

Answers for Task 5.1P

Relevant models:

- CIFAR_Model: Model trained on the CIFAR10 dataset on the local device.
- Food_Model: Model trained on the food dataset from task 4.1P on the local device.
- GPU_Model: Model trained on the CIFAR10 dataset using GPUs using google collab.

Section 2:

- The CNN CIFAR_Model appears to yield better results when using the finetuned version of the model Compared to using the Food_Model trained from scratch for the **food dataset**. The accuracy for each of the 3 food items is higher for the fine-tuned CIFAR_Model and the confusion matrix also are observed to be better as well.

Section 3:

- When adding the dropout method into the network for CIFAR_Model and using it to predict the CIFAR dataset, the accuracy seems to have reduced when using the dropout method.

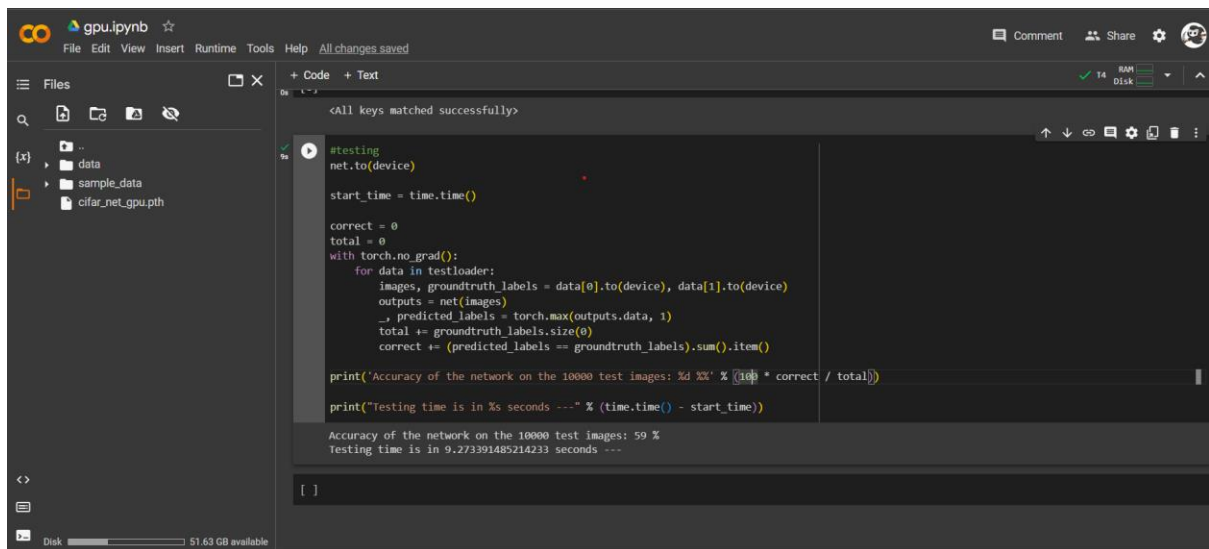
Section 4:

- Accuracy of GPU_Model used for predicting the CIFAR10 dataset, is observed to be the same as the CIFAR_Model but I did observe it to be significantly faster than CIFAR_Model in terms of processing time.

Accuracy Value:

- CIFAR_Model on CIFAR Dataset: 59%
- Fine tuned CIFAR_Model on Food Dataset: 71%
- Food_Model on Food Dataset: 67%
- CIFAR_model with dropout on CIFAR dataset: 52%
- GPU_Model on CIFAR dataset: 59%

GPU:



The screenshot shows a JupyterLab environment with a dark theme. On the left, a file browser displays a directory structure with files like 'data', 'sample_data', and 'cifar_net_gpu.pth'. The main area contains a code cell with the following Python code:

```
<All keys matched successfully>

#testing
net.to(device)

start_time = time.time()

correct = 0
total = 0
with torch.no_grad():
    for data in testloader:
        images, groundtruth_labels = data[0].to(device), data[1].to(device)
        outputs = net(images)
        _, predicted_labels = torch.max(outputs.data, 1)
        total += groundtruth_labels.size(0)
        correct += (predicted_labels == groundtruth_labels).sum().item()

print('Accuracy of the network on the 10000 test images: %d %%' % ((100 * correct) / total))

print("Testing time is in %s seconds ---" % (time.time() - start_time))
```

The output of the code cell shows the accuracy and testing time:

```
Accuracy of the network on the 10000 test images: 59 %
Testing time is in 9.273391485214233 seconds ---
```

At the bottom left, a disk usage indicator shows '51.63 GB available'.