**Restricted Boltzmann Machine**

**Machine Setup**

The provided excel document is formatted for the training set, but the machine itself can be configured for multiple data sets via the final variables in the header of the main program. These are commented to show what they change. When creating or inserting custom data, be sure the first row of the excel document contains the headers for the columns. This is for readability and the fact that the program will ALWAYS skip the first row of an excel document; the amount of rows and which columns to be read can be customized, but be aware that the columns are 0-indexed, so the ‘A’ column will be column 0.

|  |  |  |  |
| --- | --- | --- | --- |
| L1 | L2 |  | Xor Solution |
| 0 | 0 |  | 0 |
| 0 | 1 |  | 1 |
| 1 | 0 |  | 1 |
| 1 | 1 |  | 0 |

Ex1. Formatted data

In Ex1, the first row is ignored by the program so these can also be blank, but for readability they should be labeled. To read the data into the program, the NUMRECORDS variable should be set to 4, COLUMNS should be set to {0,1,3}, and NUMSHEET should be set to 1 (this is because this data set is located on the 2nd sheet in the excel document, and the program is 0-indexed).

**Running the machine**

After compilation, the program will output:

Expected Value: 1.0000 | Before CalcValue: 0.0699 | After CalcValue: 0.1062 | Error: 0.39945

This is the result from each training exercise, the expected output, the neural network’s output before the exercise, then the value after the exercise as well as its error.

After the exercises are done, the program will output:

Trained Tree:

Weight[0]: -0.5326545450063764

Weight[0]: -0.5638327767805152

InputNode

[FinishTree]

Weight[1]: -0.7565569759194543

InputNode

[FinishTree]

Weight[2]: 0.017021888288290196

BiasNode

[FinishTree]

[FinishTree]

Weight[1]: 10.080935459474302

Weight[0]: 0.283296502737596

InputNode

[FinishTree]

Weight[1]: -0.3565727131078822

InputNode

[FinishTree]

Weight[2]: 1.0253389619386106

BiasNode

[FinishTree]

[FinishTree]

Weight[2]: 14.841185985783973

BiasNode

[FinishTree]

This is the output for the entire tree, remember that this is a recursive print from the head of the tree. This shows the weight between each of the nodes in the tree, including the input and bias nodes. Lines highlighted in blue are the weights from the root node, while those in orange are from the first hidden node, and lastly those in yellow are from the second hidden node. Whenever [FinishTree] is printed, the recursive method has reached a node without any children and when it is printed twice it has finished with that branch of the tree. Lastly, it will also print BiasNode or InputNode, showing that it has reached one of these terminal nodes.

Lastly, the program will ask the user for input to test the machine:

Enter two binary values to XOR: 10  
BoltzmanMachine predicts 0.366903 as the XOR ouput.  
Enter two binary values to XOR: 01  
BoltzmanMachine predicts 0.786400 as the XOR ouput.  
Enter two binary values to XOR: 11  
BoltzmanMachine predicts 0.448582 as the XOR ouput.  
Enter two binary values to XOR: 00  
BoltzmanMachine predicts 0.712700 as the XOR ouput.

If you are using the provided example, you must input two binary digits to XOR, the machine will then output the % it has calculated for the value to be activated (1).

**Last words**

The program models an XOR operation using a Restricted Boltzmann Machine, using equations demonstrated in AIFH Vol 3. The initial logic was derived from <http://blog.echen.me/2011/07/18/introduction-to-restricted-boltzmann-machines/>.