**Electric Load Forecasting using different techniques**

***Artificial Neural Networks***

# Introduction:

The problem statement deals with hourly load forecasting. We have been given dataset from 2002 to 2005 (i.e. 4 years data) on hourly basis. I have used **2-layer feed forward artificial neural network** technique for forecasting the load for 2006 year on hourly basis.

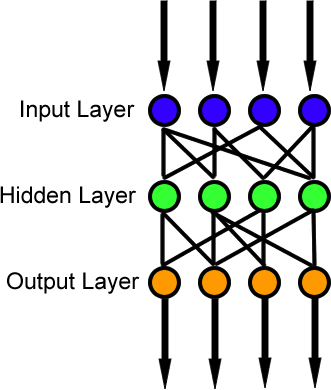
# Dataset:

The Dataset file is named **‘history.csv**’. The dataset provided has 4 variable columns namely ‘Date’ (from 1-Jan-2002 to 31st Dec 2005), ‘Hour’ ranging from 1 to 24, ‘Temperature’ & ‘Load’. The total **number of observations given is 35064**. Another excel file containing the 2006 data is provided named ’**fcst.csv**’ in which load column is kept blank to be forecasted.

# Methodology:

The ANN was built in **MATLAB** using ‘**nntool**’ feature. The various steps used in the process are elaborated further.

A feedforward neural network is an artificial neural network wherein connections between the nodes do not form a cycle. As such, it is different from its descendant: [recurrent neural networks](https://en.wikipedia.org/wiki/Recurrent_neural_networks). The feedforward neural network was the first and simplest type of artificial neural network devised. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network.



Source: Wikipedia

In our case the target variable is Load and the input/feature variable are Hour of day, Day of week, Month, Temperature.

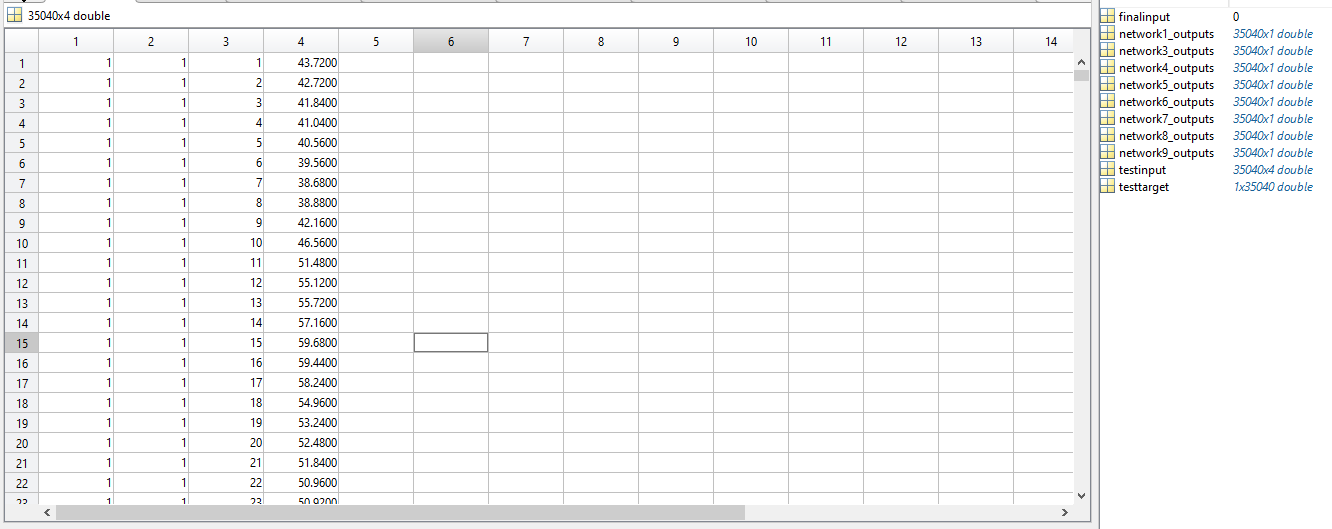
# Data-Preprocessing:

The first and foremost step taken is to delete the 24 observations for the 29th February timestamp in the leap year 2004. This is done as these values for the date 29th Feb are seen only once throughout the four years.

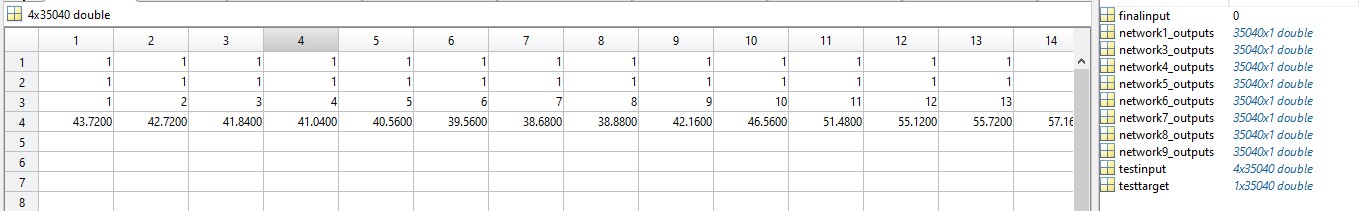
We have converted the date column into two different feature columns named day of week and month.

# MATLAB work and step-by-step process.

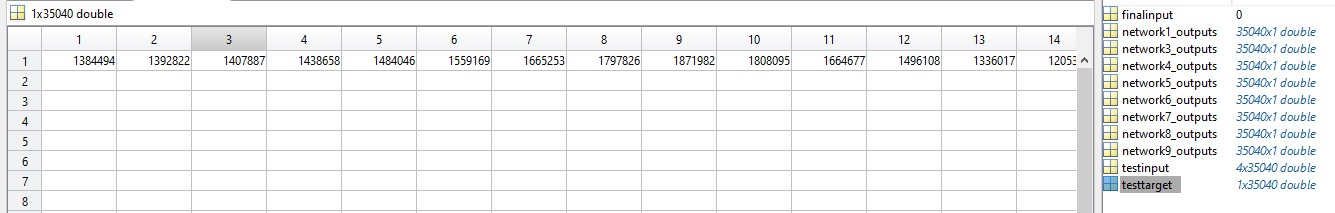
Initially we create 3 files in the workspace window named ‘testinput’, ‘testtarget’ and ‘finalinput’ to store the data from our History and Fcst files for working.



We do a transpose function on the matrix we created as MATLAB cannot process the data as it is.

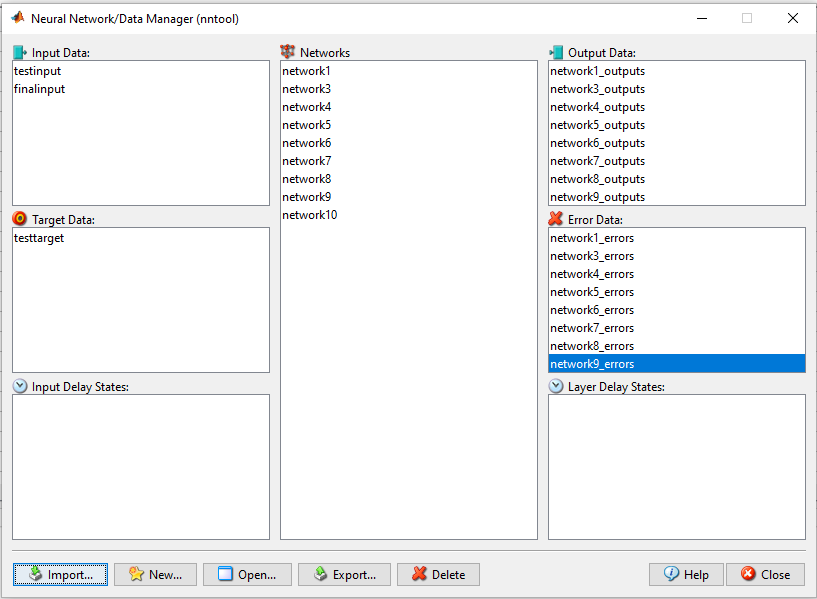


testtarget data input:-

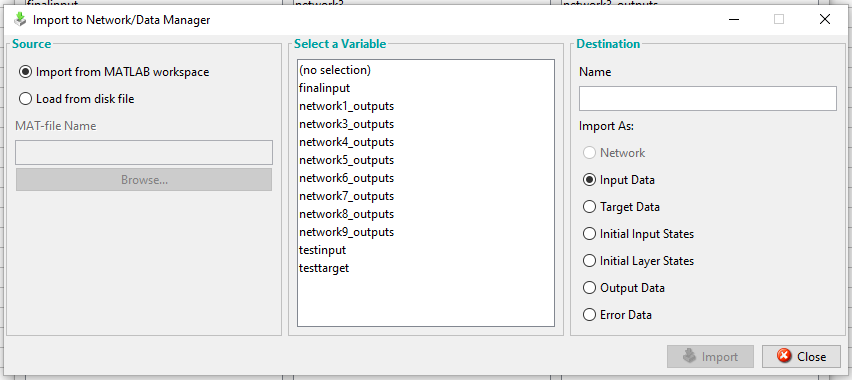


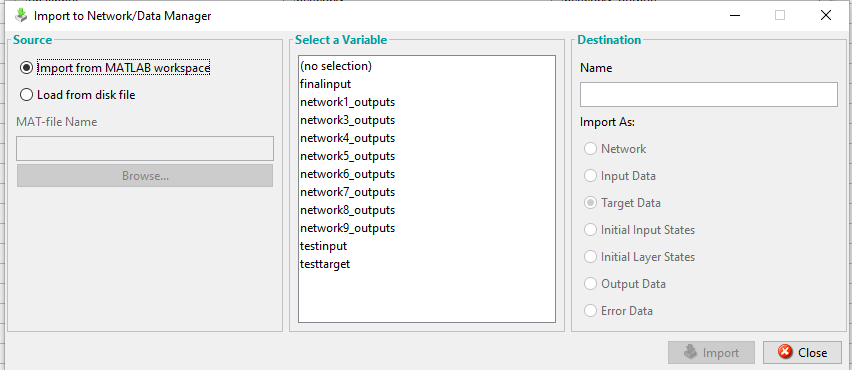
After that we call ‘***nntool’*** function from the command window of MATLAB.

nntool:- Open Network/Data Manager. Neural network fitting tool. nntool leads you through solving a data fitting problem, solving it with a two-layer feed-forward network trained with Levenberg-Marquardt.

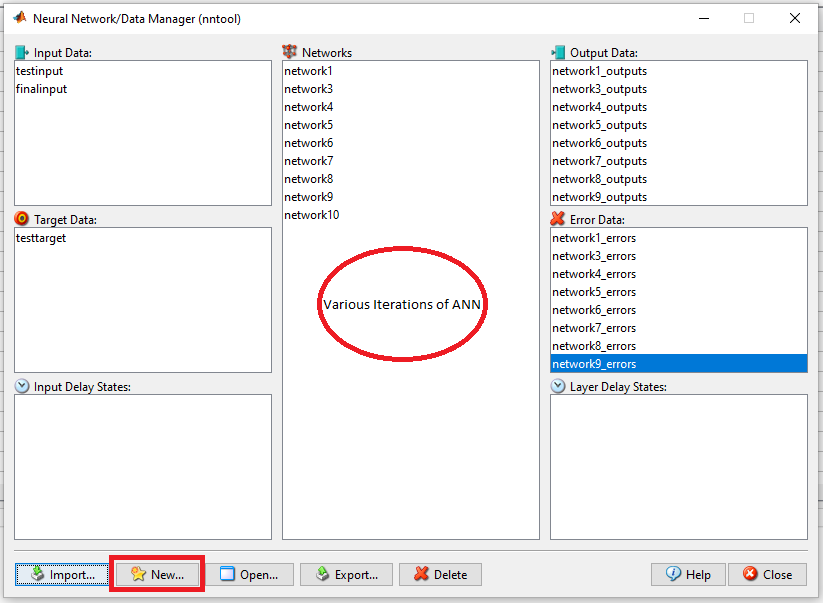


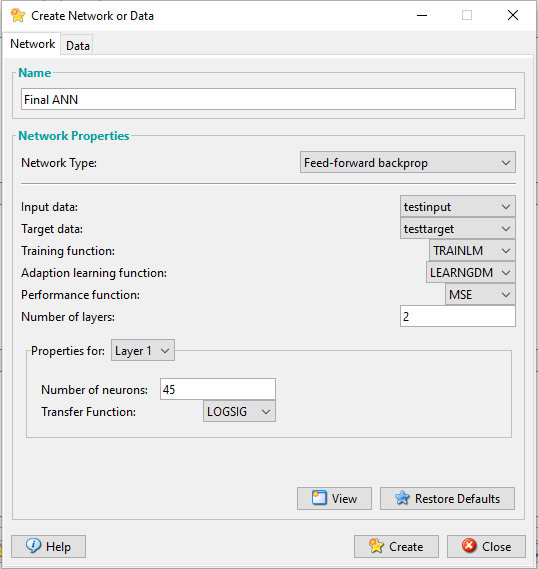
We then input data using import.





After that next step is to build the Neural Network. Click new to open the ANN window.





Network Name: FinalANN Network Type:

The Backpropagation algorithm is a supervised learning method for multilayer feed-forward networks from the field of Artificial Neural Networks. Feed-forward neural networks are inspired by the information processing of one or more neural cells, called a neuron. The principle of the backpropagation approach is to model a given function by modifying internal weightings of input signals to produce an expected output signal. The system is trained using a supervised

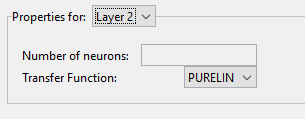
learning method, where the error between the system’s output and a known expected output is presented to the system and used to modify its internal state.

Training Function: Levenberg-Marquardt Back-Propagation Algorithm.

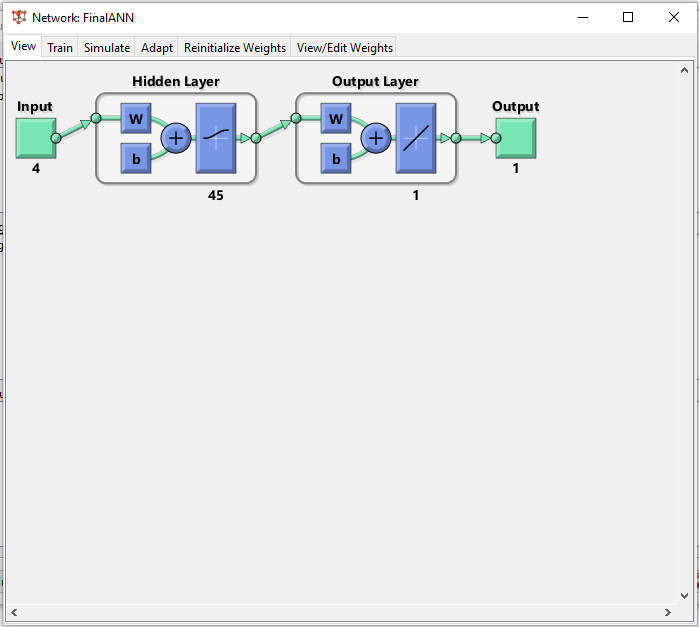
Learning Function: ‘learngdm’ is the gradient descent with momentum weight and bias learning function. Performance Function: Mean Square Error.

Number of Layers: 2

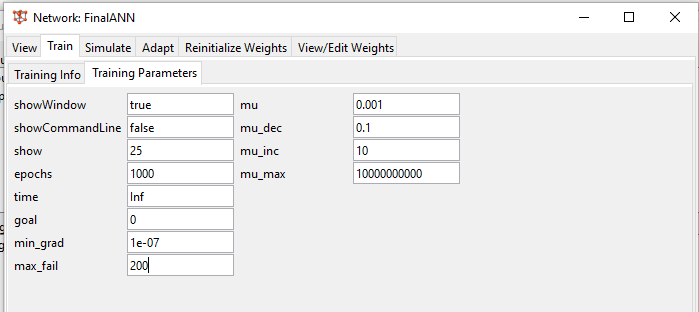
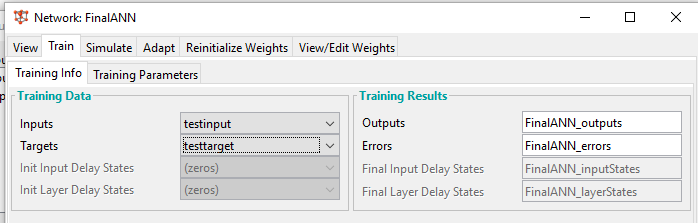
Layer 1: Number of neurons = 45, Activation function = ‘logsig’. Layer 2: Number of neurons = 1, Activation function = ‘purelin’

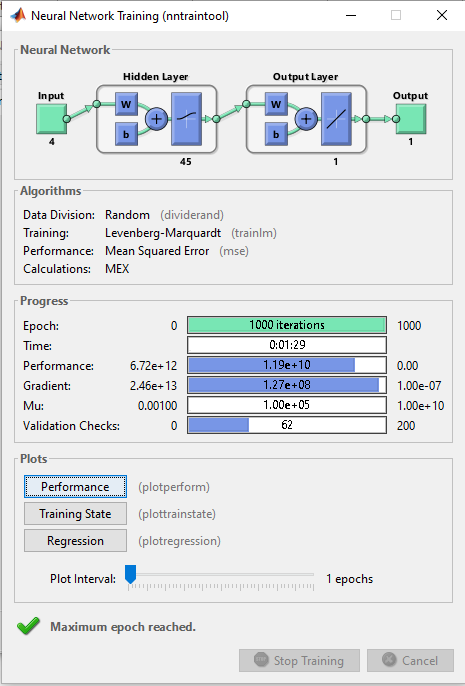


Then we train and test the model.

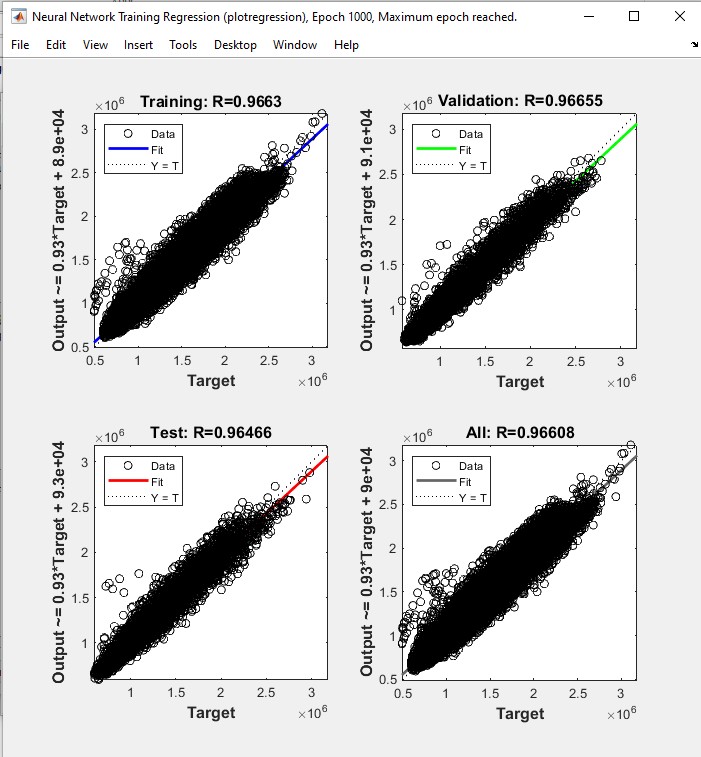


Click on train and select the test input and target values. Change the validation check number to 200 and epoch number to 1000 (iterations). Rest all is default.





The regression plot shows the R value around 0.96… for training, validation, test and all. This shows our model is satisfactory and acceptable.



# Final Step: Forecasting

We now do the forecasting for the year 2006 (8760 Values on hourly basis) using the simulate function in the network window from where we did the train command. We select the data from year 2006 stored inside the ‘finalinput’ file we earlier created and start the forecasting. The output file is exported to the workspace and finally to the Fcst.csv file. Hence, we have used the ANN model for forecasting year ahead hourly based load. Our model is working.

# Conclusion:

We have tried different iterations (Neural Nets) in the form of network 1 to 10 having different number of neurons to get the lowest ***MAPE*** for our model which comes to be ***10.62%.***