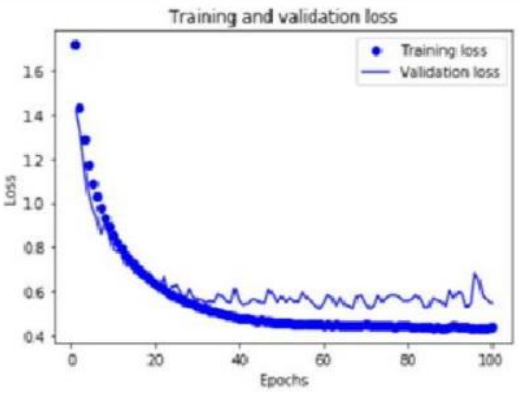
Assignment 2

Dataset: Cipher 10

Previous findings:

When we started building a model on CNN we faced overfitting and underfitting issues with our data. The task assigned to us during first assignment was to test various optimizers. The best result we got during that experiment was with RMS Prop optimizer i.e. accuracy was 82.21 as shown in figure below

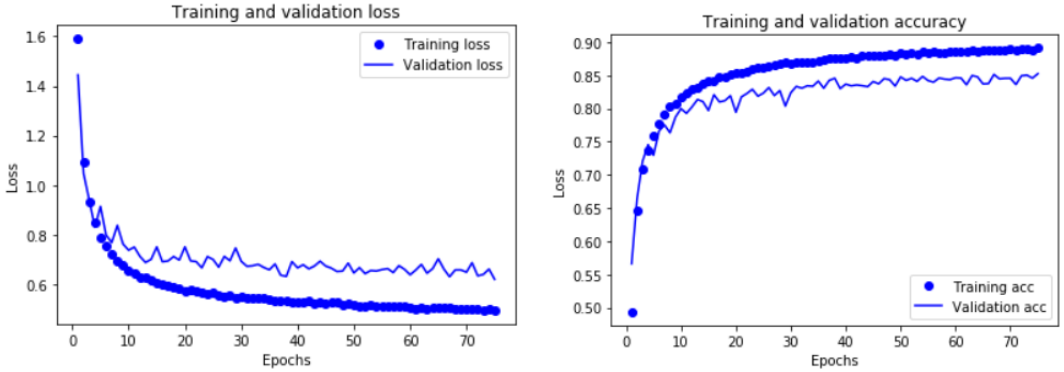



```
Saved trained model at /home/ubuntu/
10000/10000 [==============================
Test loss: 0.546309899520874
Test accuracy: 0.8221
```

New findings 1:

For our assignment 2 we started tweaking with parameters. We tried increasing/decreasing convolution layers, tried grid search to obtain hyperparameters. Following are the different strategies we used:

1) For data preprocessing of pixel values, we used the **z-score** instead of dividing the pixel values by 255. The earlier approach restricted the value of data between 0 and 1. Z-score makes the data more centered towards 0 and have a similar range. We used one-hot encoding technique for the labels which is same as before.
2) In the convolution layers, we added l1-l2 regularize to reduce noise and also used BatchNormalization to keep the data centered around 0 as per the earlier approach to find z-score
3) We added rotation for data augmentation apart from horizontal flips and height and width shift we used earlier
4) This result was running for 75 epochs and gave us around 89% accuracy for training data and around 85% accuracy for test data. The corresponding loss for training and test data were 0.49 and 0.62. They can be better viewed from the following graph:

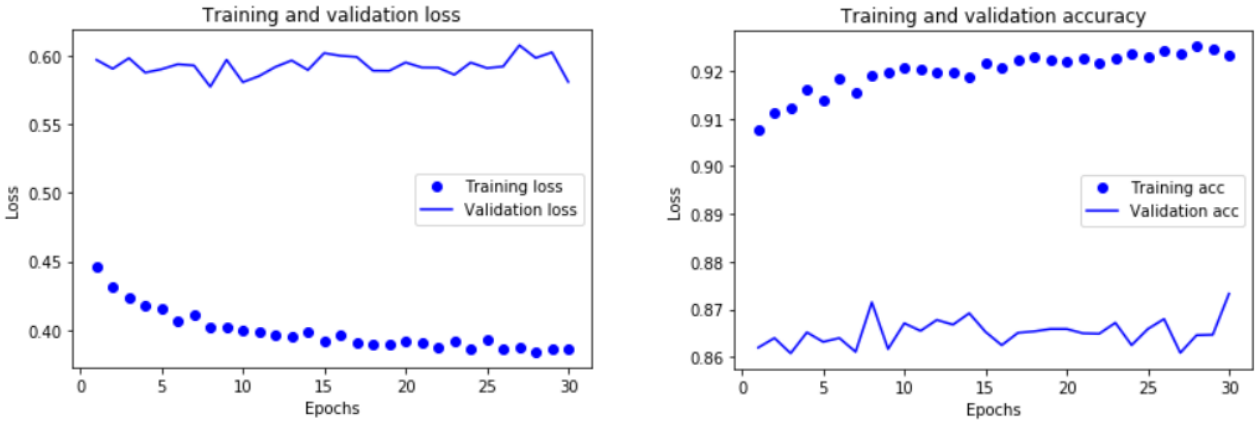



```
10000/10000 [==============================] - 3s

Test result: 87.880 loss: 0.546
Predicted label for test sample : 3
Actual label for test sample : [3]
```

New findings 2:

1) To further improve our accuracy, we decided to reuse the trained model by freezing all its layers except the last 7 starting from conv2d_6
2) We also decreased the learning rate this time from 0.001 to 0.0005. This ensured we don't get stuck on local minima. This enhanced model was trained for 30 epochs and gave us a training accuracy of around 92% and test accuracy of around 87%. The corresponding loss for training and test data were 0.38 and 0.58. They can be better viewed from the following graph:




```
Test result: 89.330 loss: 0.520
Predicted label for test sample : 3
Actual label for test sample : [3]
```

Conclusion:

Adding L2-L2 regularizers along with fine tuning of model using transfer learning we could achieve better accuracy of 89% and 85% on training and validation data respectively.