黄冈师範学院
HUANGGANG NORMAL UNIVERSITY

人工智能与机器学习

Artificial Intelligence and Machine Learning

章节：实验3-基于KNN的手写数字识别

教师：刘重

学院：计算机学院

厚德 博学 力行 致远

一、实验目的

- 1） 掌握KNN算法的原理
- 2） 了解手写数字识别的原理与过程
- 3） 利用KNN算法实现手写数字识别

- （1）收集数据：提供文本文件；
- （2）准备数据：将图像格式转换为分类器使用的List格式；
- （3）测试算法：编写函数使用提供的部分数据集作为测试样本，测试样本与非测试样本的区 别在于测试样本是已经完成分类的数据，如果预测分类和实际类别不同，则标记为一个 错误。

三、实验原理

- k-近邻算法
- ①原理：存在一个样本数据集合，也称作训练样本集，并且样本集中每一个数据都存在标签，即我们知道样本集中每一个数据与所属分类的对应关系。输入没有标签的新数据后，将新的数据的每一个特征进行比较；然后算法提取样本集中特征最相近数据（最邻近）的分类标签。一般来说，我们只选择样本数据集中前k个最相似的数据，这就是k-邻近算法中k的出处。通常k是不大于20的整数。最后选择k个最相似数据中出现次数最多的分类，最为新数据的分类。
- ②优点：精度高，对异常值不敏感。简单易用，相比其他算法，KNN算是比较简洁明了的算法。即使没有很高的数学基础也能搞清楚它的原理。预测效果好。
- ③缺点：计算复杂度高，对内存要求较高，因为该算法存储了所有训练数据，空间复杂度高。
- ④计算距离：通过测量新的测试数据和样本数据之间特征值的距离，选出最相似的前k个样本数据。计算距离的方法有：欧式距离，曼哈顿距离等等，此处不做详述。
- k-近邻算法是机器学习算法中有监督学习算法的一种，主要用于分类：适用于数据集较小的数据的分类；数据集大可用深度学习神经网络进行分类。

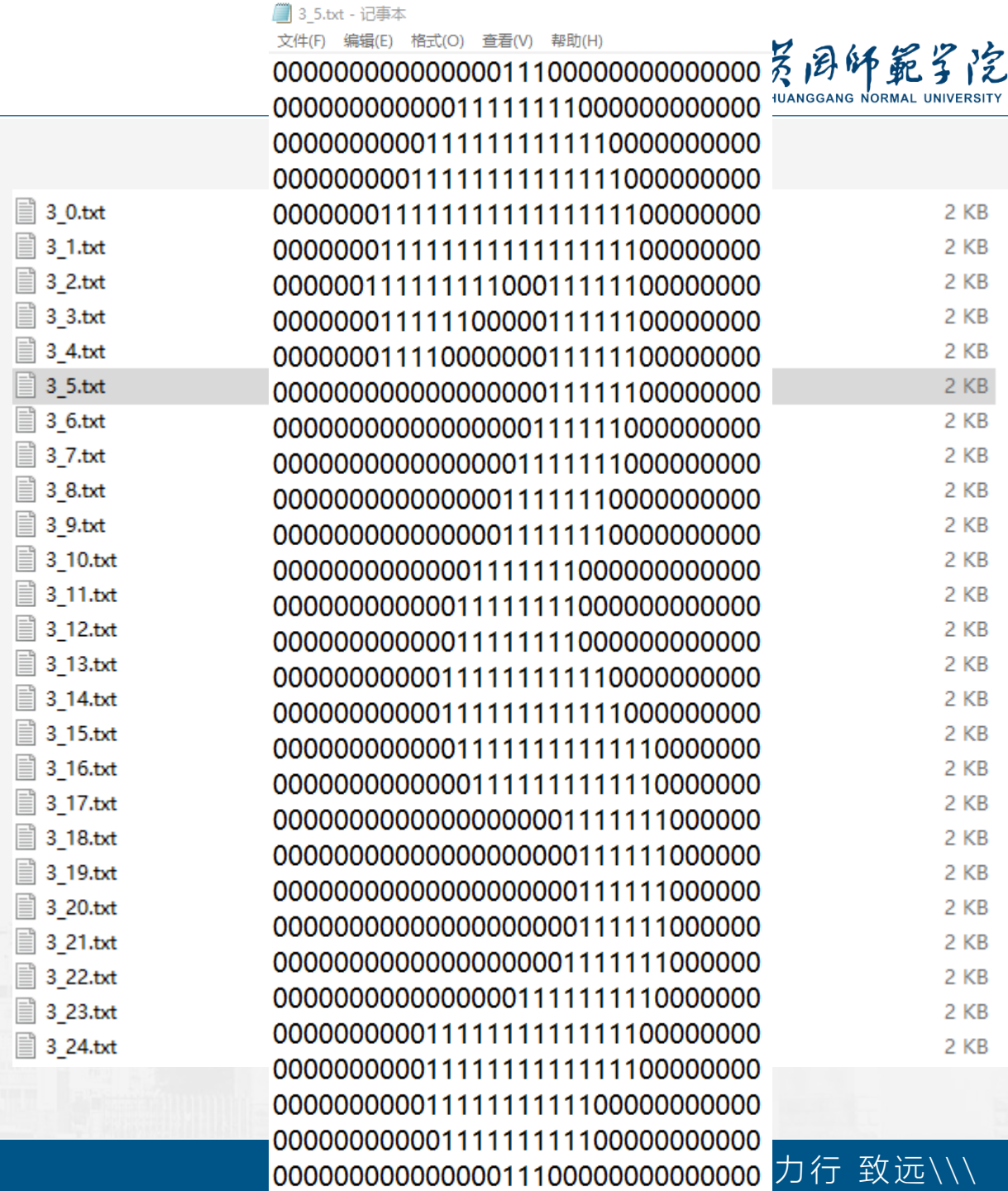
四、实验步骤

1.收集数据：提供文本文件

数据集包括两部分：

- 一部分是训练数据集，共有1934个数据；
- 另一部分是测试数据集，共有946个数据。

两个数据集中所有命名格式是统一的，例如“3_12.txt”,表示数字5 的第12个样本，这样是为了方便提取出样本的真实标签。



2.准备数据：将图像转换为测试向量

- 将图像格式化处理为一个向量，把每一个32x32的二进制图像矩阵转换为1×1024的向量。编写函数img2vector,将图像转换为向量：该函数创建1×1024的Numpy数组，然后打开给定的文件，循环读出文件的前32 行，并将每行的头32个字符值存储在Numpy数组中，最后返回数据。

```
"""
函数说明:将32x32的二进制图像转换为1x1024向量
"""
def img2vector(filename):
    #创建1x1024零向量
    returnVect = np.zeros((1, 1024))
    #打开文件
    fr = open(filename)
    #按行读取
    for i in range(32):
        #读一行数据
        lineStr = fr.readline()
        #每一行的前32个元素依次添加到returnVect中
        for j in range(32):
            returnVect[0, 32*i+j] = int(lineStr[j])
    #返回转换后的1x1024向量
    return returnVect
```

3.测试算法：使用k-近邻算法识别手写数字

- 编写handwritingClassTest()函数。
- 将trainingDigits目录中的文件内容存储在列表中，然后可以得到目录中有多少文件，并将其存储在变量m中。接着创建一个m行1024列的训练矩阵，该矩阵的每行数据存储一个图像。我们可以从文件名中解析出分类数字。如9_45.txt的分类是9，它是数字9 的第45个实例。然后我们可以将类代码存储在hwLabels向量中，使用img2vector函数载入图像。接着对testDigits目录中的文件执行相似的操作，不同之处是我们并不将这个目录下的文件载入矩阵中，而是使用KNN.predict()函数测试该目录下的每个文件。

```
46  函数说明:手写数字分类测试
47  """
48
49  def handwritingClassTest():
50      #训练集的Labels
51      hwLabels = []
52      #返回trainingDigits目录下的文件名
53      trainingFileList = listdir('trainingDigits')
54
55      #返回文件夹下文件的个数
56      m = len(trainingFileList)
57      #初始化训练的Mat矩阵,训练集
58      trainingMat = np.zeros((m, 1024))
59      #从文件名中解析出训练集的类别
60      for i in range(m):
61          #获得文件的名字
62          fileNameStr = trainingFileList[i]
63          #获得分类的数字
64          classNumber = int(fileNameStr.split('_')[0])
65          #将获得的类别添加到hwLabels中
66          hwLabels.append(classNumber)
67          #将每一个文件的1x1024数据存储到trainingMat矩阵中
68          trainingMat[i,:] = img2vector('trainingDigits/%s' % (fileNameStr))
69      #构建kNN分类器
70      neigh =KNN(n_neighbors = 3, algorithm = 'auto')
71      #拟合模型, trainingMat为训练矩阵,hwLabels为对应的标签
72      neigh.fit(trainingMat, hwLabels)
73      #返回testDigits目录下的文件列表
74      testFileList = listdir('testDigits')
75      #错误检测计数
76      errorCount = 0.0
77      #测试数据的数量
78      mTest = len(testFileList)
79      #从文件中解析出测试集的类别并进行分类测试
80      showflag= 1 #只展示第一个分错的
81      for i in range(mTest):
82          #获得文件的名字
83          fileNameStr = testFileList[i]
84          #获得分类的数字
85          classNumber = int(fileNameStr.split('_')[0])
86          #获得测试集的1x1024向量,用于训练
87          vectorUnderTest = img2vector('testDigits/%s' % (fileNameStr))
88          #获得预测结果
89          classifierResult = neigh.predict(vectorUnderTest)
90          print("分类返回结果为%d\t真实结果为%d" % (classifierResult, classNumber))
91          if(classifierResult != classNumber):
92              errorCount += 1.0
93              #一旦分类错误就显示错误结果，×掉绘图框后继续预测
94              imageDir = txt2image('testDigits/%s' % (testFileList[i]))
95              print(imageDir)
96              plt.imshow(imageDir)
97              plt.title('第%s个分类错误，testFileList/%s,真实结果为：%s，预测结果为：%s'\
98                        % (showflag,testFileList[i],classNumber,classifierResult))
99              plt.show()
100             showflag += 1
101
102     print("总共错了%d个数据\n错误率为%f%%" % (errorCount, errorCount/mTest * 100))
```

五、实验结果

- 测试handwritingClassTest()函数:

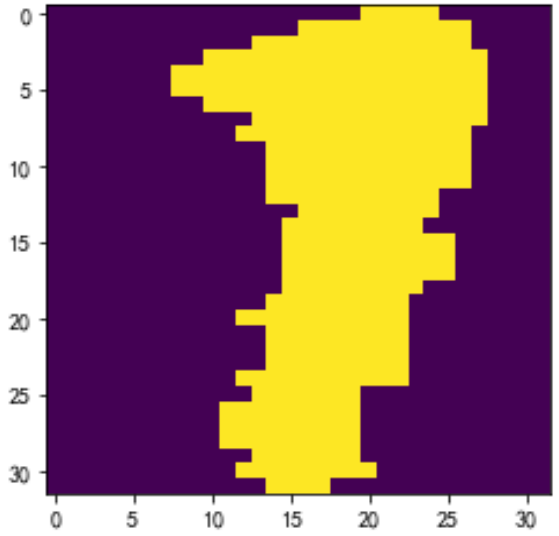
```
"""
函数说明:main函数
"""
#开始时间
start = time.perf_counter()
#测试handwritingClassTest()的输出结果
handwritingClassTest()
end = time.perf_counter()
print("运行耗时: %ds" %(end-start))
```

- 测试结果:

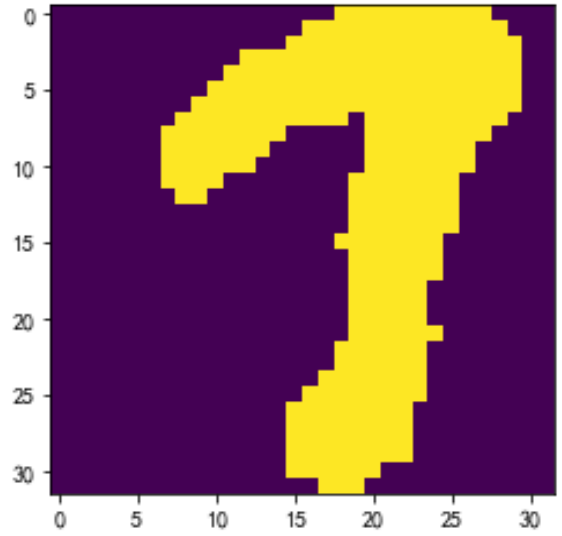
```
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
分类返回结果为9      真实结果为9
总共错了12个数据
错误率为1.268499%
运行耗时: 4s
```

运行结果中的12个分类错误

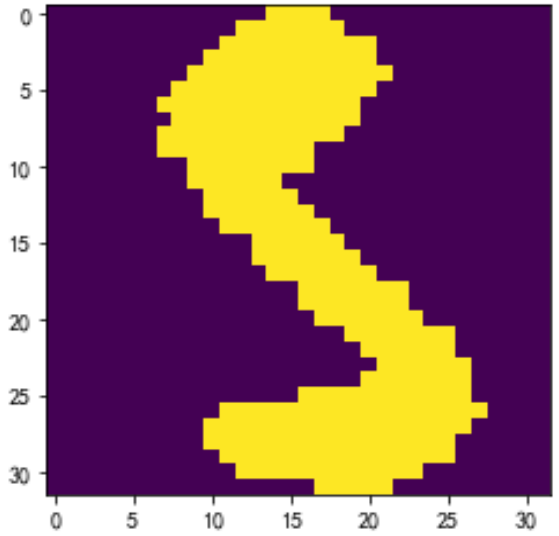
第1个错误分类，testFileList/1_86.txt,真实结果为：1，预测结果为：[7]



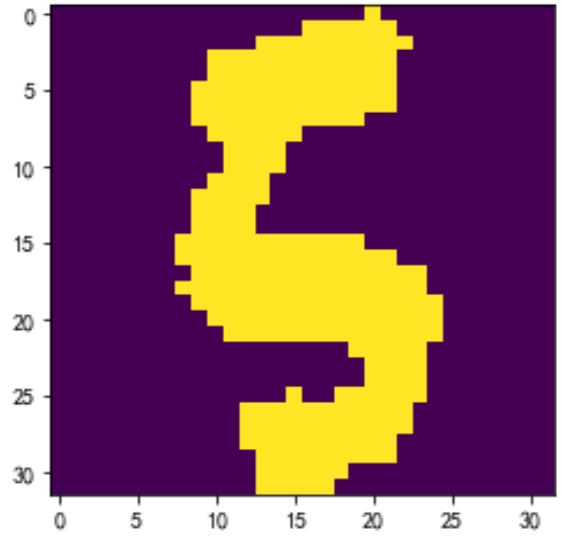
第2个错误分类，testFileList/3_11.txt,真实结果为：3，预测结果为：[9]



第3个错误分类，testFileList/5_42.txt,真实结果为：5，预测结果为：[3]

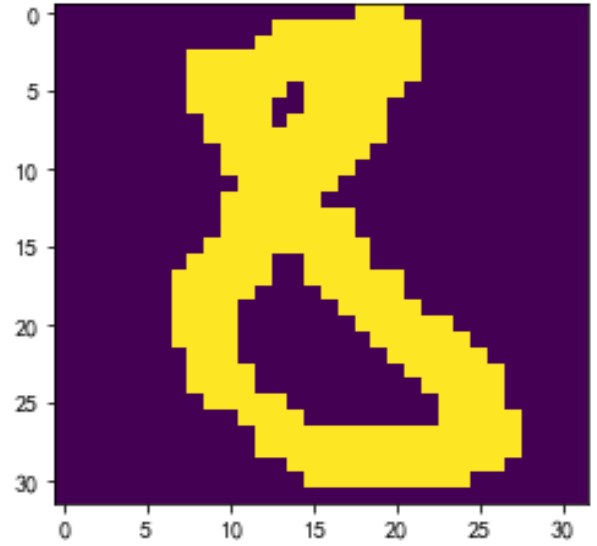


第4个错误分类，testFileList/5_43.txt,真实结果为：5，预测结果为：[6]

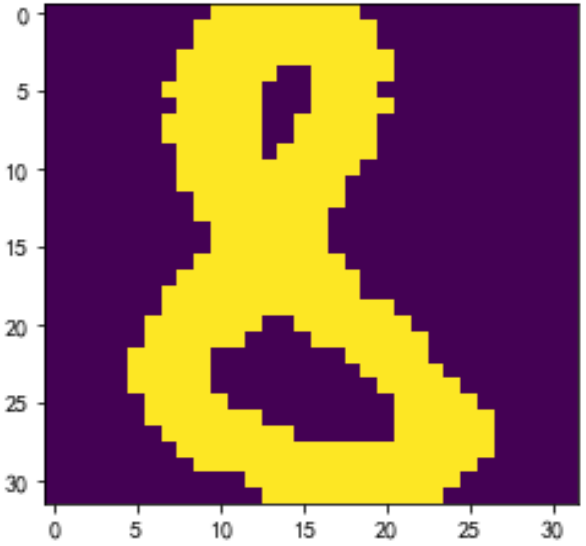


运行结果中的12个分类错误

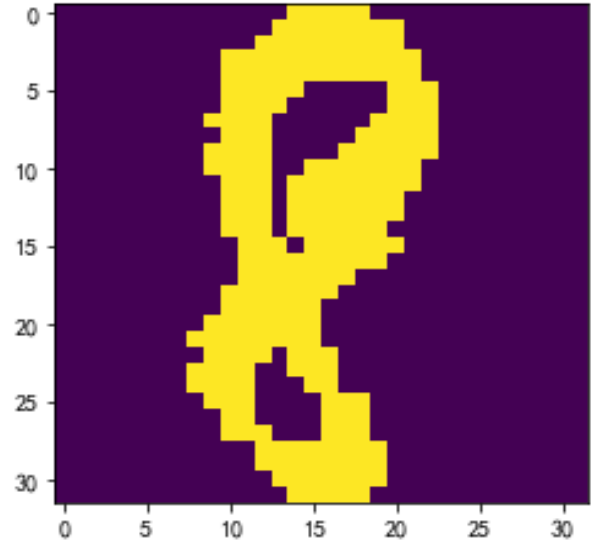
第5个错误分类，testFileList/8_11.txt，真实结果为：8，预测结果为：[6]



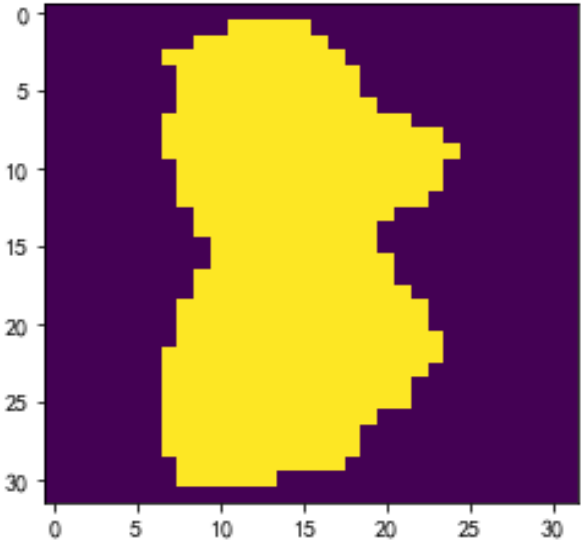
第6个错误分类，testFileList/8_23.txt，真实结果为：8，预测结果为：[3]



第7个错误分类，testFileList/8_36.txt，真实结果为：8，预测结果为：[1]

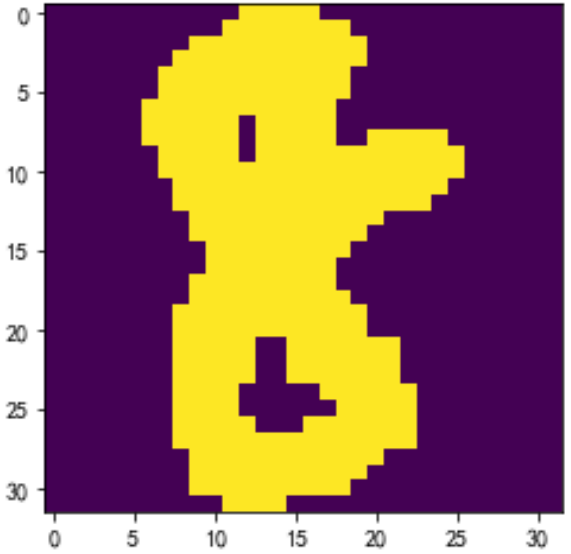


第8个错误分类，testFileList/8_45.txt，真实结果为：8，预测结果为：[1]

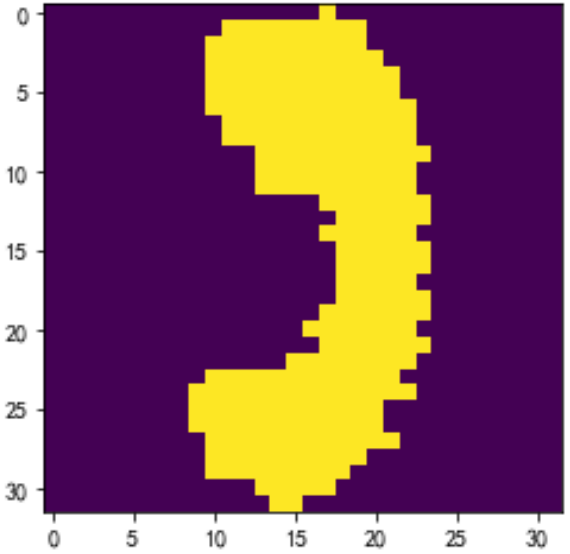


运行结果中的12个分类错误

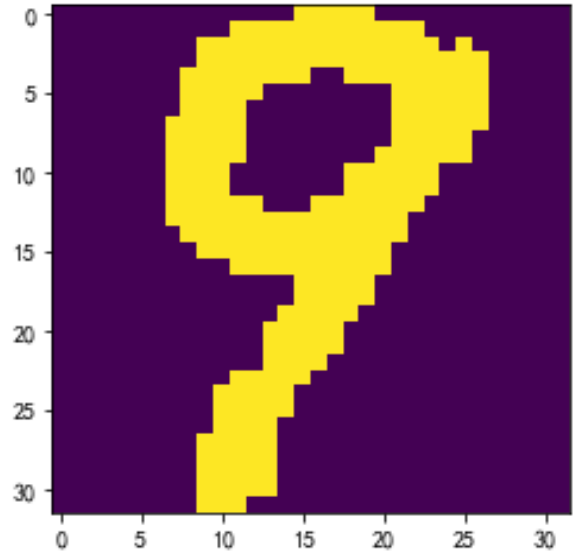
第9个错误分类，testFileList/8_68.txt,真实结果为：8，预测结果为：[1]



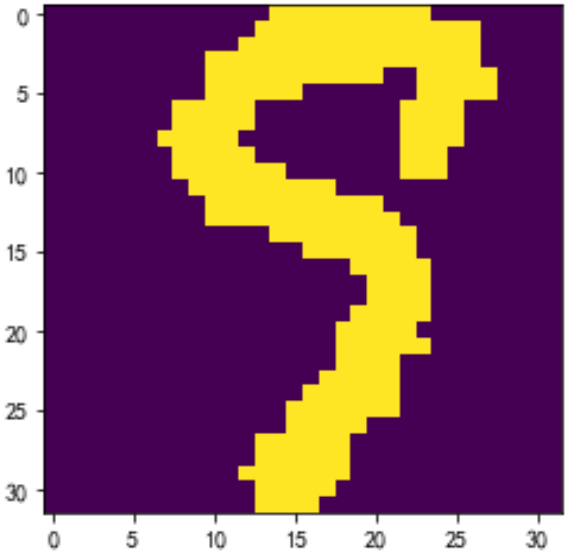
第10个错误分类，testFileList/9_14.txt,真实结果为：9，预测结果为：[1]



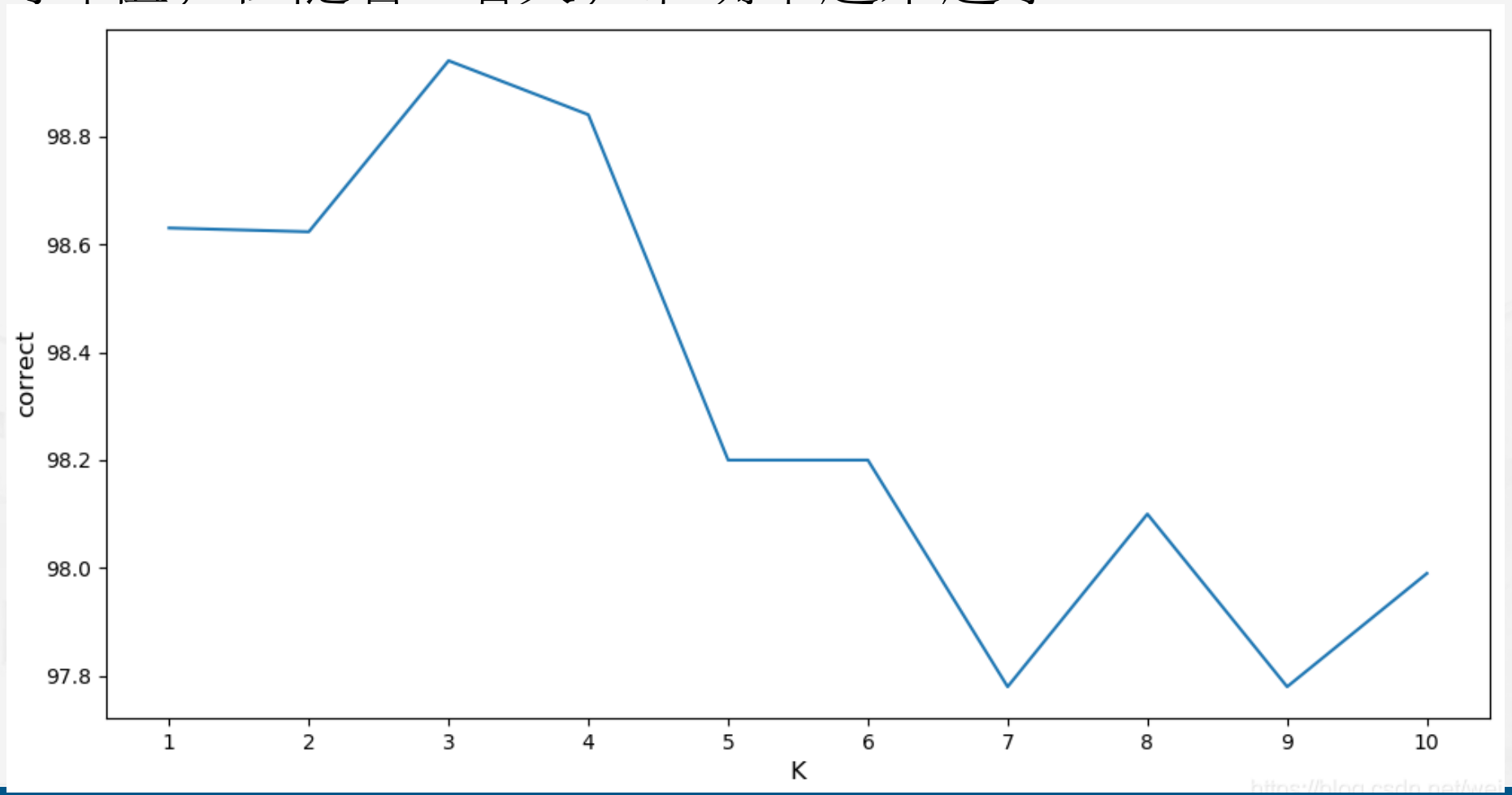
第11个错误分类，testFileList/9_60.txt,真实结果为：9，预测结果为：[7]



第12个错误分类，testFileList/9_68.txt,真实结果为：9，预测结果为：[5]



- K值：下图是K值与模型准确率的关系变化图，K = 3时，模型准确率达到峰值，但随着K增大，准确率越来越小。



六、实验报告要求

- 1、实验目的
- 2、实验内容
- 3、实验原理
- 4、实验代码
- 5、运行截图
- 6、实验小结

- 说明：每个学生都要交电子版的实验报告，命名格式：
- 01/02-XXXX（学号）-XXX（姓名）

黄冈师範学院
HUANGGANG NORMAL UNIVERSITY

Q&A

厚德 博学 力行 致远