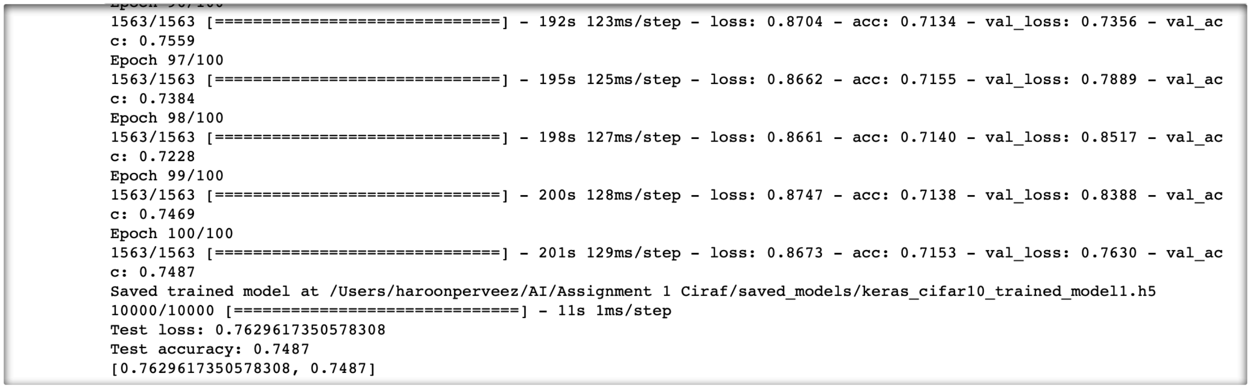
CNN – CIRAF 10 – Part B

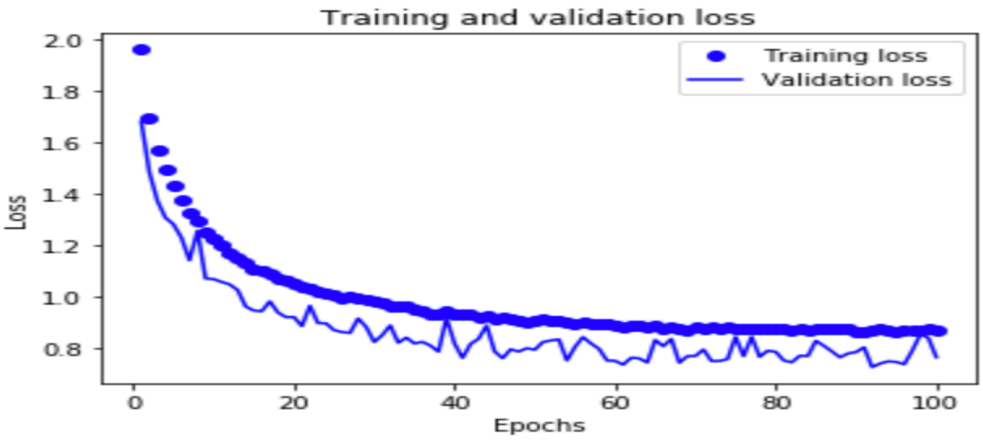
Syed Haroon Perveez

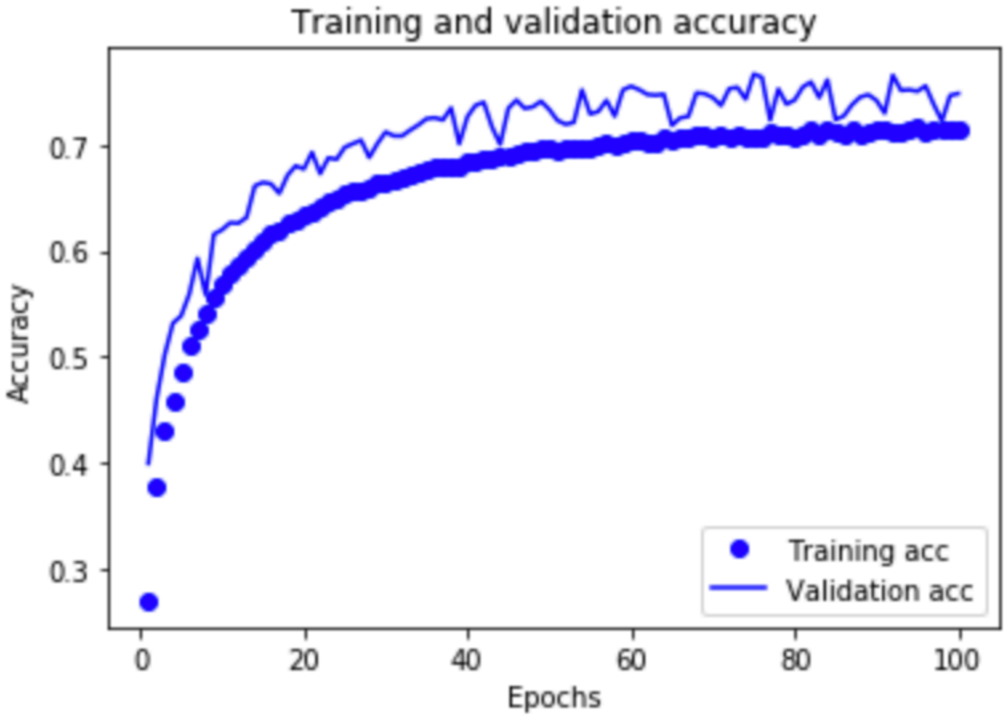
Observation 1
Evaluating Base Model

- Number of hidden layers: 4 (32,32,64,64)
- Output layer nodes: 10
- Activation: Relu,
- Dropout layers: 1(0.25), 2 (0.25) , 3(0.5)
- Optimizer: RMSprop
- Loss: categorical_crossentropy
- Metric: Accuracy
- Batch size = 32
- Epochs = 100
- **Output of this model**: Training Loss: 0.8673; Training accuracy: 71.53 %; Testing Loss: 0.7630; Testing accuracy 74.87%



```
Epoch 96/100
1563/1563 [==============================] - 192s 123ms/step - loss: 0.8704 - acc: 0.7134 - val_loss: 0.7356 - val_ac
c: 0.7559
Epoch 97/100
1563/1563 [==============================] - 195s 125ms/step - loss: 0.8662 - acc: 0.7155 - val_loss: 0.7889 - val_ac
c: 0.7384
Epoch 98/100
1563/1563 [==============================] - 198s 127ms/step - loss: 0.8661 - acc: 0.7140 - val_loss: 0.8517 - val_ac
c: 0.7228
Epoch 99/100
1563/1563 [==============================] - 200s 128ms/step - loss: 0.8747 - acc: 0.7138 - val_loss: 0.8388 - val_ac
c: 0.7469
Epoch 100/100
1563/1563 [==============================] - 201s 129ms/step - loss: 0.8673 - acc: 0.7153 - val_loss: 0.7630 - val_ac
c: 0.7487
Saved trained model at /Users/haroonperveez/AI/Assignment 1 Ciraf/saved_models/keras_cifar10_trained_model1.h5
10000/10000 [==============================] - 11s 1ms/step
Test loss: 0.7629617350578308
Test accuracy: 0.7487
[0.7629617350578308, 0.7487]
```





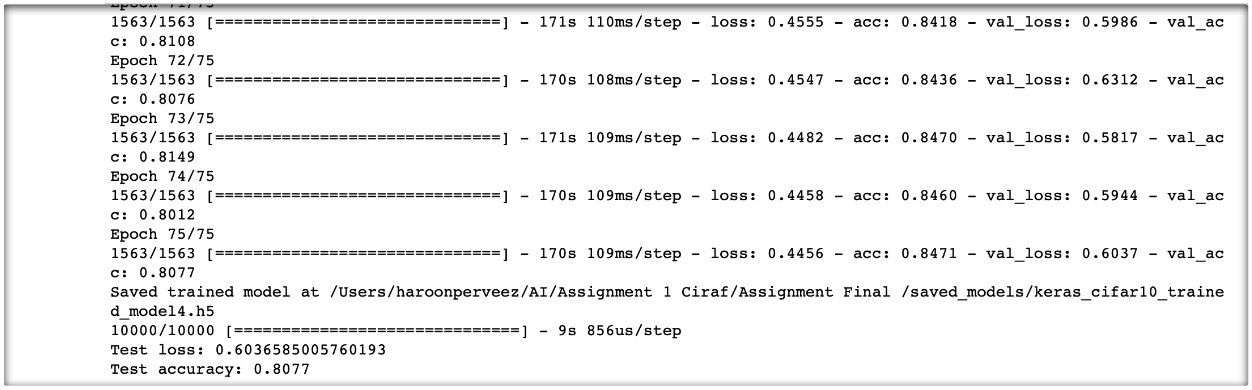
As mentioned by the author in base code, it is clear that this model is underfitting and can be optimized

After varying various parameters and running multiple test models, these were the best three models I can up with and improvements I made gradually

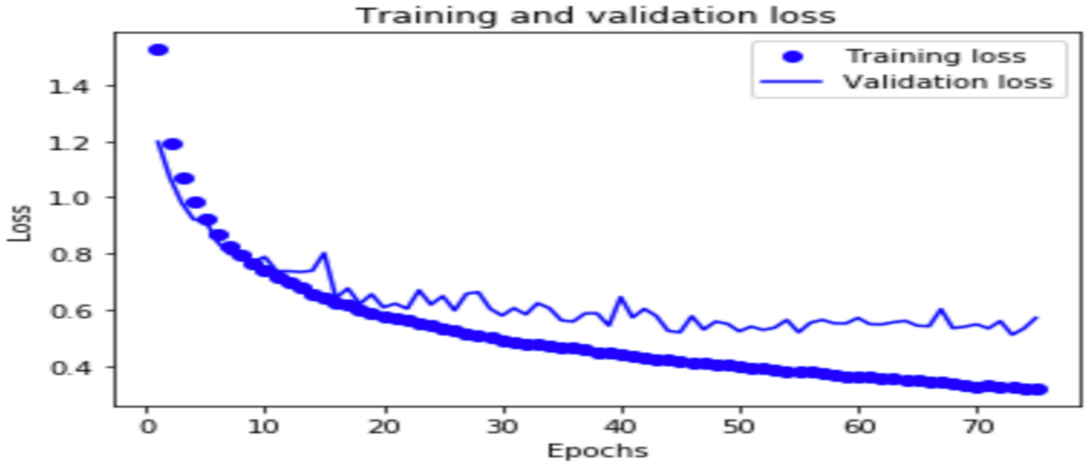
Observation 2 – Model 4 in Notebook

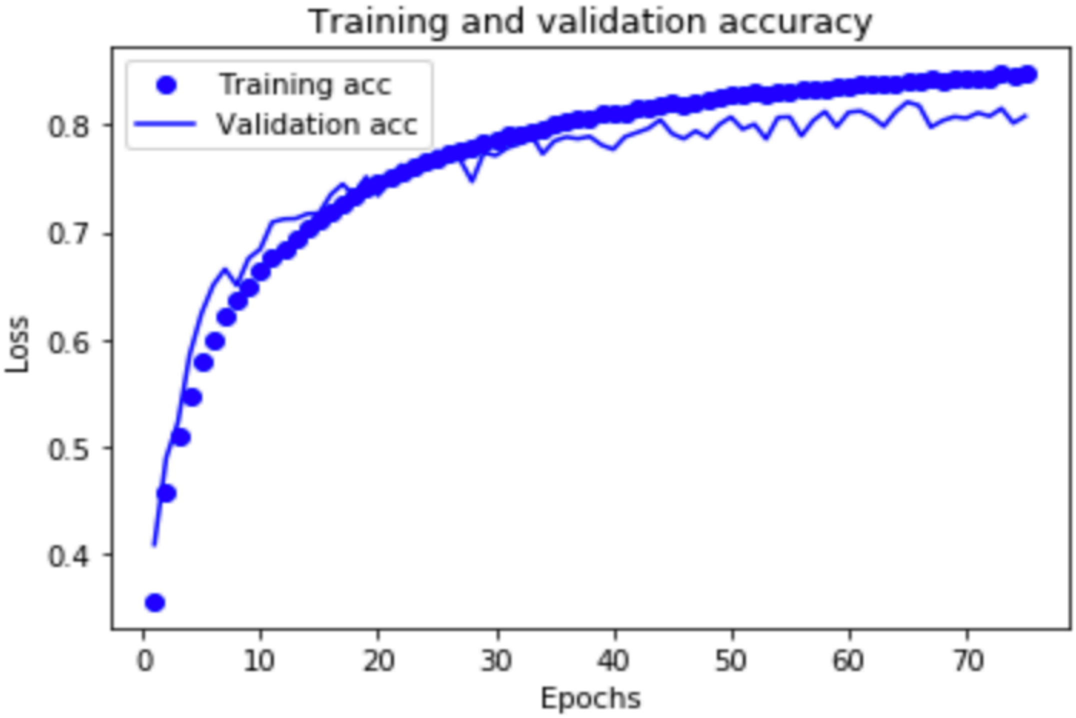
In this model as there was underfitting in previous model, I have removed the dropouts and increased the final dense layer to 512 neurons

- Number of hidden layers: 4 (32,32,64,64)
- Output layer nodes: 10
- Activation: Relu,
- Dropout layers: N.A
- Optimizer: RMSprop
- Loss: categorical_crossentropy
- Metric: Accuracy
- Batch size = 32
- Epochs = 75
- **Output of this model**: Training Loss: 0.446; Training accuracy: 84.71%; Testing Loss: 0.6037; Testing accuracy 80.77%



```
Epoch 71/75
1563/1563 [==============================] - 171s 110ms/step - loss: 0.4555 - acc: 0.8418 - val_loss: 0.5986 - val_ac
c: 0.8108
Epoch 72/75
1563/1563 [==============================] - 170s 108ms/step - loss: 0.4547 - acc: 0.8436 - val_loss: 0.6312 - val_ac
c: 0.8076
Epoch 73/75
1563/1563 [==============================] - 171s 109ms/step - loss: 0.4482 - acc: 0.8470 - val_loss: 0.5817 - val_ac
c: 0.8149
Epoch 74/75
1563/1563 [==============================] - 170s 109ms/step - loss: 0.4458 - acc: 0.8460 - val_loss: 0.5944 - val_ac
c: 0.8012
Epoch 75/75
1563/1563 [==============================] - 170s 109ms/step - loss: 0.4456 - acc: 0.8471 - val_loss: 0.6037 - val_ac
c: 0.8077
Saved trained model at /Users/haroonperveez/AI/Assignment 1 Ciraf/Assignment Final /saved_models/keras_cifar10_traine
d_model4.h5
10000/10000 [==============================] - 9s 856us/step
Test loss: 0.6036585005760193
Test accuracy: 0.8077
```





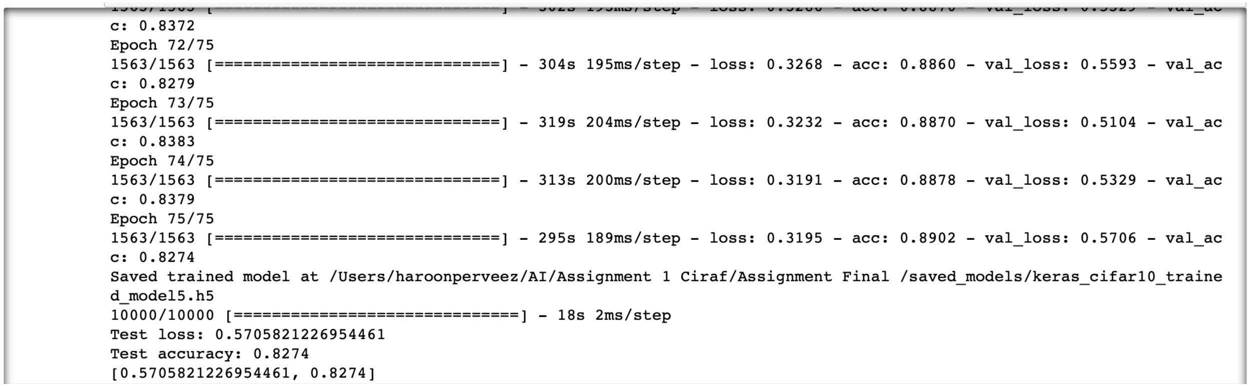
In this observation, I have improved the accuracy to 81% but now the model tends to overfit. We can see in above image that the optimum epochs seems to be between 45-60 to prevent overfitting

Observation 3 – Model 5 in Notebook

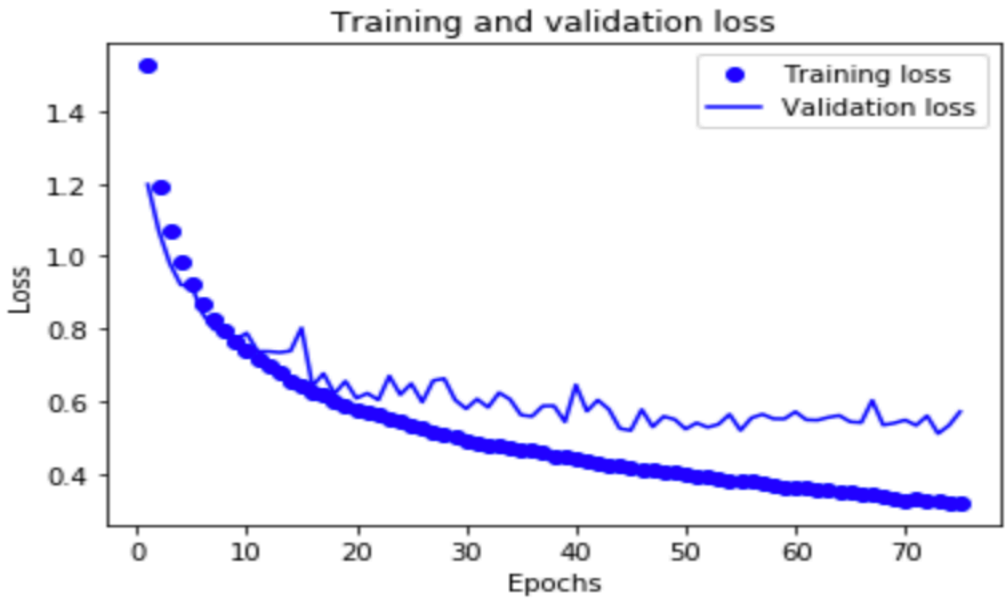
In this model I have replaced the ReLU function with Leaky ReLU again because it is better than ReLU function as it also has a small negative edge which works better with negative values.

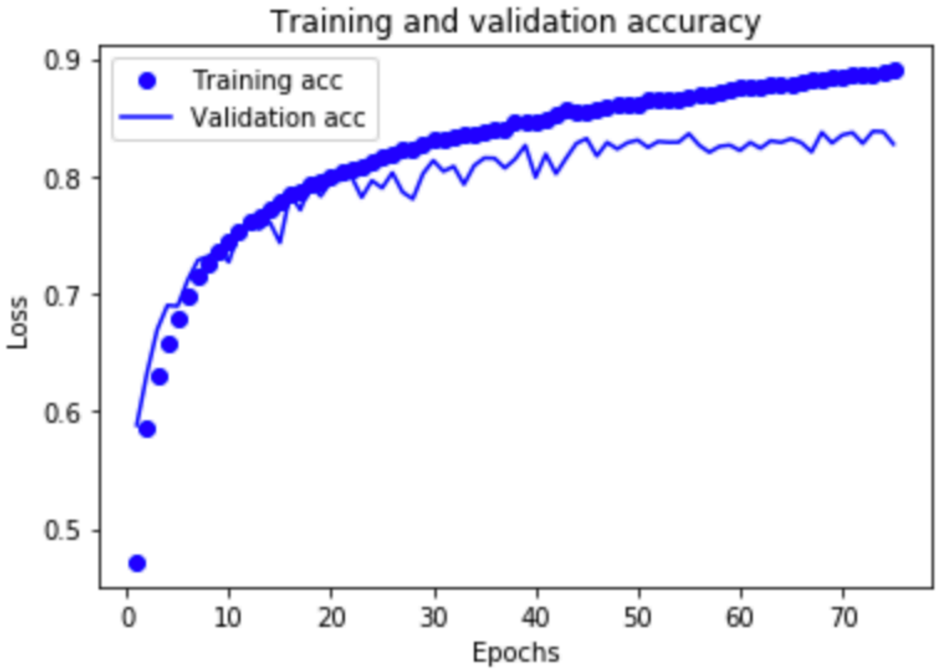
Also, as I was experimenting with batch-normalization, I have added a layer of batch-normalization, before MaxPooling Layer

- Number of hidden layers: 4 (32,32,64,64)
- Output layer nodes: 10
- Activation: leakyReLU,
- Dropout layers: N.A
- Optimizer: RMSprop
- Loss: categorical_crossentropy
- Metric: Accuracy
- Batch size = 32
- Epochs = 75
- **Output of this model**: Training Loss:0.3195; Training accuracy: 88.78%; Testing Loss: 0.5329; Testing accuracy 83.79%



```
1563/1563 [==============================] - 302s 193ms/step - loss: 0.3266 - acc: 0.8870 - val_loss: 0.5329 - val_ac
c: 0.8372
Epoch 72/75
1563/1563 [==============================] - 304s 195ms/step - loss: 0.3268 - acc: 0.8860 - val_loss: 0.5593 - val_ac
c: 0.8279
Epoch 73/75
1563/1563 [==============================] - 319s 204ms/step - loss: 0.3232 - acc: 0.8870 - val_loss: 0.5104 - val_ac
c: 0.8383
Epoch 74/75
1563/1563 [==============================] - 313s 200ms/step - loss: 0.3191 - acc: 0.8878 - val_loss: 0.5329 - val_ac
c: 0.8379
Epoch 75/75
1563/1563 [==============================] - 295s 189ms/step - loss: 0.3195 - acc: 0.8902 - val_loss: 0.5706 - val_ac
c: 0.8274
Saved trained model at /Users/haroonperveez/AI/Assignment 1 Ciraf/Assignment Final /saved_models/keras_cifar10_traine
d_model5.h5
10000/10000 [==============================] - 18s 2ms/step
Test loss: 0.5705821226954461
Test accuracy: 0.8274
[0.5705821226954461, 0.8274]
```



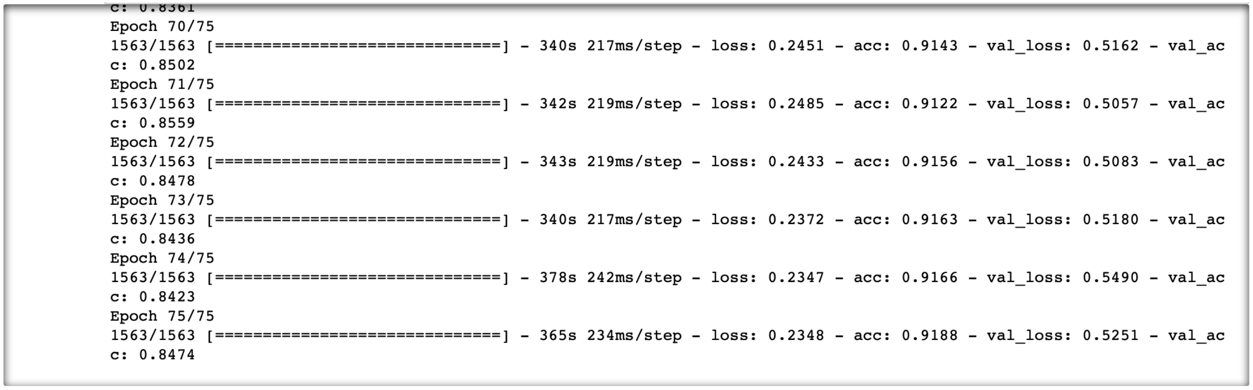


In this model, the validation accuracy is improved by 3% to ~84%. Although due to 75 epochs, the model tends to overfit after 50 epochs

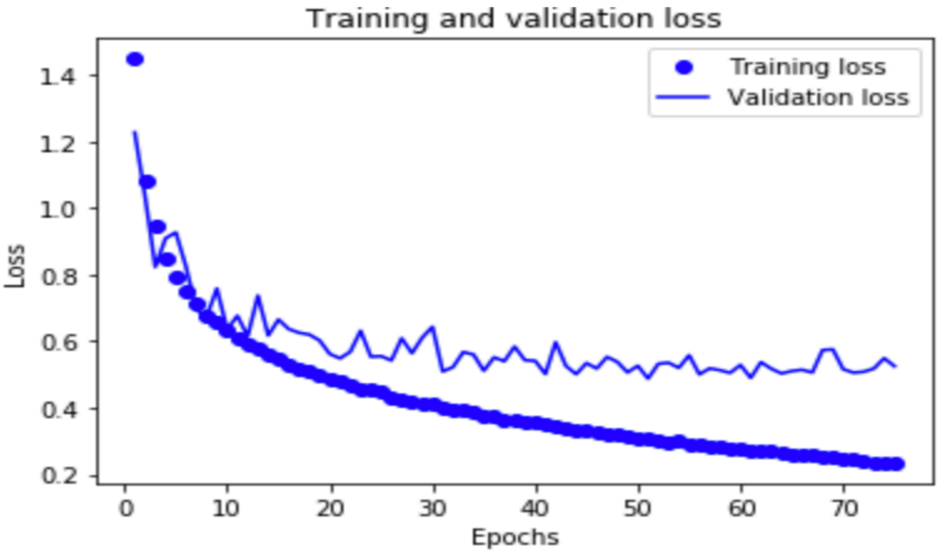
Observation 4 – Model 6 in Notebook

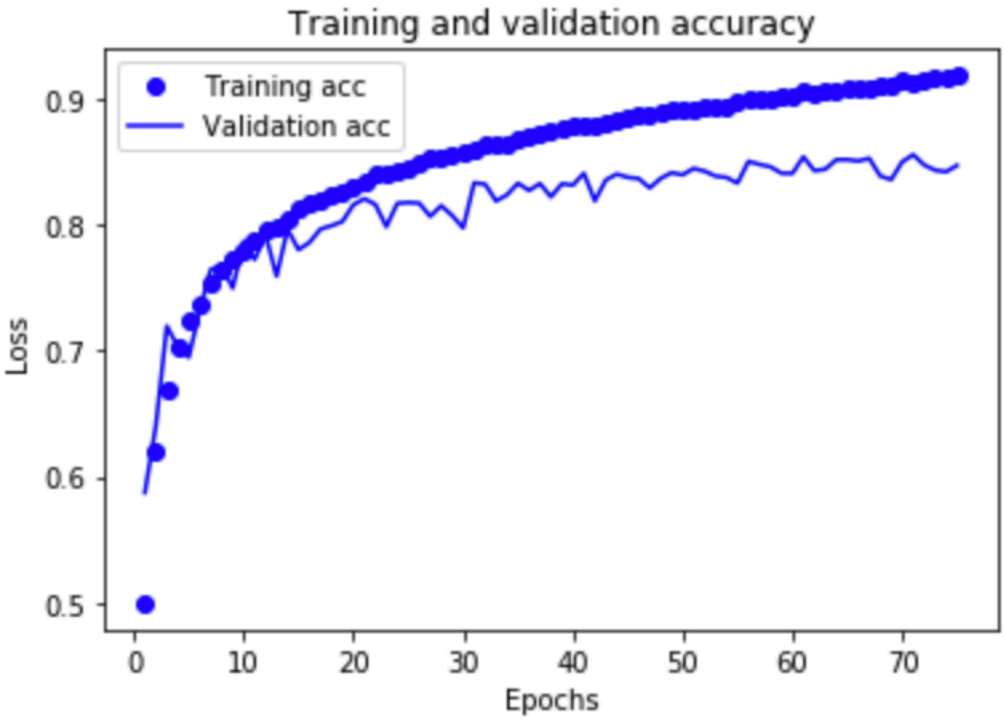
After the class discussion, I ran the model with Adam optimizer with a learning rate of 0.003(mentioned by Wenqi Cui by grid search technique) instead of RMSprop

- Number of hidden layers: 4 (32,32,64,64)
- Output layer nodes: 10
- Activation: leakyReLU,
- Dropout layers: N.A
- Optimizer: Adam
- Loss: categorical_crossentropy
- Metric: Accuracy
- Batch size = 32
- Epochs = 75
- **Output of this model**: Training Loss:0.2348; Training accuracy: 91.88%; Testing Loss: 0.5251; Testing accuracy 84.74%



```
c: 0.8361
Epoch 70/75
1563/1563 [==============================] - 340s 217ms/step - loss: 0.2451 - acc: 0.9143 - val_loss: 0.5162 - val_ac
c: 0.8502
Epoch 71/75
1563/1563 [==============================] - 342s 219ms/step - loss: 0.2485 - acc: 0.9122 - val_loss: 0.5057 - val_ac
c: 0.8559
Epoch 72/75
1563/1563 [==============================] - 343s 219ms/step - loss: 0.2433 - acc: 0.9156 - val_loss: 0.5083 - val_ac
c: 0.8478
Epoch 73/75
1563/1563 [==============================] - 340s 217ms/step - loss: 0.2372 - acc: 0.9163 - val_loss: 0.5180 - val_ac
c: 0.8436
Epoch 74/75
1563/1563 [==============================] - 378s 242ms/step - loss: 0.2347 - acc: 0.9166 - val_loss: 0.5490 - val_ac
c: 0.8423
Epoch 75/75
1563/1563 [==============================] - 365s 234ms/step - loss: 0.2348 - acc: 0.9188 - val_loss: 0.5251 - val_ac
c: 0.8474
```





It can be observed that the model accuracy has removed but it tends to overfit with increase in number of epochs.

From multiple tests, we can conclude that most optimal epochs are between 40-55 to prevent overfitting. But this may not provide a train accuracy of ~92%