**实 验 报 告**

**课程名称：** 人工智能

**实验项目：** 利用机器学习算法进行手写汉字识别

**实验仪器：** PC机

**学 院： 计算机学院**

**专 业： 数据科学与大数据技术**

**班级姓名： 大数据1701班 张丹颖**

**学 号： 2017011760**

**日 期： 2019年6月30日**

**指导教师： 蒋玉茹**

**同组成员：**

**成 绩：**

**实验目的**

注：下文中统一用 KNN 算法描述。但是你可以选用任何你喜欢的机器学习算法完成本次实验。比如可以利用 KMeans算法进行聚类。甚至利用 CNN 进行手写汉字识别。

本实验是为了让学生能够利用机器学习中经典的 KNN 算法，进行手写汉字识别。 KNN 算法的核心思想是将一个样本在特定的空间中的 K 个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类别，并具有这个类别上的样本特性。

**问题描述**

用户自己制作“田”，“甲”，“月”，“由”，“申”等汉字的手写结果图片。

编制 KNN 程序。

利用 KNN 模型，识别新的图片中的汉字。

**实验过程指导**

第一步: 安装 python

第二步: 按照 Python 集成开发环境 PyCharm 或其他。

第三步: 构建训练集和测试集。利用“画图”创建图片，大小自定义，可以是 28 \* 28 的，利用鼠标写字，用自己“汉字拼音＋姓名全拼＋编号”作为图片名字，比如 " yuejiangyuru 一 l.jpg ”代表蒋玉茹创建的第一个”月”字的手写汉字图片。

第四步: KNN 算法编写，分为以下四步：

计算出样本数据和待分类数据的距离。

为待分类数据选择 k 个与其距离最小的样本。

统计出 k 个样本中大多数样本所属的分类。

这个分类就是待分类的数据所属的分类。

**实验报告**

1. 描述系统实现所需技术

python 编程能力，KNN 算法原理，图像转为矩阵

2. 描述算法和对应程序

本实验对应两个程序，DataDeal.py 和KNN.py

① DataDeal.py 用来将图像转换为0-1文本

算法与思想：

首先从文件夹src读入图片，将每张图像的名字存入列表list:

list=os.listdir(src)

以“/”为分割线，读取图片路径，利用图片的Image.open(path).convert("1")方法， 将图片0-1文本格式，以“.txt” 结尾，保存在 dst文件夹 中

依照上述方法，生成训练集dst和测试集s=dst1，比例自行设定

②KNN.py

思想：

1、KNN思想

KNN就是K最近邻，是一种分类算法，意思是选k个最近的邻居的意思，说的是每个样本都可以用它最接近的k个邻居来代表。在k个样本中，比重最大的那一类即可把目标归为这一类。

优点：不用训练

缺点：该算法在分类时有个主要的不足是，当样本不平衡时，K临近占比概率影响结果、计算量大

2、KNN如何实现手写字体的识别

①数据处理（图片处理为数字文本）

②待测图片与训练集每一张图片的向量做欧氏距离

③排序，选取最优K，使得结果最好

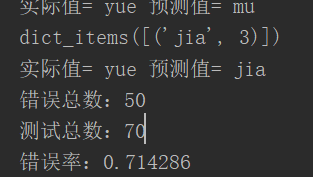
算法：

Load 函数同理读取训练样本dst中28\*28 的文本样本数据，将其转换成二维矩阵train，同时记录它的标签label，记录标签的方法是从文件名读取，碰到下划线为止。

Classifier函数执行KNN 算法。同理读入训练数据和测试数据，将测试数据同样转换为二维矩阵形式test，记录正确的实际标签值ok。 计算样本中

每个测试点矩阵和 所有训练集的距离平方和sum矩阵，这里首先K取得3， 即找出sum 中前三个距离最小的下标(ans 是一个字典，sorted 排序取前三个点即可)。然后可以将实际值和预测值算出来，并打印出测试错误率。

下面是我测试集70条，训练集280条，K 取3 的结果。



3. 讨论以下问题：

3.1 算法的执行时间和那些因素有关

①客观因素：

数据集的数量。 KNN算法中，数据集越多，代表点越多。算的两两之间的点的距离越多，从而时间增加。

机器本身的编译效率，机器指令执行的速度

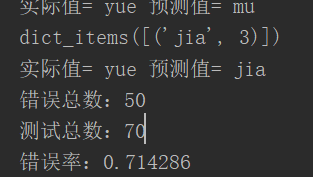
②主观因素：

算法本身的时间复杂度。比如计算两点之间的距离，是计算欧氏距离还是马氏距离。

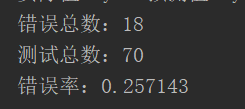
选用策略。比如计算错误率时K选的多少，选择最邻近的3个距离和最临近的十个距离，对于大量数据集来说时间是不一样的。

3.2 算法识别汉字的准确率与训练集质量和数量的关系。（以下前提是K=3）

比如，选取上述图片比较清晰的270个训练集，得出的结论是错误率0.714286



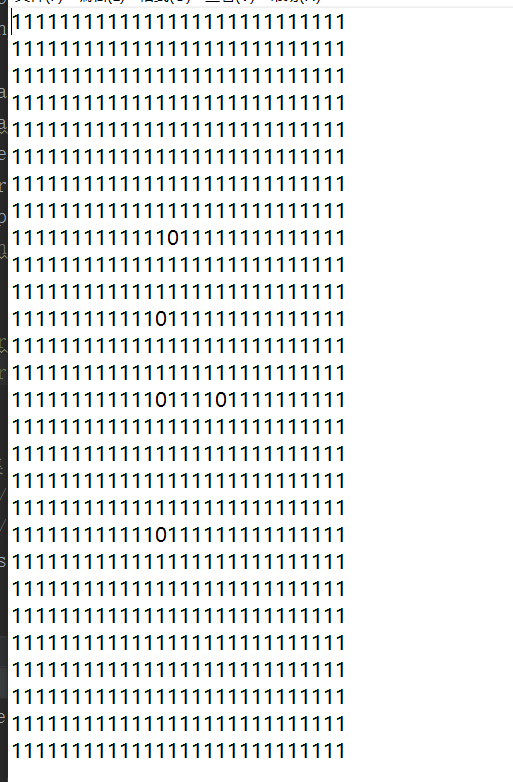
选取70个样本的训练集，错误率为0.257143.



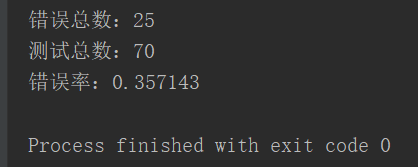
由此可见，选取的样本不是越多越好。

下面选取一些不是很清晰的样本，比如掺杂

10个我做的脏样本，当做训练集

可以看出来数据集比较差，看不出来是“甲”字，笔画太轻，识别不出来。

下面还是70个训练集训练出来的结果



可以看出，错误率有所提升。

得出结论： 训练集越少，准确率越高，但也不要非常少，比如只有几个，十几个。没有普遍性。

同理，训练集质量越高，数据越干净，准确率越高。

3.3 K值选取 （同一组，70个测试集，70个训练集）

|  |  |
| --- | --- |
| K值 | 错误率 |
| 2 | 0.271429 |
| 3 | 0.257143 |
| 4 | 0.228571 |
| 5 | 0.271429 |
| 6 | 0.271429 |
| 7 | 0.314286 |

得出结论，K取4，得到的准确率最高。

参考资料：

# <https://blog.csdn.net/wenqiwenqi123/article/details/79232536> 训练集验证集测试集合理划分比例

# <https://blog.csdn.net/hqh131360239/article/details/80250915>KNN实现手写字体的识别

# <https://www.runoob.com/python3/python3-tutorial.html>Python 3

# <https://stackoverflow.com/questions/1841565/valueerror-invalid-literal-for-int-with-base-10>程序报错[ValueError: invalid literal for int() with base 10: ''](https://stackoverflow.com/questions/1841565/valueerror-invalid-literal-for-int-with-base-10)