**Multi-Layer Implementation from Scratch**

Description

In this implementation, I am implementing the Multi Layer Perceptron with one hidden layer having two perceptron(s) using R. Using this MLP we are trying to fit the sine curve with added noise.

Solution

In this lab, we start with first generating a sine curve with added random noise. Then, we create and train multi layer perceptron with two hidden units in one hidden layer and having sigmoid activation function. Finally, training the multi-layer perceptron using back propagation algorithm. The figure below shows the structure of the MLP. It has a single input X1 and a bias X0 which is equal to 1. The hidden layer which converts linear inputs to non-linear sigmoid function are z0, z1 and z2 where z0 is again the bias which is equal to 1. We find the values of the weights W01, W02, W11, W12, V0, V1 and V2 using the back-propagation algorithm.

V0

W01

V1

W02

W11

V2

W12

Results

We ran the code for 100, 300 and 500 iterations. As, we can see from the plot below that two hidden units are not able to properly fit the sine curve. We might need more hidden layers or hidden units to fit the sine curve properly.

