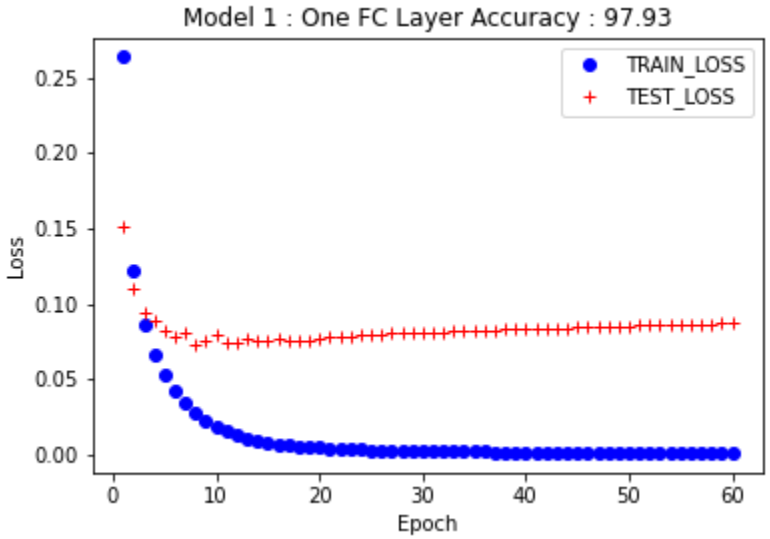
REPORT FOR QUESTION 2 PROGRAMMING ASSIGNMENT 1 CAP 5415

NAME : SARATH KUMAR MANNAM
KNIGHTS :
sarath.mannam13@knights.ucf.edu

Entire Code : Code Folder
https://drive.google.com/drive/folders/1r_PWlpY-OidZsNtxmWZ69q8_XaXsblDS?usp=sharing

TASK 1:
OUTPUT FOR MODEL 1 :

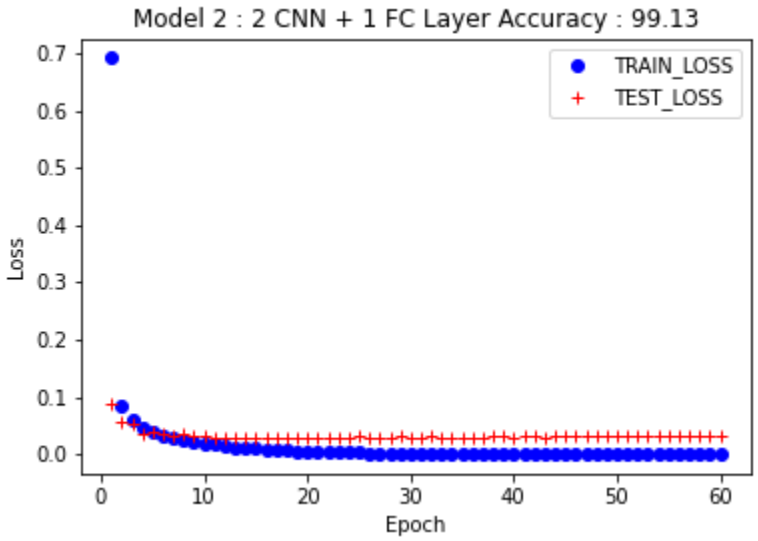


Output for Model 1 : Model1.txt
Learning Rate = 0.1
Loss = Categorical Cross Entropy
Initially the model had a high loss, but after training for a few epochs the train loss reduced exponentially, and the loss remained stable after 20 epochs. Whereas, test loss started to increase by small amounts after 20 epochs. The model had converged after 20 epochs. After 20 epochs, the model was trying to overfit, because test error was slightly going up and train error was going down.

Model Accuracy : 97.93

TASK 2:

OUTPUT FOR MODEL 2 :



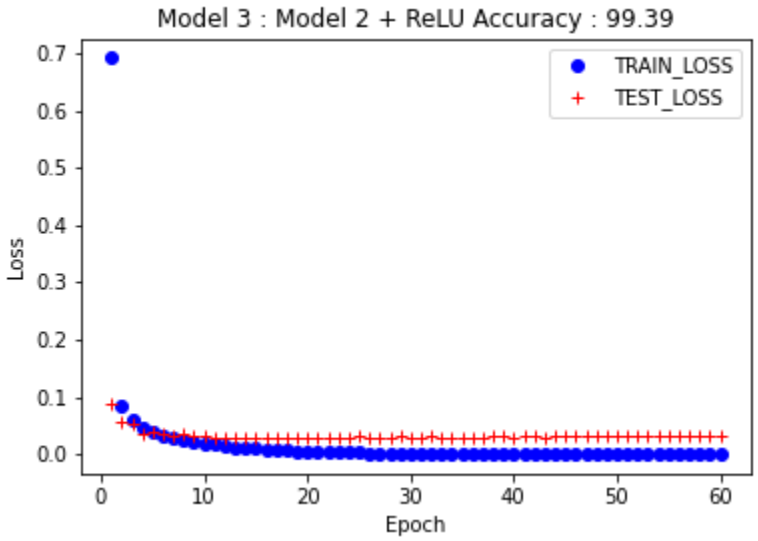
Model converged after 25 epochs. Initially training and test errors were high, but the errors reduced considerably after a few epochs. Train and test error remained stable after 25 epochs, which leads us to the conclusion that the network converged after 25 epochs.
Loss was Less as compared to Model 1, therefore using CNN layers gave good results.

Output for Model 2: Model2.txt

Model Accuracy : 99.13

TASK 3:

OUTPUT FOR MODEL 3:



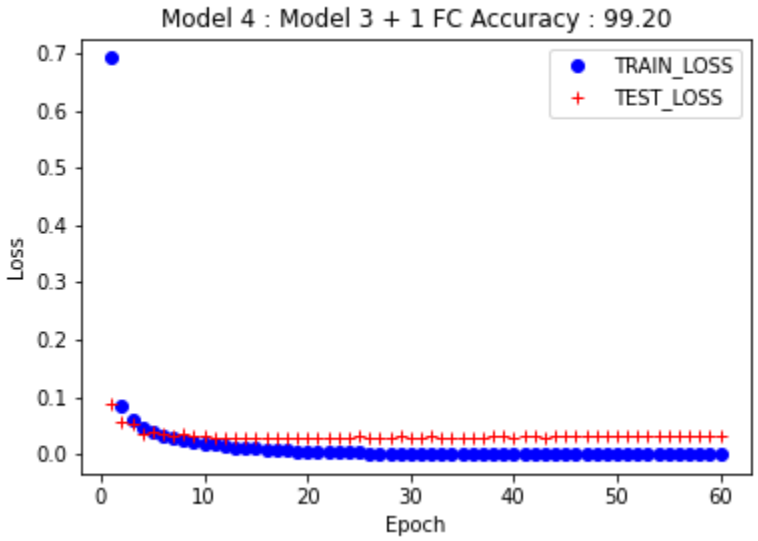
Output for Model 3: [model3.txt]

Model converged roughly after 25 epochs similar to TASK 2. The train and test loss graph was equivalent to TASK 2. After using the ReLU activation unit, there was a slight jump in accuracy.

Accuracy : 99.39

TASK 4:

OUTPUT FOR MODEL 4:



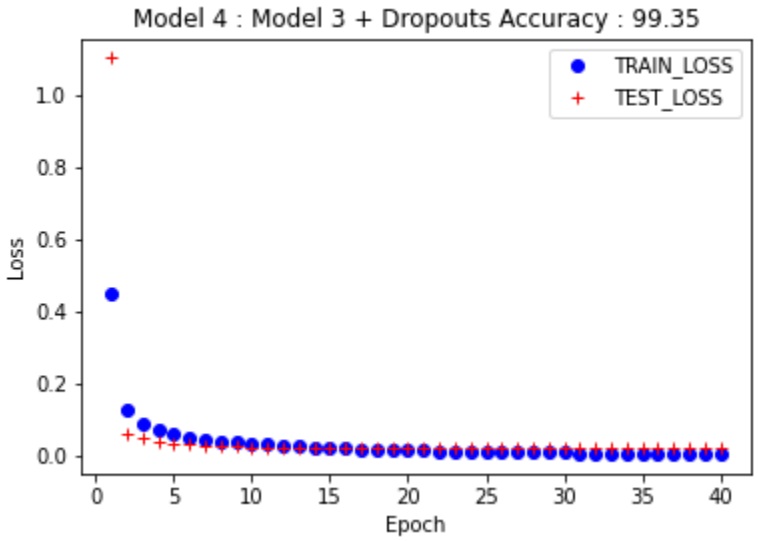
Output for Model 4 : [model4.txt]

Usage of 1 more FC layer with 100 Neurons, performed slightly better over MODEL 2 and slightly lesser than MODEL 4. The graphs looked almost similar with epoch 25-30 the curves become stable.

Accuracy : 99.20

TASK 5:

OUTPUT FOR MODEL 5:



Output for Model 5 : [model5.txt]

Observations. Here we have trained the model for 40 epochs. The graph was very interesting. We have used dropouts of 0.5 as a regularization.

The train and test error were coinciding very clearly, The model converged quickly to less error value as compared to previous models.

Regularization helped avoid overfitting and we got very good accuracy.

Accuracy : 99.35