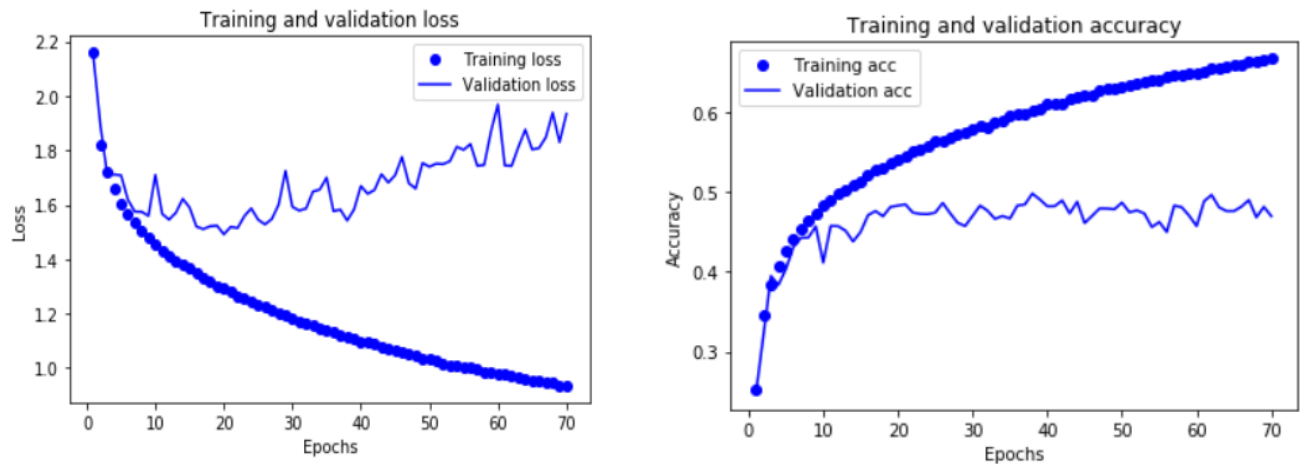


Part 1: Multilayer Perceptron

Following are the steps which we took while developing our model:

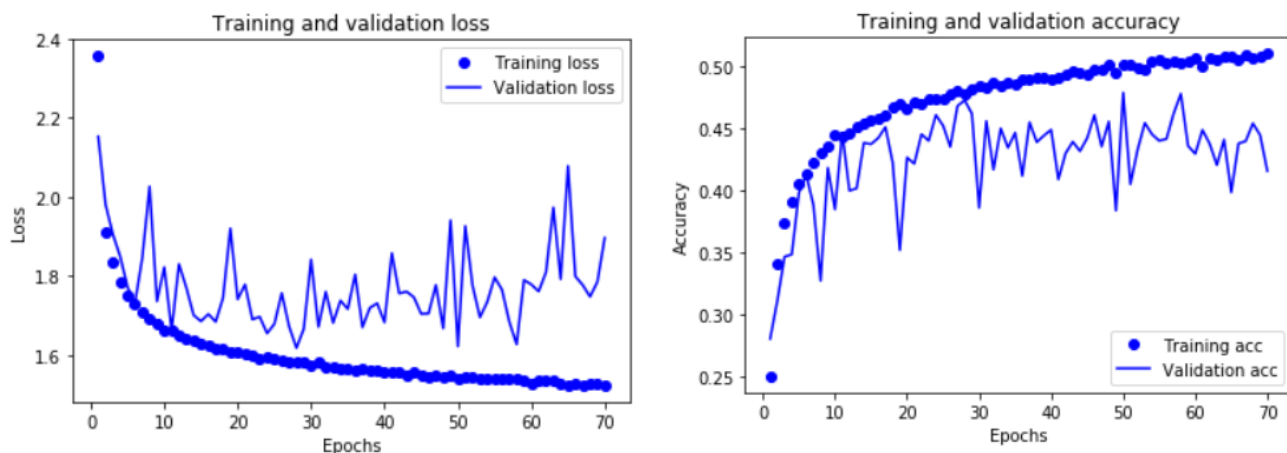
- 1) Data Preprocessing (Reshaping, normalizing, splitting data into training, validation, test sets)
- 2) Prepared config file
- 3) Build, Compile and Fit model
- 4) Plotted Loss and Accuracy

We Initially started with overfitting our model by Add layers (3), Make the layers bigger (Dense - 512), Train for more epochs (70). By doing so we got highest accuracy but our data was over fitting as shown in the graph 1A below.



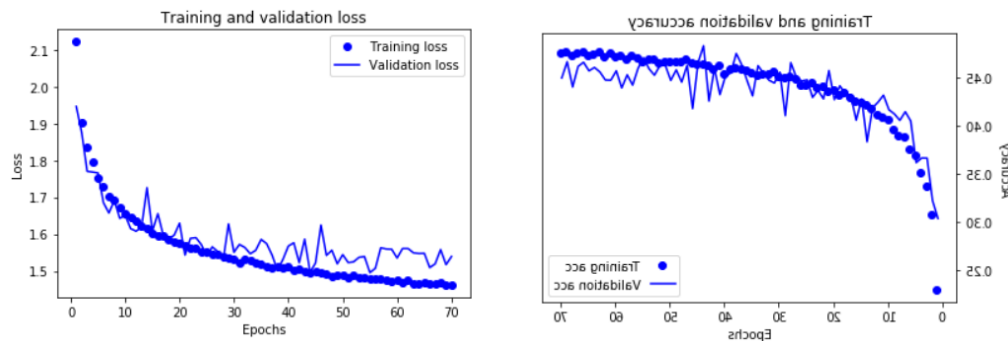
```
10000/10000 [=====] - 1s 50us/step
10000/10000 [=====] - 1s 50us/step
[1.8940791191101074, 0.4699]
[1.9337019464492797, 0.4699]
```

To Reduce over-fitting, I tried adding L1, L2, L1-L2 regularization function. As shown in graph below. It reduced our overfitting but accuracy degraded as well.



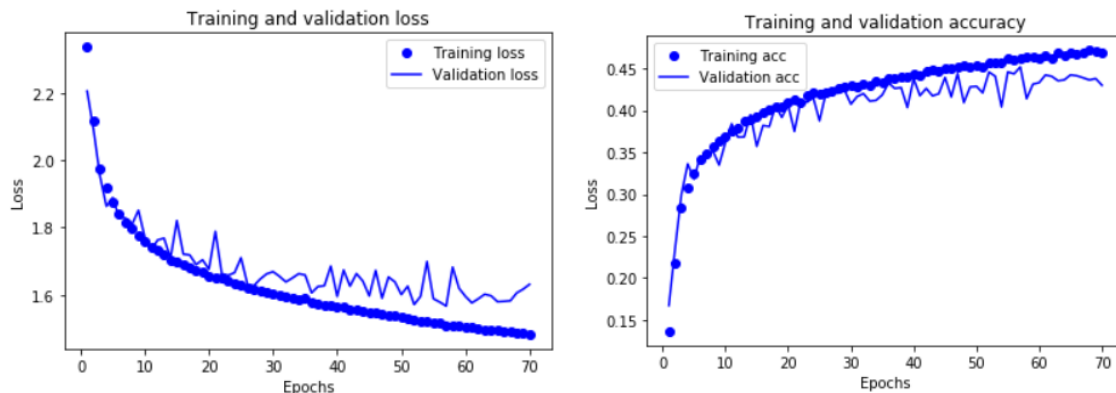
```
10000/10000 [=====] - 1s 54us/step
10000/10000 [=====] - 1s 54us/step
[1.875460443878174, 0.4101]
[1.8969592922210694, 0.4156]
```

For the 3rd time we added dropouts with values 0.2 and 0.5. As shown in graphs below. The data has lot of noise even though accuracy increased.



```
10000/10000 [=====] - 1s 52us/step
10000/10000 [=====] - 1s 52us/step
[1.5207264930725097, 0.452]
[1.5390491622924805, 0.4497]
```

Eventually we adjusted the model by removing dropout function and reducing the dense function which worked for us and we got proportionate loss and accuracy:



```
10000/10000 [=====] - 0s 48us/step
10000/10000 [=====] - 0s 49us/step
[1.529698920440674, 0.457]
[1.5591874828338623, 0.4508]
```

Conclusion:

To conclude, when we increased our batch size there was degradation in quality of our model but the values were computed faster. Later when we reduced the batch size to very small level, keeping the epochs same, then data started overfitting at the very start of 2-3rd epochs. When we tried fitting our model with very small learning rate, it increased our overfitting rate because training epochs weren't large enough for the small learning rate. Due to which we did not receive optimal solution. When we increased our learning rate with large value the graph took a curve and did not reach till the optimal solution. So finally, we took a smaller learning rate and increased our epoch size to get the optimal output.