Multilayer Perceptron Using the CIFAR-10 Dataset

Dhruv Patel, Fiona Lobo

1. Introduction:

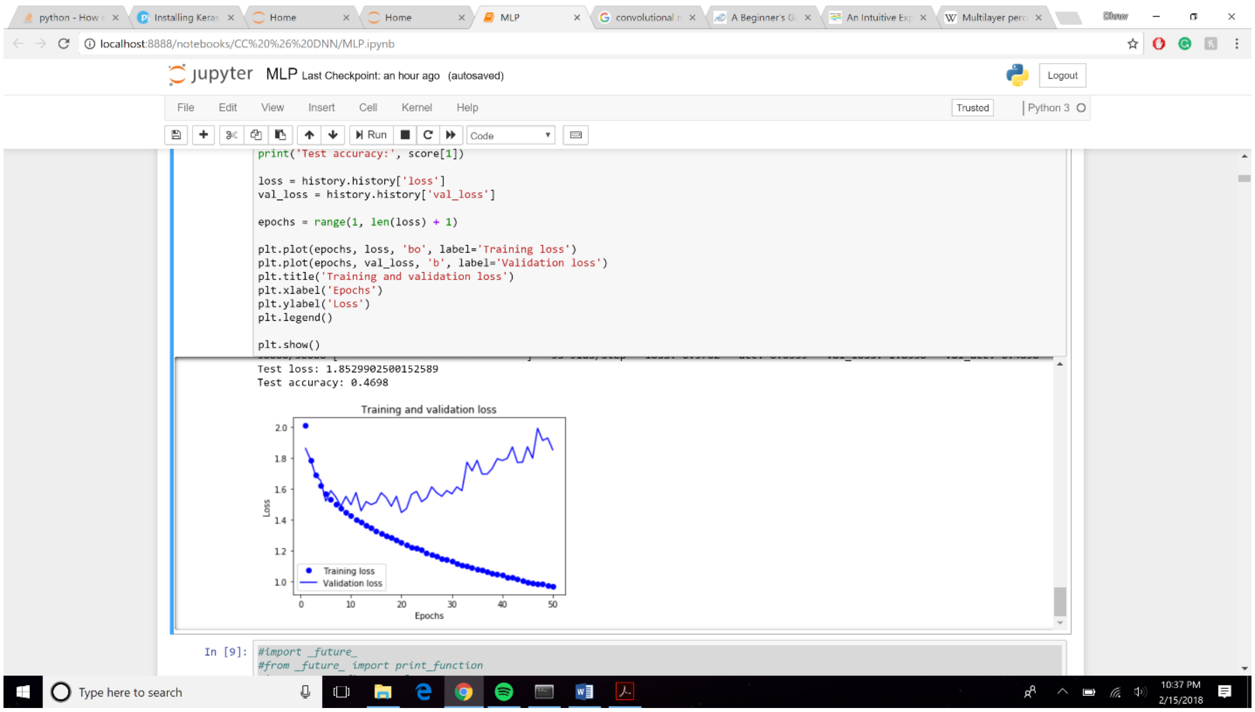
A multilayer perceptron is the basic kind of a deep neural network an here we have altered various parameters of the CIFAR-10 dataset to come up with our best model by providing different observations and provided further suggestions on how the accuracy of the model can be increased.

Observations:

Here we have note down 5 observations that we observed as a result of various permutations and combinations done on the parameters of the base code of the mlp.

Observation 1:

Number of epochs: 50
Batch size: 128
Network Configuration
  a. Number of neurons in layer: Layer 1 – 128, Layer 2 – 256, Layer 3 - 256
  b. Number of layers: 3
Learning rate:
Activation function: ReLU function
Dropout rates: NA



Here we notice the model is over-fitting as there is too much of validation loss. As the training loss decreases the validation loss increases with the number of epochs.

Observation 2:

Number of epochs: 30

Batch size: 64

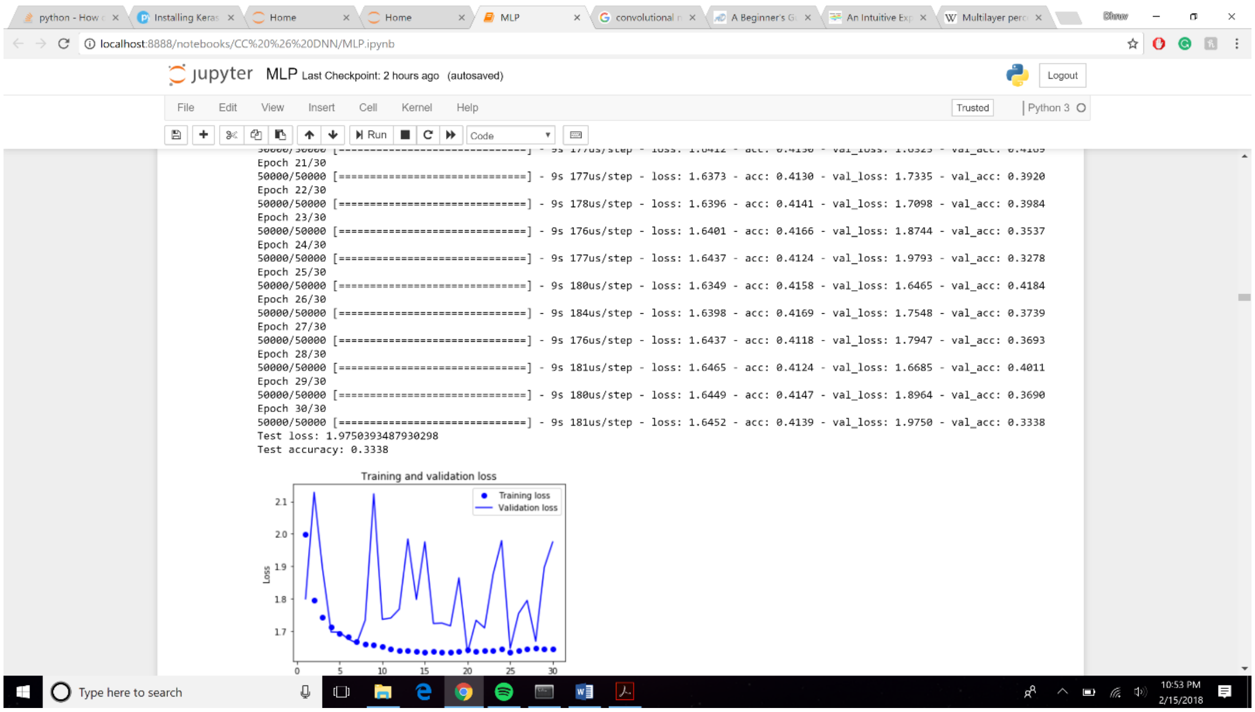
Network Configuration

a. Number of neurons in layer: Layer 1 – 128, Layer 2 –1024, Layer 3 - 1024
b. Number of layers: 3

Learning rate: NA

Activation function: ReLU function

Dropout rates: 0.2



Here we notice that there is a rapid increase and decrease in the validation loss while a steady training loss is observed.

Observation 3:

Number of epochs: 30

Batch size: 512

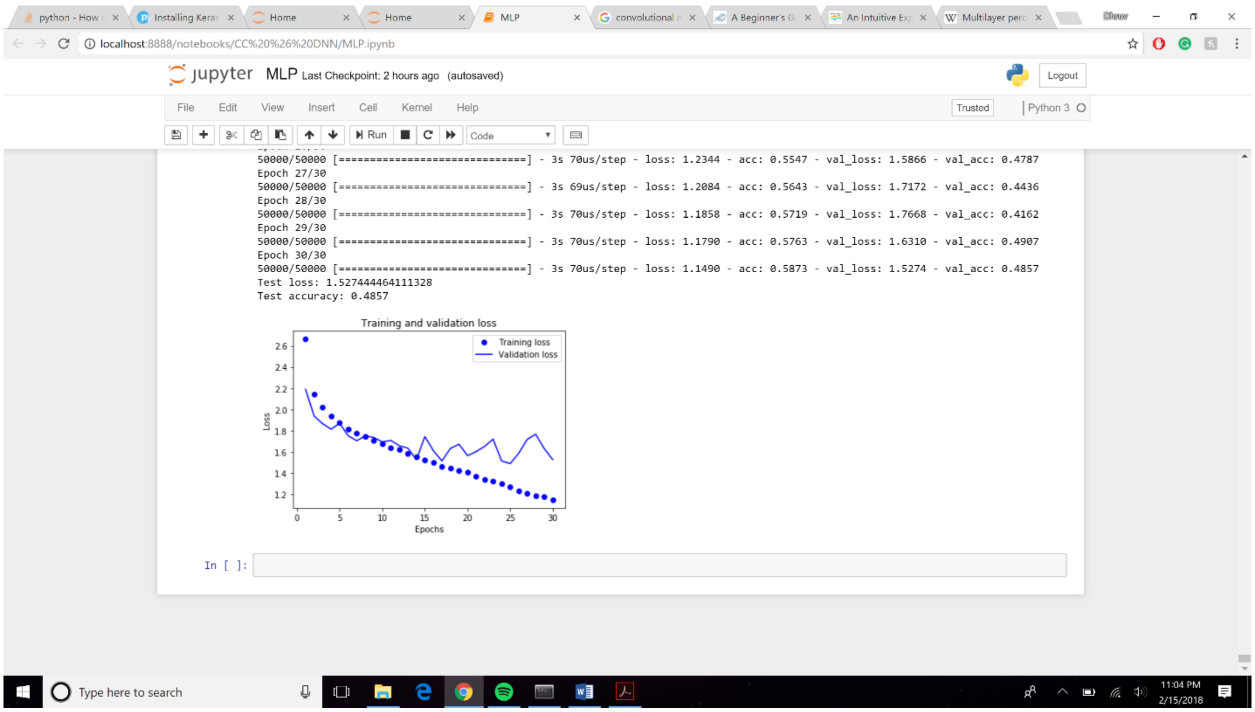
Network Configuration

a. Number of neurons in layer: Layer 1 – 512, Layer 2 –1024, Layer 3 – 1024, Layer 4 - 1024
b. Number of layers: 4

Learning rate: NA

Activation function: ReLU function

Dropout rates: 0.25



Here in this observation we can see three characteristics. Along the graph the validation is less than the training loss, after a few epochs it becomes equal to the training loss and after another few epochs, we observe over fitting in the model.

Observations 4:

Number of epochs: 30

Batch size: 512

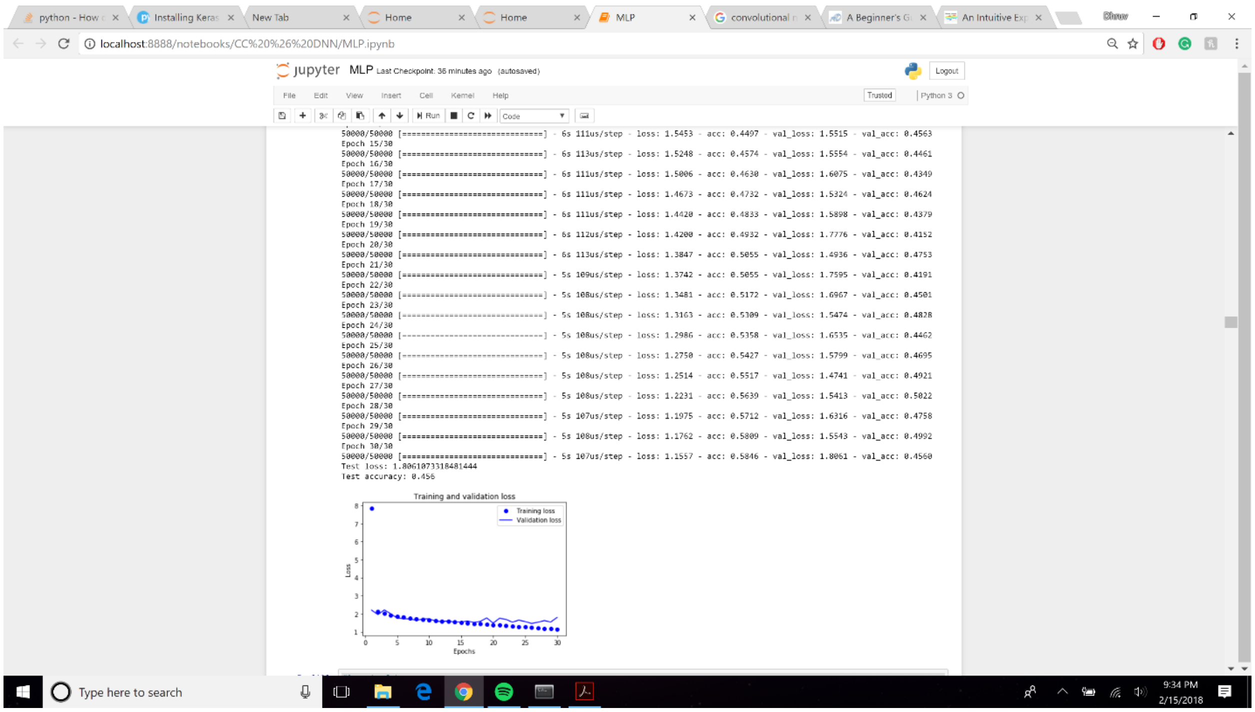
Network Configuration

a. Number of neurons in layer: Layer 1 –1024, Layer 2 –2048, Layer 3 – 2048,
b. Number of layers: 4

Learning rate: NA

Activation function: ReLU function

Dropout rates: 0.3



Here we observed the best training accuracy among all the observations. The graph of training loss and validation loss also is about the same. The inner layers have a lot of neurons providing a better learning mechanism to the model.

The above model is the best model we could come up with. It has an accuracy of 58.46% with a test accuracy of 45.60%. Our model is pretty good for the number of epochs we have chosen. If we increase the number of epochs it is a high probability that our model if overfit because it can be seen in the graph that the last few epochs, there is a spike in the validation loss which can continue to increase if it allowed run for more epochs.

If we can find a way to increase the density of the inner layers without letting the model overfit then we can come up with a better accuracy.

Conclusion:

After trying different combinations of parameters, we come up with the conclusion that increasing the density of the hidden layers, that is, increasing the number of neurons in the hidden layers increases the accuracy of the training data but it starts to overfit after a particular number of epochs.

Hence a better model of an MLP is possible if we find a way in which the data cannot be overfitted and the accuracy increases with the number if epochs.