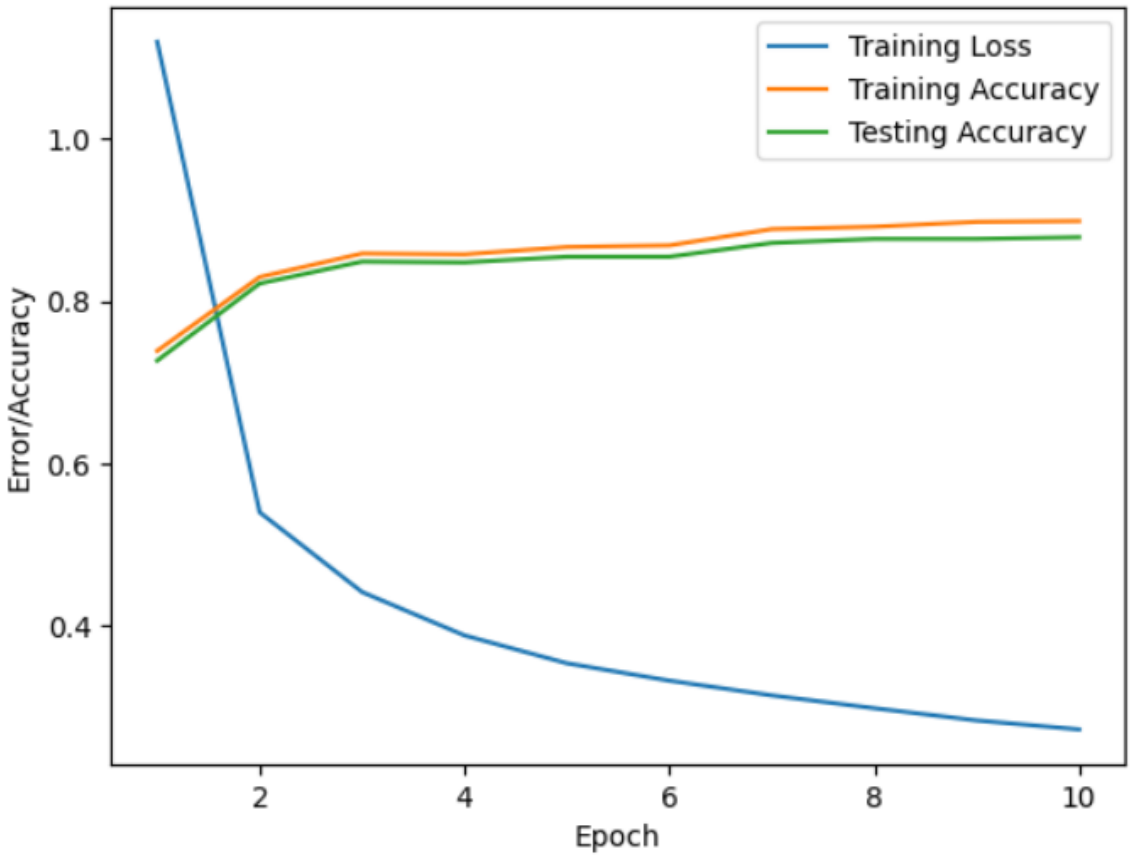
https://github.com/Christian-Martens-UNCC/ECGR-4106/tree/main/Homework_2-CNNs_%26_LeNet

Problem 1:

A) Compared to the training graph shown in the lectures for the LeNet model, this "updated" model has a higher accuracy and is a quicker learner at the same learning rates. It seems improved in nearly every way.

Figure 1 - Loss and Accuracy per Epoch for the updated LeNet Model



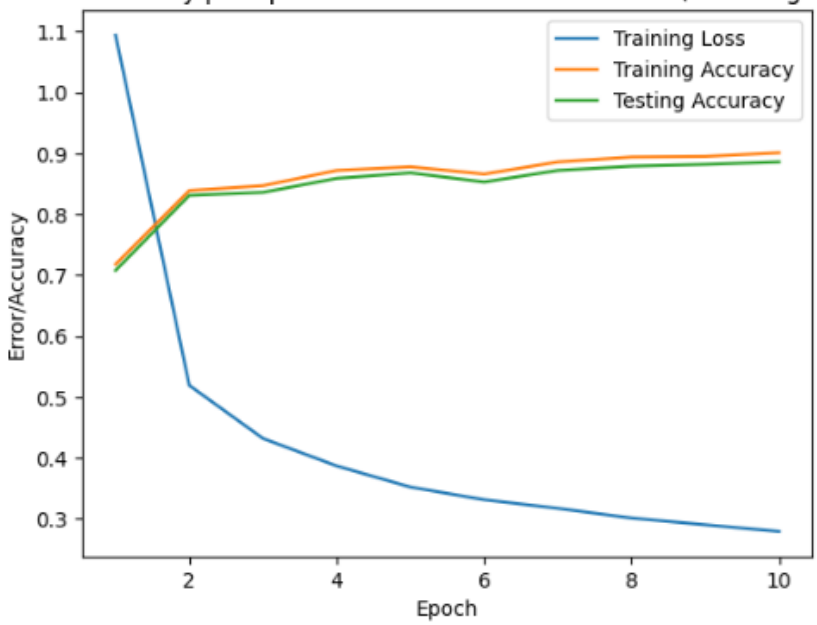

```
Epoch 10:
        Duration = 9.13 seconds
        Training Loss: 0.27235
        Training Accuracy: 0.899
        Validation Accuracy: 0.879
```

Problem 2:

A) I made two separate models for 2.a where one has a shrinking convolution window compared to LeNet (a 3x3 window instead of 5xt) and the other has an expanding convolution window (7x7). This is to test the two methods against each other to see which is more beneficial. I found that

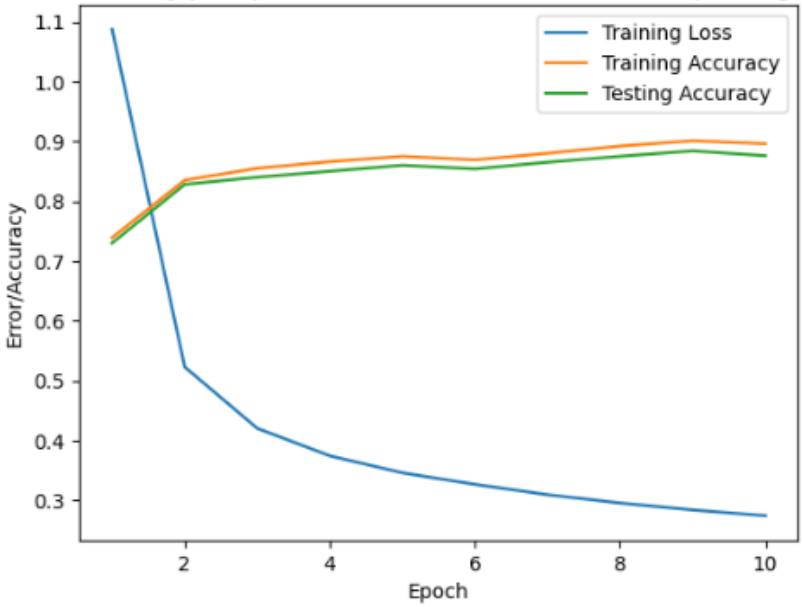
shrinking the convolution window to 3x3 was beneficial, adding roughly 1% of accuracy compared to the 5x5 and 7x7 models which had nearly identical accuracy.

Figure 2a - Loss and Accuracy per Epoch for the altered LeNet Model (shrinking convolution window)




```
Epoch 10:
        Duration = 8.199 seconds
        Training Loss: 0.27944
        Training Accuracy: 0.901
        Validation Accuracy: 0.886
```

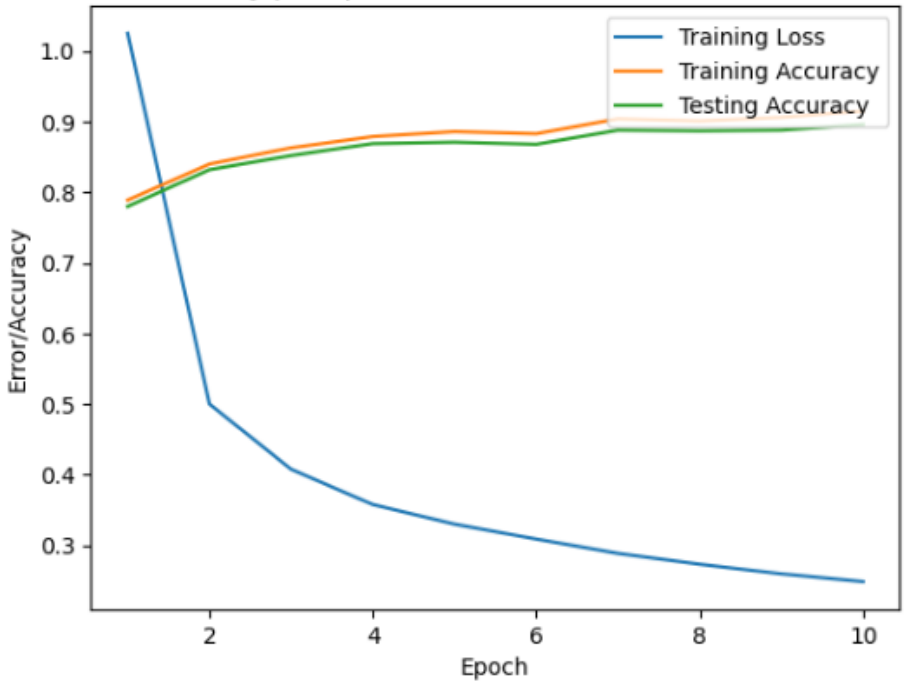
Figure 2b - Loss and Accuracy per Epoch for the altered LeNet Model (Expanding Convolution Window)




```
Epoch 10:
        Duration = 10.162 seconds
        Training Loss: 0.27412
        Training Accuracy: 0.896
        Validation Accuracy: 0.876
```

B) Doubling the layer width adds increased complexity at the expense of training time. As is shown below, the accuracy of the model is better than the standard and updated LeNet models by about 2%, but that comes at a cost of a roughly 20% increase in training time.

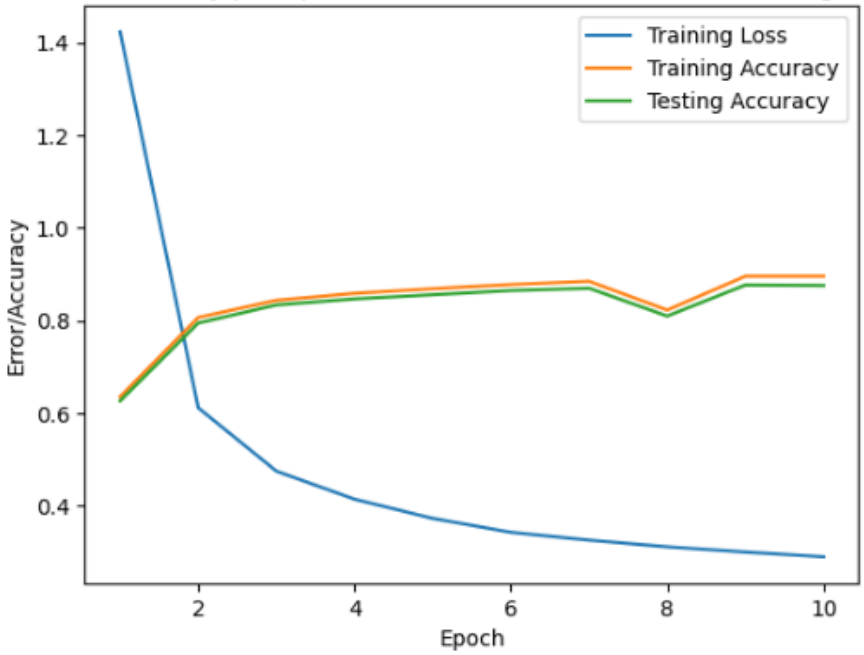
Figure 3 - Loss and Accuracy per Epoch for the altered LeNet Model (Doubling Layer Width)




```
Epoch 10:
        Duration = 11.317 seconds
        Training Loss: 0.24853
        Training Accuracy: 0.915
        Validation Accuracy: 0.896
```

C) Increasing the number of convolution layers did not seems to impact the accuracy of the model in any significant way and added to the training time. The accuracy of the model was nearly identical to that of the updated LeNet model without the extra convolution layer.

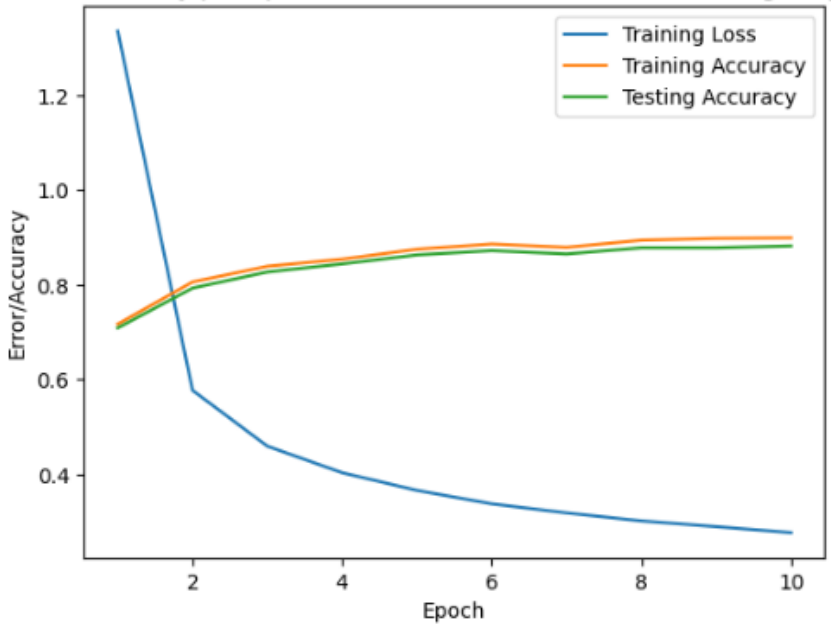
Figure 4 - Loss and Accuracy per Epoch for the altered LeNet Model (Adding Convolution Layer)




```
Epoch 10:
        Duration = 9.648 seconds
        Training Loss: 0.29061
        Training Accuracy: 0.896
        Validation Accuracy: 0.876
```

D) When adding an additional fully connected layer, we can see that the altered model has an increased validation accuracy of about 0.5% compared to the updated LeNet model. However, the added accuracy comes at a large cost to the training time (nearly 15%) so there is an argument to be made that the accuracy gain is not worth the increase in training time.

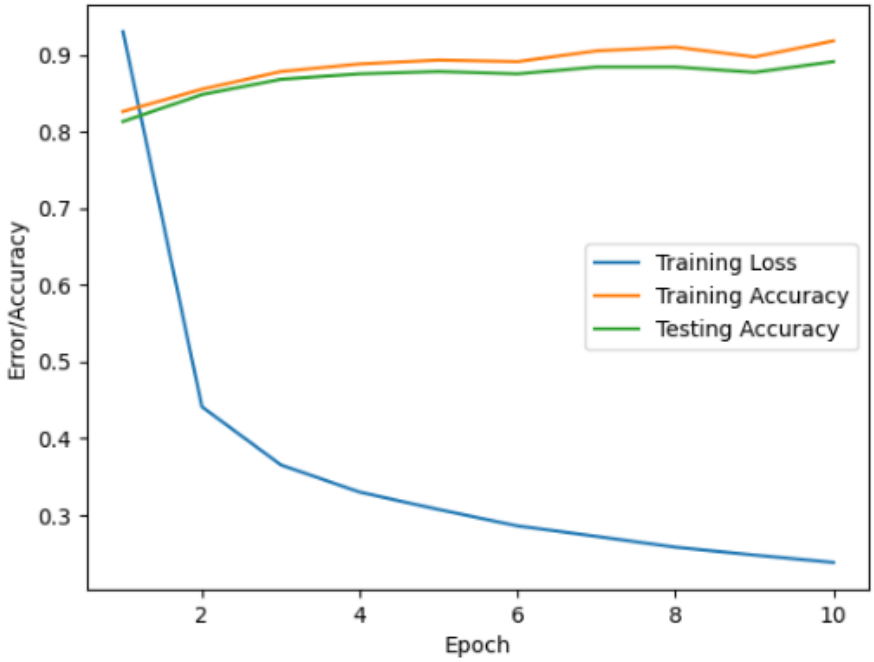
Figure 5 - Loss and Accuracy per Epoch for the altered LeNet Model (Adding Fully Connected Layer)




```
Epoch 10:
        Duration = 11.919 seconds
        Training Loss: 0.27715
        Training Accuracy: 0.899
        Validation Accuracy: 0.882
```

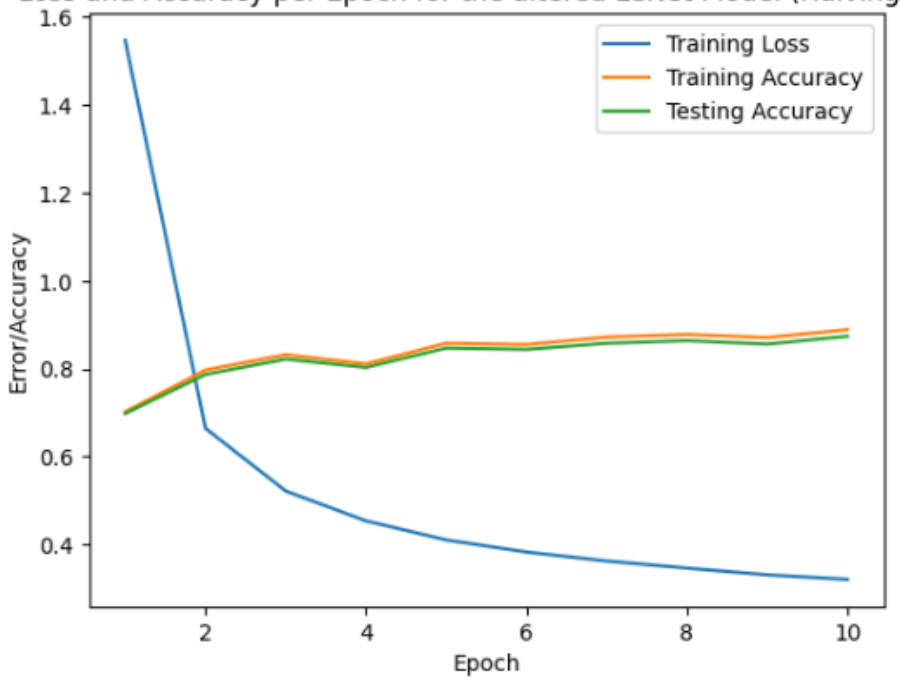
E) Two different models were tested to determine if doubling or halving the learning rate would lead to beneficial results in the training accuracy. According to the trained graphs, it appears that doubling the learning rate leads to a marked improvement in the accuracy of while halving the learning rate decreases the accuracy of the model. I suspect this is because, to keep the comparisons consistent, all trainings last for 10 epochs since there was heavy overfitting issues in problem 1. As such, the model with a lower learning rate does not have the time to reach its maximum potential while the higher learning rate can make its optimum. This seems confirmed by the overfitting occurring in the doubled learning rate model whereas no overfitting is occurring in the halved learning rate model.

Figure 6a - Loss and Accuracy per Epoch for the altered LeNet Model (Doubling Learning Rate)




```
Epoch 10:
        Duration = 10.245 seconds
        Training Loss: 0.23812
        Training Accuracy: 0.918
        Validation Accuracy: 0.891
```

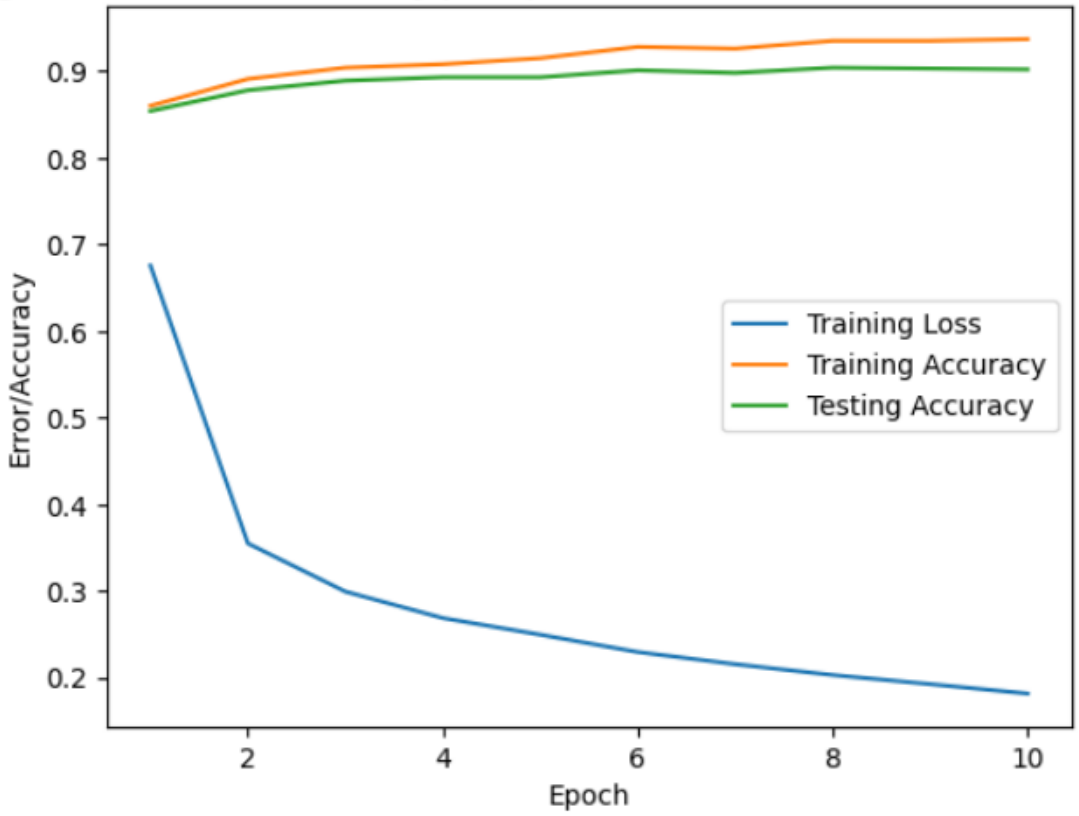
Figure 6b - Loss and Accuracy per Epoch for the altered LeNet Model (Halving Learning Rate)




```
Epoch 10:
        Duration = 11.12 seconds
        Training Loss: 0.32063
        Training Accuracy: 0.889
        Validation Accuracy: 0.874
```

F) An additional model was made using all the techniques which improved the validation accuracy of the LeNet model to satisfy my curiosity. Additionally, weight decay was used to increase the generalization. The resulting model had the highest accuracy model which was constructed. The model includes techniques such as decreasing the convolution window size, increasing the number of fully connected layers, increasing the learning rate, and increasing the layer width. This model also had an extremely high initial validation accuracy compared to all the other models by about 10%. However, the model is very large and the training time is 30% worse than the updated LeNet model. This model will be further discussed in problem 4.

Figure 7 - Loss and Accuracy per Epoch for the altered LeNet Model (Variety)

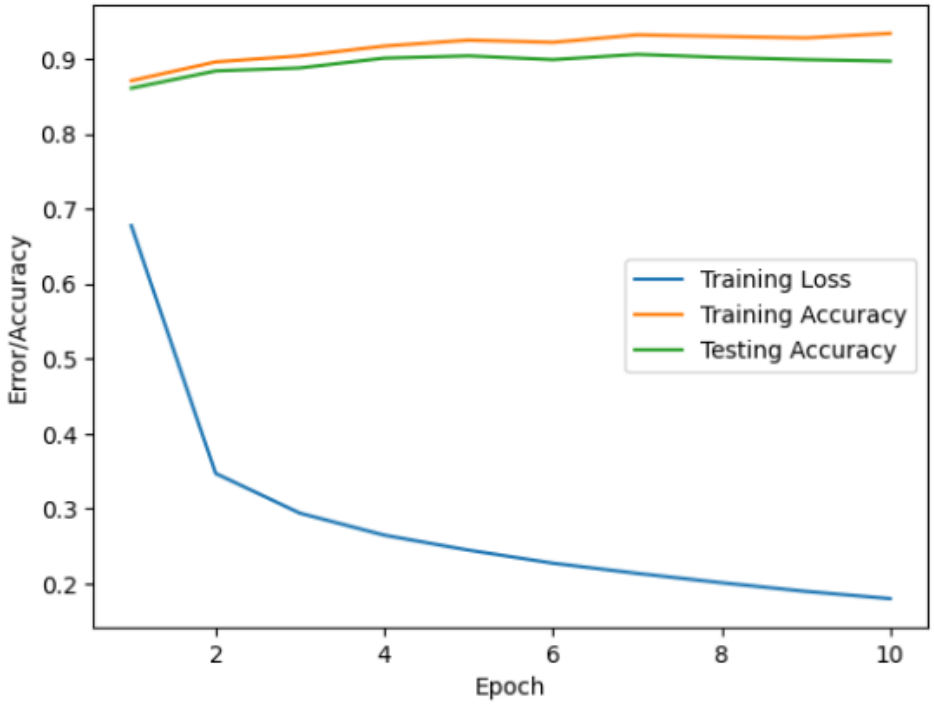



```
Epoch 10:
        Duration = 13.613 seconds
        Training Loss: 0.18169
        Training Accuracy: 0.937
        Validation Accuracy: 0.902
```

Problem 3:

A) It does not appear that the inclusion of the dropout layers had any significant effect on the training accuracy for my best-trained model.

Figure 8 - Loss and Accuracy per Epoch for the altered LeNet Model (Variety w/ Dropout)



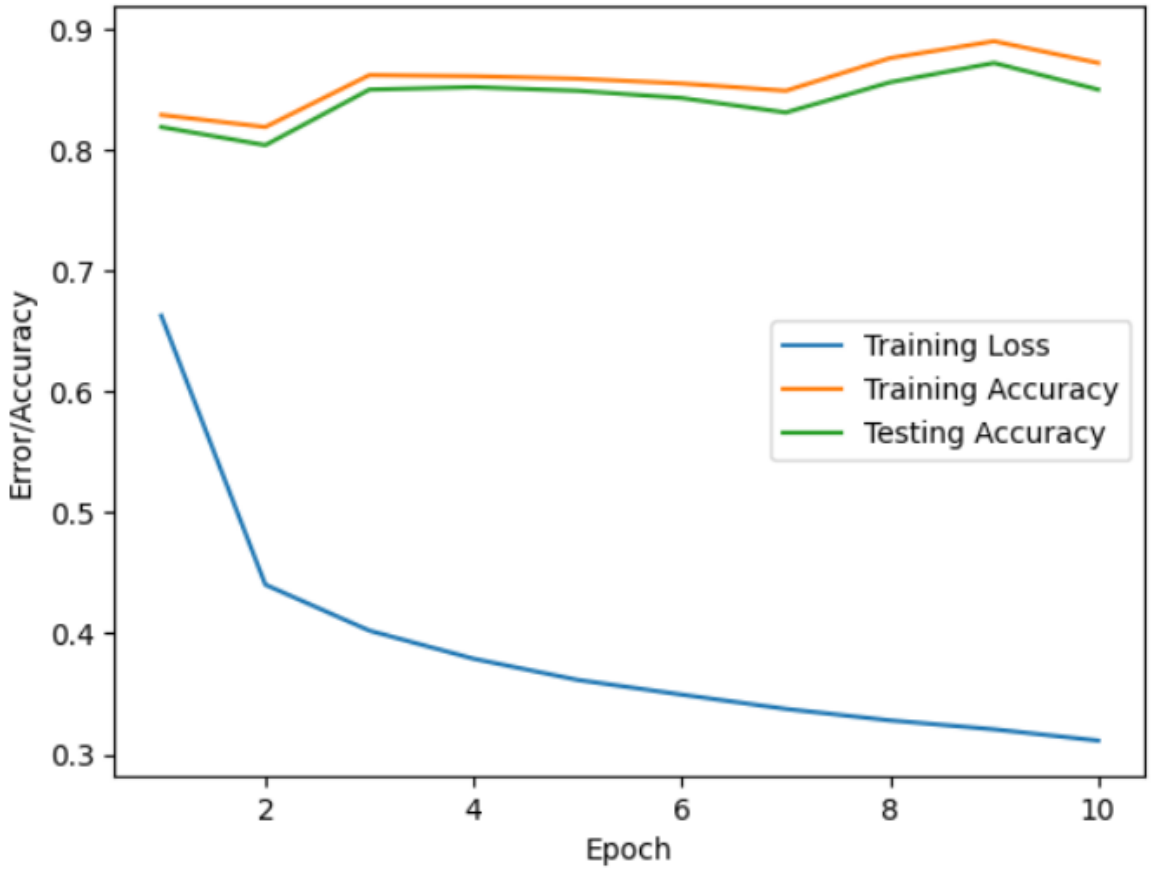

```
Epoch 10:
        Duration = 14.133 seconds
        Training Loss: 0.18009
        Training Accuracy: 0.934
        Validation Accuracy: 0.897
```

Problem 4:

A) The training accuracy, loss, and validation accuracy are all slightly worse than the updated LeNet and altered networks (a validation accuracy loss of roughly 2.5% compared to the updated LeNet model and a 5% accuracy loss compared to the altered model). However, the training is nearly 15% faster than the updated LeNet model and about 45% faster than the highest accuracy model that I constructed and the model is significantly more lightweight. The number of parameters of the simple model is an 80% reduction from the LeNet model and the number of MACs required of the simple model is a 94% reduction compared to LeNet.

```
drop_LeNet(
  (conv1): LazyConv2d(0, 8, kernel_size=(3, 3), stride=(2, 2))
  (maxp1): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_m
ode=False)
  (conv1_drop): Dropout2d(p=0.3, inplace=False)
  (fc1): LazyLinear(in_features=0, out_features=40, bias=True)
  (fc2): LazyLinear(in_features=0, out_features=10, bias=True)
  (relu): ReLU()
  (flat): Flatten(start_dim=1, end_dim=-1)
)
```

Figure 9 - Loss and Accuracy per Epoch for the simplified LeNet Model



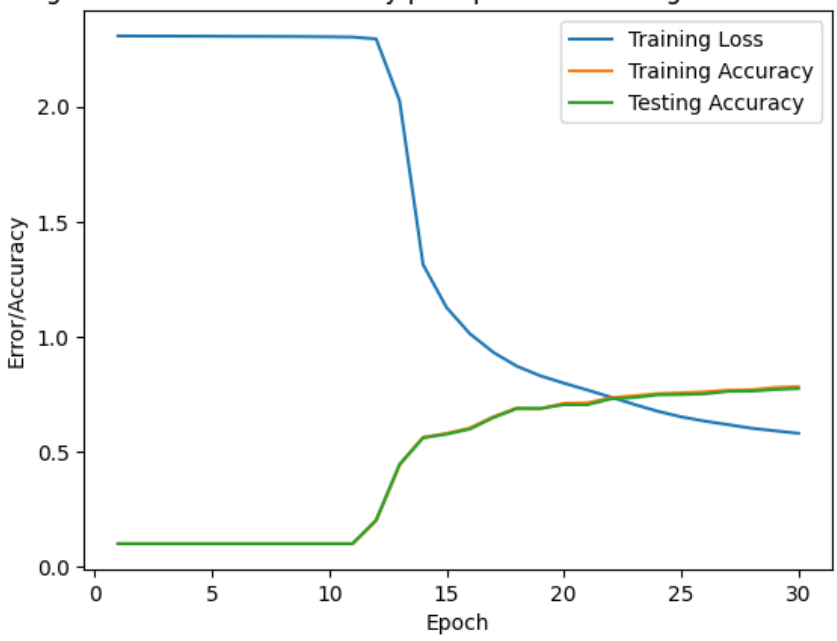

```
Epoch 10:
        Duration = 7.29 seconds
        Training Loss: 0.31137
        Training Accuracy: 0.872
        Validation Accuracy: 0.85
```

```
Computational complexity for LeNet:   435.85 KMac
Number of parameters for LeNet:   61.71 k

Computational complexity for Altered LeNet Model:   644.62 KMac
Number of parameters for Altered LeNet Model:   217.71 k

Computational complexity for Simple Model:   28.23 KMac
Number of parameters for Simple Model:   12.05 k
```

Problem 5:

A) Since I couldn't find any information about how long LeNet trained for, I created a replica of the LeNet model and trained it myself using the same techniques to verify its accuracy and have something to compare to it. The results I found were interesting to say the least. Using the original LeNet model, the accuracy capped at under 80%, meaning that the model used in Problem 4 matches the criteria for this problem as it is has less parameters and is less computationally complex than LeNet and has an accuracy of 85%. Additionally, the LeNet model needed 30 epochs to train to the final accuracy, while the model in Problem 4 reached its peak accuracy in under 10 epochs.

Figure 10 - Loss and Accuracy per Epoch for the original LeNet Model




```
Epoch 30:
        Duration = 18.923 seconds
        Training Loss: 0.57976
        Training Accuracy: 0.783
        Validation Accuracy: 0.774
```