Video Transcript

Hi, my name is Stephen and this is my video for the pytorch assignment. The assignment involves a neural network that is trained to recognize handwritten characters as numbers between 0 and 9. During testing, the neural network receives an image which is a 28x28 grid of numbers with values between 0 and 1; 0 for white and 1 for black. The neural network is then asked to classify the value of the image. For example, when the network is shown an image of a handwritten 5, it should output the value 5.

The neural network makes decisions by first converting the grid of numbers to a vector 784 numbers long. Then it multiplies this vector by weights in two hidden layers to produce a final output. The final layer has 10 neurons, one for each number. Since the input and output are fixed, the only way to improve the performance of the network is to change its weights. Initially, the weights are random and the performance of the network is poor - the accuracy of the network is initially only about 5%.

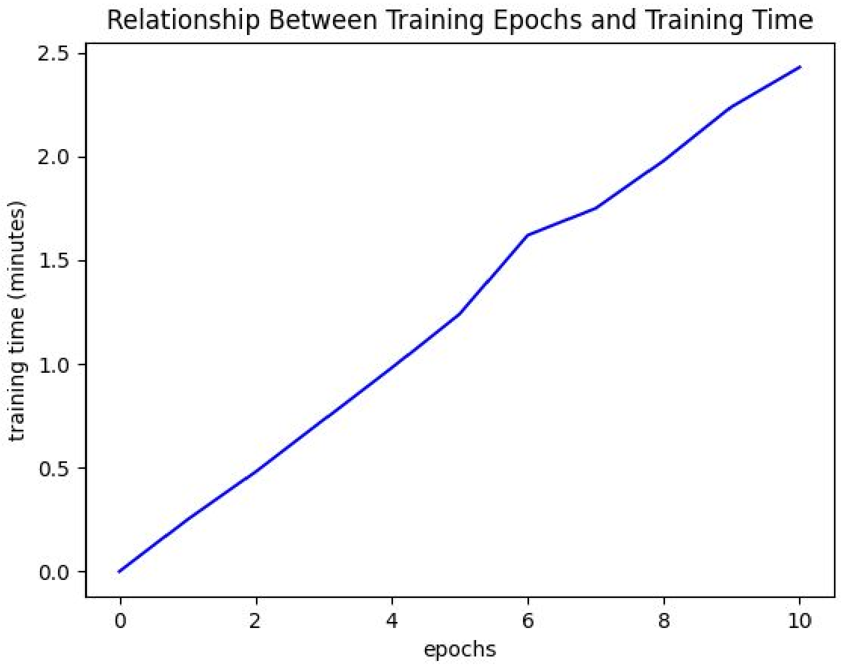
During training, the network generates a label for each image which is compared to correct the human labels. For example, if the image is a handwritten character of a three and the network outputs the number three, then the human label and the label produced by the network will be the same meaning that the network got the right answer. Otherwise, the network was wrong. Each wrong label by the network produces an error which is then used in the process of backpropagation to adjust the values of the weights. The more inaccurate the network is, the more the weights are adjusted.

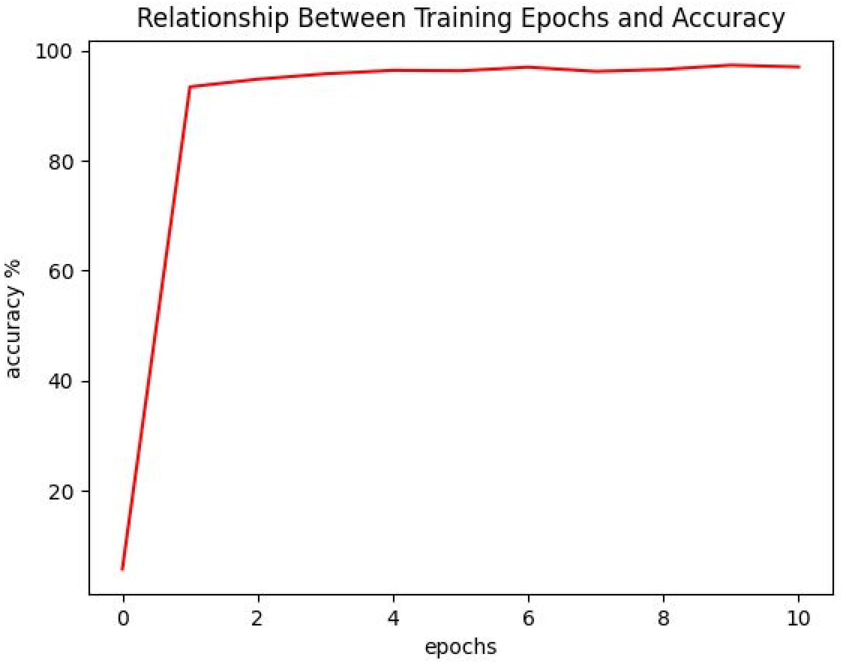
My goal in this assignment was to measure the effect of changing the number of training epochs on the accuracy of the neural network. Each epoch involves inputting 60,000 training images to the network, testing its predictions and then correcting the values of the weights based on the errors made during training. Since each epoch involves 60,000 predictions, the performance of the network improves significantly with each epoch. After just the first epoch, the accuracy of the network increased from 5% to about 93%. However, as the error decreased, the neural network changed its weights by progressively smaller increments. I then tested training the neural network with 2 epochs, 3 epochs and so on. Training the neural network with 10 epochs results in an accuracy of 97%.

I found that the training time increased linearly or in O(n) time. For example, training the neural network with 10 epochs takes about 10 times more time than training it once. Since the training time increases linearly with each epoch but the accuracy increases only logarithmically, there are diminishing returns to an increased number of training epochs and increased training.

Graphs

Below are two graphs showing the program's output. Training time increases linearly with the number of epochs whereas accuracy increases logarithmically.





Code

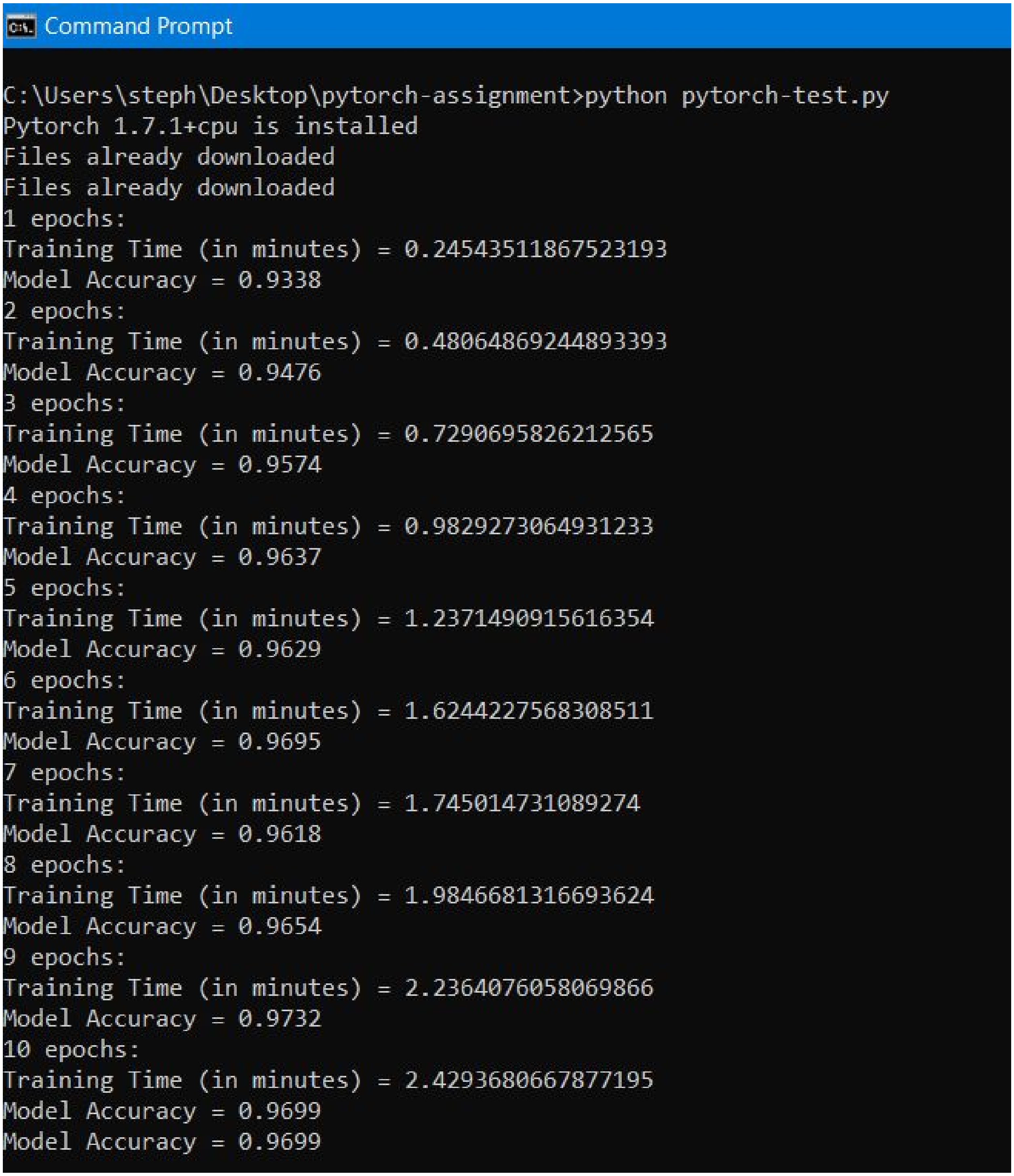
Here is the code I used to train the neural network with an increasing number of epochs.

```
for i in range(1, 11):
    print("{} epochs".format(i))
    neural_network = make_network(input_size, hidden_sizes, output_size)
    train_neural_network(neural_network, train_loader, learning_rate, i)
    test(neural_network, test_loader)
```

This is the main loop of the program, the full code is in the same directory as this document.

The code creates a new neural network, trains it and then tests its accuracy. The network is trained with 1 to 10 epochs and the accuracy and time are printed in the terminal.

Here is a screenshot of the program’s output:



```
C:\Users\steph\Desktop\pytorch-assignment>python pytorch-test.py
Pytorch 1.7.1+cpu is installed
Files already downloaded
Files already downloaded
1 epochs:
Training Time (in minutes) = 0.24543511867523193
Model Accuracy = 0.9338
2 epochs:
Training Time (in minutes) = 0.48064869244893393
Model Accuracy = 0.9476
3 epochs:
Training Time (in minutes) = 0.7290695826212565
Model Accuracy = 0.9574
4 epochs:
Training Time (in minutes) = 0.9829273064931233
Model Accuracy = 0.9637
5 epochs:
Training Time (in minutes) = 1.2371490915616354
Model Accuracy = 0.9629
6 epochs:
Training Time (in minutes) = 1.6244227568308511
Model Accuracy = 0.9695
7 epochs:
Training Time (in minutes) = 1.745014731089274
Model Accuracy = 0.9618
8 epochs:
Training Time (in minutes) = 1.9846681316693624
Model Accuracy = 0.9654
9 epochs:
Training Time (in minutes) = 2.2364076058069866
Model Accuracy = 0.9732
10 epochs:
Training Time (in minutes) = 2.4293680667877195
Model Accuracy = 0.9699
Model Accuracy = 0.9699
```