

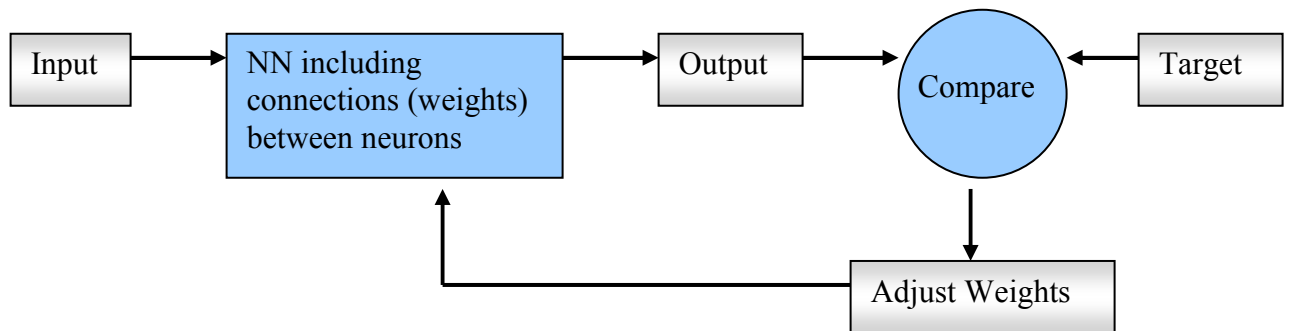
Demonstration of Artificial Neural Network in Matlab

Robyn Ball and Philippe Tissot
Division of Nearshore Research, Texas A&M University – Corpus Christi

Brief Introduction (Taken from Matlab R2006b Help, MathWorks, 2006)

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the network function is determined largely by the connections between elements. You can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements.

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown below. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are needed to train a network.



Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems.

Today neural networks can be trained to solve problems that are difficult for conventional computers or human beings. Throughout the toolbox emphasis is placed on neural network paradigms that build up to or are themselves used in engineering, financial, and other practical applications.

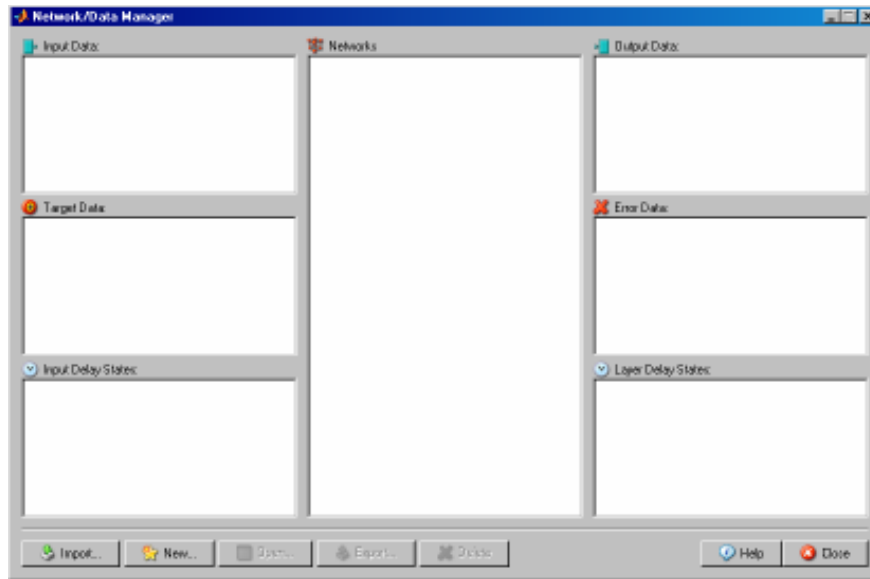
For more information on Neural Networks, go to the Help menu in Matlab, scroll down the Contents menu and click on Neural Network Toolbox.

Neural Net Graphical User Interface (GUI) in Matlab

For demonstration purposes we have ready-to-use data consisting of 3 years of salinity, air temperature, water level, and water temperature measurements at several stations. If you haven't already downloaded the Matlab data, please see that section.

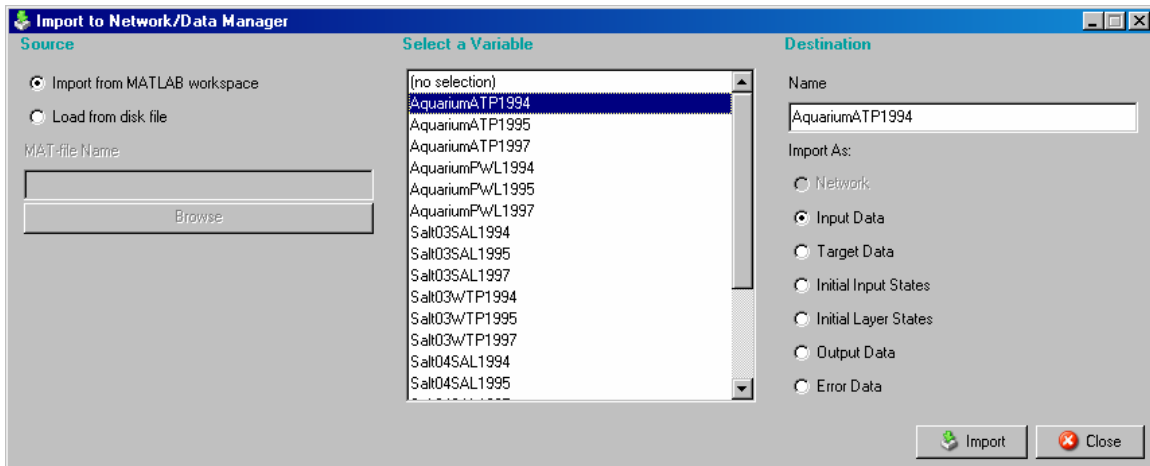
1. To begin using the NN GUI

>> nntool



2. Click on Import to import data.

Select the data you wish to use as inputs and make sure the **Input Data** button is selected.

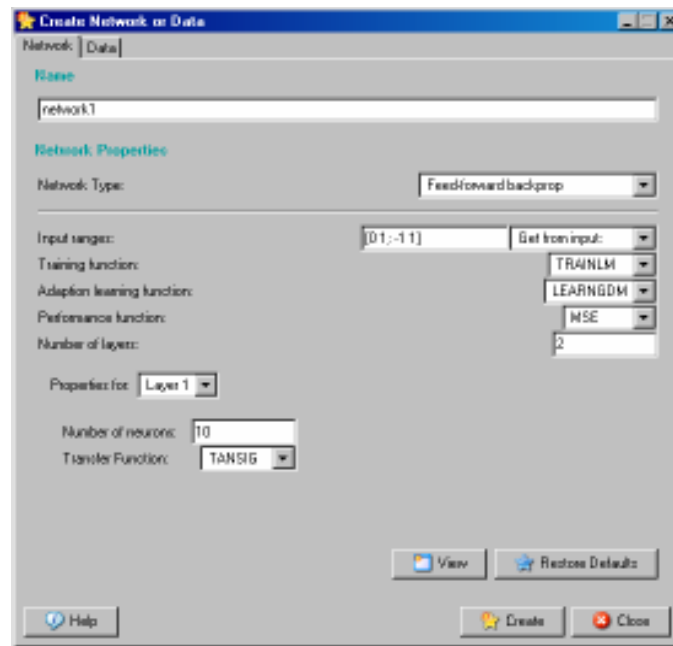


Select the data you wish to use as target data and click on the **Target Data** button.

To also include validation and testing data, import the desired data sets as Inputs or Targets.

Example: Suppose we want to construct a NNET to predict water temperature at the SALT04 Station using air temperature measurements at the Aquarium as inputs. We also want to include a validation and testing set. Thus, we import AquariumATP1994, AquariumATP1995, and AquariumATP1997 as inputs. We import Salt04WTP1994, Salt04WTP1995, and Salt04WTP1997. These will make up our training (1994), validation (1995), and testing (1997) sets.

3. Click on **New** to create your neural network



Here, you can adjust the type of network, training and transfer functions, number of neurons, etc.

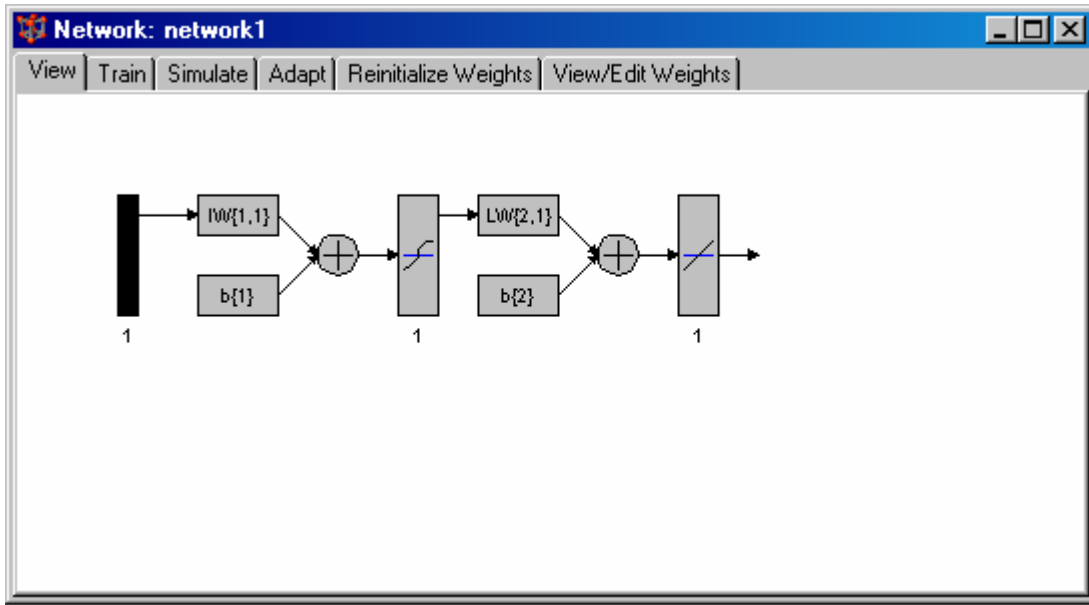
For our example we will construct a simple [1,1] NNET with a tansig transfer function as the hidden layer neuron and a purelin transfer function as the output later neuron. To do so, put 1 in the **Number of Neurons** box. Layer 1 is already set to tansig, which is what we want. Click the drop-down menu for **Properties for...** and set to Layer 2. Next, change the transfer function to purelin.

Now, set the Input ranges to our Input. Select **Get from Input:** and select AquariumATP1994.

Finally, click on **Create** and close the window.

4. Training, Validation, and Testing

Go back to the Network/Data Manager window and double click on **network1**



Here you can see the schematic of the neural net you just created. Click on the Train tab and (for the example) set **Inputs** to AquariumATP1994 and **Targets** to SALT04WTP1994.

Next, click on the **Validation and Testing** tab and check both boxes. We will include one year of data for validation and one year for testing. Using the example, set the

Validation Data: Inputs: AquariumATP1995
 Targets: Salt04WTP1995

Test Data: Inputs: AquariumATP1997
 Targets: Salt04WTP1997

The screenshot shows the 'Validation and Testing' tab in the software. It contains the following settings:

- Supply optional validation data? (Used for early stopping for many networks.)
- Supply optional testing data? (Useful for estimating a network's ability to generalize.)

Validation Data		Test Data	
Inputs	AquariumATP1995	Inputs	AquariumATP1997
Targets	Salt04WTP1995	Targets	Salt04WTP1997
Init Input Delay States	(zeros)	Init Input Delay States	(zeros)
Init Layer Delay States	(zeros)	Init Layer Delay States	(zeros)

At the bottom right, there is a button labeled "Train Network" with a flame icon.

You can adjust the training parameters as well as weights. For example, by clicking on **Training Parameters** tab, you can adjust the number of epochs and show. The default is show = 25. Try setting this value to 1.

Click **Train Network**

Note: To re-train on the same data and network, you need to reinitialize weights. Go to the **Reinitialize Weights** tab and click on **Initialize Weights**.

5. Exporting Errors/Output to Matlab Workspace

To export, return to the **Network/Data Manager** window and click on **Export**. Select the variables you wish to export and click **Export**. This is an easy way to plot and analyze the results.

6. Experiment

Try experimenting with different data sets and network architectures and see what results.

Suppose you want to include air temperature at the Aquarium and water temperature at Salt03 as inputs. To include more than one time series as an input, in the Matlab Command window, enter

```
>> Input(1,:) = AquariumATP1994;  
>> Input(2,:) = Salt03WTP1994;
```

Input can now be used in the creation of a neural network.

References

The MathWorks, Inc., 2006: MATLAB r2006b. The MathWorks, Inc.