

## Exercise: Implementing XOR in Matlab

1. To be able to work in Matlab, you'll first need to store in your workspace the data variables required for your application. Matlab can receive data in different format, including text file, and MS Excel file.

To build your input and output files in text files:

In your Input file, save

0 0 1 1

0 1 0 1

In your output file, save

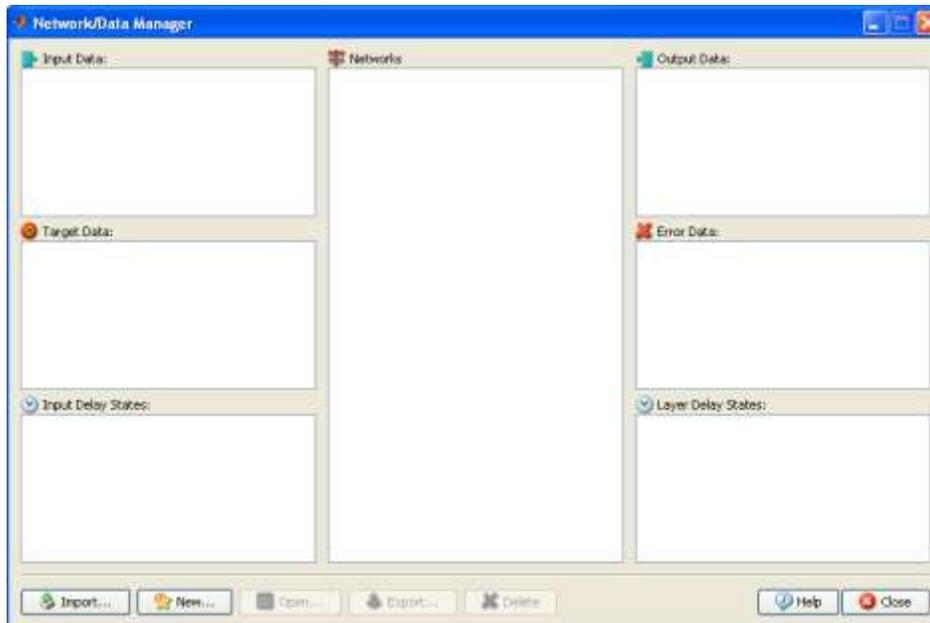
0 1 1 0

|           |   |   |   |   |
|-----------|---|---|---|---|
| <b>X1</b> | 0 | 0 | 1 | 1 |
| <b>X2</b> | 0 | 1 | 0 | 1 |
| <b>Y</b>  | 0 | 1 | 1 | 0 |

Click **File/Import Data** to import your input and output data, one at a time, from you local drive. After this you'll see the matrix of your variables under **Workspace**.

2. To begin using the ANN Tool in Matlab, you can either a) click **Start/Toolboxes/Neural Network/Neural Network Tool**, or b) type **nntool** in the Command Window.

In the below **Network/Data Manager** window (it may be slightly different depends on the version of your Matlab), you can input the data into the Workspace of Matlab, which can then be used by your ANN.



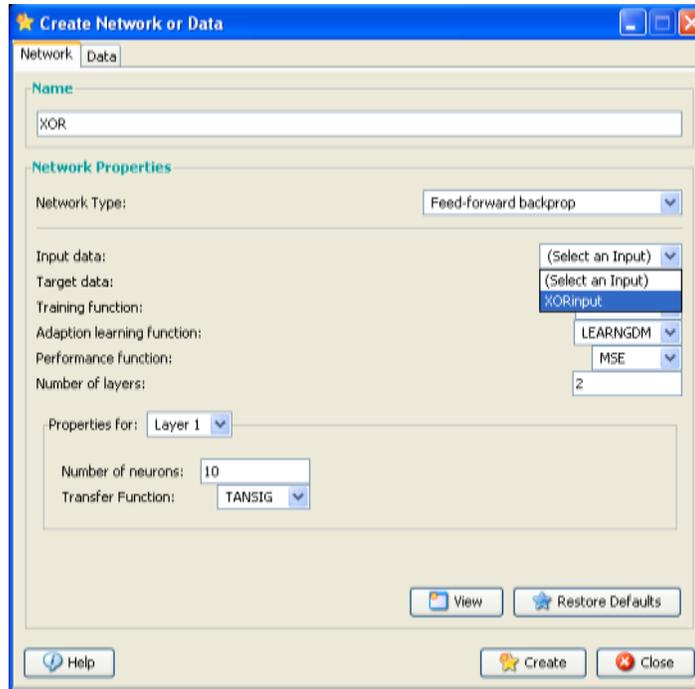
Click **Import**, select the data file you wish as the input to your ANN. Under Destination, make sure the **Input Data** button is selected. Again, select the data file you wish as the output to your ANN. Under Destination, make sure the **Target Data** button is selected.

3. Click **New** to create your ANN.

## 2011/2012 G51IAI Lab Session

The **Create Network or Data** window is where you can set up and configure your ANN, i.e. type of network, number of neurons at different layers, training function, etc.

Name your ANN as XOR, or whatever name you prefer. Select the **Input data** as XORinput and Target data as XORoutput. At the moment, leave all the other settings as default.



Click **Create** to create your ANN.

4. In the Network/Data Manager window, double click your ANN.

You can view the ANN you just created. Under **Train/Training Data** select the Inputs and Targets.

You can view the settings of the training parameters under **Training Parameters**. At the moment, leave them as default.

Click **Train Network**.

5. After the training is complete, click the **Simulate** tab to see the result of the output.

Under Simulate, choose the input XORinput, and click **Simulate Network**. Go back to the Network/Data Manager window, double click XOR\_output under **Output Data**. Check and compare the output with your expected output of XOR.

6. Click **Export** in the Network/Data Manager window to export your ANN.

Choose and output the variables you'd like to keep. After this you should be able to see them in your workspace.

7. Use/experiment your ANN.

Type **sim(XOR, [1; 0])** to simulate the result of inputting [1;0] to your ANN. Enter. Check and compare the result with the true table of XOR.

Try a different input by typing **sim(XOR, [?; ?])** and compare the result with the true table of XOR.

8. If you save your ANN as a .mat file, you'll be able to use it in your C++ or C#. This is out of the scope of this module but feel free to explore the possibilities!

### **Task: Build an ANN for the Bankloan Problem**

Download the bankloan spreadsheets from the module's web page. Build an ANN to learn to classify the customers.

Notes:

- You should follow the above same steps building the XOR network
- After the data files are imported into the Matlab workspace, the matrix should be transposed so that rows of the matrix correspond to the inputs to the ANN.

The command is:

```
Name_of_you_input_variable = Name_of_you_input_variable'
```

```
Name_of_you_output_variable = Name_of_you_output_variable'
```