

# 人工智能中的编程 Final Task2

苏王捷 \*

December 24, 2024

## 1 Introduction

Task2 使用 Pytorch 内置的 API, 在本地电脑上模拟了数据并行的操作, 以提高训练 MNIST 数据集分类的效率。

## 2 Implementation

使用 Task1 中的网络, 但在训练前, 如果选择并行, 则会调用 Pytorch 中的 API-DataParallel 进行数据并行操作。

运行代码, 首先会检测当前环境可用的 GPU 数量并报告。之后会进行两波次训练及测试, 第一次使用无数据并行的网络, 第二次使用数据并行的网络。

针对每次训练及测试, 会在训练集上进行 10 个 Epoch 的训练并在进度条中报告训练过程。训练完成后, 保存模型参数到 models 文件夹, 并在测试集上对网络进行测试。反馈各 Epoch 训练结果下的 loss 变化曲线, 及针对全体和各类别的分类准确率情况直方图。最后, 会比较不并行和并行情况下的训练时间以及准确率, 生成结果比较的直方图。图片保存在 results 文件夹, 命名格式为 *Loss/Accuracy-parallel-False/True.png*。

运行过程截图见下图<sup>1</sup>

```
(code_in_ai) E:\PKU\code_in_ai\homework\Final\Task2>python mnist.py
Number of available GPUs: 1
Start Training
Epoch: 1/10: 100%| 3750/3750 [00:11<00:00, 331.15batch/s, loss=0.171]
Epoch: 2/10: 100%| 3750/3750 [00:11<00:00, 337.92batch/s, loss=0.100]
Epoch: 3/10: 100%| 3750/3750 [00:10<00:00, 344.54batch/s, loss=0.078]
Epoch: 4/10: 100%| 3750/3750 [00:10<00:00, 357.61batch/s, loss=0.064]
Epoch: 5/10: 100%| 3750/3750 [00:10<00:00, 365.51batch/s, loss=0.047]
Epoch: 6/10: 100%| 3750/3750 [00:10<00:00, 366.83batch/s, loss=0.043]
Epoch: 7/10: 100%| 3750/3750 [00:10<00:00, 353.89batch/s, loss=0.038]
Epoch: 8/10: 100%| 3750/3750 [00:10<00:00, 364.50batch/s, loss=0.037]
Epoch: 9/10: 100%| 3750/3750 [00:10<00:00, 356.54batch/s, loss=0.036]
Epoch: 10/10: 100%| 3750/3750 [00:11<00:00, 336.93batch/s, loss=0.028]
Finished Training in 106.82 seconds
Accuracy of the total: 97.80%
Start Training
Epoch: 1/10: 100%| 3750/3750 [00:11<00:00, 325.40batch/s, loss=0.169]
Epoch: 2/10: 100%| 3750/3750 [00:11<00:00, 334.64batch/s, loss=0.092]
Epoch: 3/10: 100%| 3750/3750 [00:11<00:00, 323.09batch/s, loss=0.077]
Epoch: 4/10: 100%| 3750/3750 [00:11<00:00, 318.55batch/s, loss=0.067]
Epoch: 5/10: 100%| 3750/3750 [00:11<00:00, 320.71batch/s, loss=0.052]
Epoch: 6/10: 100%| 3750/3750 [00:11<00:00, 318.42batch/s, loss=0.048]
Epoch: 7/10: 100%| 3750/3750 [00:11<00:00, 331.03batch/s, loss=0.041]
Epoch: 8/10: 100%| 3750/3750 [00:11<00:00, 329.86batch/s, loss=0.044]
Epoch: 9/10: 100%| 3750/3750 [00:11<00:00, 332.70batch/s, loss=0.028]
Epoch: 10/10: 100%| 3750/3750 [00:11<00:00, 336.68batch/s, loss=0.036]
Finished Training in 114.69 seconds
Accuracy of the total: 98.20%
```

Figure 1: Process of training and test

\* College of Engineering, Peking University, wsu0605@stu.pku.edu.cn

在 Task2 文件夹下，打开命令行，运行命令 `python mnist.py` 即可。

### 3 Results

使用和 Task1 相同的损失函数和优化器设置，训练 10 个 Epoch 后，不进行并行的准确率为 97.80%，训练时间为 106.8 秒；进行并行的准确率为 98.20%，训练时间为 114.7 秒。结果见下图3和4, 两者对比结果见2。

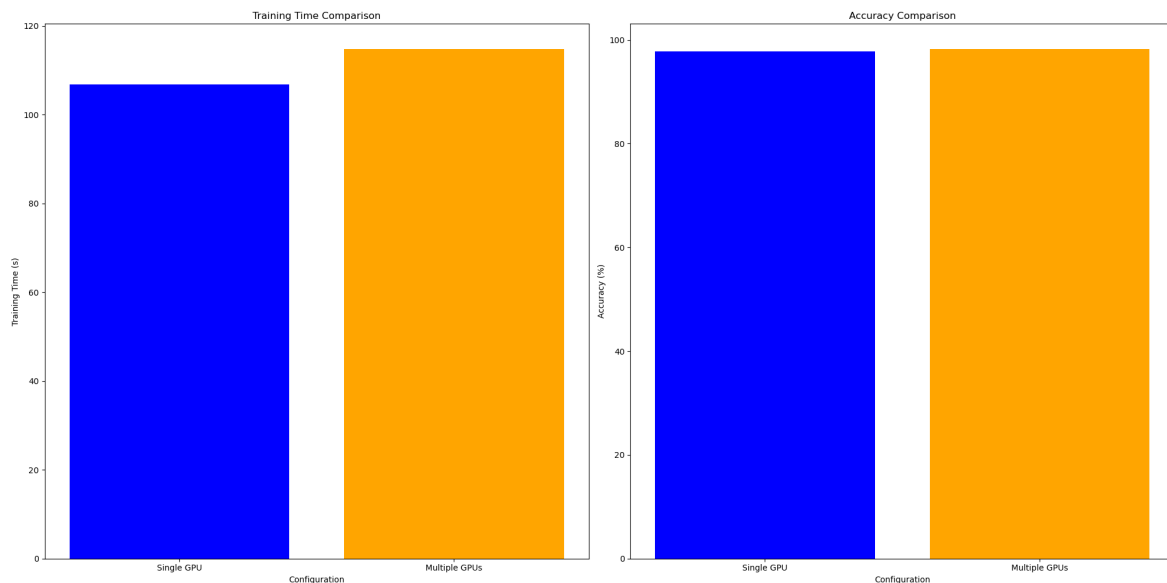
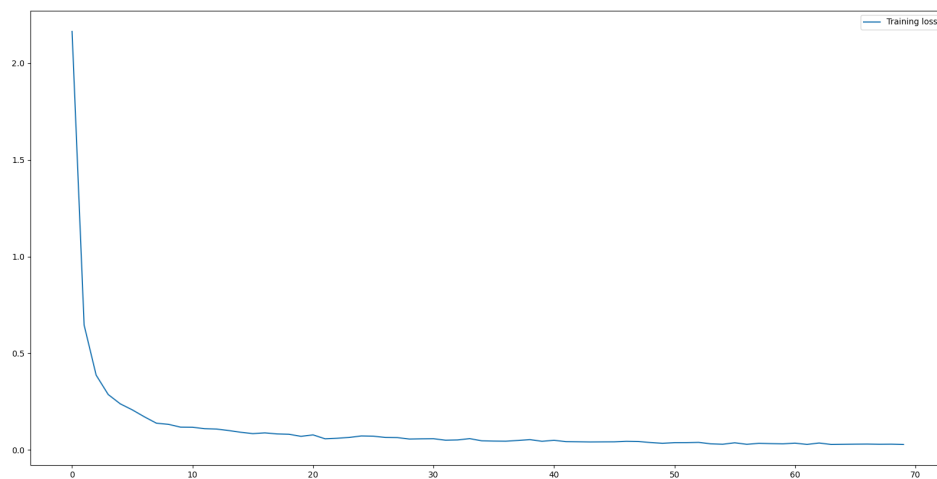
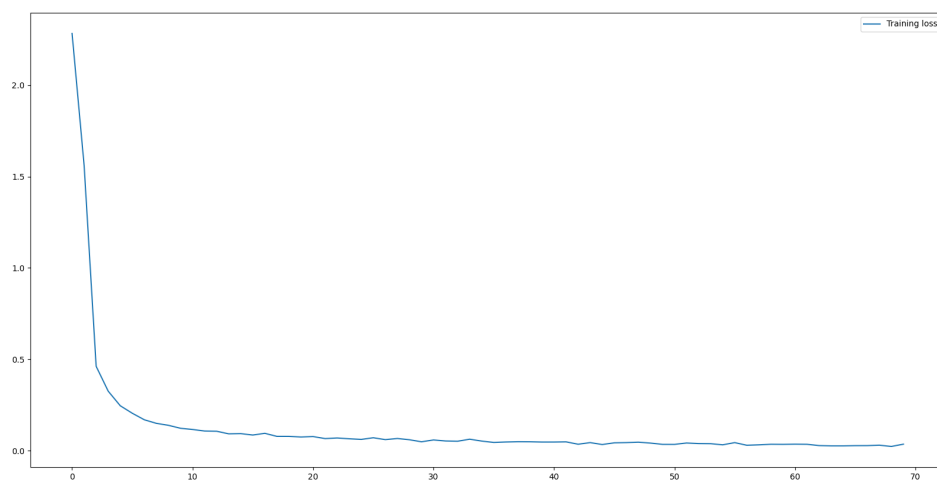


Figure 2: Results comparison

可以发现，在我本地电脑单 GPU 的情况下，并行操作和不并行操作在准确率上并无明显差别。理论上，并行操作的训练时间应当在多 GPU 条件下比不并行更快，但在本地的单 GPU 条件下，由于并行操作的数据传输时间，并行操作的训练时间更长。

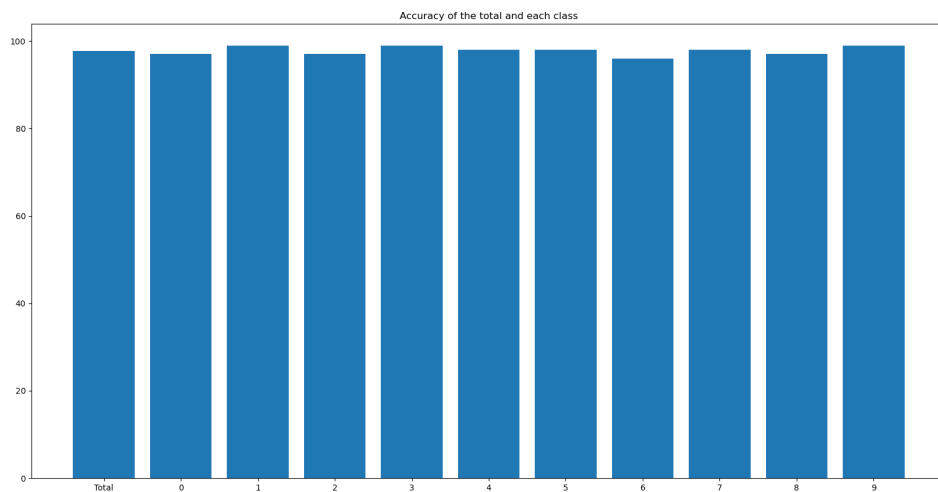


(a) Unparalleled

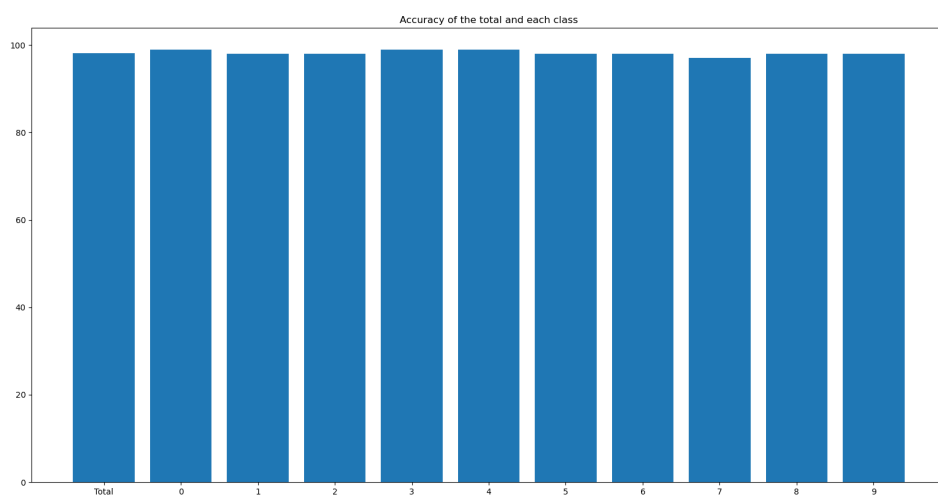


(b) Paralleled

Figure 3: Results of loss



(a) Unparallelled



(b) Paralleled

Figure 4: Results of Accuracy