

中山大学计算机学院

人工智能本科生实验报告

课程名称: Artificial Intelligence

教学班级	网安软工合班	专业 (方向)	网络空间安全
学号	20337251	姓名	伍建霖

一、实验题目

1. 在给定文本数据集完成大学生录取预测分类训练，画出训练 loss 曲线图、数据可视化图。（数据集课上发布）
2. （选做）在给定文本数据集完情感预测预测分类训练，此题不要求画曲线图。（数据集使用和朴素贝叶斯相同的数据集）

要求

- 设计合适的网络结构，选择合适的损失函数，利用训练集完成网络训练，画出数据可视化图、loss 曲线图，计算模型收敛后的分类准确率。
- 需要提交简要报告+代码
- 压缩包：学号_姓名_作业编号.zip，如 20331234_张三_实验7.zip。
- 截止日期：2022.5.25 23:59

二、实验内容

1. 算法原理

预测值 $y = g(z) = \frac{1}{1+e^{-z}}$, $z = w^T x + b$, $\dot{g} = g(z)(1 - g(z))$

损失函数 $J(w) = \frac{-1}{m} \sum_{i=1}^m (-y \log(\hat{y}) - (1 - y) \log(1 - \hat{y}))$

梯度下降 $w = w - \alpha \frac{\partial J}{\partial w}$

2. 伪代码

```
def descent:
    设置学习率, 迭代次数等参数
    while True:
        计算梯度方向
        更新参数
        计算损失值
        判断是否return
```

3.关键代码展示（带注释）

```
def descent(data,theta,batchsize,stoptype,thresh,alpha):
    i=0 #迭代次数
    k=0 #batch
    X,y=shuffleData(data) #洗牌
    grad=np.zeros_like(theta) #梯度方向
    costs=[cost(X,y,theta)] #计算代价函数的损失值

    while True:
        #1. 计算梯度方向grad, batchsize始值选择样本的数量
        grad=gradient(X[k:k+batchsize],y[k:k+batchsize],theta)
        #设置循环，每次迭代都选择新的一组[k,k+batchsize]的样本进行迭代
        k=k + batchsize
        #如果循环过程中k超过了样本总量，则让它回归至0，重新洗牌，再次循环
        if k >= n:
            k=0
            X,y=shuffleData(data)
        #2. 根据梯度下降公式求解theta值
        theta=theta - alpha*grad
        #同时将每次迭代中代价函数的损失值计算出来
        costs.append(cost(X,y,theta))
        cost1.append(cost(X,y,theta))
        #3. 选择下降停止策略：
        #循环计算次数i
        i=i+1
        #如果选择根据迭代次数停止，则value=迭代次数
        if stoptype==STOP_ITER:
            value=i
        #如果根据代价损失来停止，则value=代价函数
        elif stoptype==STOP_COST:
            value=costs
        #若根据梯度方向来停止，则value=梯度下降方向
        elif stoptype==STOP_GRAD:
            value=grad
        #直到所选择的value都满足阈值thresh，则停止循环
        if stopcriterion(stoptype,value,thresh):
            break
        #4. 返回最后更新的theta值，迭代次数，损失值和梯度方向
    return theta,i-1,grad
```

4.优化

扁平化

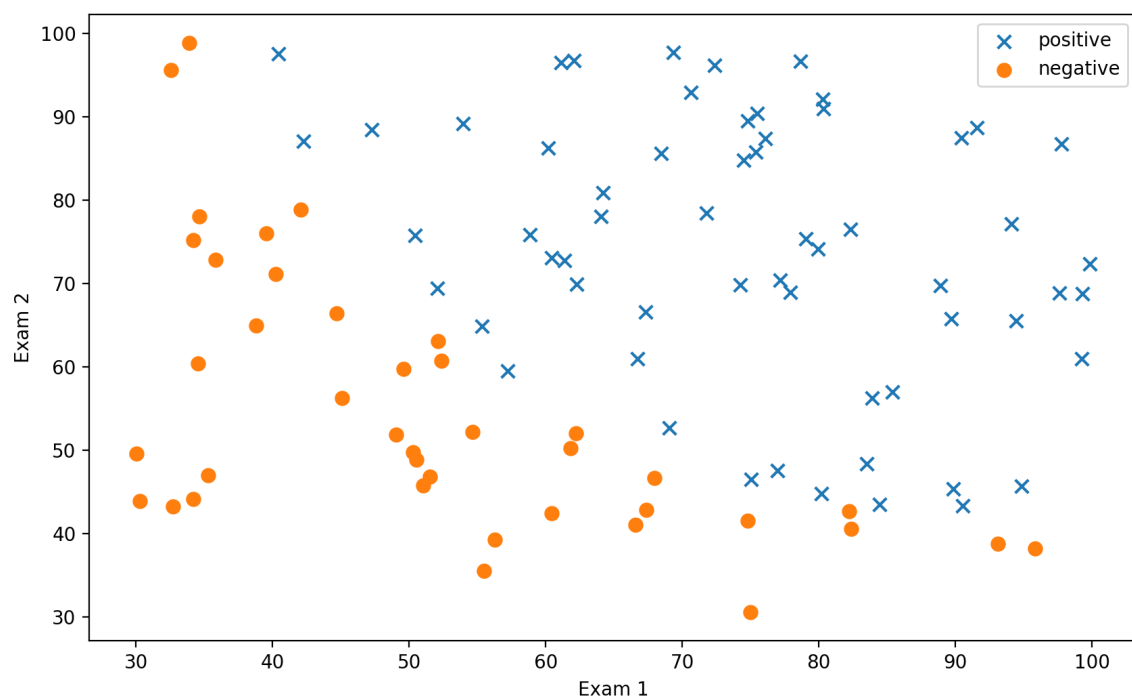
没有扁平化，那么error应该是一个(100, 1)的二维数组，X[:,j]是一个一维数组，如果通过multiply对应相乘，那么得到的结果将会是一个100X100的二维数组

标准化

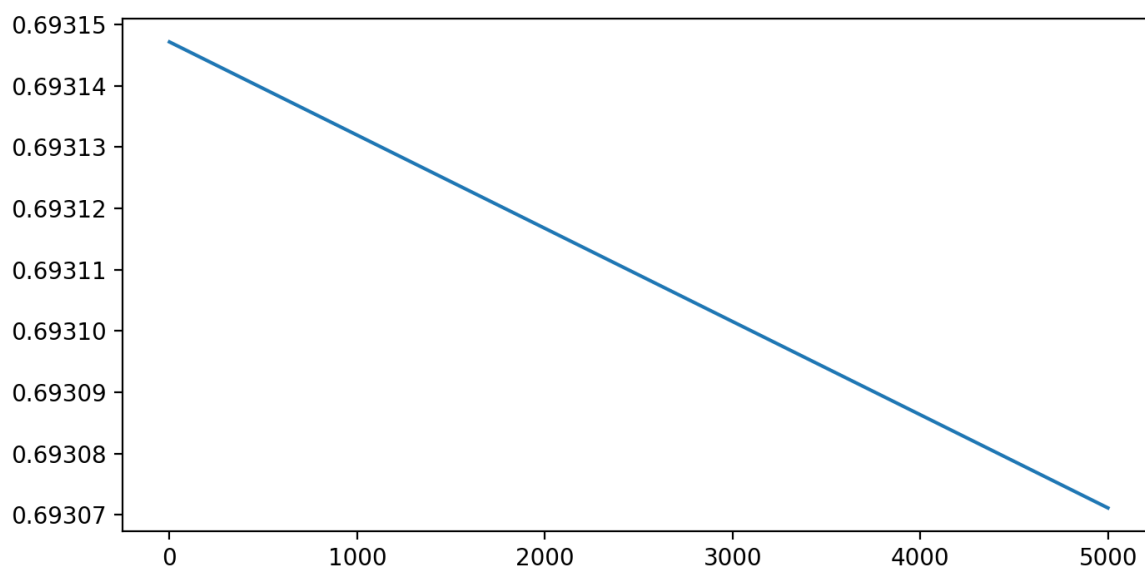
将所有特征的尺度都缩放放在 -1到1之间，这样可以帮助梯度下降算法更快的收敛

三、实验结果及分析

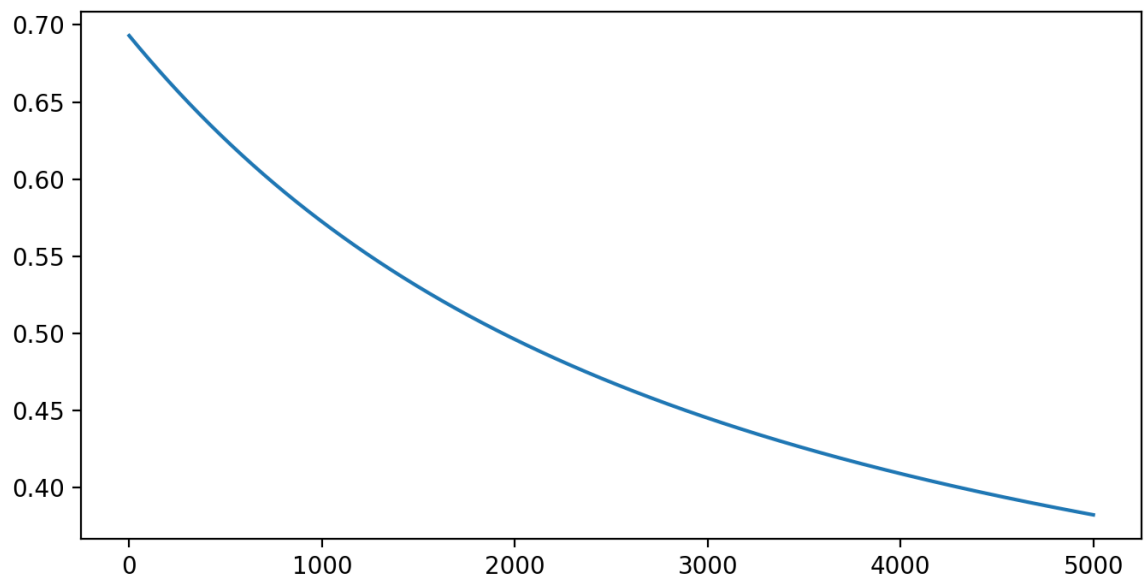
1. 实验结果展示示例（可图可表可文字，尽量可视化）



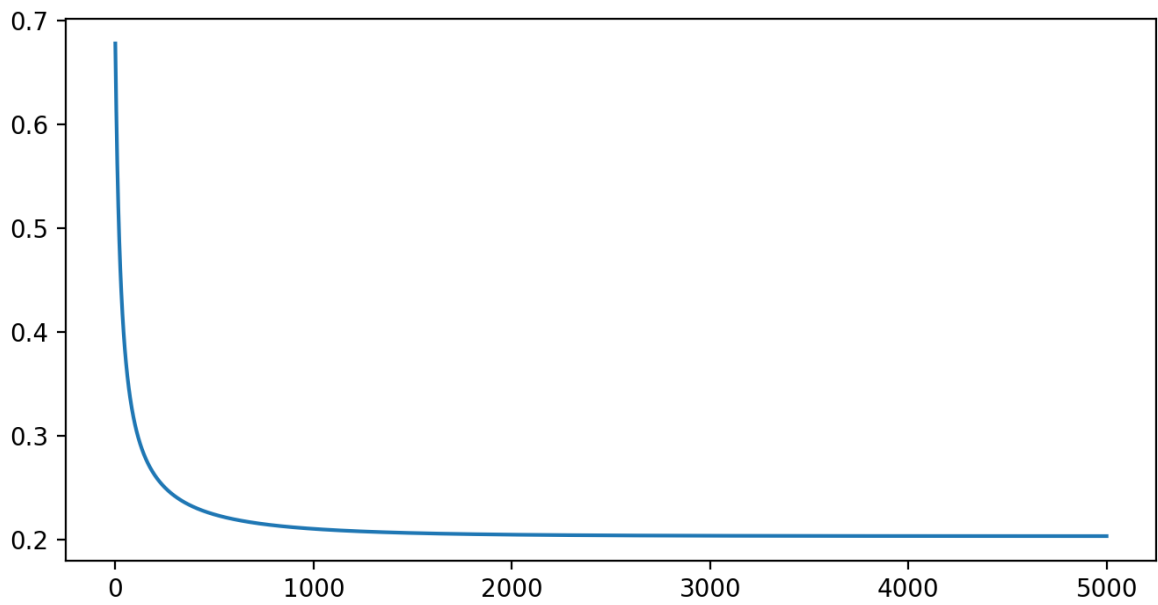
学习率为0.0000001，迭代次数为5000



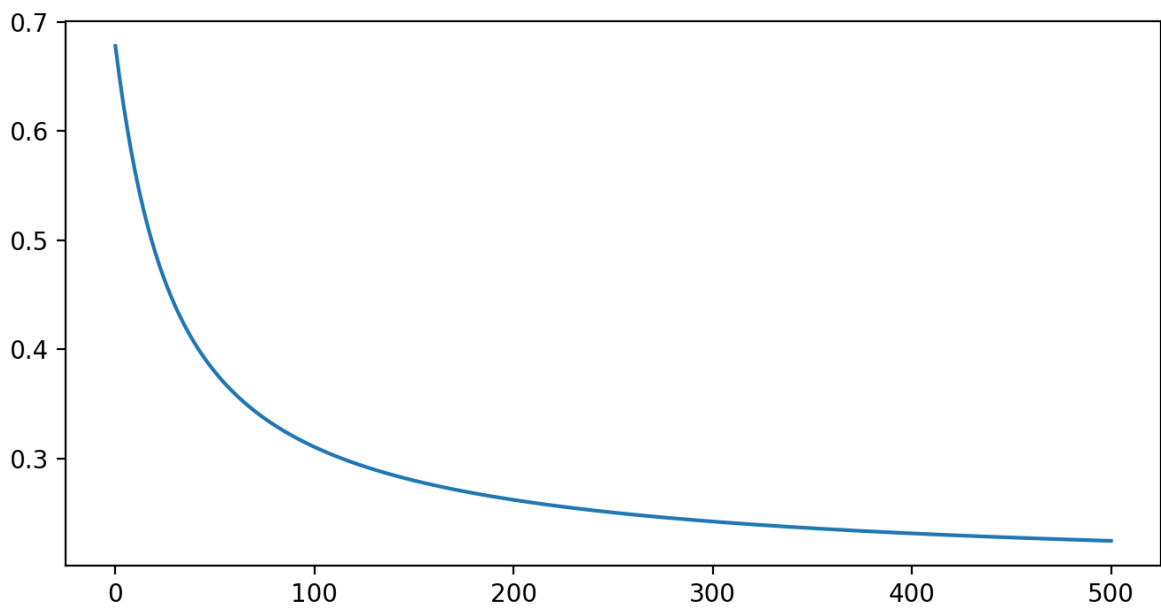
学习率为0.001，迭代次数为5000



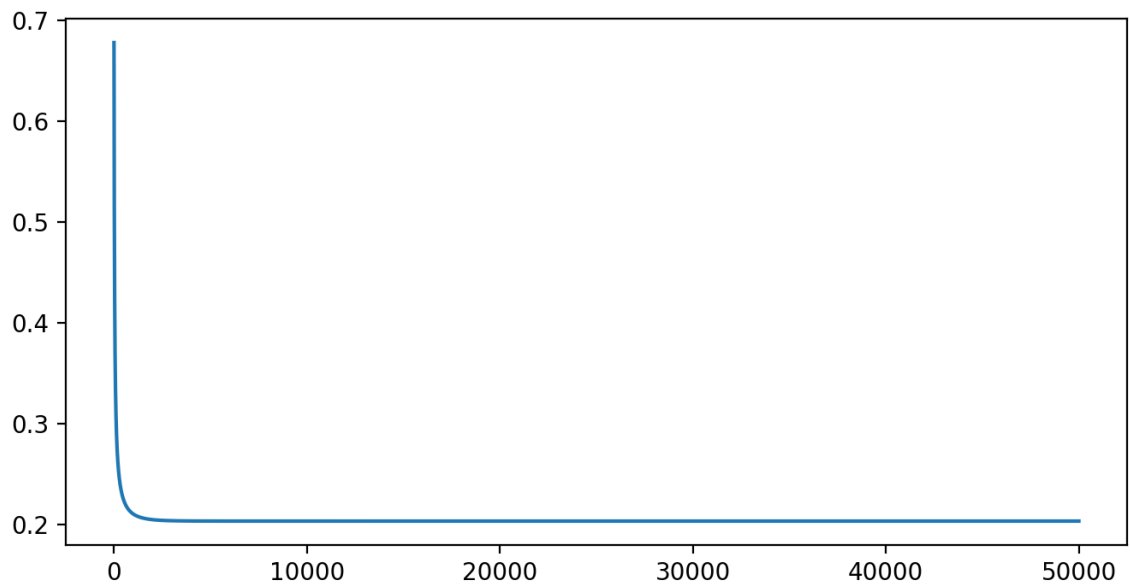
学习率为0.1，迭代次数为5000



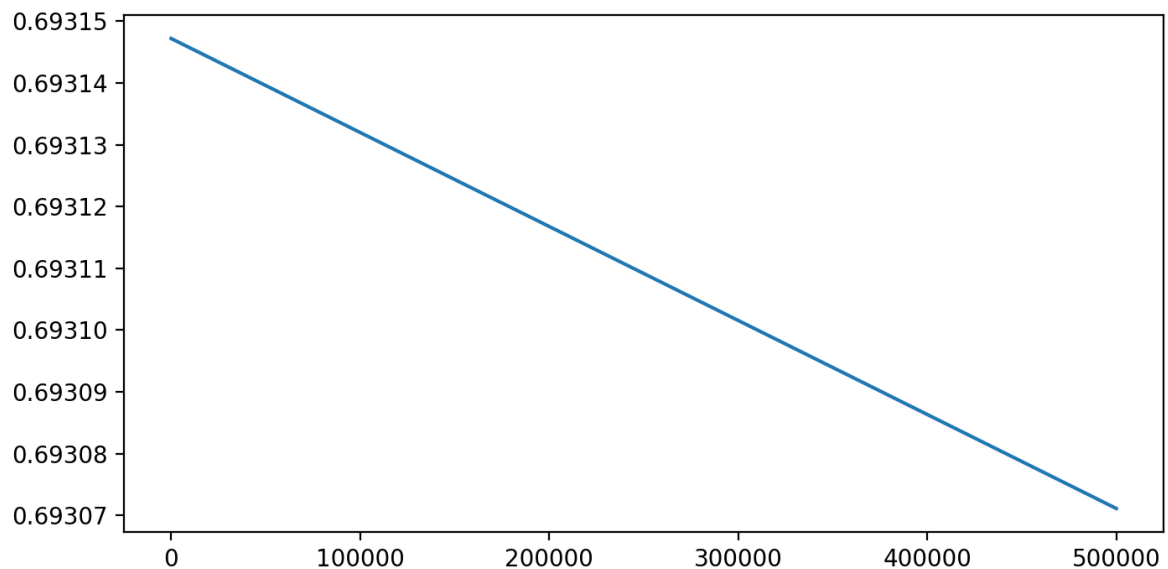
学习率为0.1，迭代次数为500



学习率为0.1，迭代次数为50000



学习率为0.000000001，迭代次数为500000



2.评测指标展示及分析（其它可分析运行时间等）

从上面的loss图可以看出，当迭代次数为50000时，学习率越低，loss降得越慢，学习率越高，loss下降得快，但是用下面的方法测出来的准确度反而降低了

```
133 data_new = pd.DataFrame(scaled_data,columns=['ones','Exam1
134 data_new['prediction']=predict(X_new,theta_new)
135 #判断预测值和观测值是否相等，相等返回True，不相等返回False
136 accuracy= (data_new['prediction']==data_new['Admitted'])
137 #查看True的个数
138 print(accuracy.value_counts())
```

问题 输出 调试控制台 终端

python + -

```
99 74.775893 89.529813 1
True 89
False 11
dtype: int64
(base) PS D:\CodeField\AI\lab\EXP7\Classification>
```

四、参考资料

理论课和实验课PPT

[逻辑回归实战练习——根据学生成绩预测是否被录取 - 知乎 \(zhihu.com\)](#)