

**《数据挖掘》实验报告**

|  |  |
| --- | --- |
| **专 业：** | **人工智能** |
| **学 号：** |  |
| **姓 名：** |  |

**2024-2025-1学期**

**南京邮电大学**

**实验1 基于深度卷积神经网络的手写体数字图像识别**

一、实验目的：

1. 熟悉MATLAB编程；

2. 使用卷积神经网络进行手写数字识别，进一步理解分类过程。

二、实验设备

1. 硬件：微型计算机。

2. 软件：操作系统Windows XP以上版本、Matlab2012及以上版本。

三、实验内容与步骤

1. 下载安装Matlab软件，配置深度学习开发环境。

2. 准备Mnist数据集，其中每个数字提供50张图片供训练与测试，共500张图片，每张手写数字图片28\*28个像素大小。

3. 熟悉Matlab图像处理工具箱，学习搭建深度神经网络步骤。

四、实验要求

1. 利用matlab读取显示手写数字图像，将数据集划分为训练集与测试集，其中训练集300张，测试集200张；

2. 利用matlab编写深度神经网络，在Mnist训练集上进行训练，在测试集取得良好的预测结果，随后输出真实数字、网络判定数字、判定概率值和分类准确率。

五、实验步骤、原理以及实验结果分析**（共70’；其中实验步骤30’，需给出阶段性运行结果或者系统部署界面截图；原理介绍20’；实验结果截图与分析20’）**

**实验步骤：**

1. **数据准备与加载**：设置数据集路径并加载本地的手写数字图片（.bmp格式）。对图像进行预处理，调整大小为28x28像素，转换为灰度图并归一化。



1. **数据划分**：将数据集随机分为训练集和测试集，训练集包含300张图片，测试集包含200张图片。
2. **构建卷积神经网络（CNN）**：定义CNN模型，包括输入层、卷积层、激活函数、池化层、全连接层和输出层。
3. **定义训练选项**：设置训练选项（如最大训练轮数、批次大小、学习率等）。
4. **训练模型**：使用训练集数据训练CNN模型。

图片包含 日历

描述已自动生成

图表

中度可信度描述已自动生成

1. **测试与评估模型**：使用测试集对训练后的模型进行评估。



**原理介绍：**

1. **卷积层（Convolutional Layer）：**

* **作用**：卷积层的主要功能是通过卷积操作从输入数据中提取局部特征。卷积操作使用多个卷积核（也称为滤波器）扫描输入图像，通过加权求和产生特征图（feature map）。
* **局部感知**：卷积操作通常只关注图像的一小部分区域（局部感知），并通过滤波器滑动整个图像，从而提取不同的局部特征。
* **共享权重**：所有的卷积操作共享同一组权重（卷积核），这使得CNN在学习过程中能有效减少参数数量，提升计算效率。

1. **激活函数（Activation Function）：**

* **作用**：激活函数引入非线性，使得神经网络能够学习和拟合更复杂的模式。常用的激活函数有ReLU（Rectified Linear Unit），它可以避免梯度消失问题，并加速训练过程。

1. **池化层（Pooling Layer）：**

* **作用**：池化层用于降低特征图的空间维度（即缩小图像尺寸），从而减少计算量并防止过拟合。池化通常有最大池化和平均池化两种方式。
* **最大池化**：选择池化区域中的最大值作为输出，保留图像的主要特征。
* **平均池化**：取池化区域中的平均值。

1. **全连接层（Fully Connected Layer）：**

* **作用**：全连接层位于网络的最后，用于将之前提取的特征映射到最终的输出标签。每个神经元都与前一层的所有神经元相连接。
* **输出**：在分类任务中，全连接层的输出通常经过Softmax层处理，以输出每个类别的概率。

1. **Softmax层：**

* **作用**：Softmax层用于将全连接层的输出转换为概率分布，确保输出值在0到1之间，并且所有输出值的和为1。

1. **反向传播与优化（Backpropagation and Optimization）：**

* 在训练过程中，CNN通过反向传播算法调整网络中的权重。通过计算预测误差（如交叉熵损失函数），并利用梯度下降法（如SGD、Adam等）优化网络参数，逐步提高模型的性能。

**实验结果截图与分析:**

**表格

描述已自动生成**

图片包含 QR 代码

描述已自动生成

经过训练后的CNN在测试集上的准确率达到了 83%，这表示网络能够准确地识别绝大多数手写数字。想要进一步提高准确率，可以尝试增加卷积层和全连接层的数量，或使用更先进的网络结构（如ResNet、VGG等）。

六、思考与总结**（20’，每题10’）**

1. 图像分类的一般流程是什么？

图像分类的一般流程可以分为以下几个主要步骤：

1. **数据收集与预处理**

收集大量的图像数据，确保数据多样性和代表性。

数据预处理通常包括图像的大小调整、标准化、增强（如旋转、缩放、翻转等）以及去噪等操作，以便于提高模型的泛化能力。

1. **数据标注**

对收集到的图像进行标注，即为每张图像指定一个类别标签。这一步对于监督学习至关重要。

1. **特征提取**

传统方法中，特征提取是关键步骤，包括颜色直方图、纹理特征、边缘检测等。

在深度学习方法中，神经网络（如卷积神经网络，CNN）自动从图像中提取有用的特征。

1. **构建和训练模型**

构建一个合适的分类模型。在深度学习中，通常使用卷积神经网络（CNN）来处理图像数据。

使用已标注的训练数据集来训练模型，通过反向传播算法调整网络中的参数，使模型能更好地预测图像类别。

1. **模型验证与调整**

使用验证集评估模型性能，查看模型在未见数据上的表现。

根据评估结果调整模型的结构和超参数（如学习率、层数、滤波器数量等）。

1. **测试与评估**

使用测试集对最终模型进行评估，计算准确率、精度、召回率等指标，确保模型的泛化能力。

可以使用混淆矩阵来分析模型的分类效果，查看哪些类别的识别效果较差。

2. 基于MATLAB的深度神经网络搭建有什么特点，与其他语言搭建过程相比有何差异？

基于MATLAB的深度神经网络搭建具有以下特点：

1. **集成化环境**：MATLAB提供完备的开发环境，便于数据处理、算法设计和可视化。
2. **强大工具箱**：MATLAB的深度学习工具箱简化了网络搭建和训练过程。
3. **易用性**：MATLAB语法简洁，适合初学者，尤其在数学计算和矩阵操作上表现突出。
4. **调试与可视化**：内置的调试和可视化工具有助于模型的分析与优化。

与其他语言（比如Python）相比，MATLAB的差异主要在于：

1. **灵活性**：MATLAB的自定义功能较少，而Python更灵活，支持更多自定义操作。
2. **社区支持**：Python有庞大的开源社区和更多框架支持，更新更快。
3. **性能**：两者在性能上差别不大，但Python框架（如TensorFlow、PyTorch）在大规模计算上更具优势。
4. **部署**：Python的部署选项更广泛，适用于更多平台。

七、实验过程遇到的问题以及心得体会**（10’）**

问题：在训练过程中，遇到了训练准确率波动较大的情况，导致模型收敛困难。

解决方法：这种情况通常是由于学习率设置过大或网络参数初始化不当导致的。可以尝试调整学习率（使用较小的学习率），或者使用更好的优化算法（如Adam优化器）。

在本次实验中，CNN通过自动从数据中提取特征，成功地完成了手写数字的分类任务。相比传统的手工特征提取方法，CNN能够通过多层卷积层自动学习图像的局部特征，并且逐渐构建出更高层次的抽象特征。这使得CNN在图像识别任务中表现出色。本次实验加深了对卷积神经网络原理和实践的理解，并通过解决问题和调优模型，获得了较好的实验结果。通过不断优化网络结构和调整超参数，我对深度学习的训练过程有了更深入的认识，为后续的项目和研究奠定了基础。

代码：

clc; clear; close all;

%% 1. 数据准备与加载

% 设置数据集路径

dataFolder = 'Mnist';

digitFolders = dir(fullfile(dataFolder, '\*.bmp'));

% 初始化变量

numImages = length(digitFolders);

imageSize = [28, 28];

images = zeros(imageSize(1), imageSize(2), 1, numImages);

labels = zeros(numImages, 1);

% 读取所有图片

fprintf('正在加载MNIST数据集...\n');

for i = 1:numImages

fileName = digitFolders(i).name;

filePath = fullfile(dataFolder, fileName);

% 读取图像

img = imread(filePath);

img = imresize(img, imageSize);

if size(img, 3) == 3

img = rgb2gray(img); % 转为灰度图

end

images(:, :, 1, i) = double(img) / 255; % 归一化

labelStr = split(fileName, '\_');

labels(i) = str2double(labelStr{1});

end

fprintf('MNIST数据集加载完成，共加载 %d 张图片。\n', numImages);

%% 2. 数据划分为训练集与测试集

% 随机划分数据集：训练集300张，测试集200张

rng(0); % 保证随机划分结果一致

indices = randperm(numImages);

trainIdx = indices(1:300);

testIdx = indices(301:500);

trainImages = images(:, :, :, trainIdx);

trainLabels = categorical(labels(trainIdx));

testImages = images(:, :, :, testIdx);

testLabels = categorical(labels(testIdx));

%% 3. 搭建卷积神经网络（CNN）

layers = [

imageInputLayer([28 28 1], 'Name', 'input')

convolution2dLayer(5, 20, 'Padding', 'same', 'Name', 'conv1') % 卷积层

tanhLayer('Name', 'tanh1')

averagePooling2dLayer(2, 'Stride', 2, 'Name', 'pool1')

fullyConnectedLayer(100, 'Name', 'fc1')

fullyConnectedLayer(10, 'Name', 'fc2')

softmaxLayer('Name', 'softmax')

classificationLayer('Name', 'output')

];

%% 4. 定义训练选项

options = trainingOptions('adam', ...

'MaxEpochs', 10, ...

'MiniBatchSize', 32, ...

'InitialLearnRate', 0.01, ...

'ValidationData', {testImages, testLabels}, ...

'ValidationFrequency', 30, ...

'Verbose', true, ...

'Plots', 'training-progress');

%% 5. 训练网络

fprintf('开始训练卷积神经网络...\n');

net = trainNetwork(trainImages, trainLabels, layers, options);

%% 6. 测试网络性能

fprintf('开始测试网络性能...\n');

[predictedLabels, scores] = classify(net, testImages);

accuracy = sum(predictedLabels == testLabels) / numel(testLabels);

fprintf('测试集准确率: %.2f%%\n', accuracy \* 100);

% 打印部分预测结果

fprintf('\n部分预测结果：\n');

for i = 1:100 % 这里只打印前10个预测结果

% 获取当前图像的真实标签和网络的预测标签

trueLabel = char(testLabels(i));

predictedLabel = char(predictedLabels(i));

% 获取当前图像的预测概率

predictedScore = scores(i, :);

predictedProbability = predictedScore(str2double(predictedLabel) + 1);

% 打印真实标签、预测标签以及预测的概率

fprintf('样本 %d: 真实标签: %s, 预测标签: %s, 预测概率: %.2f\n', ...

i, trueLabel, predictedLabel, predictedProbability);

end

% 显示部分预测结果

figure;

for i = 1:16

subplot(4,4,i);

imshow(testImages(:, :, 1, i));

title(['预测: ', char(predictedLabels(i)), ...

' | 实际: ', char(testLabels(i))]);

end

%% 7. 总结结果

fprintf('实验完成：手写数字识别任务已完成！\n');