**Assignment – 1**

**PART – A – MLP Models Report for Cifar-10 dataset**

**MLP Model -1**

**Parameters and Hyper-Parameters**

**Batch Size:** 128

**Epochs:** 40

**Number of Neurons:** 512

**Hidden Layer Activation Function:** Relu

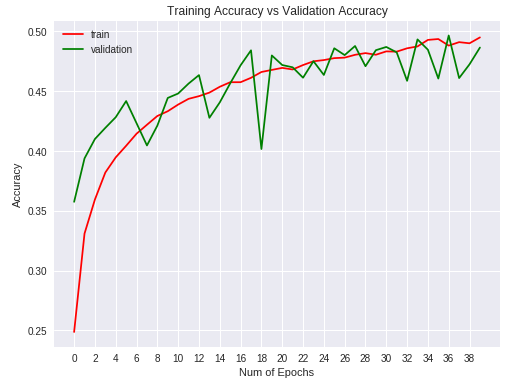
**Output Layer Activation Function:** Softmax

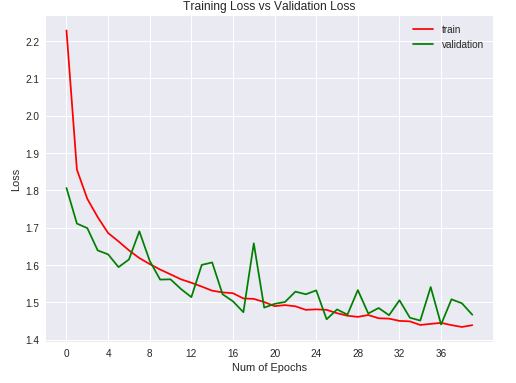
**Loss Function:** Categorical\_Crossentropy

**Optimizer:** RMSprop

**Dropout:** 0.2

**Plots**





**Conclusion:**

**Accuracy:** 48%

**Loss:** 1.466

The base model was tested out with Epoch = 20. The remaining parameters were set same. The accuracy obtained was 45% and the loss was 1.538.

Just increasing the Epoch value, we could obtain an increase of 3% in accuracy and loss reduced mariginally.

**MLP Model -2**

**Parameters and Hyper-Parameters**

**Batch Size:** 128

**Epochs:** 50

**Number of Neurons:** 512

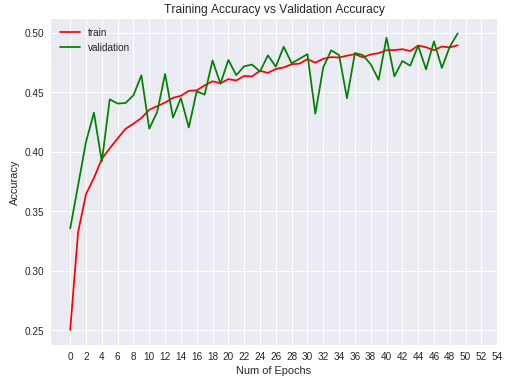
**Hidden Layer Activation Function:** Relu

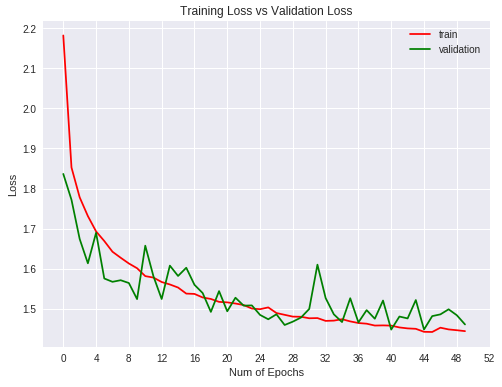
**Output Layer Activation Function:** Softmax

**Loss Function:** Categorical\_Crossentropy

**Optimizer:** RMSprop

**Dropout:** 0.2





**Conclusion:**

**Accuracy:** 49.95%

**Loss:** 1.461

From the earlier model increasing the epochs from 40 to 50, increased the accuracy by almost 1.5% which is significantly a real good sign. There was no much reduction on the loss.

So, this predicts increase in epochs is directly proportional to increase in accuracy to a limit.

**MLP Model -3**

**Parameters and Hyper-Parameters**

**Batch Size:** 256

**Epochs:** 40

**Number of Neurons:** 512

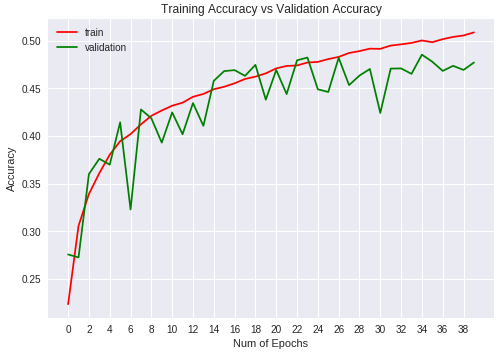
**Hidden Layer Activation Function:** Relu

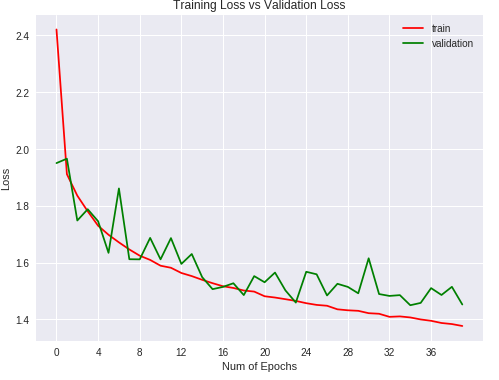
**Output Layer Activation Function:** Softmax

**Loss Function:** Categorical\_Crossentropy

**Optimizer:** RMSprop

**Dropout:** 0.2





**Conclusion:**

**Accuracy:** 47.7%

**Loss:** 1.452

Doubling the batch size had affected the accuracy. The model is also tending to be overfitting. From the previous model the accuracy has declined with no significant reduction in losses as well.

The only thing to note is the reduction in processing time. Earlier each epoch was running for a longer time. Increase in batch size had reduced the run time.

**MLP Model -4**

**Parameters and Hyper-Parameters**

**Batch Size:** 200

**Epochs:** 80

**Number of Neurons:** 512

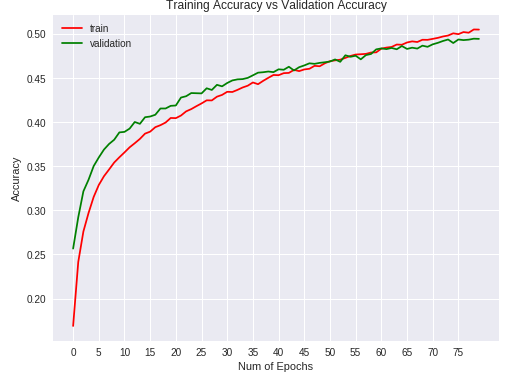
**Hidden Layer Activation Function:** Relu

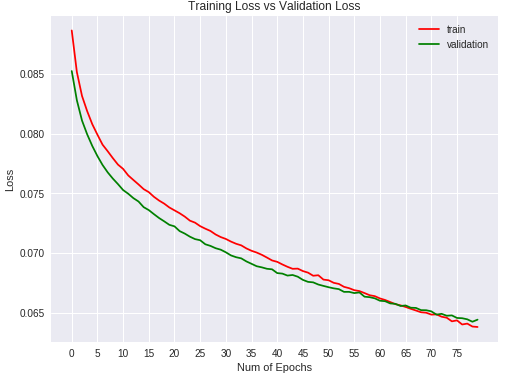
**Output Layer Activation Function:** Softmax

**Loss Function:** Mean Squared Error

**Optimizer:** Stochastic Gradient Descent

**Dropout:** 0.2





**Conclusion:**

**Accuracy:** 49.4%

**Loss:** 0.064

This model had a great performance. Changing the loss function from RMSprop to SGD made a steep reduction in loss. Also, higher epoch count with bigger batch size also reduced the processing time drastically. Just took 14 secs per Epoch.

**MLP Model -5**

**Parameters and Hyper-Parameters**

**Batch Size:** 200

**Epochs:** 100

**Number of Neurons:** 640

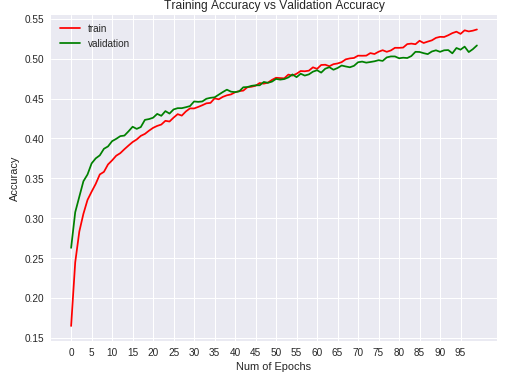
**Hidden Layer Activation Function:** Relu

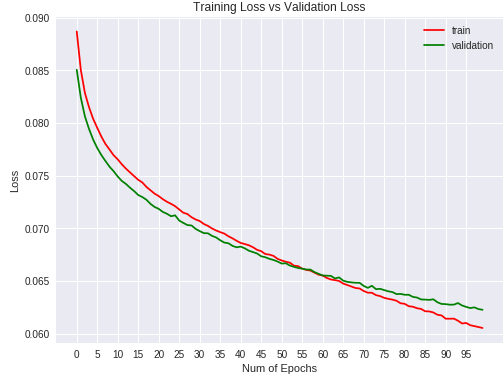
**Output Layer Activation Function:** Softmax

**Loss Function:** Mean Squared Error

**Optimizer:** Stochastic Gradient Descent

**Dropout:** 0.2





**Conclusion:**

**Accuracy:** 51.6%

**Loss:** 0.062

In this model the count of neurons was increased. This had a significant increase in accuracy. The loss remained the same compared to the previous model.

**MLP Model -6**

**Parameters and Hyper-Parameters**

**Batch Size:** 200

**Epochs:** 200

**Number of Neurons:** 640

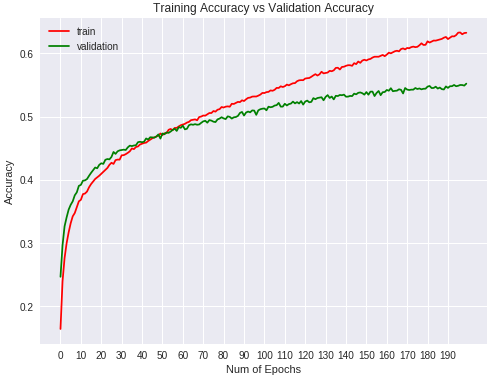
**Hidden Layer Activation Function:** Relu

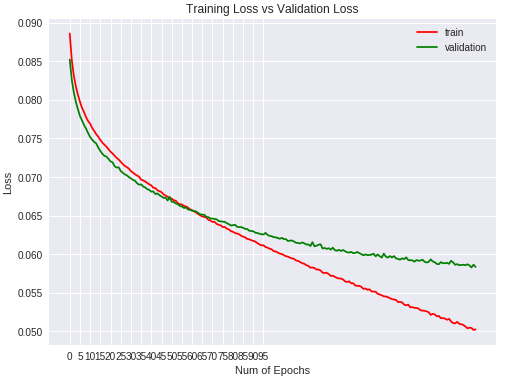
**Output Layer Activation Function:** Softmax

**Loss Function:** Mean Squared Error

**Optimizer:** Stochastic Gradient Descent

**Dropout:** 0.2





**Conclusion:**

**Accuracy:** 55.2%

**Loss:** 0.058

This model was trained with a very large epoch value expecting for high accuracy, but this model started overfitting. So, this predicts that very large epochs will start becoming overfit at some point.

**MLP Model -7**

**Parameters and Hyper-Parameters**

**Batch Size:** 200

**Epochs:** 90

**Number of Neurons:** 640

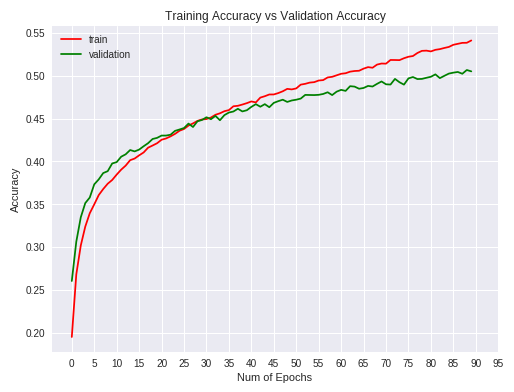
**Hidden Layer Activation Function:** LeakyRelu

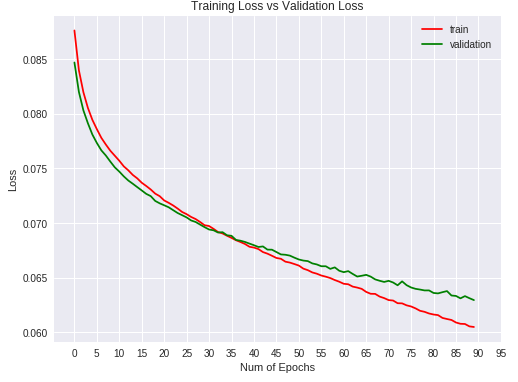
**Output Layer Activation Function:** Softmax

**Loss Function:** Mean Squared Error

**Optimizer:** Stochastic Gradient Descent

**Dropout:** 0.1





**Conclusion:**

**Accuracy:** 50%

**Loss:** 0.06

In this model the activation function was changed to LeakyRelu and drop down was reduced to 0.1. Reducing the drop down started making the model slightly overfitting.

There was no big difference in the accuracy or the loss.

**MLP Model -8**

**Parameters and Hyper-Parameters**

**Batch Size:** 32

**Epochs:** 60

**Number of Neurons:** 640

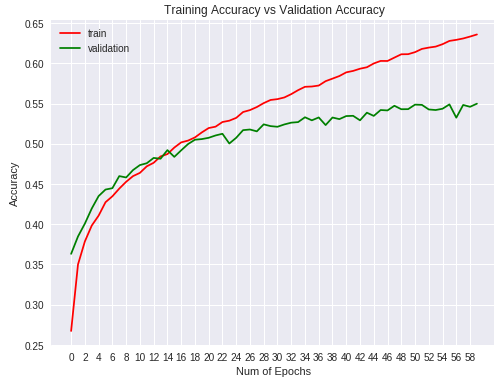
**Hidden Layer Activation Function:** Relu

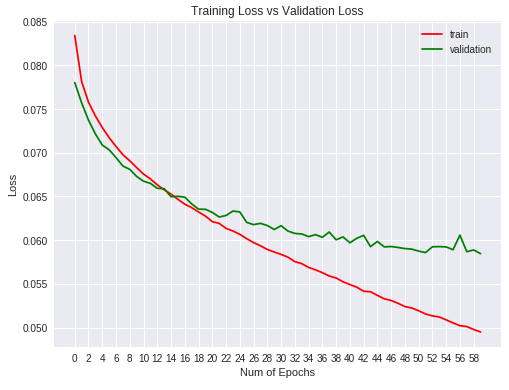
**Output Layer Activation Function:** Softmax

**Loss Function:** Mean Squared Error

**Optimizer:** Stochastic Gradient Descent

**Dropout:** 0.2





**Conclusion:**

**Accuracy:** 54.98%

**Loss:** 0.058

This model was executed with very small batch size. This was giving initially very good results but later it started overfitting.

And also, the execution time was also impacted due to smaller batch size.

**MLP Model -9**

**Parameters and Hyper-Parameters**

**Batch Size:** 32

**Epochs:** 40

**Number of Neurons:** 640

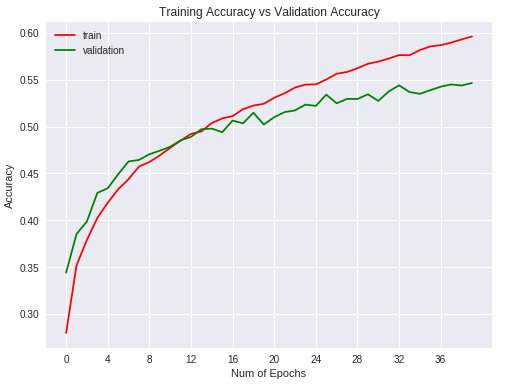
**Hidden Layer Activation Function:** Relu

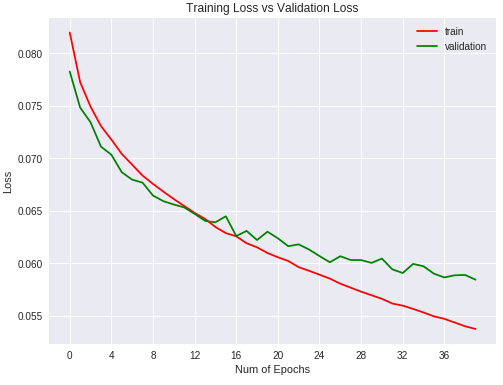
**Output Layer Activation Function:** Softmax

**Loss Function:** Mean Squared Error

**Optimizer:** Adamax

**Dropout:** 0.2





**Conclusion:**

**Accuracy:** 54.64%

**Loss:** 0.058

This model used Adamax optimizer. There was good accuracy increase from other models. The loss also was minimum.

**Final Prediction in MLP Models:**

**Batch Size:** 32

**Epochs:** 40

**Number of Neurons:** 640

**Hidden Layer Activation Function:** Relu

**Output Layer Activation Function:** Softmax

**Loss Function:** Mean Squared Error

**Optimizer:** Adamax

**Dropout:** 0.2

**Accuracy:** 54.64%

**Loss:** 0.058

This MLP model was the best among all other models.

Compared with other models this model had a good accuracy and low loss.

The execution time is also minimal.