ELL409
Assignment 3

Submitted by:
Raghav Gupta
2017EE10544

(a) Basic Implementation

- Basic implementation of fully connected neural networks was written.
- The implementation allowed flexibility over the following:
    - Number of hidden layers
    - Number of neurons in each layer
    - Gradient Descent parameters-learning rate, batch size, iterations
    - Different activation functions for each hidden layer
    - Regularization

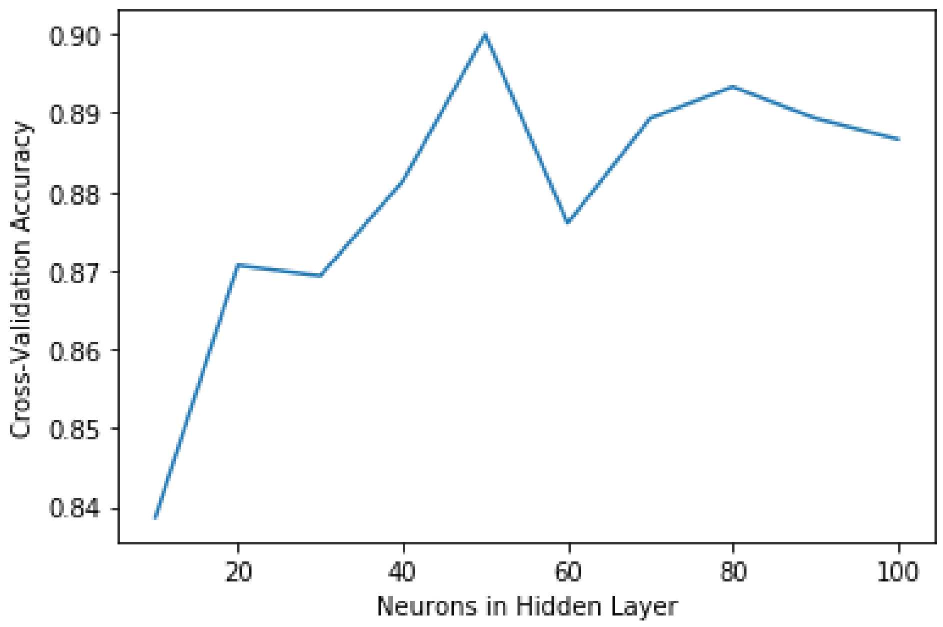
- Neural Network library used was Keras from Tensorflow (https://www.tensorflow.org/tutorials/keras/classification)

(b) Standard Backpropagation Neural Net

- Neural Network was trained on different hyperparameter settings and the model was accessed based on the 4-fold Cross Validation Accuracy.
- Tried with 1 and 2 hidden layers using activation functions like Sigmoid and RELU.
- The last layer used Softmax activation for multiclass classification.
- The results using 1 hidden layer were given as follows:

1. <u>Sigmoid Activated Hidden Layer:</u>

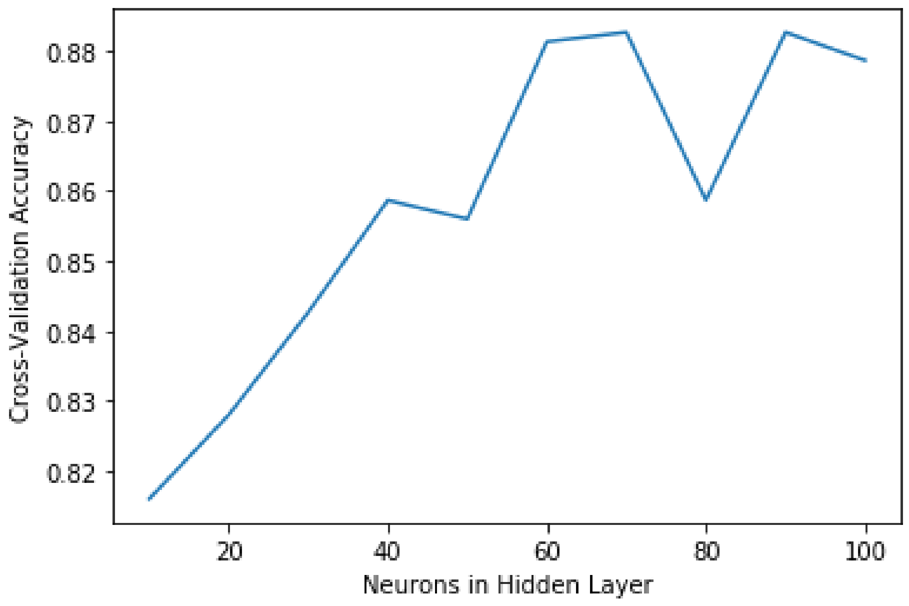
- 1 hidden layer was used with sigmoid activation function followed by Softmax for the output layer.
- Learning rate was kept 0.2 and batch size was kept as 1 (Stochastic Gradient Descent) and 13 epochs were used.
- The variation in Cross-Validation accuracy with number of neurons in hidden layer is given below:



- Best 4 fold Cross-Validation Accuracy obtained was 0.9 when 50 neurons were used in hidden layer. The model thus trained gave Training Accuracy 0.995.
- Using Keras this model gave 4 fold Cross-Validation Accuracy 0.9093 and Training Accuracy 0.984.

2. <u>RELU Activated Hidden Layer:</u>

- 1 hidden layer was used with RELU activation function followed by Softmax for the output layer.
- Learning rate was kept 0.1 and batch size was kept as 1 (Stochastic Gradient Descent) and 13 epochs were used.
- The variation in Cross-Validation accuracy with number of neurons in hidden layer is given below:



- Best 4 fold Cross-Validation Accuracy obtained was 0.8826 when 70 neurons were used in hidden layer. The model thus trained gave Training Accuracy 1.0.
- Using Keras this model gave 4 fold Cross-Validation Accuracy 0.8973 and Training Accuracy 0.9991.

- The results using 2 hidden layers were given as follows:

1. <u>Sigmoid-Sigmoid Activated Hidden Layers:</u>

    - 2 hidden layers were used with sigmoid activation function in both followed by Softmax for the output layer.
    - Learning rate was kept 0.2 and batch size was kept as 1 (Stochastic Gradient Descent) and 13 epochs were used.
    - The variation in Cross-Validation accuracy with number of neurons in each hidden layer is given below:

Neuron in 1st hidden layer	Neuron in 2nd hidden layer	Cross-Validation Accuracy
60	60	0.896
60	100	0.8973
60	200	0.904
100	60	0.8946
100	100	0.8986
100	200	0.916
200	60	0.8813
200	100	0.888
200	200	0.906

- Best 4 fold Cross-Validation Accuracy obtained was 0.916 when 100 neurons were used in 1st hidden layer and 200 neurons in 2nd hidden layer. The model thus trained gave Training Accuracy 1.0.
- Using Keras this model gave 4 fold Cross-Validation Accuracy 0.896 and Training Accuracy 0.9911.

2. <u>Sigmoid-RELU Activated Hidden Layers:</u>

- 2 hidden layers were used with sigmoid activation function in 1st and RELU in 2nd followed by Softmax for the output layer.
- Learning rate was kept 0.1 and batch size was kept as 1 (Stochastic Gradient Descent) and 13 epochs were used.
- The variation in Cross-Validation accuracy with number of neurons in each hidden layer is given below:

Neuron in 1st hidden layer	Neuron in 2nd hidden layer	Cross-Validation Accuracy
60	60	0.836
60	100	0.8626
60	200	0.8653
100	60	0.8546
100	100	0.8626
100	200	0.868
200	60	0.2813
200	100	0.606

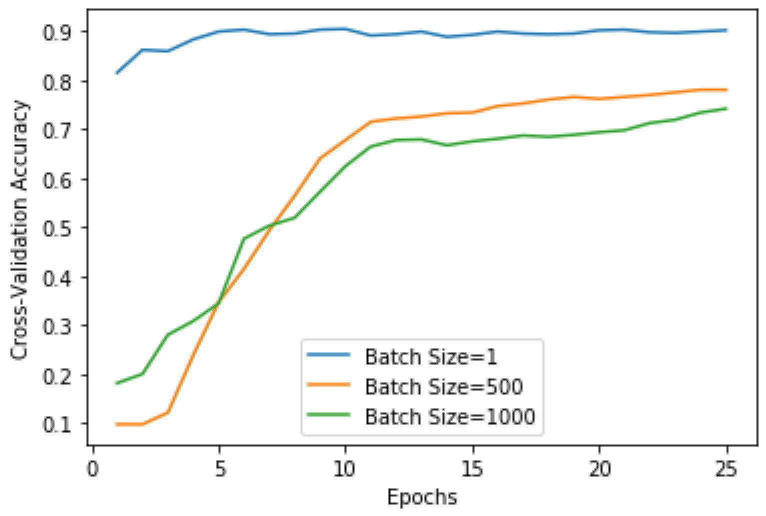
- Best 4 fold Cross-Validation Accuracy obtained was 0.868 when 100 neurons were used in 1st hidden layer and 200 neurons in 2nd hidden layer. The model thus trained gave Training Accuracy 0.977.
- Using Keras this model gave 4 fold Cross-Validation Accuracy 0.8746 and Training Accuracy 0.9964.

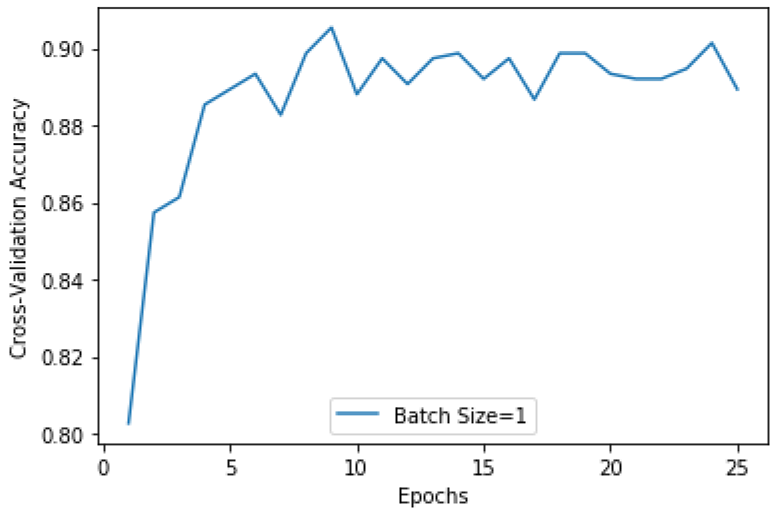
3. <u>RELU-RELU Activated Hidden Layers:</u>

- RELU-RELU gave poor results in my implementation but the library Keras gave good results.
- In my implementation RELU-Relu often resulted in overflow and when learning rate was reduced the results obtained were poor.
- Library gave Cross-Validation accuracy of 0.904 with 100 neurons in 1st hidden layer and 200 neurons in 2nd hidden layer both with RELU activation function.
- Same model gave Cross-Validation accuracy of 0.77 in my implementation. When learning rate was adjusted, it resulted in overflow or less accuracy. This could be due to Python precision error.

Variation with batch size and epochs:

- The model with 1 hidden layer (sigmoid activated) was used, with 50 neurons in hidden layer.
- Cross Validation accuracy was plotted against Epochs for batch size 1, 500, 1000.
- The results are shown below:

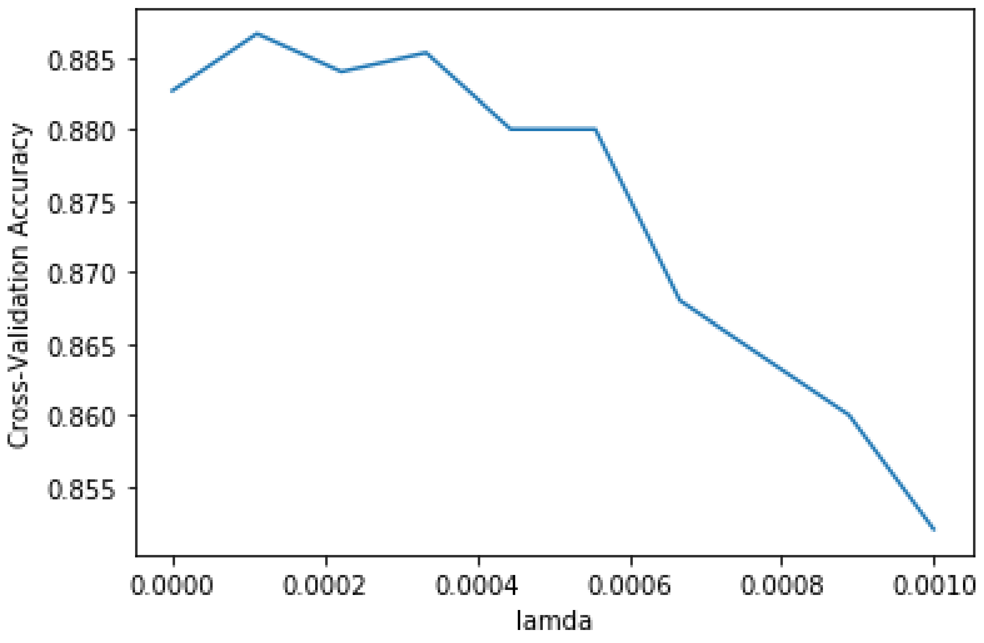




- As we can see from the plots, batch size=1 (Stochastic Gradient Descent) gave best results although it didn't vary smoothly with number of epochs.
- As we increase batch size the Cross-Validation accuracy increases smoothly as epochs are increased.

Introducing Regularization:

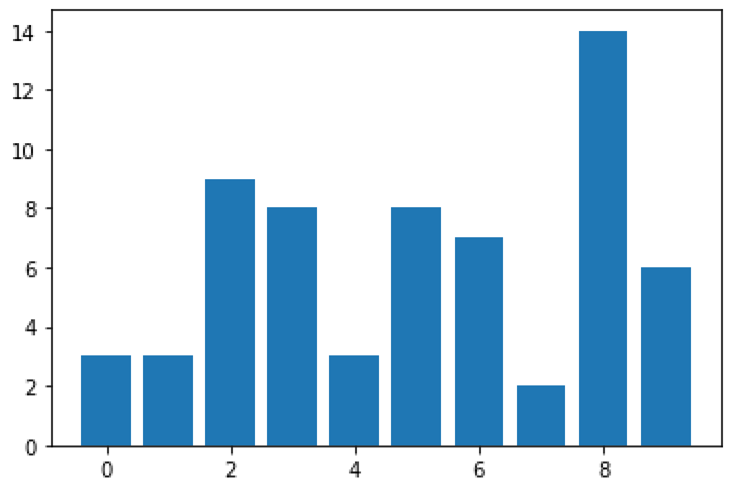
- The model with 1 hidden layer (sigmoid activated) was used, with 50 neurons in hidden layer.
- Lamda was varied from 0 to 0.001 and Cross-Validation Accuracy was plotted against lambda as shown below:



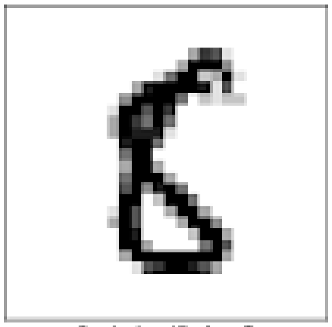
- We can see from the above plot that Cross-Validation Accuracy first increases and then decreases as we increase lambda.
- Reducing lambda results in overfitting, and thus optimum lambda was found to be 0.0001 for which Cross-Validation accuracy was found to be 0.886
- Introducing regularization didn't increase the Cross-Validation accuracy by a lot.

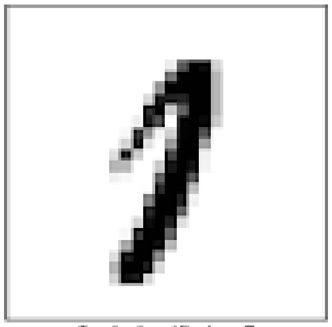
Final Model:

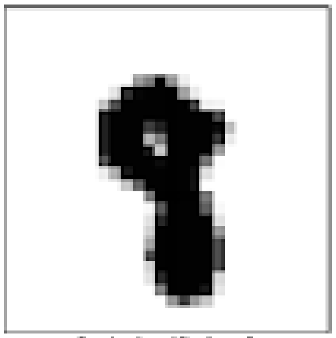
- The best model obtained so far was the one with 2 hidden layers (Sigmoid activation) with 100 and 200 neurons respectively.
- When this model was trained on 2250 points and rest 750 points were kept for validation.
- The validation accuracy obtained was 0.916 on the 750 test points.
- Similar results were obtained using Keras library but the library was much much faster.
- Following bar chart shows the number of times the digit on Horizontal axis was misclassified

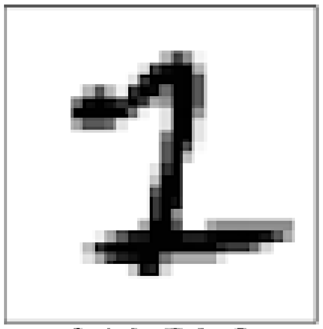


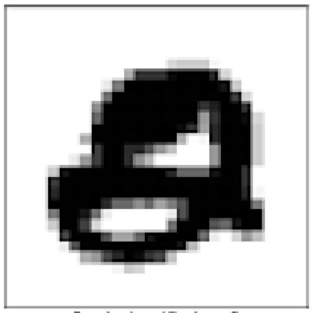
- From the above plot we can see that 8 was misclassified the most and 7 was misclassified least number of times.
- Following are some images that were misclassified by this model. Some of them are hard to recognize even by human brain.

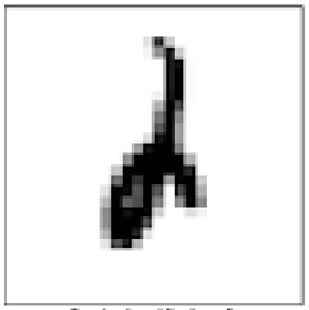

8 misclassified as 5

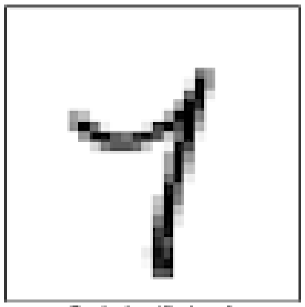

1 misclassified as 7

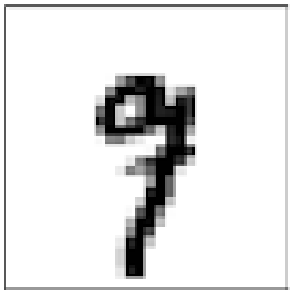

8 misclassified as 9


1 misclassified as 2


2 misclassified as 0


2 misclassified as 1


7 misclassified as 4


9 misclassified as 7

(c) PCA Comparison

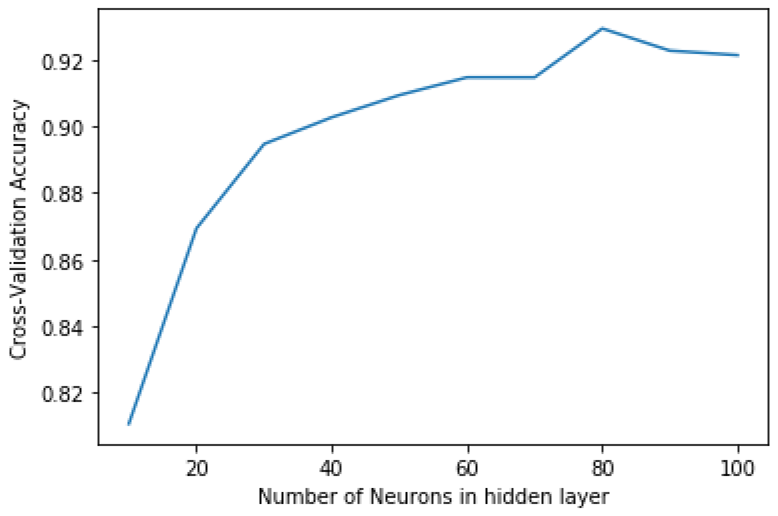
- PCA-space representation of the data was trained using Neural Network.

Logistic Regression:

- First, no hidden layer was used i.e. simple logistic regression. The learning rate was varied and best Cross-Validation Accuracy was obtained to be 0.878 when learning rate was kept 0.1.

Sigmoid Activated Hidden Layer:

- Now hidden layer was introduced (Sigmoid Activation) and neurons in the layer were varied. The plot for Cross-Validation Accuracy versus neurons in hidden layer is given below:



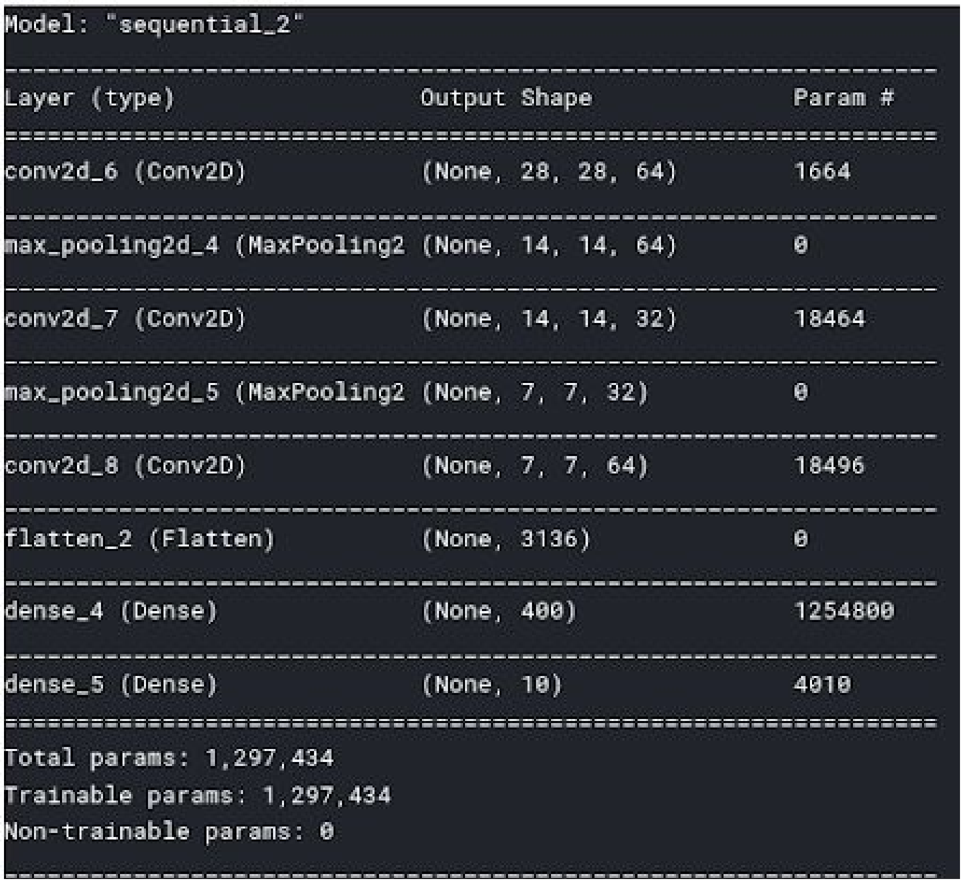
- The best Cross-Validation accuracy achieved was 0.9293 when 80 neurons were used.

Conclusions:

- It is clear that adding a hidden layer improves the cross-validation accuracy from 0.878 to 0.9293.
- PCA representation gave better results compared to that with raw pixels and the computation was much faster. The best Cross-Validation accuracy obtained by training on raw pixels was 0.916.
- This is because PCA maps the images to a lower dimensional space thus, discarding the useless features and making computations faster.

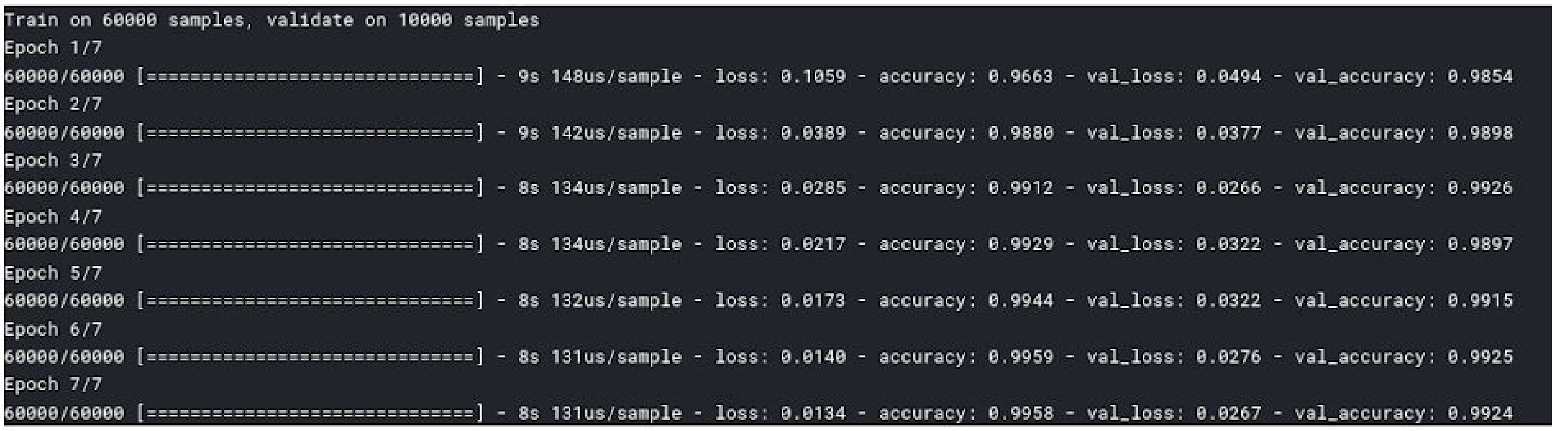
(d) Advanced Neural Networks

- Keras of Tensorflow was used to train MNIST data set on Convolutional Neural Network (https://www.tensorflow.org/tutorials/images/cnn).
- The following model gave best Testing Accuracy of 0.9924 on the 10000 test images.



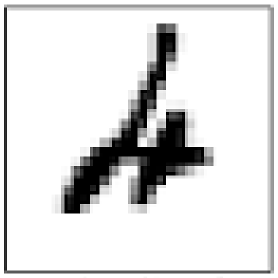
```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_6 (Conv2D)            (None, 28, 28, 64)        1664
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 14, 14, 64)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 14, 14, 32)        18464
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 7, 7, 32)          0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 7, 7, 64)          18496
_________________________________________________________________
flatten_2 (Flatten)          (None, 3136)              0
_________________________________________________________________
dense_4 (Dense)              (None, 400)               1254800
_________________________________________________________________
dense_5 (Dense)              (None, 10)                4010
=================================================================
Total params: 1,297,434
Trainable params: 1,297,434
Non-trainable params: 0
_________________________________________________________________
```

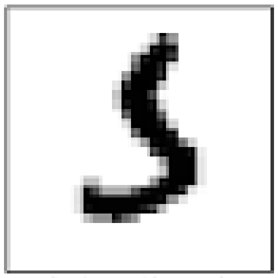
- Layers used are as follows:
    1. Convolution Layer (RELU Activated) - 64 feature maps and 5*5 Kernel
    2. MaxPool
    3. Convolution Layer (RELU Activated) - 32 feature maps and 3*3 Kernel
    4. MaxPool
    5. Convolution Layer (RELU Activated) - 64 feature maps and 3*3 Kernel
    6. Hidden Layer (RELU Activated) - 400 Neurons
    7. Output Layer (SoftMax)

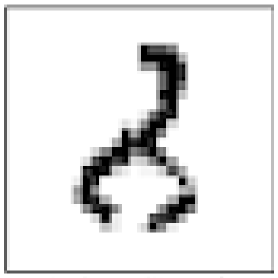


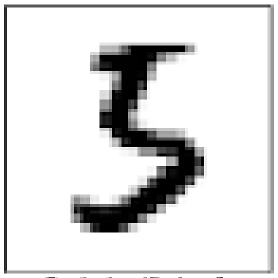
```
Train on 60000 samples, validate on 10000 samples
Epoch 1/7
60000/60000 [==============================] - 9s 148us/sample - loss: 0.1059 - accuracy: 0.9663 - val_loss: 0.0494 - val_accuracy: 0.9854
Epoch 2/7
60000/60000 [==============================] - 9s 142us/sample - loss: 0.0389 - accuracy: 0.9880 - val_loss: 0.0377 - val_accuracy: 0.9898
Epoch 3/7
60000/60000 [==============================] - 8s 134us/sample - loss: 0.0285 - accuracy: 0.9912 - val_loss: 0.0266 - val_accuracy: 0.9926
Epoch 4/7
60000/60000 [==============================] - 8s 134us/sample - loss: 0.0217 - accuracy: 0.9929 - val_loss: 0.0322 - val_accuracy: 0.9897
Epoch 5/7
60000/60000 [==============================] - 8s 132us/sample - loss: 0.0173 - accuracy: 0.9944 - val_loss: 0.0322 - val_accuracy: 0.9915
Epoch 6/7
60000/60000 [==============================] - 8s 131us/sample - loss: 0.0140 - accuracy: 0.9959 - val_loss: 0.0276 - val_accuracy: 0.9925
Epoch 7/7
60000/60000 [==============================] - 8s 131us/sample - loss: 0.0134 - accuracy: 0.9958 - val_loss: 0.0267 - val_accuracy: 0.9924
```

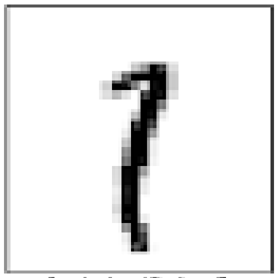
- Training on 60000 samples and Validation on 10000 samples gave the following results:
- Training Accuracy of 0.9958 and Testing Accuracy of 0.9924 was achieved.
- Some of the images misclassified by the final model are given below:

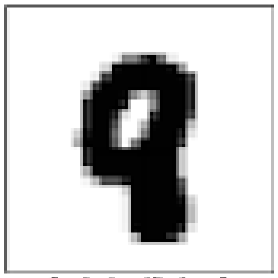

4 misclassified as 2

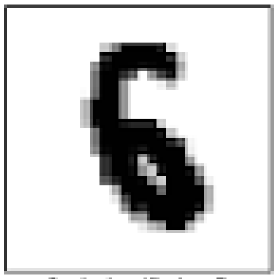

5 misclassified as 3

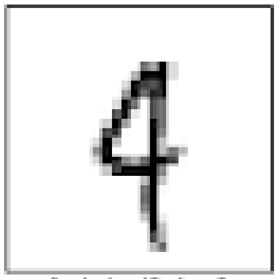

8 misclassified as 2

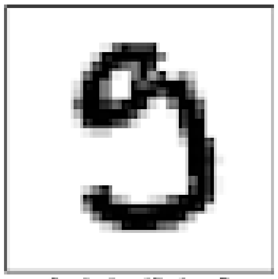

5 misclassified as 3

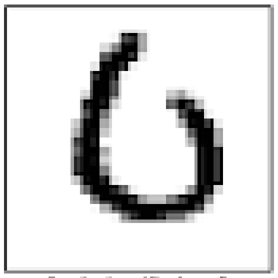

1 misclassified as 7

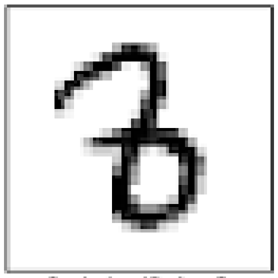

8 misclassified as 9

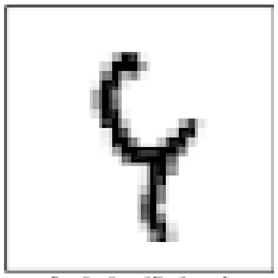

6 misclassified as 5

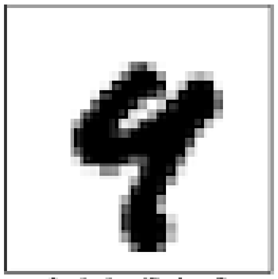

4 misclassified as 9

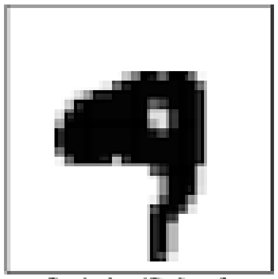

9 misclassified as 5

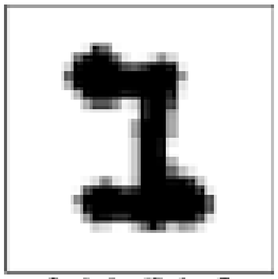

0 misclassified as 6


8 misclassified as 3


9 misclassified as 4


4 misclassified as 9


9 misclassified as 4


1 misclassified as 7

Conclusions:

- We can see that the images being misclassified by the model are somewhat ambiguous and hard to interpret even by human brain.
- Clearly this model outperforms the Standard Neural Network model.
- The representations being learned by Convolution Neural Network are more natural or intuitive than the representations learnt by the standard neural net trained earlier which is quite evident from the above images that are being misclassified.