Project Report

Aditya Prakash axp171931

Megan Yociss mxy122430

Optical Character Recognition over MNIST dataset using Optimized Neural Network

<u>Note</u>: We have changed the project topic.

(1) <u>Introduction</u>

We are dealing with Optical Character Recognition which is a relatively common application of Machine Learning. Here a model is trained on MNIST dataset which consists of 10 digits from 0 to 9. Then we attempt to classify the digits into classes and test the accuracy of our trained model.

(2) <u>Problem Definition and Algorithm</u>

(a) <u>Task Definition</u>

We want to achieve the best possible training accuracy for Optical Character Recognition over MNIST dataset. We are using Keras Backend and a Neural Network to accomplish the task. First, we need to train the system over the training data. Then we do validation testing and then we test our model over the test set.

Then we plot 2 graphs to evaluate our model. First – Accuracy vs Epoch, Second – Loss vs Epoch.

(b) <u>Algorithm Definition</u>

We define Batch Size, Number of classes as 10(0-9), number of epochs, size of image Rows and Columns (28 X 28).

We split the training and testing data and then reshape the data to fit the input shape to be given to the first layer.

We specify the input image as a linear vector and normalize its value for efficient classification. The class vector is specified as a vector with length 10 due to Keras requirements.

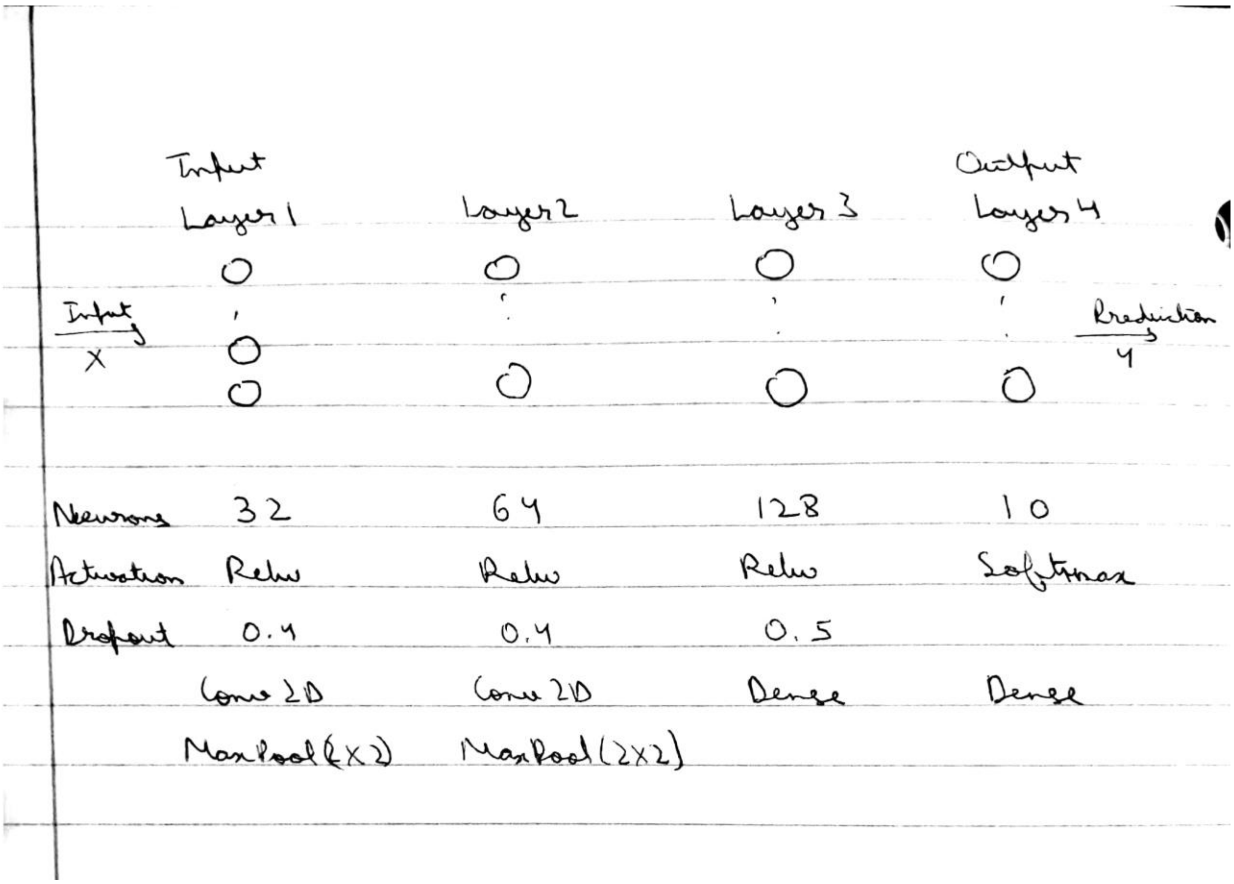
The model specified is sequential with four layers.

Layer 1: Input layer, Convolutional, Hidden, gets input shape, Max Pooling, Dropout, Relu Activation

Layer 2: Convolutional, Hidden, Max Pooling, Dropout, Relu Activation

Layer 3: Dense, Flattened, Dropout, Relu Activation

Layer 4: Output Layer, Dense, Softmax Activation



(3) <u>Experimental Evaluation</u>

(a) <u>Methodology</u>

As mentioned in [5], there are multiple models to achieve this task. The best possible error rates for these models are:

Classifier	Best Error Rate %(Test)
Linear	7.6
KNN(L2)	1.8
Boosted Trees	1.53
Quadratic	3.3
SVM	1.1
2 Layer NN	3.8
35 Convolution Nets	**0.23**

Our model achieves:

Test Loss = 0.01782 = 1.782%

Test Accuracy = 0.9947 = 99.47%

(b) <u>Results</u>

We have created cases where some parameters of our model have been changed to find out the optimal model.

The default value of parameters set is:

Batch Size = 128

Number of Classes = 10

Image rows, Image Columns = 28 X 28

<u>Layer 1 : Input</u>

Neurons 32, Conv2D, Kernel Size 5 X 5, Activation = Relu, Pool Size 2 X 2, Dropout 0.4

<u>Layer 2</u>

Neurons 64, Conv2D, Kernel Size 5 X 5, Activation = Relu, Pool Size 2 X 2, Dropout 0.4

<u>Layer 3</u>

Neurons 128, Dense, Activation = Relu, Dropout 0.5

<u>Layer 4 : Output</u>

Neurons 10, Dense, Activation = Softmax

In all the cases, only one parameter is changed, the rest remain same.

Cases:

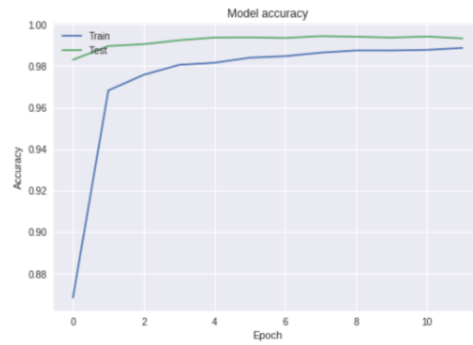
1. Add another convolutional layer

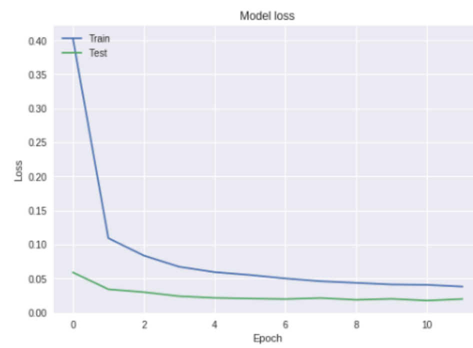
Total Time: 143 seconds

Test Loss: 1.991%

Test Accuracy: 99.33%

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 13s - loss: 0.4025 - acc: 0.8684 - val_loss: 0.0590 - val_acc: 0.9830
Epoch 2/12
 - 12s - loss: 0.1093 - acc: 0.9682 - val_loss: 0.0342 - val_acc: 0.9896
Epoch 3/12
 - 12s - loss: 0.0838 - acc: 0.9758 - val_loss: 0.0301 - val_acc: 0.9905
Epoch 4/12
 - 12s - loss: 0.0673 - acc: 0.9806 - val_loss: 0.0240 - val_acc: 0.9924
Epoch 5/12
 - 12s - loss: 0.0594 - acc: 0.9815 - val_loss: 0.0216 - val_acc: 0.9937
Epoch 6/12
 - 12s - loss: 0.0553 - acc: 0.9840 - val_loss: 0.0206 - val_acc: 0.9938
Epoch 7/12
 - 12s - loss: 0.0501 - acc: 0.9847 - val_loss: 0.0198 - val_acc: 0.9935
Epoch 8/12
 - 12s - loss: 0.0460 - acc: 0.9864 - val_loss: 0.0213 - val_acc: 0.9944
Epoch 9/12
 - 12s - loss: 0.0436 - acc: 0.9875 - val_loss: 0.0187 - val_acc: 0.9941
Epoch 10/12
 - 12s - loss: 0.0413 - acc: 0.9875 - val_loss: 0.0201 - val_acc: 0.9937
Epoch 11/12
 - 11s - loss: 0.0407 - acc: 0.9877 - val_loss: 0.0176 - val_acc: 0.9942
Epoch 12/12
 - 11s - loss: 0.0382 - acc: 0.9887 - val_loss: 0.0199 - val_acc: 0.9933
Test loss: 0.01991353547919498
Test accuracy: 0.9933
```





2. Change in Batch Size

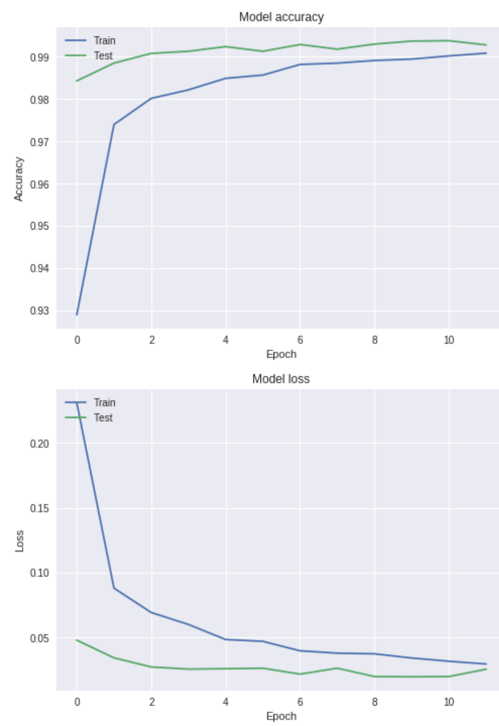
a. 64

Total Time: 174 seconds

Test Loss: 2.5546%

Test Accuracy: 99.28%

```
Using TensorFlow backend.
Downloading data from https://s3.amazonaws.com/img-datasets/mnist.npz
11493376/11490434 [==============================] - 1s 0us/step
11501568/11490434 [==============================] - 1s 0us/step
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 17s - loss: 0.2310 - acc: 0.9290 - val_loss: 0.0479 - val_acc: 0.9843
Epoch 2/12
 - 15s - loss: 0.0880 - acc: 0.9740 - val_loss: 0.0343 - val_acc: 0.9885
Epoch 3/12
 - 15s - loss: 0.0693 - acc: 0.9802 - val_loss: 0.0273 - val_acc: 0.9908
Epoch 4/12
 - 14s - loss: 0.0600 - acc: 0.9822 - val_loss: 0.0256 - val_acc: 0.9913
Epoch 5/12
 - 14s - loss: 0.0484 - acc: 0.9849 - val_loss: 0.0259 - val_acc: 0.9924
Epoch 6/12
 - 14s - loss: 0.0470 - acc: 0.9857 - val_loss: 0.0263 - val_acc: 0.9913
Epoch 7/12
 - 14s - loss: 0.0397 - acc: 0.9882 - val_loss: 0.0218 - val_acc: 0.9929
Epoch 8/12
 - 14s - loss: 0.0379 - acc: 0.9885 - val_loss: 0.0263 - val_acc: 0.9918
Epoch 9/12
 - 14s - loss: 0.0374 - acc: 0.9891 - val_loss: 0.0199 - val_acc: 0.9930
Epoch 10/12
 - 15s - loss: 0.0342 - acc: 0.9895 - val_loss: 0.0197 - val_acc: 0.9937
Epoch 11/12
 - 14s - loss: 0.0317 - acc: 0.9902 - val_loss: 0.0199 - val_acc: 0.9938
Epoch 12/12
 - 14s - loss: 0.0295 - acc: 0.9908 - val_loss: 0.0255 - val_acc: 0.9928
Test loss: 0.025546745175798604
Test accuracy: 0.9928
```

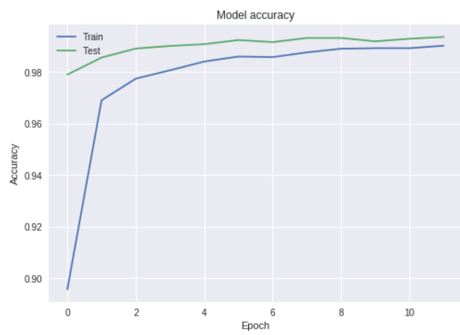


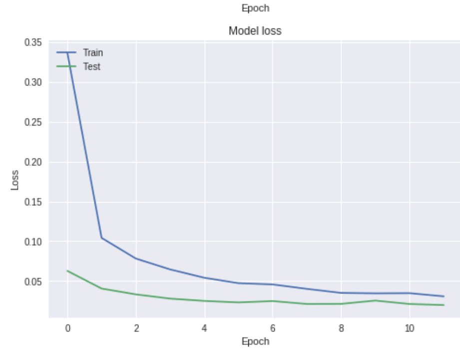
b. 256

Total Time: 85 seconds

Test Loss: 1.9904%

Test Accuracy: 99.36%





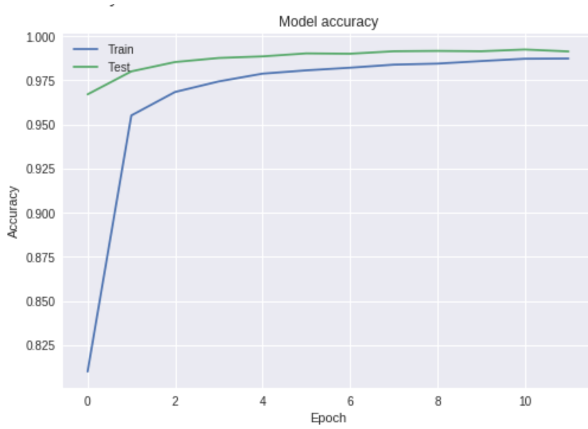

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 8s - loss: 0.3366 - acc: 0.8957 - val_loss: 0.0628 - val_acc: 0.9790
Epoch 2/12
 - 7s - loss: 0.1042 - acc: 0.9691 - val_loss: 0.0406 - val_acc: 0.9856
Epoch 3/12
 - 7s - loss: 0.0783 - acc: 0.9774 - val_loss: 0.0333 - val_acc: 0.9891
Epoch 4/12
 - 7s - loss: 0.0647 - acc: 0.9807 - val_loss: 0.0280 - val_acc: 0.9901
Epoch 5/12
 - 7s - loss: 0.0542 - acc: 0.9841 - val_loss: 0.0251 - val_acc: 0.9908
Epoch 6/12
 - 7s - loss: 0.0474 - acc: 0.9860 - val_loss: 0.0232 - val_acc: 0.9924
Epoch 7/12
 - 7s - loss: 0.0458 - acc: 0.9858 - val_loss: 0.0249 - val_acc: 0.9916
Epoch 8/12
 - 7s - loss: 0.0402 - acc: 0.9876 - val_loss: 0.0213 - val_acc: 0.9932
Epoch 9/12
 - 7s - loss: 0.0352 - acc: 0.9890 - val_loss: 0.0214 - val_acc: 0.9932
Epoch 10/12
 - 7s - loss: 0.0346 - acc: 0.9892 - val_loss: 0.0256 - val_acc: 0.9919
Epoch 11/12
 - 7s - loss: 0.0348 - acc: 0.9892 - val_loss: 0.0213 - val_acc: 0.9929
Epoch 12/12
 - 7s - loss: 0.0309 - acc: 0.9902 - val_loss: 0.0199 - val_acc: 0.9936
Test loss: 0.01990412887528437
Test accuracy: 0.9936
```

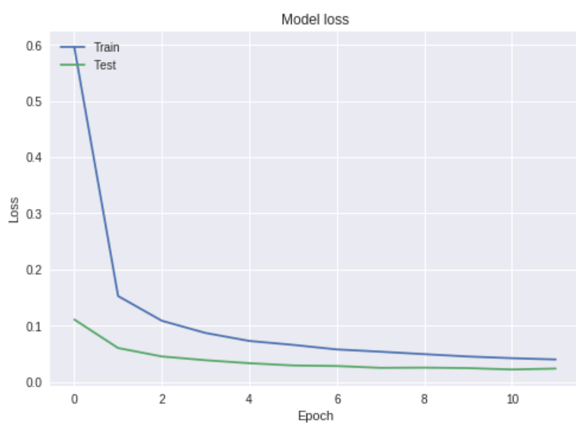
c. 1024

Total Time: 73 seconds

Test Loss: 2.3634%

Test Accuracy: 99.14%






```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 7s - loss: 0.5953 - acc: 0.8100 - val_loss: 0.1109 - val_acc: 0.9671
Epoch 2/12
 - 6s - loss: 0.1527 - acc: 0.9551 - val_loss: 0.0602 - val_acc: 0.9801
Epoch 3/12
 - 6s - loss: 0.1087 - acc: 0.9684 - val_loss: 0.0452 - val_acc: 0.9854
Epoch 4/12
 - 6s - loss: 0.0870 - acc: 0.9744 - val_loss: 0.0385 - val_acc: 0.9877
Epoch 5/12
 - 6s - loss: 0.0728 - acc: 0.9788 - val_loss: 0.0332 - val_acc: 0.9886
Epoch 6/12
 - 6s - loss: 0.0658 - acc: 0.9807 - val_loss: 0.0292 - val_acc: 0.9903
Epoch 7/12
 - 6s - loss: 0.0577 - acc: 0.9822 - val_loss: 0.0282 - val_acc: 0.9901
Epoch 8/12
 - 6s - loss: 0.0537 - acc: 0.9839 - val_loss: 0.0249 - val_acc: 0.9915
Epoch 9/12
 - 6s - loss: 0.0493 - acc: 0.9845 - val_loss: 0.0253 - val_acc: 0.9917
Epoch 10/12
 - 6s - loss: 0.0451 - acc: 0.9860 - val_loss: 0.0245 - val_acc: 0.9915
Epoch 11/12
 - 6s - loss: 0.0421 - acc: 0.9873 - val_loss: 0.0220 - val_acc: 0.9925
Epoch 12/12
 - 6s - loss: 0.0399 - acc: 0.9874 - val_loss: 0.0236 - val_acc: 0.9914
Test loss: 0.023634856985054284
Test accuracy: 0.9914
```

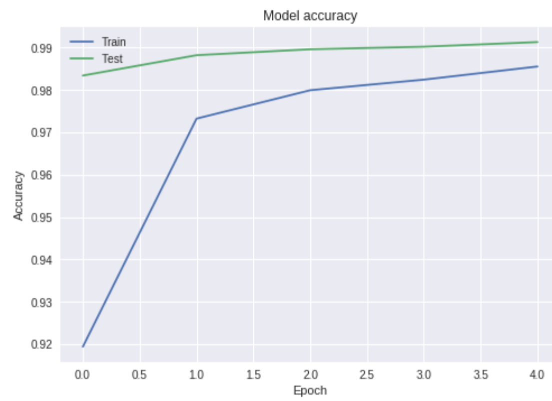
3. Change in Epochs

a. 5

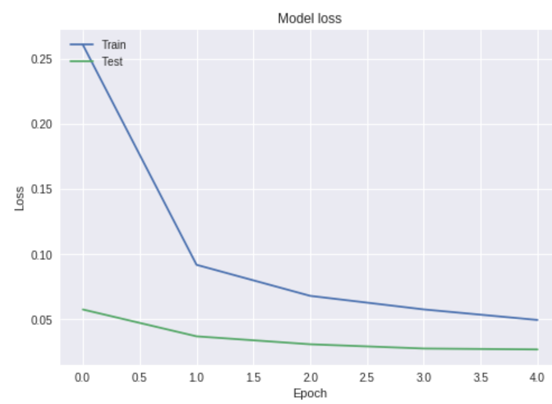
Total Time: 50 seconds

Test Loss: 2.6717%

Test Accuracy: 99.13%




```
Train on 60000 samples, validate on 10000 samples
Epoch 1/5
 - 10s - loss: 0.2609 - acc: 0.9194 - val_loss: 0.0574 - val_acc: 0.9834
Epoch 2/5
 - 10s - loss: 0.0918 - acc: 0.9732 - val_loss: 0.0368 - val_acc: 0.9882
Epoch 3/5
 - 10s - loss: 0.0679 - acc: 0.9799 - val_loss: 0.0307 - val_acc: 0.9896
Epoch 4/5
 - 10s - loss: 0.0575 - acc: 0.9824 - val_loss: 0.0274 - val_acc: 0.9902
Epoch 5/5
 - 10s - loss: 0.0494 - acc: 0.9855 - val_loss: 0.0267 - val_acc: 0.9913
Test loss: 0.02671768427727293
Test accuracy: 0.9913
```

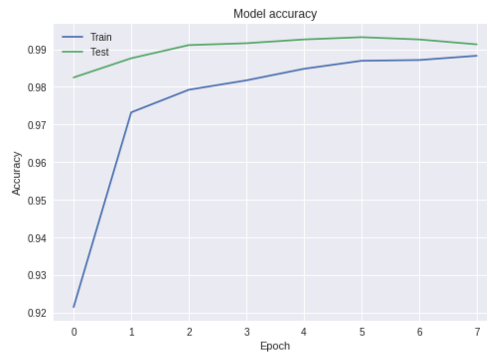


b. 8

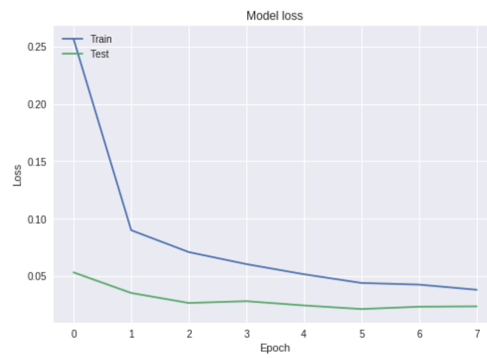
Total Time: 80 seconds

Test Loss: 2.2347%

Test Accuracy: 99.13%




```
Train on 60000 samples, validate on 10000 samples
Epoch 1/8
 - 10s - loss: 0.2566 - acc: 0.9215 - val_loss: 0.0531 - val_acc: 0.9825
Epoch 2/8
 - 10s - loss: 0.0899 - acc: 0.9732 - val_loss: 0.0352 - val_acc: 0.9876
Epoch 3/8
 - 10s - loss: 0.0708 - acc: 0.9792 - val_loss: 0.0264 - val_acc: 0.9911
Epoch 4/8
 - 10s - loss: 0.0604 - acc: 0.9817 - val_loss: 0.0280 - val_acc: 0.9916
Epoch 5/8
 - 10s - loss: 0.0515 - acc: 0.9848 - val_loss: 0.0243 - val_acc: 0.9926
Epoch 6/8
 - 10s - loss: 0.0439 - acc: 0.9869 - val_loss: 0.0211 - val_acc: 0.9932
Epoch 7/8
 - 10s - loss: 0.0424 - acc: 0.9871 - val_loss: 0.0232 - val_acc: 0.9926
Epoch 8/8
 - 10s - loss: 0.0380 - acc: 0.9883 - val_loss: 0.0235 - val_acc: 0.9913
Test loss: 0.023466508609434823
Test accuracy: 0.9913
```



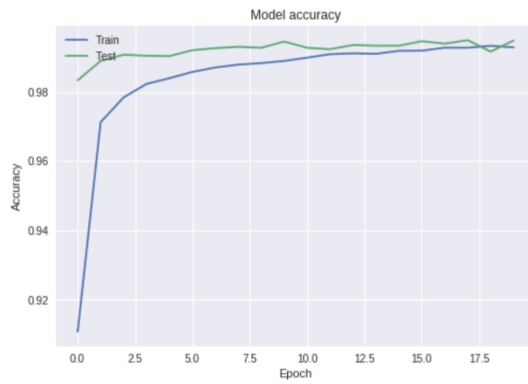
c. 20

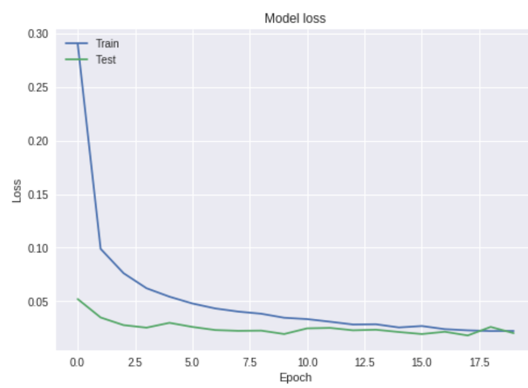
Total Time: 200 seconds

Test Loss: 2.0139%

Test Accuracy: 99.48%

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/20
 - 10s - loss: 0.2906 - acc: 0.9107 - val_loss: 0.0520 - val_acc: 0.9833
Epoch 2/20
 - 10s - loss: 0.0989 - acc: 0.9712 - val_loss: 0.0348 - val_acc: 0.9889
Epoch 3/20
 - 10s - loss: 0.0761 - acc: 0.9784 - val_loss: 0.0276 - val_acc: 0.9907
Epoch 4/20
 - 10s - loss: 0.0621 - acc: 0.9823 - val_loss: 0.0253 - val_acc: 0.9904
Epoch 5/20
 - 10s - loss: 0.0542 - acc: 0.9839 - val_loss: 0.0298 - val_acc: 0.9903
Epoch 6/20
 - 10s - loss: 0.0478 - acc: 0.9857 - val_loss: 0.0260 - val_acc: 0.9920
Epoch 7/20
 - 10s - loss: 0.0432 - acc: 0.9870 - val_loss: 0.0230 - val_acc: 0.9926
Epoch 8/20
 - 10s - loss: 0.0403 - acc: 0.9878 - val_loss: 0.0223 - val_acc: 0.9930
Epoch 9/20
 - 10s - loss: 0.0383 - acc: 0.9883 - val_loss: 0.0225 - val_acc: 0.9927
Epoch 10/20
 - 10s - loss: 0.0345 - acc: 0.9889 - val_loss: 0.0193 - val_acc: 0.9945
Epoch 11/20
 - 10s - loss: 0.0333 - acc: 0.9899 - val_loss: 0.0246 - val_acc: 0.9927
Epoch 12/20
 - 10s - loss: 0.0308 - acc: 0.9909 - val_loss: 0.0250 - val_acc: 0.9923
Epoch 13/20
 - 10s - loss: 0.0282 - acc: 0.9911 - val_loss: 0.0228 - val_acc: 0.9935
Epoch 14/20
 - 10s - loss: 0.0284 - acc: 0.9910 - val_loss: 0.0233 - val_acc: 0.9933
Epoch 15/20
 - 10s - loss: 0.0255 - acc: 0.9918 - val_loss: 0.0212 - val_acc: 0.9933
Epoch 16/20
 - 10s - loss: 0.0268 - acc: 0.9919 - val_loss: 0.0193 - val_acc: 0.9946
Epoch 17/20
 - 10s - loss: 0.0239 - acc: 0.9927 - val_loss: 0.0214 - val_acc: 0.9939
Epoch 18/20
 - 10s - loss: 0.0227 - acc: 0.9927 - val_loss: 0.0180 - val_acc: 0.9949
Epoch 19/20
 - 10s - loss: 0.0221 - acc: 0.9933 - val_loss: 0.0259 - val_acc: 0.9916
Epoch 20/20
 - 10s - loss: 0.0222 - acc: 0.9928 - val_loss: 0.0201 - val_acc: 0.9948
Test loss: 0.020139422235911753
Test accuracy: 0.9948
```





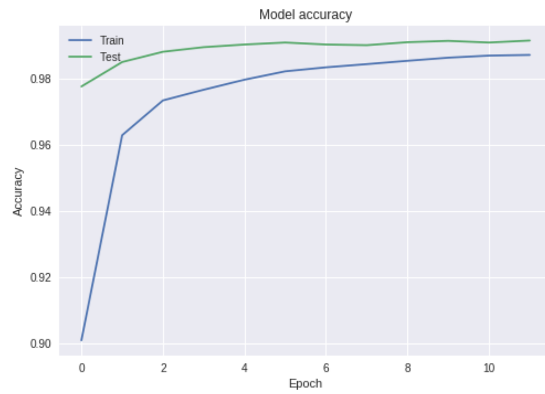
4. Change in Kernel Size

a. 3 X 3

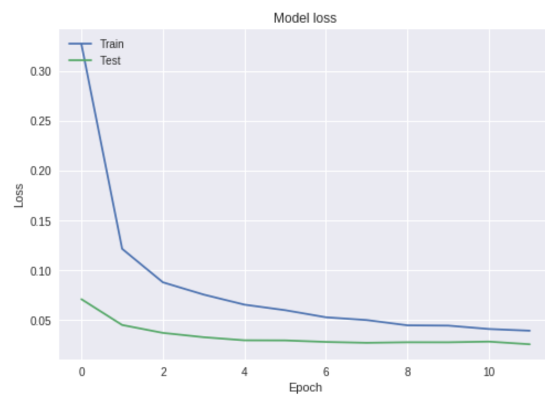
Total Time: 143 seconds

Test Loss: 2.5930%

Test Accuracy: 99.15%




```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 12s - loss: 0.3269 - acc: 0.9008 - val_loss: 0.0711 - val_acc: 0.9776
Epoch 2/12
 - 11s - loss: 0.1216 - acc: 0.9628 - val_loss: 0.0453 - val_acc: 0.9850
Epoch 3/12
 - 11s - loss: 0.0880 - acc: 0.9734 - val_loss: 0.0373 - val_acc: 0.9881
Epoch 4/12
 - 11s - loss: 0.0758 - acc: 0.9766 - val_loss: 0.0330 - val_acc: 0.9895
Epoch 5/12
 - 11s - loss: 0.0657 - acc: 0.9797 - val_loss: 0.0299 - val_acc: 0.9903
Epoch 6/12
 - 11s - loss: 0.0601 - acc: 0.9822 - val_loss: 0.0298 - val_acc: 0.9909
Epoch 7/12
 - 11s - loss: 0.0530 - acc: 0.9834 - val_loss: 0.0283 - val_acc: 0.9903
Epoch 8/12
 - 11s - loss: 0.0502 - acc: 0.9844 - val_loss: 0.0274 - val_acc: 0.9901
Epoch 9/12
 - 11s - loss: 0.0449 - acc: 0.9854 - val_loss: 0.0280 - val_acc: 0.9910
Epoch 10/12
 - 11s - loss: 0.0446 - acc: 0.9863 - val_loss: 0.0279 - val_acc: 0.9914
Epoch 11/12
 - 11s - loss: 0.0412 - acc: 0.9869 - val_loss: 0.0286 - val_acc: 0.9909
Epoch 12/12
 - 11s - loss: 0.0395 - acc: 0.9872 - val_loss: 0.0259 - val_acc: 0.9915
Test loss: 0.025930335229737103
Test accuracy: 0.9915
```

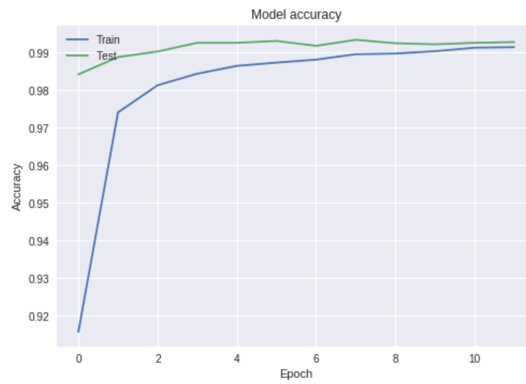


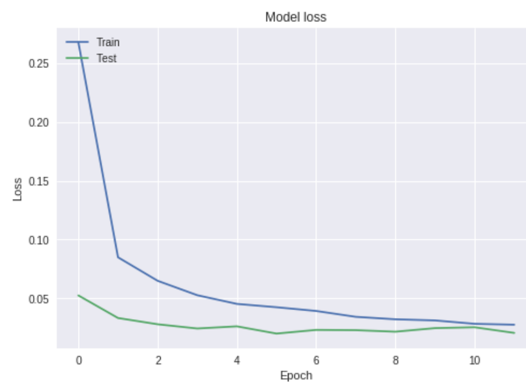
b. 8 X 8

Total Time: 107 seconds

Test Loss: 2.0547%

Test Accuracy: 99.28%






```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 10s - loss: 0.2679 - acc: 0.9157 - val_loss: 0.0524 - val_acc: 0.9842
Epoch 2/12
 - 8s - loss: 0.0849 - acc: 0.9741 - val_loss: 0.0332 - val_acc: 0.9888
Epoch 3/12
 - 9s - loss: 0.0649 - acc: 0.9813 - val_loss: 0.0279 - val_acc: 0.9903
Epoch 4/12
 - 9s - loss: 0.0526 - acc: 0.9844 - val_loss: 0.0243 - val_acc: 0.9926
Epoch 5/12
 - 9s - loss: 0.0452 - acc: 0.9865 - val_loss: 0.0261 - val_acc: 0.9926
Epoch 6/12
 - 9s - loss: 0.0424 - acc: 0.9873 - val_loss: 0.0200 - val_acc: 0.9931
Epoch 7/12
 - 9s - loss: 0.0392 - acc: 0.9881 - val_loss: 0.0231 - val_acc: 0.9918
Epoch 8/12
 - 8s - loss: 0.0342 - acc: 0.9896 - val_loss: 0.0229 - val_acc: 0.9934
Epoch 9/12
 - 9s - loss: 0.0321 - acc: 0.9898 - val_loss: 0.0216 - val_acc: 0.9925
Epoch 10/12
 - 9s - loss: 0.0312 - acc: 0.9904 - val_loss: 0.0246 - val_acc: 0.9922
Epoch 11/12
 - 9s - loss: 0.0283 - acc: 0.9913 - val_loss: 0.0254 - val_acc: 0.9926
Epoch 12/12
 - 9s - loss: 0.0275 - acc: 0.9914 - val_loss: 0.0205 - val_acc: 0.9928
Test loss: 0.020546274933998666
Test accuracy: 0.9928
```

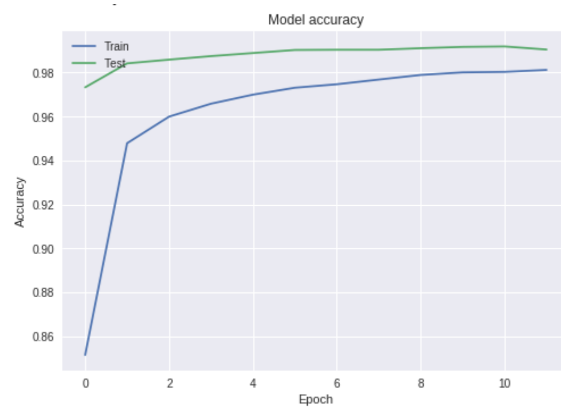
5. Change in number of Neurons

a. 16, 32, 64 combination

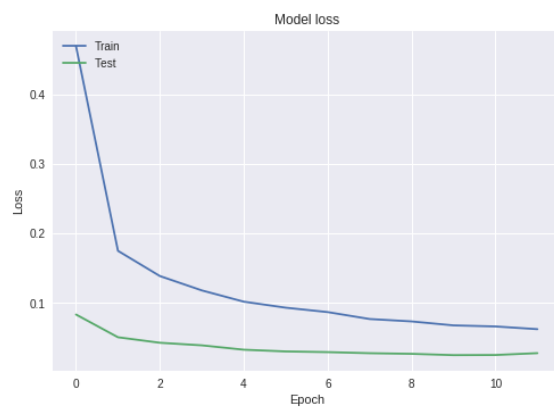
Total Time: 97 seconds

Test Loss: 2.747%

Test Accuracy: 99.05%




```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 9s - loss: 0.4694 - acc: 0.8515 - val_loss: 0.0831 - val_acc: 0.9733
Epoch 2/12
 - 8s - loss: 0.1748 - acc: 0.9479 - val_loss: 0.0503 - val_acc: 0.9842
Epoch 3/12
 - 8s - loss: 0.1385 - acc: 0.9600 - val_loss: 0.0424 - val_acc: 0.9859
Epoch 4/12
 - 8s - loss: 0.1178 - acc: 0.9659 - val_loss: 0.0387 - val_acc: 0.9875
Epoch 5/12
 - 8s - loss: 0.1016 - acc: 0.9700 - val_loss: 0.0324 - val_acc: 0.9889
Epoch 6/12
 - 8s - loss: 0.0929 - acc: 0.9731 - val_loss: 0.0299 - val_acc: 0.9903
Epoch 7/12
 - 8s - loss: 0.0866 - acc: 0.9747 - val_loss: 0.0290 - val_acc: 0.9904
Epoch 8/12
 - 8s - loss: 0.0766 - acc: 0.9768 - val_loss: 0.0274 - val_acc: 0.9904
Epoch 9/12
 - 8s - loss: 0.0732 - acc: 0.9789 - val_loss: 0.0265 - val_acc: 0.9911
Epoch 10/12
 - 8s - loss: 0.0674 - acc: 0.9801 - val_loss: 0.0246 - val_acc: 0.9917
Epoch 11/12
 - 8s - loss: 0.0659 - acc: 0.9803 - val_loss: 0.0248 - val_acc: 0.9919
Epoch 12/12
 - 8s - loss: 0.0620 - acc: 0.9812 - val_loss: 0.0275 - val_acc: 0.9905
Test loss: 0.027470250113546445
Test accuracy: 0.9905
```

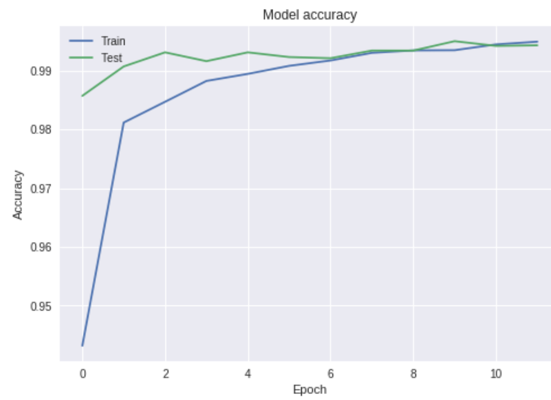


b. 64, 128, 256 combination

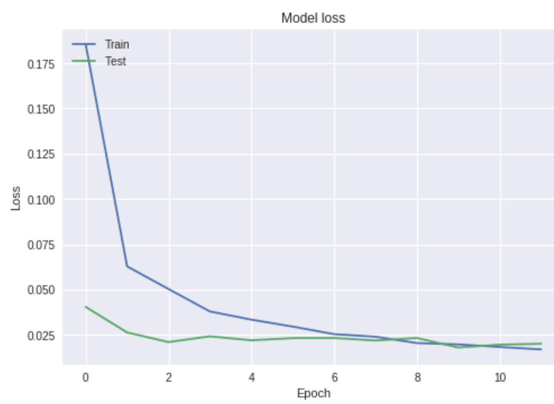
Total Time: 230 seconds

Test Loss: 2.006%

Test Accuracy: 99.43%




```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 21s - loss: 0.1854 - acc: 0.9432 - val_loss: 0.0404 - val_acc: 0.9857
Epoch 2/12
 - 19s - loss: 0.0628 - acc: 0.9811 - val_loss: 0.0263 - val_acc: 0.9907
Epoch 3/12
 - 19s - loss: 0.0503 - acc: 0.9847 - val_loss: 0.0210 - val_acc: 0.9931
Epoch 4/12
 - 19s - loss: 0.0379 - acc: 0.9882 - val_loss: 0.0242 - val_acc: 0.9916
Epoch 5/12
 - 19s - loss: 0.0334 - acc: 0.9894 - val_loss: 0.0220 - val_acc: 0.9931
Epoch 6/12
 - 19s - loss: 0.0295 - acc: 0.9908 - val_loss: 0.0232 - val_acc: 0.9923
Epoch 7/12
 - 19s - loss: 0.0254 - acc: 0.9917 - val_loss: 0.0232 - val_acc: 0.9921
Epoch 8/12
 - 19s - loss: 0.0239 - acc: 0.9930 - val_loss: 0.0219 - val_acc: 0.9934
Epoch 9/12
 - 19s - loss: 0.0204 - acc: 0.9934 - val_loss: 0.0232 - val_acc: 0.9934
Epoch 10/12
 - 19s - loss: 0.0197 - acc: 0.9935 - val_loss: 0.0180 - val_acc: 0.9950
Epoch 11/12
 - 19s - loss: 0.0183 - acc: 0.9944 - val_loss: 0.0196 - val_acc: 0.9942
Epoch 12/12
 - 19s - loss: 0.0169 - acc: 0.9949 - val_loss: 0.0201 - val_acc: 0.9943
Test loss: 0.02006073566749992
Test accuracy: 0.9943
```

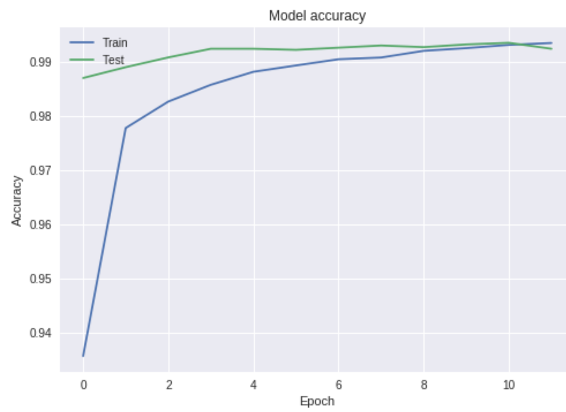


6. Without Max Pooling

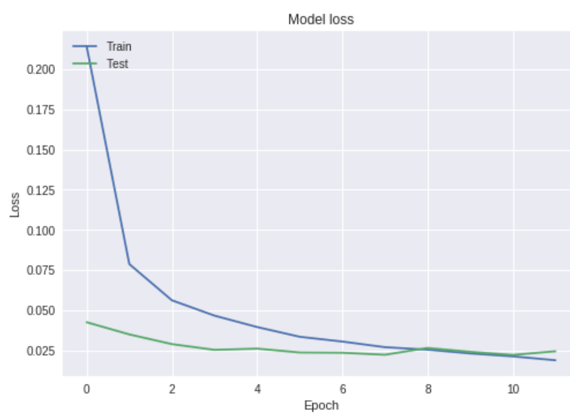
Total Time: 157 seconds

Test Loss: 2.4528%

Test Accuracy: 99.24%




```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 14s - loss: 0.2139 - acc: 0.9357 - val_loss: 0.0426 - val_acc: 0.9870
Epoch 2/12
 - 13s - loss: 0.0787 - acc: 0.9778 - val_loss: 0.0350 - val_acc: 0.9890
Epoch 3/12
 - 13s - loss: 0.0562 - acc: 0.9827 - val_loss: 0.0290 - val_acc: 0.9908
Epoch 4/12
 - 13s - loss: 0.0467 - acc: 0.9857 - val_loss: 0.0254 - val_acc: 0.9924
Epoch 5/12
 - 13s - loss: 0.0396 - acc: 0.9881 - val_loss: 0.0262 - val_acc: 0.9924
Epoch 6/12
 - 13s - loss: 0.0335 - acc: 0.9893 - val_loss: 0.0238 - val_acc: 0.9922
Epoch 7/12
 - 13s - loss: 0.0306 - acc: 0.9905 - val_loss: 0.0236 - val_acc: 0.9926
Epoch 8/12
 - 13s - loss: 0.0271 - acc: 0.9908 - val_loss: 0.0224 - val_acc: 0.9930
Epoch 9/12
 - 13s - loss: 0.0255 - acc: 0.9920 - val_loss: 0.0266 - val_acc: 0.9927
Epoch 10/12
 - 13s - loss: 0.0232 - acc: 0.9925 - val_loss: 0.0242 - val_acc: 0.9932
Epoch 11/12
 - 13s - loss: 0.0214 - acc: 0.9931 - val_loss: 0.0223 - val_acc: 0.9935
Epoch 12/12
 - 13s - loss: 0.0190 - acc: 0.9935 - val_loss: 0.0245 - val_acc: 0.9924
Test loss: 0.024528377351882773
Test accuracy: 0.9924
```

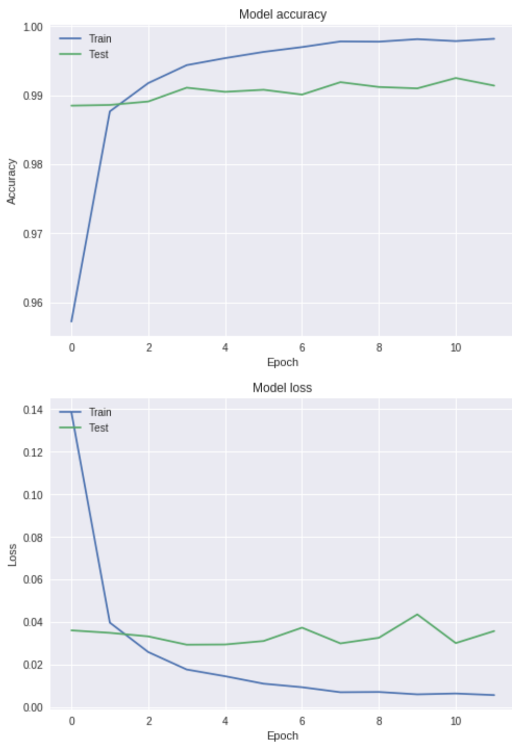


7. Without Dropout

Total Time: 96 seconds

Test Loss: 3.5775%

Test Accuracy: 99.14%



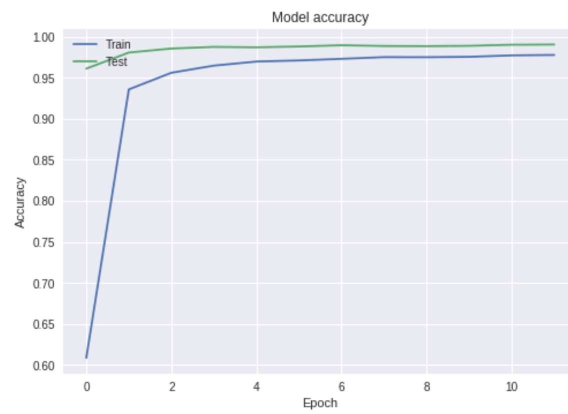

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 8s - loss: 0.1385 - acc: 0.9572 - val_loss: 0.0361 - val_acc: 0.9885
Epoch 2/12
 - 8s - loss: 0.0397 - acc: 0.9877 - val_loss: 0.0349 - val_acc: 0.9886
Epoch 3/12
 - 8s - loss: 0.0259 - acc: 0.9918 - val_loss: 0.0333 - val_acc: 0.9891
Epoch 4/12
 - 8s - loss: 0.0177 - acc: 0.9944 - val_loss: 0.0294 - val_acc: 0.9911
Epoch 5/12
 - 8s - loss: 0.0146 - acc: 0.9954 - val_loss: 0.0295 - val_acc: 0.9905
Epoch 6/12
 - 8s - loss: 0.0111 - acc: 0.9963 - val_loss: 0.0311 - val_acc: 0.9908
Epoch 7/12
 - 8s - loss: 0.0094 - acc: 0.9970 - val_loss: 0.0374 - val_acc: 0.9901
Epoch 8/12
 - 8s - loss: 0.0070 - acc: 0.9978 - val_loss: 0.0300 - val_acc: 0.9919
Epoch 9/12
 - 8s - loss: 0.0072 - acc: 0.9978 - val_loss: 0.0326 - val_acc: 0.9912
Epoch 10/12
 - 8s - loss: 0.0060 - acc: 0.9981 - val_loss: 0.0436 - val_acc: 0.9910
Epoch 11/12
 - 8s - loss: 0.0064 - acc: 0.9978 - val_loss: 0.0301 - val_acc: 0.9925
Epoch 12/12
 - 8s - loss: 0.0057 - acc: 0.9982 - val_loss: 0.0358 - val_acc: 0.9914
Test loss: 0.03577511838136388
Test accuracy: 0.9914
```

8. Without Normalization

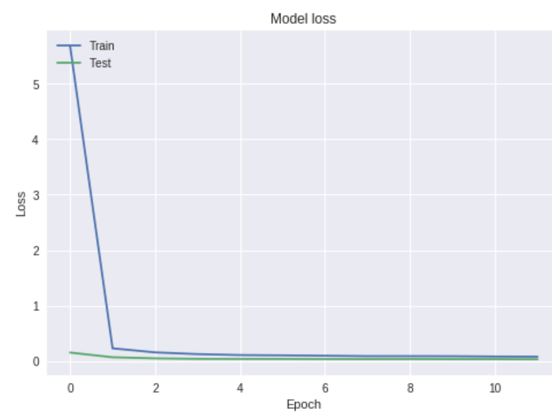
Total Time: 120 seconds

Test Loss: 3.136%

Test Accuracy: 99.07%




```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 10s - loss: 5.6814 - acc: 0.6085 - val_loss: 0.1525 - val_acc: 0.9613
Epoch 2/12
 - 10s - loss: 0.2288 - acc: 0.9359 - val_loss: 0.0677 - val_acc: 0.9809
Epoch 3/12
 - 10s - loss: 0.1559 - acc: 0.9562 - val_loss: 0.0478 - val_acc: 0.9858
Epoch 4/12
 - 10s - loss: 0.1247 - acc: 0.9649 - val_loss: 0.0388 - val_acc: 0.9878
Epoch 5/12
 - 10s - loss: 0.1074 - acc: 0.9699 - val_loss: 0.0372 - val_acc: 0.9873
Epoch 6/12
 - 10s - loss: 0.1022 - acc: 0.9712 - val_loss: 0.0368 - val_acc: 0.9883
Epoch 7/12
 - 10s - loss: 0.0956 - acc: 0.9732 - val_loss: 0.0356 - val_acc: 0.9898
Epoch 8/12
 - 10s - loss: 0.0880 - acc: 0.9753 - val_loss: 0.0342 - val_acc: 0.9889
Epoch 9/12
 - 10s - loss: 0.0880 - acc: 0.9752 - val_loss: 0.0357 - val_acc: 0.9887
Epoch 10/12
 - 10s - loss: 0.0866 - acc: 0.9757 - val_loss: 0.0332 - val_acc: 0.9891
Epoch 11/12
 - 10s - loss: 0.0809 - acc: 0.9775 - val_loss: 0.0333 - val_acc: 0.9904
Epoch 12/12
 - 10s - loss: 0.0782 - acc: 0.9780 - val_loss: 0.0314 - val_acc: 0.9907
Test loss: 0.03136393967232098
Test accuracy: 0.9907
```

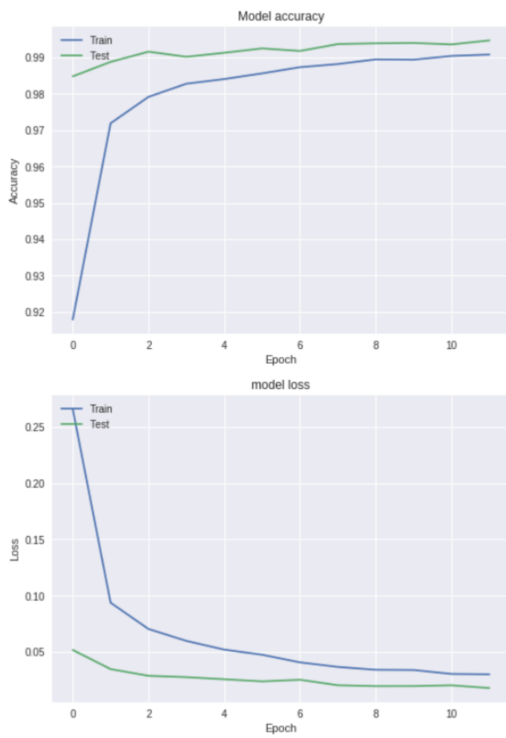


9. Default and Optimized

Total Time: 121 seconds

Test Loss: 1.782%

Test Accuracy: 99.47%




```
Train on 60000 samples, validate on 10000 samples
Epoch 1/12
 - 11s - loss: 0.2661 - acc: 0.9179 - val_loss: 0.0517 - val_acc: 0.9848
Epoch 2/12
 - 10s - loss: 0.0938 - acc: 0.9719 - val_loss: 0.0347 - val_acc: 0.9888
Epoch 3/12
 - 10s - loss: 0.0704 - acc: 0.9792 - val_loss: 0.0287 - val_acc: 0.9916
Epoch 4/12
 - 10s - loss: 0.0598 - acc: 0.9828 - val_loss: 0.0275 - val_acc: 0.9902
Epoch 5/12
 - 10s - loss: 0.0520 - acc: 0.9841 - val_loss: 0.0257 - val_acc: 0.9913
Epoch 6/12
 - 10s - loss: 0.0475 - acc: 0.9856 - val_loss: 0.0238 - val_acc: 0.9925
Epoch 7/12
 - 10s - loss: 0.0407 - acc: 0.9873 - val_loss: 0.0252 - val_acc: 0.9918
Epoch 8/12
 - 10s - loss: 0.0366 - acc: 0.9882 - val_loss: 0.0203 - val_acc: 0.9937
Epoch 9/12
 - 10s - loss: 0.0341 - acc: 0.9895 - val_loss: 0.0196 - val_acc: 0.9939
Epoch 10/12
 - 10s - loss: 0.0339 - acc: 0.9894 - val_loss: 0.0196 - val_acc: 0.9940
Epoch 11/12
 - 10s - loss: 0.0304 - acc: 0.9904 - val_loss: 0.0203 - val_acc: 0.9936
Epoch 12/12
 - 10s - loss: 0.0301 - acc: 0.9908 - val_loss: 0.0178 - val_acc: 0.9947
Test loss: 0.017822115575458885
Test accuracy: 0.9947
```

(c) Discussion

To achieve higher accuracy, we had to implement a more complex model. Complex models lead to more computations and thus more time. So, to make it time efficient, complexity had to be reduced.

Normalization improves the accuracy by 0.4%(approx.) Dropout improves the accuracy by 0.3%(approx.) Max Pooling improves the accuracy by 0.2%(approx.)

Reducing the number of Neurons decreases the accuracy, increasing them leads to more computation time. Any change in Kernel size leads to a reduce in accuracy. We are unable to figure out why a 5X5 Kernel leads to better accuracy. Increasing the number of Epochs leads to increase in accuracy but the time required to complete the process increases exponentially. Lesser Epochs lead to less accuracy.

No solid conclusion can be said for change in Batch Size and change in number of layers. The accuracy fluctuates in these cases. Computation time increases as the complexity increases.

(4) Related Work

A number of techniques have been implemented for an efficient task of Optical Character recognition like:

Fast Contour Matching using Approximate Earth Mover's Distance - [1]

Fast Pruning using Generalized Shape Contexts – [2]

Gradient-based learning applied to document recognition. – [3]

A Global Geometric Framework for Nonlinear Dimensionality Reduction. – [4]

(5) Future Work

There is a possibility of using an efficient GPU or a GPU efficient framework like Caffe. There might be other combinations of the parameters which would lead to better results. The structure could be modified in terms of layers to get better results.

There have been a lot of algorithms implemented for efficient Optical character recognition for MNIST dataset. Some algorithms using Neural Nets get errors <1%.

There also have been SVM implementation using different kernels. But it is yet to be seen how ours and all other implementations fare against another dataset. MNIST is a pretty standard dataset that is used widely, thus it cannot give an idea of the real world performance of the algorithms.

We could use some algorithms like ISOMAP - [4] which are extensions of PCA to get the intrinsic dimension of the data. By doing this we can find out the algorithms which give us better classification accuracy for a non MNIST dataset.

We could also use some computer vision algorithm for shape detection to get lesser error rate.

(6) Conclusion

We have presented a Neural Network with Optimized parameters to perform the task of Optical Character Recognition. Any change in the parameters leads to either reduced accuracy or slightly improved accuracy with exponential increase in computation time.

The optimized parameters are:

Batch Size = 128

Number of Classes = 10

Image rows, Image Columns = 28 X 28

Layer 1 : Input

Neurons 32, Conv2D, Kernel Size 5 X 5, Activation = Relu, Pool Size 2 X 2, Dropout 0.4

Layer 2

Neurons 64, Conv2D, Kernel Size 5 X 5, Activation = Relu, Pool Size 2 X 2, Dropout 0.4

Layer 3

Neurons 128, Dense, Activation = Relu, Dropout 0.5

Layer 4 : Output

Neurons 10, Dense, Activation = Softmax

Total Time: 121 seconds

Test Loss: 1.782%

Test Accuracy: 99.47%

(7) Bibliography

[1] K Grauman and T Darrell. Fast contour matching using approximate earth movers distance. Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2004 CVPR 2004, 1(June):220–227, 2004.

[2] Greg Mori, Serge Belongie, Jitendra Malik, and Senior Member. Efficient shape matching using shape contexts. IEEE Trans. Pattern Analysis and Machine Intelligence, 27:1832–1837, 2005.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998

[4] J. B. Tenenbaum, V. de Silva and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science 290 5500,2319, (2000)

[5] http://yann.lecun.com/exdb/mnist/