

《神经网络技术》课程大作业要求说明

使用 Python NumPy 从零实现一个多层感知机，包括完整的前向传播与反向传播过程，掌握神经网络的基本原理与数值实现方法。

■ 目标

- 1、理解并实现 MLP 的数学原理。
- 2、掌握前向传播和反向传播的实现流程。
- 3、使用 NumPy 完成矩阵运算，禁止调用深度学习框架（如 PyTorch、TensorFlow、Keras 等）中的现成神经网络模块。
- 4、在小规模数据集上完成训练与测试，验证实现的正确性。

■ 要求

- 1、网络结构
 - 1) 至少支持输入层、隐藏层、输出层的基本三层网络（可扩展为多隐藏层）。
 - 2) 每一层需要实现：仿射变换和非线性激活函数（如 Sigmoid、ReLU、Tanh 等，自行选择）。
 - 3) 前向传播：需要支持分类任务（如二分类/多分类 Softmax 输出）。
 - 4) 反向传播：根据损失函数（如 MSE、交叉熵）推导并实现参数梯度计算；对权和偏置的梯度进行更新；更新方式自行选择，如 SGD、Adam。
- 2、训练与测试
 - 1) 数据集：可选用手写数字数据集（MNIST 的子集，或自制小数据集）。
 - 2) 输出训练过程中损失值随迭代的变化曲线。
 - 3) 最终展示在测试集上的性能，如测试样例、ROC、AUC。

■ 代码规范

- 1、必须使用 NumPy 实现所有矩阵运算，不得调用深度学习框架。
- 2、代码需包含注释（公式对应关系、变量含义）。
- 3、提交报告需简要说明：代码设计思路、实验过程、实验结果与分析

■ 评分标准

- 1、基本实现（40%）：正确完成前向传播与反向传播，能在小数据集上训练。
- 2、代码规范（20%）：结构清晰、注释完整、可运行。
- 3、实验结果（20%）：提供训练曲线、测试集准确率。
- 4、扩展（20%）：多隐藏层结构、激活函数对比、优化算法对比（Momentum/Adam）、泛化能力分析等

■ 提交内容

- 1、源代码
- 2、实验报告（PDF/Word），含训练日志、可视化图表、测试结果