COMPARISON OF THE PERFORMANCE OF TWO MULTI-LAYER PERCEPTRONS IN MNIST DATASET

Submitted by :

Allen Y
S5 CSE
Roll no: 11
TVE19CS011

SOURCE CODE

Github repository link
https://github.com/Alleny244/Allen.Y_NNDL1

Link to google Colab Notebook
https://colab.research.google.com/drive/1i9dfifGz6yrHi7_un_PhEaeK6Gb4Qhd5#scrollTo=e0Oz8t_gCQAf

I trained the MNIST dataset in two models. The model summary of the two models is as shown in the images below

Model 1

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 flatten_1 (Flatten)         (None, 784)               0

 dense_2 (Dense)             (None, 64)                50240

 dense_3 (Dense)             (None, 10)                650

=================================================================
Total params: 50,890
Trainable params: 50,890
Non-trainable params: 0
_________________________________________________________________
```

Model 2

```
Model: "sequential_10"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 flatten_10 (Flatten)         (None, 784)               0

 dense_40 (Dense)             (None, 1024)              803840

 dense_41 (Dense)             (None, 512)               524800

 dense_42 (Dense)             (None, 256)               131328

 dense_43 (Dense)             (None, 128)               32896

 dense_44 (Dense)             (None, 10)                1290

=================================================================
Total params: 1,494,154
Trainable params: 1,494,154
Non-trainable params: 0
_________________________________________________________________
```

A comparison on both the model training is tabulated below:

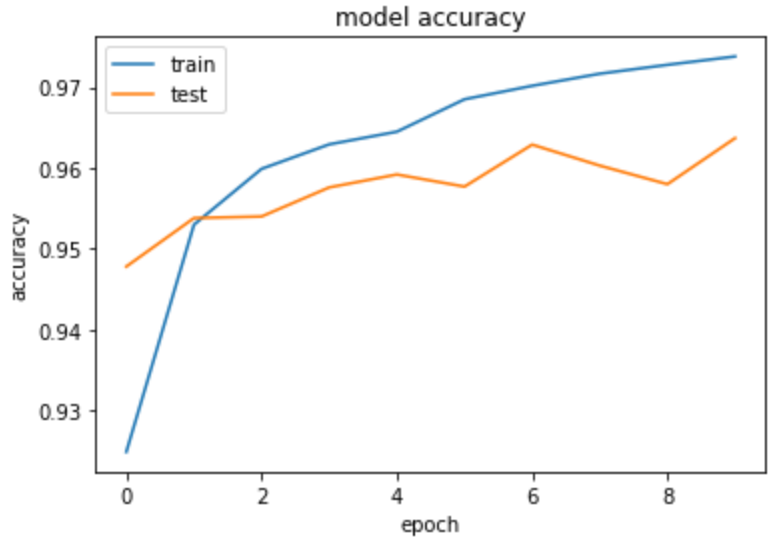
Criteria	Model 1	Model 2
# of Hidden layers	3	6
# of parameters	50,890	1,494,154
Total # of neurons	74	1930

Both the models were trained with same Loss function (cross_entropy), same optimization function(Adam) and the same number of epochs(ten).

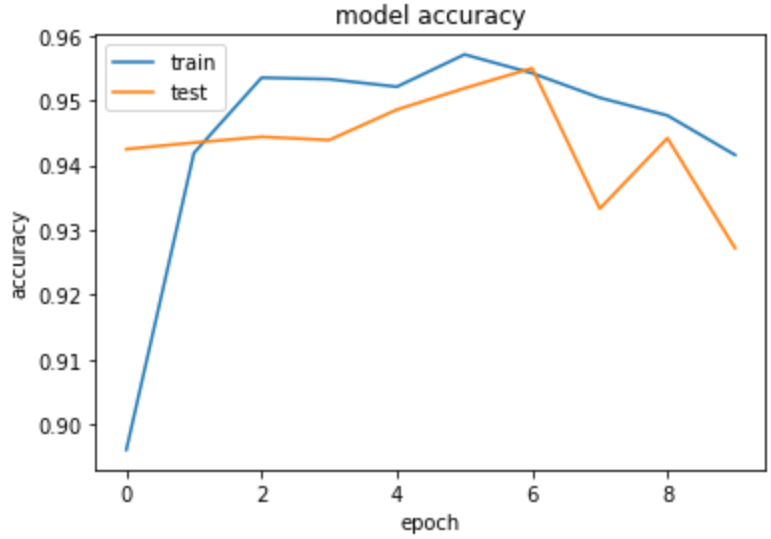
Validation Accuracy of Model 1 = 96.37%
Validation Accuracy of Model 2 =92.72%

Accuracy vs Epoch Plots

Model 1



Model 2



Inference

Clearly the second model is a more complex model compared to the first one as it contains more hidden layers, more neurons, and hence more trainable parameters. From the plots, we can infer that the second model is getting overfitted towards the end epochs. Thus Model 1 shows slightly less accuracy than Model 2. The second model is getting overfitted because it is even learning the noise in the training samples and is losing its generalization ability.

It is also observed that there was a significant difference in the time, required to train these models.
Model 1 took around 5 seconds for each epoch, whereas Model 2 took around 38 seconds for the same. This can be explained by the more amount of trainable parameters in Model 2 as compared to Model 1.