COMPUTER VISION (CAP5415)-PA-2 REPORT

CONVOLUTIONAL NEURAL NETWORK

Step 1:

I've created a fully connected hidden layer comprising of 100 neurons with Sigmoid activation function. In addition, trained it with Stochastic Gradient Descent (SGD) which is used for training with the following specifications such as learning rate=0.1 (total of 60 epochs) with mini-batch size=10 and no regularization.

Train Accuracy: 100%

Test Accuracy: 97.81

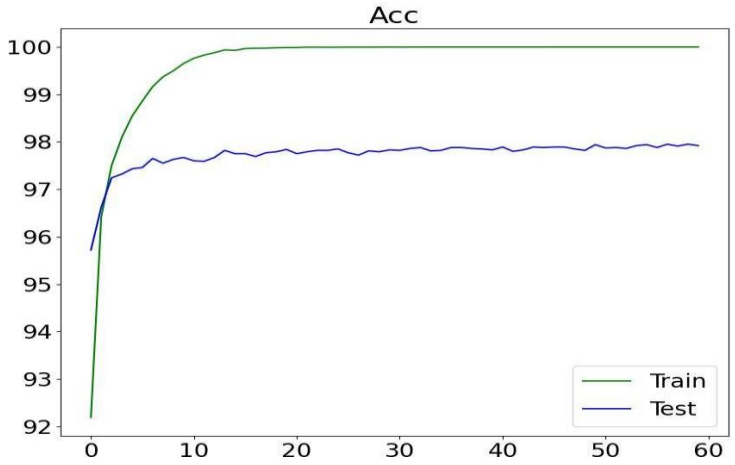

Fig a(1)

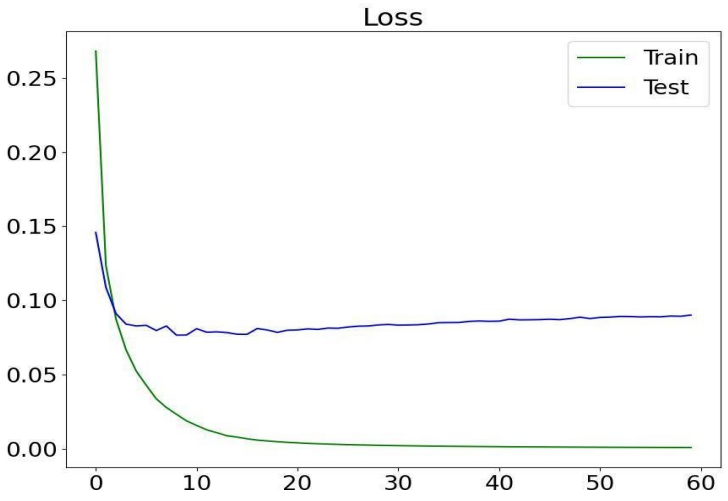

Fig b(1)

Observations:

From the above graphical observations we can depict that w.r.t epoch numbers there is rise in accuracies, where in the training accuracy reaches 100% approximately at an epoch of 30. While the testing accuracy lies around 97.81%. The training accuracy shows a sharp early rise followed by a gradual increase. On the other hand, testing loss begins a slow ascent and the loss curve indicates a point where accuracy plateaus, indicating possible overfitting. But by the 60th epoch, things are clearly more stable and in balance.

Step 2:

So, here we have inserted two convolutional layers which is built in previous step (Add the sigmoid activation and pooling layer to every convolutional layer) with following specifications:

Pool Over: 2x2 regions

kernels: 40

stride :1

kernel size :5x5

Train Accuracy: 100%

Test Accuracy: 99.18%

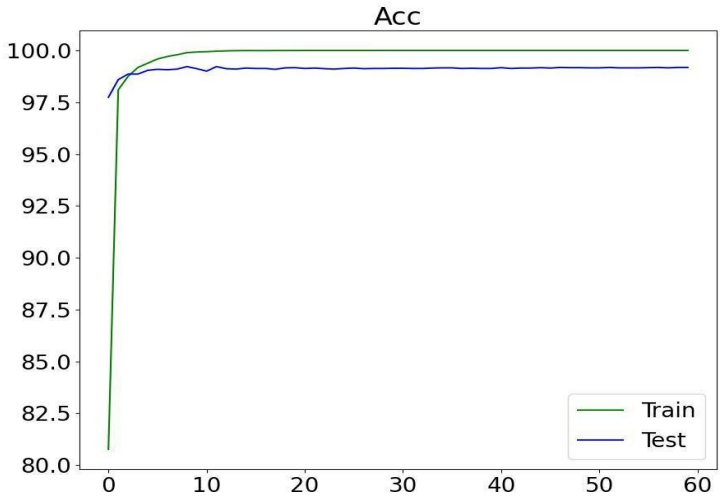

Fig a(2)

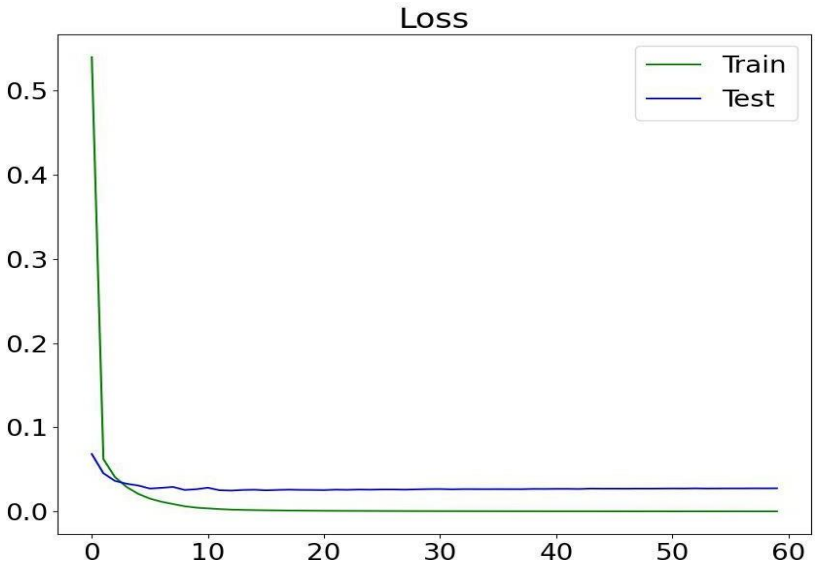

Fig b(2)

Observations:

The model performed better after convolutional layers were added. The optimizer's progress stalled in the tenth epoch after hitting a minimum point, which is consistent with the accuracy plot's behavior.

Step 3:

I've replaced the Sigmoid activation function with the Rectified Linear Unit function (ReLU) and inculcated with new learning rate as 0.03. Simultaneously, retrained the system with these settings.

Train Accuracy: 100%

Test Accuracy: 99.32%

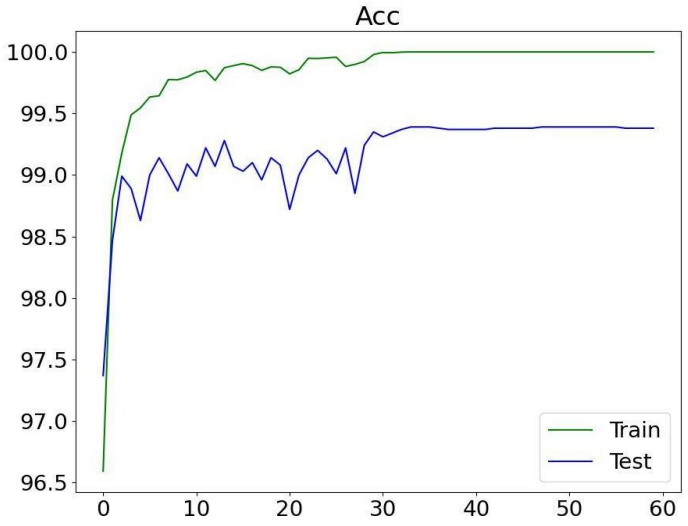

Fig a(3)

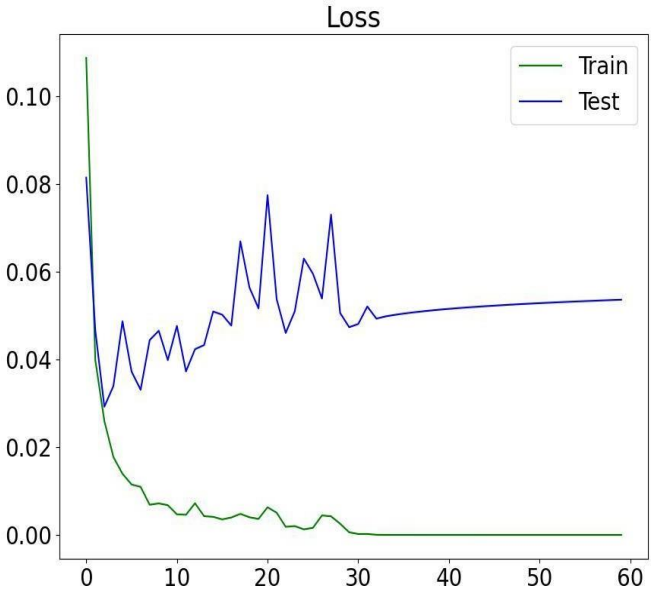

Fig b(3)

Observations:

As the model undergoes a transition from the Sigmoid activation function to Rectified Linear Unit (ReLU), training and testing accuracies reach close to 100%. Notably, around the 30th epoch, both accuracy and loss stabilized, with signs of overfitting becoming apparent in the testing loss graph thereafter.

Step 4:

For the network built in Step 3; I've added another fully connected layer with 100 Neurons.
Train Accuracy: 100%
Test Accuracy: 99.11%

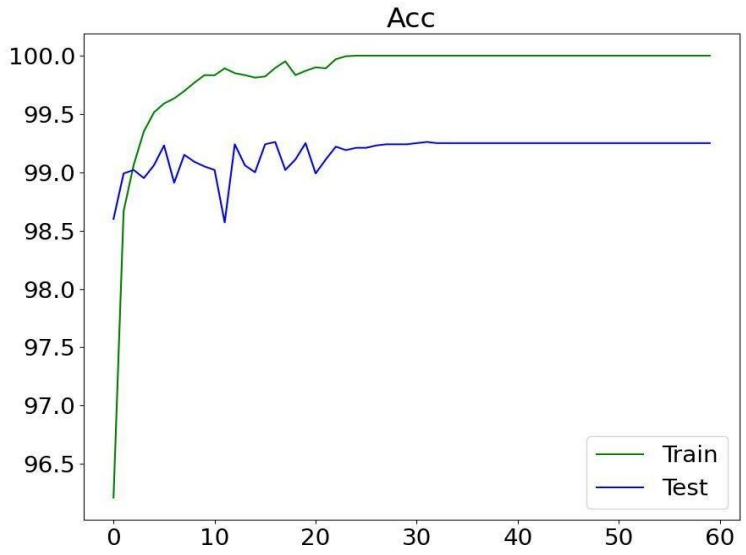

Fig a(4)

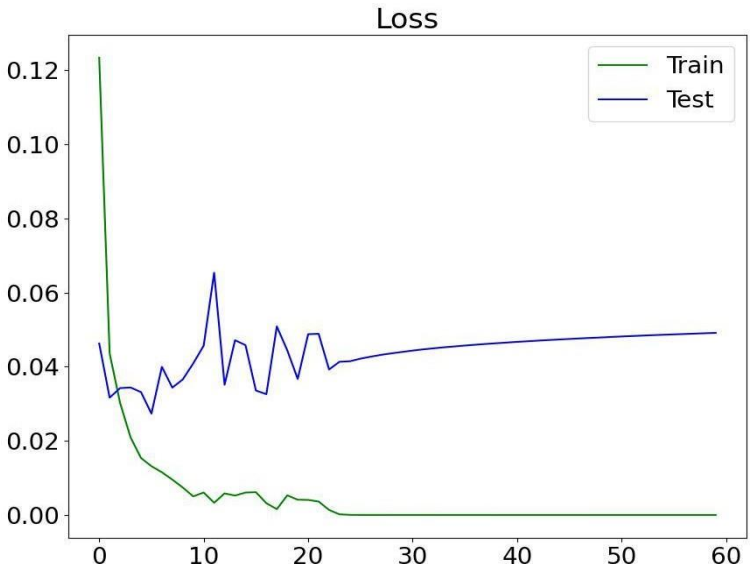

Fig b(4)

Observations:

The pattern is quite like Step 3, with the exception of a little overfitting event around the 20th epoch, which is probably caused by changes in the parameters.

Step 5:

Updating the number of neurons to 1000 and regularize it with following specifications:
Dropout Rate: 0.5
Epochs: 40
learning rate: 0.03
Train Accuracy: 99.91%
Test Accuracy: 99.30%

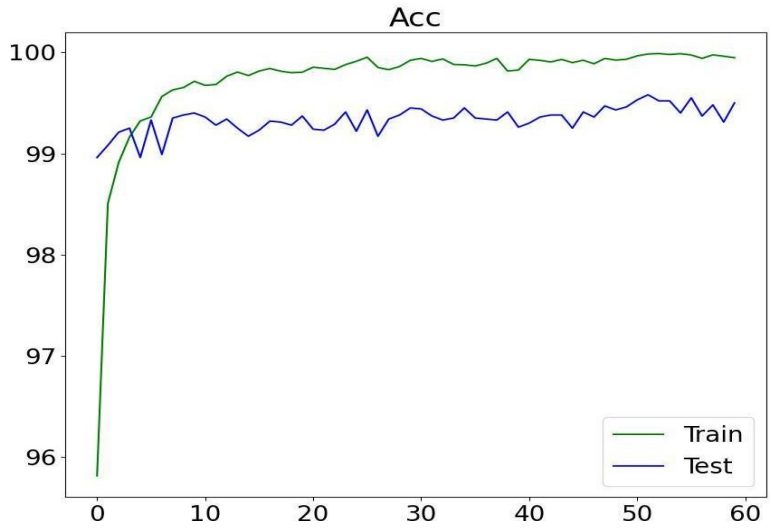

Fig a(5)

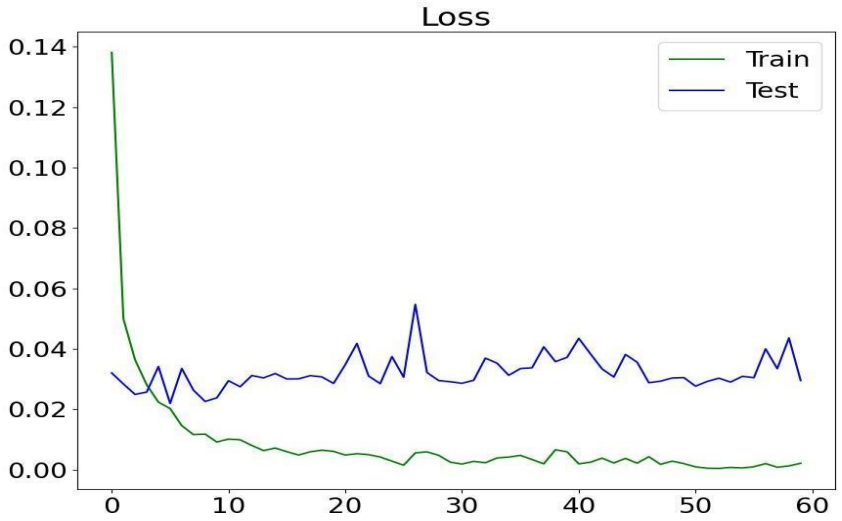

Fig b(5)

Observations:

The addition of dropout in this model serves to mitigate overfitting, although it introduces some noise in the loss curve. Nevertheless, the model performs effectively and attains higher accuracy when compared to the previous steps.