AI 417 – DL

Lab 5

جامعة الأمير مقرن بن عبد العزيز

University of Prince Mugrin

Department of Artificial Intelligence

College of Computer and Cyber Sciences

Introduction to Deep Learning

Fine-Tuning a Pre-Trained Model in PyTorch

1. Learning Objectives

By the end of this lab, students will:

- Learn what is a fine-tuned model.
- Be able to use a fine-tuned ResNet model for MNIST Classification.
- Differentiate between a pre-trained model and a fine-tuned model.

2. Explanation of Key Concepts

- **Fine-Tuning:**

  Fine-tuning in deep learning refers to the process of adapting a pre-trained model to perform better on specific tasks or datasets. This technique has become essential, especially when working with foundation models in generative AI. By starting with a model that has already learned general patterns from large datasets, fine-tuning allows for more efficient training and better performance on specialized tasks.

- **ResNet Model:**

  ResNet (Residual Network) is a deep convolutional neural network (CNN) that introduced the concept of residual connections, enabling the training of much deeper networks. By utilizing these connections, ResNet alleviates the vanishing gradient problem, allowing for more effective learning in networks with hundreds or even thousands of layers. Trained on large datasets like ImageNet, ResNet has become a powerful model for image classification and other computer vision tasks.

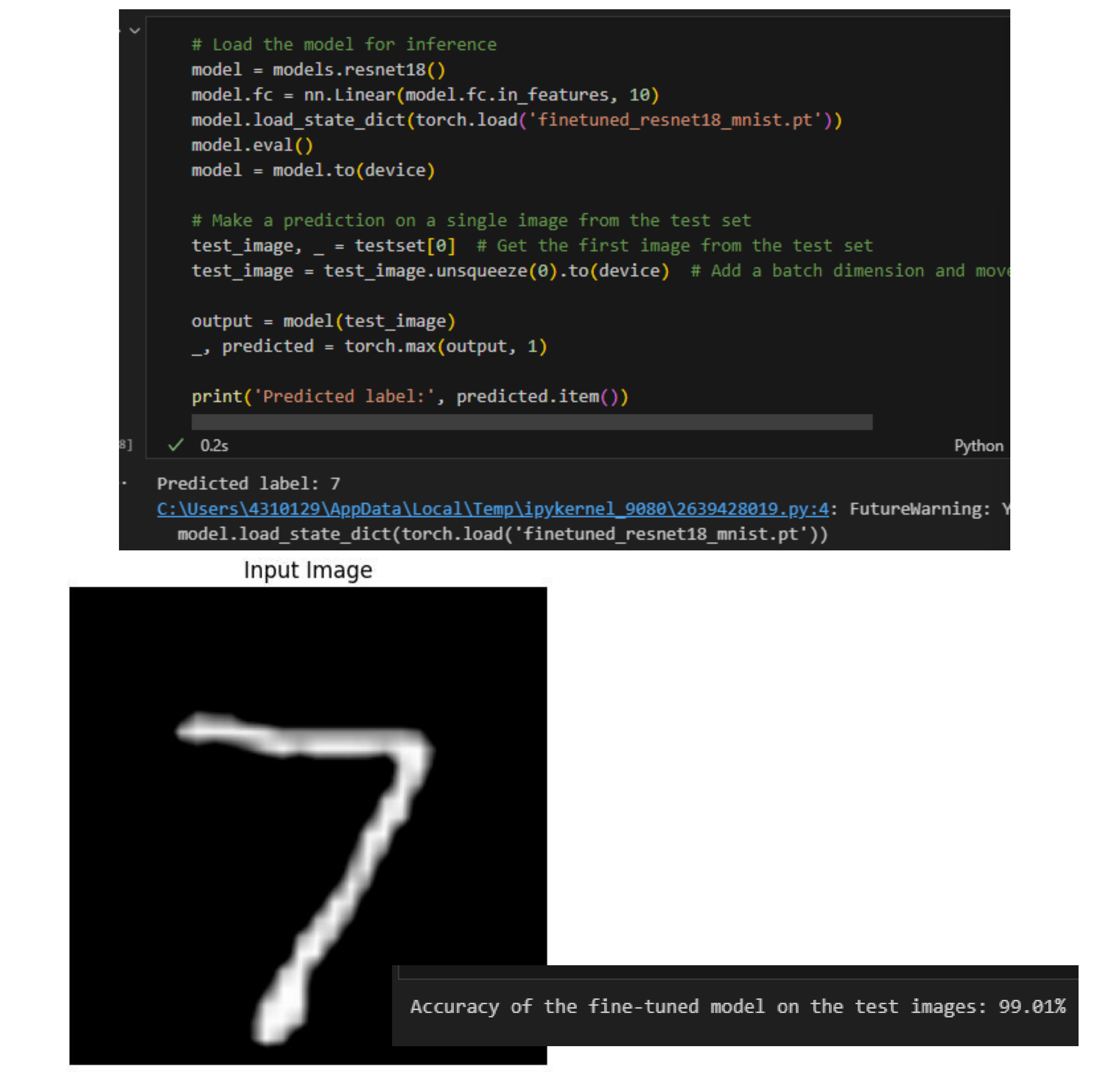
3. Activities

➢ Exercise 1: Using ResNet for MNIST Classification.

In this exercise, we will fine-tune a pre-trained ResNet model for image classification on the MNIST dataset using PyTorch. You will load the model, modify its architecture to match the number of classes in MNIST, preprocess the dataset, train the model, and evaluate its performance.

```
# Code provided in the notebook
```

Screenshot of the result:



```python
# Load the model for inference
model = models.resnet18()
model.fc = nn.Linear(model.fc.in_features, 10)
model.load_state_dict(torch.load('finetuned_resnet18_mnist.pt'))
model.eval()
model = model.to(device)

# Make a prediction on a single image from the test set
test_image, _ = testset[0]  # Get the first image from the test set
test_image = test_image.unsqueeze(0).to(device)  # Add a batch dimension and move
```
```python
output = model(test_image)
_, predicted = torch.max(output, 1)

print('Predicted label:', predicted.item())
```

```
Predicted label: 7
C:\Users\4310129\AppData\Local\Temp\ipykernel_9080\2639428019.py:4: FutureWarning: Y
  model.load_state_dict(torch.load('finetuned_resnet18_mnist.pt'))
```

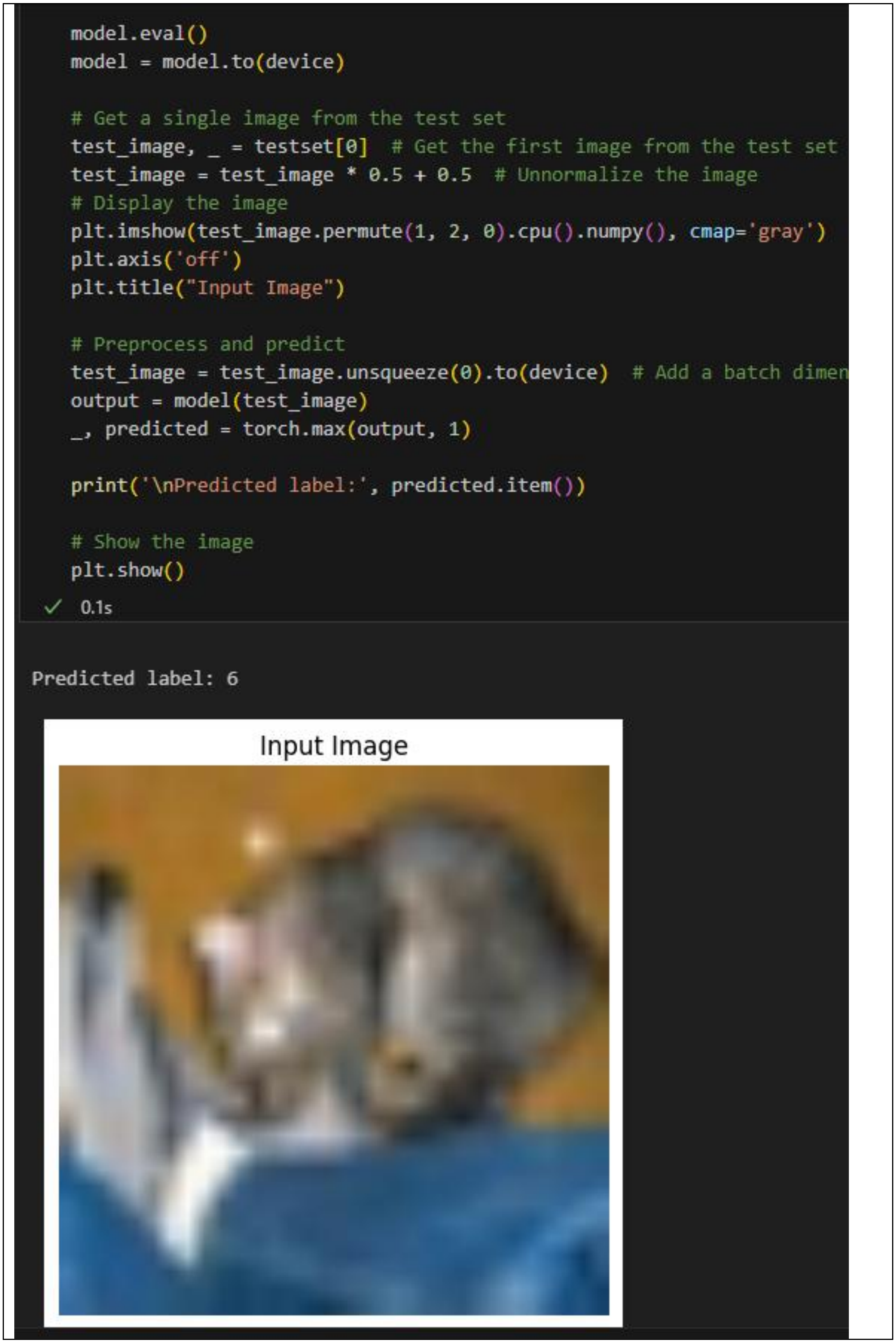
```
Accuracy of the fine-tuned model on the test images: 99.01%
```

4. Tasks

➢ Task 1:

In this task, you will modify the code from Exercise 1 to fine-tune ResNet on Cifar10 dataset instead of MNIST.

Submit your jupyter notebook & Add a screenshot of the result:



```
model.eval()
model = model.to(device)

# Get a single image from the test set
test_image, _ = testset[0]  # Get the first image from the test set
test_image = test_image * 0.5 + 0.5  # Unnormalize the image
# Display the image
plt.imshow(test_image.permute(1, 2, 0).cpu().numpy(), cmap='gray')
plt.axis('off')
plt.title("Input Image")

# Preprocess and predict
test_image = test_image.unsqueeze(0).to(device)  # Add a batch dimen
output = model(test_image)
_, predicted = torch.max(output, 1)

print('\nPredicted label:', predicted.item())

# Show the image
plt.show()
```

✓ 0.1s

```
Predicted label: 6
```

➢ Task 2:

Complete the table below by identifying the key differences between **pre-training** and **fine-tuning**.
(Using ChatGPT or any other Chatbot is completely prohibited).

Feature	Pre-Training	Fine-Tuning
Definition	A model that is trained for a specific task using some dataset. Then it can be used by others for the exact same task without any re-training or changes.	Taking an existing model that was trained on a specific dataset, and re-training on my own dataset to make it more suitable for my task.
Dataset	TinyImage	CIFAR-10 If my task only requires the 10 classes of CIFAR10, then I could fine-tune an existing model on this dataset to improve accuracy.
Training Time	Hours I'm assuming, because TinyImage is huge	1 minute 17 seconds
Example	ResNet, YOLO	Re-training the ResNet model on CIFAR10

5. References

What is Fine-Tuning? | IBM

Fine tuning Vs Pre-training. The objective of my articles is to… | by Eduardo Ordax | Medium

Residual Networks (ResNet) - Deep Learning - GeeksforGeeks

Fine-Tuning a Pre-Trained Model in PyTorch: A Step-by-Step Guide for Beginners - DEV Community