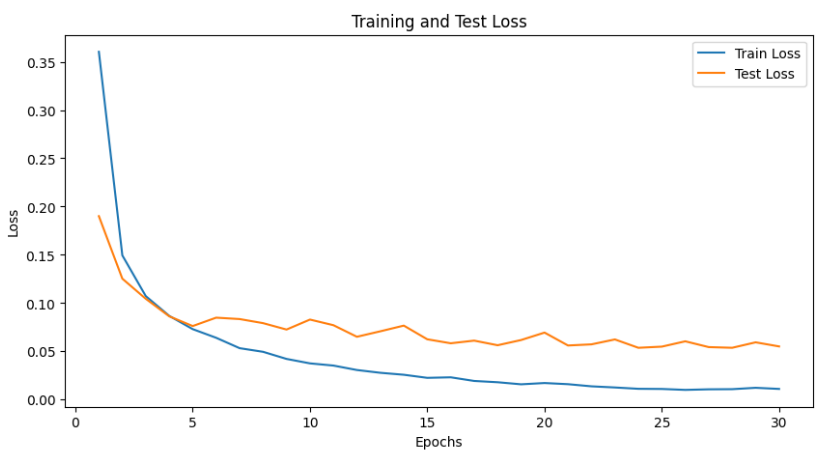
# Machine Learning - Neural Networks

Ruoheng Du

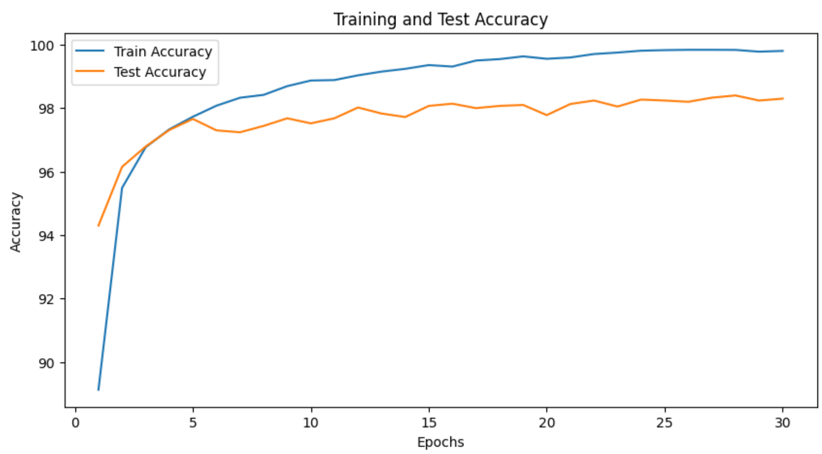
## 1 Multi-layer Perception (MLP) on MNIST

:

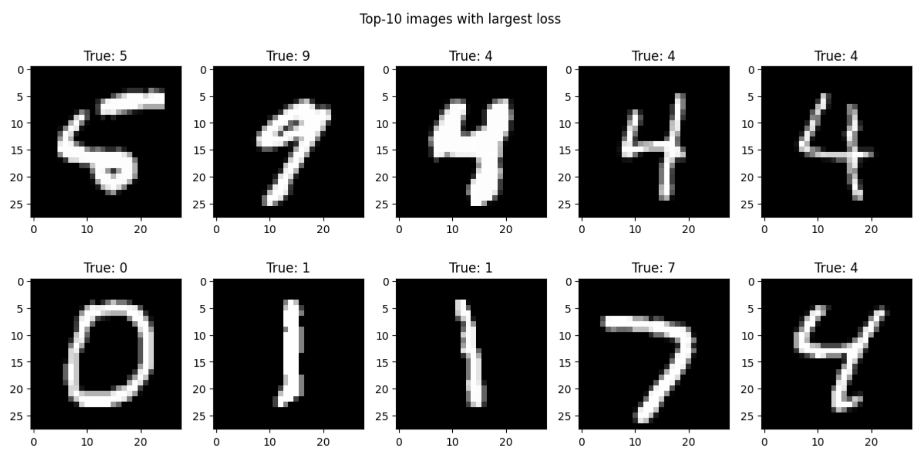
1. Show the training loss and test loss.

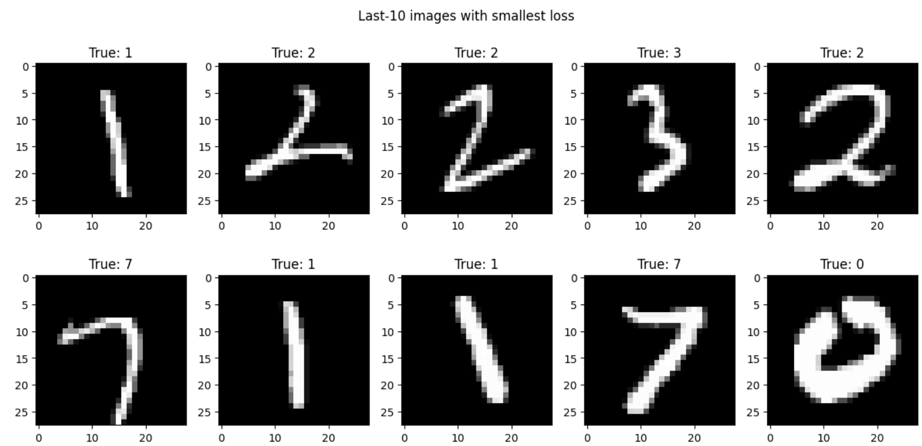



2. Show the training accuracy and test accuracy.




3. Show the top-10 test images with the largest loss, and the last-10 images with the small- est loss.



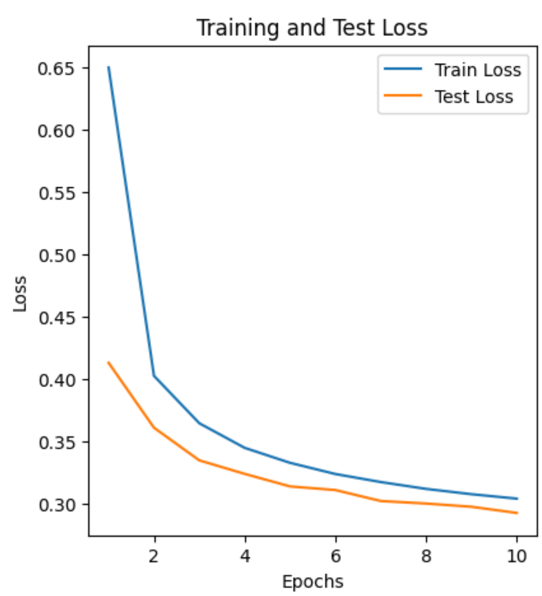


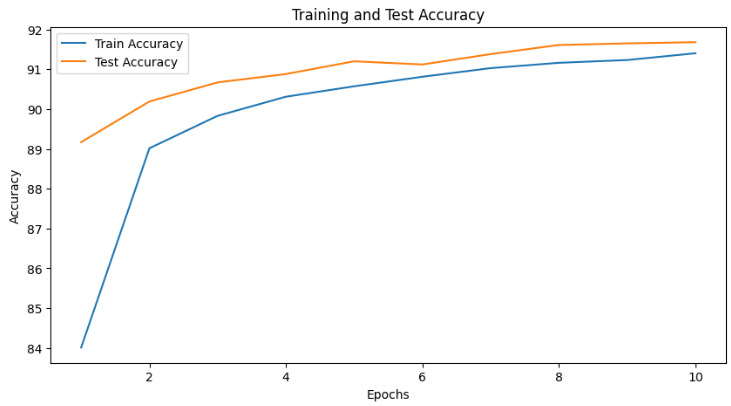

## 2 Depth and Width Trade-offs

:

1. Train MLPs with number of hidden layers equal to [1, 2, 3, 4] and all hidden layers have the same number of neurons. Also, train a MLP with 0 hidden layers and compare the results to MLPs of various depths.

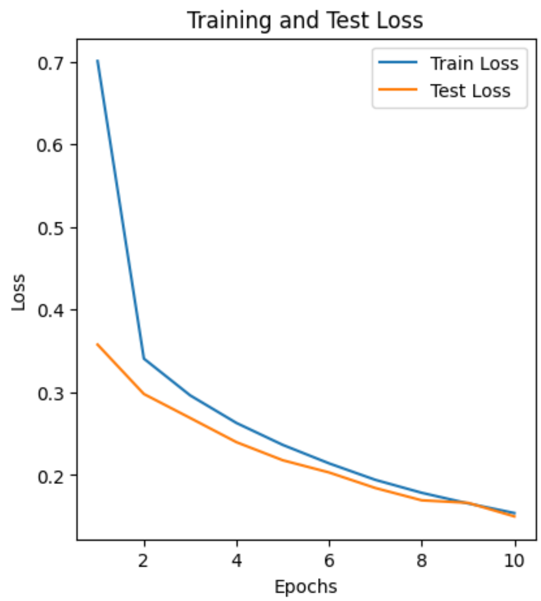
Depth: 0

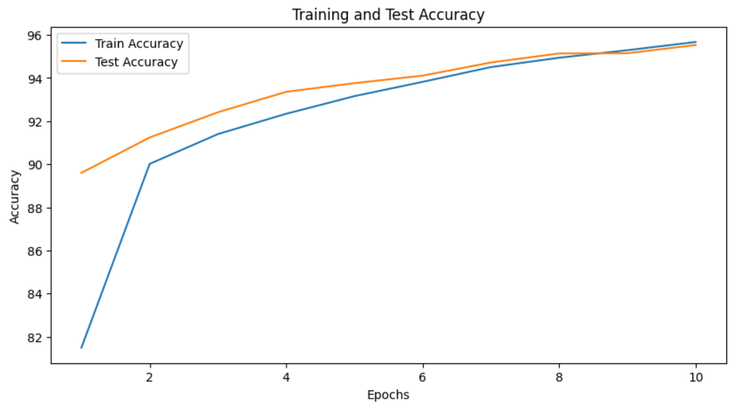





Epoch [10/10], Train Loss: 0.3036, Train Accuracy: 91.40%, Test Loss: 0.2922, Test Accuracy: 91.68%

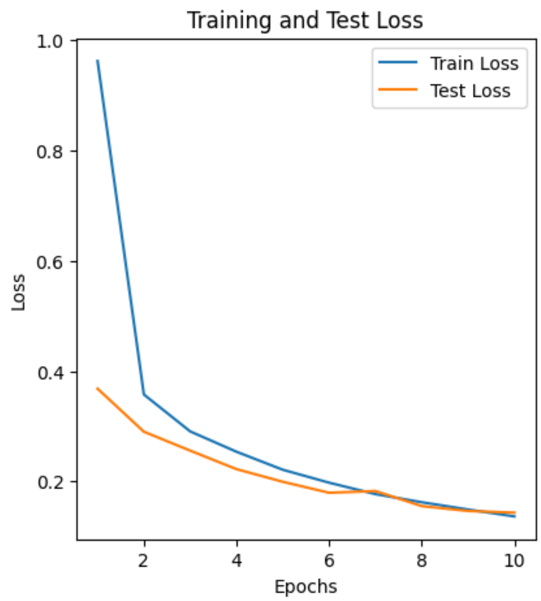
Depth: 1

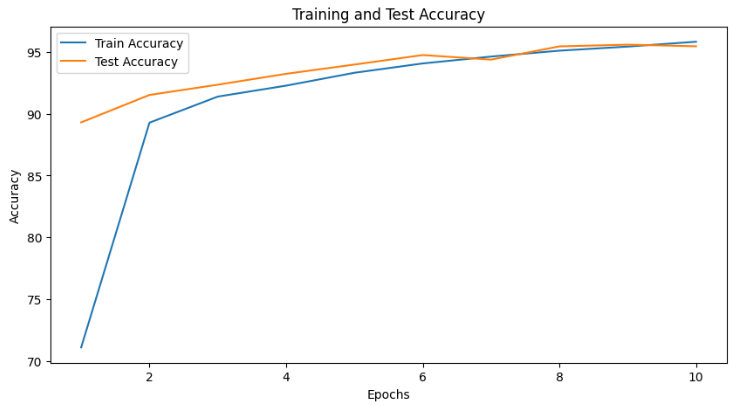





Epoch [10/10], Train Loss: 0.1540, Train Accuracy: 95.67%, Test Loss: 0.1500, Test Accuracy: 95.53%

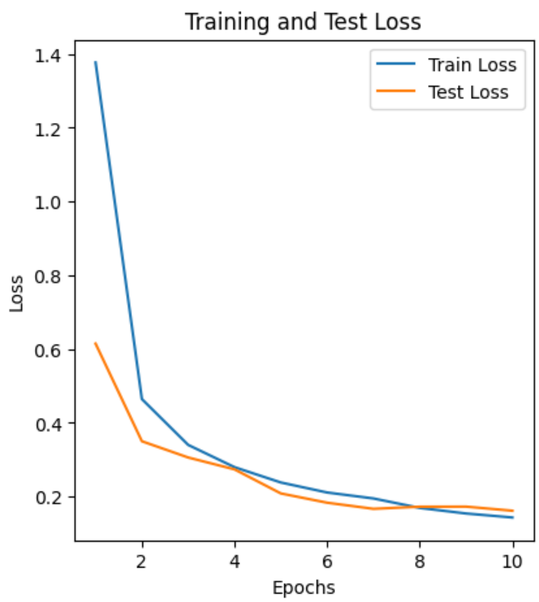
Depth: 2

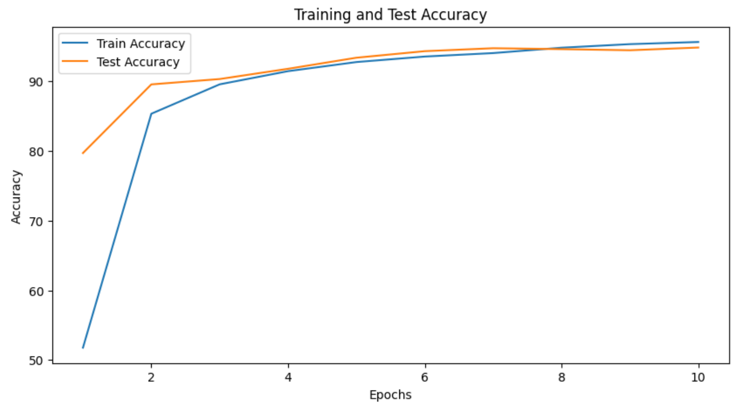





Epoch [10/10], Train Loss: 0.1365, Train Accuracy: 95.84%, Test Loss: 0.1434, Test Accuracy: 95.47%

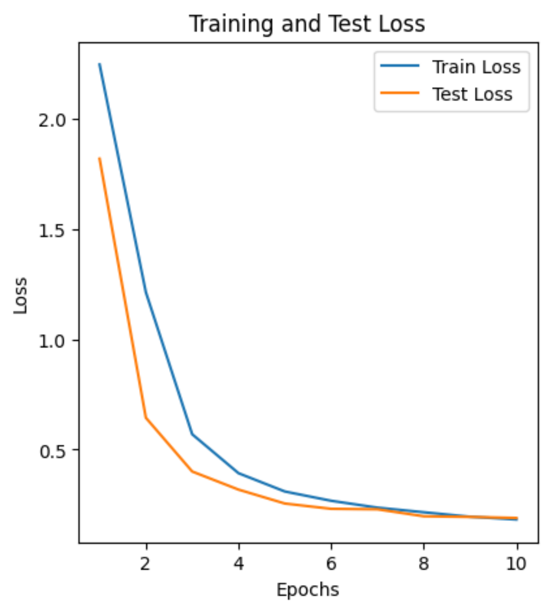
Depth: 3

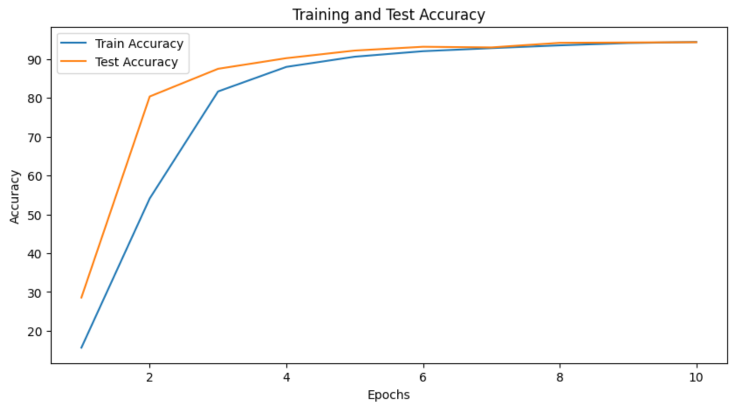





Epoch [10/10], Train Loss: 0.1433, Train Accuracy: 95.65%, Test Loss: 0.1616, Test Accuracy: 94.85%

Depth: 4






Epoch [10/10], Train Loss: 0.1832, Train Accuracy: 94.41%, Test Loss: 0.1900, Test Accuracy: 94.35%

As we can see from the above:

Depth 0: This represents a MLP with no hidden layers. It achieves an accuracy of 91.68% on the test set after 10 epochs.

Depth 1: A single-layer network with identical neuron counts in each hidden layer. It reaches a test accuracy of 95.53% after 10 epochs.

Depth 2: A two-layer network with identical neuron counts in each hidden layer. It achieves a test accuracy of 95.47% after 10 epochs.

Depth 3: A three-layer network with identical neuron counts in each hidden layer. It attains a test accuracy of 94.85% after 10 epochs.

Depth 4: A four-layer network, all layers with equal neuron counts. It reaches a test accuracy of 94.35% after 10 epochs.

It seems that the networks with depths 1 and 2 perform slightly better than the others, achieving higher test accuracies. With fixed total number of hidden neurons, deeper networks don't necessarily guarantee better performance.

2. Compute the total number of weights given the structure of a MLP. Report the total number of parameters for the MLPs trained in the previous question.

Depth 0 - Total Parameters: 7840 - Total Parameters with Bias: 7850

Depth 1 - Total Parameters: 79400 - Total Parameters with Bias: 79510

Depth 2 - Total Parameters: 42200 - Total Parameters with Bias: 42310

Depth 3 - Total Parameters: 28380 - Total Parameters with Bias: 28489

Depth 4 - Total Parameters: 21725 - Total Parameters with Bias: 21835

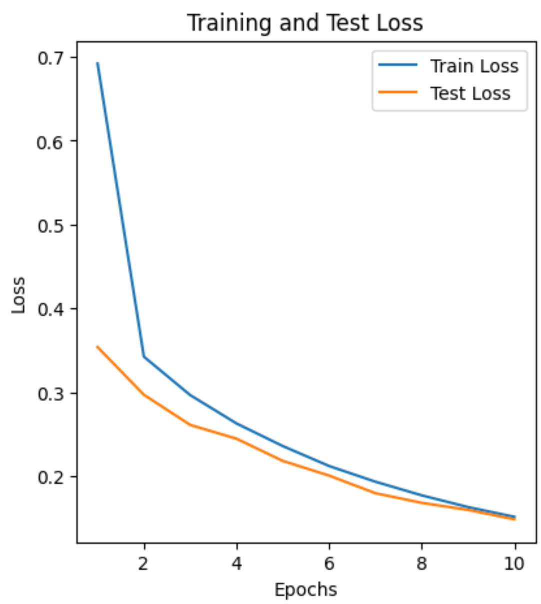
We can see that: Depth 1 has significantly more parameters than other depths, indicating a higher model complexity. On the other hand, Depth 0 has the fewest parameters but still achieves a respectable accuracy.

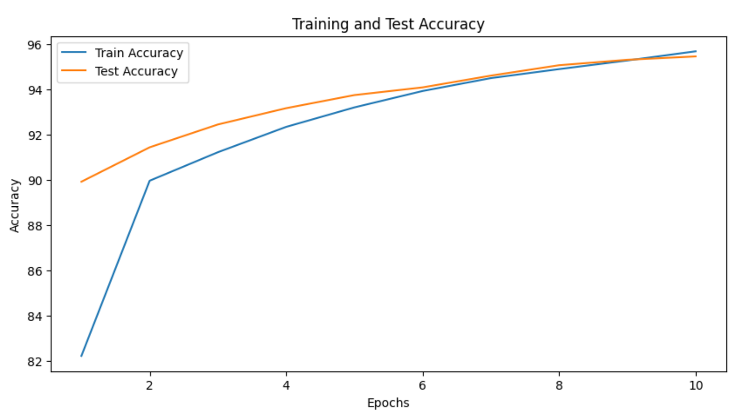
This shows an accuracy-complexity trade-off: Depth 1 achieves the highest accuracy but requires substantially more parameters compared to other depths.

Meanwhile, Depth 2 achieves a competitive accuracy with fewer parameters than Depth 1, indicating a more efficient use of parameters.

3. Now suppose we fix the total number of parameter to be the number of parameters of a single hidden layer MLP with 100 hidden neurons. Train MLPs with number of hidden layers equal to [1, 2, 3, 4] and all hidden layers have the same number of neurons. Report the number of neurons for MLPs with various depths. Compare the results to MLPs of various depths.

Depth: 1, Neurons per Hidden Layer: 100

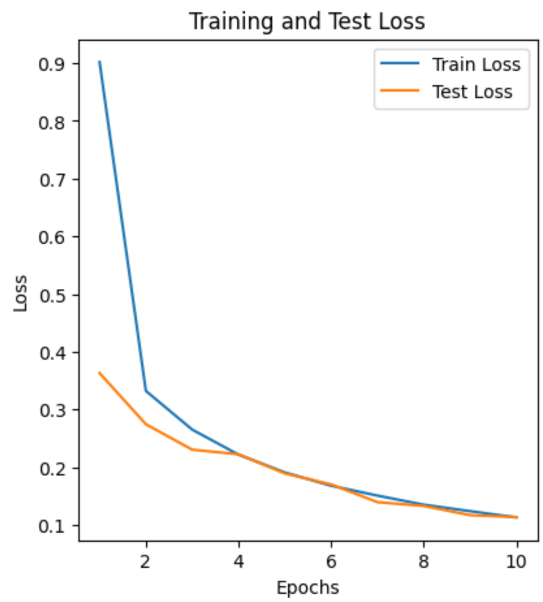


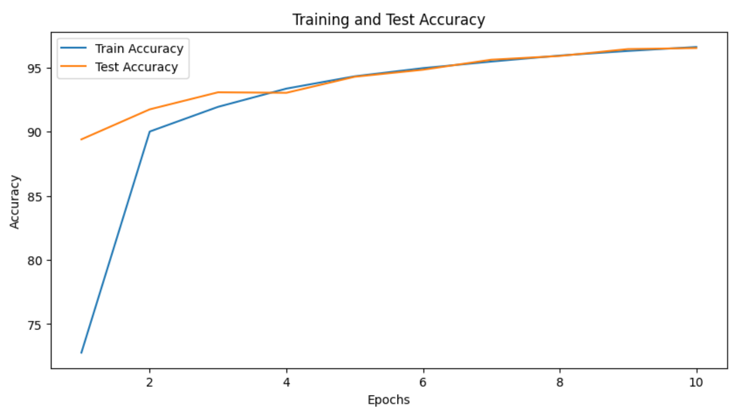




Epoch [10/10], Train Loss: 0.1517, Train Accuracy: 95.69%, Test Loss:

0.1487, Test Accuracy: 95.47%

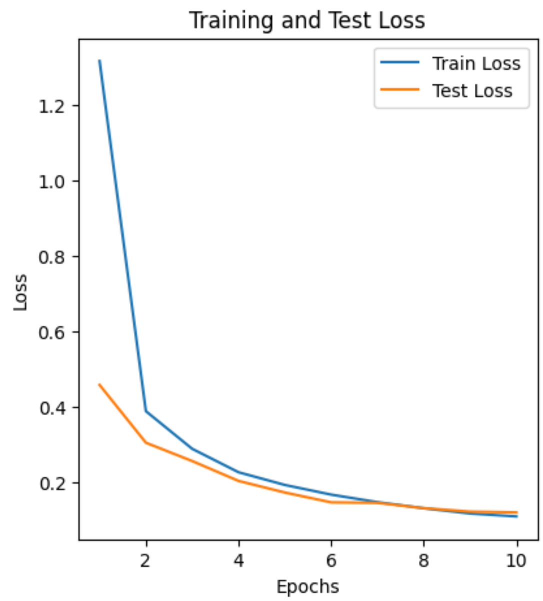
Depth: 2, Neurons per Hidden Layer: 89

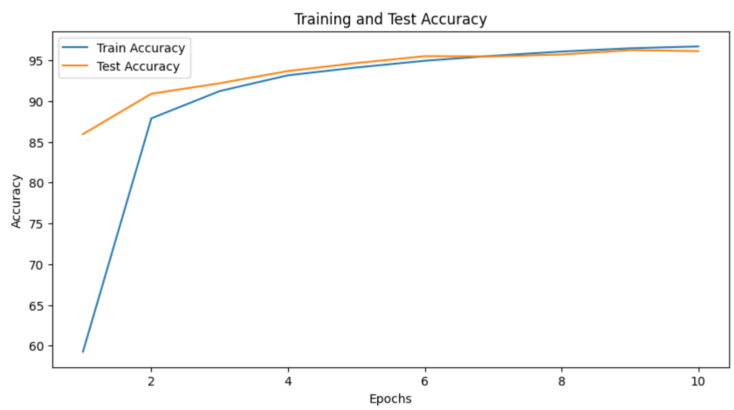



Epoch [10/10], Train Loss: 0.1130, Train Accuracy: 96.60%, Test Loss: 0.1135, Test Accuracy: 96.52%

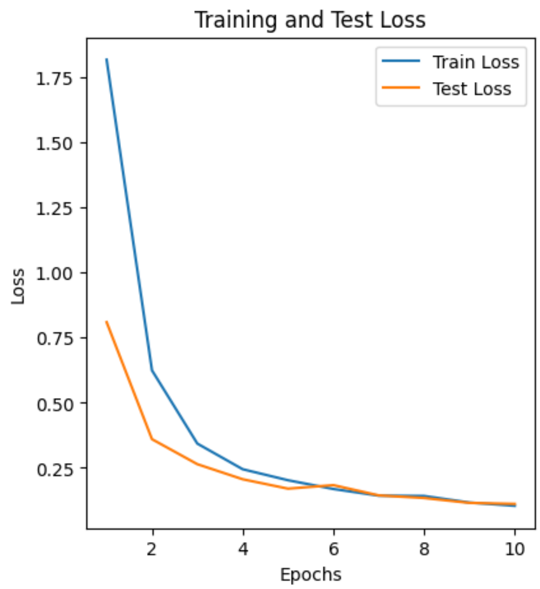
Depth: 3, Neurons per Hidden Layer: 82

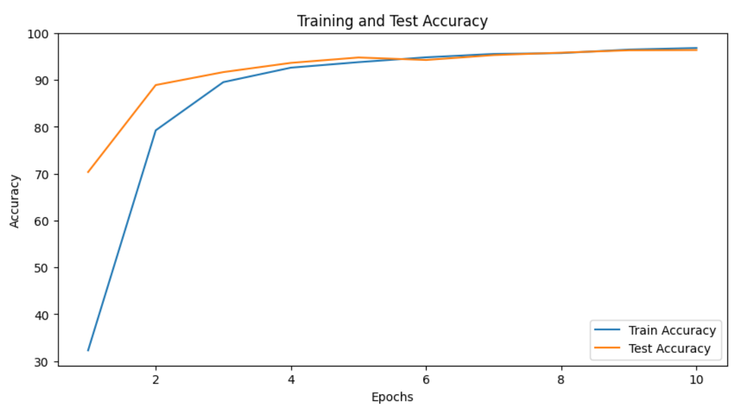



Epoch [10/10], Train Loss: 0.1091, Train Accuracy: 96.71%, Test Loss: 0.1199, Test Accuracy: 96.13%

Depth: 4, Neurons per Hidden Layer: 77




Epoch [10/10], Train Loss: 0.1044, Train Accuracy: 96.81%, Test Loss: 0.1117, Test Accuracy: 96.38%

As we can see from the above:

Depth 1: Neurons per Hidden Layer: 100; Test Accuracy: 95.47% after 10 epochs

Depth 2: Neurons per Hidden Layer: 89; Test Accuracy: 96.52% after 10 epochs

Depth 3: Neurons per Hidden Layer: 82; Test Accuracy: 96.13% after 10 epochs

Depth 4: Neurons per Hidden Layer: 77; Test Accuracy: 96.38% after 10 epochs

It seems that as the depth increases while keeping the total number of parameters constant, deeper networks don't consistently lead to higher accuracies. Moreover, as the depth increases, the number of neurons per hidden layer reduces. However, this reduction shallower networks and it doesn't necessarily translate to a linear increase in accuracy.

4. For a 4 hidden layer MLP, assume the total number of hidden neurons to be 100. Tune the number of hidden neurons on each layer. Report the best structure you find and the corresponding train/test accuracies.

Configuration Tried: [25, 25, 25, 25], [10, 30, 40, 20], [40, 30, 20, 10], [40, 30, 15, 15], [40, 20, 20, 20], [15, 15, 30, 40], [20, 30, 40, 10], [5, 20, 50, 25], [35, 40, 20, 5], [30, 20, 25, 25]

Best Configuration: [40, 20, 20, 20]

Train Accuracy: 98.18%

Test Accuracy: 96.51%

# 3 Intermediate Layer Representation

:

1. For the network with 3 hidden layers and 33 neurons on each layer, extract the intermediate layer outputs as feature representations of the data, and train logistic regression models using those features. Also, train a logistic regression using the original 784 features. Report the train and test accuracies and explain/compare the results.

Logistic Regression on Features from Intermediate Layer 1 - Train Accuracy: 96.06% - Test Accuracy: 95.55%

Logistic Regression on Features from Intermediate Layer 2 - Train Accuracy: 96.66% - Test Accuracy: 95.83%

Logistic Regression on Features from Intermediate Layer 3 - Train Accuracy: 96.68% - Test Accuracy: 95.96%

Logistic Regression on Original Features - Train Accuracy: 94.16% - Test Accuracy: 92.03%

As we move deeper into the network (from Layer 1 to Layer 3), the extracted features become increasingly effective for classification, reflected in the marginal but consistent improvements in accuracy. In other words, utilizing features from intermediate layers as representations seems to offer better performance compared to the original features. This suggests that the representations learned in the intermediate layers capture more relevant information for classification.

# 4 Convolutional NN (LeNet5) on MNIST

:

1. Implement a convolutional network called LeNet-5. Report the techniques tried, the training and test loss curve over epochs, and the training and test accuracy over epochs.

Batch normalization applied; Dropout applied;

Epoch: 100;

Learning rate: 0.001;

Train Loss: 0.0818, Train Accuracy: 97.99%.

Test Loss: 0.0426, Test Accuracy: 98.75%.