# 机器学习原理与实践

课程报告



**题 目** 乳腺癌二分类预测

**姓 名** 全兴晓

**专 业**  计算机技术

**学 号**  2272019

**学 院**  计算机科学与工程学院

**2022 年** 10 **月** 14 **日**

概要表

|  |
| --- |
| 硬件环境（CPU/GPU）：  CPU：Intel(R) Core(TM) i5-8300H  GPU：NVIDIA GeForce GTX 1050 |
| 操作系统：  Windows 10 |
| 采用的语言、库、开发工具：  语言：Python，库：numpy、matplotlib、pandas，开发工具：VsCode |
| 任务描述/问题定义：  该作业的数据集是对多个维度的乳腺癌特征记录，使用该数据集实现对乳腺癌良性或恶心的二分类预测，数据集内容如下  # Attribute Domain  -- -----------------------------------------  1. Sample code number id number  2. Clump Thickness 1 - 10  3. Uniformity of Cell Size 1 - 10  4. Uniformity of Cell Shape 1 - 10  5. Marginal Adhesion 1 - 10  6. Single Epithelial Cell Size 1 - 10  7. Bare Nuclei 1 - 10  8. Bland Chromatin 1 - 10  9. Normal Nucleoli 1 - 10  10. Mitoses 1 - 10  11. Class: (0 for benign, 1 for malignant)  由于Bare Nuclei所在的列存在未知数据，因为舍弃这列。由于第一列为样本的id信息并不会对Class列造成影响，因此也删去这一列，将前8列作为特征输入X，最后一列作为实际的分类值。使用交叉熵代价函数，画出代价函数、算出精度、召回率、f1值等性能指标。 |
| 数据集及来源（相关链接）：  http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Original) |
| 最终设置的超参数：  Learning rate：0.1、m\_train=500：500，iteration=500 |
| 模型的效果（如准确率等与任务相关的评价指标）：  训练集：分类错误个数:26、TP:192,FP:12,TN:332,FN:14、recall:0.932、precision:0.941,f1:0.937  测试集：分类错误个数:2、TP:35,FP:2,TN:112,FN:0、recall:1.0,precision:0.946,f1:0.972 |

1. 源代码及注释

1.1 导入需要的库

import numpy as np

import matplotlib.pyplot as plt

import pandas。

1.2 特征缩放

def Feature\_scaling(data,d):#data：数据集,d：维度

x\_mean=np.mean(data[:,:d],axis=0)#求平均值

x\_std=np.std(data[:,:d],axis=0,ddof=1)#求出方差

data[:,:d]=(data[:,:d]-x\_mean)/x\_std#进行吗标准化

1.3 划分数据集

def split\_dataset(data,m\_train,m\_all,d):

#dada：数据集，m\_train:训练集大小,m\_all：数据集大小，d：维度

x\_train=data[:m\_train,:d].T #划分训练集的特征输入

y\_train=data[:m\_train,d].reshape(1,-1) #划分训练集的结果

x\_test=data[m\_train:,:d].T#划分测试集的特征输入

y\_test=data[m\_train:,d].reshape(1,-1) #划分测试集的结果

train\_set=data[:m\_train,:] #划分训练集

return x\_train,y\_train,x\_test,y\_test,train\_set#返回划分好的结果

1.4 对数据进行训练

def Train(iteration,learning\_rate,batch\_size,train\_set):#iteration：迭代次数,learning\_rate：学习率,batch\_size：批长,train\_set：训练集

rng=np.random.default\_rng(1)#设置随机种子

w=np.zeros((1,d)).reshape(-1,1)#设置权重w的初始值

b=0#设置偏移b的初始值

cost\_save=[]#用来保存代价函数的列表

for i in range(iteration):#循环迭代

rng.shuffle(train\_set)#对训练集进行随机打乱

for start\_sample in range(0,m\_train,batch\_size):# start\_sample间隔步长

batch\_sample=min([batch\_size,m\_train-start\_sample])

‘’’start\_sample的长度等于二者的最小值，防止最后一次取训练集剩余的长度小于步长’’’

v=np.ones((1,batch\_sample)).reshape(1,-1)

X=train\_set[start\_sample:batch\_sample+start\_sample,:d].T#划分每批的训练样本

Y=train\_set[start\_sample:batch\_sample+start\_sample,d].T

z=np.dot(w.T,X)+b

y\_hat=1./(1+np.exp(-z))

y\_hat\_ba=1-y\_hat

y\_train\_ba=1-Y

e=y\_hat-Y

w=w-learning\_rate\*np.dot(X,e.T)/batch\_size

b=b-learning\_rate\*np.dot(v,e.T)/batch\_size

cost=-(np.dot(np.log(y\_hat),Y.T)+np.dot(np.log(y\_hat\_ba),y\_train\_ba.T))/batch\_size

'''cost值域为0-2'''

cost\_save.append(cost.item(0))

return w,b,cost\_save

1.5 对数据集预测

def Prediction(w,b,x,y,m\_train):#w：权重，b：偏移，x：特征输入，y：实际结果，m\_train：输入的长度

z=(np.dot(w.T,x)+b)

y\_hat=1/(1+np.exp(-z))

z=z>=0

errors=np.sum(np.abs(z-y))

print("分类错误个数：",errors.astype(int))

#训练数据集上的混淆矩阵

FP=np.sum(np.logical\_and(y==0,z==1))

TP=np.sum(np.logical\_and(y==1,z==1))

TN=np.sum(np.logical\_and(y==0,z==0))

FN=np.sum(np.logical\_and(y==1,z==0))

print(f'TP:{TP},FP:{FP},TN:{TN},FN:{FN}')

recall=TP/(TP+FN)#召回率

precision=TP/(TP+FP)#精度

f1=2\*recall\*precision/(recall+precision)

print(f'recall:{recall:.3},precision:{precision:.3},f1:{f1:.3}')

1.6 导入数据进行训练分类

df=pandas.read\_csv('breastCancer.csv')

df=df.drop(labels='bare\_nucleoli' ,axis=1)#删掉标签名为bare\_nucleoli的列，此列有脏数据

df=df.drop(labels='id' ,axis=1)#删掉标签名为id的一列，此列没有用处

data=np.array(df,dtype = float)

m\_all=np.shape(data)[0]#计算出数据样本总数

d=np.shape(data)[1]-1#计算出维度

#特征缩放

Feature\_scaling(data,d)

#设定参数

m\_train=500

m\_test=m\_all-m\_train

iteration=500

learning\_rate=0.1

batch\_size=m\_train

x\_train,y\_train,x\_test,y\_test,train\_set=split\_dataset(data,m\_train,m\_all,d)#划分数据集

w,b,cost\_saved=Train(iteration,learning\_rate,batch\_size,train\_set=train\_set)#训练参数

print('训练集分类预测')

Prediction(w,b,x\_train,y\_train,m\_train)#训练集分类预测

print('测试集分类预测')

Prediction(w,b,x\_test,y\_test,m\_test)#测试集分类预测

#画出图像

plt.ylabel('Costs')

plt.xlabel('Iterations')

plt.title('Learning rate = ' + str(learning\_rate))

plt.plot(range(1,len(cost\_saved)+1),cost\_saved,'r')

plt.show()

1. 训练曲线（Loss-Epoch）

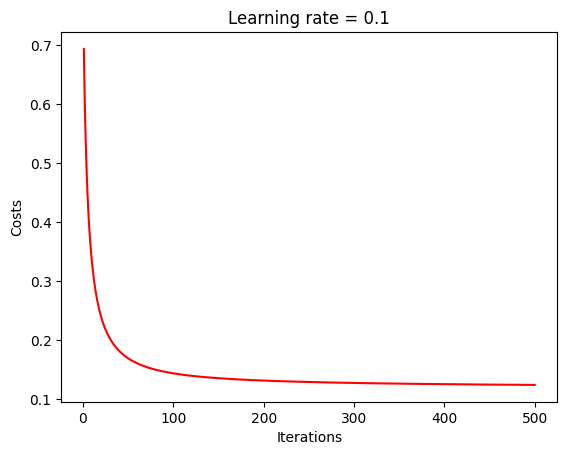
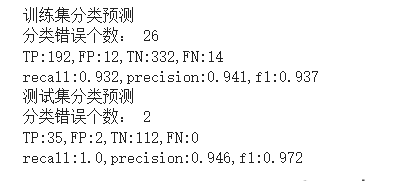


图2.1 训练曲线

1. **运行或运行结果截图**

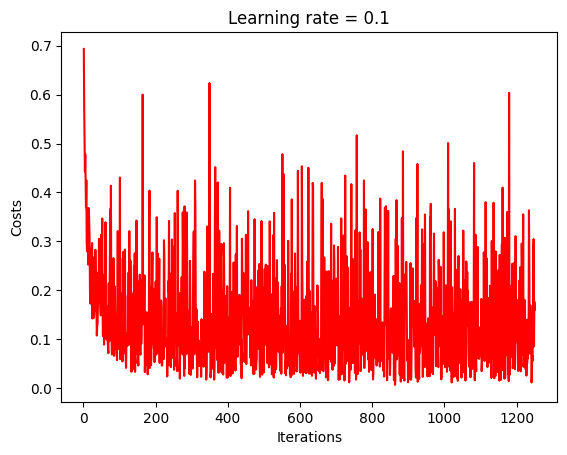


1. **总结**

在做整个作业的时候我也遇到了许多问题，首先第一个问题是数据集。当我对数据集进行特征缩放的时候程序报错提示我数据里有字符串，检查发现原始数据集的“bare\_nucleol”列的某几行会出现“？”的数据，可能是数据采集人员当时没有记录到真实数据因此用“？”字符填充，并且该列的其他数据全为字符串型，因此我舍弃了该列。

第二个问题出现在我特征缩放的时候，由于我不知道特征缩放出来的数据是错误的导致我最后算代价函数里出现了“INF”即出现了无穷大，当我不断向前排查时发现经过特征缩放后的训练集x\_train里的数据全变成了非0即1，继续排查发现x\_mean均值与x\_std方差结果都没有问题，但是当执行“data[:,:d]=(data[:,:d]-x\_mean)/x\_std”语句之后就发生了错误，最后的解决办法是将原来的语句” data=np.array(df)”替换为“data=np.array(df,dtype = float)”，即指定数据集里面的元素为float类型就解决了这个问题，我认为造成该错误的原因是在执行“data[:,:d]=(data[:,:d]-x\_mean)/x\_std”语句时进行了整除操作，所以才导致最后的x\_train元素里出现了非0即1的结果。

第三个问题是在进行二分类任务时我使用了mini\_batch小批梯度下降法，当批长为20迭代训练50次时代价函数的图像如下：

 图4.1 小批梯度下降法代价函数图像

该图像并没有像线性回归使用小批梯度下降法的代价函数图像那样平滑并最终收敛，我又对酒驾检测实验同样使用mini\_batch随机梯度下降法训练同样也得到了不收敛的代价函数图像，同时我考虑过是否是因为使用了交叉熵代价函数而导致此情况，我又对均方误差代价函数使用mini\_batch小批梯度下降法进行训练，发现其同样不收敛。因此我得出结论：在逻辑回归进行二分类预测时，由于使用了sigmoid函数必然会使批长小于训练集长度的小批梯度下降法不收敛。因此真对逻辑回归二分类问题中，建议使用批梯度下降法。