

东北大学

人工智能基础作业

基于 BP 神经网络的手写数字识别



专业： 测绘工程

姓名： 栗浩宇

学号： 20222490

班级： 测绘 2202

指导教师： 张恩德

1 项目背景的介绍

1.1 项目意义

手写字体识别是图像识别领域的一个重要分支，具有广泛的应用前景。解决手写字体识别问题对提高自动文档处理、智能识别系统的准确性和效率具有重要意义，能够帮助人们更快速、准确地处理文本信息。过去几十年间，研究者们在手写字体识别领域做出了大量工作，提出了各种各样的方法和技术。其中，基于神经网络的方法因其能够模拟人脑的信息处理方式，特别是 BP 神经网络作为一种广泛应用的模型，被广泛用于手写字体识别任务中。BP 神经网络通过学习大量的手写字符样本，能够自动学习并提取特征，具有很好的识别性能。本项目通过对手写 0-9 数字的识别，旨在提供一种有效的自动识别方法。

1.2 数据来源

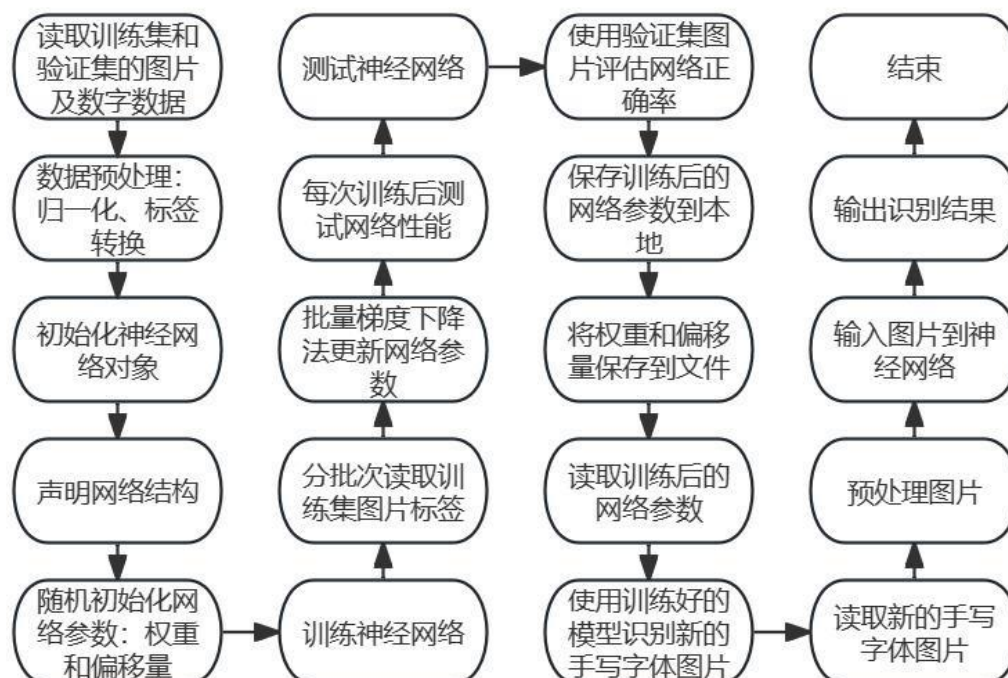
本项目选用 MNIST 数据集作为实验数据，MNIST 数据集来自美国国家标准与技术研究所，由来自 250 个不同人手写的数字构成，其中 50% 是高中学生，50% 来自人口普查局的工作人员。测试集也是同样比例的手写数字数据，但保证了测试集和训练集的作者集不相交。MNIST 数据集一共有 7 万张图片，其中 6 万张是训练集，1 万张是测试集。每张图片是 28×28 的 0-9 手写数字图片。每个图片是黑底白字的形式，黑底用 0 表示，白字用 0-1 之间的浮点数表示，越接近 1，颜色越白。

1.3 编程环境

本实验选用 python 作为编程语言，主要库有：NumPy, PIL, stuct 等。

2 项目技术路线及算法原理

2.1 技术路线



2.2 算法原理

2.2.1 BP 神经网络的概述

在人工神经网络发展历史中，很长一段时间里没有找到隐层的连接权值调整问题的有效算法。直到误差反向传播算法（BP 算法）的提出，成功地解决了求解非线性连续函数的多层前馈神经网络权重调整问题。

BP (Back Propagation)神经网络，即误差反传误差反向传播算法的学习过程，由信息的正向传播和误差的反向传播两个过程组成。输入层各神经元负责接收来自外界的输入信息，并传递给中间层各神经元；中间层是内部信息处理层，负责信息变换，根据信息变化能力的需求，中间层可以设计为单隐层或者多隐层结构；最后一个隐层传递到输出层各神经元的的信息，经进一步处理后，完成一次学习的正向传播处理过程，由输出层向外界输出信息处理结果。当实际输出与期望输出不符时，进入误差的反向传播阶段。误差通过输出层，按误差梯度下降的方式修正各层权值，向隐层、输入层逐层反传。周而复始的信息

正向传播和误差反向传播过程，是各层权值不断调整的过程，也是神经网络学习训练的过程，此过程一直进行到网络输出的误差减少到可以接受的程度，或者预先设定的学习次数为止。

2.2.2 神经网络训练过程

1) 前向传播 (Forward Propagation): 输入样本数据经过输入层，并通过网络中的隐藏层逐层传播，最终到达输出层。在每一层中，通过将输入与相应的权重相乘并加上偏置 (bias)，然后应用激活函数，计算每个节点的输出。对于隐藏层和输出层的每个节点，输出值都是通过激活函数 (如 Sigmoid、ReLU 等) 将加权输入进行非线性转换得到的。

2) 反向传播 (Backward Propagation): 通过与标签数据进行比较，计算输出层节点的误差。常见的误差函数包括均方误差 (Mean Squared Error) 等。误差然后沿着网络反向传播，通过链式法则计算每一层的梯度，以便更新权重和偏置。根据梯度下降法则，更新网络参数以最小化误差函数。这里使用了梯度下降的变体，如小批量梯度下降 (Mini-batch Gradient Descent) 来提高训练效率。

3) 权重和偏置的更新: 使用梯度下降算法，更新每一层的权重和偏置。这一步通过计算误差对参数的偏导数来实现。学习率 (Learning Rate) 是一个重要的超参数，它决定了每次参数更新的步长。合适的学习率能够加快收敛速度，但过大的学习率可能导致算法无法收敛，而过小的学习率则会降低收敛速度。

4) 迭代训练: 重复执行前向传播和反向传播过程，直到达到预设的迭代次数或者达到了收敛条件。在每次迭代中，随机从训练集中选择一批样本进行参数更新，以减少计算开销和提高训练效率。神经网络训练过程的目标是通过调整网络参数，使得网络的预测结果与真实标签尽可能接近，从而达到准确识别手写字符的目的。

2.2.3 小批量梯度下降法

小批量梯度下降法 (Mini-batch Gradient Descent) 是一种常用的优化算法，用于训练神经网络和其他机器学习模型。与批量梯度下降和随机梯度[8]下降相比，小批量梯度下降法结合两者的优点，在训练过程中每次更新参数时使用一个小批量样本。

- 1) 初始化参数：初始化模型的权重和偏置，设置学习率、迭代次数等超参数。
- 2) 分批次处理数据：将训练集分成多个小批量（mini-batches），每个小批量包含一定数量的样本，通常是 2 的幂次方，例如 64、128 等。
- 3) 迭代更新参数：对于每个小批量数据，执行前向传播和反向传播，计算梯度，使用计算得到的梯度来更新模型的参数（权重和偏置）。
- 4) 重复迭代：重复执行 2 和 3 步骤，直到达到指定迭代次数或者满足收敛条件。

2.2.4 归一化处理和标签转换

2.2.4.1 归一化处理

在机器学习和深度学习中，归一化处理是一种常用的数据预处理技术，用于将不同特征的值范围缩放到相同的尺度。归一化能够消除特征之间的量纲差异，有助于提高模型训练的稳定性和收敛速度。

- 1) 确定归一化范围：确定归一化的范围，通常将特征的值缩放到 $[0, 1]$ 或者 $[-1, 1]$ 范围内。
- 2) 计算归一化因子：对于每个特征，计算归一化所需的因子（例如最小值和最大值，或者均值和标准差）。
- 3) 应用归一化变换

消除量纲差异：归一化处理消除了不同特征的量纲差异，使得特征之间的比较更具有可比性。

提高训练效果：归一化有助于加快模型的收敛速度，提高训练的稳定性和效果。

2.2.4.2 标签转换

在机器学习和深度学习中，对标签进行适当的转换是数据预处理的重要环节之一。特别是在分类任务中，将类别标签转换为模型可接受的格式是必要的。

- 1) 确定标签编码方式：确定将类别标签编码为模型可以接受的格式，通常使用独热编码（One-Hot Encoding）或者整数编码（Integer Encoding）。
- 2) 应用标签转换：对每个类别标签进行转换，使用以下两种方式之一：

独热编码：将每个类别标签编码为一个向量，其中只有一个元素为 1，其

余元素为 0。例如，[0, 1, 0] 表示类别 2。

整数编码：将每个类别标签映射为一个整数值。例如，类别 1 映射为整数 0，类别 2 映射为整数 1，依此类推。

模型可接受性：将类别标签转换为模型可接受的格式，有助于提高模型的训练效果和泛化能力。

3 运行结果

3.1 BP 神经网络的运行结果

3.1.1 准确率

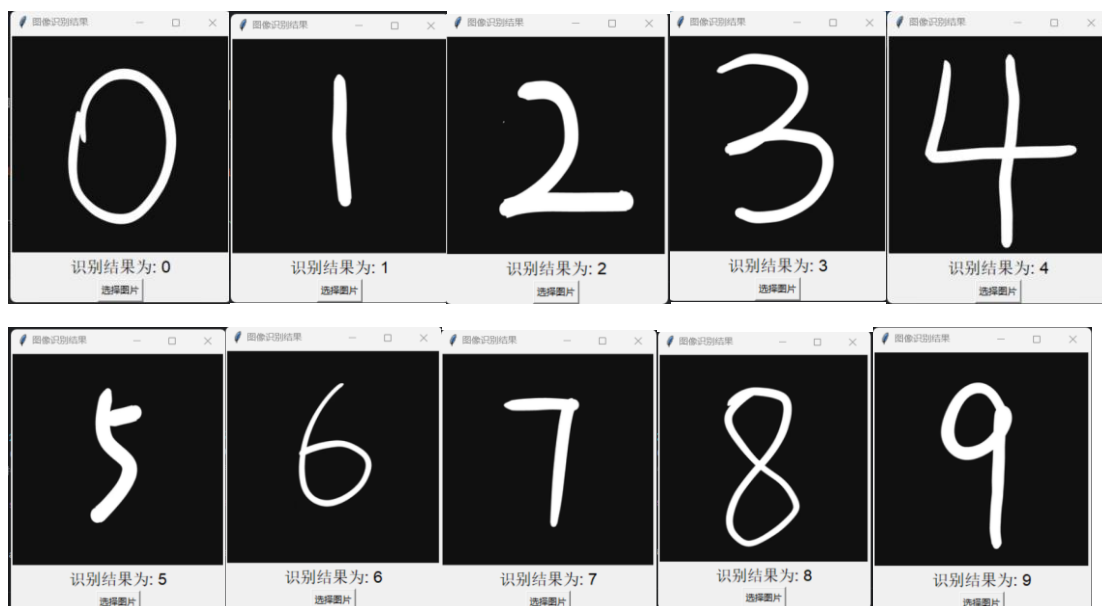
经过 30 次迭代训练后，测试集上的识别正确率达到了 95%左右

```
C:\Users\栗浩宇\AppData\Local\Microsoft\WindowsApps\python3.10.exe D:\字符识别\main.py
魔数:2051, 图片数量: 60000张, 图片大小: 28*28
已解析 10000张
已解析 20000张
已解析 30000张
已解析 40000张
已解析 50000张
已解析 60000张
魔数:2049, 图片数量: 60000张
已解析 10000张
已解析 20000张
已解析 30000张
已解析 40000张
已解析 50000张
已解析 60000张
魔数:2051, 图片数量: 10000张, 图片大小: 28*28
已解析 10000张
魔数:2049, 图片数量: 10000张

第25次学习结束
正确率: 9486/10000
第26次学习结束
正确率: 9490/10000
第27次学习结束
正确率: 9492/10000
第28次学习结束
正确率: 9473/10000
第29次学习结束
正确率: 9486/10000
第30次学习结束
正确率: 9524/10000
```

3.1.2 泛化能力

泛化能力是指模型在未见过的数据上表现良好的能力。为了测试 BP 神经网络的泛化能力，除测试集的 10000 张手写字体图片之外，我又自行准备了 0-9 的手写字体图片各一张供验证模型的泛化能力，并将识别结果做了可视化处理，以下是测试结果展示：



3.2 与其他算法的比较

为了进一步评价 BP 神经网络的结果，我选择与用相同的 MNIST 数据集在随机森林算法和 KNN 算法得到的结果进行比较。

3.2.1 与随机森林算法比较

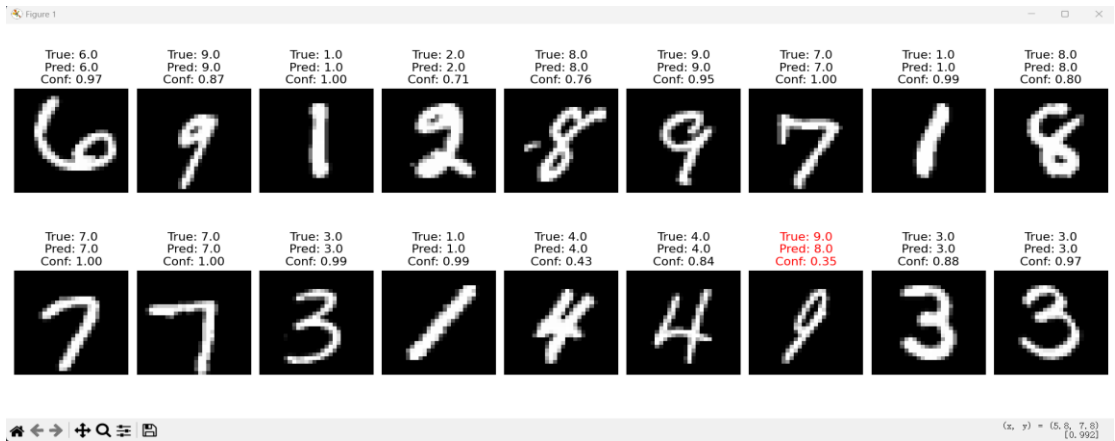
3.2.1.1 精度评价

```
Overall Accuracy: 96.90%
正确率: 9690/10000
Classification Report:
```

	precision	recall	f1-score	support
0.0	0.97	0.99	0.98	980
1.0	0.99	0.99	0.99	1135
2.0	0.96	0.97	0.96	1032
3.0	0.96	0.96	0.96	1010
4.0	0.97	0.97	0.97	982
5.0	0.97	0.96	0.97	892
6.0	0.97	0.98	0.97	958
7.0	0.97	0.96	0.97	1028
8.0	0.96	0.96	0.96	974
9.0	0.96	0.95	0.95	1009
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000

可以看到在该实验模型中我们得到了测试集 10000 张图片的总体识别准确率为 96.9%，以及每个数字的准确率。

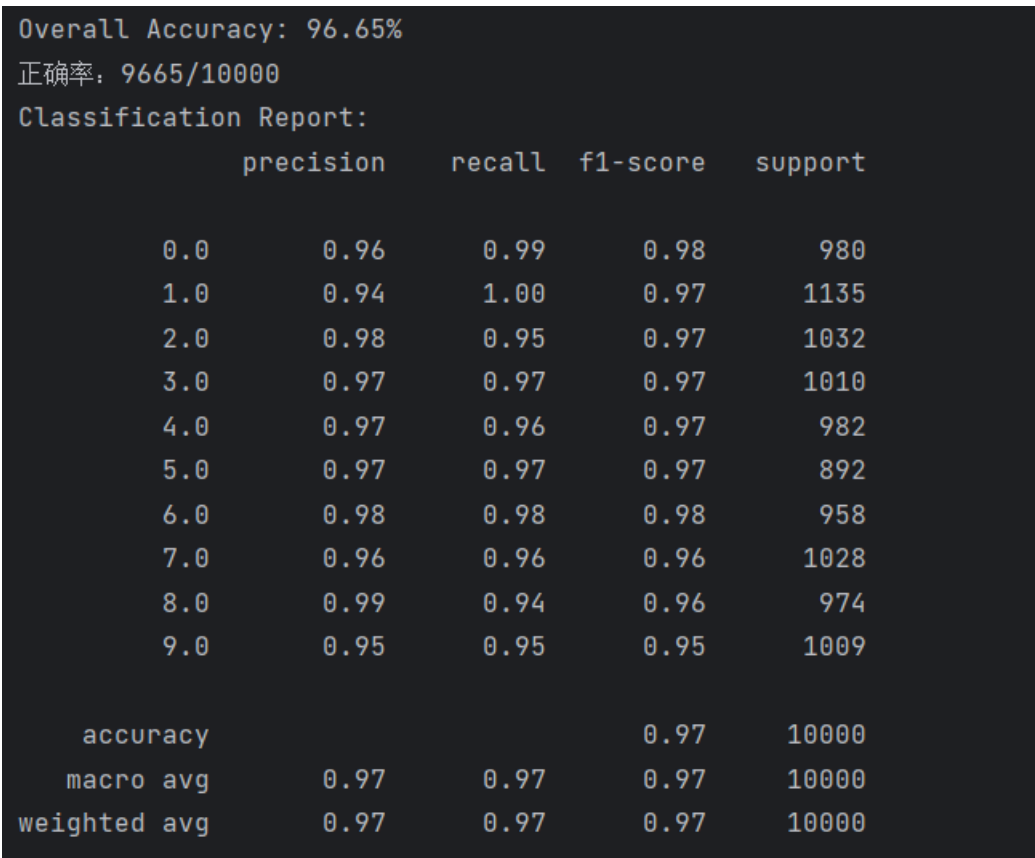
3.2.1.2 可视化展示



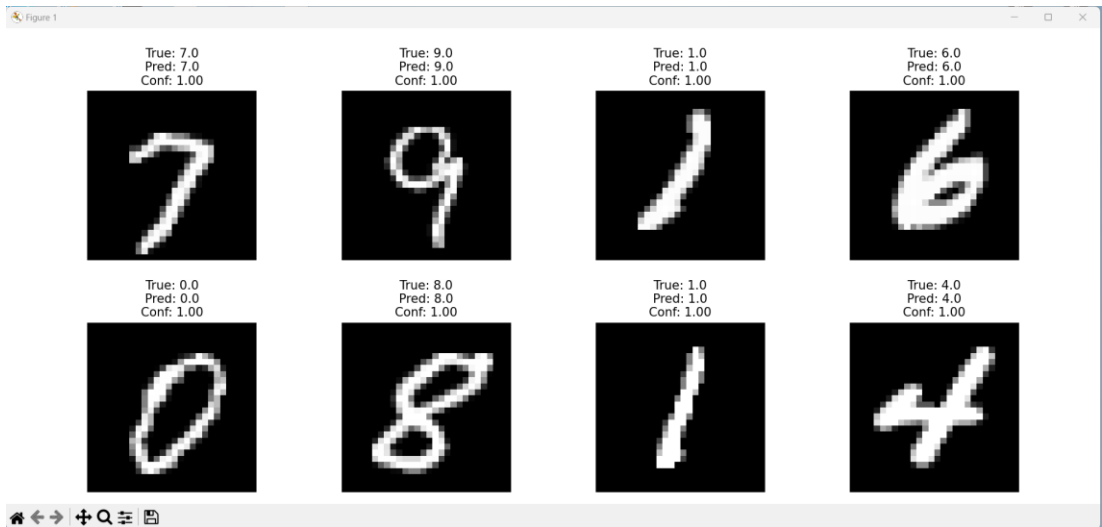
随机将测试集中的图片展示出来，并进行预测，如果预测错误则显示红色。可视化展示的数量可以进行调整(为了得到错误的结果改成了 18 个)。

3.2.2 与 KNN 算法比较

3.2.2.1 精度评价



3.2.2.2 可视化展示



3.3 结论

本项目通过基于 BP 神经网络的手写数字识别任务，深入探讨了神经网络模型在数字识别领域的应用和性能。BP 神经网络在数字识别任务中展现出了较好的泛化能力和适应性，能够有效处理不同风格和书写习惯的手写数字。与传统的机器学习算法相比，BP 神经网络能够自动学习和提取特征，无需手动设计特征提取器，从而简化了任务流程并提高了识别速度。

在 MNIST 数据集上，随机森林和 KNN 由于其算法特性和数据集的简单结构，能够取得较高的准确率。MNIST 数据集的预处理和归一化使得这些基于简单特征和距离度量的算法能够高效地进行分类。具体来说，随机森林通过集成多棵决策树，有效降低了过拟合风险并提高了模型的稳定性；而 KNN 通过简单的距离度量方法，能够在标准化的数据上快速、准确地进行分类。

然而，这并不意味着 BP 神经网络在其他场景中表现不好。BP 神经网络在处理复杂、非线性和高维度数据方面具有独特优势。在更复杂的数据集和任务中，BP 神经网络及其变种（如卷积神经网络）通常能够展示出更强的特征提取和泛化能力。

因此，根据具体任务和数据集的特点选择合适的算法是非常重要的。对于简单、结构化的数据集，如 MNIST，随机森林和 KNN 可能表现更好。而在处理复杂任务和非结构化数据时，BP 神经网络及其变种通常具有更强的优势和潜力。

4 项目心得

在完成基于 BP 神经网络的手写数字识别项目过程中，我遇到了许多挑战，同时也收获了丰富的经验。

起初，由于我的编程水平有限，基于 BP 神经网络的初步实现中存在许多错误。通过逐个在网上查询和改正这些错误，我最终训练好了模型。然而，训练完成后，我只获得了在测试集上的正确率。为了更好地验证模型，我决定使用自己手写的数字图片进行识别。因此，我添加了对手写图片的预处理过程以及相应的调用函数，成功实现了对自己手写数字的识别。在最开始，识别结果只能在运行框中显示，这显得不够直观。于是，我通过修改调用函数，将结果以对话框的形式显示，使其更加直观和易于理解。

在完成基于 BP 神经网络的数字识别后，我又想尝试一下别的算法，于是有了结果中的随机森林算法和 KNN 算法，在做这些算法之前，我也上网查询了一些资料，得到了初步的了解：BP 神经网络的准确度应该要大于随机森林算法和 KNN 算法的准确度，所以我做了这两个算法，本意是通过这两个算法来突出 BP 神经网络的优越性，但结果让我很意外，怀疑是不是自己做错了，出于这个原因，我又上网查询了一些资料，发现可能是由于 MNIST 数据集不够复杂，不能完全发挥 BP 神经网络的优势，并且这个数据集很适合随机森林算法和 KNN 算法，BP 神经网络更适合用于更加复杂的场景里，这个过程也让我学到了很多。

通过本项目，我对 BP 神经网络的工作原理有了深入的理解，包括前向传播和反向传播的具体过程。我学会了如何通过调整网络参数来优化模型性能，并掌握了数据归一化和标签转换的各种方法及其在不同场景下的应用。在面对各种挑战时，我通过查阅文献、询问其他同学，利用网上资源，不断尝试不同的解决方案，解决了一个个问题，这个过程很充实。

此外，我的 Python 编程能力技能也得到了显著提升，说实话，刚开始去做这个项目时，我很怀疑自己能否完成，但一个接着一个算法的实现，也让我意识到我进步了许多。

不仅仅是这个项目，人工智能基础课的学习也让我受益匪浅，张老师上课总会涉及许多课外的知识，让我眼前一亮，吸引我去了解，感谢张老师的悉心指导，让我打下了坚实的人工智能基础，相信我会在以后的学习中更进一步。