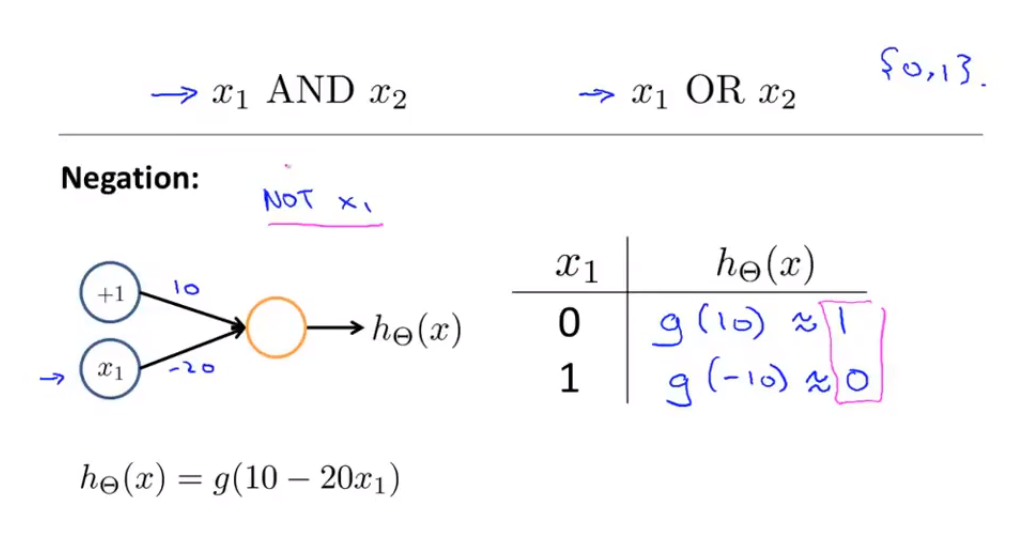
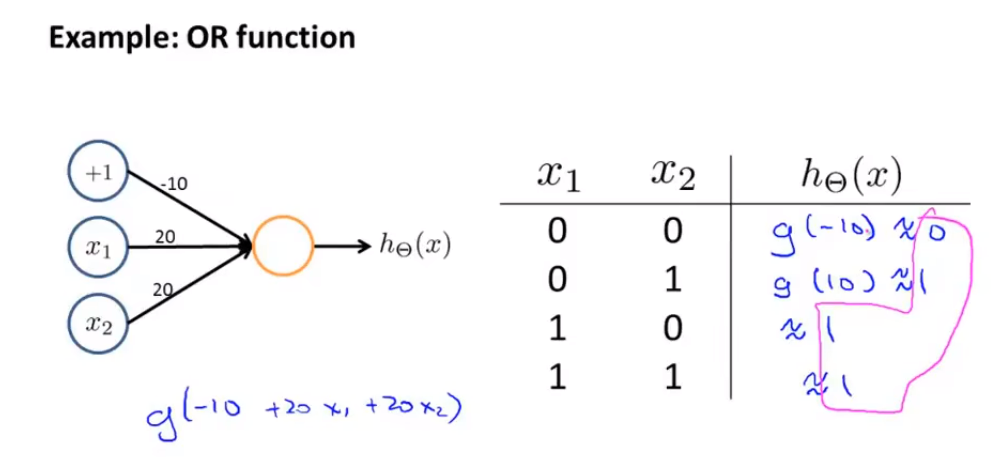


1. 神经网络向量化形式(前向传播 forward propagation):

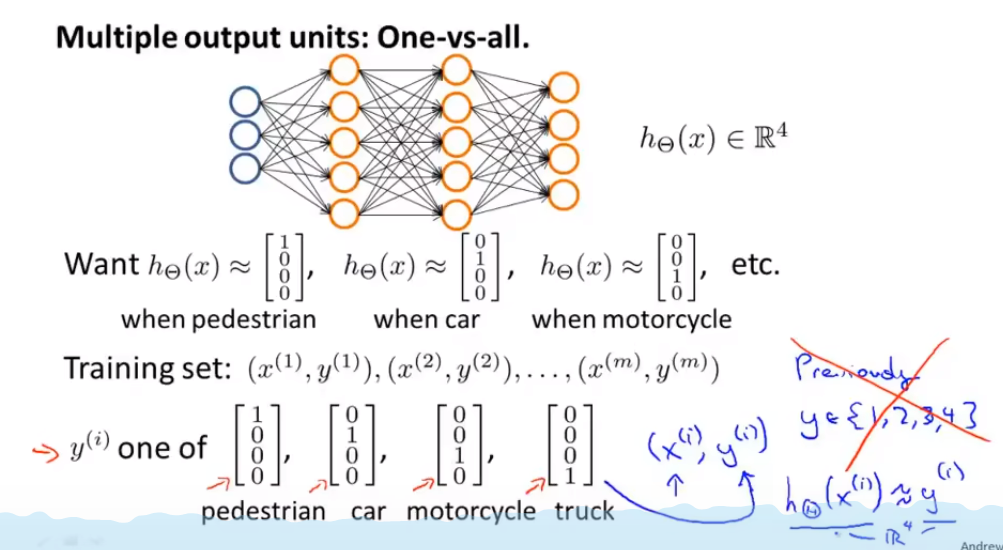


1. 神经网络的应用举例：代替逻辑异或(XOR)、逻辑与(AND)、逻辑或(OR)





解决多分类问题



#### 2.4.2 神经网络的价值函数



#### 2.4.3 神经网络反向传播（用于求导以进行梯度下降可当做黑盒理解一般不需要自行实现）

代表第4层第j个结点的误差值

核心思想:通过从后往前求解每一层的偏差值进而确定梯度下降法的导数值

核心流程：

1. 准备训练集
2. 遍历每个样本i，在每个样本中重复以下流程



1. Matlab应用fminunc的修正接口矩阵技巧



#### 2.4.4 梯度检验（Gradient Checking）（计算量很大记得要禁用）

该式子近似计算导数，并与反向传播的结果对比验证反向传播算法计算是否正确

#### 2.4.5 随机初始化(Random Initialization)

不能简单的把参数全初始化为0，这样会导致学出的函数过分简单，将每个变量在maltab可如下初始化，INIT\_EPSILON为超参数

### 2.5 调试机器学习模型（监督学习）

#### 2.5.1 改进模型几种方法

首先要判断出现了欠拟合（高偏差）还是过拟合（高方差）的问题，通过学习曲线判断（后面有详细描述）具体的问题类型

得到更多的训练集样本 → 解决高方差问题

尝试较少的特征 → 解决高方差问题

获取更多的特征 → 解决高偏差问题

增加多项式特征 → 解决高偏差问题

减小正则化参数 → 解决高偏差问题

增大正则化参数 → 解决高方差问题

#### 2.5.2 评估假设函数

1. 按照7:3划分训练集为训练集和测试集，用训练训练出，计算出测试集的误差
2. 60%作为训练集,20%作为验证集,20%作为测试集

训练集用于训练模型，验证集用于对模型的选择，测试集用于对模型的泛化能力进行测试

1. Underfit（Bias）:训练集和验证集误差都很大

Overfit(Variance):训练集误差很小，验证集误差很大

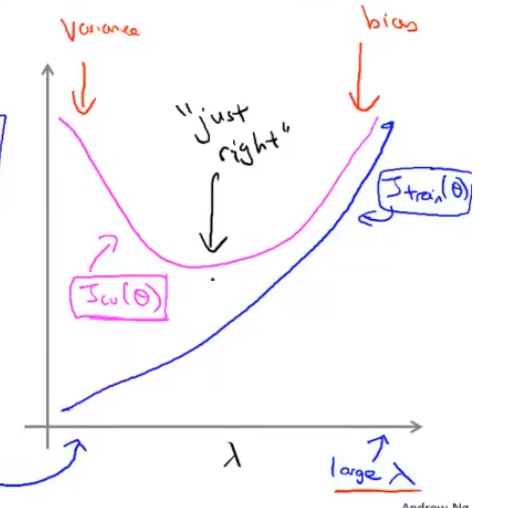
#### 2.5.3正则化与偏差及方差

正则化参数过大会导致欠拟合，过小会导致过拟合

按一定步长选取不同的正则化参数，通过梯度下降来拟合不同的参数模型（这里是有正则化参数的）

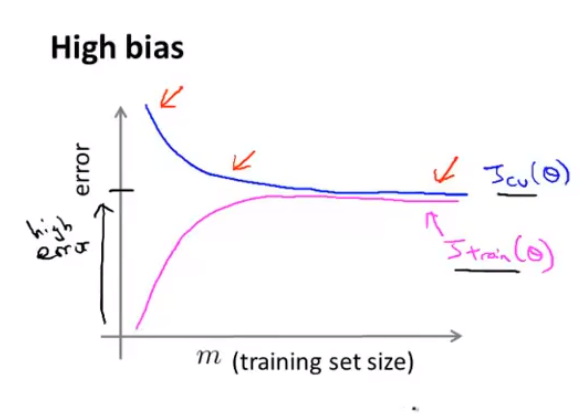
通过以下两式计算模型在验证集的误差（注意没有正则化参数）

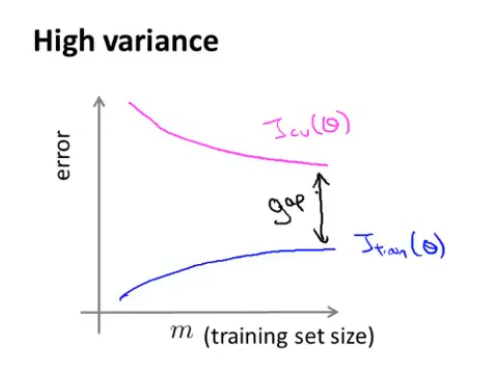




#### 2.5.4 学习曲线(Learning curves)

欠拟合时，训练集和验证集误差随着样本的数目变化情况如下，可以发现高偏差情况下增加样本并不能够提升模型的精度了 ，并且最终训练集和验证集的误差是差不多的

过拟合时训练集和验证集误差随着样本的数目变化情况如下，可以看出增加训练集样本的数目是有效的



#### 2.5.5 误差分析

1. 人工寻找出算法中预测错误的样本，从中分析出新的特征
2. 数值验证误差评价标准：
3. 偏斜类存在(不对称样本)的误差分类问题，即正确的样本比错误的样本多很多（反之亦然），这样正确率并不能代表算法好坏，假设全预测为正确，正确率依然很高

|  |  |  |
| --- | --- | --- |
| Predict Actual | 1 | 0 |
| 1 | True Positive | False Positive |
| 0 | False Negative | True Negative |



两者均越高越好，但是往往存在权衡，以癌症为例，当逻辑回归阈值偏大，那么意味着查准率较高，但是召回率较低；逻辑回归阈值较小，意味着查准率较低，但是召回率较高；因此为了衡量不同模型的效果，需要将两者统一计算F值比较好坏

越大越好

#### 2.5.6 数据对于模型算法的影响

用有许多参数的算法模型解决→解决高偏差问题

用有非常多数据的数据集→解决高方差问题（不太可能过拟合）

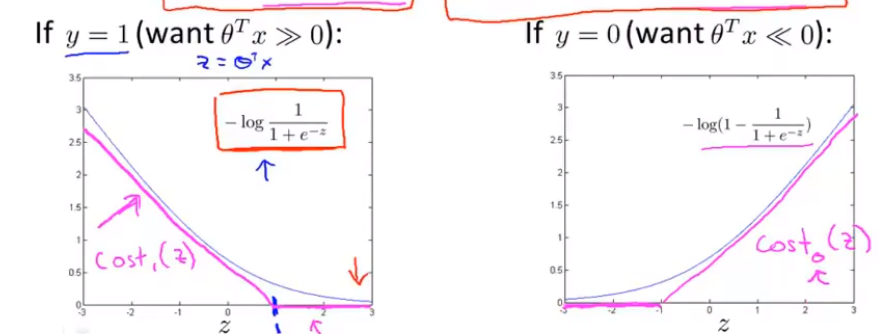
### 2.6支持向量机(SVM) 大间距分类器（监督学习）

#### 2.6.1 代价函数（C是衡量正则化和目标函数的正则化参数）

是在逻辑回归之上进行修改的函数，是对逻辑回归的一种近似处理



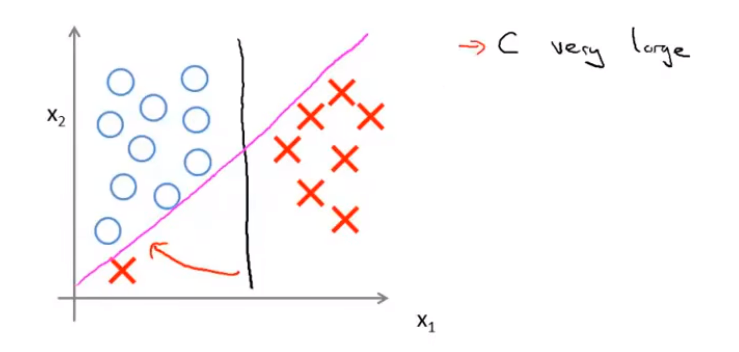
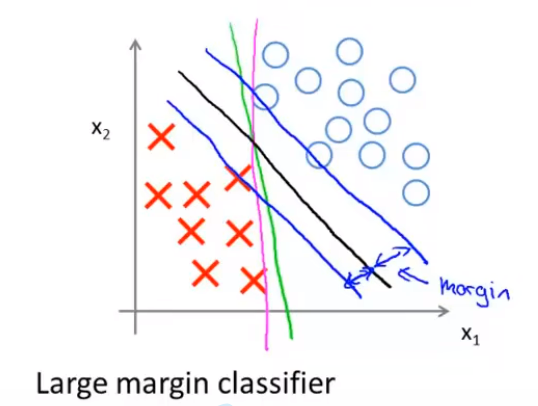
Cost1和cost2的函数对应下图，是逻辑回归函数的一种近似



假设函数：



优化上述的问题所得到的分类器是大间距分类器，拥有最大的margin（数学推导比较复杂，略过），同时SVM对于异常样本十分敏感（在C非常大的情况下），C较小的时候也会使得其对异常样本降低敏感度



#### 2.6.2核函数(Kernel)-用于定义新的特征，适用于复杂非线性边界

1. 高斯核函数



如果那么 标志着训练点和标记点离的很近

如果标志着训练点离标记点的距离

1. 标记点的产生

令每个标记点等于每个训练点从而产生标记点



1. 有核函数的SVM分类器价值函数



C越大意味着高方差低偏差

C越小意味着高偏差低方差

偏大，特征越光滑导致了高偏差低方差

偏小，特征越剧烈，导致了高方差，低偏差

#### 2.6.3 使用SVM

1. 选择正则化参数C
2. 选择核函数（线性核函数意味着原始特征、高斯核函数要选择、多项式核函数）
3. 逻辑回归和SVM对比(n表示特征数，m表示样本数)：

如果特征数目远大于训练集样本，那么采用logistics回归（n=10000,m=10~1000）

如果特征数目较小，训练集样本居中，那么采用SVM高斯核函数(n=1~1000,m=10~10000)

如果特征数目小，训练集样本很多(n=1~1000,m=50000)那么采用逻辑线性回归或者SVM不含核函数

### 2.7 K-means聚类算法(无监督学习)

#### 2.7.1 K-means算法流程

1. 随机初始化K个聚类中心
2. 循环以下过程：

聚类中心分配。根据每个样本点距离每个聚类中心的距离比较确定该点属于哪个聚类中心相当于固定价值函数的；

移动聚类中心。对所有聚类中心所属的点求平均值作为新的聚类点的位置（没有点的聚类中心一般移除，如果要求类数则重新随机初始化）相当于固定价值函数的

#### 2.7.2 优化目标（失真代价函数、价值函数）

表示样本所属的聚类中心的索引

表示第k个聚类中心的位置

表示样本i所属的聚类索引所对应的聚类中心

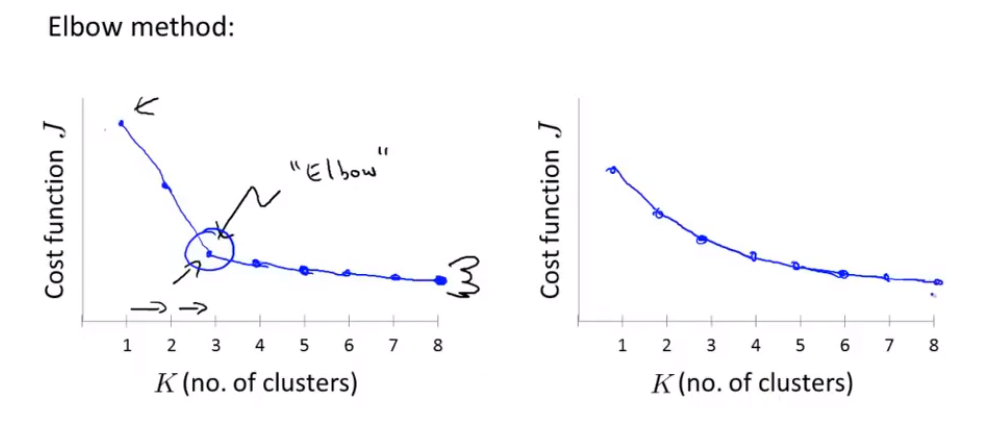


#### 2.7.3 随机初始化

从训练集随机选择K个样本点作为初始聚类中心

解决局部最优的方法：多次进行随机初始化（聚类数较小的情况下）

#### 2.7.4 选取聚类数量



聚类突变点选择聚类数量，但有时不会有突变点如右图，由业务下游决定

### 2.8 降维（Dimension Reduction）

## Baseline逐行精读

### 数据处理

1. （笔记本需要加fn功能键）Ctrl+shift+c复制文件地址，在Python里面因为“\”为转义字符，因此直接粘贴文件绝对路径不可行，需要把绝对路径的“\”修正为“\\”如"C:\Users\Lenovo\Desktop\下载.zip"=》"C:\\Users\\Lenovo\\Desktop\\下载.zip"或者“r"C:\Users\Lenovo\Desktop\下载.zip"”也可以

### 3.2 Pytorch入门

#### 3.2.1 dir和help

dir用来探索工具包的组成

help用来理解某个工具包的函数/类的使用

Dateset ??也可以用来看说明

#### 3.2.2 pytorch加载数据 对应read\_data.py

Dataset类:提供一种方式去获取数据及其label，告诉使用者如何获取每一个数据及其label，告诉使用者总共有多少数据

数据集几种组织形式：文件夹名称为label，内部为数据集



一个文件夹作为数据集，另一个文件夹作为label



直接将label写在图片的名称上一般是目标检测