

华东师范大学计算机科学与技术实验报告

实验课程：数据挖掘	年级：2018	实验成绩：
指导教师：兰曼	姓名：董辰尧	提交作业日期：2021/6/8
实践编号：2	学号：10185102144	实践作业编号：2

华东师范大学计算机科学与技术实验报告

- 一、实验名称：手写数字识别
- 二、实验目的
- 三、实验内容
 - 3.1训练集处理
 - 3.1.1观察数据格式
 - 3.1.2所有文件的读取
 - 3.1.3单个文件的读取
 - 3.2测试集读取
 - 3.3KNN
 - 3.3.1训练
 - 3.3.2预测
 - 3.4写入预测文件
- 四、实验结果及其分析
 - 4.1预测结果截图
- 五、问题讨论（实验过程中值得交待的事情）
 - 5.1结果乱序
 - 5.2图片转txt
 - 5.3KNN
- 六、结论

一、实验名称：手写数字识别

识别手写数字


















二、实验目的

掌握本地数据的读写，实现手写数字识别

三、实验内容

3.1训练集处理

3.1.1观察数据格式

 0_0.txt	2021/6/8 13:06	文本文档	2 KB
 0_1.txt	2021/6/8 13:06	文本文档	2 KB
 0_2.txt	2021/6/8 13:06	文本文档	2 KB
 0_3.txt	2021/6/8 13:06	文本文档	2 KB
 0_4.txt	2021/6/8 13:06	文本文档	2 KB
 0_5.txt	2021/6/8 13:06	文本文档	2 KB
 0_6.txt	2021/6/8 13:06	文本文档	2 KB
 0_7.txt	2021/6/8 13:06	文本文档	2 KB
 0_8.txt	2021/6/8 13:06	文本文档	2 KB
 0_9.txt	2021/6/8 13:06	文本文档	2 KB
 0_10.txt	2021/6/8 13:06	文本文档	2 KB
 0_11.txt	2021/6/8 13:06	文本文档	2 KB
 0_12.txt	2021/6/8 13:06	文本文档	2 KB
 0_13.txt	2021/6/8 13:06	文本文档	2 KB
 0_14.txt	2021/6/8 13:06	文本文档	2 KB
 0_15.txt	2021/6/8 13:06	文本文档	2 KB
 0_16.txt	2021/6/8 13:06	文本文档	2 KB

发现训练集的文件名称是“答案_文件编号”的形式，在读取数据的时候可以考虑把这两部分分开读。

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
000000000000000111100000000000000000
000000000000000111111100000000000000
000000000000011111111110000000000000
000000001111111111111110000000000000
000000001111111011111110000000000000
000000011111110000001111000000000000
000000011111110000000011100000000000
000000011111110000000011110000000000
000000011111110000000011110000000000
000000011111110000000001110000000000
000000011111110000000001110000000000
000000011111110000000000011100000000
000000011111110000000000001110000000
000000011111110000000000001110000000
000000011111110000000000001110000000
000000011111110000000000001110000000
000000011111110000000000001111000000
000000011110110000000000011110000000
000000011110000000000000011110000000
000000011110000000000000011110000000
0000000111100000000000000111110000000
000000011110000000000111111000000000
000000001110000000111111100000000000
000000001111000111111111000000000000
000000000111111111111110000000000000
```

0000000001111111111111111111111100000000000

所有的数据形式是32*32的数字矩阵，所以在读取单个文件的数据的时候要注意数据的格式。

3.1.2所有文件的读取

```
1 # 获取数据文件
2 fileList = os.listdir('./data/trainingDigits/')
3
4 # 定义数据标签列表
5 trainingIndex = []
6 # 添加数据标签
7 for filename in fileList:
8     trainingIndex.append(int(filename.split('_')[0]))
9
10 # 定义矩阵数据格式
11 trainingData = np.zeros((len(trainingIndex),1024))
12 print(trainingData.shape)#(1498, 1024)
```

这里trainingData训练集的矩阵是（文件数1498，单个文件的数字个数32*32）

3.1.3单个文件的读取

```
1 # 获取矩阵数据
2 index = 0
3 for filename in fileList:
4     with open('./data/trainingDigits/%s'%filename, 'rb') as f:
5
6         # 定义一个空矩阵
7         vect = np.zeros((1,1024))
8
9         # 循环32行
10        for i in range(32):
11            # 读取每一行数据
12            line = f.readline()
13
14            # 遍历每行数据索引 line[j] 即为数据
15            for j in range(32):
16                vect[0,32*i+j] = int(line[j])
17
18        trainingData[index,:] = vect
19        index+=1
```

这里单个文件每次读一行，读取完毕后把读出来的数据矩阵vect填入上一步定义好的trainingData训练集矩阵中去。

3.2测试集读取

这个读取步骤和上一步相同。但是需要注意的是测试集的文件名称只有序号，由于提交的预测结果是“序号+结果”的形式，所以在这里略有不同，需要得到序号的列表。

3.3KNN

3.3.1训练

```
1 from sklearn.neighbors import KNeighborsClassifier
2 # 定义k为3个，即寻找最近的3个邻居
3 knn = KNeighborsClassifier(n_neighbors=3)
4 # 训练数据
5 knn.fit(trainingData,trainingIndex)
```

这里直接导包进行训练

3.3.2预测

```
1 %%time
2 # 预测数据
3 predict_data = knn.predict(testData)
4 print(predict_data)
```

这里直接打出预测数据

```
[6 7 4 8 5 7 9 2 9 4 3 2 5 0 9 4 7 9 7 4 4 1 9 6 0 3 0 5 2 3 4 0 5 4 2 8 6
 1 2 0 2 9 8 6 0 2 9 3 3 7 2 9 5 9 9 0 2 4 3 1 6 2 7 3 3 5 6 2 5 8 6 5 2 5
 0 1 7 6 8 6 4 8 8 7 6 8 0 8 2 2 5 8 7 6 9 2 6 4 4 5 4 6 9 5 7 1 4 5 2 4 1
 0 7 7 8 1 8 0 9 9 9 0 1 6 3 9 7 1 2 8 6 8 7 5 4 8 8 6 4 5 4 8 5 6 7 7 1 2
 6 3 7 5 8 1 6 6 5 6 3 6 4 1 1 9 5 1 4 3 6 1 7 7 3 9 7 4 7 2 4 3 8 2 6 7 8
 9 3 0 6 0 9 8 8 3 3 9 2 8 5 3 1 0 5 9 4 1 7 3 0 7 0 5 8 7 7 4 0 9 0 1 0 6
 2 2 7 1 6 3 4 6 4 3 6 8 0 8 3 5 0 4 6 5 1 4 4 4 8 7 5 1 2 9 4 0 4 6 1 0 4
 1 8 6 1 7 4 8 3 6 4 6 3 4 3 0 6 2 1 2 0 0 6 5 3 4 3 2 0 4 1 4 8 2 2 9 4 9
 0 4 7 0 4 1 2 6 8 5 8 7 6 9 8 1 6 9 5 5 4 2 3 4 9 3 9 6 1 9 4 8 2 4 1 8 8
 3 4 2 9 7 7 7 1 5 2 7 9 0 9 7 9 7 5 7 7 4 1 9 1 2 0 1 5 1 2 3 1 0 6 6 5 7
 9 0 0 4 7 0 9 0 0 9 2 2 5 1 4 0 9 0 3 9 3 5 1 3 8 0 8 3 5 0 8 4 5 5 2 0 4
 7 1 0 5 4 5 2 6 2 7 8 0 8 4 5 1 4 0 1 2 5 5 0 2 7 2 4 3 5 9 3 6 2 5 7 8 0
 6 9 3 5 0 3 5 2 0 6 8 6 5 4 5 8 3 2 2 4 7 4 5 9 2 8 4 6 6 1 3 4 6 9 6 3 0
 7 1 4 8 7 1 3 6 9 5 2 4 9 8 3 4 5 8 7 7 6 9 6 2 4 1 3 1 1 6 2 4 5 3 1 0 5
 5 7 0 4 8 4 6 4 6 0 2 0 1 4 3 3 6 1 1 1 1 8 7 3 3 3 7 0 5 2 5 4 5 3 1 7 7
 4 4 0 3 1 5 9 0 7 8 6 1 7 6 5 4 5 4 0 5 7 6 7 4 3 4 9 1 4 9 7 1 2 3 1 8 5
 1 8 8 5 5 1 9 8 4 8 7 2 1 4 0 7 0 4 3 7 9 2 5 1 1 5 8 5 3 1 8 2 8 0 2 1 3
 4 3 6 2 0 7 9 7 8 5 4 3 9 8 7 6 4 5 5 3 3 1 9 7 3 6 4 7 7 9 4 5 5 7 2 4 6
 9 2 8 1 9 8 0 7 8 8 1 7 6 1 6 1 3 7 3 9 7 6 5 8 0 3 1 8 5 2 0 0 7 5 9 9 7
 6 5 5 5 1 2 0 5 0 7 9 4 2 9 8 8 6 5 7 3 4 6 8 1 3 5 2 6 3 1 7 5 1 1 4 5 2
 0 2 5 9 0 7 9 4 1 6 2 7 7 2 8 9 9 8 7 2 3 2 4 5 0 1 4 6 7 2 5 2 4 0 7 5 3
 9 9 9 3 9 5 6 1 2 2 7 3 1 6 2 4 1 5 4 9 1 0 2 8 8 2 2 1 8 8 7 9 4 0 0 6 7
 7 0 6 2 9 6 1 1 1 1 5 6 4 5 4 4 8 5 2 2 4 4 2 5 9 1 2 5 3 5 8 9 1 6 3 1 4
 7 5 9 9 8 7 4 5 6 9 5 4 0 5 3 7 6 7 9 4 4 5 1 0 8 8 2 3 8 0 2 3 1 0 1 9 3
 1 5 5 9 1 8 2 8 3 6 9 0 8 7 9 9 4 4 3 3 1 8 2 2 6 5 3 4 0 4 8 0 6 0 7 4 3
 6 3 5 9 6 1 4 4 2 9 3 0 2 4 7 7 7 6 0 5 3]
```

Wall time: 4.02 s

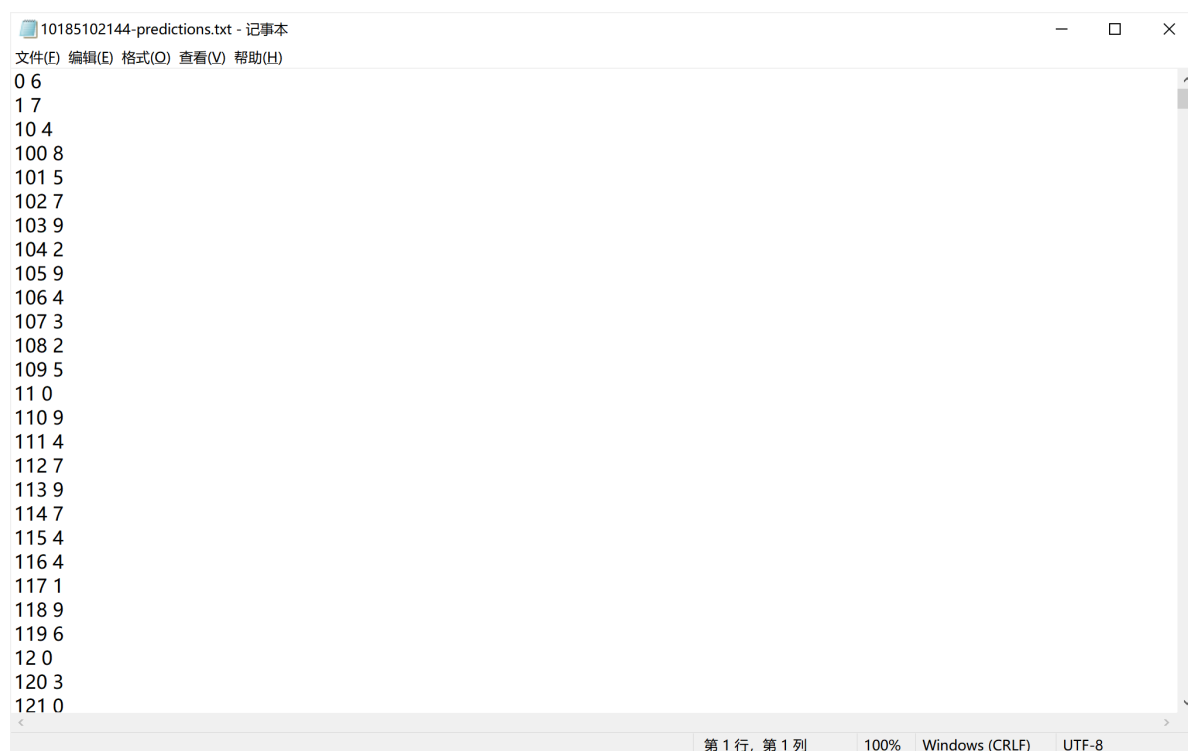
3.4写入预测文件

```
1 # 我们生成一个新的数据文本，并将所有结果写入新文件
2 with open('10185102144-predictions.txt', 'w', encoding='utf-8') as f:
3     for i in range(len(testIndex)):
4         f.write(str(testIndex[i])+' '+str(predict_data[i])+'\n')
5 f.close()
```

这里注意要求的序号 + 结果的格式

四、实验结果及其分析

4.1预测结果截图



五、问题讨论（实验过程中值得交待的事情）

5.1结果乱序

写结果的时候一开始以为自己的预测结果顺过来就对应着1、2、3、4.....结果出来之后比较好奇自己做的对不对，结果我发现正确率很低，这让我很是惊讶，因为在验证机上进行验证的时候正确率一般是97%以上，后来发现我的文件序号列表是不是按顺序来的，答案也不是按照顺序来的，所以要一一对应。

5.2图片转txt

本来我以为这次给的数据集是图片，所以我参考老师的代码，写了图片转化成txt的程序

```
1 import sys
2 from PIL import Image
3 # 将256灰度映射到16个字符上
4 def image_to_text(pixels, width, height):
5     symbols = list("01")
6     string = ""
7     for h in range(height):
```

```

8         for w in range(width):
9             rgb = pixels[w, h]
10            string += symbols[int(sum(rgb) / 3.0 / 256.0 * len(symbols))]
11            string += "\n"
12        return string
13        # 加载并调整大小
14    def load_and_resize_image(imgname, width, height):
15        img = Image.open(imgname)
16        if img.mode != 'RGB':
17            img = img.convert('RGB')
18        w, h = img.size
19        rw = width * 1.0 / w
20        rh = height * 1.0 / h
21        r = rw if rw < rh else rh
22        rw = int(r * w)
23        rh = int(r * h)
24        img = img.resize((rw, rh), Image.ANTIALIAS)
25        return img
26        # 图片转为文本
27    def image_file_to_text(img_file_path, dst_width, dst_height):
28        img = load_and_resize_image(img_file_path, dst_width, dst_height)
29        pixels = img.load()
30        width, height = img.size
31        string = image_to_text(pixels, width, height)
32        return string

```

5.3KNN

这次结果直接导包过于偷懒，所以我还完成了一个自己写的knn算法，试验过，发现准确率并不算高

```

1    def draw(X):
2        group,label = init()
3        plt.scatter(group[:,0],group[:,1])
4        plt.scatter(X[0],X[1])
5        plt.show()
6    def knn(group,k,labels,input):
7        x = group.shape[0]#行数
8        new_array = np.tile(input,(x,1))#线性代数矩阵思维
9        new_array -= group
10       new_array **= 2
11       new_array = np.sum(new_array,axis = 1)#每行相加获得的是行向量 axi=0 每列求和
12       diatance = new_array**0.5 #距离列表
13       #对距离进行排序
14       sorted_distance = np.argsort(diatance) #小到大排序返回下标的列表
15       map = {}#装k个
16       for i in range(k):
17           #sorted_distance labels k
18           #3 A 0
19           #1 A 1
20           #0 B 2
21           string = labels[sorted_distance[i]]
22           map[string] = map.get(string,0)+1
23       cnt = 0
24       for key,value in map.items():
25           if value > cnt:
26               cnt = value
27               res_string = key

```

六、结论

这次作业学习到了很多知识，从数据的挖掘，处理，利用，每一步都有更深刻的理解。