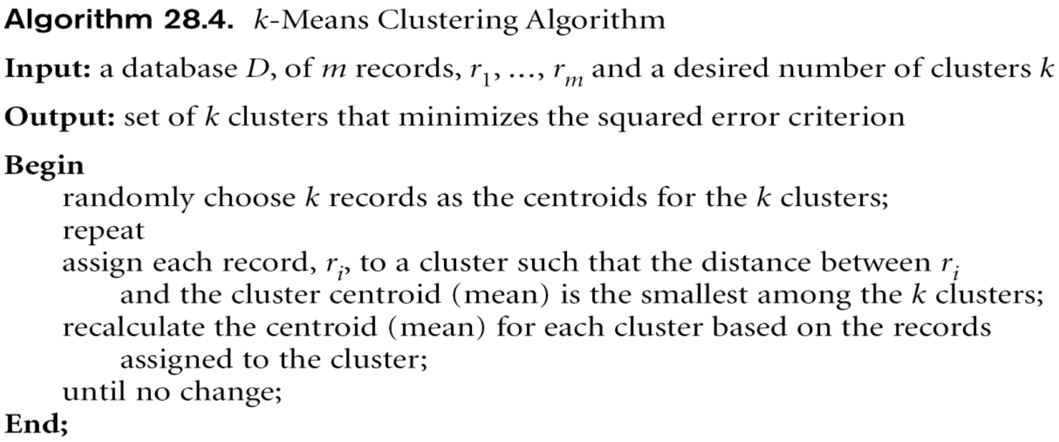
Assignment 3 (10 mark)
PART 1 - K-Means - 3 marks

Data Mining in databases means analyzing records and related attributes for finding patterns, trends and relationships to form model to be used in business decision making, prediction, and etc. One of the data mining method is clustering and a common method of clustering is k-Means. General k-Means algorithm when applies on database data is shown below which is from Chapter 28th of Database book by Elmasri:



Algorithm 28.4. *k*-Means Clustering Algorithm

Input: a database D, of m records, $r_1, \ldots, r_m$ and a desired number of clusters k

Output: set of k clusters that minimizes the squared error criterion

```
Begin
    randomly choose $k$ records as the centroids for the $k$ clusters;
    repeat
    assign each record, $r_i$, to a cluster such that the distance between $r_i$
        and the cluster centroid (mean) is the smallest among the $k$ clusters;
    recalculate the centroid (mean) for each cluster based on the records
        assigned to the cluster;
    until no change;
End;
```

a) First use the above algorithm and by using a common similarity metric distance between a record, and also using a value of 3 for K, manually cluster the data of following table. You can assume that the records with RIDs 1, 3, and 5 are used for the initial cluster centroids (means). Try to follow the algorithm and calculate the centroids in a way that clusters to be optimum. Print the cluster members and centroid information.

RID	Dimension 1	Dimension 2
1	8	4
2	5	4
3	2	4
4	2	6
5	2	8
6	8	6

b) Now you need to implement the manual k-means clustering you did above, on a software to verify and visualize your results. You can use any software (R or Python) for generating a graphical view of the clusters or a free trail of SPSS platform at http://spss.en.softonic.com (or any platform of your choice) and answer the following questions:

1. Use non hierarchical k-means (k=3) and show clusters' memberships of the above dataset.
2- Use hierarchical clustering with dendrogram which is a cluster tree for visualization of hierarchical clustering. Print dendogram and explain how many cluster options we can have for a given cutting points.
*There will be bonus mark if you do the above clustering with large dummy data of one of the tables (with several attributes) of the database that you created in Assignment1.

Part 2 – Supervised and Unsupervised ML models- 6 marks

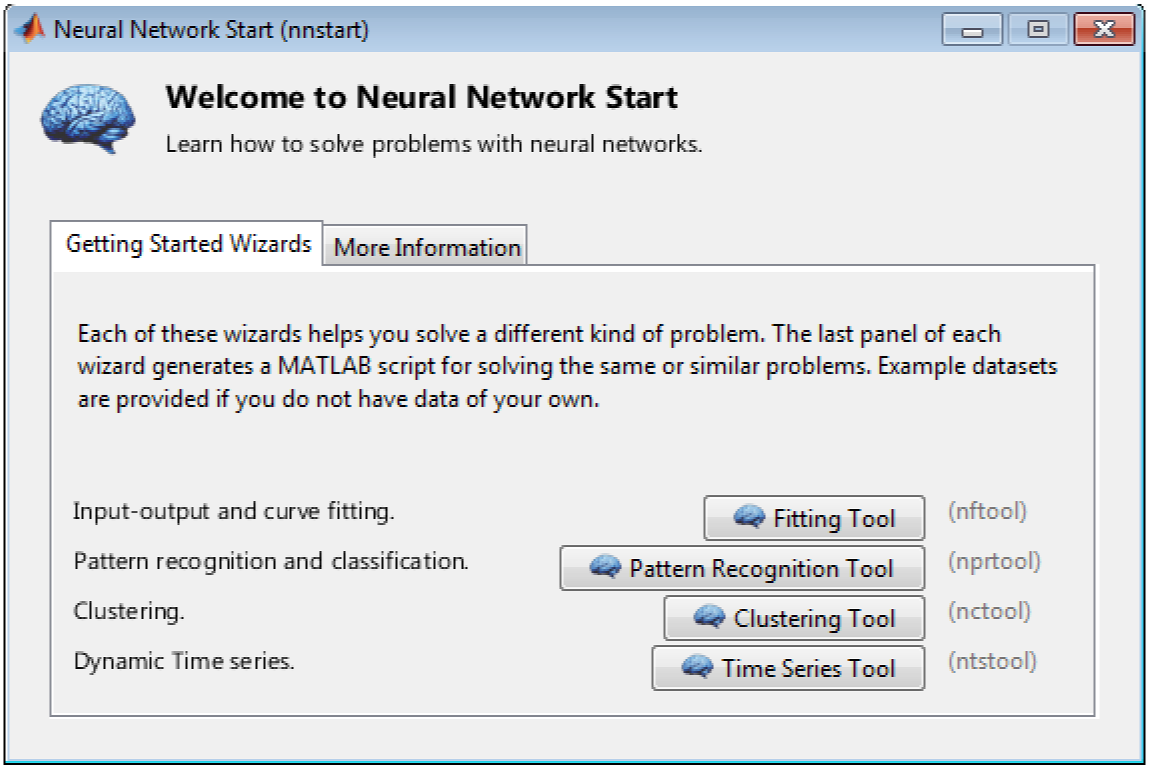
Section A- Supervised Machine Learning: In this section, you will use the heart disease data set to train an artificial neural network as explained in next pages compare it with logistic regression analysis (or any other nonlinear classifier of your choice) and then answer which method is better. First, use two models as the estimator (with the numerical result). Here you need to compare both methods by calculating Errors such as Mean Absolute Error (MAE) and other performance metrics to find which method can do prediction more accurately. Then use two models as the classifier and produce categorical result and confusion matrix. If it is needed in heart disease dataset change the final result to a class with two labels of severe or non-severe (for example, by considering a threshold).Make sure you separate training set and testing data and there is no overfitting. Explain what is training errors and tests errors in your used methods. Try to use R or Python (the one that you used in Regression analysis) for implementing ANN. However because it is your first assignment of ANN you can use a simple interactive tool in Matlab to build ANN shown as appendix on the next page.. Please note the preference is to implement ANN with R or Python. The following figures in the appendix show different steps of building the model and training of ANN and how you can access the data set..

Section B- Unsupervised Machine Learning: In this section use an unsupervised learning method k-means clustering on hear disease data set. Create and visualize the k-means clusters (with k=5) for the given heart disease dataset. For visualization, draw the scatter plot using the age and cholesterol features on each group of clusters. Apply k-means clusters on the heart disease dataset with varying numbers of clusters from 1 to 10 and compute their corresponding Sum of squared Error (SSE) value. Plot the graph and estimate the right “k” value . Use SSE vs the number of clusters to estimate “k” value. Finally, discuss if applying an unsupervise machine learning method such as clustering on the heart disease dat set can provide any additional information about this data set or not.

Appendix related to Part2- Section A: An example shows how to use ANN in Matlab with Heart Disease Data Set.

ANN model can be first trained using training data set according to following steps:

1. In Matlab command prompt type **nnstart** and press enter



Neural Network Start (nnstart)

Welcome to Neural Network Start

Learn how to solve problems with neural networks.

Each of these wizards helps you solve a different kind of problem. The last panel of each wizard generates a MATLAB script for solving the same or similar problems. Example datasets are provided if you do not have data of your own.

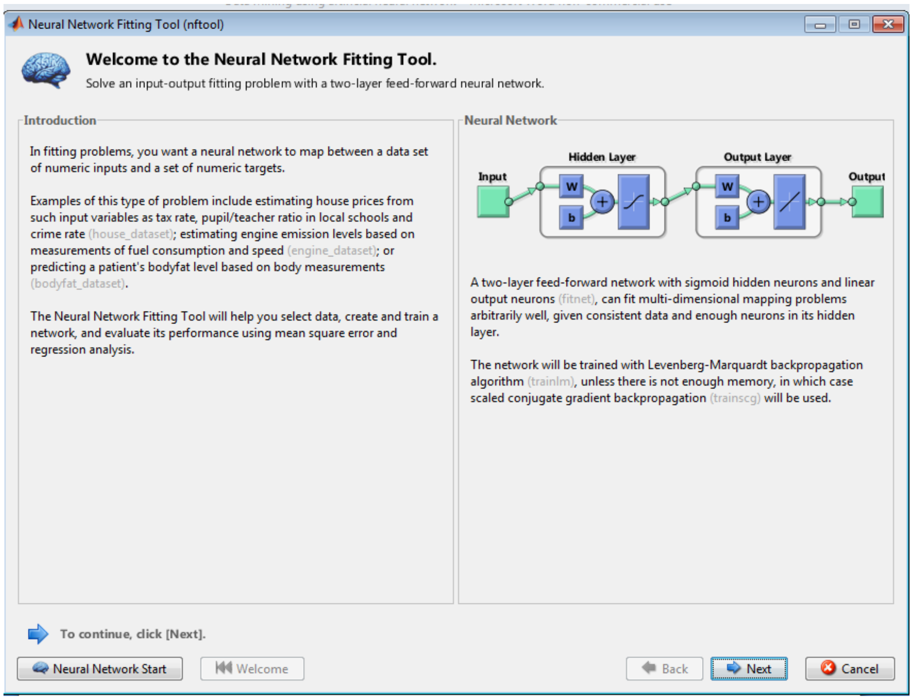
Input-output and curve fitting. Fitting Tool (nftool)

Pattern recognition and classification. Pattern Recognition Tool (nprtool)

Clustering. Clustering Tool (nctool)

Dynamic Time series. Time Series Tool (ntstool)

2. Select any tool which suits your ANN problem. In this example fitting tool was selected.



Welcome to the Neural Network Fitting Tool.

Solve an input-output fitting problem with a two-layer feed-forward neural network.

Introduction

In fitting problems, you want a neural network to map between a data set of numeric inputs and a set of numeric targets.

Examples of this type of problem include estimating house prices from such input variables as tax rate, pupil/teacher ratio in local schools and crime rate (house_dataset); estimating engine emission levels based on measurements of fuel consumption and speed (engine_dataset); or predicting a patient's bodyfat level based on body measurements (bodyfat_dataset).

The Neural Network Fitting Tool will help you select data, create and train a network, and evaluate its performance using mean square error and regression analysis.

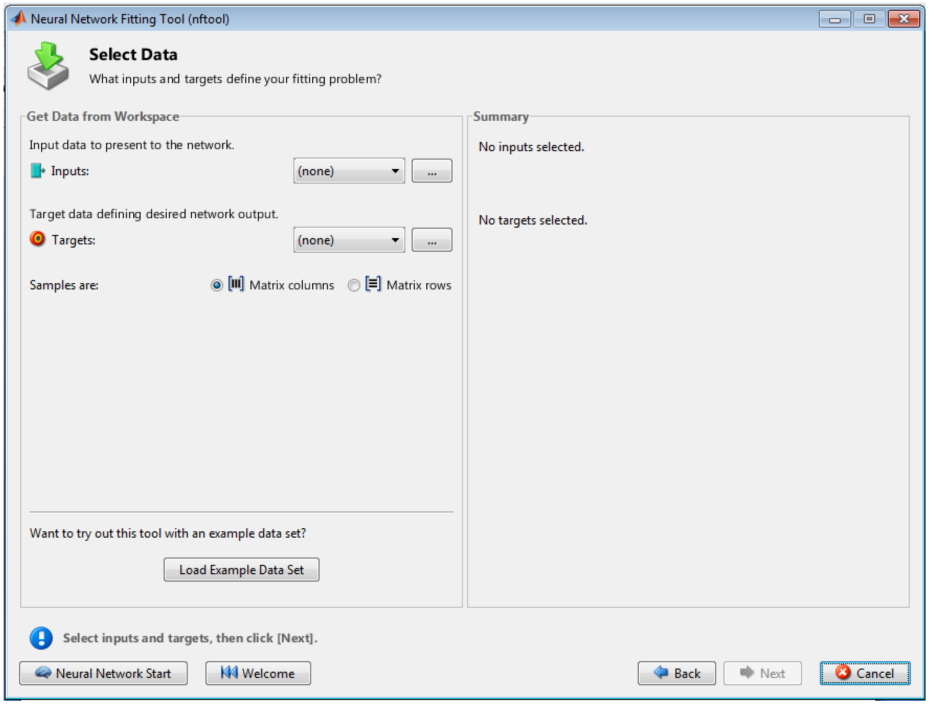
Neural Network

A two-layer feed-forward network with sigmoid hidden neurons and linear output neurons (fitnet), can fit multi-dimensional mapping problems arbitrarily well, given consistent data and enough neurons in its hidden layer.

The network will be trained with Levenberg-Marquardt backpropagation algorithm (trainlm), unless there is not enough memory, in which case scaled conjugate gradient backpropagation (trainscg) will be used.

To continue, click [Next].

3. Select data sets, example is loaded by using Load Example Data Set button.



Select Data

What inputs and targets define your fitting problem?

Get Data from Workspace

Input data to present to the network.

Inputs: (none)

Target data defining desired network output.

Targets: (none)

Samples are: ● Matrix columns ○ Matrix rows

Want to try out this tool with an example data set?

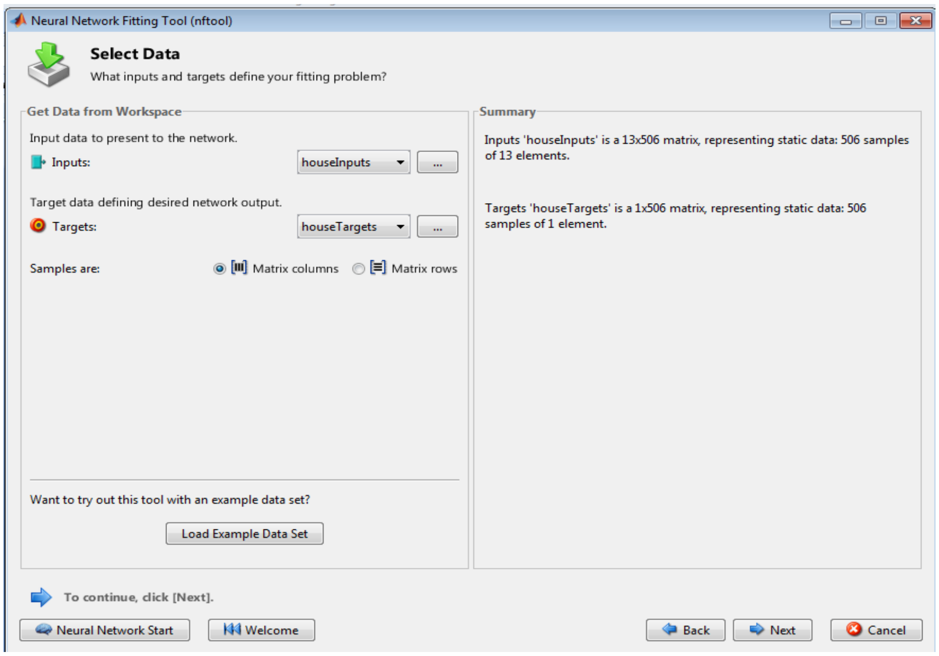
Summary

No inputs selected.

No targets selected.

Select inputs and targets, then click [Next].

4. Select Housing Pricing data set. You will notice that Input and target data field are filled with the selected data.



Select Data

What inputs and targets define your fitting problem?

Get Data from Workspace

Input data to present to the network.

Inputs: houseInputs

Target data defining desired network output.

Targets: houseTargets

Samples are: ● Matrix columns ○ Matrix rows

Want to try out this tool with an example data set?

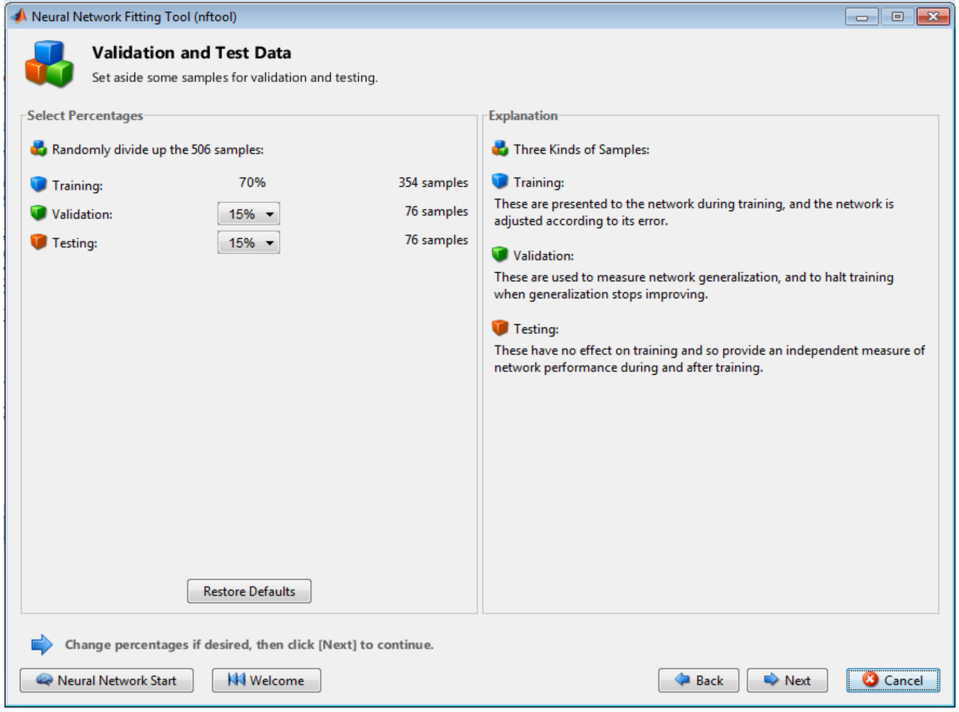
Summary

Inputs 'houseInputs' is a 13x506 matrix, representing static data: 506 samples of 13 elements.

Targets 'houseTargets' is a 1x506 matrix, representing static data: 506 samples of 1 element.

To continue, click [Next].

5. Data set is divided as 70-15-15 ratio by default but you can change it as you like.



Validation and Test Data

Set aside some samples for validation and testing.

Select Percentages

Randomly divide up the 506 samples:

Training:	70%	354 samples
Validation:	15%	76 samples
Testing:	15%	76 samples

Explanation

Three Kinds of Samples:

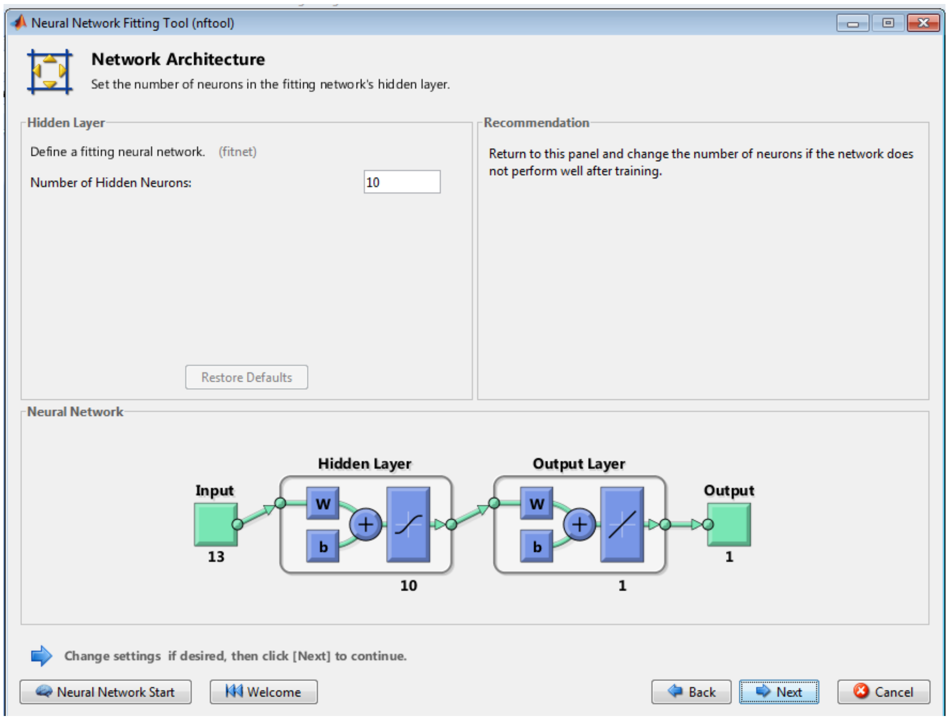
Training:
These are presented to the network during training, and the network is adjusted according to its error.

Validation:
These are used to measure network generalization, and to halt training when generalization stops improving.

Testing:
These have no effect on training and so provide an independent measure of network performance during and after training.

Change percentages if desired, then click [Next] to continue.

6. This example has 13 inputs and 1 output, therefore ANN model has 13 inputs with 1 output, with 1 hidden layer. Number of neuron in hidden layer can be increase or decrease.



Network Architecture

Set the number of neurons in the fitting network's hidden layer.

Hidden Layer

Define a fitting neural network. (fitnet)

Number of Hidden Neurons: 10

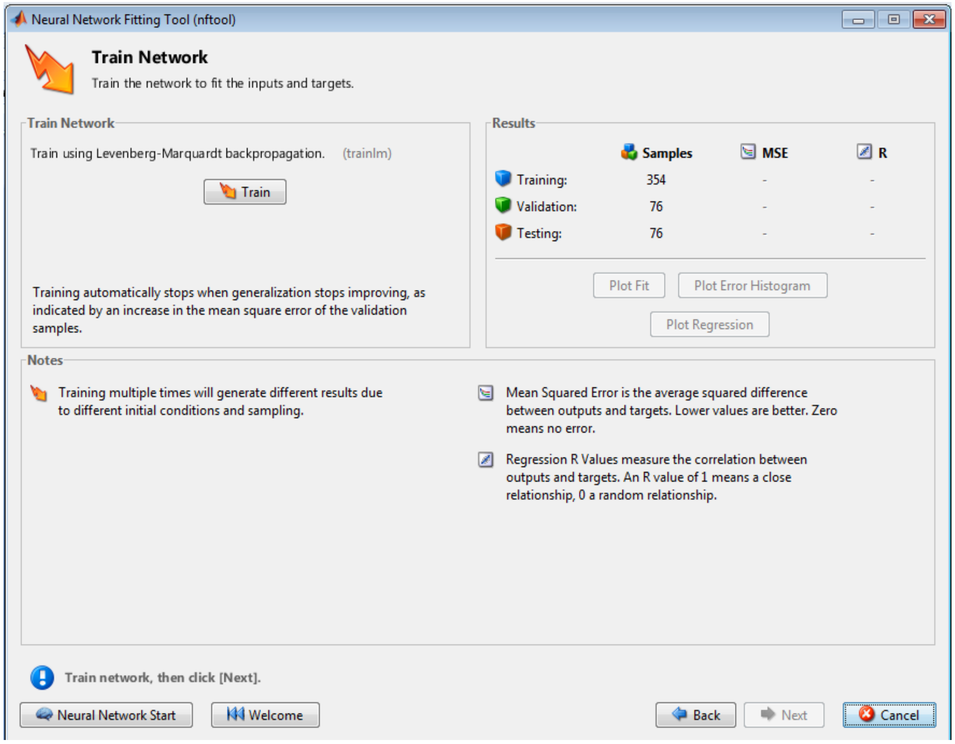
Recommendation

Return to this panel and change the number of neurons if the network does not perform well after training.

Neural Network

Change settings if desired, then click [Next] to continue.

7. To train ANN model click Train button.



Train Network

Train the network to fit the inputs and targets.

Train Network

Train using Levenberg-Marquardt backpropagation. (trainlm)

Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Results

	Samples	MSE	R
Training:	354	-	-
Validation:	76	-	-
Testing:	76	-	-

Notes

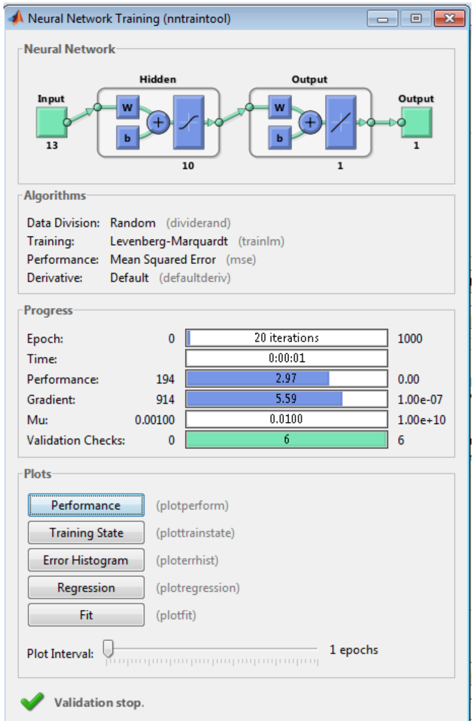
Training multiple times will generate different results due to different initial conditions and sampling.

Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.

Train network, then click [Next].

8. Once you start training the ANN network, the following screen shows the progress.



Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Progress

Epoch:	0	20 iterations	1000
Time:		0:00:01	
Performance:	194	2.97	0.00
Gradient:	914	5.59	1.00e-07
Mu:	0.00100	0.0100	1.00e+10
Validation Checks:	0	6	6

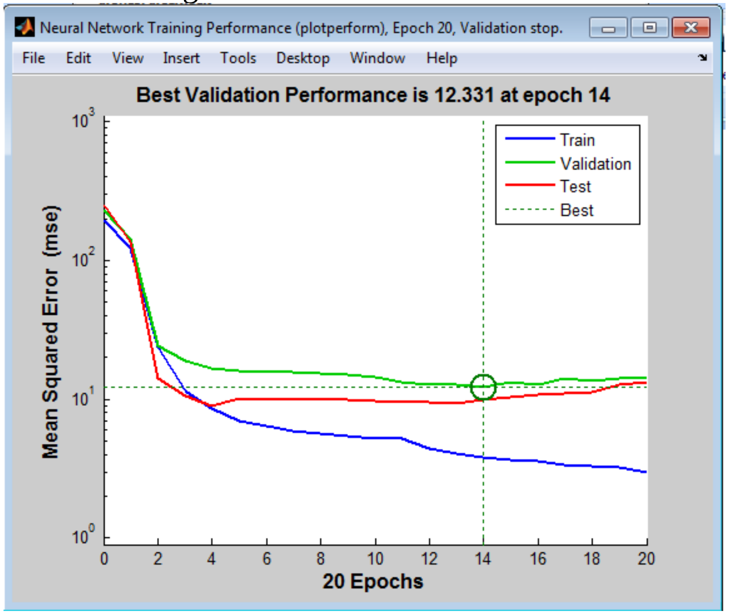
Plots

Performance (plotperform)
Training State (plottrainstate)
Error Histogram (ploterrhist)
Regression (plotregression)
Fit (plotfit)

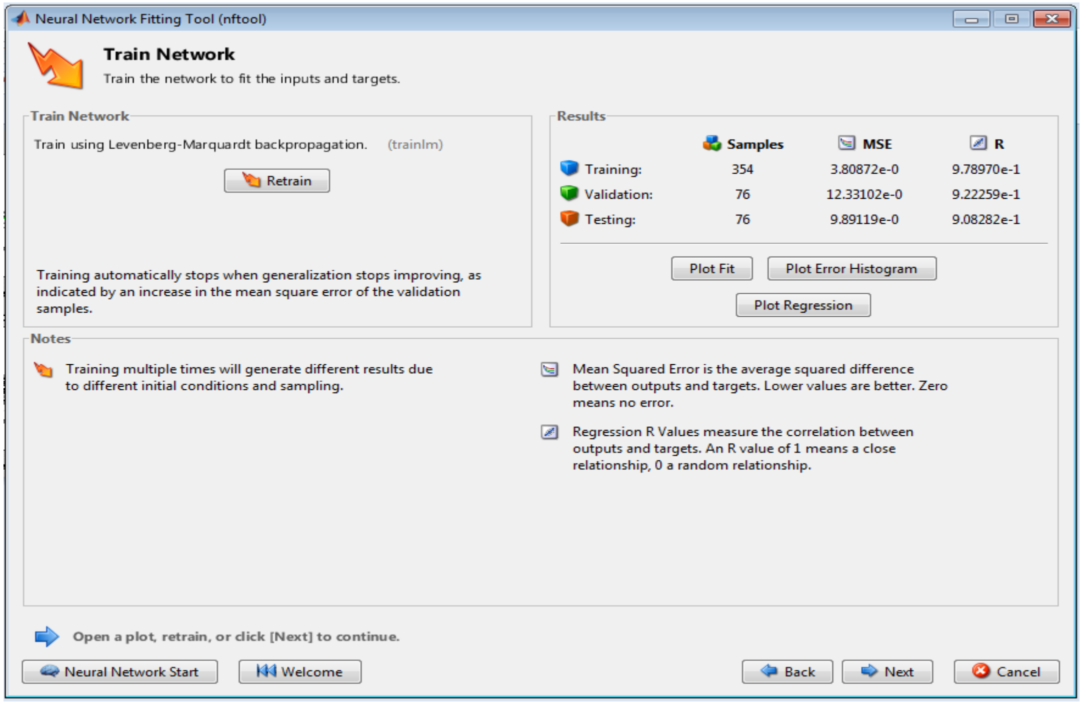
Plot Interval: 1 epochs

✓ Validation stop.

9. When training is complete. The performance of trained ANN model can be seen by clicking Performance Button. Here Best Performance is achieved at 12.331 at epoch 14.



10. Once training is done and if you like the performance then you can test you newly build ANN model with the data which was not included in your training data set. You can also retrain the model. By the way you need to retrain your model multiple times to get best result.



Neural Network Fitting Tool (nftool)

Train Network

Train the network to fit the inputs and targets.

Train Network

Train using Levenberg-Marquardt backpropagation. (trainlm)

Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Results

	Samples	MSE	R
Training:	354	3.80872e-0	9.78970e-1
Validation:	76	12.33102e-0	9.22259e-1
Testing:	76	9.89119e-0	9.08282e-1

Notes

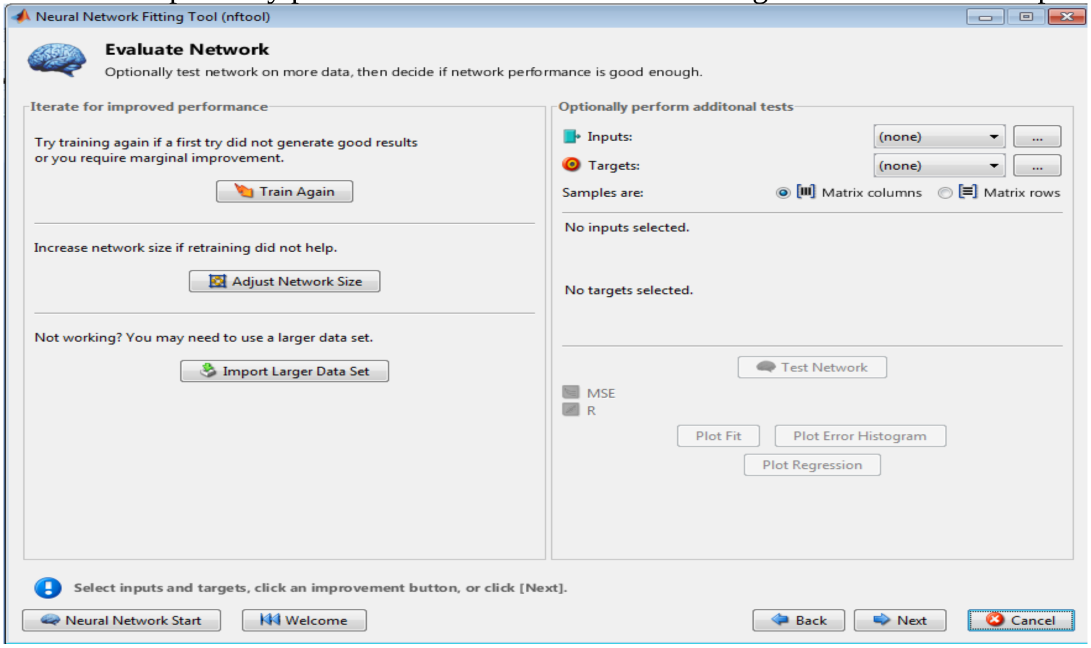
Training multiple times will generate different results due to different initial conditions and sampling.

Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.

Open a plot, retrain, or click [Next] to continue.

11. Use optionally perform additional tests button on right side to test with separate data set.



Neural Network Fitting Tool (nftool)

Evaluate Network

Optionally test network on more data, then decide if network performance is good enough.

Iterate for improved performance

Try training again if a first try did not generate good results or you require marginal improvement.

Increase network size if retraining did not help.

Not working? You may need to use a larger data set.

Optionally perform additonal tests

Inputs: (none)

Targets: (none)

Samples are: ● Matrix columns ○ Matrix rows

No inputs selected.

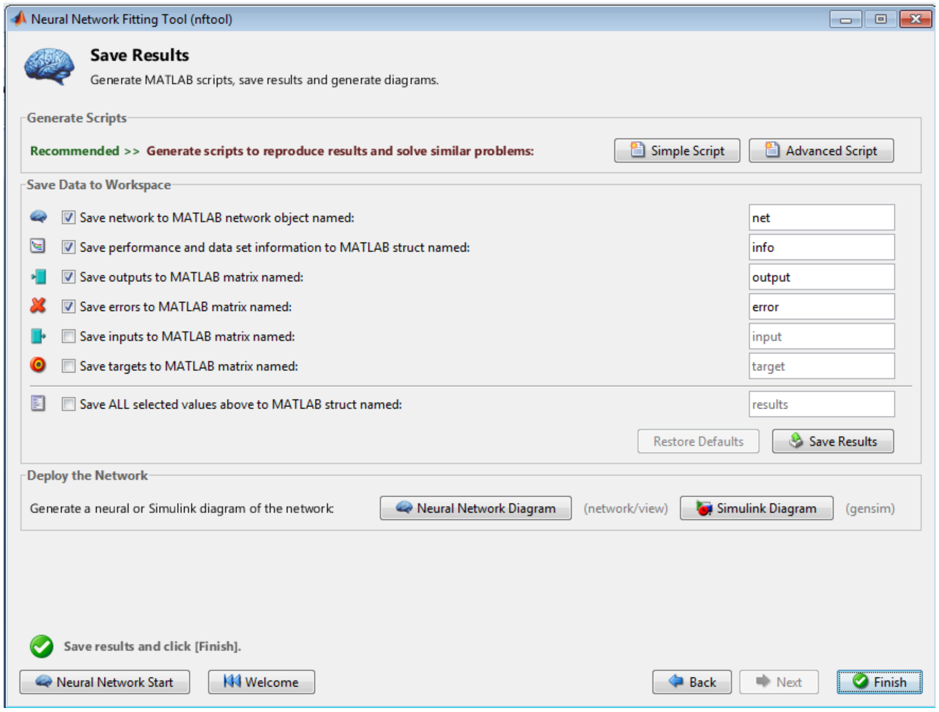
No targets selected.

MSE

R

Select inputs and targets, click an improvement button, or click [Next].

12. All the results and the ANN model can be save for further use.



Neural Network Fitting Tool (nftool)

Save Results

Generate MATLAB scripts, save results and generate diagrams.

Generate Scripts

Recommended >> Generate scripts to reproduce results and solve similar problems:

Save Data to Workspace

☑ Save network to MATLAB network object named: net

☑ Save performance and data set information to MATLAB struct named: info

☑ Save outputs to MATLAB matrix named: output

☑ Save errors to MATLAB matrix named: error

□ Save inputs to MATLAB matrix named: input

□ Save targets to MATLAB matrix named: target

□ Save ALL selected values above to MATLAB struct named: results

Deploy the Network

Generate a neural or Simulink diagram of the network: (network/view) (gensim)

✓ Save results and click [Finish].

Show your ANN design and write a brief analysis of your ANN result for the above example

Heart Disease Dataset: To use ANN in Matlab or R or Python with the heart disease dataset of patients go to the following link: http://archive.ics.uci.edu/ml/datasets/Heart+Disease

After going to this link you will find two folders: One: Data Folder and two: Dataset description. Data folder that has the dataset. It is better to use processed cleveland data. In the dataset description folder, you will find the description about the columns' names referring to the14 column of the dataset as the following: The last one attribute (number 14) is the result. You want to use neural network that you created before for training and predicting the result attribute. After designing and training the neural net with an appropriate amount of data set (e.g. 70%), test it with the remaining data and report the accuracy of the prediction for both data mining techniques by Mean absolute error (MAE) and Root mean squared error (RMSE) and any other performance metrics that you think is useful.