

华中科技大学

2023

计算机视觉实验 · 实验报告 ·

专 业： 计算机科学与技术

班 级： CS2106

学 号： U202115514

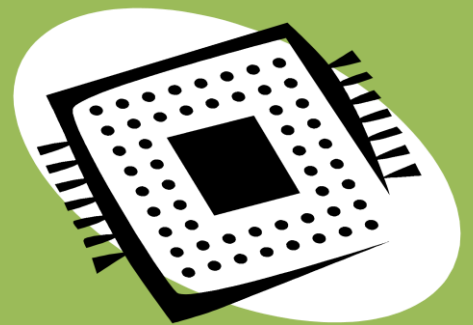
姓 名： 杨明欣

电 话： 13390396012

邮 件： ymx@hust.edu.cn

指导老师： 刘康

完成日期： 2023-12-19



计算机科学与技术学院

目录

1	实验内容	2
1.1	实验要求	2
1.2	实验内容	2
1.2.1	前馈神经网络	2
1.2.2	反向传播算法	2
1.3	实验方法	3
1.3.1	网络架构设计	3
1.3.2	网络性能比较	3
1.4	实验设计	4
1.4.1	数据集	4
1.4.2	实验环境	4
1.5	实验分析与结果	5
1.5.1	分类性能	5
1.5.2	分类性能	6
1.6	实验总结	8
2	实验心得	9

1 实验内容

1.1 实验要求

设计一个前馈神经网络，对一组数据实现分类任务。

具体要求：下载“dataset.csv”数据集，其中包含四类二维高斯数据和它们的标签。设计至少含有一层隐藏层的前馈神经网络来预测二维高斯样本($data_1, data_2$)所属的分类label。这个数据集需要先进行随机排序，然后选取 90%用于训练，剩下的 10%用于测试。

1.2 实验内容

1.2.1 前馈神经网络

前馈神经网络（Feedforward Neural Network, FNN），是一种最简单的神经网络，各神经元分层排列，每个神经元只与前一层的神经元相连。接收前一层的输出，并输出给下一层，是应用最广泛、发展最迅速的人工神经网络之一。

前馈神经网络在实际应用中有多用途，包括模式识别和分类（如图像识别、语音识别）、函数逼近、数据降维、异常检测、预测和控制等。它通过学习复杂的非线性关系，可以处理各种任务，例如在回归分析、金融市场预测、工业控制等领域发挥作用，展现出灵活性和多功能性。

1.2.2 反向传播算法

反向传播（Back propagation, BP）是一种用于训练神经网络的算法，它是神经网络发展历程中的重大突破，也是现代深度学习训练方法的基础之一。该算法通过计算神经网络中损失函数对各参数的梯度，并结合优化方法来更新参数，从而降低损失函数。

1.3 实验方法

为了实现对一组数据的分类任务，需要设计前馈神经网络的模型架构，通过前向和后向的传播，对前馈神经网络内部参数进行训练，得到最终的分类模型。本实验在此分成两个部分，第一部分设计一个整体的网络架构完成分类任务，第二部分则更改第一部分网络中的部分架构对比网络性能。

1.3.1 网络架构设计

本实验中主要通过使用前馈神经网络进行分类任务，最终设计实现了四层的前馈神经网络，包括一层输入层（包含两个神经元，对应输入的两个特征维度），一层输出层（包含四个神经元，对应输出的四种分类），两层隐藏层（每层均为四个神经元）。具体结构示意图如图 1-1 所示。

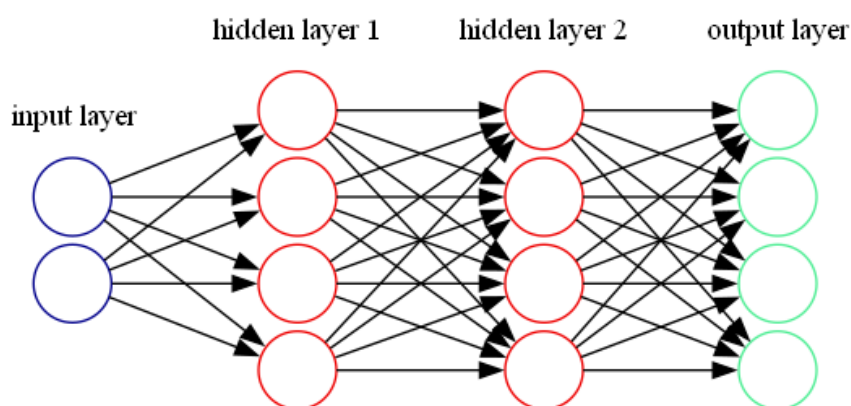


图 1-1 前馈神经网络网络架构示意图

输入层只保存输入数据，不执行任何计算，因此不需要使用激活函数。在神经网络的隐藏层中需要使用非线性激活函数，隐藏层对应的激活函数在这里首先选用 ReLU 激活函数。

1.3.2 网络性能比较

在前一部分的网络架构中，我们设计实现了一个以 ReLU 为激活函数的四层神经网络架构。在此部分，我们通过控制变量法的形式，比较不同神经元层数，每层神经元个数以及激活函数的选择对于网络性能的影响。

华中科技大学课程实验报告

比较不同神经元层数对于网络性能的影响,我们设计每层隐藏层均为4个神经元,使用 ReLU 激活函数,不断改变隐藏层数量比较网络性能。

比较每层神经元个数对于网络性能的影响,我们设计两层隐藏层,使用 ReLU 激活函数,不断改变每层隐藏层的神经元数目比较网络性能。

分析激活函数的选择对于网络性能的影响,我们设计两层隐藏层,每层隐藏层均为4个神经元,分别使用 Sigmoid、Tanh、ReLU、Leaky ReLU、GeLU、ELU 激活函数比较网络性能。

1.4 实验设计

1.4.1 数据集

本实验采用 `dataset.csv` 数据集,其为一个四分类的数据集,数据总量为 4000。首先需要对四类进行规范化,使其变成 0、1、2、3 四类。进一步分析数据集得到四类数据的比例为 1:1:1:1,特征维度为二维,将数据点绘制在二维平面的分布如图 1-2 所示。

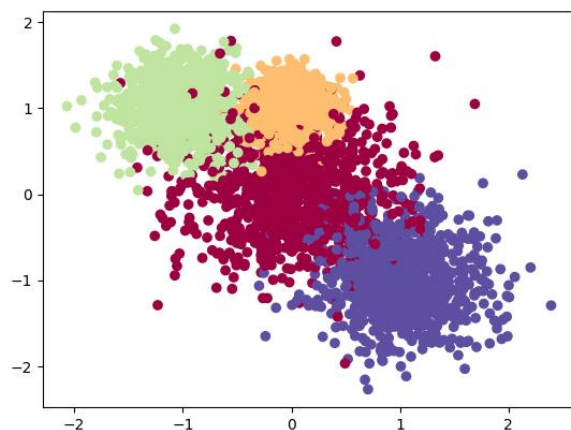


图 1-2 数据集分布

接下来首先将数据集进行打乱 (shuffle),然后将数据集划分为训练集 : 测试集 = 9 : 1, 以备模型训练和测试。

1.4.2 实验环境

华中科技大学课程实验报告

系统版本: Windows 11 专业版

处理器: 13th Gen Intel(R) Core(TM) i9-13900HX 2.20 GHz

机带 RAM: 16.0 GB (15.7 GB 可用)

系统类型: 64 位操作系统, 基于 x64 的处理器

显卡版本: NVIDIA GeForce RTX 4060 Laptop GPU

显存: 8188MiB

CUDA 版本: 12.0

Python 版本: 3.9.18

torch 版本: 2.1.1+cu118

1.5 实验分析与结果

1.5.1 分类性能

对于 1.3.1 中的网络架构进行训练, 使用交叉熵损失函数 (Cross_entropy), 使用 SGD 优化器, epoch 为 100 轮, batchsize 为 128, 学习率设置为 0.1, 训练过程中训练集的损失和准确率如图 1-3 所示。

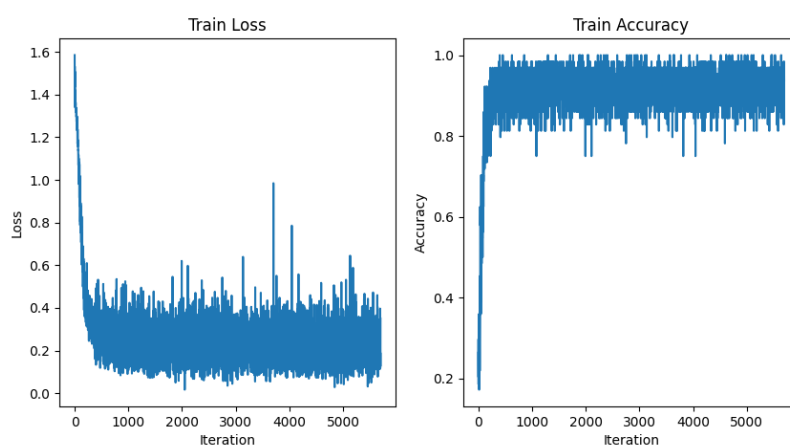


图 1-3 训练过程训练集的损失和准确率

最终在训练集上进行测试可得整体的准确率为 91.94%, 其中第一类的准确率为 85.03%, 第二类的准确率为 96.15%, 第三类的准确率为 94.28%, 第四类的准确率为 92.27%。

训练过程中测试集的损失和准确率如图 1-4 所示。

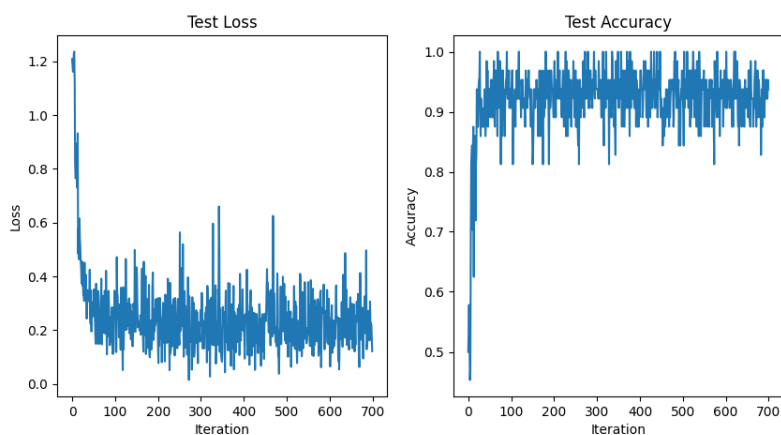


图 1-4 训练过程测试集的损失和准确率

最终在测试集上进行测试可得整体的准确率为 93.75%，其中第一类的准确率为 88.57%，第二类的准确率为 97.83%，第三类的准确率为 96.33%，第四类的准确率为 92.55%。

通过模型训练得到的决策边界如图 1-5 所示。

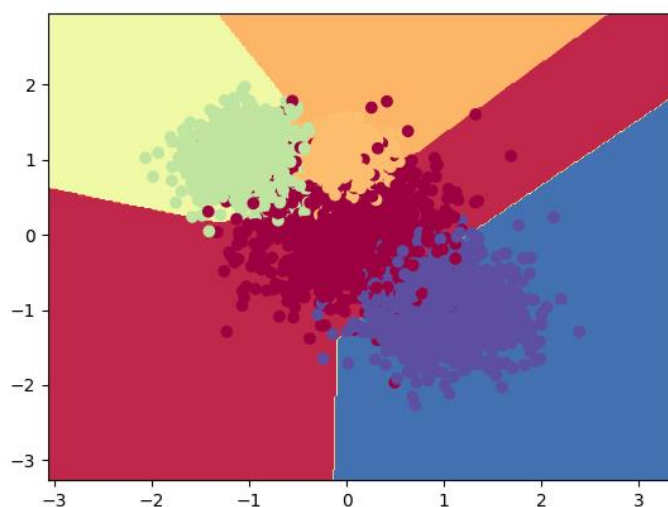


图 1-5 决策边界

从图 1-5 可以看出获得的决策边界基本符合预期结果。最终可以得出结论我们的模型最终收敛到了一个比较好的效果，能够很好地完成四分类任务。

1.5.2 分类性能

首先比较不同神经元层数对于网络性能的影响。具体测试结果如表 1-1 所示。

华中科技大学课程实验报告

表 1-1 不同神经元层数对网络性能的影响

#	隐藏层层数	隐藏层神经元个数	激活函数	时间性能/s	准确度/%
1	1	4	ReLU	8.37	93.00
2	2	4	ReLU	9.16	93.75
3	3	4	ReLU	9.62	92.75
4	4	4	ReLU	9.81	91.75
5	5	4	ReLU	10.06	90.25
6	6	4	ReLU	10.41	-
7	7	4	ReLU	10.77	-
8	8	4	ReLU	10.91	- ¹

由此可以得出如下结论：

1. 隐藏层层数对时间性能有很大的影响，因为隐藏层数越多，模型中需要训练的参数数量越多，所需要的时间就越长。
2. 并不是隐藏层层数越多（参数量越大），模型的性能就越好，有时候过少的参数量会导致模型欠拟合，而过多的模型则会使模型过拟合，当模型参数量过大时，则有可能导致模型崩溃，分类性能极低，因此需要选择适当的隐藏层层数。

进一步比较每层神经元个数对于网络性能的影响。具体测试结果如表 1-2 所示。

表 1-2 不同神经元个数对于网络性能的影响

#	隐藏层层数	隐藏层神经元个数	激活函数	时间性能/s	准确度/%
1	2	2	ReLU	9.07	90.05
2	2	3	ReLU	9.23	91.50
3	2	4	ReLU	9.16	93.75
4	2	5	ReLU	9.08	93.00
5	2	6	ReLU	9.31	90.10
6	2	7	ReLU	9.05	92.00
7	2	8	ReLU	9.35	90.25

¹ 在隐藏层层数过多时，由于数据量较小，训练轮次有效，可能会导致训不动的情况，准确度极低，没有比较的意义，用 - 代替

华中科技大学课程实验报告

由此可以得出结论：单独增加隐藏层神经元个数对于模型时间性能有一定的损耗，但是相较于增加隐藏层层数来说影响有限（局限在我们设计的隐藏层神经元个数很少，隐藏层层数也少）。同时，也并不是每个隐藏层的神经元个数越多，最终模型的性能越好。

最终比较不同激活函数对于模型的性能的影响。具体测试结果如表 1-3 所示。

表 1-3 不同激活函数对于分类性能的影响

#	隐藏层层数	隐藏层神经元个数	激活函数	收敛速度排名	准确度/%
1	2	4	Sigmoid	2	90.25
2	2	4	Tanh	2	92.25
3	2	4	ReLU	1	93.75
4	2	4	Leaky ReLU	1	93.25
5	2	4	GeLU	1	92.50
6	2	4	ELU	1	90.75

由此可以得出结论：不同的激活函数收敛速度不同，ReLU、Leaky ReLU、GeLU 和 ELU 明显比 Sigmoid 和 Tanh 收敛速度快，同时不同的激活函数最终的性能也有所不同，需要进行比较后选择最合适的激活函数。

1.6 实验总结

本次实验主要完成了如下几点工作：

- 1) 设计实现了一个以 ReLU 为激活函数的四层神经网络架构，对一组数据实现分类任务。
- 2) 通过控制变量法的形式，比较不同神经元层数，每层神经元个数以及激活函数的选择对于网络性能的影响。
- 3) 得出结论，不同神经元层数，每层神经元个数以及激活函数的选择对于网络性能的影响不同，需要在设计前馈神经网络的时候多维度进行测评，选择最合适的网络架构。

2 实验心得

在完成前馈神经网络实现分类任务的实验过程中，我获得了许多宝贵的经验。

- 1) 选择合适的网络结构和超参数对于实验成功的重要性。通过不断调整网络的层数、每层的神经元数量、激活函数、学习率等参数，我逐渐找到了适合当前数据集的最佳配置。这个过程需要耐心和细心，因为每个参数的调整都可能对实验结果产生重大影响。
- 2) 学会了如何使用各种工具和库来辅助实验。例如，使用 NumPy 和 Pandas 进行数据处理，使用 Matplotlib 进行数据可视化，构建决策边界，使用 logger 记录日志，以及使用 Pytorch 深度学习框架来构建和训练神经网络。这些工具的使用大大提高了实验效率，使我能够更快地完成实验并分析结果。

总的来说，这次实验让我对前馈神经网络在分类任务中的应用有了更深入的了解。我不仅学会了如何构建和训练神经网络模型，还掌握了如何优化模型性能和处理实验过程中遇到的各种问题，收获很大。

• 指导教师评定意见 •

一、原创性声明

本人郑重声明本报告内容，是由作者本人独立完成的。有关观点、方法、数据和文献等的引用已在文中指出。除文中已注明引用的内容外，本报告不包含任何其他个人或集体已经公开发表的作品成果，不存在剽窃、抄袭行为。

特此声明！

作者签字：杨明欣



二、对课程实验的学术评语（教师填写）

三、对课程实验的评分（教师填写）

评分项目 (分值)	课程目标 1 工具应用 (10 分)	课程目标 2 设计实现 (70 分)	课程目标 3 验收与报告 (20 分)	最终评定 (100 分)
得分				

指导教师签字：_____