Assignment 1 Report

Faisal Omari – 325616894
Saji Assi – 314831207

1. Report your architecture, training time and performance on the test set. Do this for each network.

Answer:

Network 1 Architecture:

Layer	Input shape	Output shape	Parameters
FC (ReLU)	28*28	62	48670
FC	62	7	189

Network 2 Architecture:

Layer	Input shape	Output shape	Parameters
FC (ReLU)	28*28	62	48670
FC	62	7	441

Network 3 Architecture:

Layer	Input shape	Output shape	Parameters
FC (ReLU)	28*28	60	47100
FC (ReLU)	60	28	1708
FC (ReLU)	28	14	406
FC	14	7	105

Network 4 Architecture:

Layer	Kernel	Channels	Parameters
CNN (ReLU)	3	20	200
Dropout	0.25		
MaxPooling	$2x2$		
CNN (ReLU)	3	40	7240
Dropout	0.25		
MaxPooling	$2x2$		
CNN (ReLU)	3	90	32490
MaxPooling	$2x2$		
FC	OUTPUT LAYER		5677

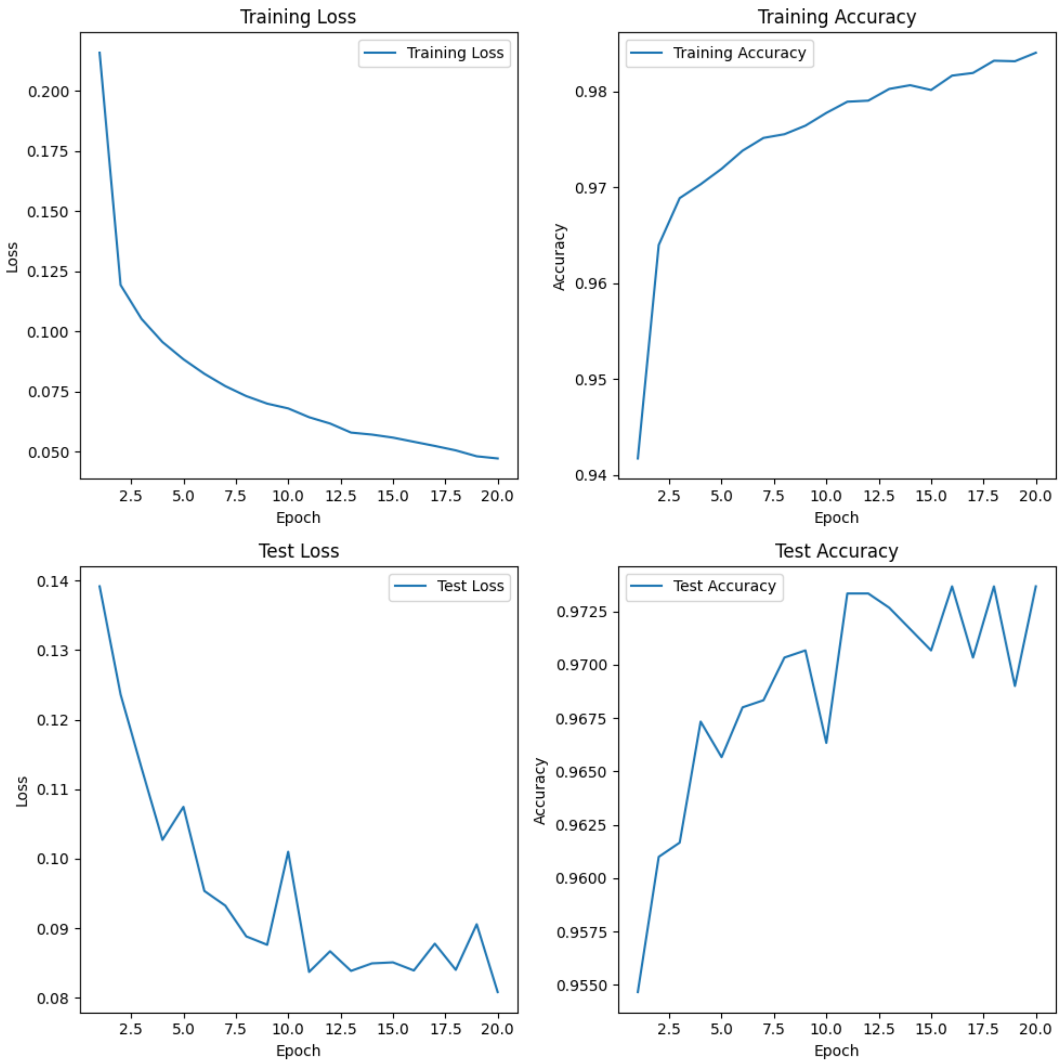
Networks Results (GPU - GTX1060Ti 6GB):

Network No.	Num of Epochs	Test Accuracy	Training Time (seconds)	Model Capacity
1	20	97.37%	5.4	48859
2	25	85.27%	15.5	49111
3	23	85.57%	31.3	49319
4	40	89.84%	352	45607

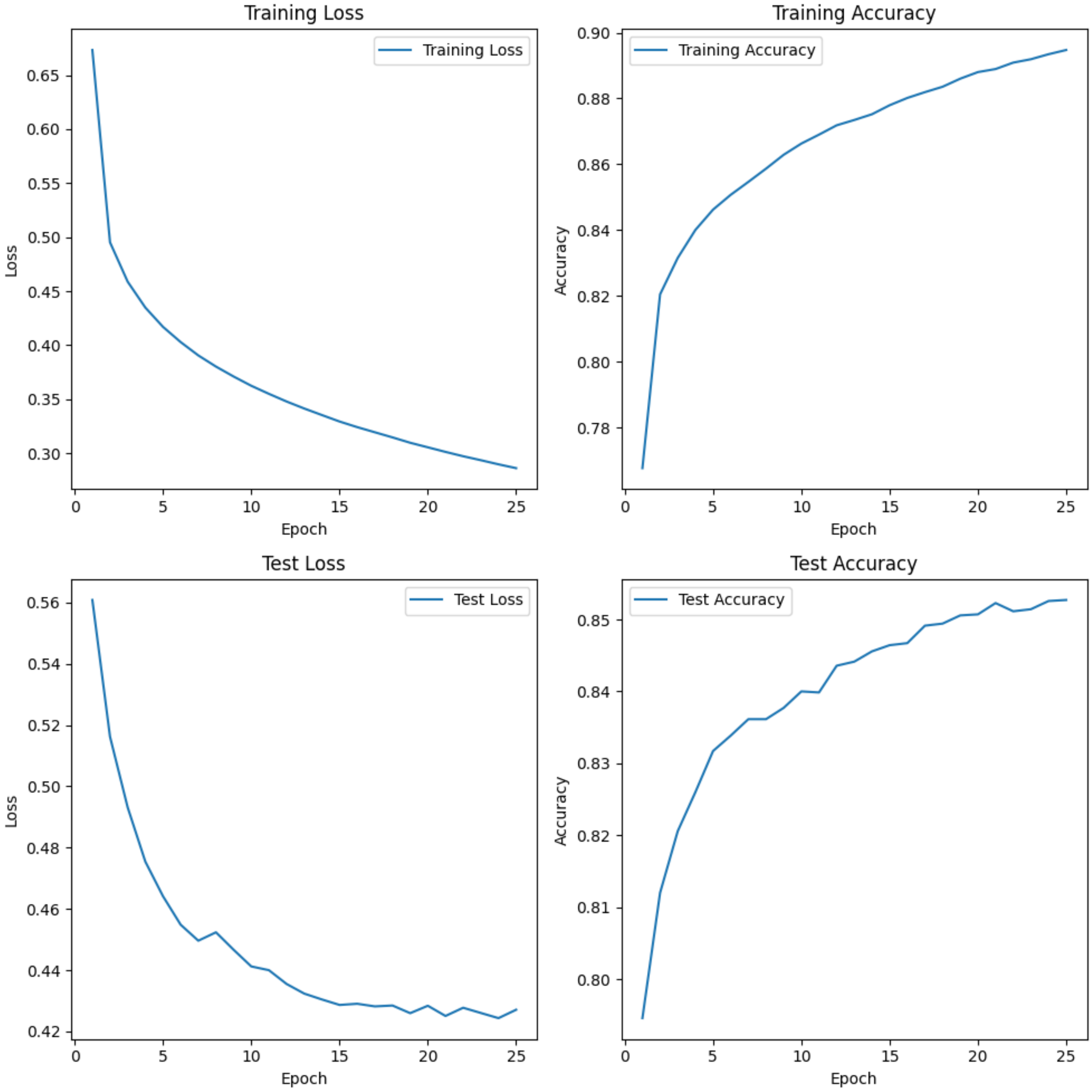
2. Add a plot showing train error, test error, and accuracy as a function of the number of epochs. Do this for each network.

Answer:

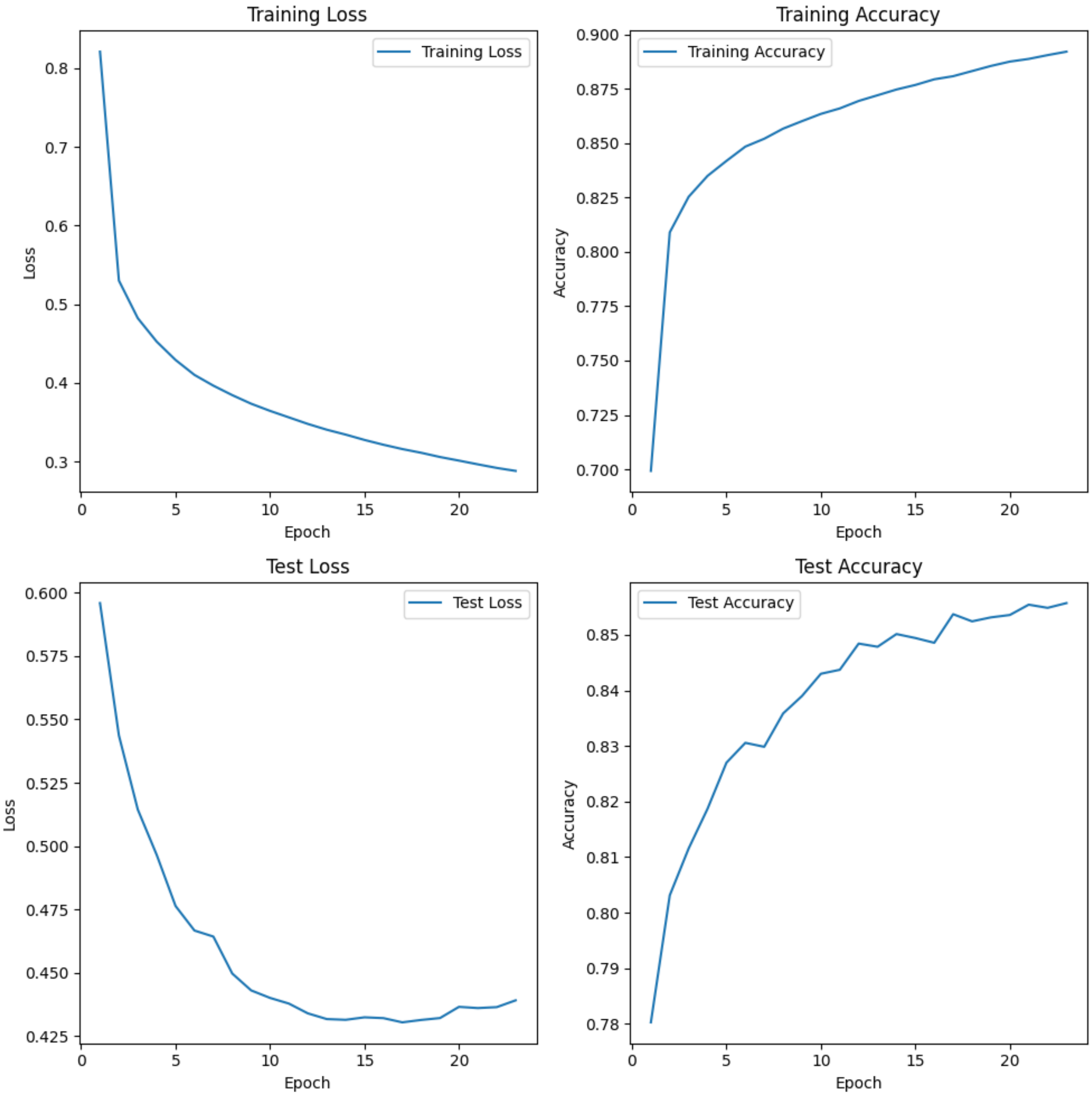
Model_1:



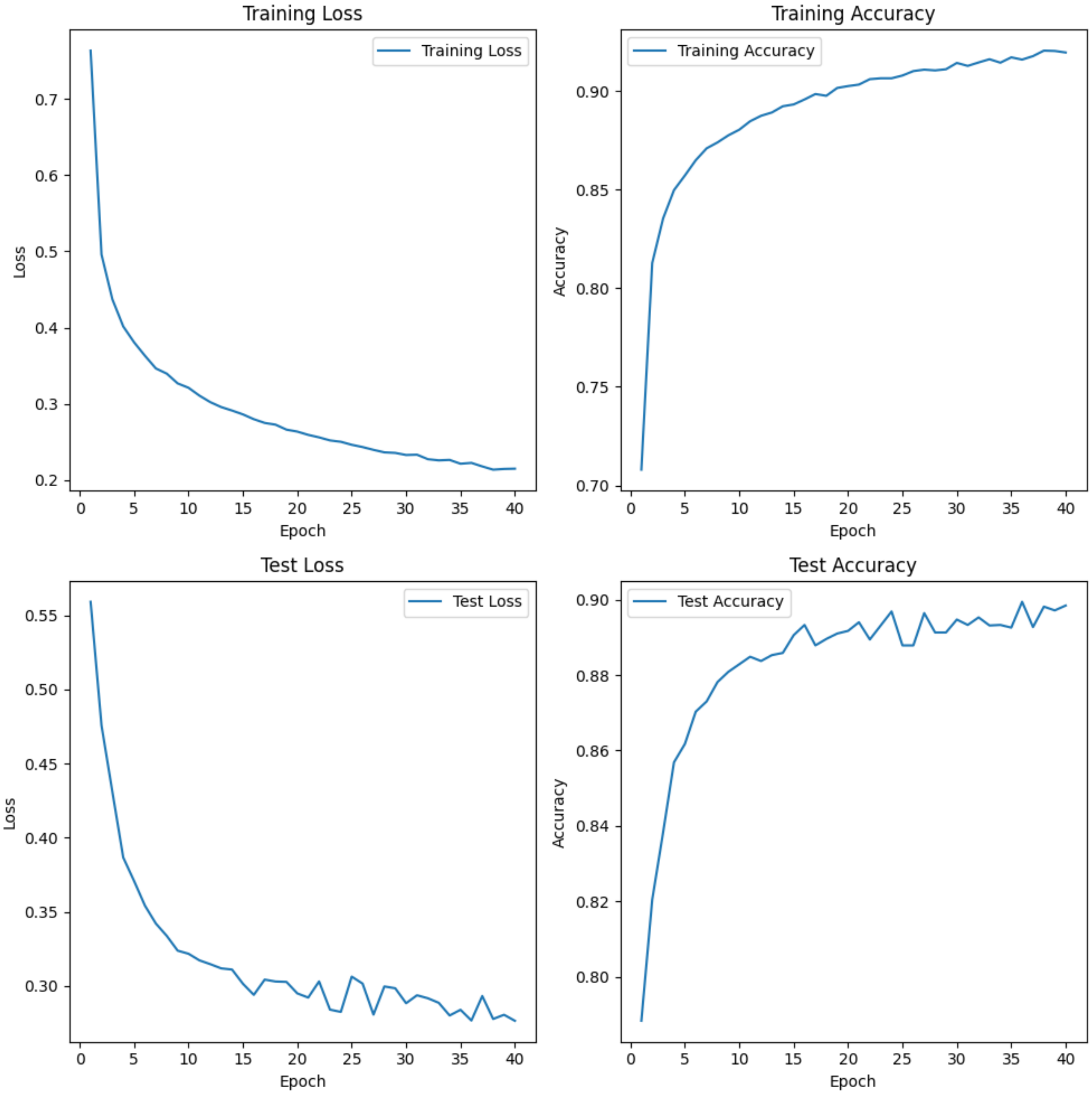
Model_2:


Training Loss
Training Loss
Loss
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
0
5
10
15
20
25
Epoch
Training Accuracy
Training Accuracy
Accuracy
0.90
0.88
0.86
0.84
0.82
0.80
0.78
0
5
10
15
20
25
Epoch
Test Loss
Test Loss
Loss
0.56
0.54
0.52
0.50
0.48
0.46
0.44
0.42
0
5
10
15
20
25
Epoch
Test Accuracy
Test Accuracy
Accuracy
0.85
0.84
0.83
0.82
0.81
0.80
0
5
10
15
20
25
Epoch

Model_3:


Training Loss
Training Loss
Loss
0.8
0.7
0.6
0.5
0.4
0.3
Epoch
0
5
10
15
20
Training Accuracy
Training Accuracy
Accuracy
0.900
0.875
0.850
0.825
0.800
0.775
0.750
0.725
0.700
Epoch
0
5
10
15
20
Test Loss
Test Loss
Loss
0.600
0.575
0.550
0.525
0.500
0.475
0.450
0.425
Epoch
0
5
10
15
20
Test Accuracy
Test Accuracy
Accuracy
0.85
0.84
0.83
0.82
0.81
0.80
0.79
0.78
Epoch
0
5
10
15
20

Model_4:


Training Loss
Training Loss
Loss
0.7
0.6
0.5
0.4
0.3
0.2
Epoch
0
5
10
15
20
25
30
35
40
Training Accuracy
Training Accuracy
Accuracy
0.90
0.85
0.80
0.75
0.70
Epoch
Test Loss
Test Loss
Loss
0.55
0.50
0.45
0.40
0.35
0.30
Epoch
Test Accuracy
Test Accuracy
Accuracy
0.90
0.88
0.86
0.84
0.82
0.80
Epoch

3. Explain why the performance of Network 1 is better than that of Network 2.

Answer:

The first network and the second network has almost the same architecture, the same number of fully-connected layers, but still we see according to the results that the first network has higher accuracy on the test set, and that is not weird, because the main difference between these two models, is that the first one is working on classifying only 3 classes, which makes the job easier, because it needs to learn only the features and the classification of 3 digits {0,1,2}, and for this job it has 50k parameters to use, which make it about 17k parameters per class, while the second model has to classify 7 classes, which are the digits {0,1,2,3,4,5,6}, and we see that here the job is harder, because there is more features and patterns it has to learn to differentiate between the classes with high accuracy such as the first model, while the number of parameters of the network which is limited by 50k can't help on this mission because it can't catch deep features to help classifying the 7 classes, and we see that according to the number of classes, each class gets about 7k parameters, that's right that this may not mean anything, but if we think about making the model architecture wider and make it with more parameters it can really get to an accuracy that is closer to the accuracy of the first model.

4. Explain why the performance of Network 3 is better than the performance of Network 2.

Answer:

The second and third network share the same parameters limit of 50k, and the same train and test datasets, with the same type of layers (Fully-Connected), while the main difference is that the third layer is constructed of 4 layers, while the second one is of 2 layers, but these 2 layers really helps the network to catch deeper information about the classifying process, in addition to learning more complex features that can help in classifying, and the non-linearity on the 3rd network can also help in catching more details about each class which helps the network to classify in a better way with risk of overfitting which must be avoided when training the model more carefully than the 2nd network.

5. Explain why the performance of Network 4 is better than the performance of Network 3. 6. Conclude, by questions 3,4,5 above, which architecture is best suited for the FMNIST classification problem.

Answer:

According to what we've learnt on the class, the CNN was discovered mainly to help in classifying images and to extract and catch more features from the image, while the fully-connected networks doesn't have the feature of catching spatial relationships in the image, also that CNN architecture that we've chosen can exploit parameters limited number with higher efficiency which leads to higher accuracy with the same number of layers and parameters comparing to another FC network, and the features that the CNN can catch is more generalized over the dataset which leads the model to have better accuracy for data strange data that haven't been trained on, finally, the CNN has more tuning options on the CNN layer itself on PyTorch which leads to make the tuning process easier with higher performance and accuracy comparing to the FC, and according to the results we've got that is represented in the following table, we can easily see how the CNN overcome FC network, so eventually we can see that CNN architecture is best suited for the FMNIST classification problem with the assignment limitations comparing to the other networks!

$* \ Note \ *$: the submitted python notebook file (ipynb) do work with Google Collab, while the code was tested and trained on the local machine as mentioned above with CPU: $i5 - 9400f$ and GPU: $GTX1060ti \ 6GB$, the submitted code does the test when running it, to change it to train you have to change the MODE to “$TRAIN$” instead of “$TEST$”, and to train only one of the models $(1 - 4)$ change the MODE to “$TRAIN1$” or “$TRAIN2$” … accordingly.