Neural Networks Demo using Matlab 6.5.

The XOR problem cannot be solved using Perceptron Method, and it requires one hidden layer & one output layer, since it's NOT linearly separable. Please note that Input Signal is NOT counted as a layer, in Matlab implementation.

Step 1: Design Phrase:

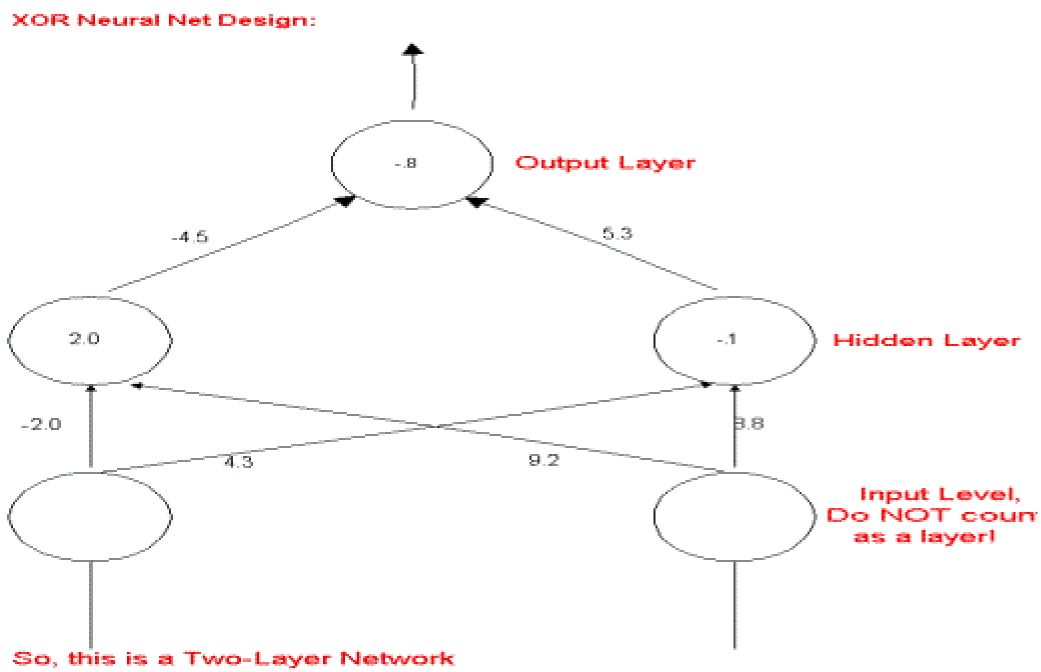

Figure 1

Getting Started:

Open the Matlab Toolbox and look for the Neural Network Toolbox. You should be able to see the NN Network/Data Manager screen (figure 2).

Now, let's look at the truth table of XOR:

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

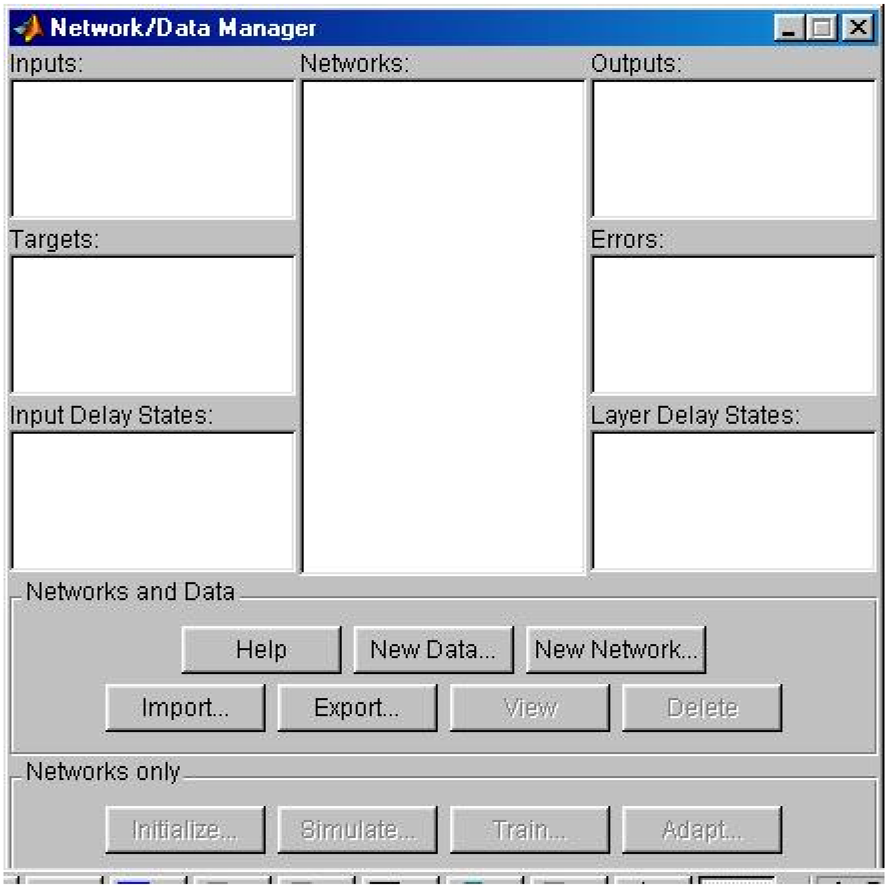

Figure 2

Let P denote the input and T denote the target/output. In Matlab as per the guidelines of implementation these are to be expressed in the form of matrices:

P = [0 0 1 1; 0 1 0 1]
T = [0 1 1 0]

In order to use a network we need to first design it, then train it. After this the network is ready for simulations to be performed on it. We follow the steps in order to do the above:

Design Phase

Step 1 : First we have to enter P and T to the NN Network Manager. This is done by clicking New Data once. You should see a screen like following figure 3:

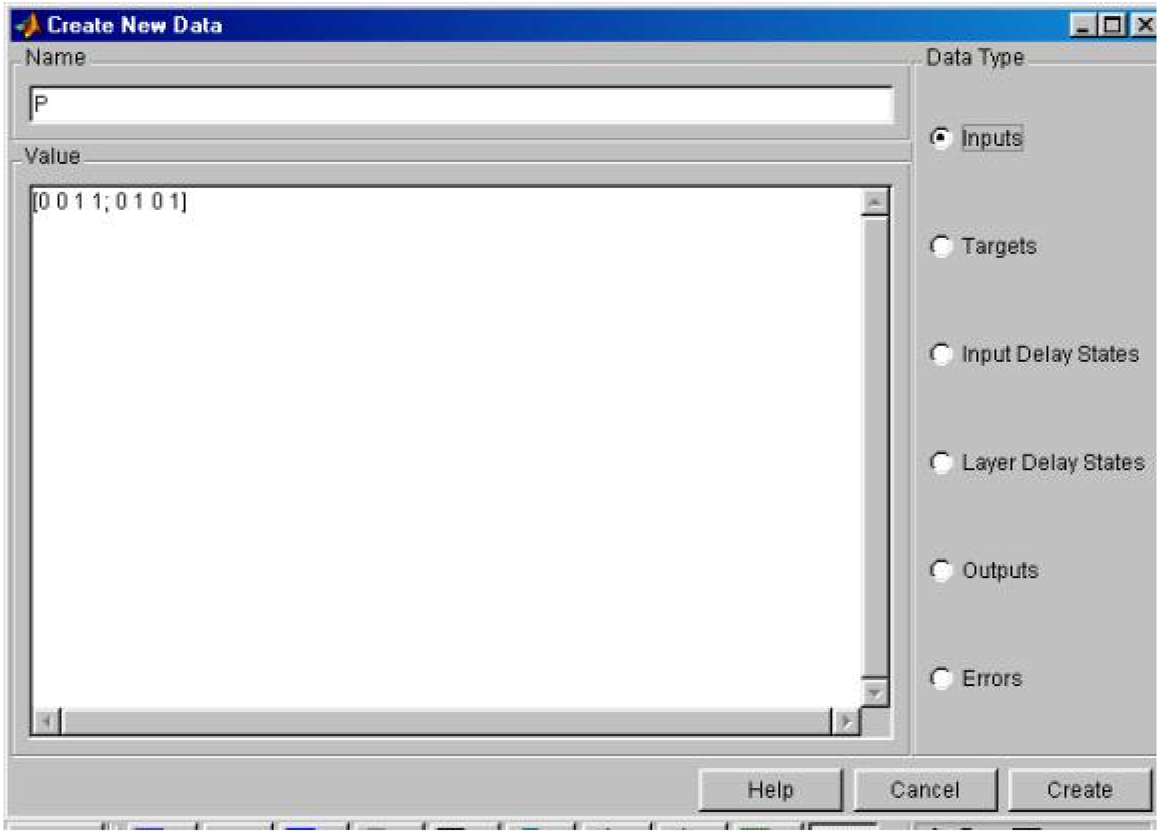

figure 3

Step 2: Type **P** as the Name, and corresponding matrix [0 0 1 1; 0 1 0 1] as the Value, select Inputs under DataType (if it's not already selected), then confirm by clicking on Create.

Step 3: Similarly, type in **T** as the Name, and corresponding matrix =[0 1 1 0] as the Value, select Targets, under DataType , then confirm by clicking Create.

Step 4: Now we try to create a XORNet. For this click on New Network. You should see a screen like in the following figure 4.

Now change all the parameters on the screen to the values as indicated on the following screen:

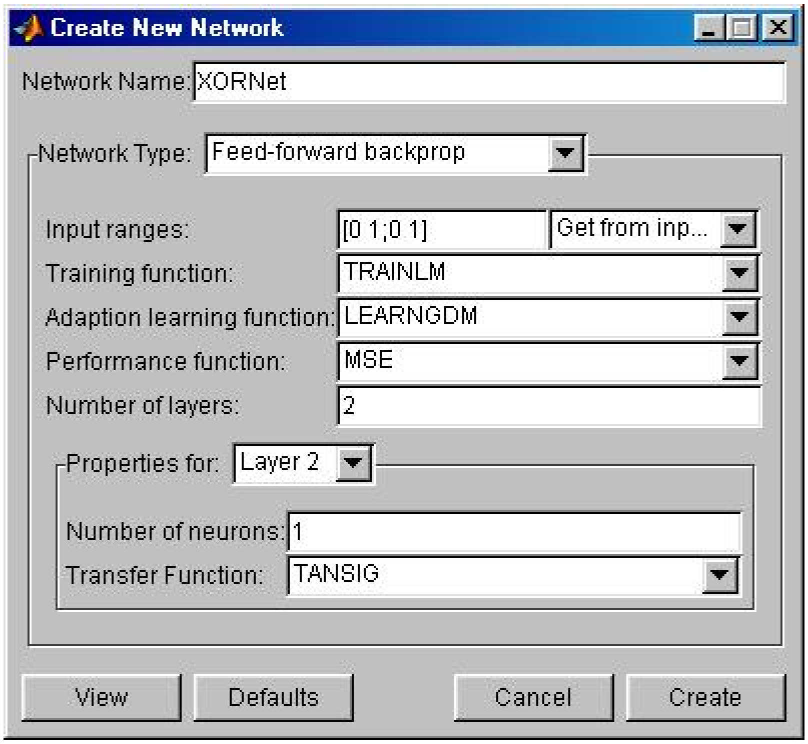

Figure 4

Make Sure the parameters are as follows:

Network Type = Feedforword Backprop
Input Ranges = [0 1; 0 1]
Train Function = TRAINLM
Adaption Learning Function = LEARNGDM
Performance Function = MSE
Numbers of Layers = 2

Step 5: Select Layer 1, type in 2 for the number of neurons, & select TANSIG as Transfer Function.

Select Layer 2, type in 1 for the number of neurons, & select TANSIG as Transfer Function.

Step 6: Then, confirm by hitting the Create button, which concludes the XOR network implementation phrase.

Network Training:

Step 7 : Now, highlight XORNet with ONE click, then click on Train button. You will get the following screen indicated in figure 5:

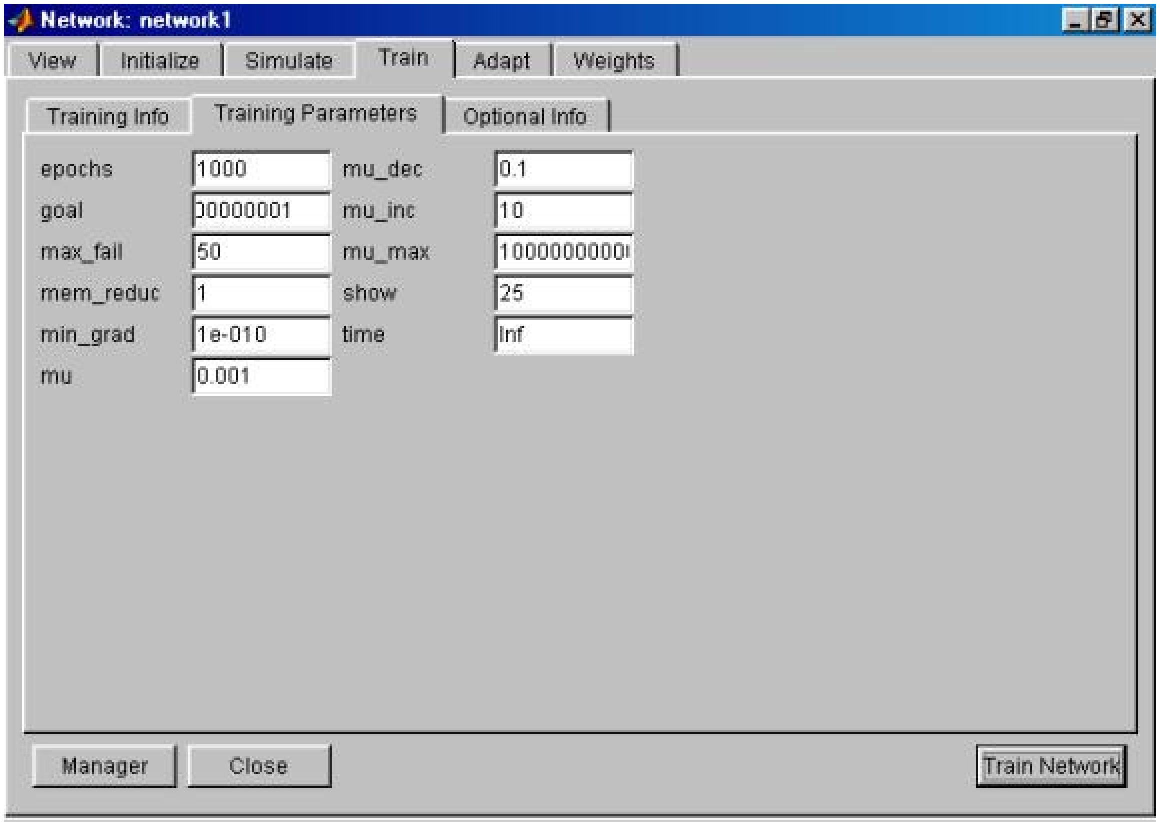

Figure 5

Step 8 : On Training Info, select P as Inputs, T as Targets.

On Training Parameters, specify:

epochs = 1000 (since we would like to train the network for a longer duration).

goal = 0.000000000000001 (since we would like to see if the XORNet that we implemented earlier is capable of producing precise results).

max_fail = 50

After, confirming all the parameters have been specified as indented, hit Train Network. This will give you a training and performance plot.

You should get a decaying plot (since you are trying to minimize the error) similar to that in figure 6, but it may NOT be the exact shape, due to the randomness of the calculation:

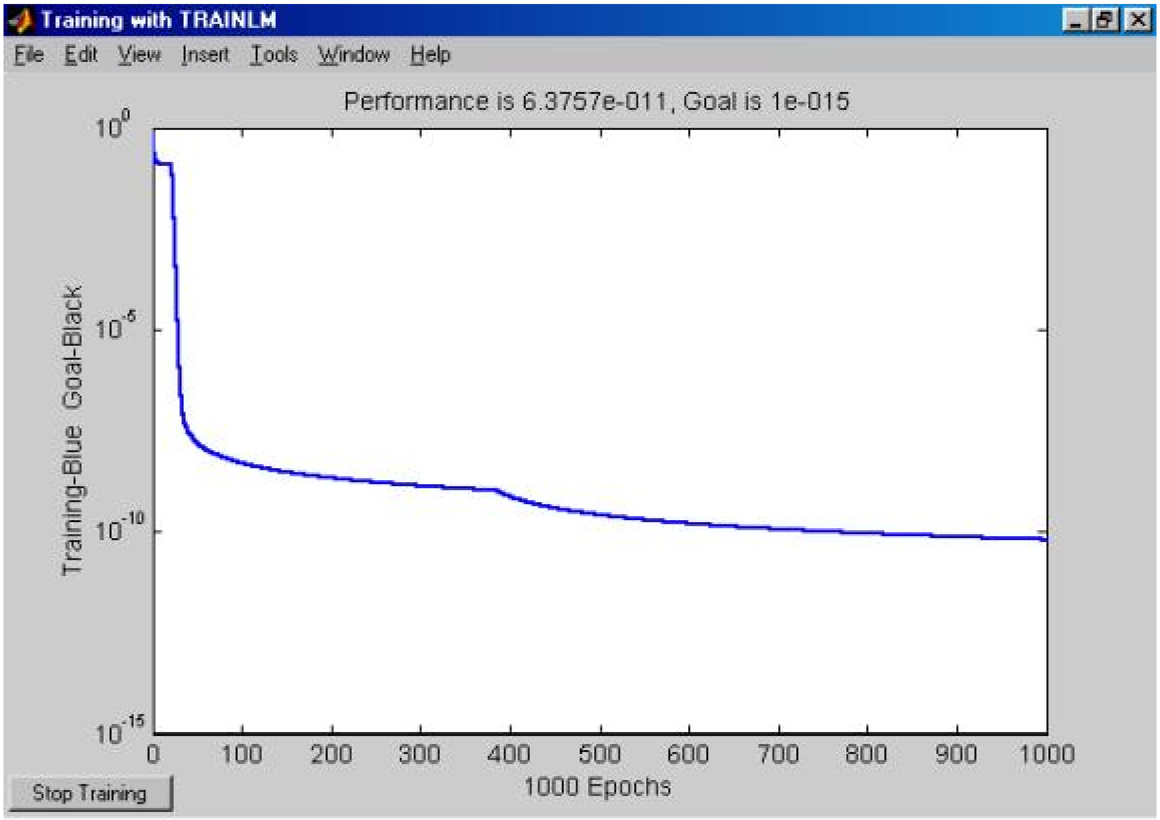

Figure 6

Step 8 : Now we want to confirm the XORNet structure and values of various Weights and Bias of the trained network. For this click on View on the Network/Data Manager window (see figure7).

NOTE : If for any reason, you don't get the figure as expected, click on Delete and recreate the XORNet and follow the procedure indicated in the proceeding steps.

Now, the XORNet has been trained successfully and is ready for simulation if needed.

Network simulation:

With trained network, simulation is a way of testing on the network to see if it meets our expectation.

Step 9 : Now, create a new test data **S** (with a matrix [1; 0] representing a set of two inputs) on the NN Network Manager, follow the same procedure indicated before (like for input P).

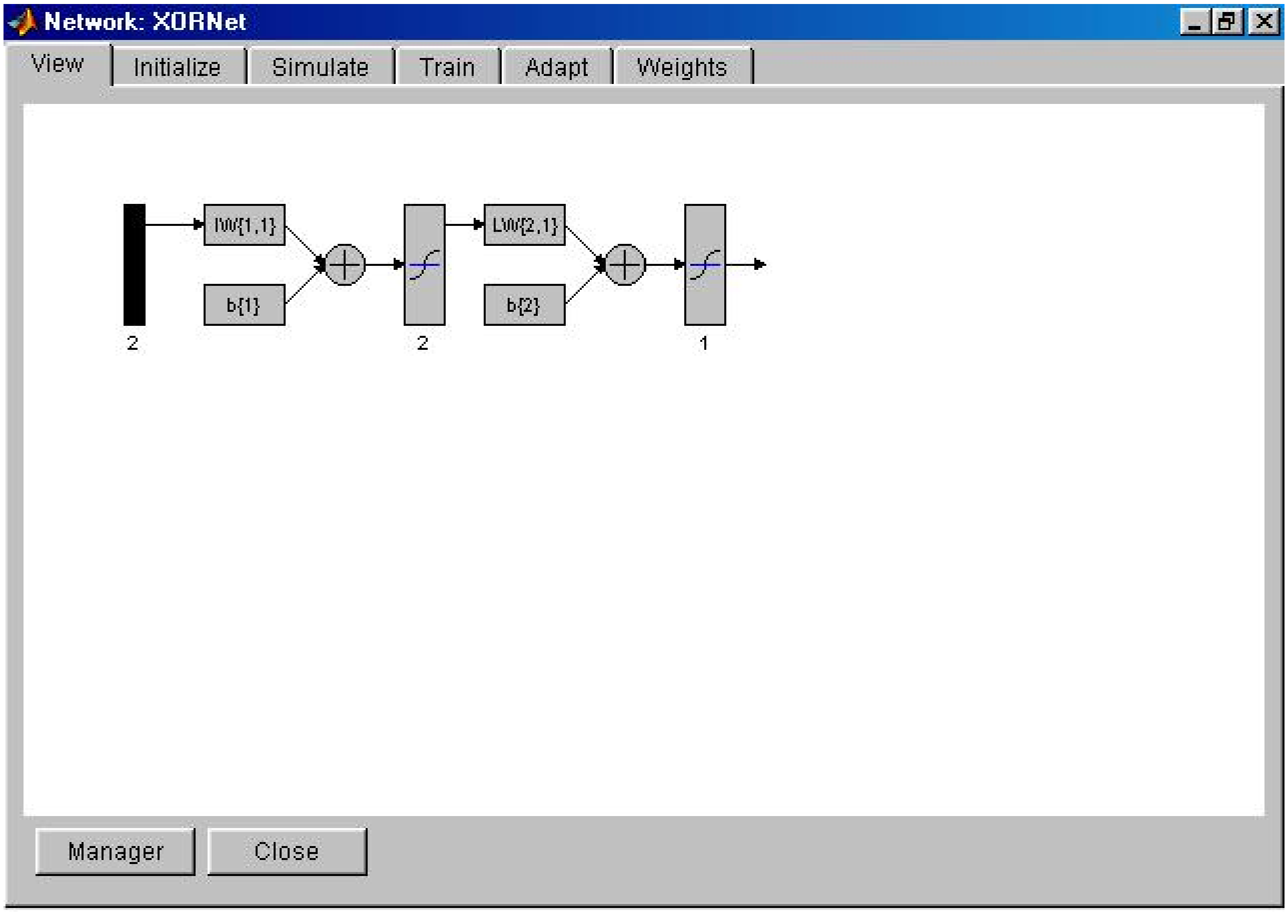

Figure 7

Step 10 : HighLight XORNet again with one click, then click on the Simulate button on the Network Manager. Select S as the Inputs, type in XORNet_outputsSim as Outputs, then hit the Simulate Network button and check the result of XORNet_outputSim on the NN Network Manager, by clicking View.

This concludes the whole process of XOR network design, training & simulation.

Exercise : Now try to train the network with the data that was given to you in the Neural Networks exercise .

For more reference, visit http://www.mathworks.com/ & http://www.mathtools.net/.

Parts taken from document prepared By: Y. Michael Jiang, City College of New York, Feb. 4, 2001, michael212@onebox.com.