4.7. Experiment No. B-1

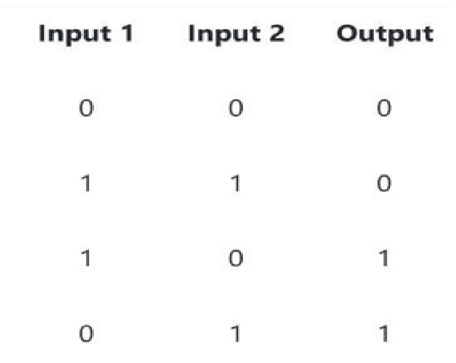
Aim:

Write a python program to show back propagation network for XOR function with Binary Input and Output.

Theory:

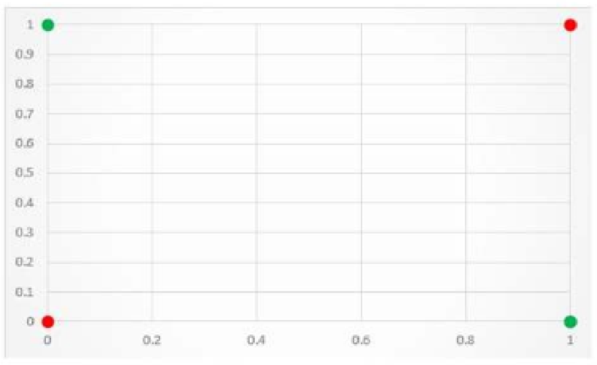
Solving XOR problem with Back Propagation Algorithm

XOR or Exclusive OR is a classic problem in Artificial Neural Network Research. An XOR function takes two binary inputs (0 or 1) & returns True if both inputs are different & False if both inputs are same.



Input 1	Input 2	Output
0	0	0
1	1	0
1	0	1
0	1	1

On the surface, XOR appears to be a very simple problem, however, Minksy and Papert (1969) showed that this was a big problem for neural network architectures of the 1960s, known as perceptrons. A limitation of this architecture is that it is only capable of separating data points with a single line. This is unfortunate because the XOR inputs are not linearly separable. This is particularly visible if you plot the XOR input values to a graph. As shown in the figure, there is no way to separate the 1 and 0 predictions with a single classification line.



Solution

The backpropagation algorithm begins by comparing the actual value output by the forward propagation process to the expected value and then moves backward through the network, slightly adjusting each of the weights in a direction that reduces the size of the error by a small degree. Both forward and back propagation are re-run thousands of times on each input combination until the network can accurately predict the expected output of the possible inputs using forward propagation.

Algorithm:

Step1: Import the required Python libraries
Step2: Define Activation Function : Sigmoid Function
Step3: Initialize neural network parameters (weights, bias) and define model hyper parameters (number of iterations, learning rate)
Step4: Forward Propagation
Step5: Backward Propagation
Step6: Update weight and bias parameters
Step7: Train the learning model
Step8: Plot Loss value vs Epoch
Step9: Test the model performance

Input:

input vector {x} : (l{x_1, {x_2})

Output:

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Conclusion:

Hence, the model predicted output for each of the test inputs are exactly matched with the XOR logic gate conventional output, according to the truth table and the cost function is also continuously converging.

Outcome:

Upon completion of this experiment, students will be able to:

Experiment level outcome (ELO1): Students will be able to model a neural network to separate linearly non-separable input patterns using decision plane.

Questions:

1. What is Back propagataion for XOR?
2. How do you solve XOR with a neural network?
3. How can we design a neural network that acts as an XOR gate?
4. What are the properties of XOR gate?
5. What is a real life example of XOR?