Programming Assignment 2
CAP 5415
Fazle Rahat
UCF ID: 5499542

Observation for model 1:

The model structure is: ConvNet(
 (fc1): Linear(in_features=784, out_features=100, bias=True)
 (fc2): Linear(in_features=100, out_features=10, bias=True)
)

Hyperparameters:
 Batch size: 10
 Number of epochs: 60
 Learning rate: 0.1

Best testing accuracy is 97.81%

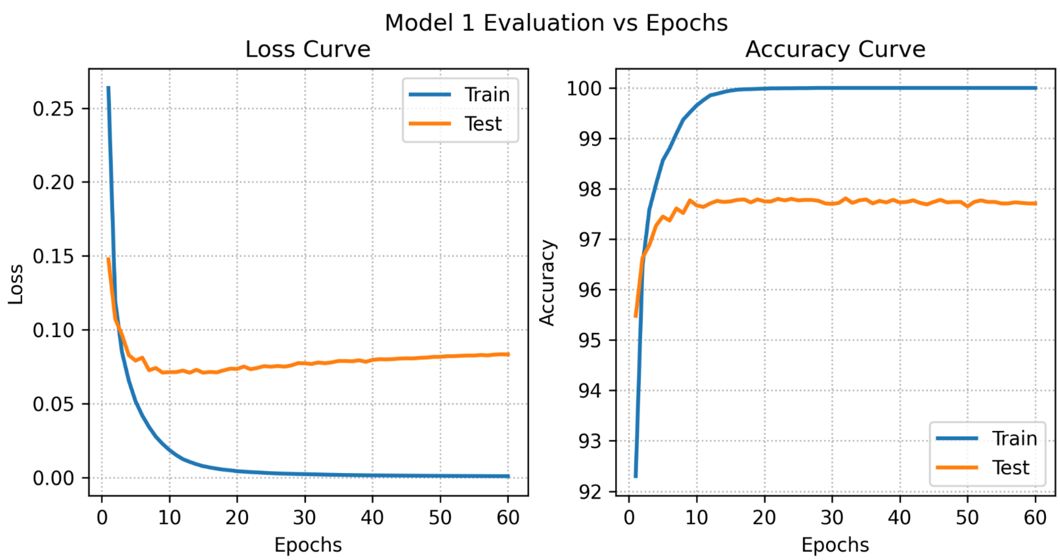

Figure 1: Loss and accuracy curve with respect to corresponding epochs for model 1

From Model 1, for the loss curve, both training and test losses exhibit a downward trend from the beginning. The training loss gradually decreases, but after 9 or 10 epochs, the test loss starts to increase. This likely indicates the overfit nature of the model.

On the other hand, for the accuracy curve, the training accuracy shows an upward trend and eventually reaches its highest point. However, the test accuracy, while also showing an upward trend, consistently performs at a lower level than the training accuracy, fluctuating between the 97% and 98% range.

Observation for model 2:

The model structure is: ConvNet(
  (conv1): Conv2d(1, 40, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(40, 40, kernel_size=(5, 5), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=640, out_features=100, bias=True)
  (fc2): Linear(in_features=100, out_features=10, bias=True)
)

Hyperparameters:
 Batch size: 10
 Number of epochs: 60
 Learning rate: 0.1
Best testing accuracy is 99.22%

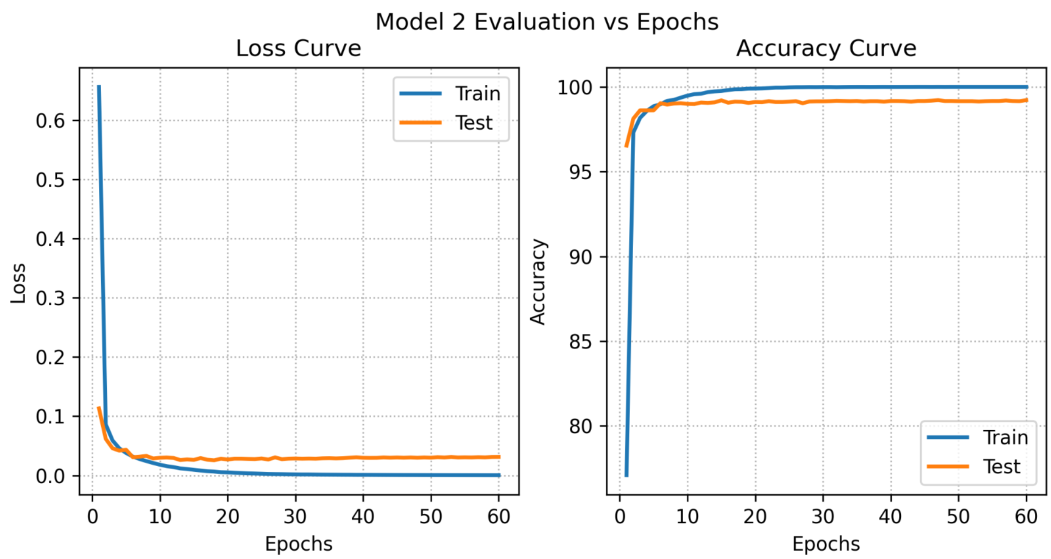

Figure 2: Loss and accuracy curve with respect to corresponding epochs for model 2

In Model 2, we introduced two convolution layers along with max-pooling layers and achieved better test loss performance than in Model 1. In this model, both training and test losses exhibit downward trends, but test losses consistently remain higher than training losses after 8 epochs.

On the other hand, the training and test accuracy curves both display an upward trend from the beginning. The training curve reaches its peak performance after 16 epochs and maintains that level consistently. However, the test performance fluctuates slightly, and most of the time, it lags behind the training performance.

Observation for model 3:

The model structure is: ConvNet(
  (conv1): Conv2d(1, 40, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(40, 40, kernel_size=(5, 5), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=640, out_features=100, bias=True)
  (fc2): Linear(in_features=100, out_features=10, bias=True)
)

Hyperparameters:
Batch size: 10
Number of epochs: 60
Learning rate: 0.03
Best testing accuracy is 99.38%

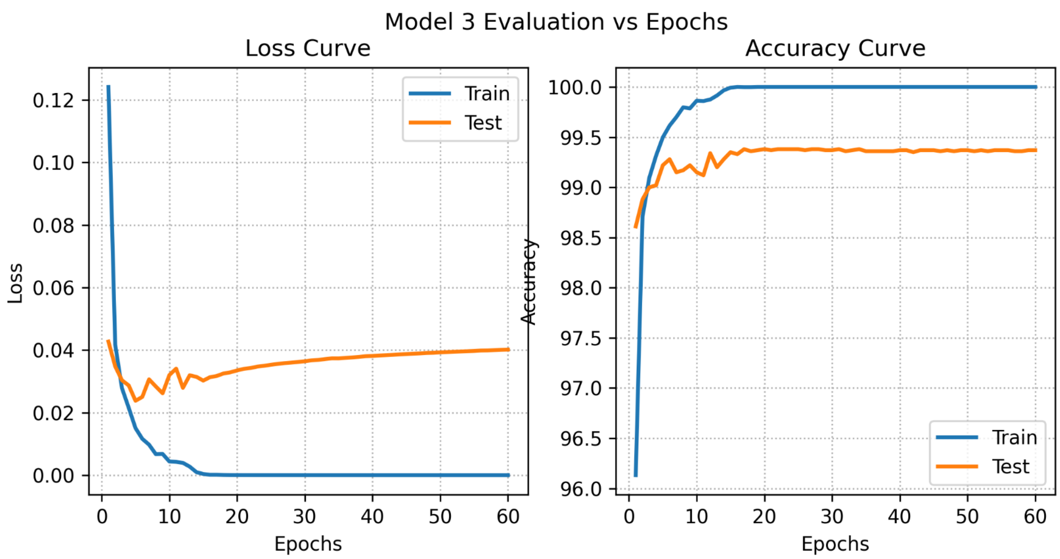

Figure 3: Loss and accuracy curve with respect to corresponding epochs for model 3

In Model 3, we changed the activation function from sigmoid to ReLU. Figure 3 shows that the test loss curves exhibit some fluctuations during the initial epochs compared to the previous two models. However, the test loss curve gradually increases starting from epoch 15.

Additionally, we modified the learning rate from 0.1 to 0.03 in Model 3. This change resulted in higher test accuracy compared to the other two models.

Observation for model 4:

The model structure is: ConvNet(
  (conv1): Conv2d(1, 40, kernel_size=(5, 5), stride=(1, 1))
  (conv2): Conv2d(40, 40, kernel_size=(5, 5), stride=(1, 1))
  (pool): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilation=1, ceil_mode=False)
  (fc1): Linear(in_features=640, out_features=100, bias=True)
  (fc2): Linear(in_features=100, out_features=100, bias=True)
  (fc3): Linear(in_features=100, out_features=10, bias=True)
)

Hyperparameters:
 Batch size: 10
 Number of epochs: 60
 Learning rate: 0.03
Best testing accuracy is 99.43%

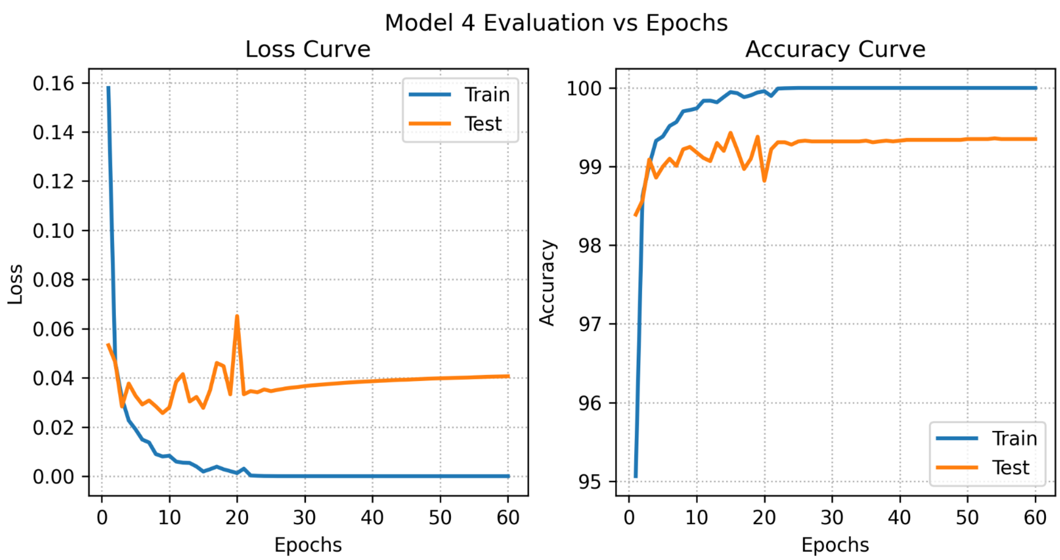

Figure 4: Loss and accuracy curve with respect to corresponding epochs for model 4

This model exhibits a similar trend to Model 3 in terms of both loss and accuracy, with a greater fluctuation of test loss and test accuracy. However, the addition of another fully connected layer has further improved test accuracy compared to Model 3.

Observation for model 5:

The model structure is: ConvNet(
 (conv1): Conv2d(1, 40, kernel_size=(5, 5), stride=(1, 1))
 (conv2): Conv2d(40, 40, kernel_size=(5, 5), stride=(1, 1))
 (pool): MaxPool2d(kernel_size=(2, 2), stride=(2, 2), padding=0, dilation=1, ceil_mode=False)
 (fc1): Linear(in_features=640, out_features=1000, bias=True)
 (fc2): Linear(in_features=1000, out_features=1000, bias=True)
 (fc3): Linear(in_features=1000, out_features=10, bias=True)
 (dropout): Dropout(p=0.5, inplace=False)
)

Hyperparameters:
Batch size: 10
Number of epochs: 40
Learning rate: 0.03
Best testing accuracy is 99.54%

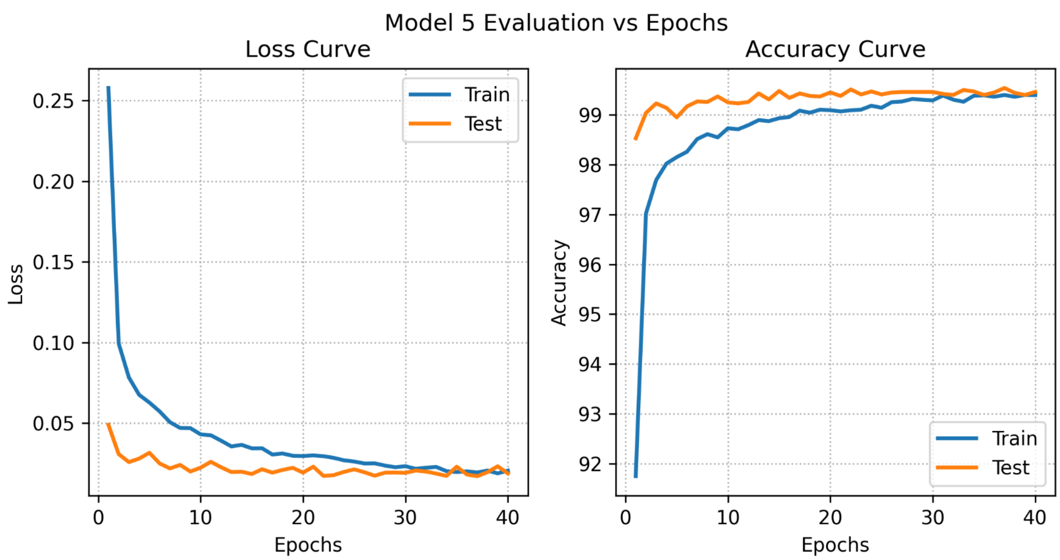

Figure 5: Loss and accuracy curve with respect to corresponding epochs for model 5

In Model 5, we introduced a dropout layer with a probability of 0.5. In this model, we achieved an overall lower test loss than the training loss and higher overall accuracy than the training set. This result indicates a reduction in overfitting. Notably, Model 5 delivers the best test accuracy, reaching 99.54%. Therefore, among all the models, Model 5 provides the best results based on both testing and training loss and accuracy.