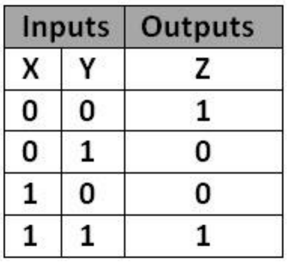
Analysis Report for Back Propagation In Multilayer Perceptron (XNOR)

- Ratnesh Kumar Rai(MIT2019098)

Dataset:

We have to implement the X-NOR gate using multilayer perceptron neural network via back propagation. For this we have training data as the truth table for XNOR Gate with two input and 1 output:



Inputs		Outputs
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1

As we can XOR data is not linear, so a single perceptron will not be able to classify the given problem, thus we will use multilayer perceptron.

Implementation:

For the purpose assignment we have used:

- 2 -Hidden Layers, each having 3 neurons(nodes)
- 1 – Output Layer, having only one node.
- Activation Function: Sigmoid Function
- Adaptive Learning Rate
- Optimiser: Stochastic gradient descent
- Convergence Point: when squared sum error < 0.011
- Number of epochs: 10,000
- Initial learning rate 0.3

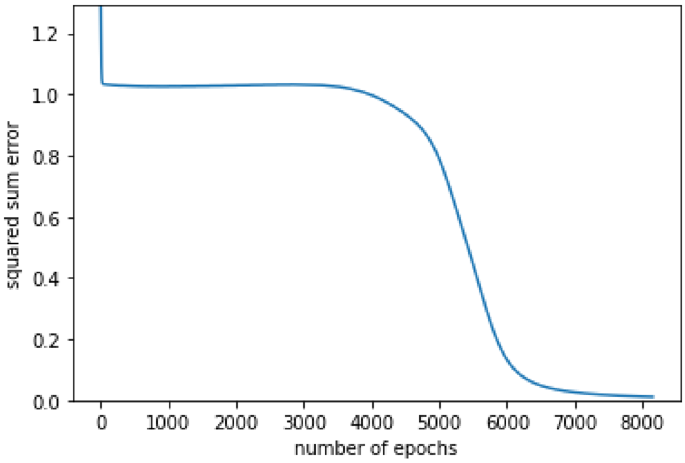
Steps:

1. We first initialised our network with (2 nodes in Input layer, 3 nodes in each hidden layer and 1 node in output layer) by randomly assigning weights and bias(threshold) between -0.5 to 0.5 of each node.
2. 1st pass is Forward Propagation.
3. Then, we applied backward propagation for the error in output layer storing the new delta(error) at each node and passing it back to the previous layer.
4. Then we updated the weights according to calculated delta (or error) in backward propagation.
5. Applied adaptive learning rate for faster convergence.
6. Repeat From step 2 until convergence is achieved.

Result:

We achieved convergence at 8,152 epochs, with initial learning rate as 0.3 and adaptive increase or decrease in learning by 0.005.

Here is the graph showing the how sum squared error changed over number of epochs.



Q) How you will verify your trained algorithms? Justify your solution.
Sol) First, we are always going to get the correct result as our training dataset contain all possible combination of data, we can skip testing because we will always get result with 100% accuracy.