**作业1：用numpy实现训练MLP网络识别手写数字MNIST数据集**

**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

**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。