

ML

LAB 1 report

王一鸣

PB21000024

目录

1. [实验目的](#)
 2. [实验环境](#)
 3. [实验步骤](#)
 4. [实验结果与分析](#)
 5. [总结与反思](#)
-

实验目的

实现逻辑斯蒂回归

实验环境

python3.11

windows11H23

实验步骤

清洗数据

- 处理缺失值
 - 数量类型填充平均值
 - 类别类型填充种类最多的类别

- 处理非数量类型的变量

独热编码转化为0, 1类型向量供模型预测

注意, 这里不要对目标属性进行变换

- 数据正则化

对数据作归一化, 使得所有维度的数据落在统一的区间内方便模型进行预测

模型建立

- sigmoid函数

```
1/(1+np.exp(-x))
```

- 训练

- 损失函数

```
linear_output = np.dot(X, self.coef_)
loss=np.sum(-
np.multiply(y,linear_output)+np.log(np.exp(linear_output)+1))
```

- 梯度

```
grad=np.dot(X.T,y_pred-y)
```

梯度的惩罚项，防止模型过拟合

```
if self.penalty=="l2":
    grad=grad-self.gamma*self.coef_
elif self.penalty=="l1":
    grad=grad-self.gamma*np.sign(self.coef_)
```

- 预测

```
for i in range(X.shape[0]):
    key=self.sigmoid(linear_output[i])
    if key >= 0.5:
        probs[i]=1
    else:
        probs[i]=0
return probs
```

离散化预测值

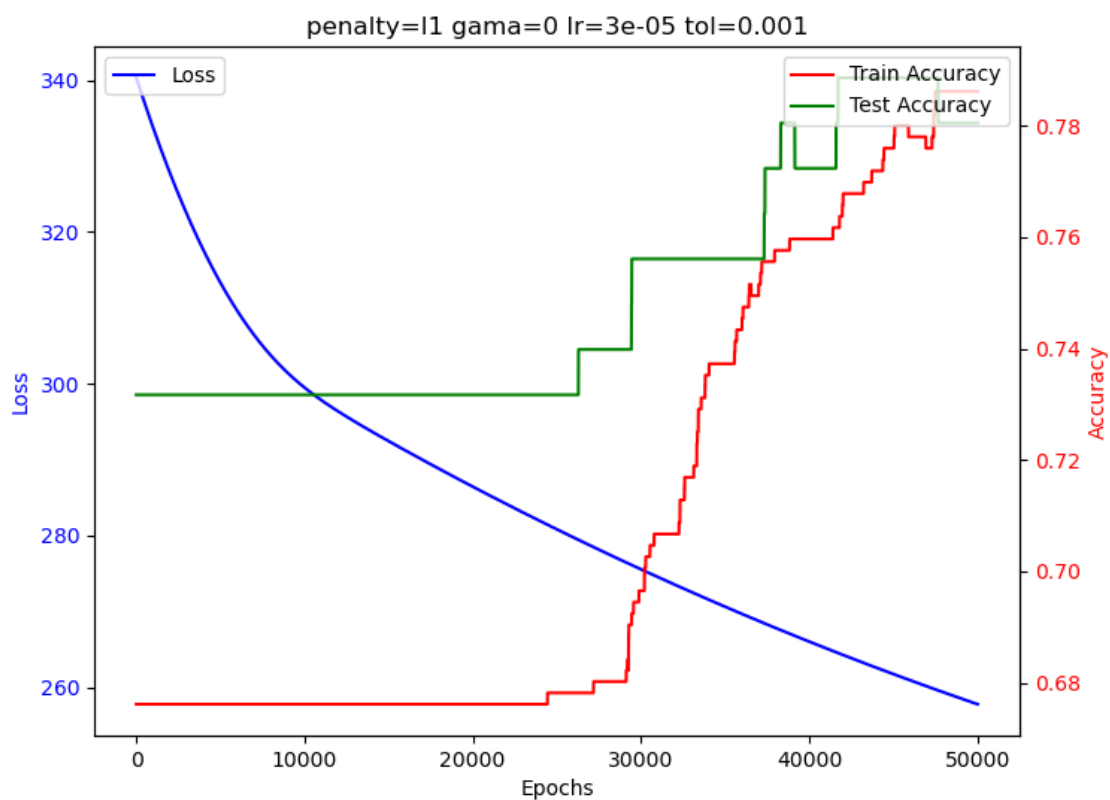
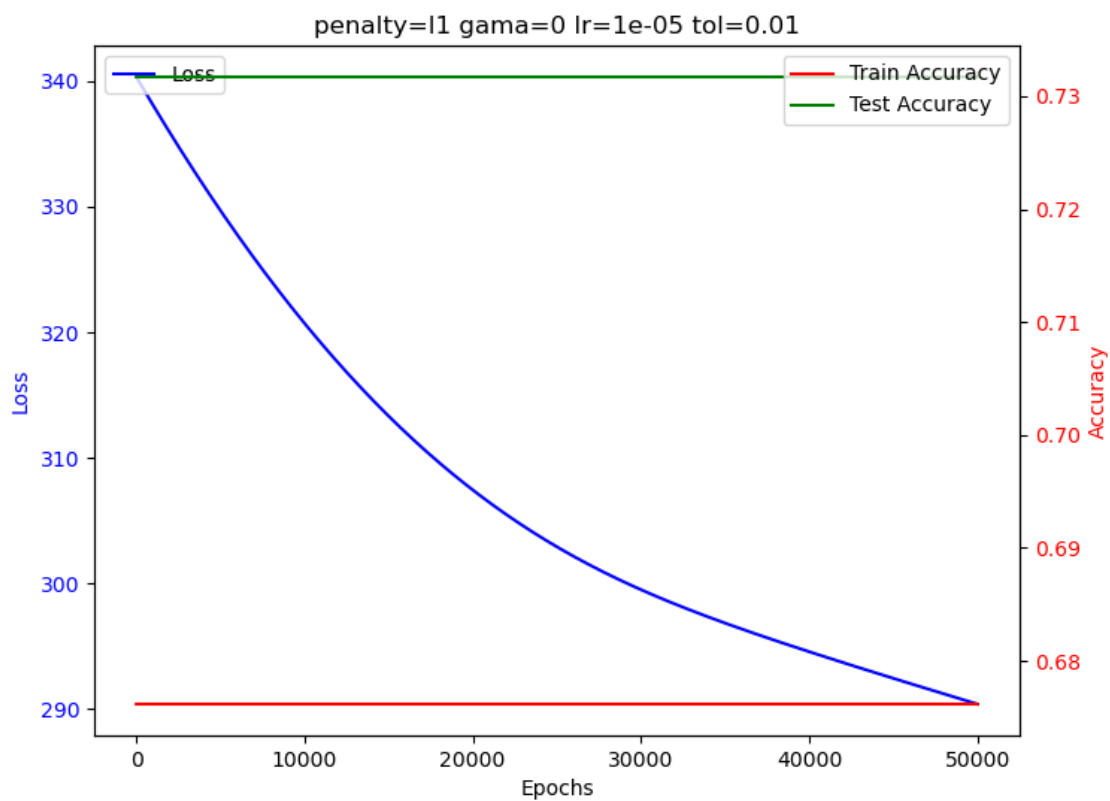
实验结果

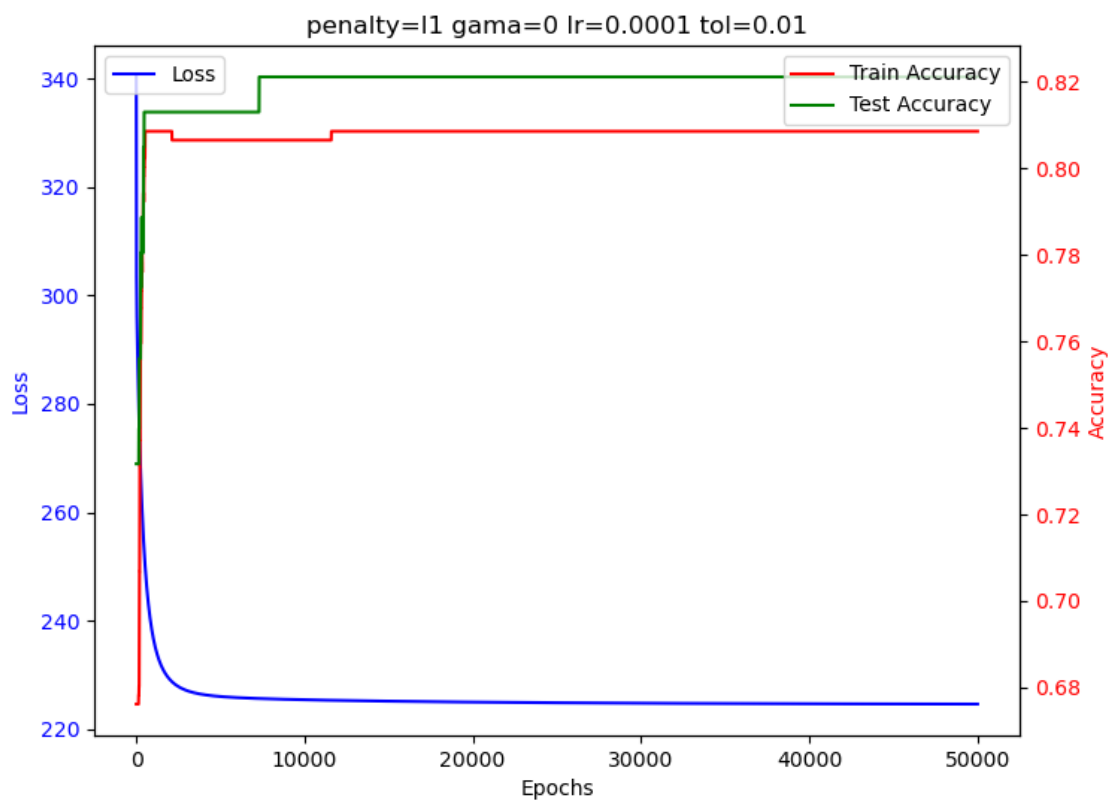
```
train acc:
0.8044806517311609
test acc:
0.8211382113821138
```

参数在gama=0 max_iter=1e5 lr=1e-4 penalty=l2的一次结果

- 参数的调整

首先我们对学习率作一些微调，尝试在收敛速度和拟合结果找到一个平衡

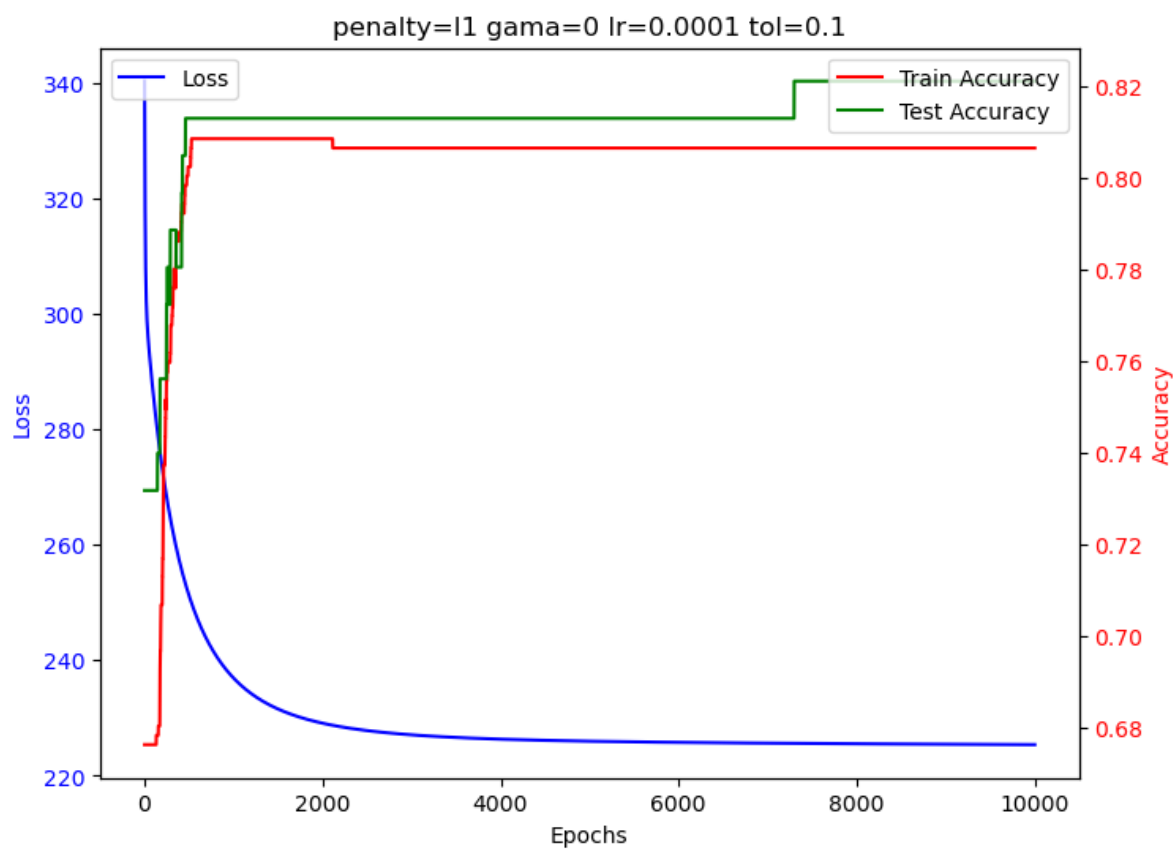


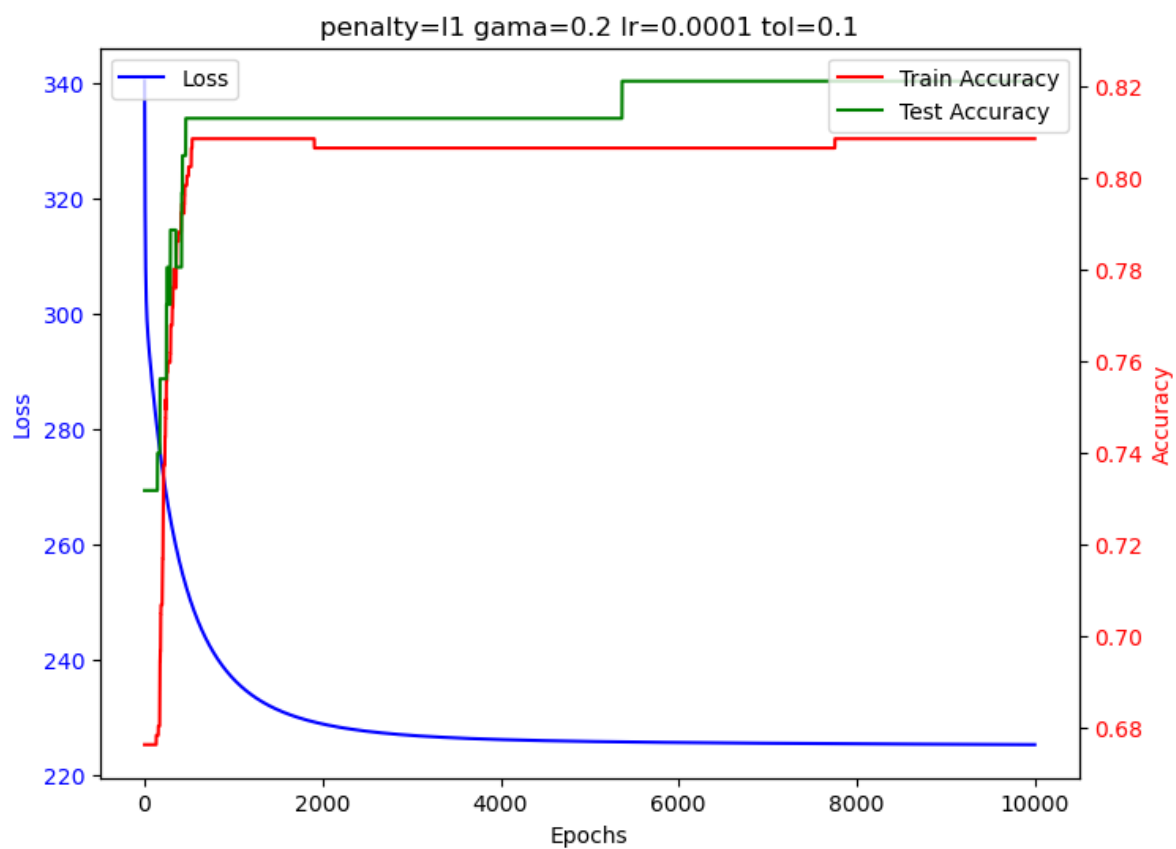
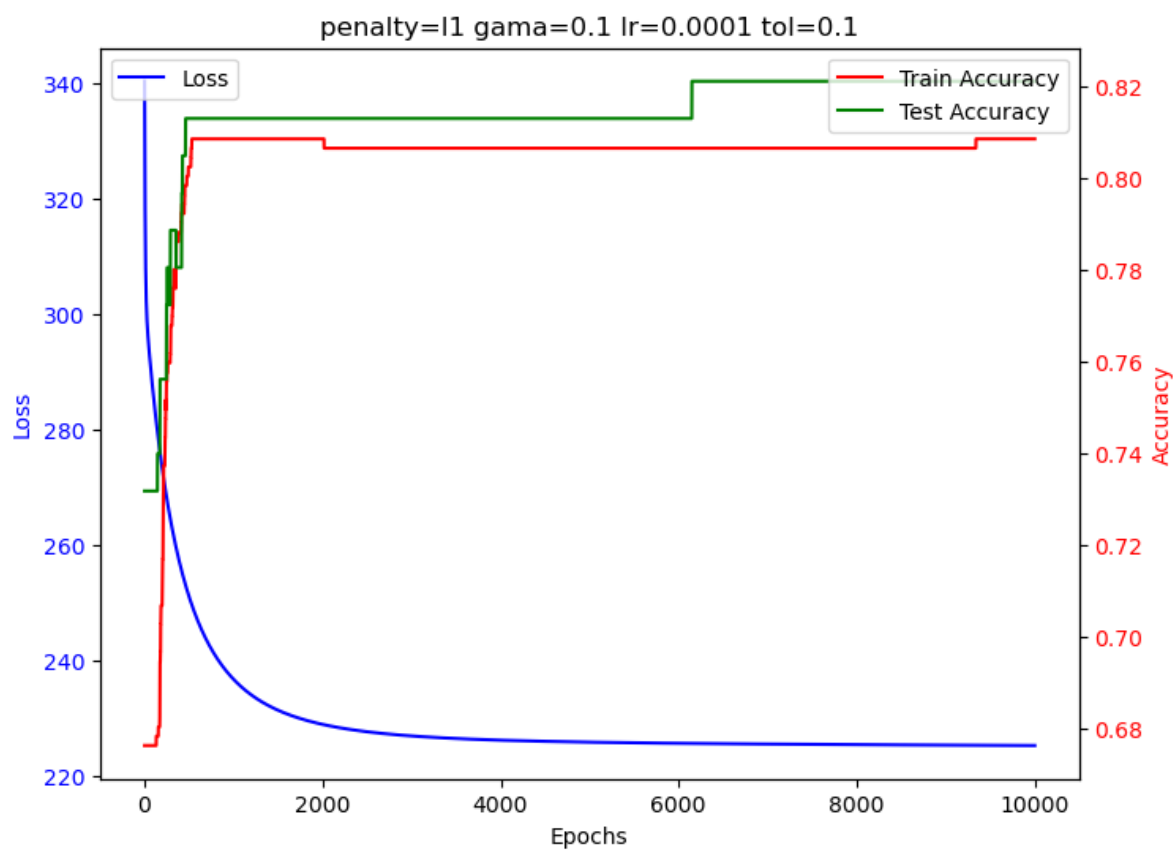


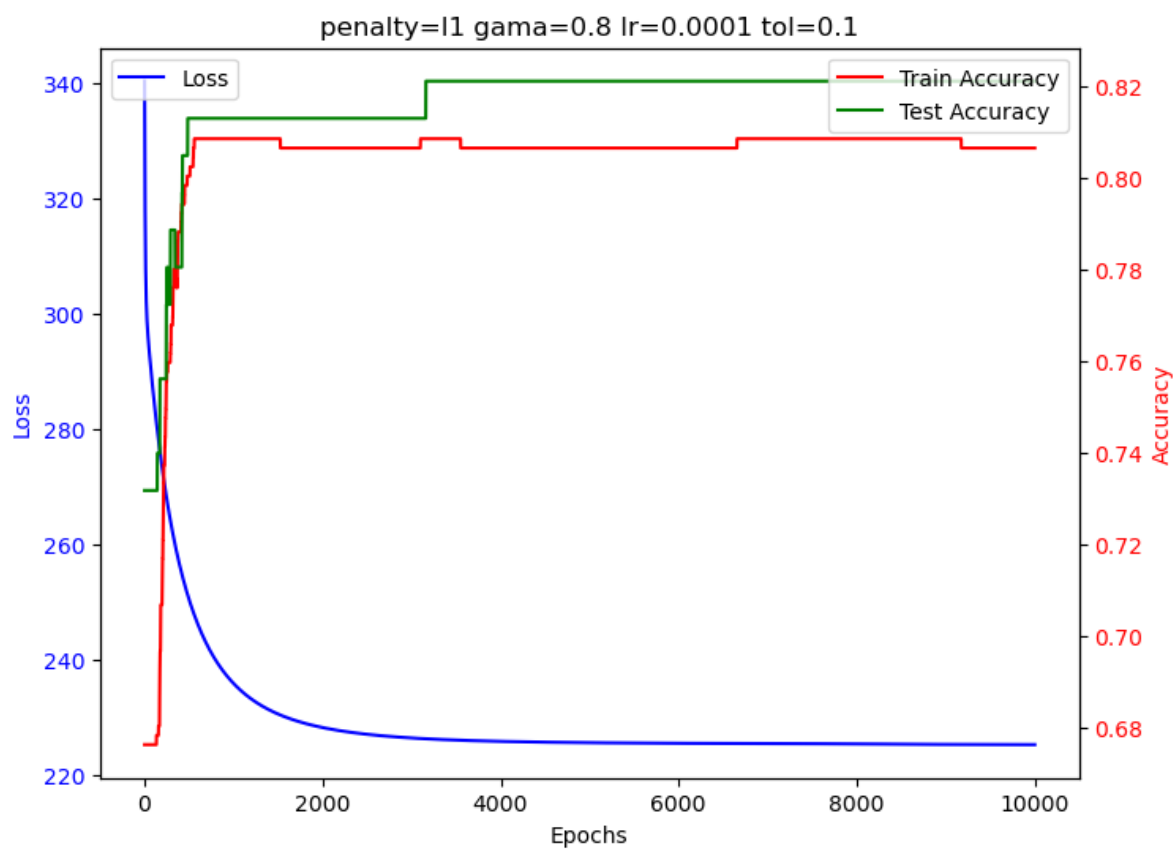
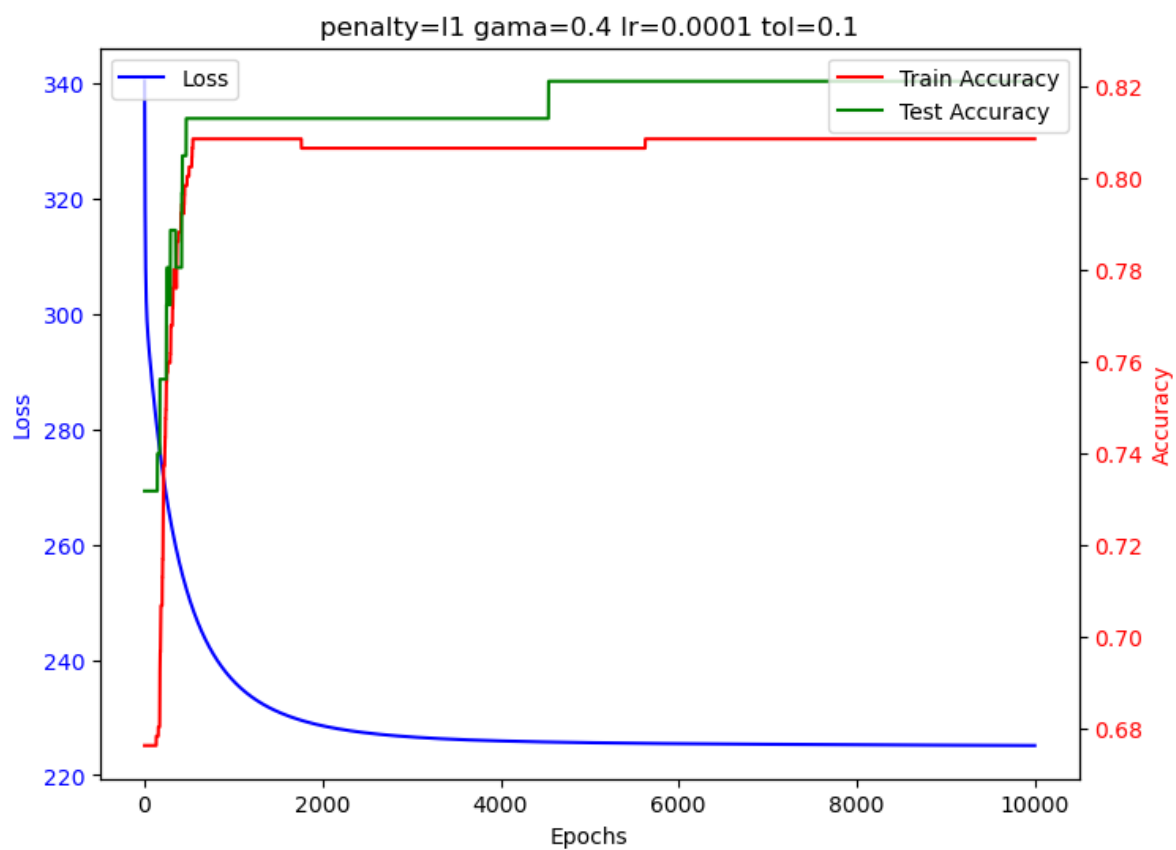
反复尝试后发现 $1e-4$ 时模型的收敛速度比较快，在1w步就收敛了

接着我们尝试不同的惩罚方式以及惩罚程度对于训练的影响

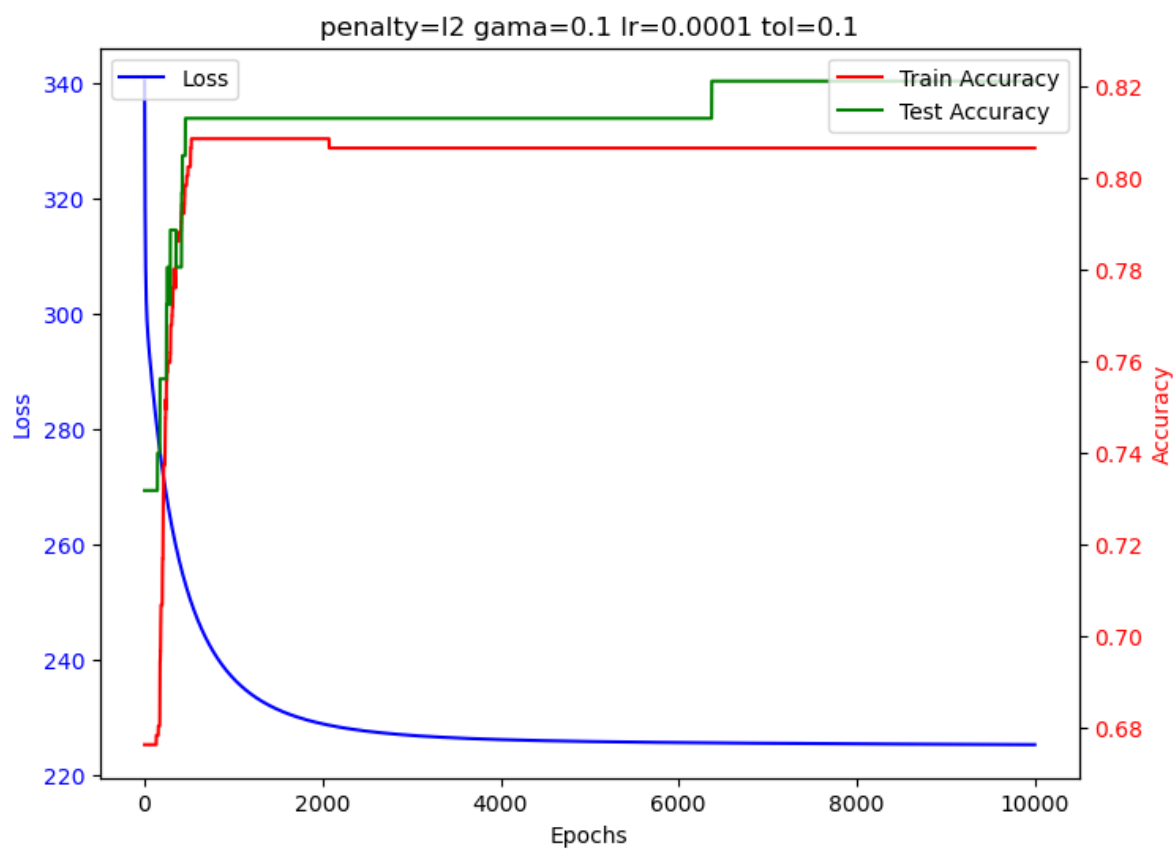
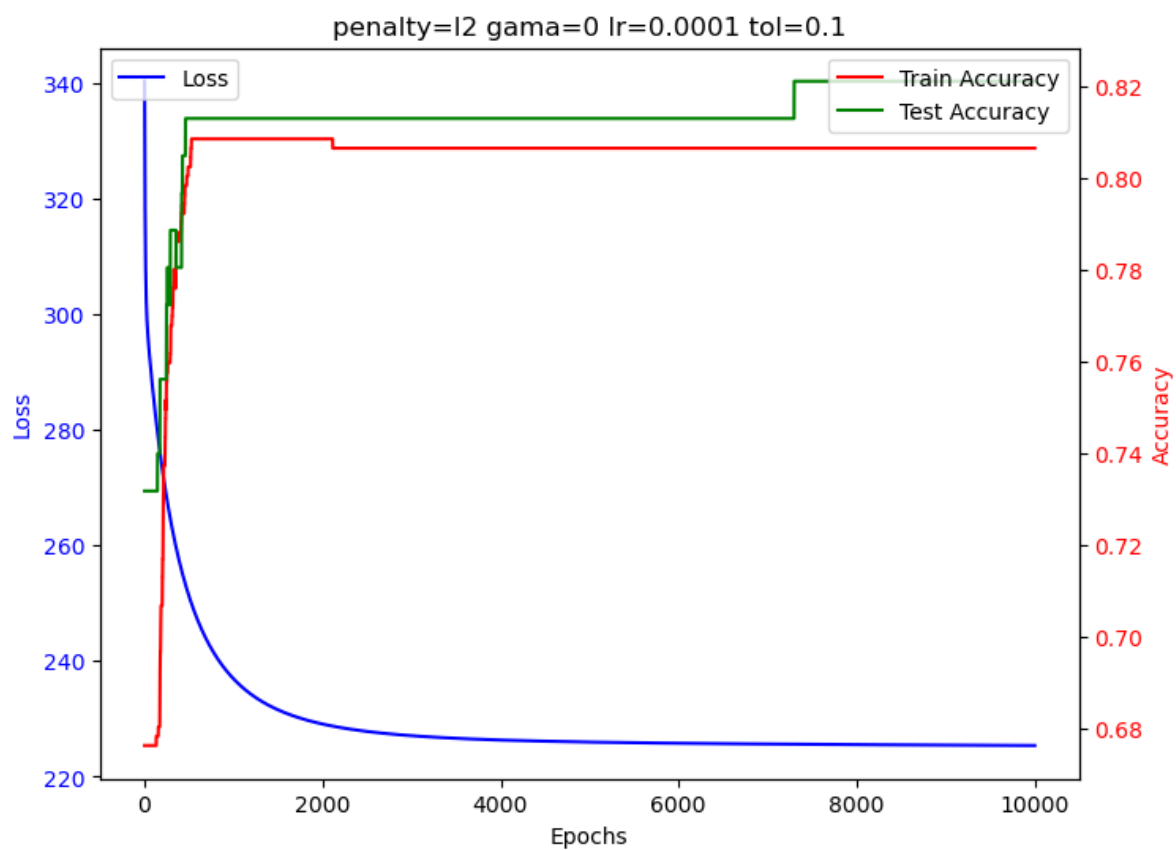
- l1惩罚

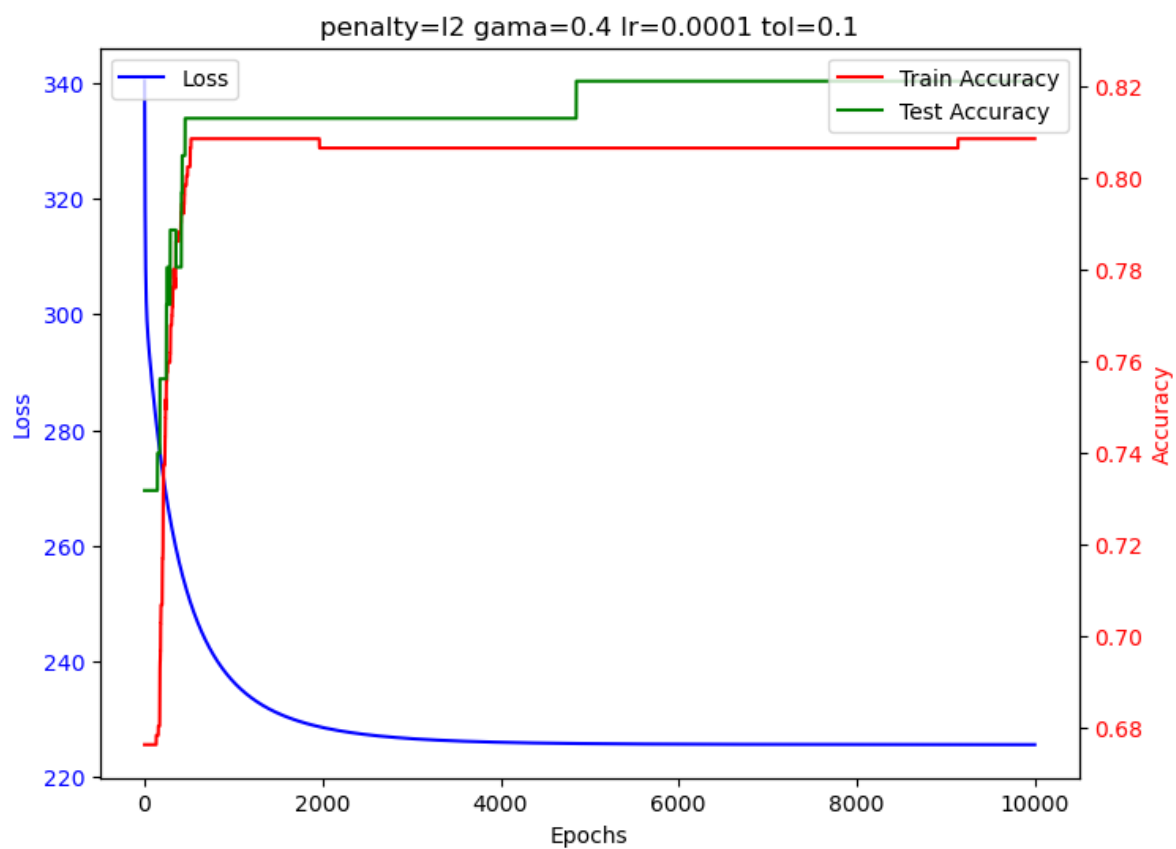
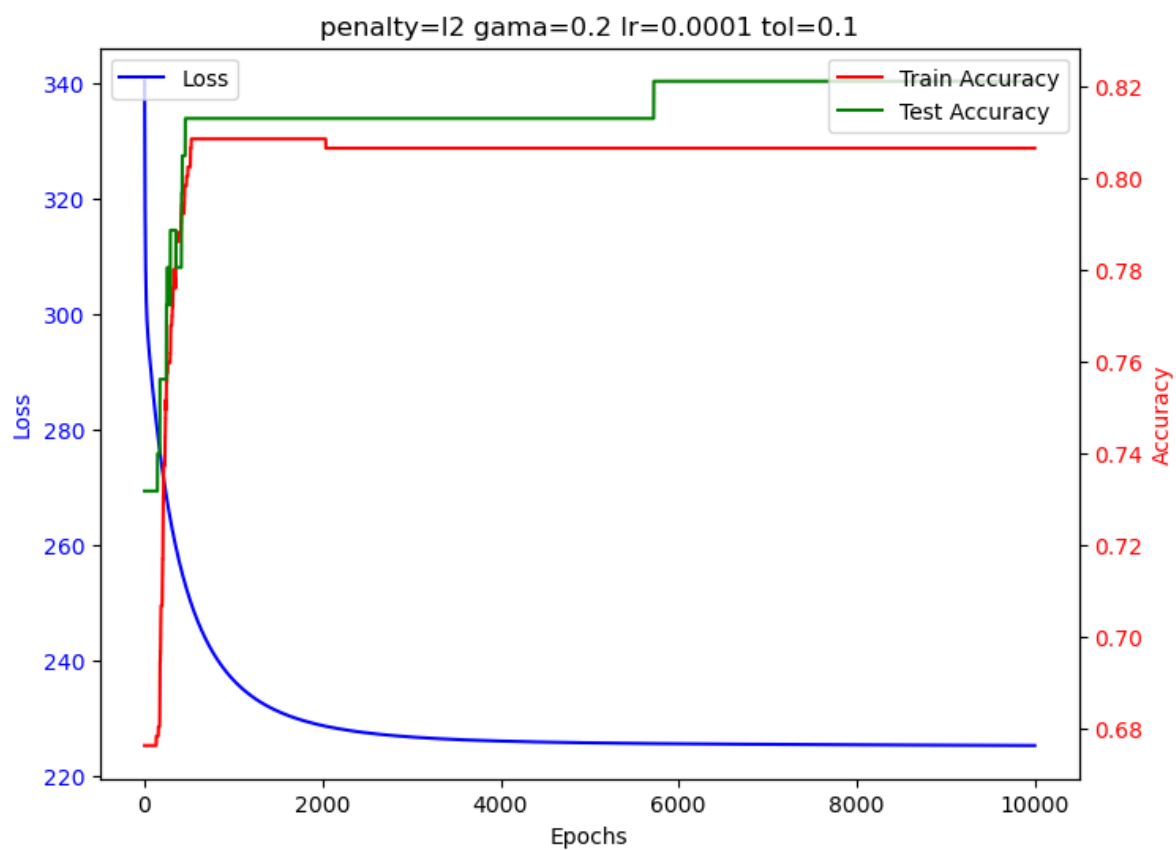


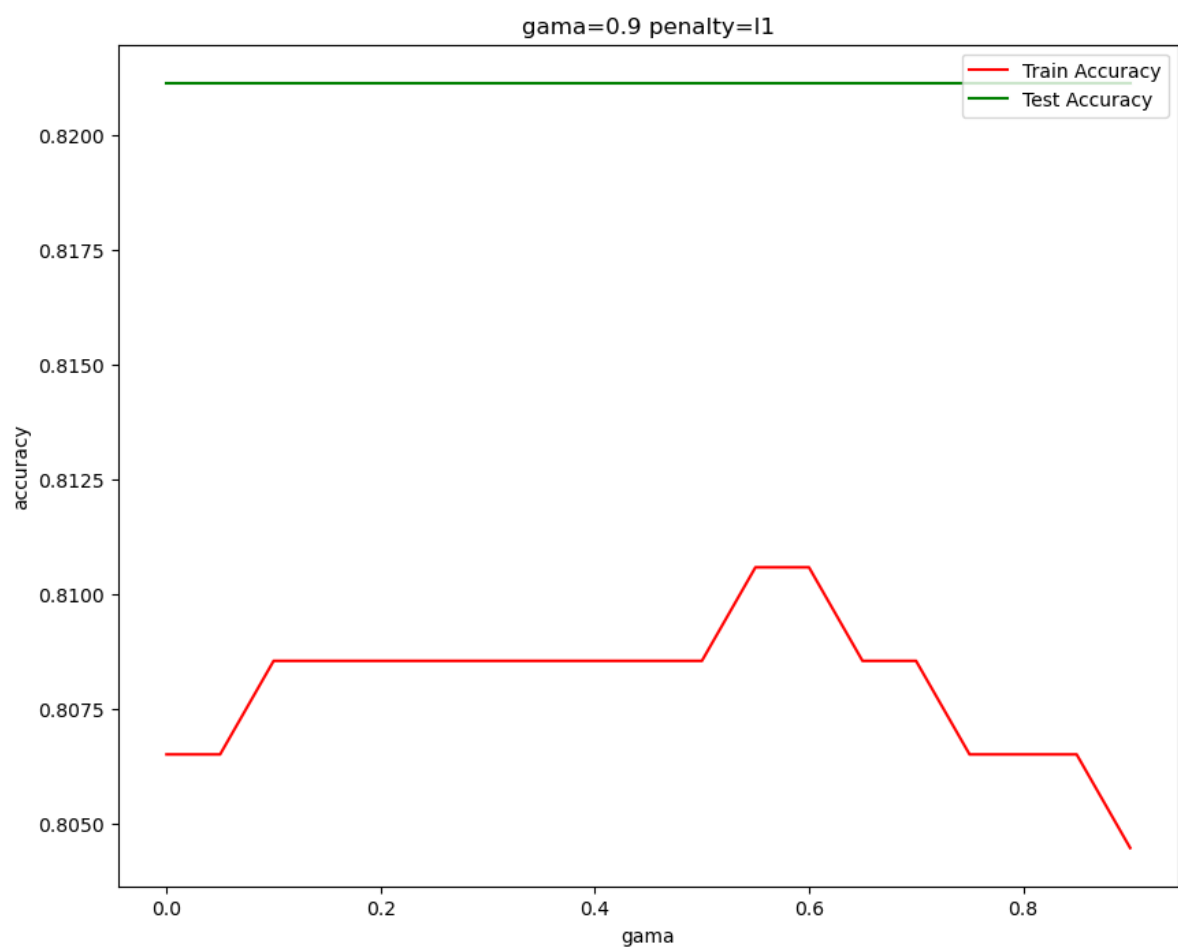
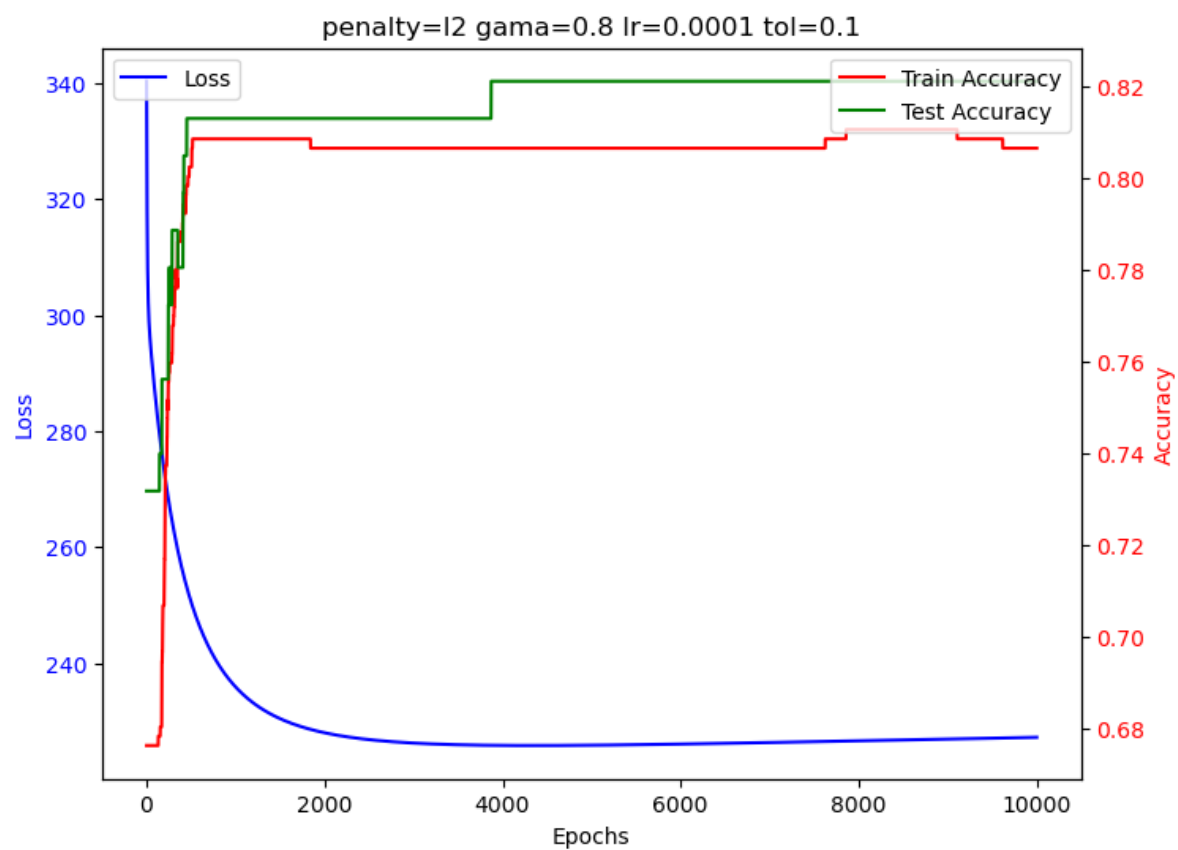


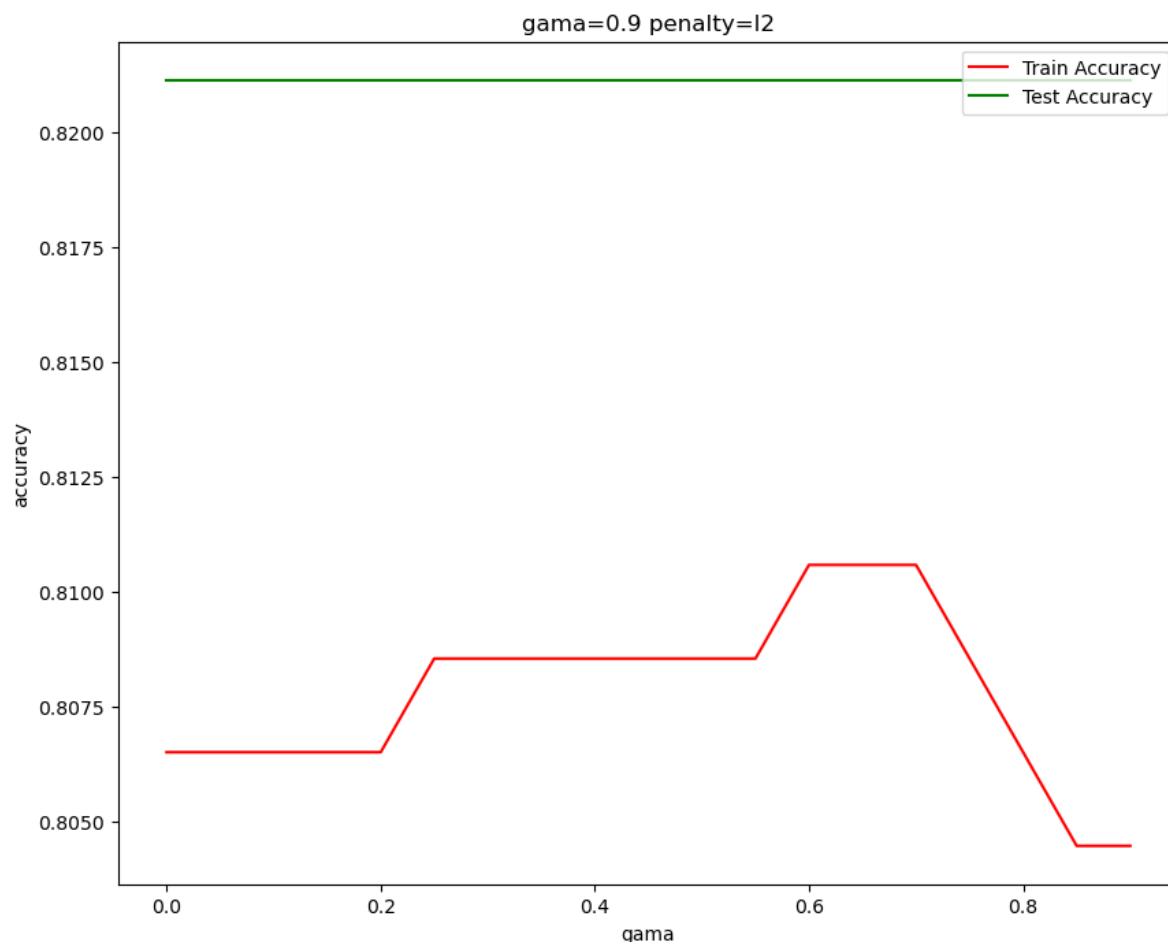


- l2 惩罚









观察图线可以发现，惩罚参数在0.6左右的时候训练集上的预测表现较好,不过在测试集上没有什么变化，推测是数据量太小导致现有参数的分类平面变化无法对样本点产生影响。

- 分析

关于实验中一些有趣的现象

测试集的分类准确率高于训练集，推测是数据划分的偶然性导致的，调整了几个随机种子，测试和训练误差在训练过程中保持一致。

对于收敛性的判定，模型始终没有按照收敛条件终止，检查每一步的梯度后发现梯度在模型接近收敛时维持在一个比较大的下界，可能需要我们在实际训练的时候观察梯度值进行手动设置。这里收敛时梯度的l2范数大概维持在0.02左右，但对收敛判定不如直接设置训练步长和时间花费上来的稳定，所以直接设置了一个比较大的步长训练，不考虑收敛的问题了。

实验中并没有出现过拟合的现象，随着训练迭代次数的增大，损失函数维持在一个恒定值，测试和训练误差也保持不变。单层分类器的学习能力有限，没有过拟合的问题。

总结与反思

在实验中踩得比较大的坑还是数据清洗部分，分类器需要非常规整的数据，而且由于样本量太小，对数据的处理会非常影响实验结果

一开始我直接把缺失值全部丢弃了，发现效果并不好，于是只能塞平均值或者出现最多的项进去

后面发现在训练过程中容易产生溢出的问题（因为损失函数取了exp在特征很大的情况下非常容易溢出），于是对数据进行了归一化处理解决这个问题，一通操作下来模型的性能才趋于稳定。

对参数的调节也算是个比较讲究的活，在调节过程中对训练的实时情况的可视化可以帮助我们理解训练器的状态，比如实时的画出训练和测试误差。
