Assignment 2B

Artificial Neural Networks

Brief Description of the Model

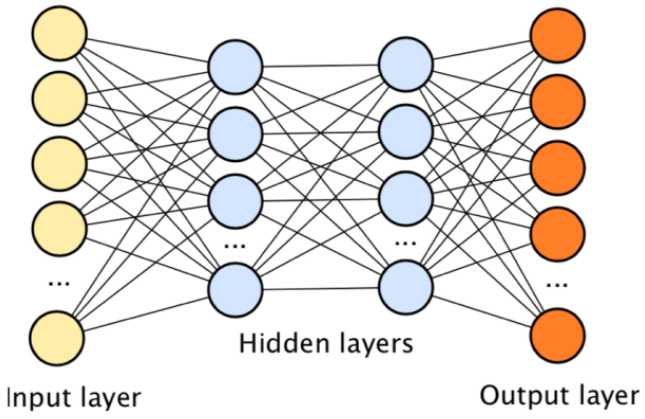
Artificial Neural Network is a computational model that is inspired by the way biological neural networks in the human brain process information. They are efficient data-driven modelling tools widely used for nonlinear systems dynamic modelling and identification, due to their universal approximation capabilities and flexible structure that allow them to capture complex nonlinear behaviors. Feed-forward multi-layer perceptron ANNs type is frequently used in engineering applications. Some of the steps involved in the implementation are -

Forward Propagation is the technique to generate predictions during training that will need to be corrected, and it is the method we need after the network is trained to make predictions on new data.
We can break forward propagation down into three parts:

- Neuron Activation.
- Neuron Transfer.
- Forward Propagation.

In the Backpropagation step, the error is calculated between the expected outputs and the outputs forward propagated from the network. These errors are then propagated backward through the network from the output layer to the hidden layer, assigning blame for the error and updating weights as they go.

We then train the network based on the parameters and measure accuracies based on the predictions by the models



Network Architecture

The parameters and hyper-parameters we used for the models are:

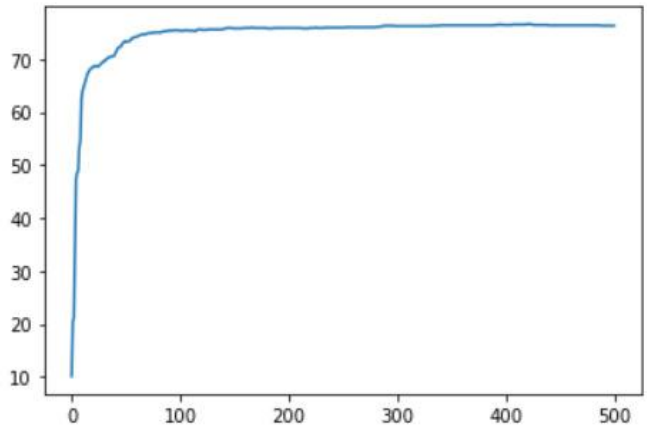
- Number of neurons:
    - Input Layer: 6
    - Hidden Layers: 8
    - Output Layer: 10
- Number of hidden Layers: 1 and 2
- Types of networks: 6-8-10 and 6-8-8-10
- Activation function used: Sigmoid and Softmax
- Loss function: Sum of Squared Error (SSE)
- Gradient Descent: Stochastic Gradient Descent
- Learning rates: 0.3, 0.1, 0.05
- Epochs: 500

Training Accuracy and Loss Plots for every 50-iterations are in the “plots” folder

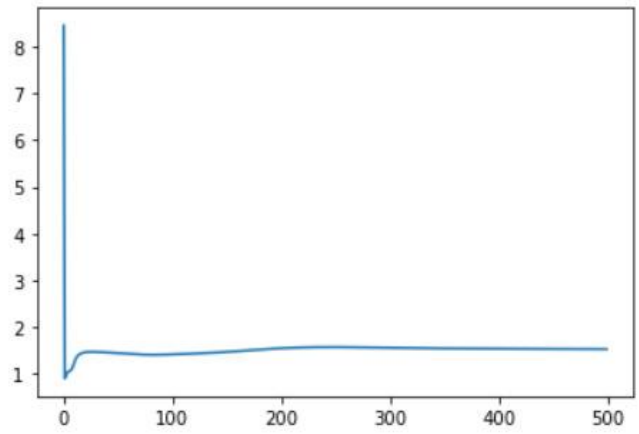
ANN Model Results using 1 hidden layer

1. Learning Rate: 0.3

Training Plots



Accuracy vs Number-of-Epochs



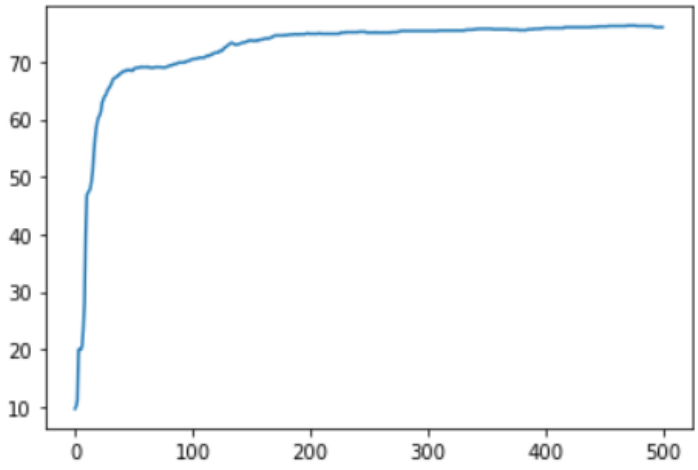
Loss vs Number-of-Epochs

Final Scores

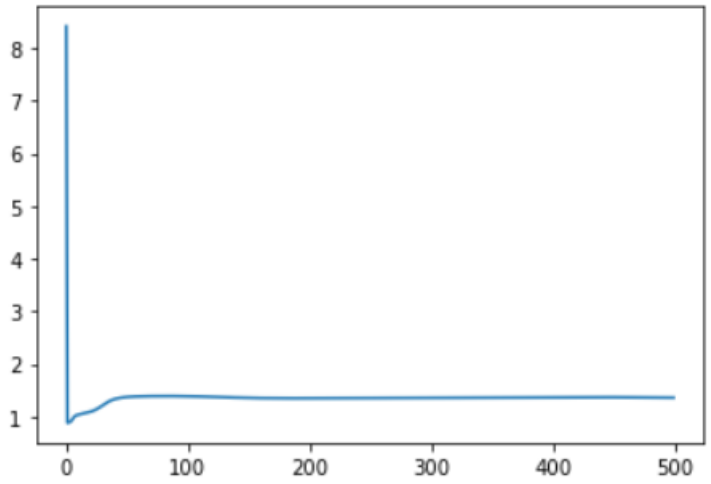
Training Accuracy : 76.285%
Test Accuracy : 74.667%

2. Learning Rate = 0.1

Training Plots



Accuracy vs Number-of-Epochs



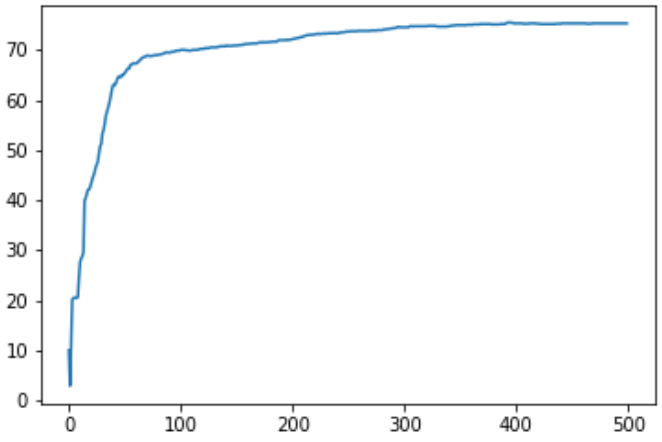
Loss vs Number-of-Epochs

Final Scores

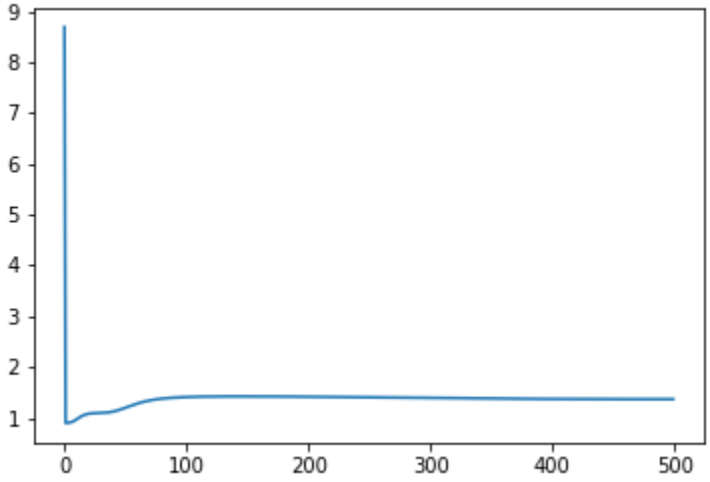
Training Accuracy: 76.0%
Test Accuracy: 73.833%

3. Learning Rate = 0.05

Training Plots



Accuracy vs Number-of-Epochs



Loss vs Number-of-Epochs

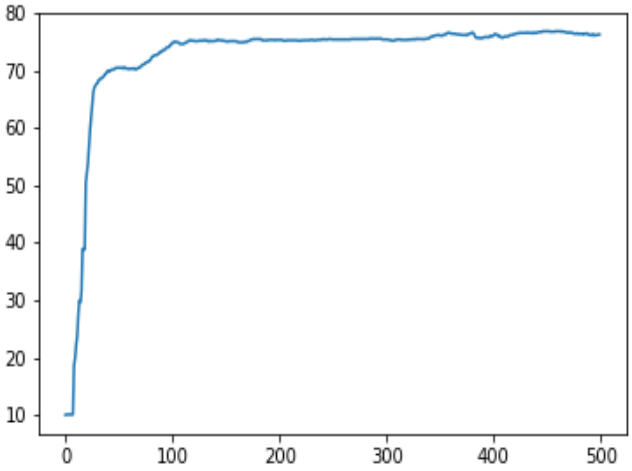
Final Scores

Training Accuracy: 75.285%
Test Accuracy: 73.5%

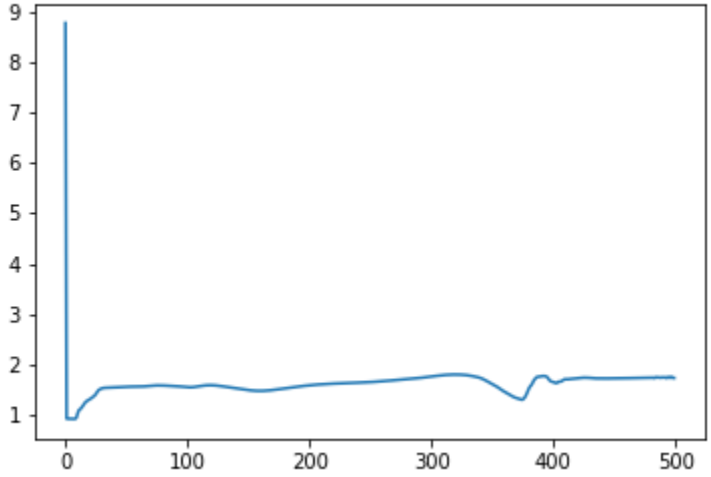
ANN Model Results using 2 hidden layers

1. Learning Rate = 0.3

Training Plots



Accuracy vs Number-of-Epochs



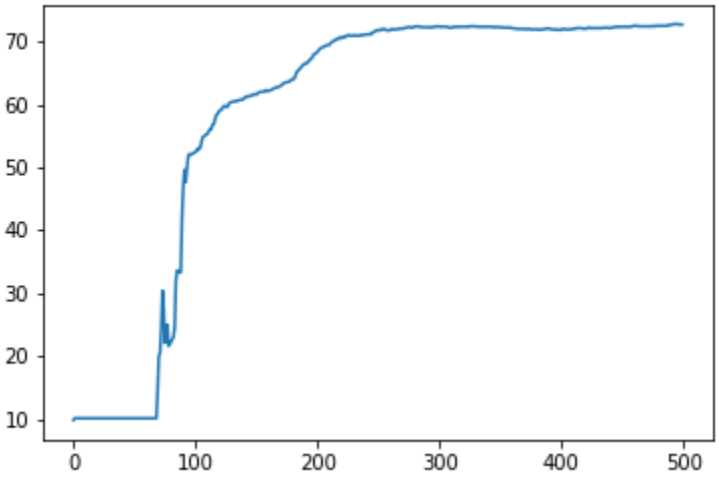
Loss vs Number-of-Epochs

Final Scores

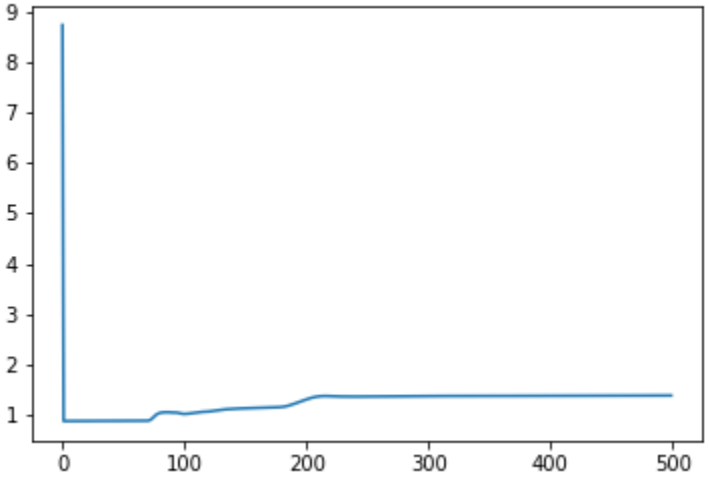
Training Accuracy:	76.071%
Test Accuracy:	73.0%

2. Learning Rate = 0.1

Training Plots



Accuracy vs Number-of-Epochs



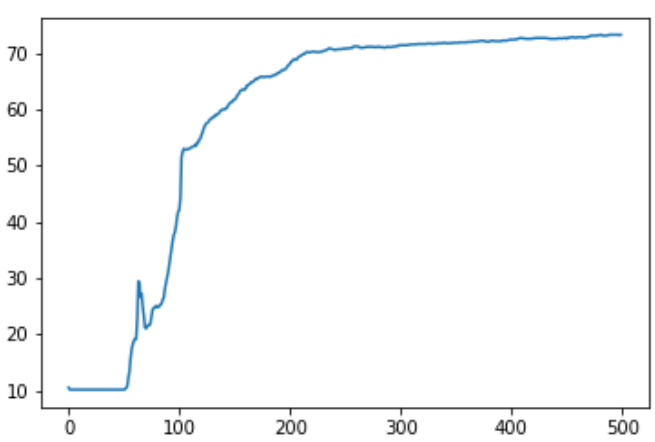
Loss vs Number-of-Epochs

Final Scores

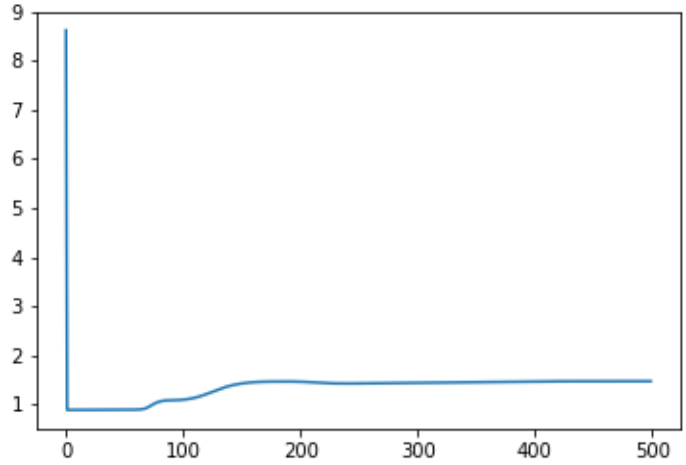
Training Accuracy:	74.714%
Test Accuracy:	73.0%

3. Learning Rate=0.05

Training Plots



Accuracy vs Number-of-Epochs



Loss vs Number-of-Epochs

Final Scores

Training Accuracy:	73.357%
Test Accuracy:	72.167%

Conclusion

We observe that the accuracies for different architectures differ based on the number of hidden layers, learning rate and number of epochs.
The training and test accuracies are converging which tells us that the model is able to fit the data well and may reach higher accuracy when given more data or running for more number of epochs.

The best Test accuracy achieved is **74.667%** using Single Hidden Layer 6-8-10 Architecture with 0.3 learning rate.