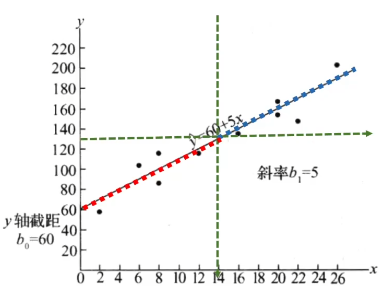
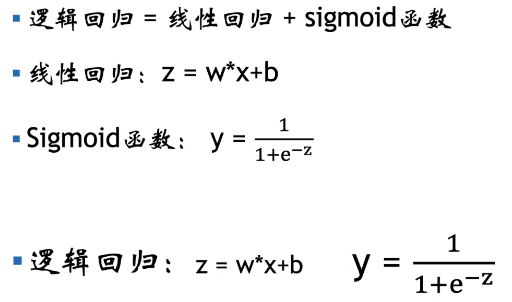
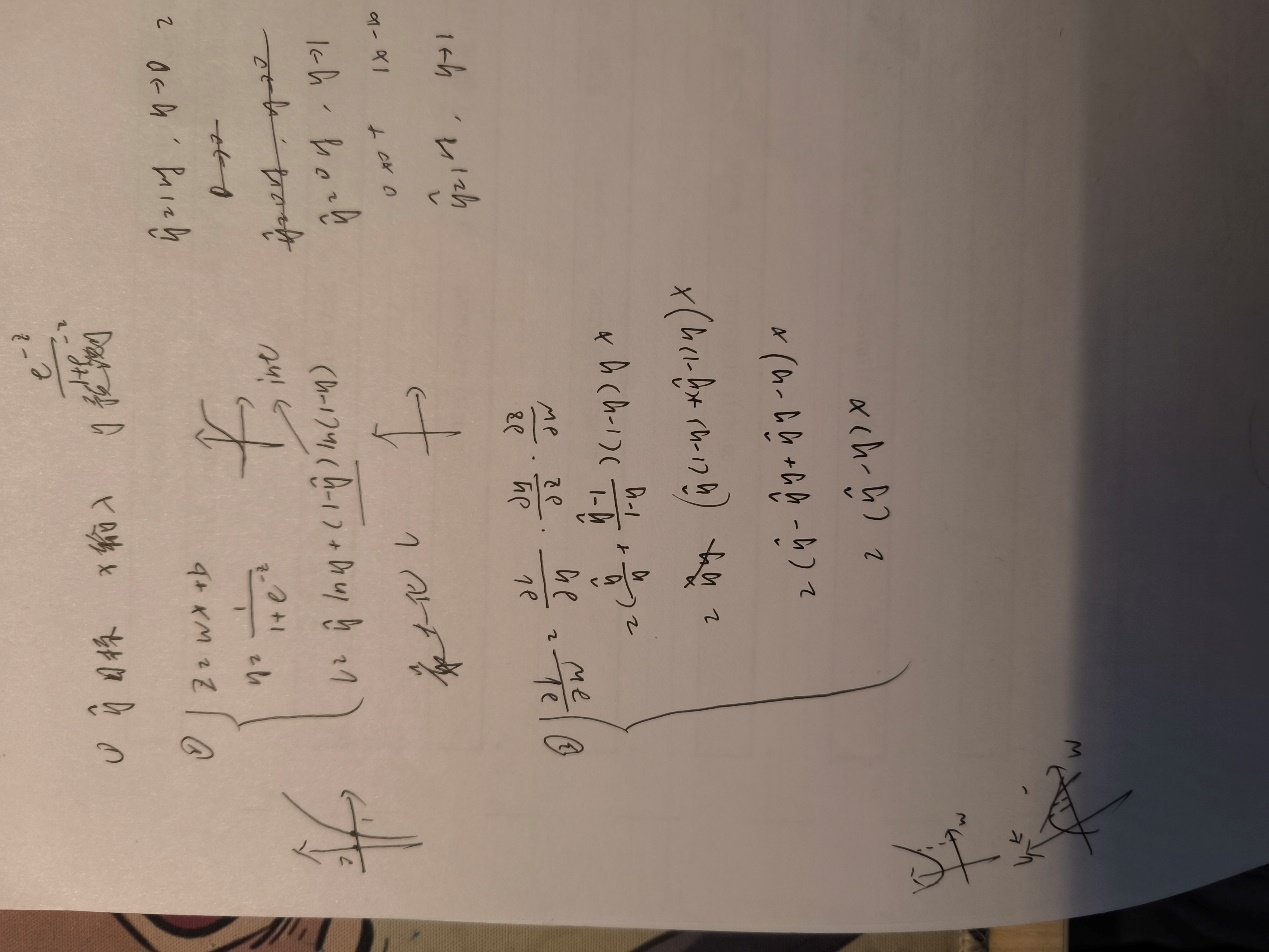
# 基础知识

1. 逻辑回归用于解决二分类问题
2. 逻辑回归=线性回归+sigmoid函数
   1. 线性回归解决回归问题，拟合出一条符合数据的一条直线  
      
   2. 在数据中添加一条y=130的线，线的上下各为一类，将原数据的y都换为0或1，那么回归问题就变成了分类问题
   3. 这时候子在线性回归的结果（拟合线）的基础上嵌套sigmoid函数就能实现分类：给出x，拟合线预测了y，再将y通过sigmoid映射到0-1间可以实现二分类（以0.5为分界面）
   4. 
   5. 
      1. y为样本目标类别{0,1}
      2. h为y的估计值
      3. 加log是为了当预测错误时（log0）给无穷的惩罚，而预测正确时（log1）不给惩罚

# 实验过程

## Logistic回归1

1. 实现logistic模型
   1. Linear函数计算线性回归部分，即前文所提
   2. Sigmoid函数调用linear函数，计算逻辑回归结果，即前文所提
   3. Loss函数调用sigmoid函数，计算损失函数
   4. Grad函数用于计算梯度，通过计算，得到w的梯度为(y’-y)x，b的梯度为(y’-y)，其中y’为专家标签，y为预测值。梯度推导过程如下：  
      
   5. Learn函数用于更新梯度
2. 构造logistic实例，并使用learn方法进行梯度回传，同时绘制出决策边界（令sigmoid函数=0.5，得z=0（决策边界），给定一系列的x1，通过函数z=0求对应的x2，并绘制（x1,x2）即可）

## Logistic回归2

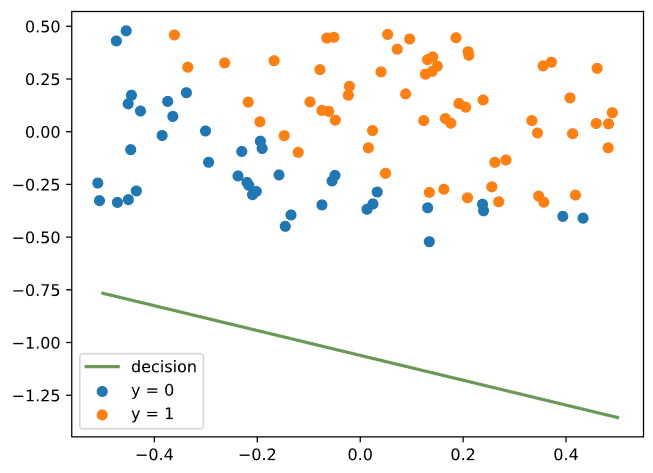
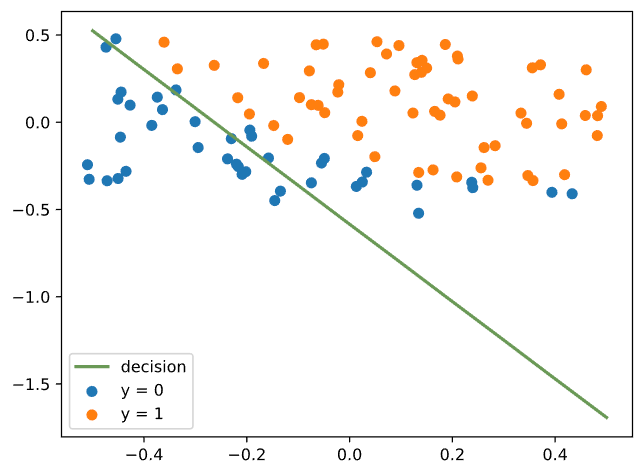
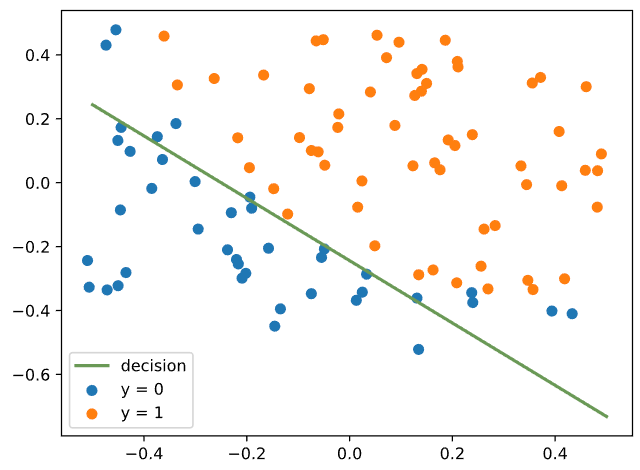
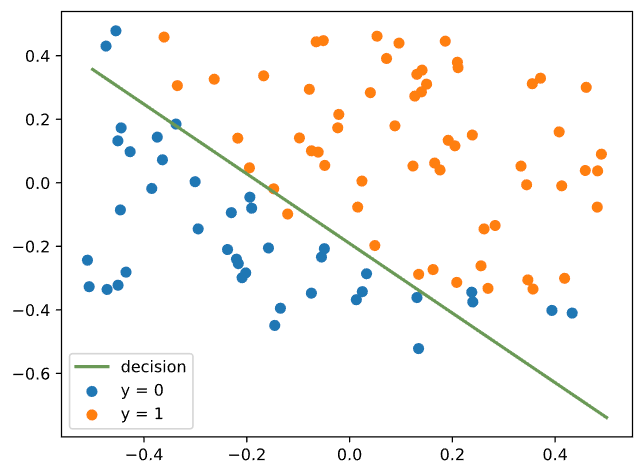
1. 构造高维特征，将原模型变为：

## 正则化

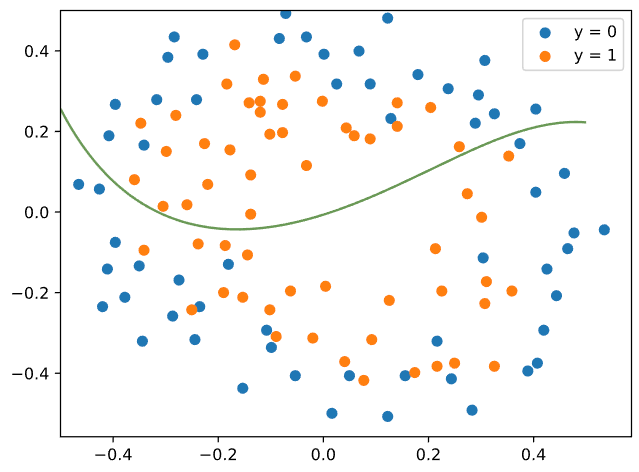
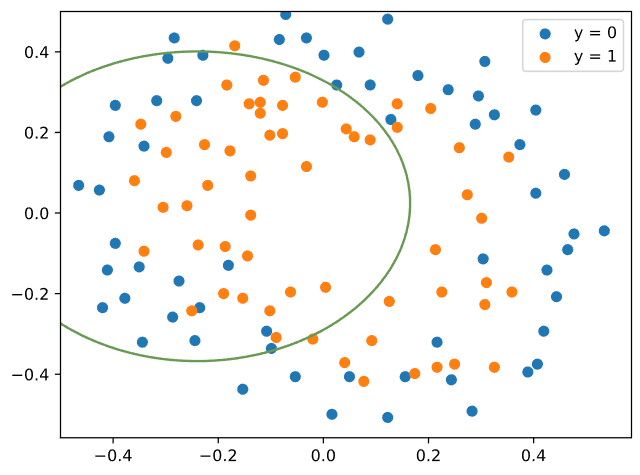
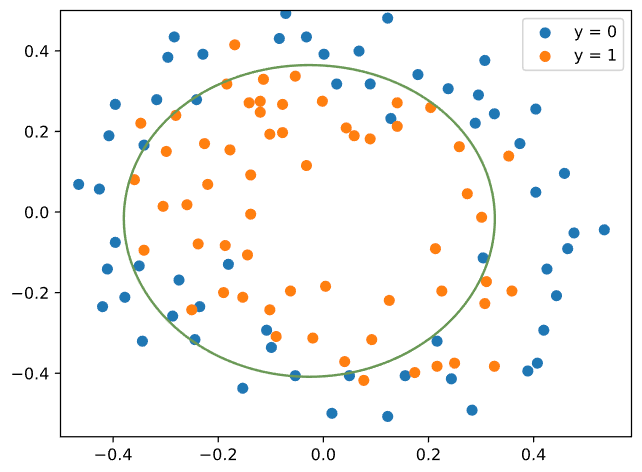
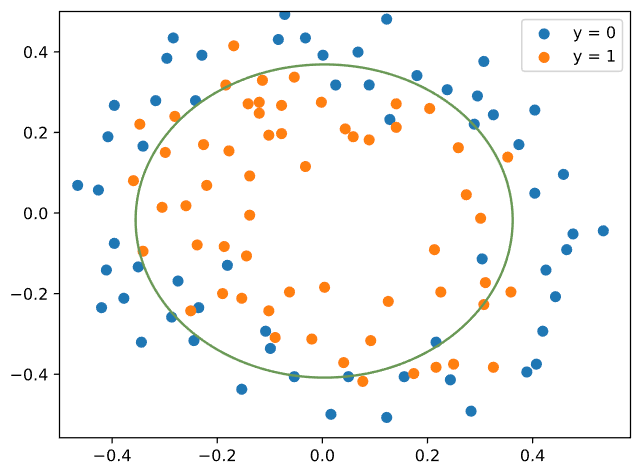
1. 使用L2正则化，设置权重为0.5

# 实验结果

## Logistic回归1

1. 学习率：0.1，epoch：1  
   
2. 学习率：0.1，epoch：10  
   
3. 学习率：0.1，epoch：200  
   
4. 学习率：0.1，epoch：500  
   

## Logistic回归2

1. 学习率：0.3，epoch：100  
   
2. 学习率：0.3，epoch：500  
   
3. 学习率：0.3，epoch：2000  
   
4. 学习率：0.3，epoch：6000  
   

## 正则化

1. 学习率：0.3，正则化权重：0.5，epoch：50000

# 实验心得

1. 训练前需要对数据进行特征规范化，否则训练会不成功，具体表现为：学习率很难调，进而导致收敛速度慢、无法收敛、收敛到不正确的地方