

文件描述：

bp_np.py: 反向传播示例代码，仅供学习，不涉及作业。

examples.py: 第一课课上配套代码，仅供学习，不涉及作业。

np_mnist_template.py: 下方**作业 1**的代码，具体操作见作业 1

torch_mnist_template.py: 下方**作业 2**的代码，具体操作见作业 2

mnist: 作业数据集

作业 1: 用 numpy 实现训练 MLP 网络识别手写数字 MNIST 数据集

任务介绍：

本作业目的是通过 numpy 理解梯度回传的计算方式。

任务描述：

1. 运行、阅读并理解反向传播算法示例 bp_np.py，确保理解其工作原理。
2. 复制 np_mnist_template.py 至 np_mnist.py，并根据以下要求对其进行修改：
 - 补充代码内容（例如 relu, relu_prime, 反向传播等）(2.5 分)
 - 更改损失函数（例如，可以尝试使用交叉熵损失函数）。(0.5 分)
 - 调整网络结构，包括但不限于更改隐藏层数量、神经元数量。(0.5 分)
 - 尝试不同的激活函数（如 ReLU、sigmoid、tanh 等）。(0.5 分)
3. 使用修改后的 np_mnist.py 完成训练 MLP 网络以识别手写数字 MNIST 数据集。(1 分)

提交要求：

- 在 10 个 epoch 后，测试集的准确率需达到 94%以上。(1 分)，提交代码
- 提交一份实验报告，内容包括训练过程中打印的准确率截图（确保截图中展示了每个 epoch 结束时的准确率），所做的修改描述。实验报告格式不限，勿长篇大论。

作业 2：使用 Pytorch 训练 MNIST 数据集的 MLP 模型

任务介绍：

本作业目的是观察网络结构、优化器、超参数对网络性能的影响。

任务描述：

1. 运行、阅读并理解 `mnist_mlp_template.py`，了解其网络结构和训练流程。
2. 复制 `mnist_mlp_template.py` 至 `mnist_mlp.py`，并根据以下要求对其进行修改：
 - 修改网络结构，增加隐藏层，调整神经元数量。(0.5 分)
 - 尝试使用不同的优化器，如 Adam、SGD 等，观察其对训练效果的影响。(1 分)
 - 添加 Dropout 层，以减少过拟合，并观察其对训练效果的影响。(1 分)

提交要求：

- 在 10 个 epoch 后，测试集的准确率需达到 97%以上。(1.5 分)
- 提交一份实验报告，内容包括训练过程中打印的准确率截图（确保截图中展示了每个 epoch 结束时的准确率），所做的修改描述。实验报告格式不限，勿长篇大论。