

**计算机与信息 学院实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验课程： | Python机器学习实验 | | | | |
| 实验编号： | 实验二 | | | | |
| 实验名称： | **softmax回归模型** | | | | |
| 实验人员： | 学号 | 18111207248 | | | |
| 姓名 | 吴钰 | | | |
| 班级 | 创新班 | | | |
| 实验日期： | 2021.3.28 | | | | |
| 实验室： | 2060301 | | | | |
|  |  | | | | |
| 实验评价： |  | | | | |
| 实验成绩： | |  | 评价日期： |  |
|  | 指导教师： | |  | | |

# 实验目的

# 请编程实现Softmax回归模型在二分类问题上的应用，提交开题报告、训练集和测试集

# 实验要求

# 请编程实现Softmax回归模型在二分类问题上的应用，提交开题报告、训练集和测试集

# 实验内容

# 请编程实现Softmax回归模型在二分类问题上的应用，提交开题报告、训练集和测试集

# 实验过程及结果

代码：

# -\*- coding: utf-8 -\*-

"""

Created on Mon Mar 29 16:03:13 2021

@author: wuyu

"""

import numpy as np

import random as rd

import matplotlib.pyplot as plt

def load\_weights(weights\_path):

'''导入训练好的Softmax模型

input: weights\_path(string)权重的存储位置

output: weights(mat)将权重存到矩阵中

m(int)权重的行数

n(int)权重的列数

'''

f = open(weights\_path)

w = []

for line in f.readlines():

w\_tmp = []

lines = line.strip().split("\t")

for x in lines:

w\_tmp.append(float(x))

w.append(w\_tmp)

f.close()

weights = np.mat(w)

m, n = np.shape(weights)

return weights, m, n

def load\_data(num, m):

'''导入测试数据

input: num(int)生成的测试样本的个数

m(int)样本的维数

output: testDataSet(mat)生成测试样本

'''

testDataSet = np.mat(np.ones((num, m)))

for i in range(num):

testDataSet[i, 1] = rd.random() \* 6 - 3#随机生成[-3,3]之间的随机数

testDataSet[i, 2] = rd.random() \* 15#随机生成[0,15]之间是的随机数

return testDataSet

def predict(test\_data, weights):

'''利用训练好的Softmax模型对测试数据进行预测

input: test\_data(mat)测试数据的特征

weights(mat)模型的权重

output: h.argmax(axis=1)所属的类别

'''

h = test\_data \* weights

return h.argmax(axis=1)#获得所属的类别

def save\_result(file\_name, result):

'''保存最终的预测结果

input: file\_name(string):保存最终结果的文件名

result(mat):最终的预测结果

'''

f\_result = open(file\_name, "w")

m = np.shape(result)[0]

for i in range(m):

f\_result.write(str(result[i, 0]) + "\n")

f\_result.close()

def draw(weight, file\_name):

x0List = []

y0List = []

x1List = []

y1List = []

x2List = []

y2List = []

x3List = []

y3List = []

f = open(file\_name, 'r')

for line in f.readlines():

lines = line.strip().split()

if lines[2] == '0':

x0List.append(float(lines[0]))

y0List.append(float(lines[1]))

elif lines[2] == '1':

x1List.append(float(lines[0]))

y1List.append(float(lines[1]))

elif lines[2] == '2':

x2List.append(float(lines[0]))

y2List.append(float(lines[1]))

else:

x3List.append(float(lines[0]))

y3List.append(float(lines[1]))

fig = plt.figure()

ax = fig.add\_subplot(111)

ax.scatter(x0List, y0List, s=10, c='pink')

ax.scatter(x1List, y1List, s=10, c='red')

ax.scatter(x2List, y2List, s=10, c='blue')

ax.scatter(x3List, y3List, s=10, c='green')

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

# 1、导入Softmax模型

print ("1.加载模型 ------------")

w, m , n = load\_weights("weights")

# 2、导入测试数据

print ("2.加载数据------------")

test\_data = load\_data(4000, m)

# 3、利用训练好的Softmax模型对测试数据进行预测

print ("3.利用训练好的Softmax模型对测试数据进行预测 ------------")

result = predict(test\_data, w)

# 4、保存最终的预测结果

print ("4.保存预测结果 ------------")

save\_result("result", result)

# 4，绘制图像

draw(w, 'D:\\大三（下）专业课\\机器学习实验\\18111207248吴钰（实验二）[\\softmax\\softInput.txt](file:///\\softmax\\softInput.txt)')

结果截图：









