**MLP – MNIST Dataset**

**Report details:**

* The MNIST dataset contains 6000 images each of 28\*28 pixels and labels assigned to each image
* The data is split into training and validation set.
* Various Neural network configuration are done to train and test the network.
* Changes in number of epochs, layers, activation functions, optimization techniques and learning rates are examined to understand the effect and importance of each hyperparameter for training the network.
* Relu and tanh activation is used in both the layers and a sigmoid activation in the last output layer.
* Optimization technique such as Adam and RMSprop are used for back propagation technique for reducing gradient descent.
* Various networks along with hyper parameter tuning are explained in detail in the notebook along with their results in excel sheet.
* The best network after the hyperparameter tuning is the model with less testing loss and high accuracy resulting in less overfitting of data while training.

**Model:**

Different models are created by varying layers, epochs, batch size, activation function, optimizers and the best model with high testing accuracy is picked up.

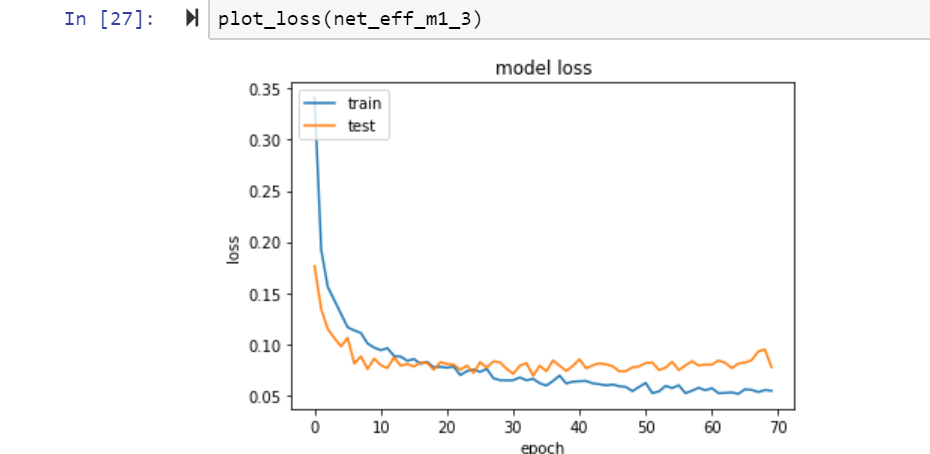
The excel sheet illustrates the details of each model.

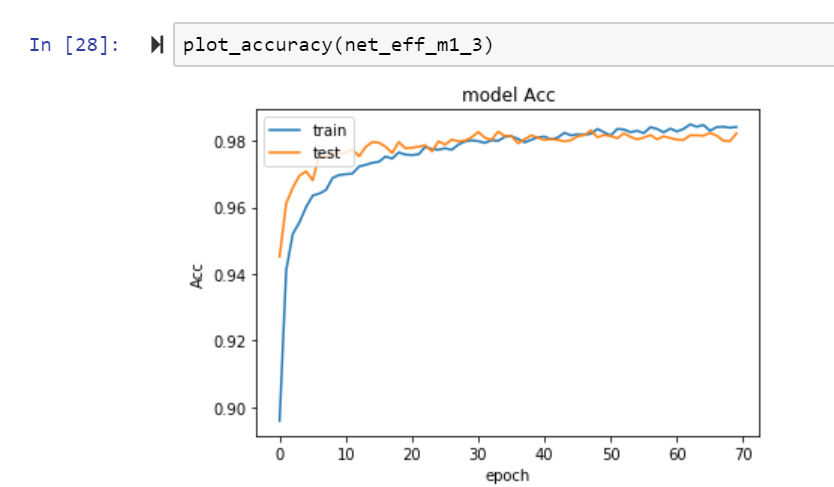


**Best models:**

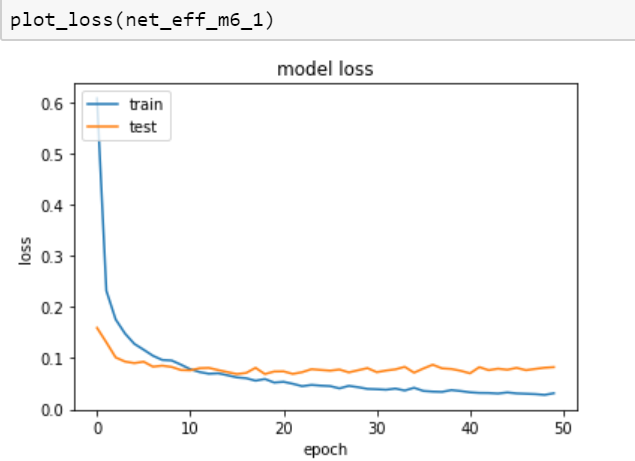
Network Architecture:

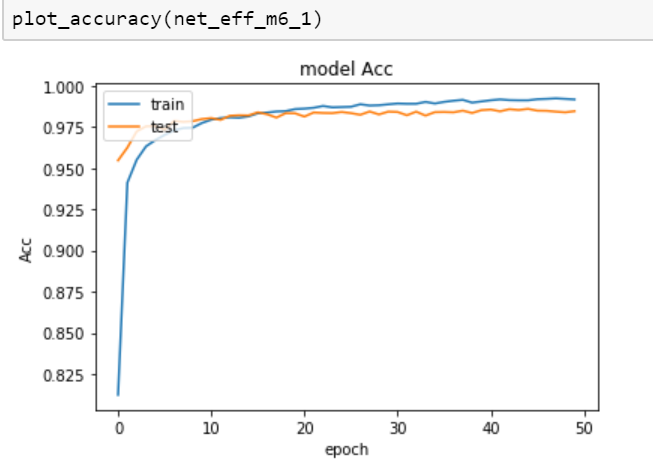
|  |  |
| --- | --- |
| Models network\_effect1\_3 |  |
| Layer 1 - 512 - dropout - 0.4 ,tanh Layer 2 - 512 Dropout-0.4,tanh Optimizer - Adam - lr - 0.002 |  |
| Batch size | 128 |
| Epochs | 70 |
| Training loss | 0.0544 |
| Testing loss | 0.0775 |
| Training Accuracy | 0.9841 |
| Testing Accuracy | 0.9822 |





|  |  |
| --- | --- |
| network\_effect5\_1 |  |
|  |  |
| Layer 1 - 512 - dropout - 0.4 ,relu Layer 2 - 256 Dropout-0.4,relu Layer3 - 128 dropout - 0.4,relu Layer4 - 64 dropout - 0.4,relu Optimizer - Adam - lr - 0.001 |  |
|  |
| Batch size | 128 |
| Epochs | 50 |
| Training loss | 0.0315 |
| Testing loss | 0.0825 |
| Training Accuracy | 0.9916 |
| Testing Accuracy | 0.9845 |





**Conclusions:**

|  |
| --- |
| From the 2 models, I choose Models network\_effect1\_3 to be the best model,  since it has a smaller number of layers reducing computation time and  cost as compared to the other model |

**Observation:**

Increasing the epochs sometimes causes the network to overfit

Small batch size causes the gradient update in small steps and results in better accuracy at the early epochs.