**k-NN课程报告**

学号：20307110315

姓名：周训哲

# 1.任务描述

利用k-NN算法实现手写数字的识别与分类。解决了多样化模糊图片的识别分类问题。

# 2.数据集描述

**采用何种数据集开展实验？**

利用MNIST手写数字集作为学习以及测试的数据集。里面包含了图片集作为识别对象，标签集作为学习判断的标准。

**数据集有什么特点？**

MNIST数据集来自美国国家标准与技术研究所, National Institute of Standards and Technology (NIST)。训练集 (training set) 由来自 250 个不同人手写的数字构成, 其中 50% 是高中学生，50% 来自人口普查局 (the Census Bureau) 的工作人员。测试集(test set) 也是同样比例的手写数字数据。具有极大的测试数据量。

数据集分为训练集和测试集两个部分，其中又细分为图片集和标签集。

图片集主要存储的是图像信息，每一张图片包含28\*28个像素，在数据集中展开转换为一个长度为784的向量数据。在Mnist的训练数据集中mnist.train.images是一个形状为[60000, 784]的张量，第一个维度数字用来索引图片，第二个维度数字用来索引每张图片中的像素点，图片里的某个像素的强度值介于0-1之间。

标签集主要存储的是分类的信息，将每张图片所表示的数字进行存储。Mnist数据集中所有的标签mnist.train.labels是一个[60000, 10]的数字矩阵。在训练集中主要用来学习，以及kNN算法中的分类（利用距离）。在测试集中主要用来对比计算算法的识别准确率。

**数据集的各项统计量是多少？**

训练集中的数据量有60000个，测试集中数据量有10000个。

# 3.方法介绍

## 3.1数据预处理（如有）

**从原始数据到输入模型的数据之间，经过了哪些处理？这些处理有何作用？**

首先，原始数据为ubyte二进制文件，需要对二进制数据进行处理。需要利用open（）函数打开，其中参数为 ‘rb’，其含义为将文件以二进制文件形式打开。

对于图像数据（image）是一个3维的数据集（60000\*28\*28）。开头为信息标识，有四个变量，魔数（magic）为2051，数据集大小（size）为60000，图像的行、列都是28，均通过unpack（）函数读取。然后将图像数据从28\*28的二维数据进行转换为长度为748的一维向量images。

对于标签数据（label）是一个一维的数据集（60000），开头也是信息标识，有两个变量，魔数（magic）为2049，数据集大小（size）为60000，也是通过unpack（）函数读取，最后利用append（）函数，将标签存储在向量labels中。

注：以上所述unpack（）函数中引入两个参数，一个为每次读取的数据位数“>II”，另一个为利用read（）函数将文件设置读取格式（16进制/8进制）。

将文件以二进制文件打开的目的是为了将01数据转换为可以进行操作的数据。读取信息标识的作用是为了识别数据的数量等信息，为后续信息处理做铺垫。将二维图像转换为一维向量的目的是为了方便后续的距离计算（可以直接用sum（）函数）。

## 3.2算法描述

**核心思想：**

kNN算法的核心思想是，如果一个样本在特征空间中的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，并具有这个类别上样本的特性。该方法在确定分类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。kNN方法在类别决策时，只与极少量的相邻样本有关。由于kNN方法主要靠周围有限的邻近的样本，而不是靠判别类域的方法来确定所属类别的，因此对于类域的交叉或重叠较多的待分样本集来说，kNN方法较其他方法更为适合。

**整体流程：**

总体来说，kNN分类算法包括以下4个步骤：

①准备数据，对数据进行预处理。

②计算测试样本点（也就是待分类点）到其他每个样本点的距离。

③对每个距离进行排序，然后选择出距离最小的k个点。

④对k个点所属的类别进行比较，根据少数服从多数的原则，将测试样本点归入在k个点中占比最高的那一类。

**缺点：**

该算法在分类时有个主要的不足是，当样本不平衡时，如一个类的样本容量很大，而其他类样本容量很小时，有可能导致当输入一个新样本时，该样本的K个邻居中大容量类的样本占多数。该方法的另一个不足之处是计算量较大，因为对每一个待分类的文本都要计算它到全体已知样本的距离，才能求得它的K个最近邻点。

**改进策略：**

目前对kNN算法改进的方向主要可以分为4类：

①寻求更接近于实际的距离函数以取代标准的欧氏距离，典型的工作包括WAKNN、VDM；

②搜索更加合理的k值以取代指定大小的k值典型的工作包括SNNB、DKNAW；

③运用更加精确的概率估测方法去取代简单的投票机制，典型的工作包括 kNNDW、LWNB、ICLNB；

④建立高效的索引，以提高kNN算法的运行效率，代表性的研究工作包括 KDTree、NBTree。还有部分研究工作综合了以上的多种改进方法。

**训练和预测过程:（具体实现思路）**

首先利用tile（）函数，将测试数据沿x轴复制size倍，与学习数据做差。一种是欧式距离，矩阵本身与自己做数量积，得到每个距离的平方，求和以后取1/2次方得到欧式距离，最后进行排序分类；另一种是曼哈顿距离，做差后取绝对值，然后绝对值求和，对和进行排序分类。取前k个数据，得到一个二维的列表，取k个数据中数量最多的一类作为测试数据的分类。

# 4.实验结果分析

## 4.1评价指标

该算法训练集学习之后，对测试集样例的图像文件进行识别与预测，将预测结果与标签集比对，最后计算得到10000个测试数据的准确率。

## 4.2定量评价结果

**基于曼哈顿距离kNN算法的准确率与错误率**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 |
| 准确率 | 0.964 | 0.9629 | 0.9618 | 0.9605 | 0.9597 | 0.9592 | 0.9587 | 0.9579 | 0.9573 |
| 错误率 | 0.0360 | 0.0371 | 0.0382 | 0.0395 | 0.0403 | 0.0408 | 0.0413 | 0.0421 | 0.0427 |

**基于欧几里得距离kNN算法的准确率与错误率**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| K | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 |
| 准确率 | 0.9717 | 0.9693 | 0.9699 | 0.9669 | 0.9673 | 0.9662 | 0.9643 | 0.9635 | 0.9635 |
| 错误率 | 0.0283 | 0.0307 | 0.0301 | 0.0331 | 0.0327 | 0.0338 | 0.0357 | 0.0365 | 0.0365 |

## 4.3可视化结果

# 总结

由上图可知，不管是使用欧几里得距离还是曼哈顿距离，其总的趋势都是随着k的增大，准确率下降，其中利用欧几里得距离进行分类总体的准确率都要高于曼哈顿距离的测试准确率，且欧式距离判断的准确率下降的速度快于曼哈顿距离。

kNN算法作为最简单的机器学习算法之一，凭借其思路的直观性，对大量数据处理的稳定性，在机器学习算法中取得不可或缺的一席之地。同时，对于少量学习数据而言容易误判，计算量过大等缺点也不可忽视。该次kNN算法实现让我初步了解了机器学习算法及其实现，为之后其他复杂算法的学习打下坚实基础。