**作业1：用numpy实现训练MLP网络识别手写数字MNIST数据集**

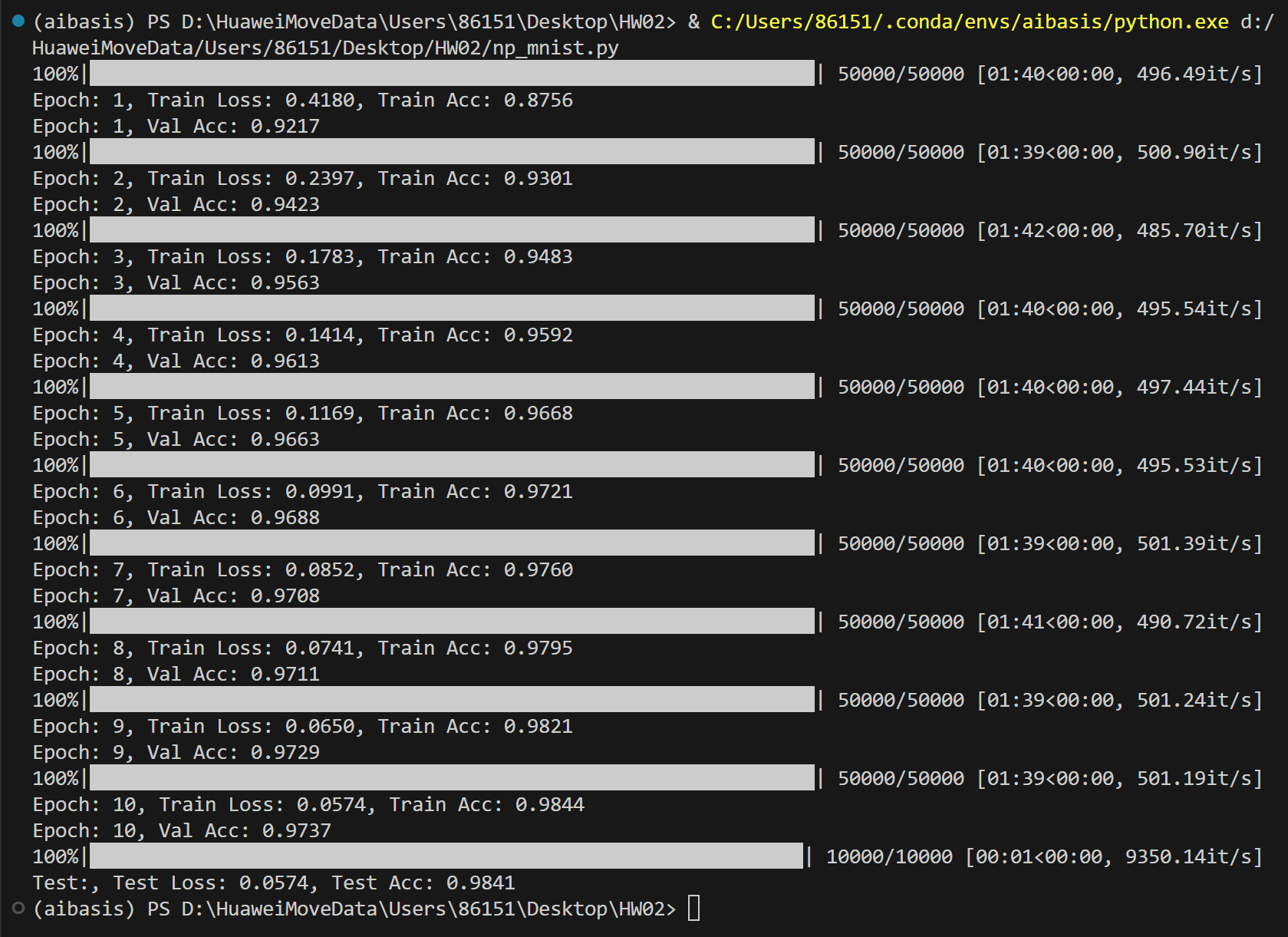
**任务描述：**

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：
   * 更改损失函数（例如，可以尝试使用交叉熵损失函数）。
   * 调整网络结构，包括但不限于更改隐藏层数量、神经元数量。
   * 尝试不同的激活函数（如ReLU、sigmoid、tanh等）。
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。

**要求：**

* 在10个epoch后，测试集的准确率需达到94%以上。
* 完成，2024年3月27日19:01

**提交方式：**

* 提交np\_mnist.py代码文件。
* 提交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。
* 

图中每个进度条代表一个epoch，最后一个进度条代表test测试集，最终准确率达到98.41%。

**作业2：使用Pytorch训练MNIST数据集的MLP模型**

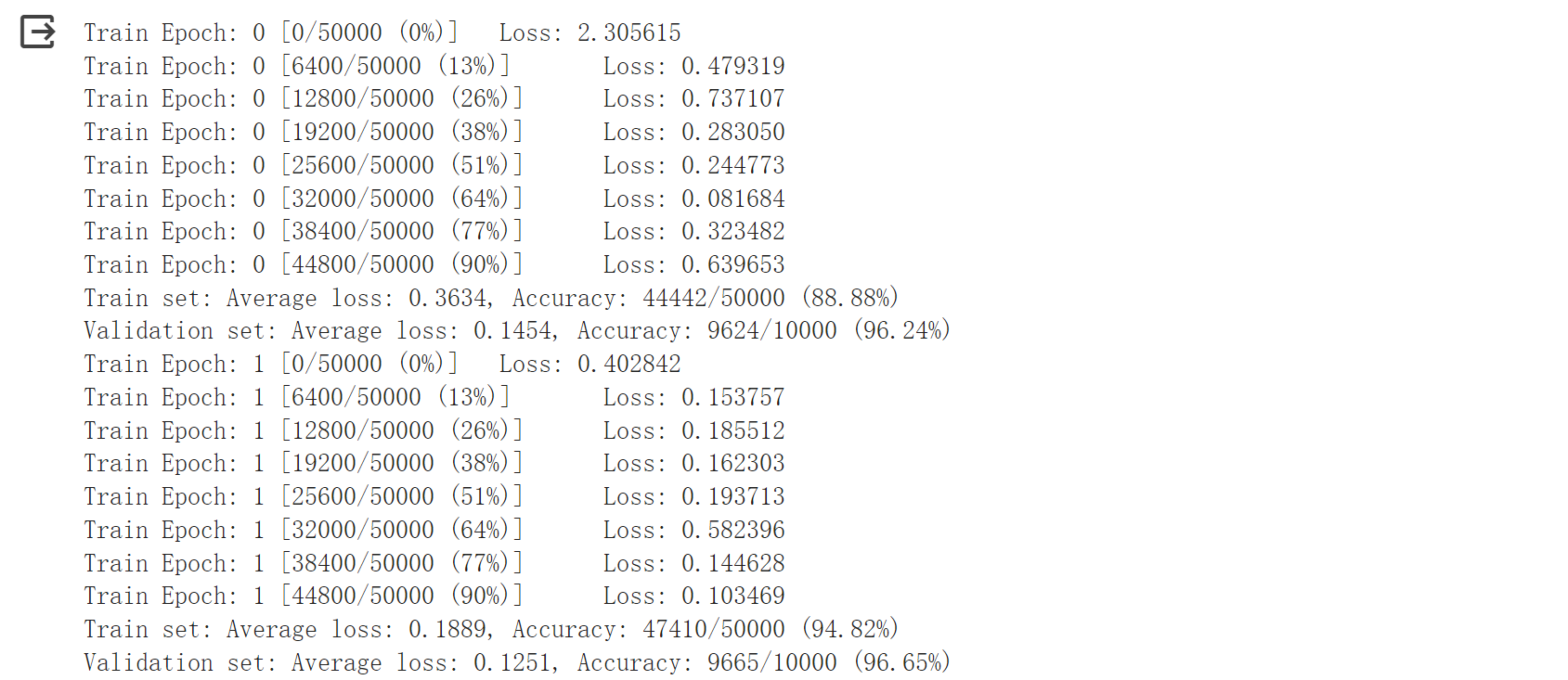
**任务描述：**

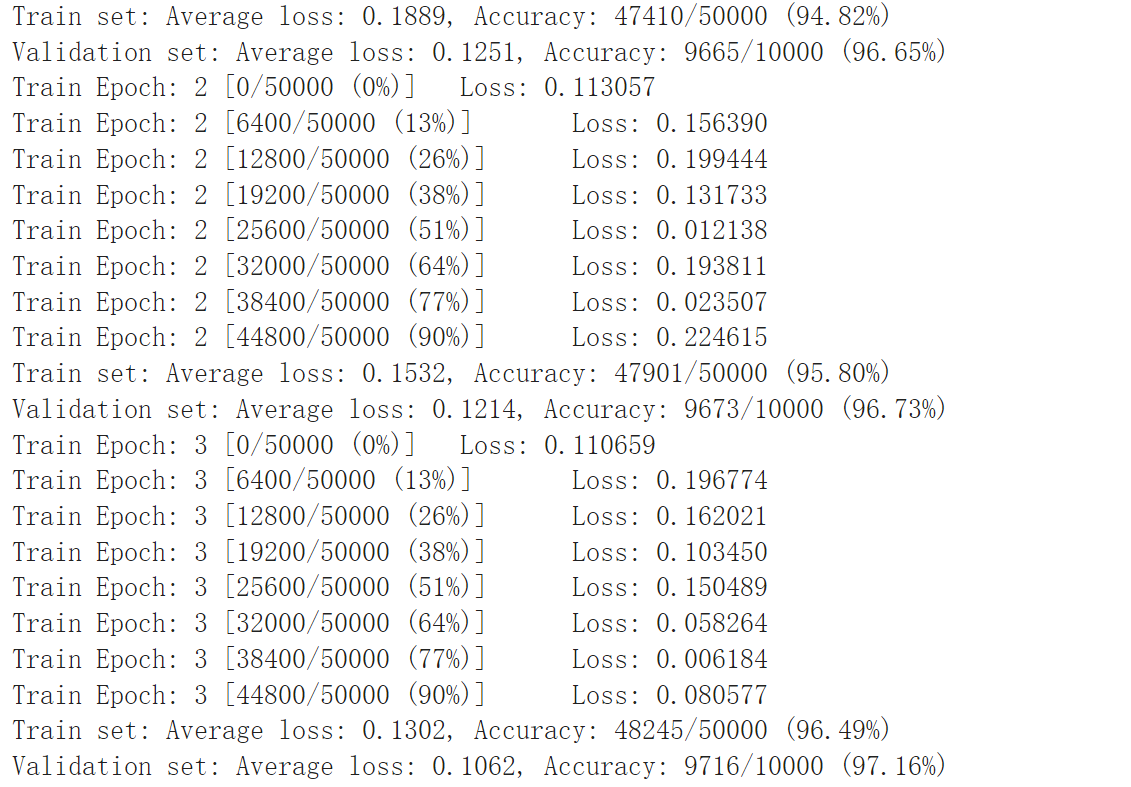
1. 运行、阅读并理解mnist\_mlp\_template.py，了解其网络结构和训练流程。
2. 复制mnist\_mlp\_template.py至mnist\_mlp.py，并根据以下要求对其进行修改：
   * 修改网络结构，增加隐藏层，调整神经元数量。
   * 尝试使用不同的优化器，如Adam、SGD等，观察其对训练效果的影响。
   * 添加Dropout层，以减少过拟合，并观察其对训练效果的影响。

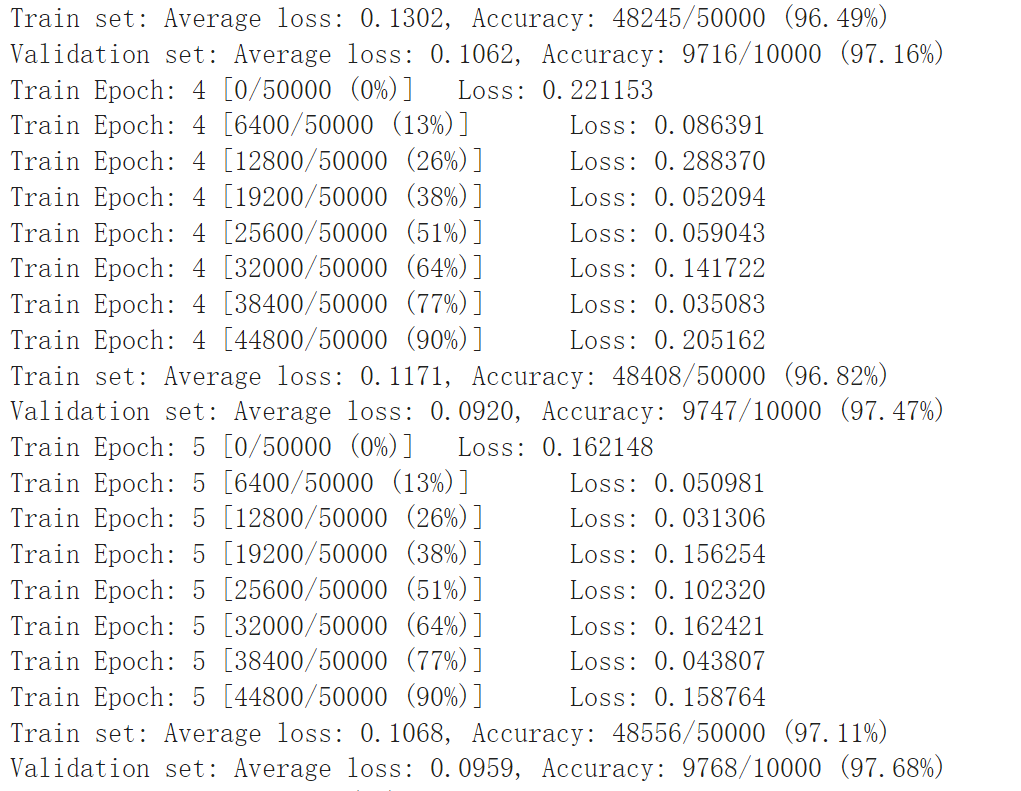
**要求：**

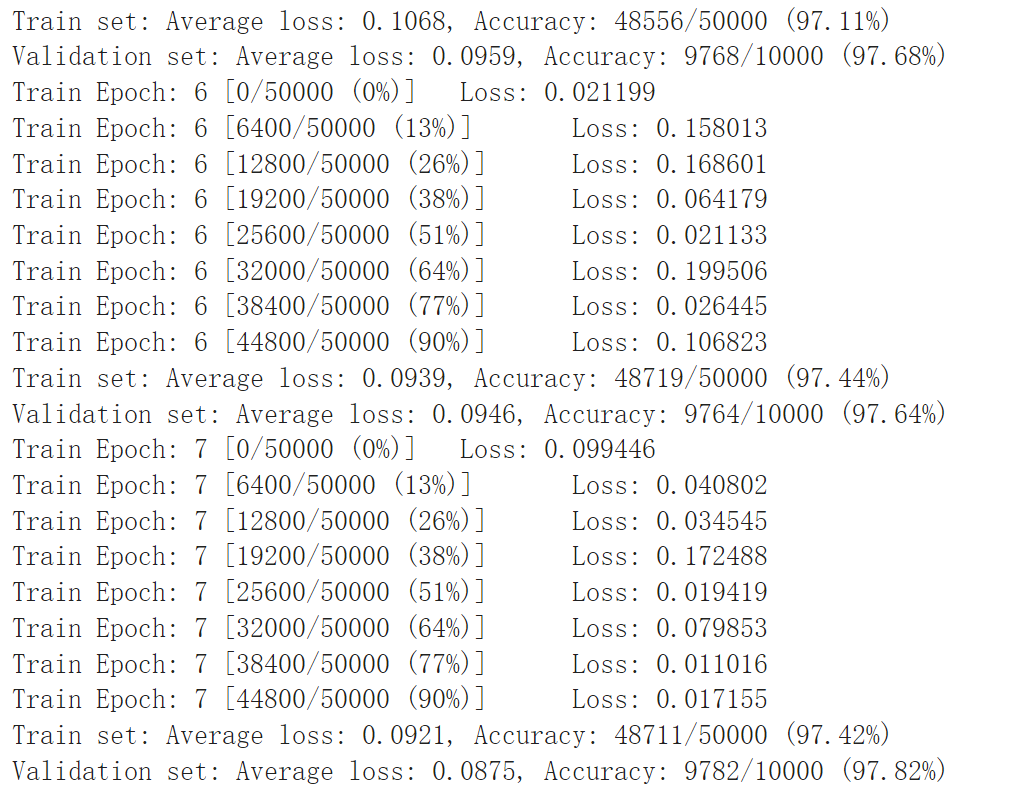
* 在10个epoch后，测试集的准确率需达到97%以上。

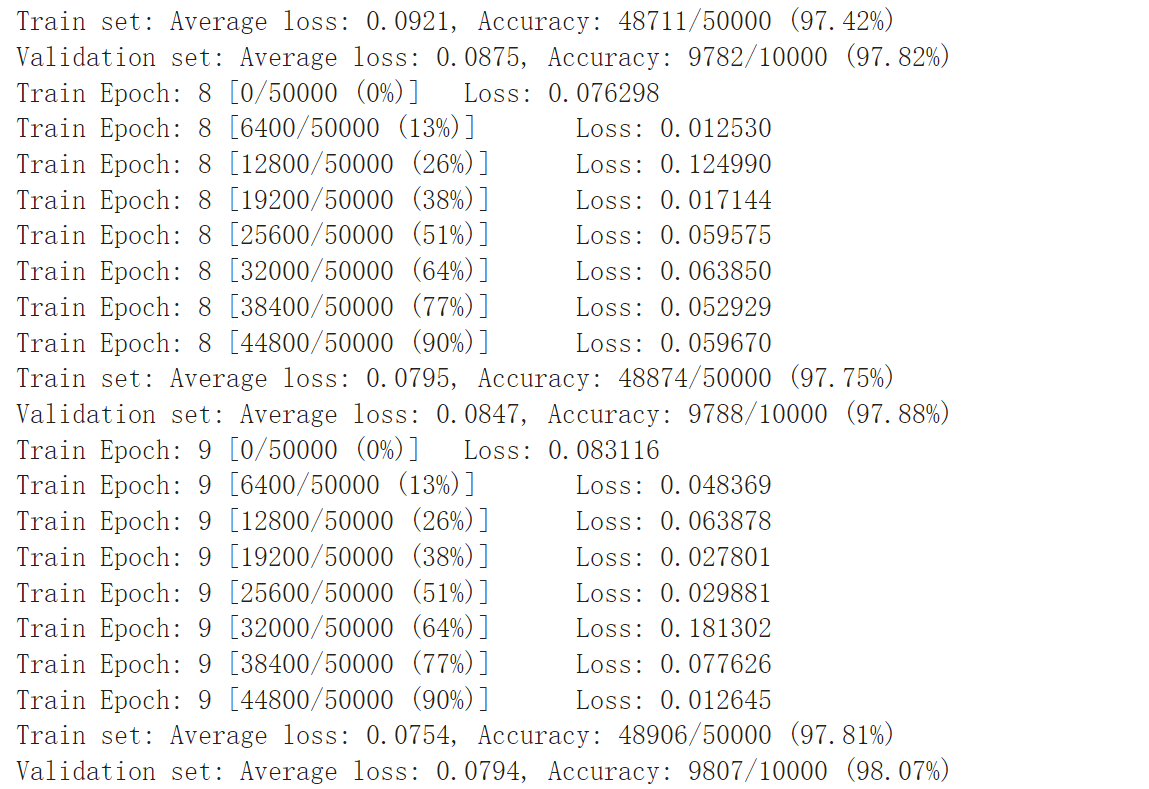
**提交方式：**

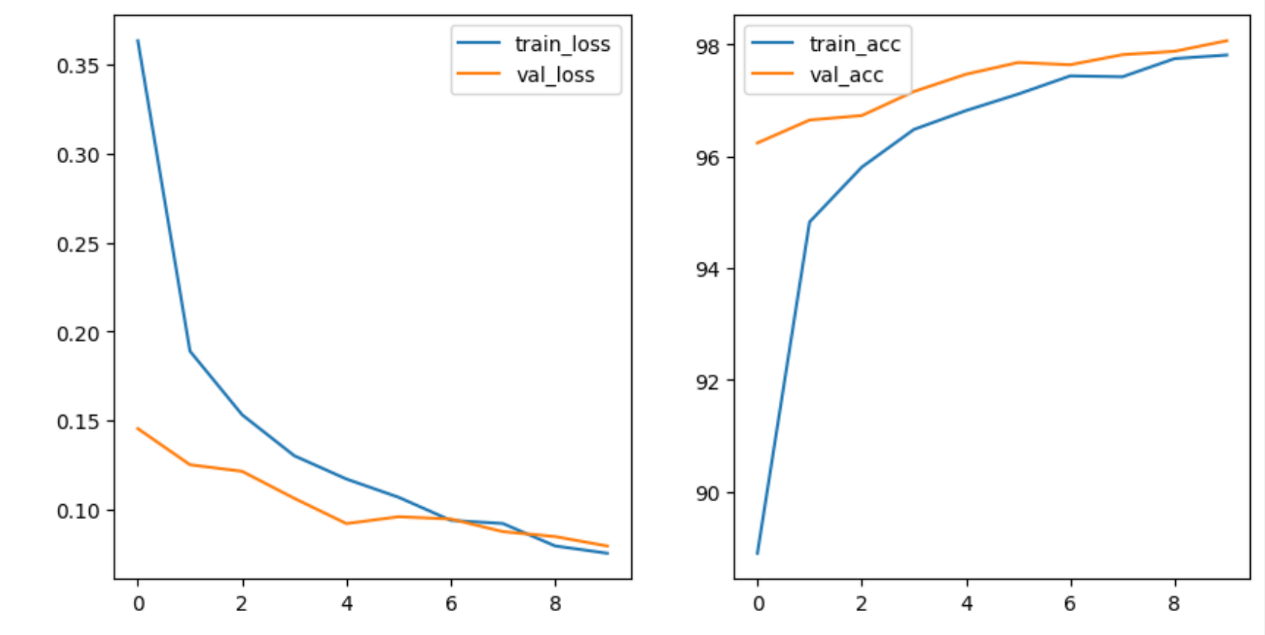
* 提交mnist\_mlp.py代码文件。
* 提交训练过程中打印的准确率截图，确保截图中展示了每个epoch结束时的准确率。
* 第一次训练：在google colab中，调用cuda，但忘记加入测试集：
* 











在google colab上的运行结果如图所示，最终验证集准确率突破了98%

第二次训练：在vs code上，调用cpu，加入了测试集，测试集准确率97.79%：

