CSE574 Spring 2019 Introduction to Machine Learning
Programming Assignment 2
Handwritten Digits Classification and Celebrity Face Detection

Group no. 7
Aditi Asthana
Sidharth Pati
Stuti Shukla

1

Choosing the hyper parameters for neural network

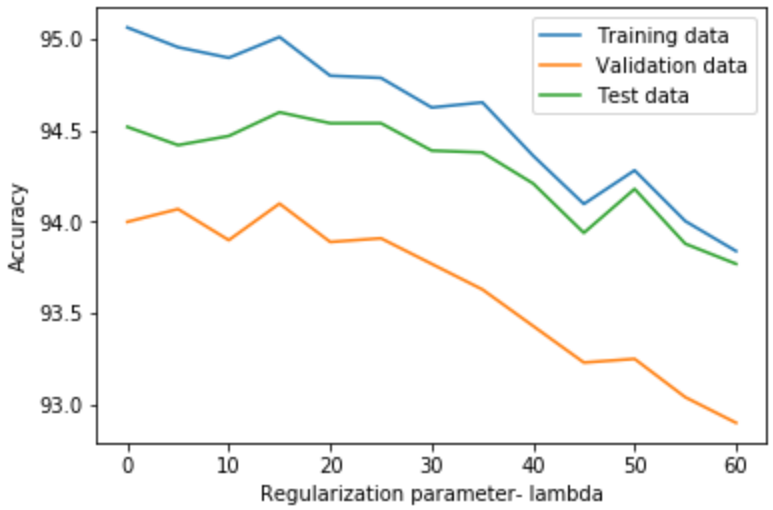
Performance of a neural network depends on the following factors:

1. Number of Hidden units
2. Regularisation term

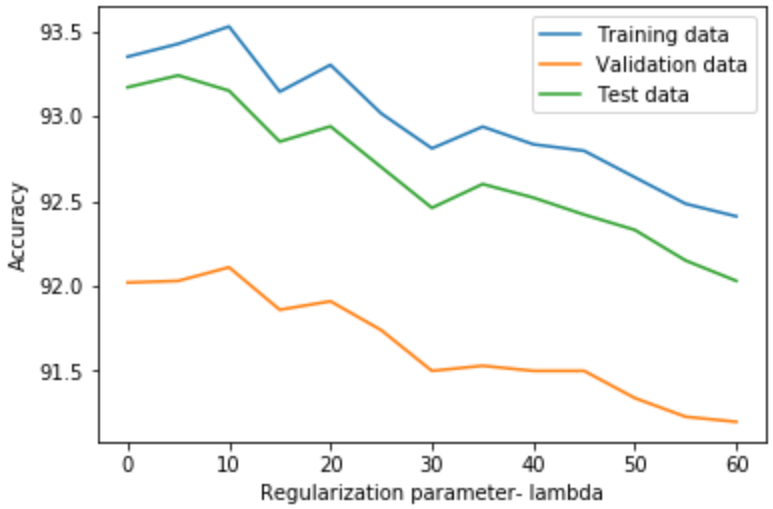
We performed various tests on our neural network by varying both these parameters and observed the following results:

Lambda vs accuracy

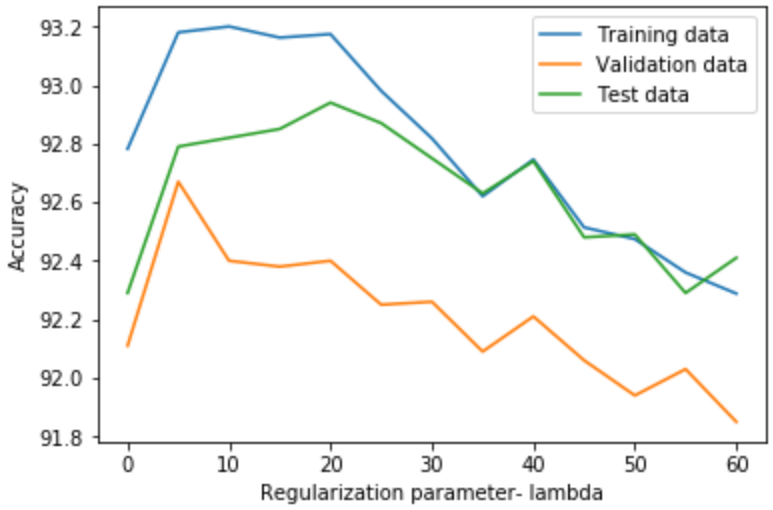
When number of **hidden units were 50**, we observed that the accuracy for test data is maximum when **lambda is 15**.



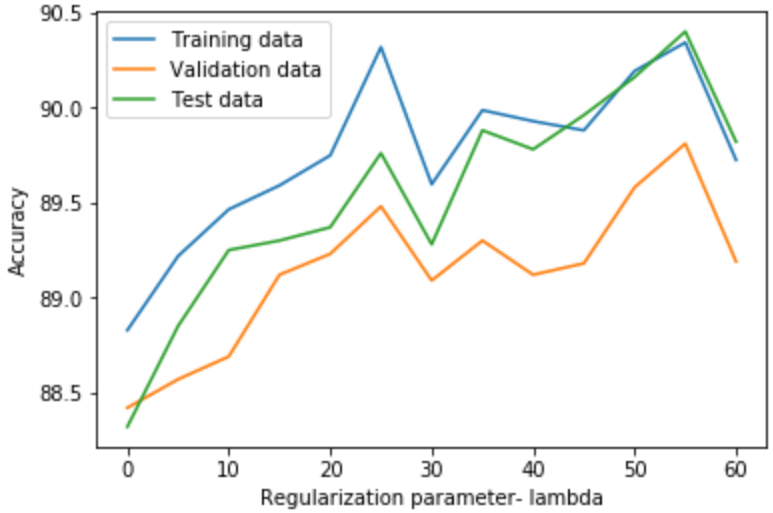
With 20 hidden units:



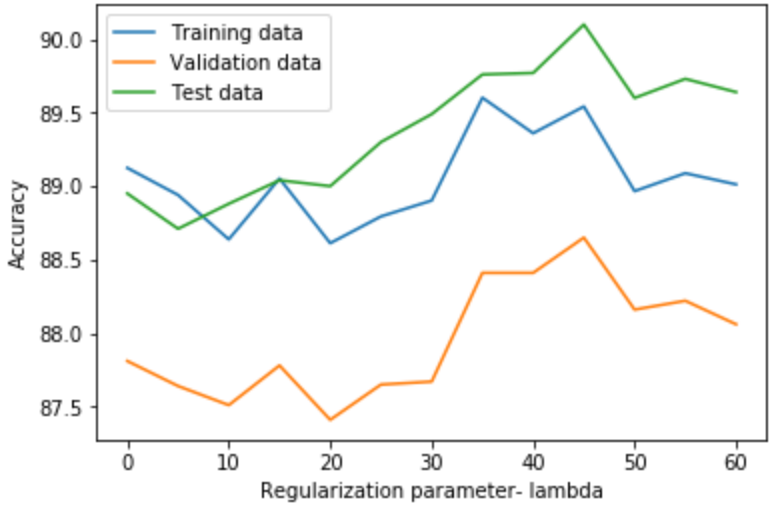
With 16 hidden units:



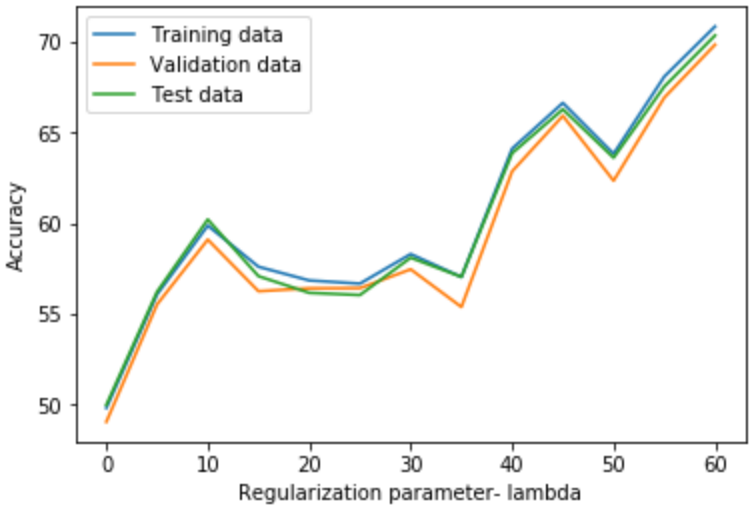
With 12 hidden units:



With 8 hidden units:



With 4 hidden units:



Observations:

The reason we use regularisation is to prevent overfitting of model with training data, to make our model more robust and perform better for any given test data.

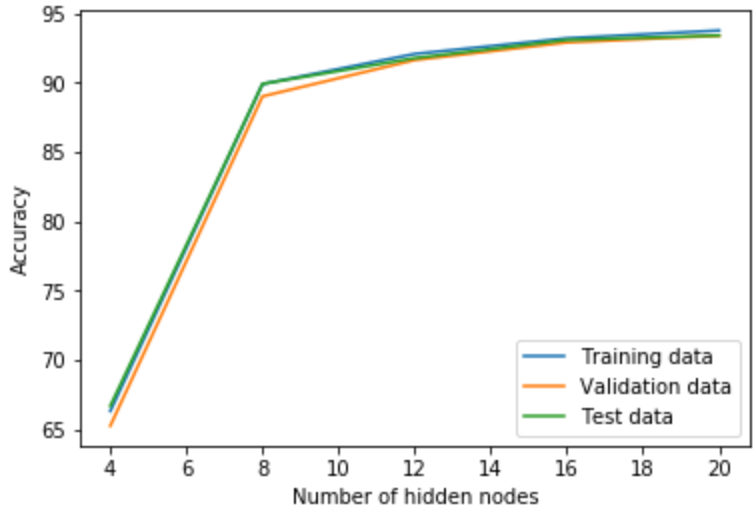
To find optimal lambda value, we varied lambda over small intervals(0, 5, 10,15, 20, 25, 30, 35, 40, 45, 50, 55, 60) keeping number of hidden units constant- 50 and plotting the respective accuracies.

It can be observed that **the accuracy is maximum for lambda=15.**

In addition to that, we also varied lambda for hidden units 4, 8 , 12, 16, 20. As observed in graphs, the results are not stable for hidden units <= 12. However, as the number of hidden nodes increases, the graph stabilises. So we conclude that more the number of hidden units, better the accuracy.

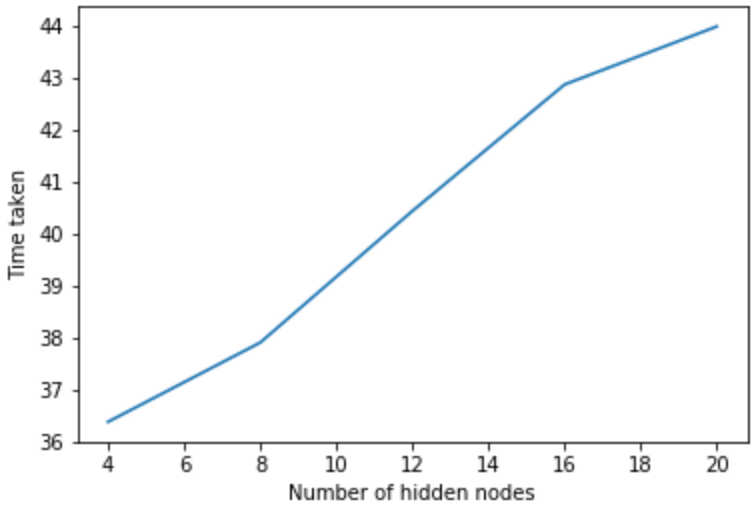
To further prove it, we changed the values of hidden nodes and observed the accuracies keeping lambda constant(15).

No. of Hidden units vs Accuracy



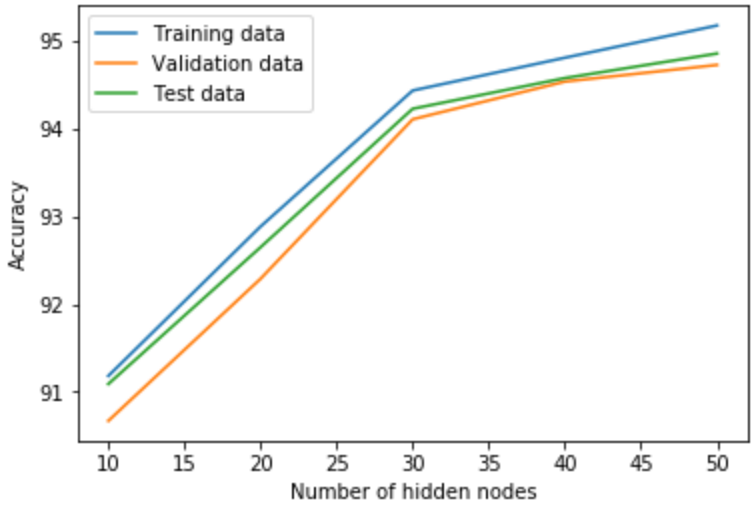
As seen above, lambda the accuracy is increasing as we vary hidden units from 4 to 20.

No. of Hidden units vs Time



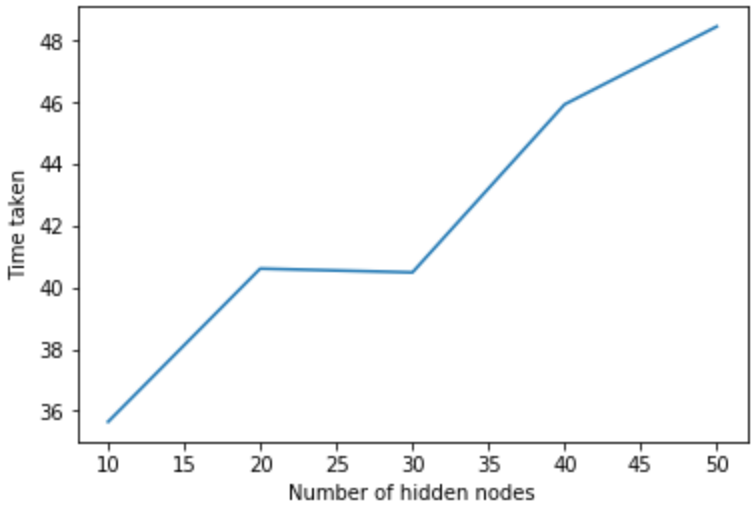
As the number of hidden units increases, the computation time of our neural network increases as evident from the above graph.

Accuracies when number of hidden units varies from 10 to 50



The accuracy is maximum when the hidden unit is 50.

Time taken when number of hidden units varies from 10 to 50



The optimal value for our hyper parameters is:

- Lambda=15
- Hidden node=50

2 **Accuracy of classification method on the handwritten digits test data**

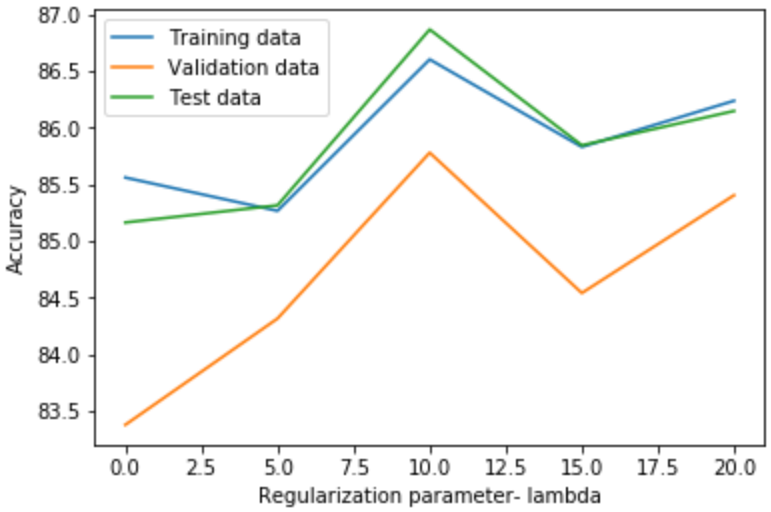
When number of hidden units is 50 and lambda value is 15, the accuracies for handwritten digits data set are as follows:
Training set: 95.062%
Validation set 94.77%
Test set Accuracy: 94.94%

3 **Accuracy of classification method on the CelebA dataset**

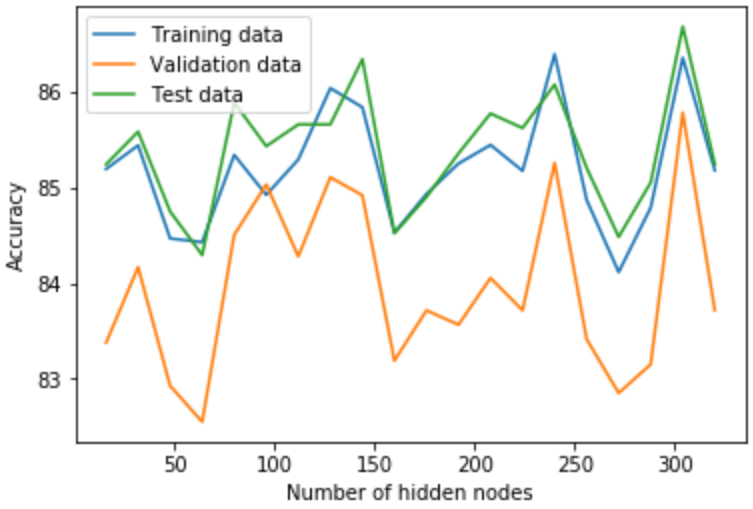
Accuracy of single hidden layer Neural Network on CelebA dataset using optimal parameters
Training set Accuracy:85.0521327014218%
Validation set Accuracy:83.67729831144464%
Test set Accuracy:85.04920514761545%
Time 223.36601638793945 sec

Accuracy vs Lambda for CelebA dataset

We observed that accuracy was maximum for lambda=10



Accuracy vs Hidden Units for CelebA dataset

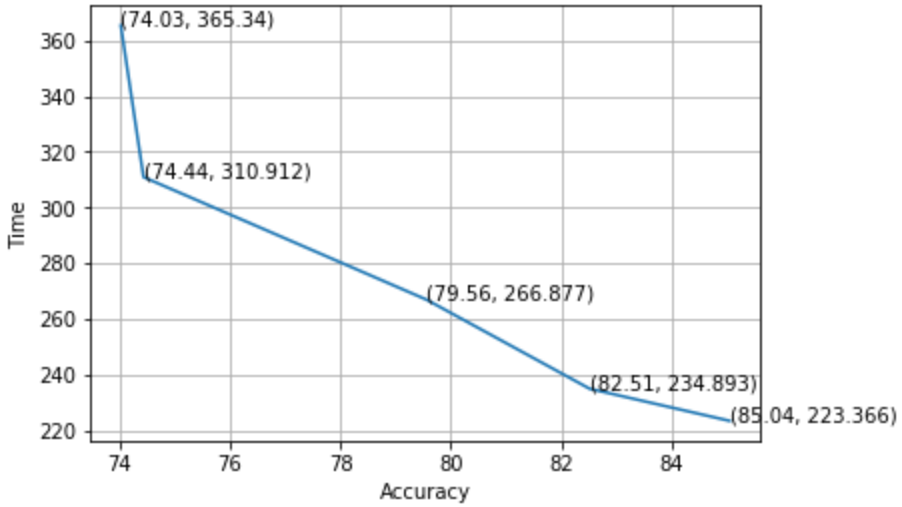


The above graphs show the varying accuracies when the lambda value changes between 0 & 20 and when the hidden units change between 16 & 320 respectively.

4 **Comparison of single hidden layer neural network vs deep neural network**

We used **Google Colab**, a free cloud service based on Jupyter Notebooks that supports free GPU and help develop deep learning applications using popular libraries such as PyTorch, TensorFlow, Keras, and OpenCV.

Layers	Accuracies(test data)	Time
Single Layer neural network	85.04%	223.366 sec
Two layer deep NN	82.51%	234.893 sec
Three layer deep NN	79.56%	266.877 sec
Five layer deep NN	74.44%	310.912 sec
Seven layer deep NN	74.03%	365.340 sec



Result show that as we increase the number of hidden layers, the computation time increases. However the accuracies not necessarily increases, this is in accordance with Occam's Razor which says that **'More complicated neural network will not give better result'.**

It can be observed that accuracy was highest for single layer neural network and it is decreasing as we increase the number of hidden layers.

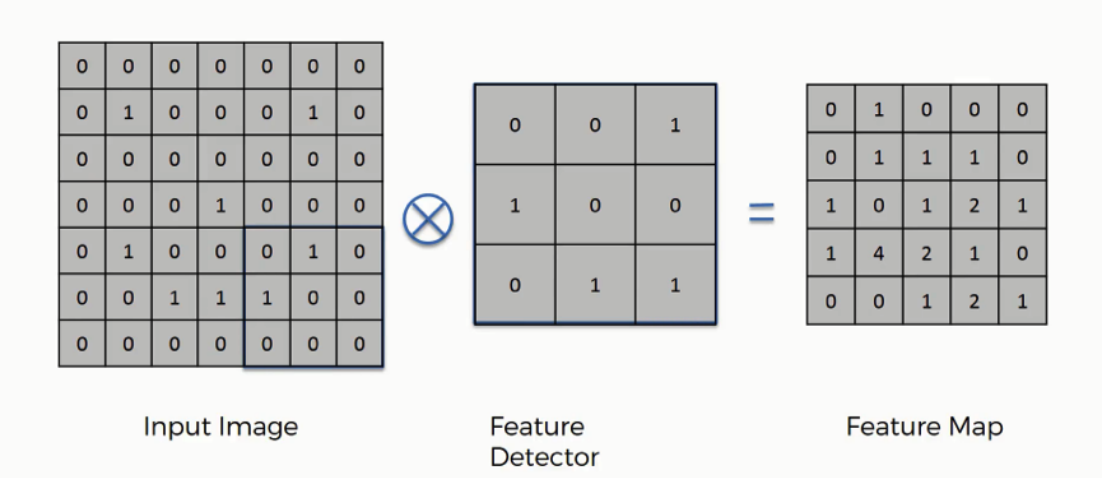
5 **Convolution Neural Network (CNN)**

Reference:https://medium.com/datadriveninvestor/convolution-neural-networks-vs-fully-connected-neural-networks-8171a6e86f15

Both convolution neural networks and neural networks have learn able weights and biases. In both networks the neurons receive some input, perform a dot product and follows it up with a nonlinear function like ReLU(Rectified Linear Unit).

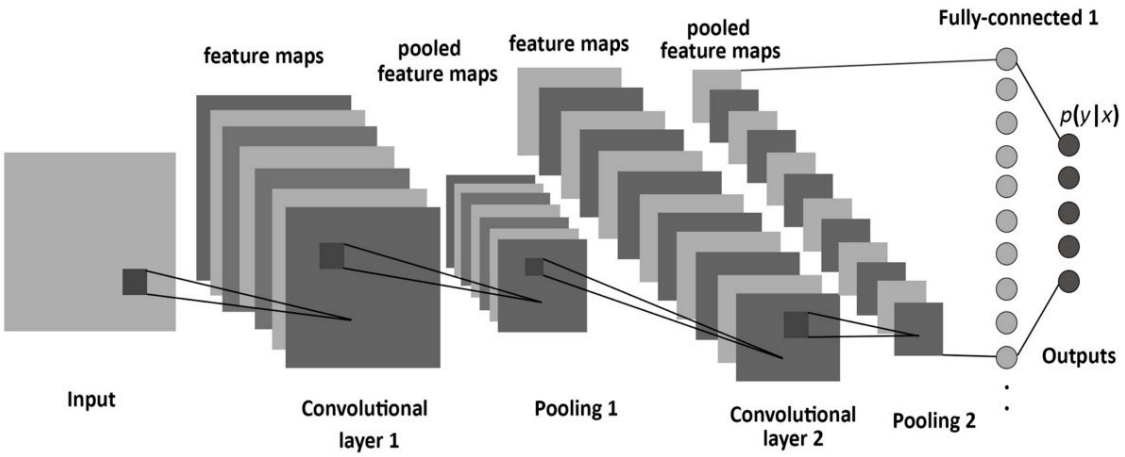
A fully connected neural network for image with dimension 64*64*3 will have 12288 weights in the first layer itself. The neural network will face several problems like slower training time, chance of overfitting.
There are additional components involved in CNN which makes it perform well for image classifications:

1. **Input layer** - a single raw image is given as an input. For a RGB image its dimension will be AxBx3, where 3 represents the colours Red, Green and Blue
2. **Convolution layer** - a convolution layer is a matrix of dimension smaller than the input matrix. It performs a convolution operation with a small part of the input matrix having same dimension. The sum of the products of the corresponding elements is the output of this layer.



3. **ReLU/Rectified Linear Unit** - ReLU is mathematically expressed as max(0,x). It means any number below 0 is converted to 0 while any positive number is allowed to pass as it is
4. **Maxpool** - Maxpool passes the maximum value from amongst a small collection of elements of the incoming matrix to the output. Usually it is a square matrix.
5. **Fully connected layer** - The final output layer is a normal fully-connected neural network layer, which gives the output.
   Usually the convolution layers, ReLUs and Maxpool layers are repeated number of times to form a network with multiple hidden layer commonly known as deep neural network.

Group 7 (Aditi Asthana, Sidharth Pati and Stuti Shukla)



For convolution neural network, the accuracy observed is 98.7% and time taken is 12:48 minutes.
Please note that the above results were obtained when we ran the code in Google Colab.