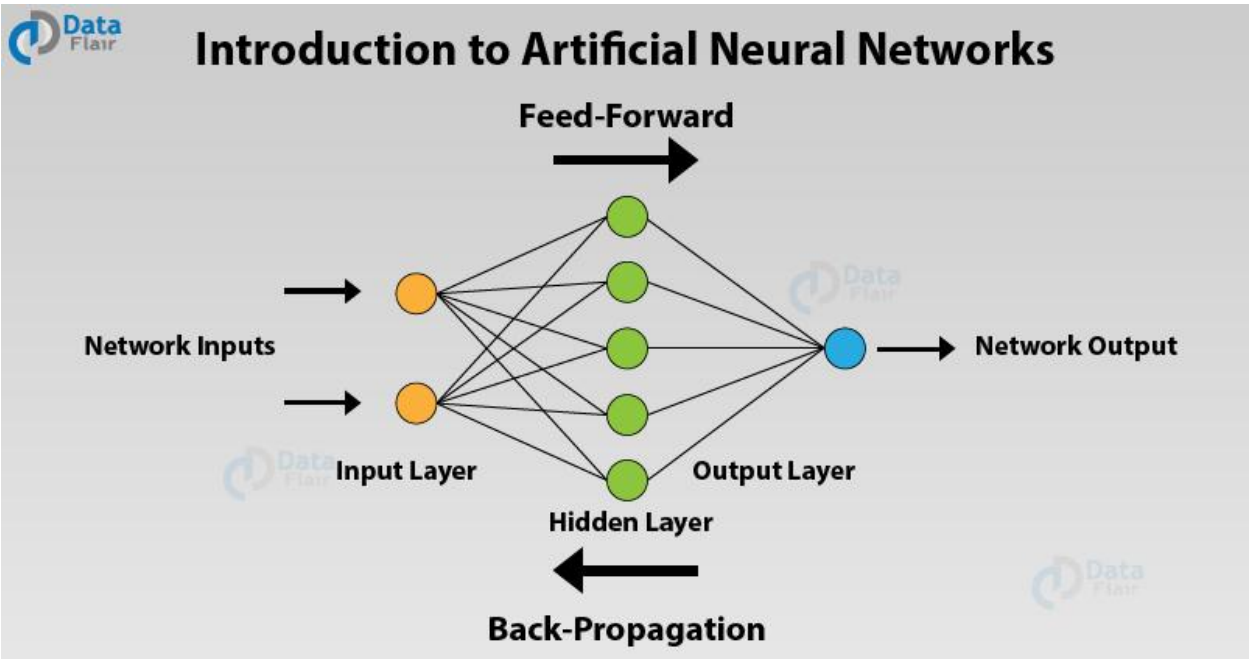
2B - Artificial Neural Networks

1. Brief description of the model and its implementation:



Artificial Neural Networks are a particular type of machine learning algorithms that are modelled after the human brain. Like how the neurons in our nervous system can learn from past data, the ANN can learn from the data and provide responses in the form of predictions or classifications.

ANNs are nonlinear statistical models which display a complex relationship between the inputs and outputs to discover a new pattern. A variety of tasks such as image recognition, speech recognition, machine translation, and medical diagnosis make use of these artificial neural networks.

In a neural network, there are three essential layers –

Input Layers

The *input layer* is the first layer of an ANN that receives the input information in various texts, numbers, audio files, image pixels, etc.

Hidden Layers

In the middle of the ANN model are the *hidden layers*. There can be a single hidden layer, as in a perceptron or multiple hidden layers. These hidden layers perform various types of mathematical computation on the input data and recognise the patterns that are part of it.

Output Layer

In the *output layer*, we obtain the result through rigorous computations performed by the middle layer.

Implementation:

a. We scaled the dataset using min-max-scaling.
b. We randomised the dataset.
c. We split the dataset in a 70:30 split.
d. We applied mini-batch gradient descent with batch size varying from 100 to 150. We varied the learning rate from 0.05 to 0.2
e. We calculated the accuracy and loss for the models.

2. A brief description of the chosen hyper-parameters for the model:

We are using rectified linear units as our activation functions in this ANN. We have run it with three different learning rates - 0.1, 0.2 and 0.05. The batch size was set to 100, which is the size of each mini-batch involved in gradient descent.

There is also an attempt at using stochastic gradient descent at the end for this model using the same function (we simply restrict the batch size to 1).

3. **The final train and test metrics (loss and accuracy) achieved by the model for ANN with one hidden layer and two hidden layers:**

For Neural Network with 1 Hidden Layer

Accuracy for training = 0.7442857142857143

Accuracy for test = 0.76

Loss for training = 0.6301883498153874

Loss for test = 0.6125326976501727

For Neural Network with two hidden layers

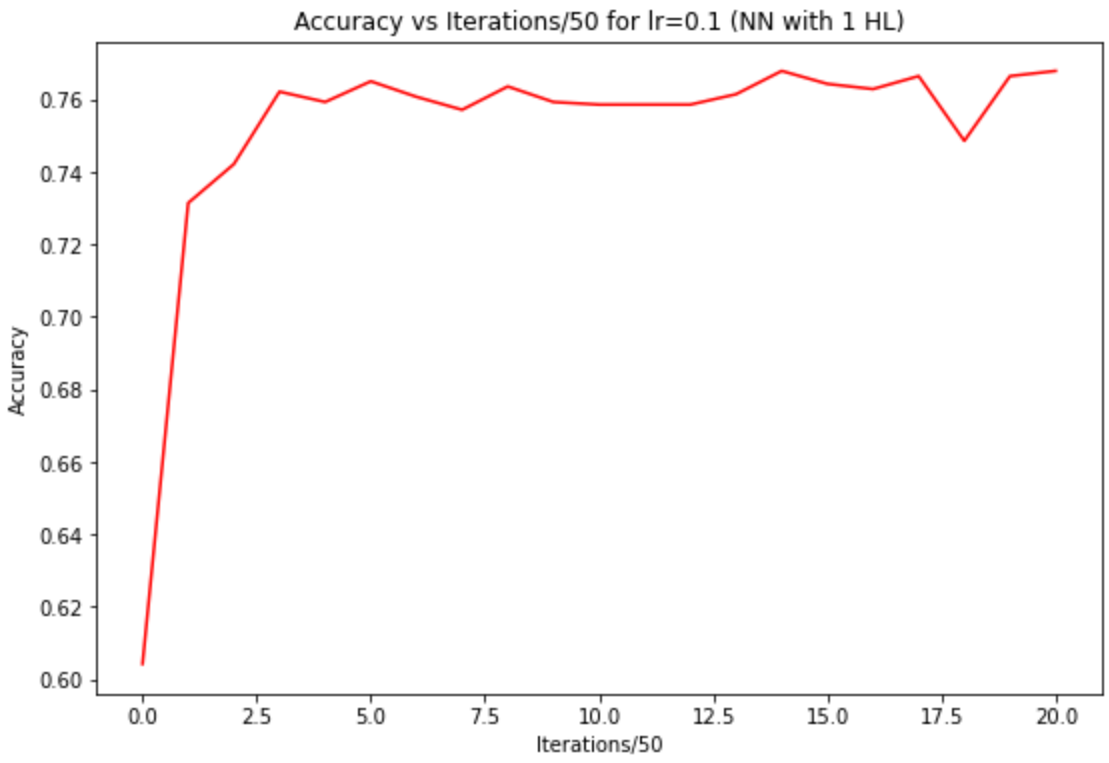
Accuracy for training = 0.6714285714285714

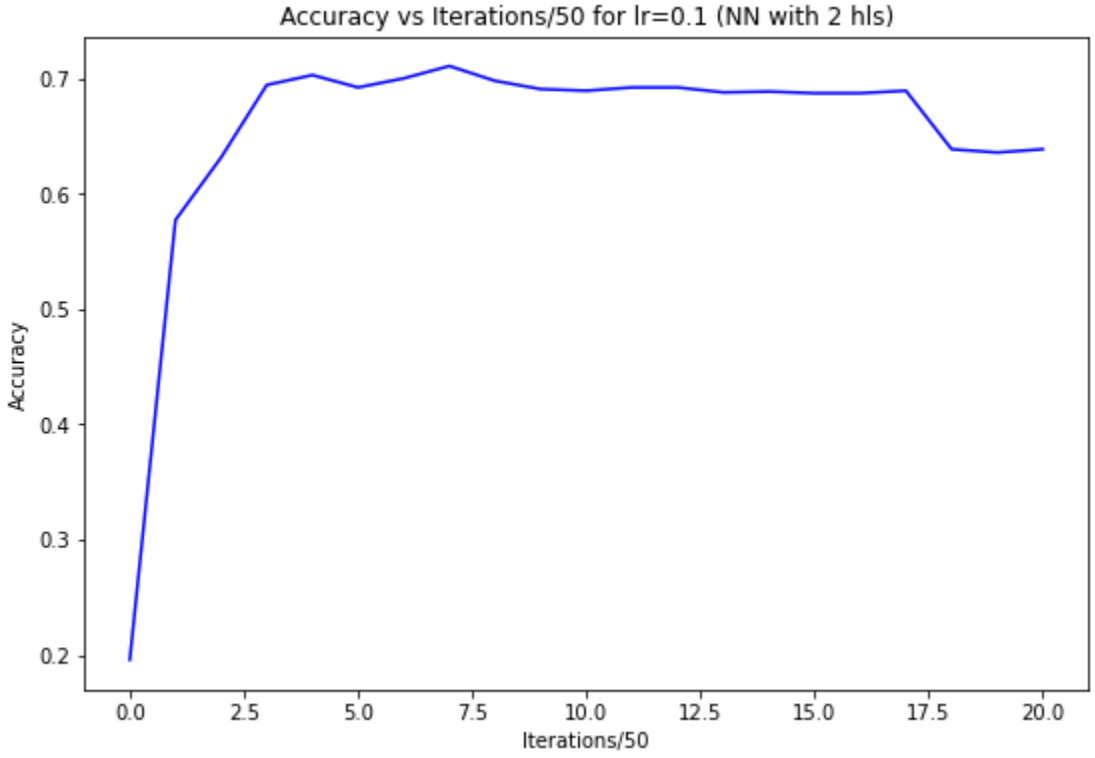
Accuracy for test = 0.685

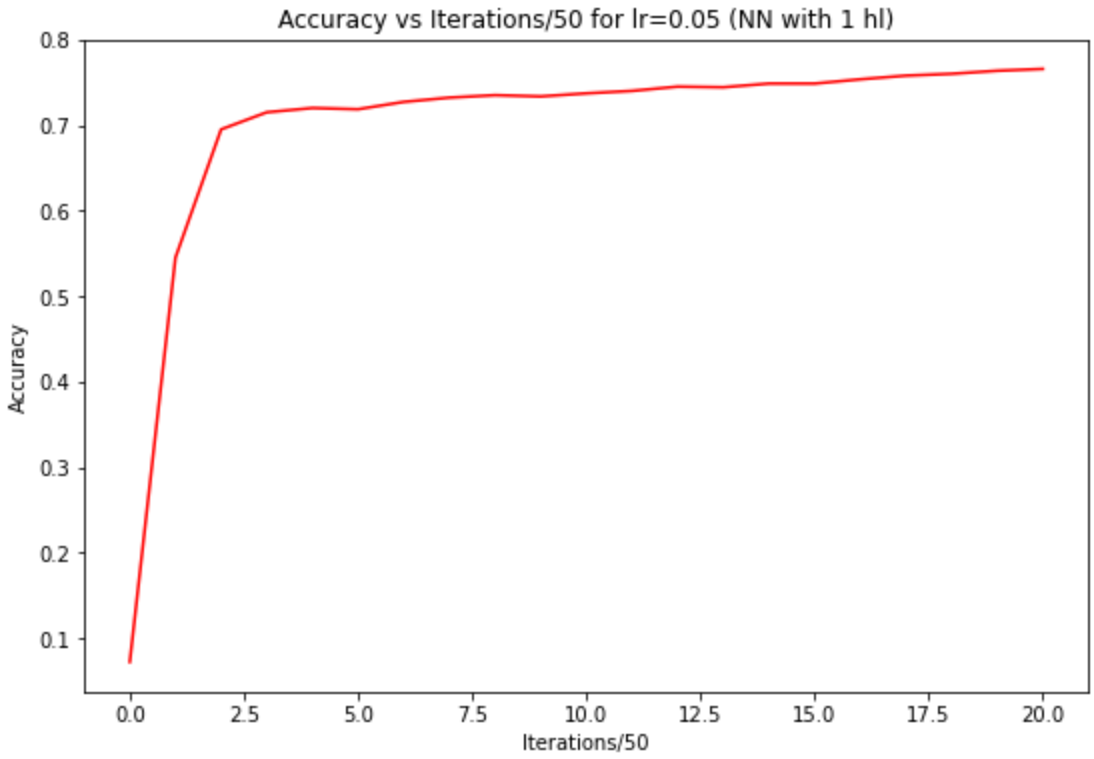
Loss for training = 0.9756736481443982

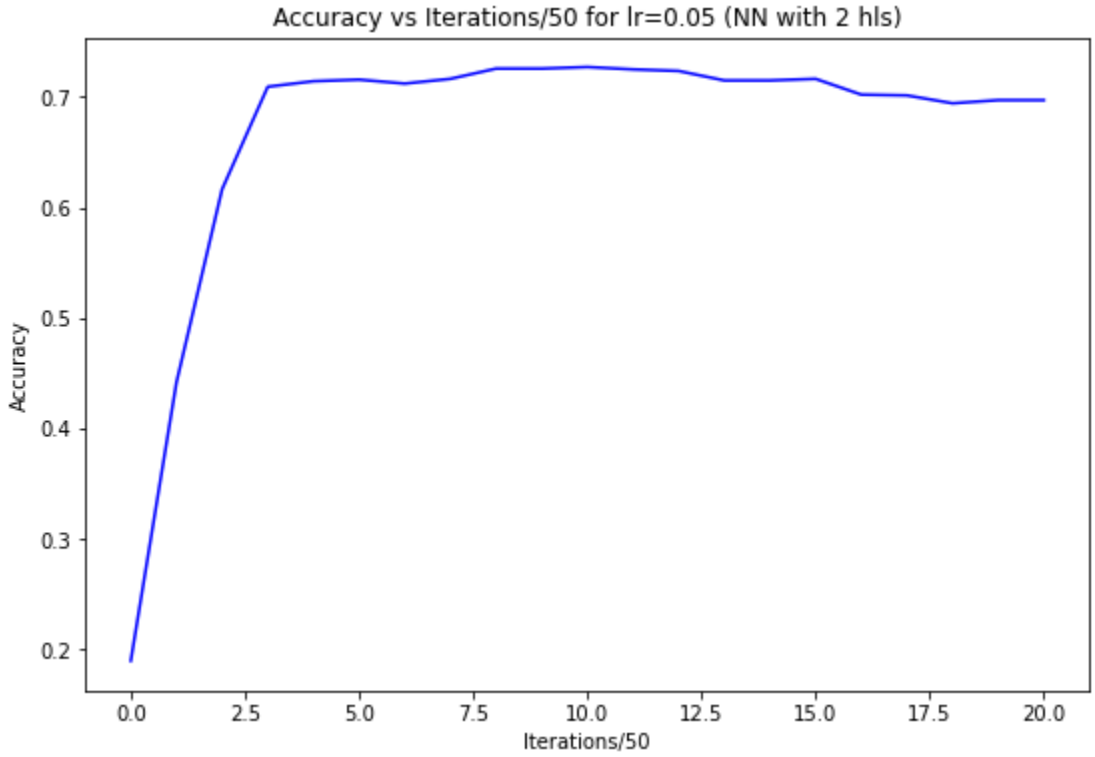
Loss for test = 1.0690141772728463

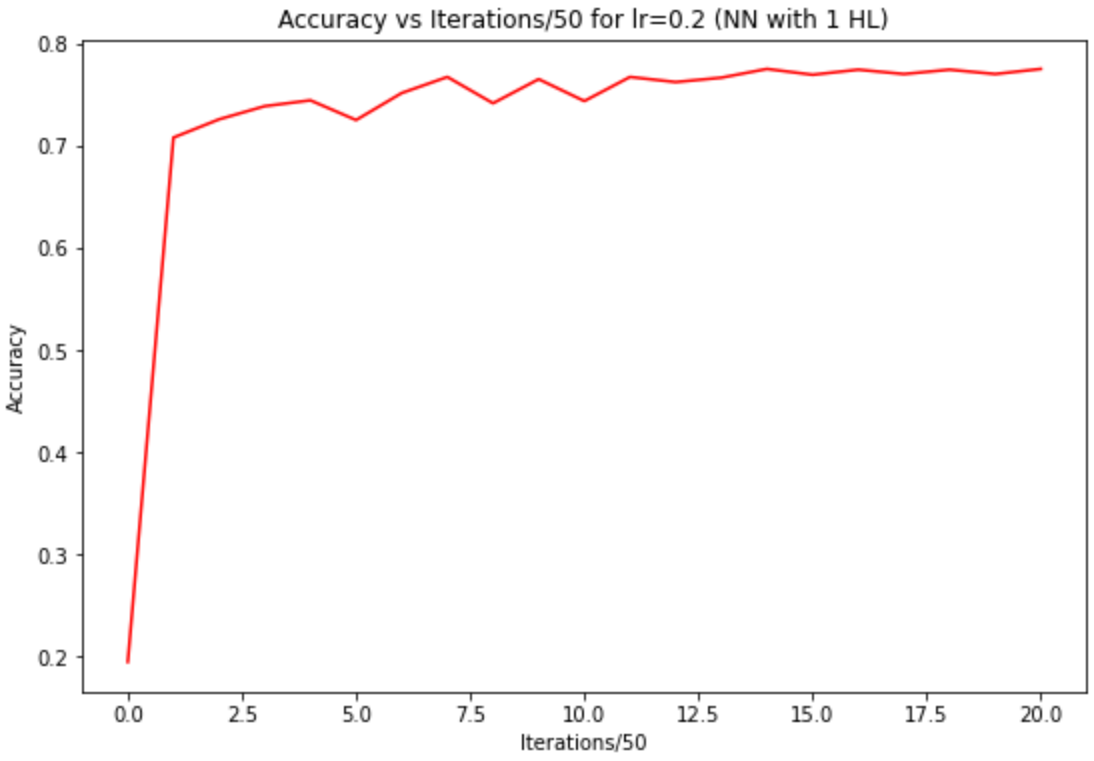
4. **Plots of accuracy for three different learning rates for ANN with one hidden layer and two hidden layers:**

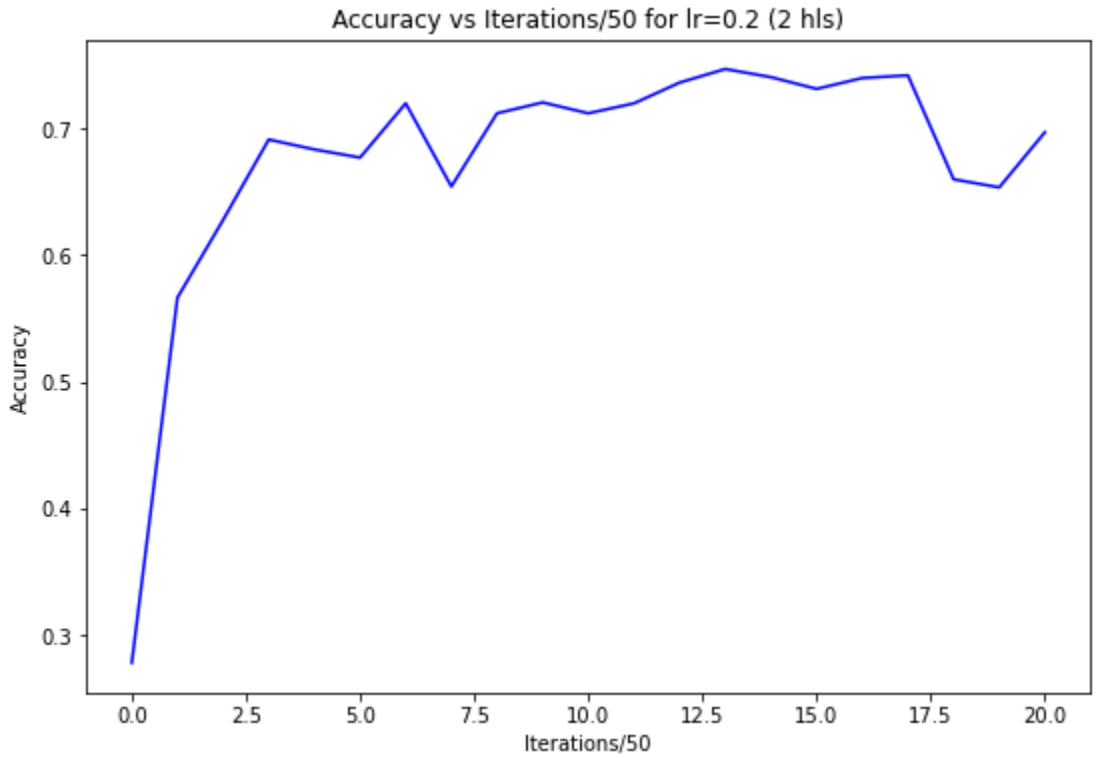

Accuracy vs Iterations/50 for lr=0.1 (NN with 1 HL)
Accuracy
Iterations/50
0.60
0.62
0.64
0.66
0.68
0.70
0.72
0.74
0.76
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Accuracy vs Iterations/50 for lr=0.1 (NN with 2 hls)
Accuracy
Iterations/50
0.2
0.3
0.4
0.5
0.6
0.7
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Accuracy vs Iterations/50 for lr=0.05 (NN with 1 hl)
Accuracy
Iterations/50
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Accuracy vs Iterations/50 for lr=0.05 (NN with 2 hls)
Accuracy
Iterations/50
0.7
0.6
0.5
0.4
0.3
0.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Accuracy vs Iterations/50 for lr=0.2 (NN with 1 HL)
Accuracy
Iterations/50
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Accuracy vs Iterations/50 for lr=0.2 (2 hls)
Accuracy
Iterations/50
0.7
0.6
0.5
0.4
0.3
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0