高级人工智能大作业

姓 名： 赵路路

学 号： 2023202210120

小组成员： 王亚龙、沈静远、甘泽昊、赵家璇

编写日期： 2023年12月26日

目 录

[1手写数字识别实验 1](#_Toc154523668)

[1.1实验要求 1](#_Toc154523669)

[1.2数据集介绍 1](#_Toc154523670)

[1.3算法一：KNN 2](#_Toc154523671)

[1.3.1算法介绍 2](#_Toc154523672)

[1.3.2代码实现 3](#_Toc154523673)

[1.3.3实验结果 3](#_Toc154523674)

[1.4算法二：全连接神经网络 3](#_Toc154523675)

[1.4.1算法介绍 3](#_Toc154523676)

[1.4.2代码实现 4](#_Toc154523677)

[1.4.3实验结果 4](#_Toc154523678)

[1.5算法三：卷积神经网络 4](#_Toc154523679)

[1.5.1算法介绍 4](#_Toc154523680)

[1.5.2代码实现 6](#_Toc154523681)

[1.5.3实验结果 6](#_Toc154523682)

[1.6性能差异分析 6](#_Toc154523683)

[2小组论文分享报告 6](#_Toc154523684)

[3个人技术报告 6](#_Toc154523685)

# 1手写数字识别实验

## 1.1实验要求

使用以下算法实现手写数字识别任务：

1. KNN算法，可以使用sklearn库，也可以自己实现。
2. 全连接的神经网络模型，可以使用sklearn库。
3. 全连接的神经网络模型，仅可使用numpy库，手动实现。
4. CNN卷积神经网络模型，可以使用sklearn库。

最终使用测试集评估算法的性能，比较各算法的差异，并形成实验报告：

1. 对于KNN算法，需设置不同的K值，比较结果差异；
2. 对于神经网络模型，需设置不同的学习率、隐层节点数，比较结果差异；
3. 对比上述所有模型的性能差异。

## 1.2数据集介绍

DBRHD （Pen-Based Recognition of Handwritten Digits）是UCI机器学习中心提供的手写数字数据集，他们使用一台WACOM PL-100V压感平板，连接到基于Intel 486的PC的串口收集手写样本，最终收集到来源于44位不同书写者的手写数字数据，每个数据条项包括16维特征及标签。

1. 训练集：来源于30位书写者的7494张手写数字特征及其对应标签。
2. 测试集：来源于14位书写者的3498张手写数字特征及其对应标签。

数据分布如图1所示：

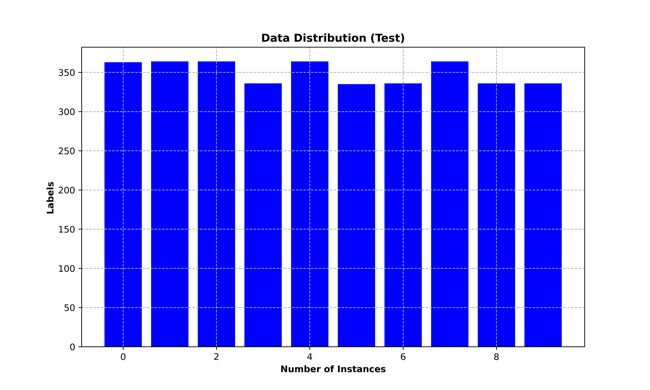
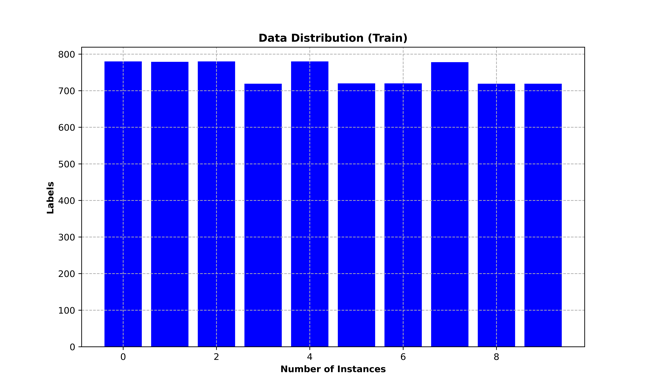


图1 数据集分布柱状图

## 1.3算法一：KNN

1.3.1算法介绍

K最近邻算法（K-Nearest Neighbors，KNN）是一种常用的监督学习算法，用于分类和回归问题。它基于样本之间的距离度量进行预测，即根据最邻近的K个训练样本的标签来确定新样本的标签。如图2所示，KNN算法的基本思想是，如果一个样本在特征空间中的K个最近邻居中的大多数属于某个类别，则该样本也属于这个类别。

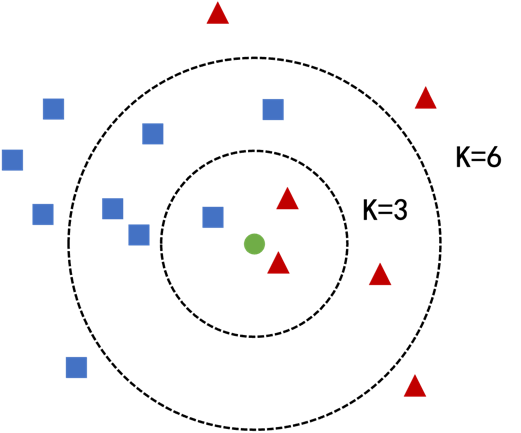


图2 KNN算法示意图

KNN算法的主要步骤如下：

1. 准备数据集：收集训练样本数据集，包括输入特征和对应的标签。

选择K值：确定K的取值，即要考虑的最近邻居的数量。K值的选择会影响算法的性能，通常需要通过交叉验证等方法进行选择。

1. 计算距离：对于一个新的测试样本，计算它与训练样本集中每个样本之间的距离。常用的距离度量方法包括欧氏距离、曼哈顿距离等。
2. 选择最近邻居：根据距离的大小选择与测试样本最近的K个训练样本。
3. 进行预测：对于分类问题，根据K个最近邻居的标签进行投票，将得票最多的类别作为测试样本的预测类别。对于回归问题，可以取K个最近邻居的平均值作为测试样本的预测值。

KNN算法的优点包括简单易用、无需训练过程、对异常值不敏感等。然而，由于需要计算测试样本与所有训练样本之间的距离，KNN算法的计算复杂度较高。此外，特征空间的维度较高时，需要进行降维或特征选择等处理。在应用中，KNN算法常用于分类问题，特别是在样本分布较为均匀、类别间边界不明显的情况下效果较好。此外，KNN算法还可以用于异常检测、推荐系统等领域。

1.3.2代码实现

1.3.3实验结果

## 1.4算法二：全连接神经网络

1.4.1算法介绍

全连接神经网络（Fully Connected Neural Network，FCNN），也被称为多层感知机（Multilayer Perceptron，MLP），是一种经典的人工神经网络模型。它由多个神经元层组成，其中每个神经元与前一层的所有神经元连接，并将其输入加权求和后通过激活函数进行非线性转换。

如图3所示，为全连接神经网络的一般结构示意。

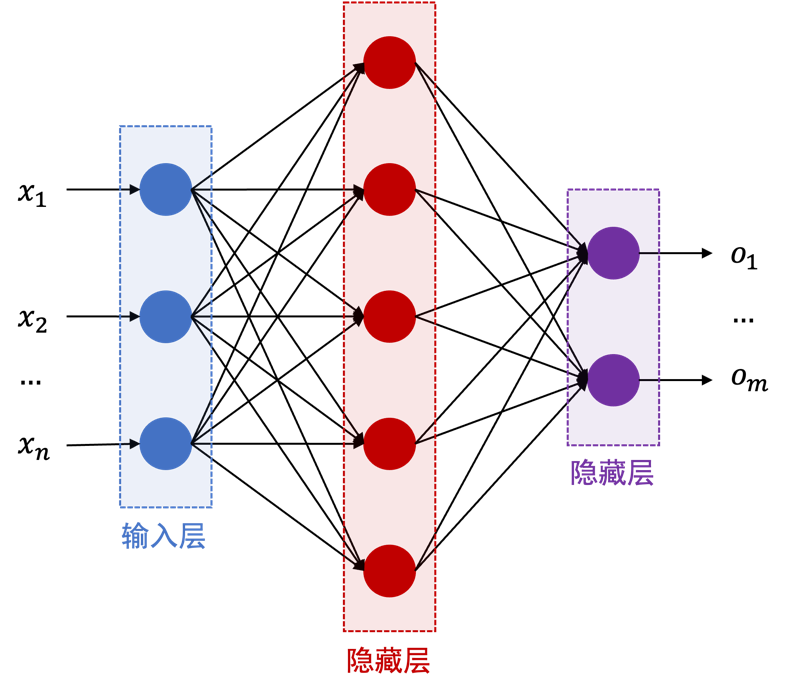


图3 全连接神经网络一般结构示意图

1. 输入层（Input Layer）：接受输入数据的层。每个输入特征都对应一个输入神经元。
2. 隐藏层（Hidden Layers）：中间层，位于输入层和输出层之间。每个隐藏层包含多个神经元，这些神经元与前一层的所有神经元相连接。
3. 输出层（Output Layer）：产生模型的输出结果。输出层的神经元数量取决于任务的要求，例如二分类问题通常使用一个神经元，多分类问题可能使用多个神经元。

全连接神经网络的训练过程主要包括前向传播和反向传播两个阶段：

1. 前向传播（Forward Propagation）：在前向传播中，输入数据通过网络的各个层，每个神经元将其输入加权求和并应用激活函数。这样，网络逐层计算输出，直到达到输出层，产生模型的预测结果。
2. 反向传播（Backpropagation）：在反向传播中，通过比较模型的预测结果和真实标签，计算损失函数的梯度。然后，梯度从输出层向输入层传播，根据链式法则更新每个神经元的权重，以最小化损失函数。

全连接神经网络在许多领域广泛应用，包括图像分类、语音识别、自然语言处理等。然而，全连接神经网络的参数量随着网络层数和神经元数量的增加而增加，容易导致过拟合问题。为了应对这个问题，可以使用正则化技术、Dropout、批归一化等方法来提高模型的泛化能力。

1.4.2代码实现

1.4.3实验结果

## 1.5算法三：卷积神经网络

1.5.1算法介绍

卷积神经网络（Convolutional Neural Networks，CNN）是一类主要用于处理具有网格结构数据，如图像和视频的深度学习模型。CNN在计算机视觉领域取得了很大成功，其设计灵感来自生物学中对视觉系统的理解，尤其是视觉皮层中的神经元的工作方式。

如图4所示，为卷积神经网络的一般结构示意。

1. 输入层（Input Layer）：用于接收输入数据，通常表示为图像。每个输入节点对应图像中的一个像素值或通道。
2. 卷积层（Convolutional Layer）：卷积层是CNN的核心组件。它通过在输入图像上滑动卷积核（filter），学习图像中的特征。卷积操作可以有效地捕捉图像的局部模式，如边缘、纹理等。
3. 池化层（Pooling Layer）：池化层用于降低特征图的空间维度，减少计算复杂度，并提取图像的重要特征。常见的池化操作包括最大池化和平均池化。
4. 全连接层（Fully Connected Layer）：全连接层通常位于网络的顶部，将前面层的输出转换为最终的输出。全连接层的每个节点都与前一层的所有节点相连，通过权重连接。在图像分类任务中，全连接层通常用于输出类别概率。
5. 输出层（Output Layer）：输出层产生最终的预测结果。

如表1所示，是一些常见的卷积神经网络及其特点。

表1 常见卷积神经网络表

|  |  |
| --- | --- |
| 网络名 | 网络特点 |
| LeNet-5 | LeNet-5是早期的卷积神经网络，主要用于手写数字识别。包含卷积层、池化层和全连接层。 |
| AlexNet | AlexNet是在2012年ImageNet图像分类比赛中获胜的模型，推动了深度学习在计算机视觉中的应用。具有8个卷积层和3个全连接层。 |
| VGGNet | VGGNet的核心思想是使用多个3x3的小卷积核来代替一个大的卷积核，以增加网络的深度。VGGNet有16层或者19层的深度。 |
| GoogLeNet | GoogLeNet使用了Inception模块，该模块同时使用多个不同大小的卷积核，提供了不同尺度的信息。具有22层的深度。 |
| ResNet | ResNet是使用残差块（Residual Block）的模型，通过引入残差连接，解决了深度网络中梯度消失问题。ResNet赢得了2015年ImageNet图像分类比赛。 |
| MobileNet | MobileNet是一种轻量级的卷积神经网络，专注于在资源受限的环境中运行，如移动设备。使用深度可分离卷积来减少参数数量。 |
| EfficientNet | EfficientNet基于网络缩放的思想，通过在深度、宽度和分辨率上进行均衡调整，以在相对较少的参数下提高性能。 |

1.5.2代码实现

1.5.3实验结果

## 1.6性能差异分析

# 2小组论文分享报告

# 3个人技术报告