

Aashish Adhikari

CS 535 Homework 3, Winter 2019

NOTE: Even though the assignment asks only the figures and no explanation, I have included some extra figures and description. Thanks.

Question 1.

Answer: A batch-normalization layer was added after the first fully-connected layer as it can be seen in the program file named “CIFAR_10_1_b.py”. The parameters were saved in a file named “mytraining1b.pth”. The model was converted into training and evaluation mode as implemented in the program.

The table below illustrates the findings **before and after** using Batch Normalization.

	Num of Epochs	Learning Rate	Final Training Loss	Final Training Accuracy In %	Final Testing Loss	Final Testing Accuracy In %	Maximum Testing Accuracy, Acc-Max	Epoch Number for Acc-Max
Without Batch Norm	20	0.01	0.06973	97.924	2.06767	72.880	74.910	4
With Batch Norm	20	0.01	0.00140	99.992	0.98144	79.970	79.970	20

Following are the plots for

a) Training and Testing Accuracies vs. Number of Epochs for the model without Batch Normalization.

b) Training and Testing Losses vs. Number of Epochs for the model without Batch Normalization

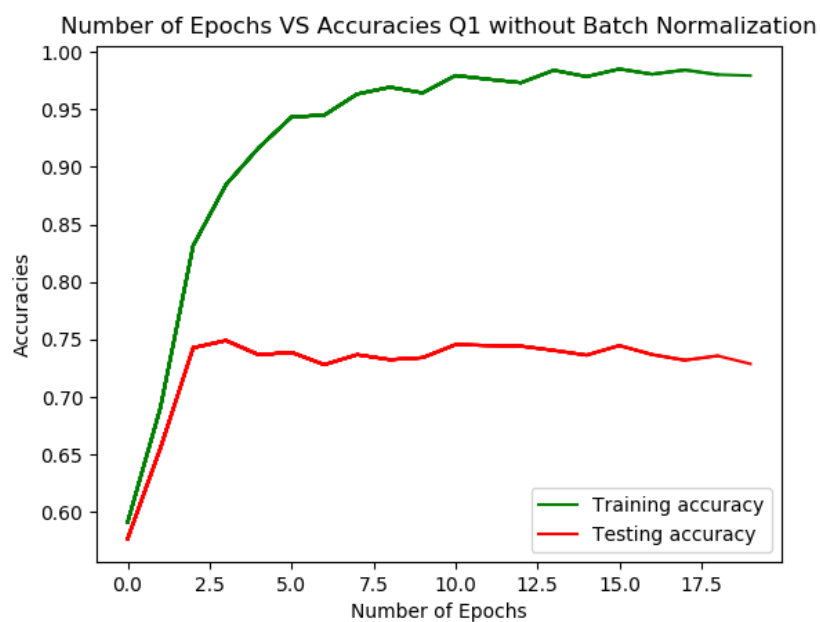


Figure 1.a.Accuracy vs Number of Epochs

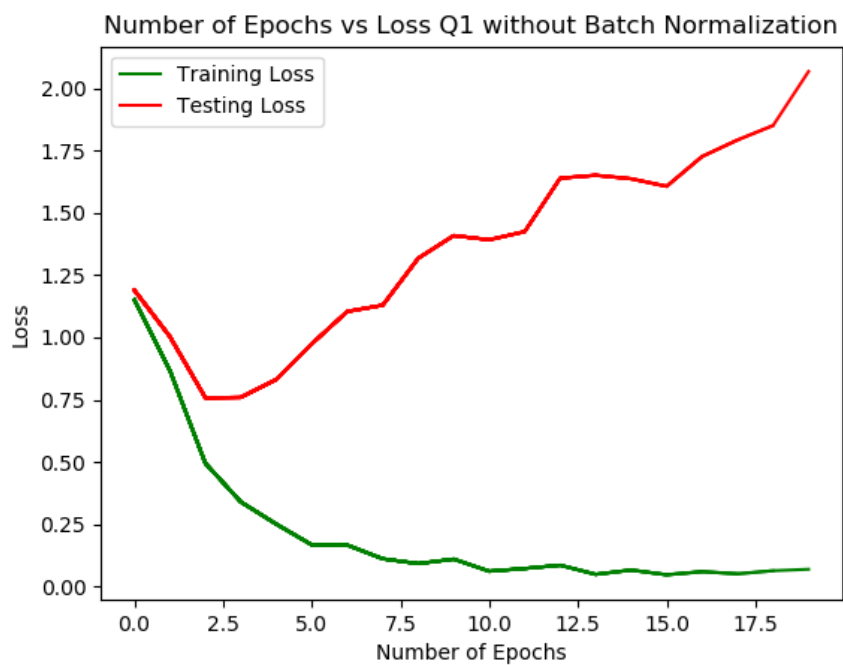


Figure 1.b. Loss vs Number of Epochs

Following are the plots for

a) Training and Testing Accuracies vs. Number of Epochs for the model with Batch Normalization.

b) Training and Testing Losses vs. Number of Epochs for the model with Batch Normalization

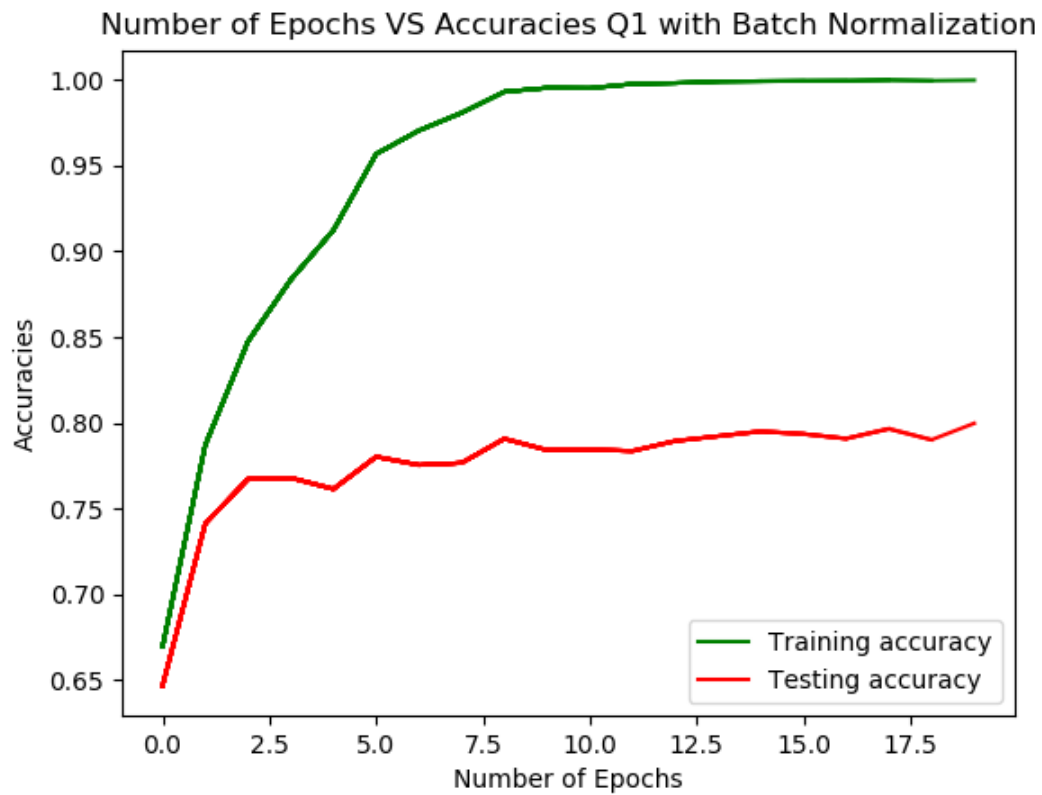


Figure 1.c. Accuracy vs Number of Epochs

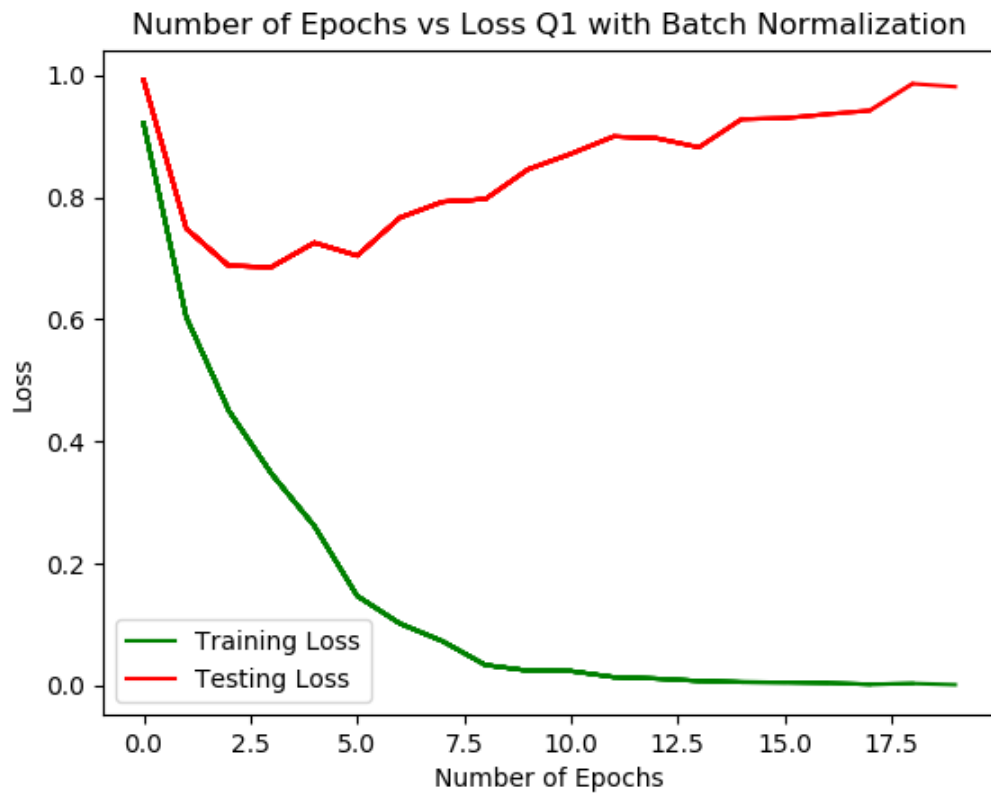


Figure 1.d. Loss vs Number of Epochs

For both the cases, we could see a decrease in the testing loss at first which started increasing after some epoch regardless of the decrease in testing accuracy.

Question 2

Answer: A new fully-connected layer with 512 nodes was added as mentioned in the question. The parameters saved previously in the file “mytraining1b.pth” were applied only up to the layer before this newly created layer.

Following table summarizes the findings.

	Num of Epochs	Learning Rate	Final Training Loss	Final Training Accuracy in %	Final Testing Loss	Final Testing Accuracy in %	Maximum Testing Accuracy, Acc-Max	Epoch Number for Acc-Max
At the end of 20 th epoch	20	0.01	0.00005	100	1.44908	79.850	80.120	8
Right after the pretrained weight was applied for the first time	20	0.01	0.00402	99.998	0.81736	79.850	-	-

My understanding : As shown in the table above and also illustrated in the plots below, the pretrained weights yielded a very good training accuracy of 99 percent regardless of an extra fully-connected layer. This first seemed quite odd to me given that the newly added fully-connected layer is randomly initialized and it should have prevented from achieving such a good accuracy in the first run, but many classmates I talked to also received similar results. I tried looking for the reason behind it on the internet. A post explained that since we are using many convolutional layers to achieve these weights, the weights have reached a saturation point such that adding a single fully-connected layer behind the convolutional layers does not affect much in the accuracy even if it has been randomly initialized. Hence, the training accuracy curve is almost a straight line with slight variations in the testing accuracy. Similar is the case for training loss.

Following are the plots for

a) Training and Testing Accuracies vs. Number of Epochs for the model with partial pre-trained weights.

b) Training and Testing Losses vs. Number of Epochs for the model with partial pre-trained weights

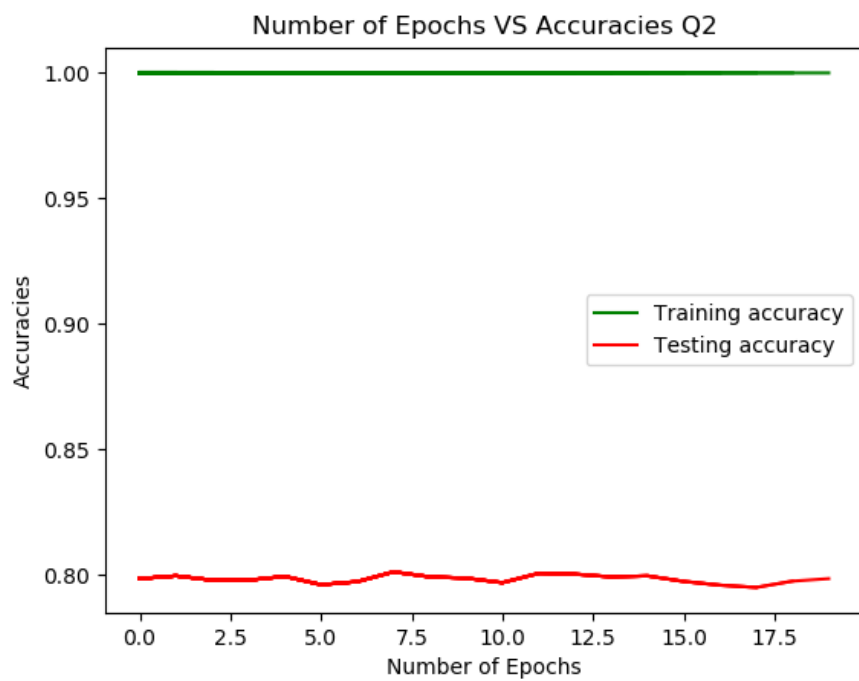


Figure 2.a. Accuracy vs Number of Epochs

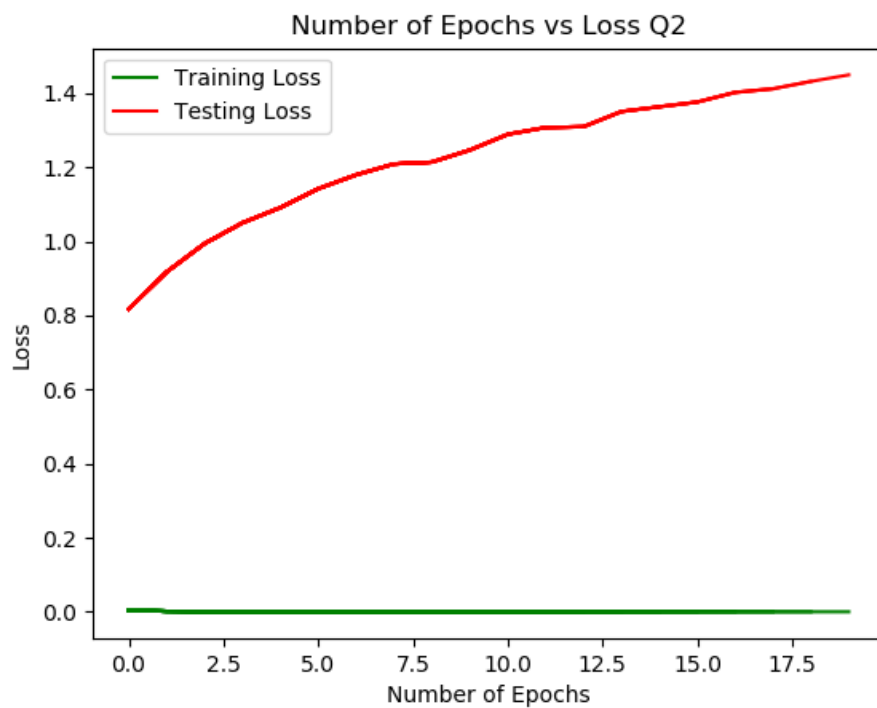


Figure 2.b. Loss vs Number of Epochs

Question 3

Answer: Adaptive scheduling for the learning rate was implemented as it can be seen in the code. I chose ADAM among the three. The pre-trained weights were not used for this question. Please note that it was run for **50 epochs**.

Following table summarizes the findings.

Num of Epochs	Learning Rate	Final Training Loss	Final Training Accuracy in %	Final Testing Loss	Final Testing Accuracy in %	Maximum Testing Accuracy, Acc-Max	Epoch Number for Acc-Max
50	0.01	0.04777	98.410	1.66531	73.240	74.33	14

Following are the plots for

a) Training and Testing Accuracies vs. Number of Epochs for the model with ADAM as the optimizer.

b) Training and Testing Losses vs. Number of Epochs for the model with ADAM as the optimizer

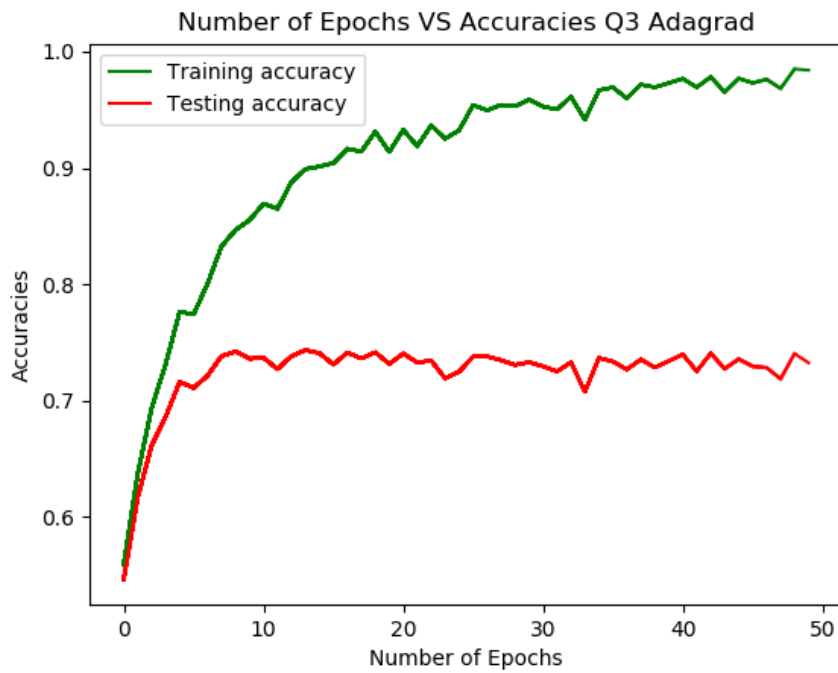


Figure 3 Accuracy vs Number of Epochs

Note: The above plot title has Adagrad instead of Adam by mistake. Thanks.

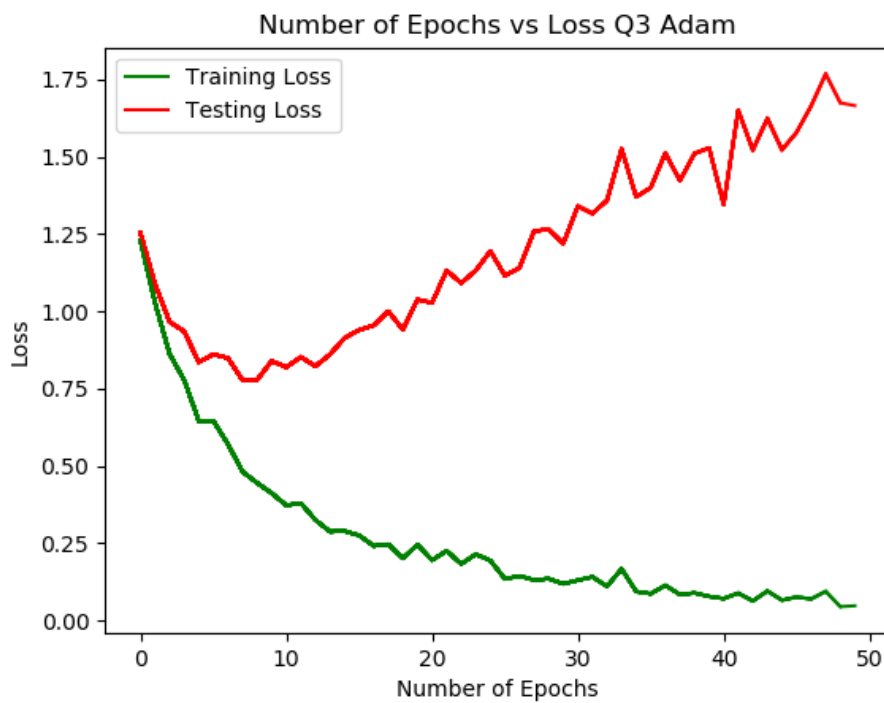


Figure 4 Loss vs Number of Epochs

I observed that running it for 50 epochs was of no use since it obtained its maximum testing accuracy before 20 epochs.

Question 4

Answer: I tried changing the learning rate, adding/ removing layers, using Leaky Relu, and some other ways. They did not yield good results. Hence I tried augmenting the data. For that, I used random cropping and random horizontal flip and it significantly improved the testing accuracy.

So, for the first setting the only change that I made was by data augmentation. Note that I ran the program for 100 epochs. One thing to notice here is that this is the first case where the testing loss has not increased after a decline in the beginning in general. We can see that the testing loss is decreasing with a curve that looks smooth in general and stays low instead of increasing after some epochs.

Following were the findings for the first setting.

Num of Epochs	Learning Rate	Final Training Loss	Final Training Accuracy in %	Final Testing Loss	Final Testing Accuracy in %	Maximum Testing Accuracy, Acc-Max	Epoch Number for Acc-Max
100	0.01	0.10143	96.444	0.47378	86.630	86.630	100

Following are the plots for the first setting.

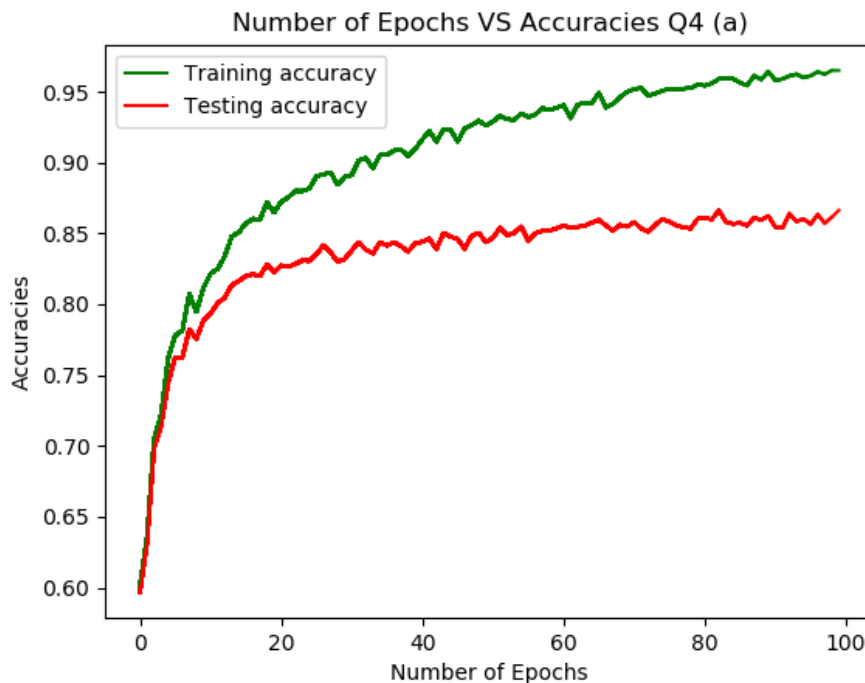


Figure 4.a. Accuracy vs Number of Epochs

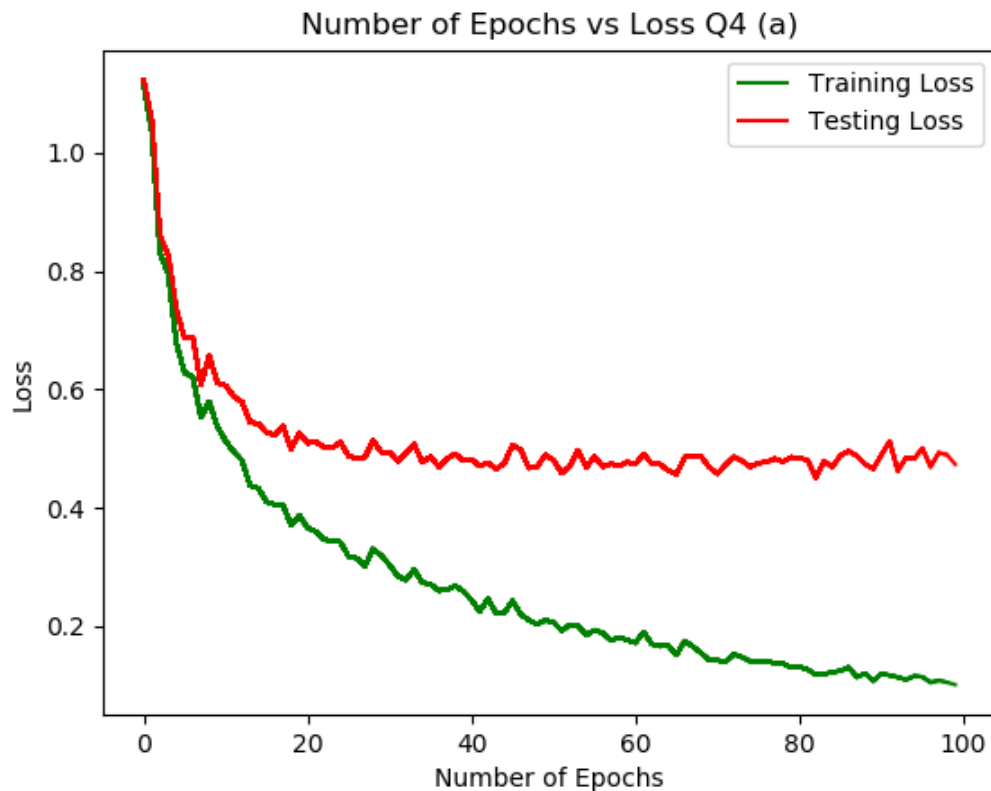


Figure 4.b. Loss vs Number of Epochs

After that, for the second setting, I made the following changes.

- a. I added a batch normalization layer after the first convolutional layer.
- b. I changed the number of output channels of the second convolutional layer from 32 to 64.
- c. I added another batch normalization layer after the second convolutional layer.
- d. I added a dropout layer after this second batch normalization layer
- e. I changed the number of output channels for the third convolutional layer to 128.
- f. I added another batch normalization layer after this third convolutional layer.
- g. I changed the number of output channels in the fourth convolutional layer to 256.
- h. I added another batch normalization layer after this fourth convolutional layer.
- i. I added a dropout layer after this new batch normalization layer.
- j. I added a new convolutional layer after this dropout layer which has 512 output channels.
- k. I added a dropout layer after the first fully-connected layer.
- l. I added another batch normalization layer after the last dropout layer.
- m. I changed all my Relu functions to Leaky Relu.
- n. I used a weight decay factor of $1e-05$ for the optimizer.
- o. I also used data augmentation, namely random cropping and random horizontal flip.

Num of Epochs	Learning Rate	Final Training Loss	Final Training Accuracy in %	Final Testing Loss	Final Testing Accuracy in %	Maximum Testing Accuracy, Acc-Max	Epoch Number for Acc-Max
100	0.01	0.04360	98.688	0.38110	89.080	89.350	78

Following are the plots for the second setting.

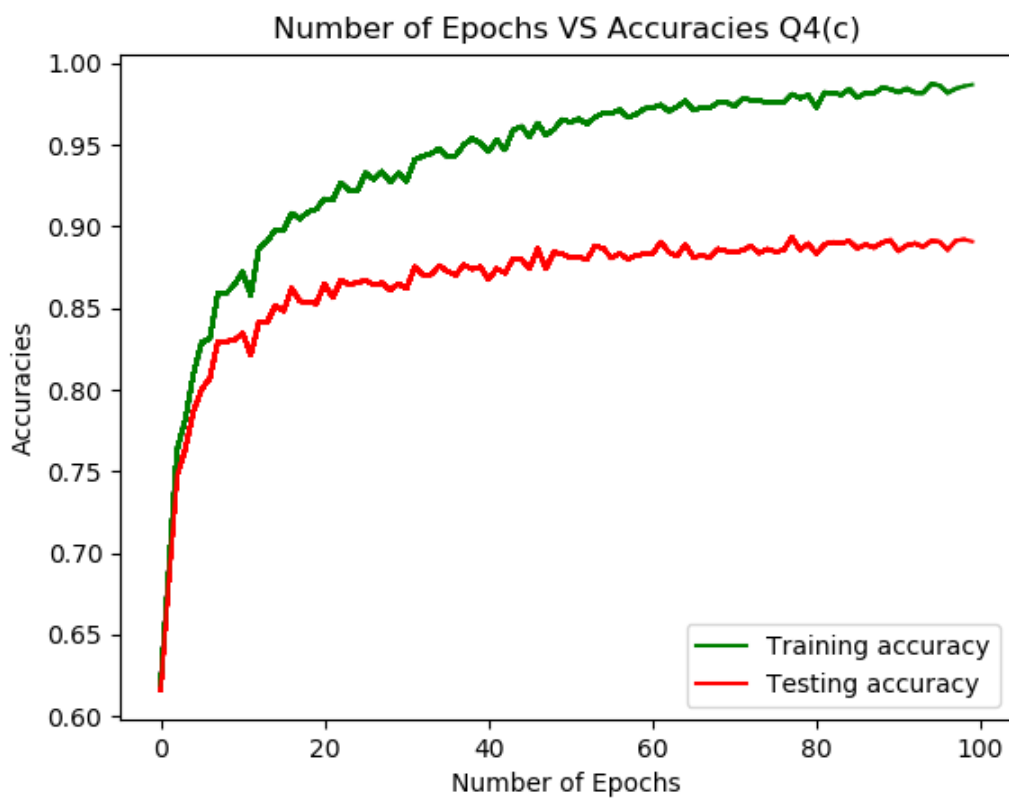


Figure 4.a. Accuracy vs Number of Epochs

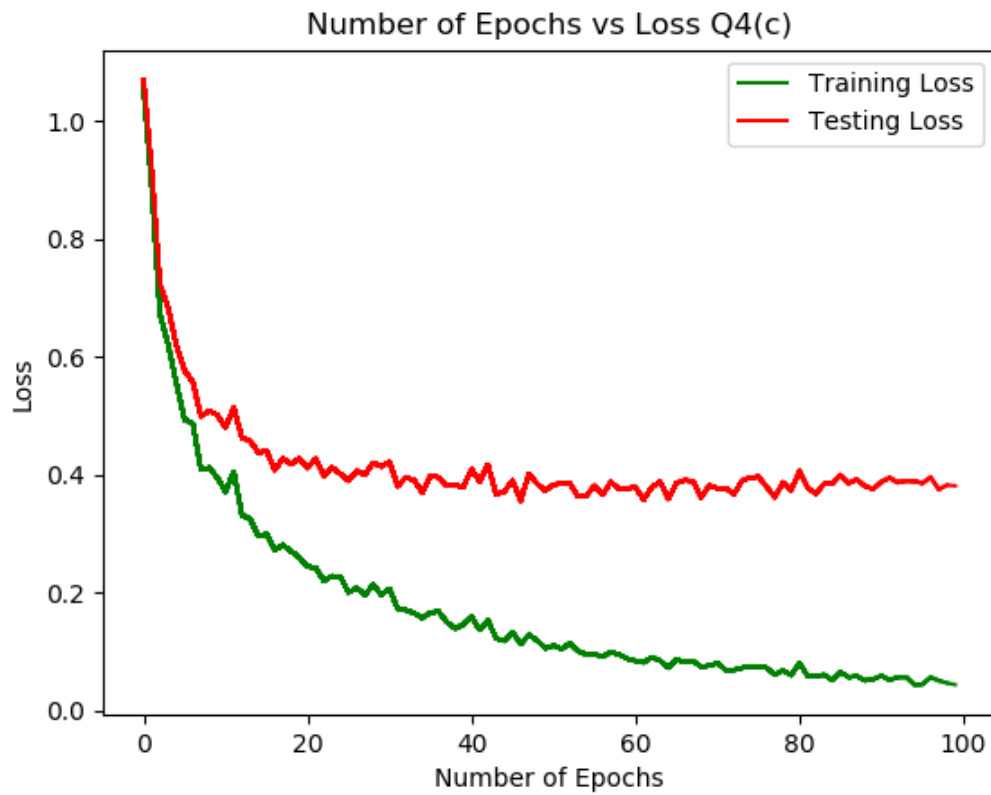


Figure 4.b. Loss vs Number of Epochs

As we can see, with these changes, we achieve a maximum testing accuracy of 89.350% in the 78th epoch. Also, I believe, running it for more epochs could have yielded a minor increment in the testing accuracy.