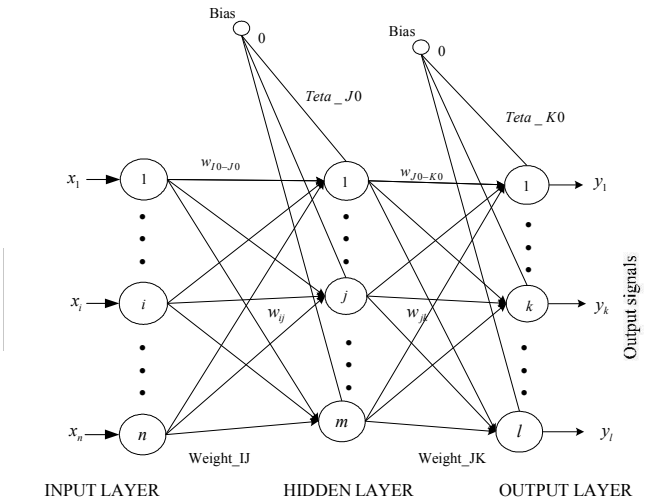
SPICE-NEURO NEURAL NETWORK PROGRAM

USERS' GUIDE

Cao Thang 2003 – 2007



1. INTRODUCTION

This is a user's guide for the Spice-Neuro Neural Network Program and does not intend to introduce about neural network theories.

The purpose of this program is to get you started quickly with Neural Network without having to go through lengthy theory of the Neural Network background. Once you understand these programs you will be able to consult the Neural Network materials on a need basis.

Spice-Neuro is a 3 layer Neural Network Program with multi-inputs and outputs. Spice-Neuro was written with the arm to introduce NN to students studying NN and modeling various data by NN. Currently Spice-Neuro has been using by many students around the world. Spice-Neuro has interfaces in Vietnamese, English and Japanese.

Spice-Neuro was written by CAO THANG when he did his researches in the Soft Intelligence Laboratory, Ritsumeikan University, Japan, 2003-2007.

Spice-Neuro and Spice-SOM can be downloaded at download.cnet.com

Having read this material, you may want to read neural_network_practical_use_en.pdf that illustrates classifications for face, pedestrians and car, stock price prediction, exchange forecast and other example

If you have questions or requirements about Spice-Neuro, please contact the author at http://spiceneuro.wordpress.com or spiceneuro AT gmail DOT com, Thank you.

2. INSTALL THE SPICE-NEURO

Download setup file of the Spice-Neuro and run setup.exe, setup welcome window appears on the screen:

Fig. 1. Installing

Select Next, and then select a folder into that you want to install Spice-Neuro, then select Next and Next. Spice-Neuro will be installed into the selected folder.

Fig. 2. Select Folder

Note:

In the old Windows version, if the Spice-Neuro does not run after installing, you may need to install *Microsoft .NET Framework Redistributable Package 3.5.21022* before installing Spice-Neuro.

If your data is in MDB format, you may need to install *Microsoft Data Access Components.*

3. USING SPICE-NEURO

Run Spice-Neuro by clicking onto Spice-Neuro icon on your desktop or selecting "Start → Programs → Cao Thang's Spice-Neuro → Spice-Neuro ".

The program runs with an English interface, you can select Vietnamese of Japanese by selecting "Options → Languages".

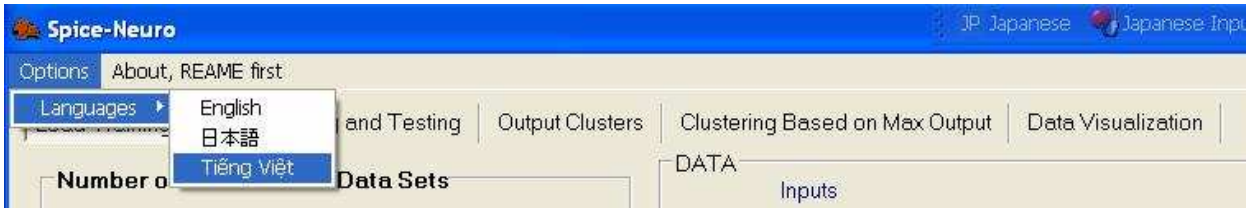


Fig. 3. Select Language

Menu "About, README first" is a briefly introduction about Spice-Neuro and users' agreement. You should read it carefully before using Spice-Neuro.

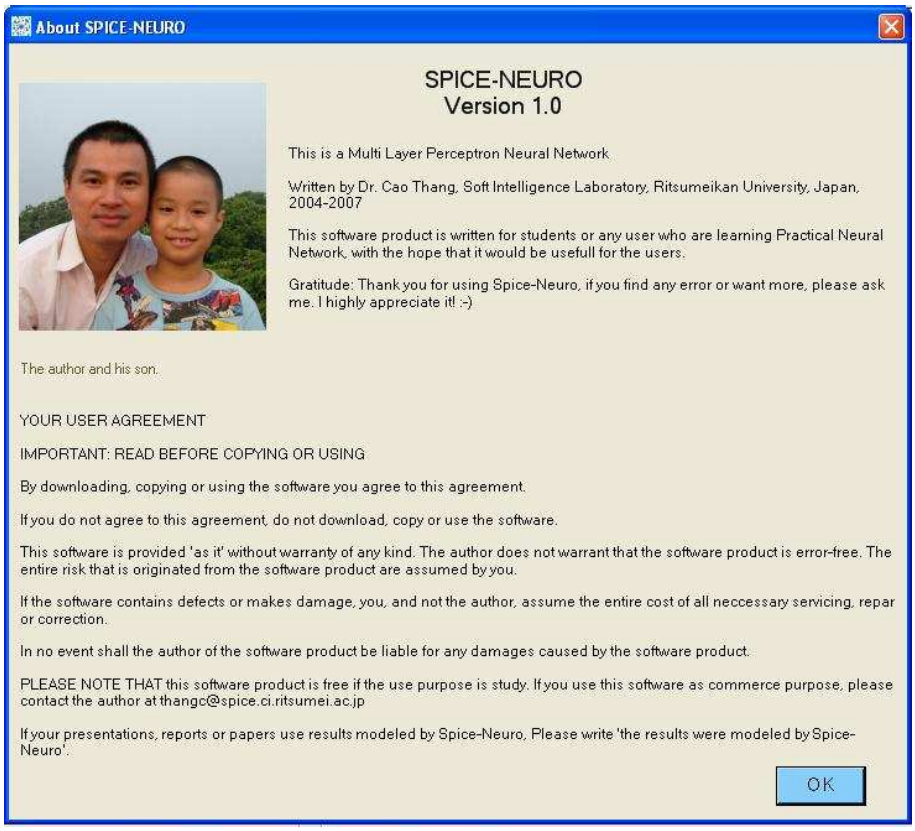


Fig. 4. About program. The image was taken on 2007

3.1. Data Preparation

For using your data by Spice-Neuro, you should prepare your data by the following format.

3.1.1. Text format

Data in text format should be prepared in rows and columns. The first column is ID, and then Inputs and Outputs. Values are separated by comma (Comma Separated Value File Format) with CSV file type, or Tab (Tab Separated Value File Format) with TXT file type. You may use MS Excel to edit your data, and then save it in text or csv format. For example data with 2 inputs and 3 outputs is organized as shown in Table 1.

Table 1. Data with 2 inputs and 3 outputs in text format

ID	X0	X1	Y0	Y1	Y2	LABEL
0	0	0	0	0	0	Data 1
1	0	1	1	0	1	Data 2
2						
3					a	

ID: Order of DataSet

X: Input Data

Y: Output Data

LABEL: Label of each Dataset

Note:

Data should be numeral, except label and input and output symbols.

Spice-Neuro cannot read your data if there is blank or null value. In the testing data, if there is no output, please set is as 0, 1 or other number.

There are some good examples in "\Data" folder of the Spice-Neuro:

"Boolean functions.csv" is an example with 4 datasets, 2 inputs and 3 outputs. Inputs are binary values 0 and 1, outputs are values of XOR (Y0), AND (Y1) OR(Y2) functions.

"Herbal data.csv" is an example with 640 datasets, 16 inputs and 33 outputs. Inputs are symptoms' severities; outputs are co-efficiencies of treatment herbs normalized in [0, 1].

"sincos.csv" is an example with 100 datasets, 1 input and 2 outputs. Input is argument with values in [0, 2π], and outputs are values of Sin and Cos of the input argument.

- "iris_for_mlp_4inputs_1output.csv", "iris_for_mlp_4inputs_3outputs.csv" is data of 3 flower species (Iris setosa, Iris virginica and Iris versicolor), their detailsare in http://archive.ics.uci.edu/ml/datasets/Iris. "iris_for_mlp_4inputs_1output.csv" is data with 4 inputs and 1 output, "iris_for_mlp_4inputs_3outputs.csv" is data with 4 inputs and 3 outputs.
- "CAD_USD_JPN.csv", "CAD_USD_JPN_Normalized.csv", 2489 datasets about exchange rate CAD ⇒USD, CAD ⇒JPN with 30 inputs and 2 outputs.
- NASDAQ_5026_data_15inputs_1output.csv is 5026 datasets of NASDAQ indexes with 15 inputs and 1 output.

3.2. Load Data

Suppose that we are using data in "sincos.txt" file, 100 datasets, 1 input and 2 outputs. In the "Number of Neurons and Data Sets", we select parameters as shown in fig. 5:

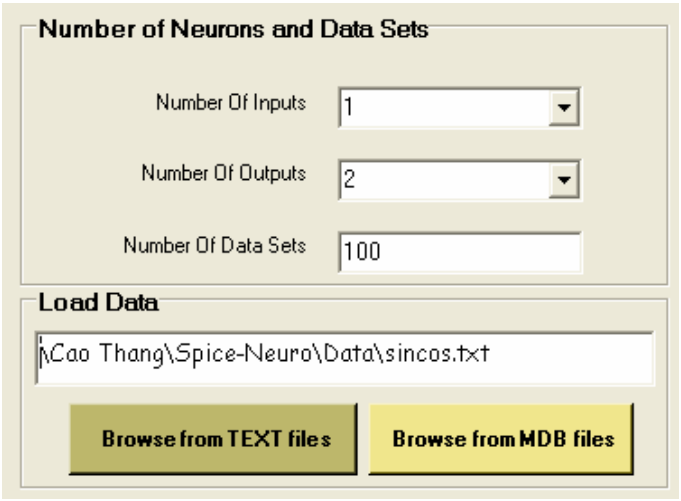
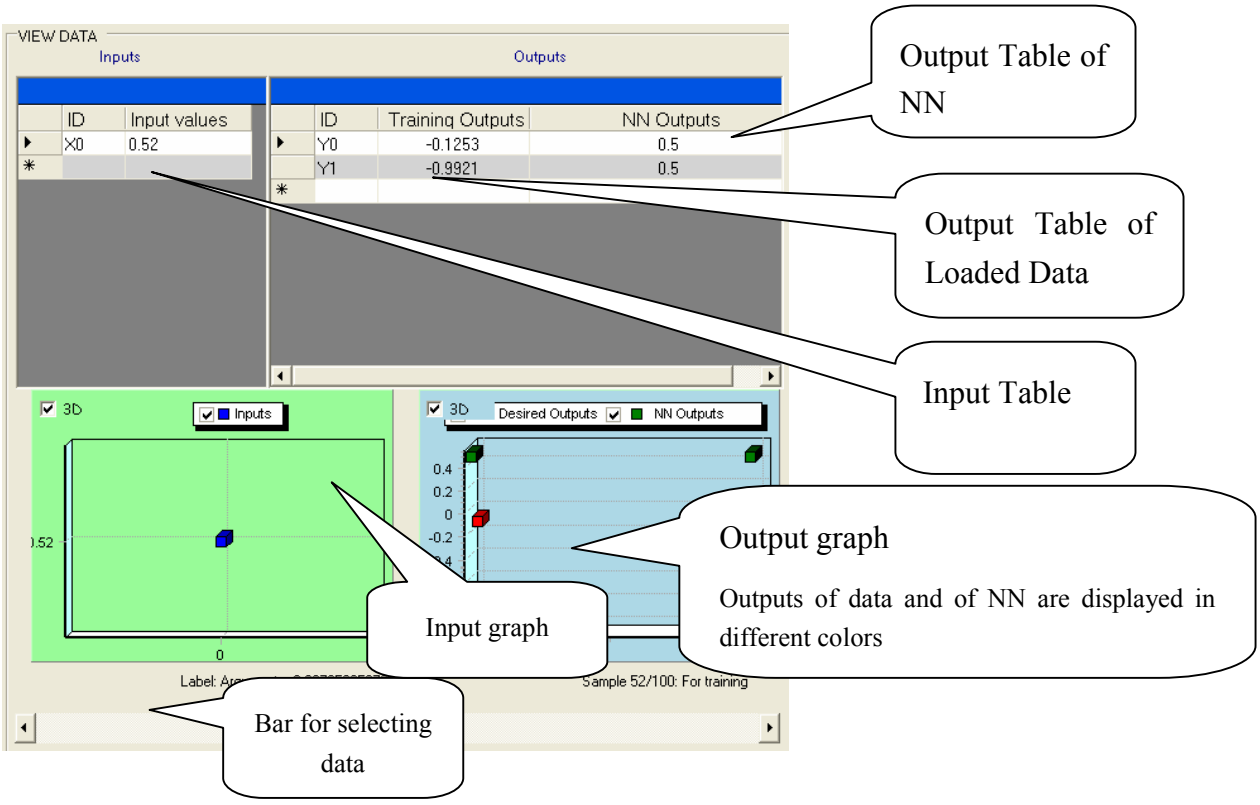


Fig. 5. Select parameters to load data

Select "Brose from TEXT files" button, data will be loaded into memory. In the "VIEW DATA" on the right, you can view each dataset of the loaded data:



3.3. Data Normalization

If your data is not normalized, you can use "data normalization" function as shown in fig. 7. You can normalize input, output data, or both.

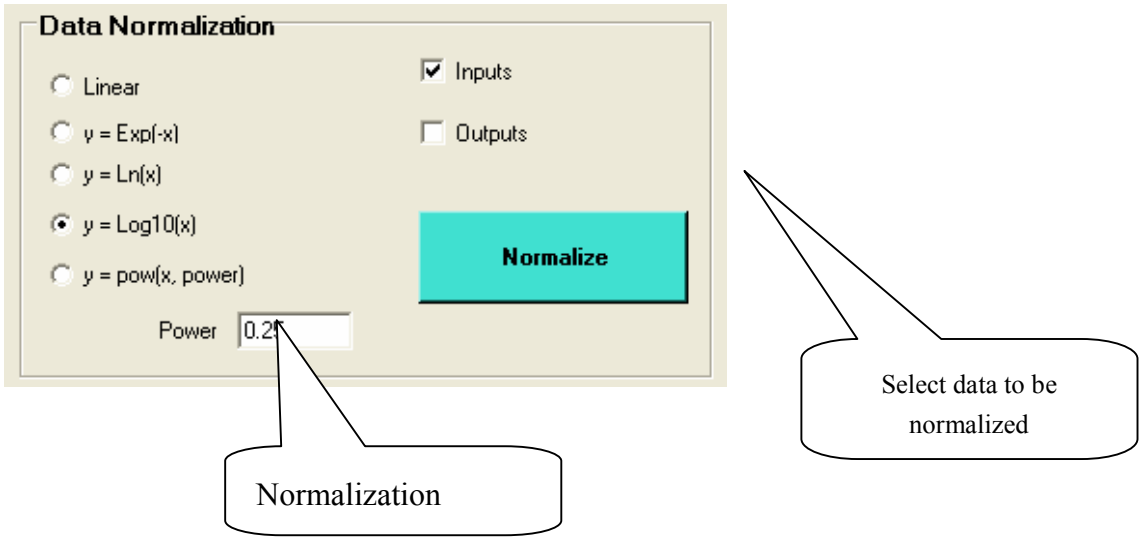


Fig. 7. Data Normalization

3.4. Training

3.4.1. Splitting Data

If you want to split data into two parts, training and testing parts, you can use "Splitting Data" function of Spice-Neuro. Fig. 8 illustrates randomly splitting data into parts of 70% and 30%.

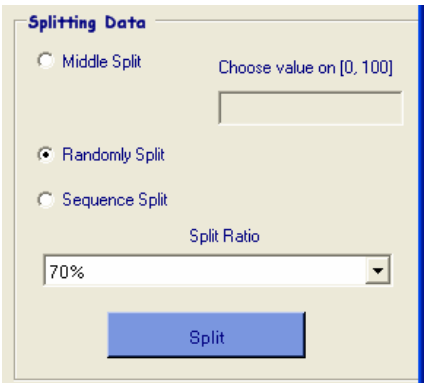


Fig. 8. Splitting Data

3.4.2. Select training data and parameters

Next, you should select number of neuron for hidden layer, number of iterations (or epochs), learning time, required MSE (Mean of Square Error). You can also select "Adaptive Learning" (learning rate varies depending on training MSE) and "Enable Testing" (testing while learning)

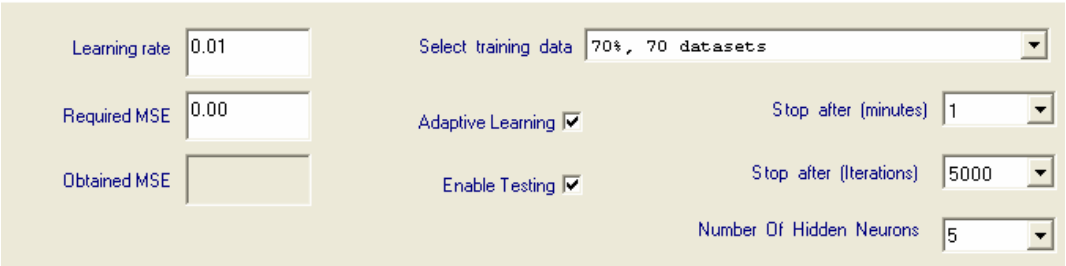


Fig. 9. Select Training data and Parameters

Activated Functions (AF): you should select Activated Functions for hidden and output layers. Spice-Neuro provides you many functions. If you are beginners, please select Sigmoid, HyperTanh, Tanh, ArcTan of, ArcSinh functions.

Selecting inputs in learning, you can select random or in turn inputs.

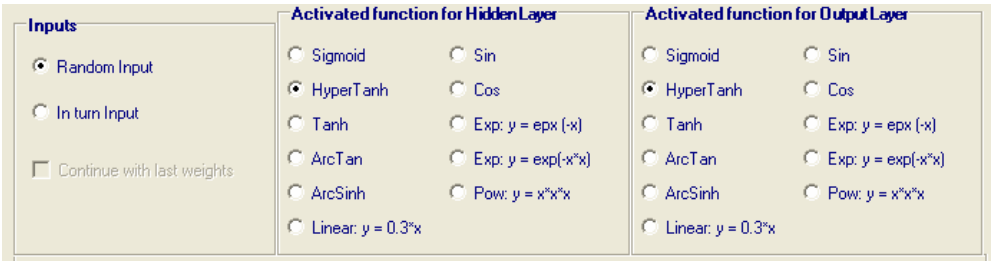
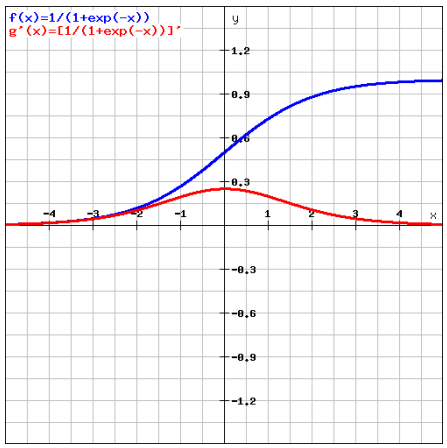
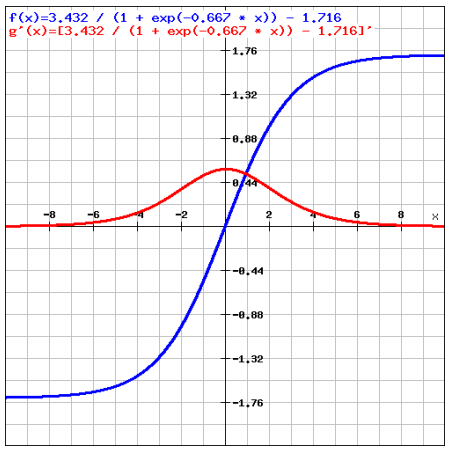


Fig. 10. Select Activated Functions and order of inputs

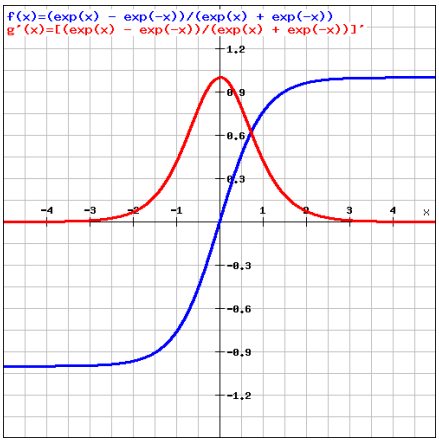
Formulas and shapes of the activated functions are in the following table 2.



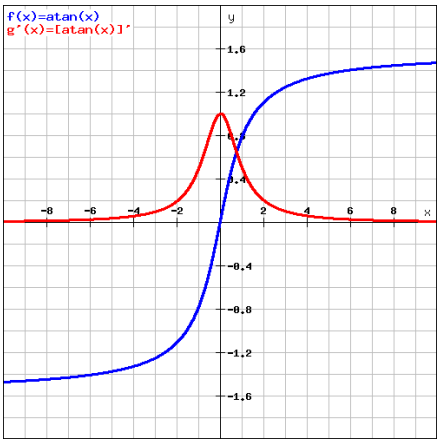
Sigmoid



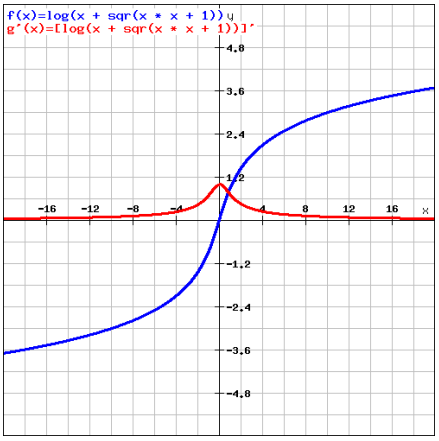
HyperTanh



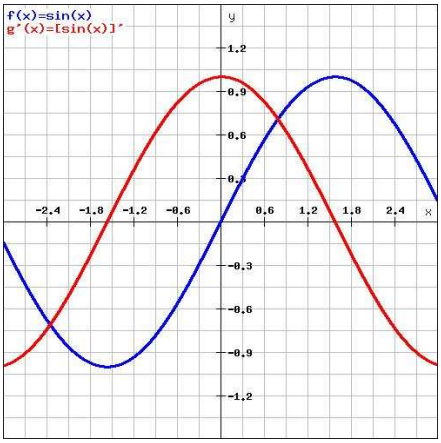
Tanh



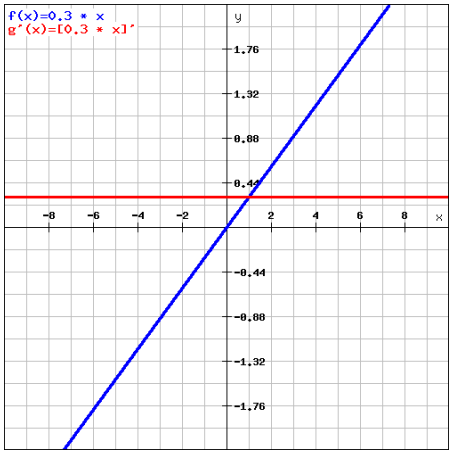
ArcTan



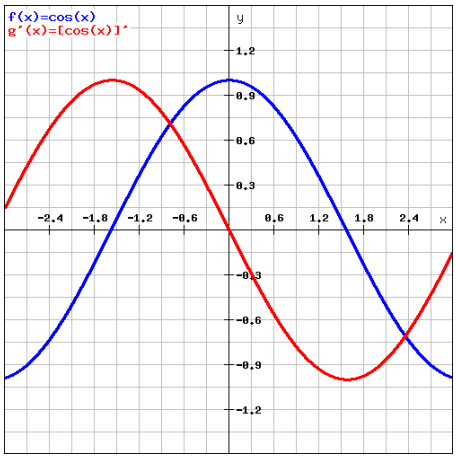
ArcSinh



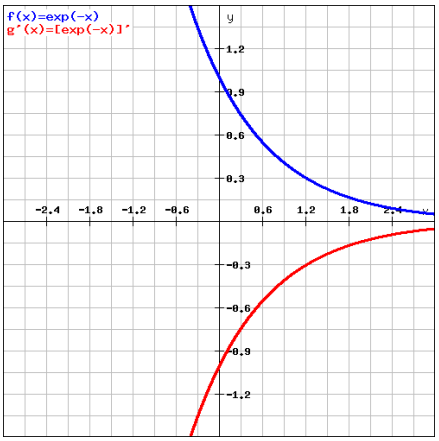
Sin



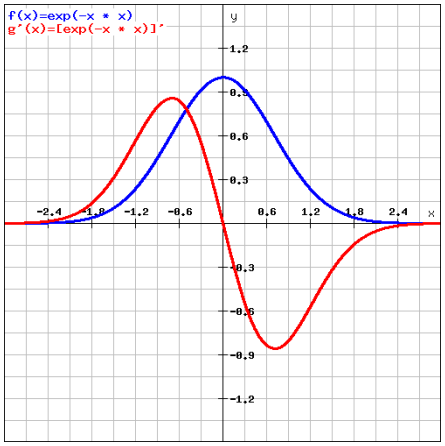
Linear



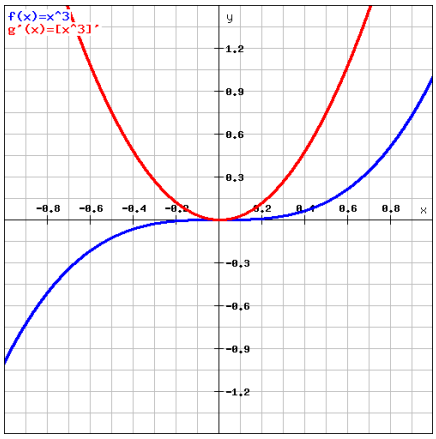
Cos



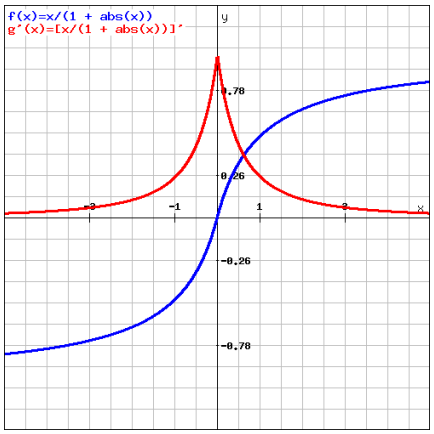
Exp(-x)



Exp(-x*x)



x*x*x



InvertAbs

3.4.3. Network Training

After selecting necessary parameters, you can begin to train the NN. Here are main buttons for the train.

- *Reset initial weights*: reset the weights for all connections of the NN.
- *Train*: train the NN.
- *Load Weight from Binary File*: load weights of a NN saved in a binary file. Note that if your NN parameters and the loaded NN parameters are different, the program may not work or give unexpected results.
- *Save Weight to Binary File*: save weights of current NN to a binary file. Each weight is written by 4-byte length (double value). The order of the weights are the following:

  Weight_IJ: I0J0, I0J1, …

  Weight_TetaJ: Teta_J0, Teta_J1, …

  Weight_JK: J0K0, J0K1, …

  Weight_TetaK:Teta_K0, Teta_K1, …

- *Save Weight to Text File*: save weights of current NN to a text file, so you can inspect values of each connection of the NN.

For example, weights of a NN with 2 inputs, 2 hiddens and 3 outputs are stored in a text file as shown in Table 3.

Table 3. Content of a NN with 2 inputs, 2 hiddens and 3 outputs

Weights of the NN. Activated Function was Hyperbolic Tangent

Saved on 2/4/2008 2:06:11 PM

Weight_IJ: array contain the weights in the first connections between input and hidden layers

	J0	J1
I0	2.26013784930634000000	3.92545735150867000000
I1	2.26013782334056000000	3.92545726757851000000

Weight_TetaJ: array contain the weights from Bias of neurons in hidden layers

Teta_J0	2.75109071396534000000
Teta_J1	0.85361332621648900000

Weight_JK: array contain the weights in the second connections between hidden and output layers

	K0	K1	K2
J0	-2.41085860219597000000	2.00917738577701000000	-0.40168121642706000000
J1	2.40321691280414000000	-1.07731644733275000000	1.32590046568425000000

Weight_TetaK: array contain the weights from Bias of neurons in Output layers

Teta_K0	1.85485947371567000000
Teta_K1	-1.98609819228695000000
Teta_K2	-0.13123871869928800000

After training, the last trained information of the NN is displayed in a text box on the right left.

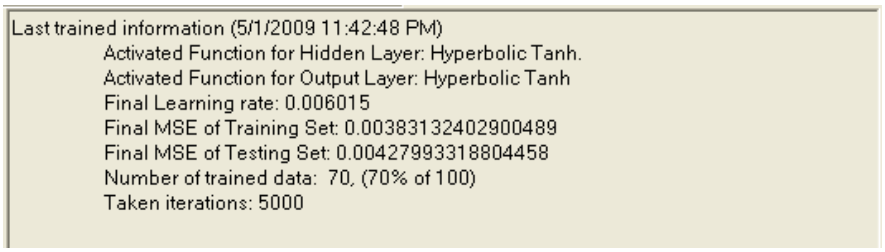


Fig. 11. The last trained Information

3.4.4.a. View the Error Graph

After training, you can view error graph by selecting the "See Final Training Graph" button or selecting checkbox "Show Detailed Training Graph" as shown in Fig 12. You can also save error data by the "Save" button as shown in Fig. 14.

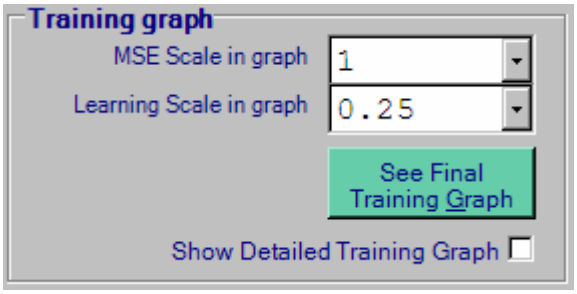


Fig. 12. View Error Data

If you select checkbox "Show Detailed Training Graph " before training, the error graph will be displayed online when training, however the training time will increase much because your PC need to graph together with to train the NN.

Figs. 13, 14 illustrate error graphs by selecting the "See Final Training Graph" button or selecting checkbox "Show Detailed Training Graph".

If viewing error data by checkbox "Show Detailed Training Graph", you can view detailed error for each iteration by clicking on the graph. Fig. 15 illustrates error graph on the iterations of 50 to 150.

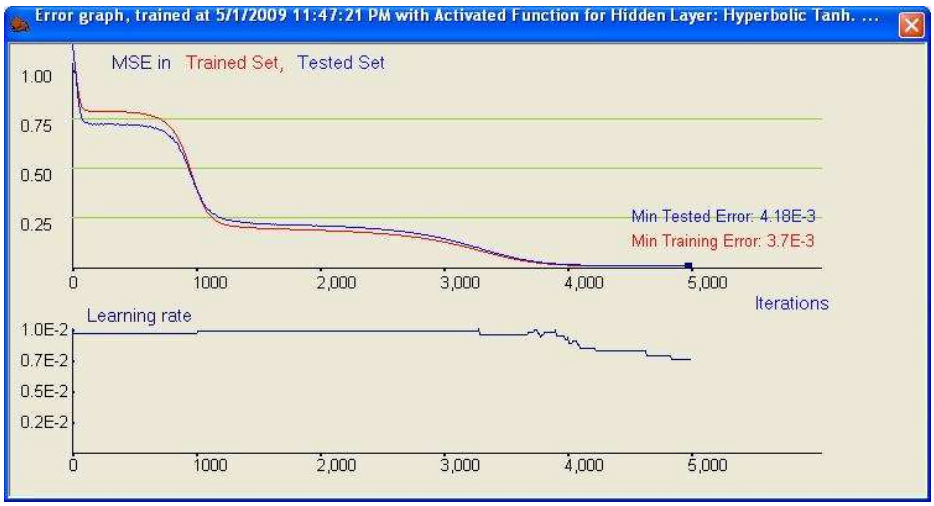


Fig. 13. Error graph viewed by selecting the "Show Final Training Graph" button

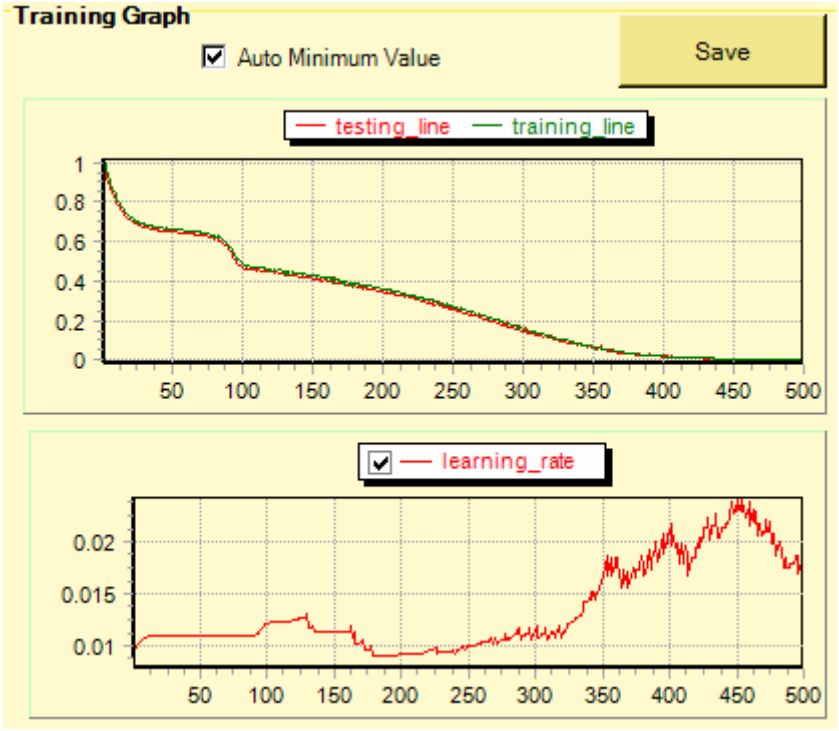


Fig. 14. Error graph viewed by selecting checkbox “Show Detailed Training Graph”

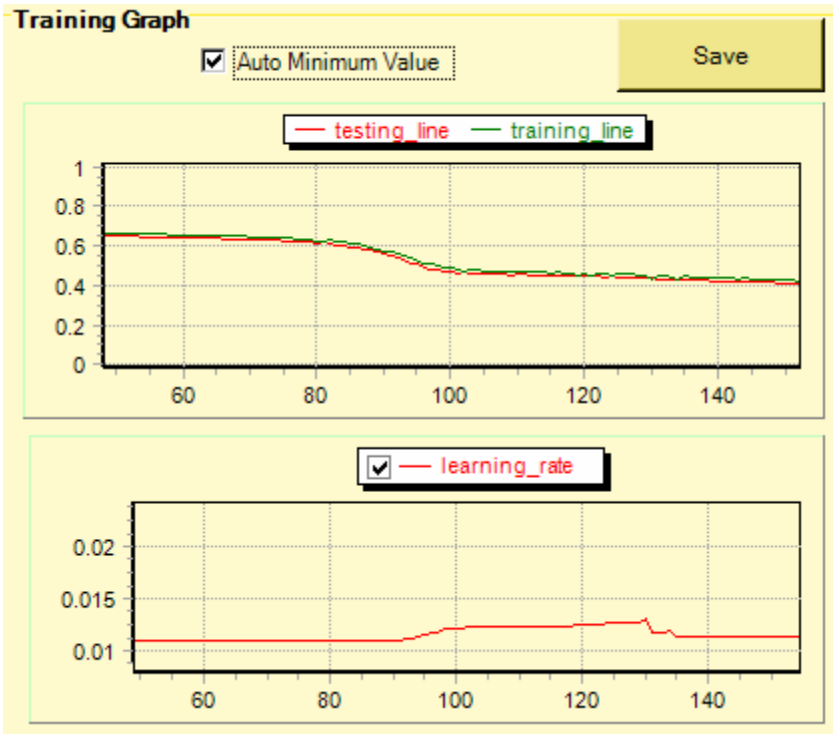


Fig. 15. Detailed Error Graph

Table 4. Error Data saved on text file

SPICE-NEURO by Cao Thang

Training Error

Last trained information (5/5/2009 6:34:04 PM)

Activated Function for Hidden Layer: Hyperbolic Tanh.

Activated Function for Output Layer: Hyperbolic Tanh

Final Learning rate: 0.006492

Final MSE of Training Set: 0.00363515839240401

Final MSE of Testing Set: 0.00488268815319086

Number of trained data: 70, (70% of 100)

Taken iterations: 5000

Iterations	TrainingError	TestingError	LearningRate
1	1.006932778	0.999370971	0.00918
2	0.989821872	0.990941336	0.00918
3	0.976928521	0.986117834	0.00918
4	0.969271326	0.981868921	0.00918
5	0.963313559	0.977746958	0.00918
6	0.958254105	0.973941811	0.00918
7	0.953665223	0.971387056	0.00918
8	0.949320804	0.967659064	0.00918
9	0.945233933	0.964940659	0.00918
10	0.941177193	0.961143105	0.00918
11	0.936883505	0.956635027	0.00918
…	…	…	…
4995	0.003240048	0.003576933	0.007842627
4996	0.003238211	0.003618928	0.007842627
4997	0.003234612	0.003602332	0.007842627
4998	0.003227556	0.003564963	0.007842627
4999	0.003221500	0.003589969	0.007842627
5000	0.003231968	0.003597997	0.007842627

3.4.4. b. Check weight graphs and averages of inputs of a neuron

Spice-Neuro allows users check the changes of some weights and average inputs of a neuron in training. Left part of Fig 16 illustrates selecting weight from neuron 0 of input layer to neuron 2 of hidden layer WIJ[0][2], and weight from neuron 3 of hidden layer to neuron 1 of output layer WJK[3][1]. Right part of Fig 16 shows selecting average input of neuron 2 of hidden layer and average input of neuron 1 of output layer.

Like selecting checkbox "Show Detailed Training Graph, if you select checkbox "Show These Graphs" before training, these graph will be displayed online when training, however the training time will increase much because your PC need to graph together with to train the NN.

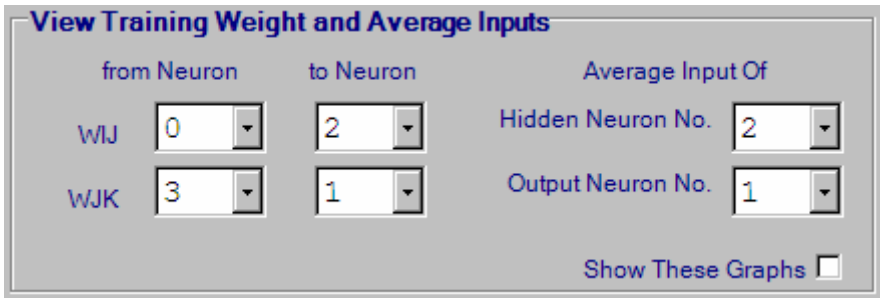


Fig 16. Select weights and averages of inputs of neurons to view

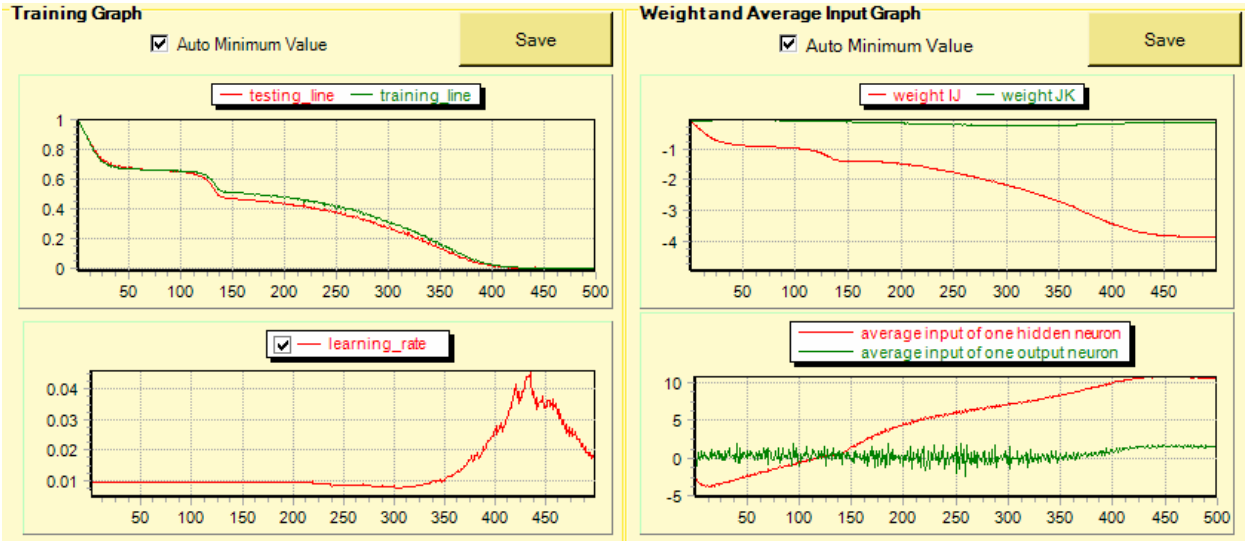


Fig 17. Graphs of errors, weights and input averages in training

Fig 17 illustrates error graph and graphs of two selected weights and averages of two inputs (as selected as in Fig 16). It is easy to note that when the network converges (when training error comes to 0), weights and averages of inputs of neurons also converge to stable values.

3.4.5. View graphs of inputs and outputs of data and outputs of NN

In the "Load Data" in 3.2, you can see values of each loaded dataset as well as output values of the NN. In the "Data Visualization" Tab, you can see graph of all data (if number of inputs and outputs are not large).

You can view graph of all data, training data only or testing data only. The following figures illustrate the input, outputs of Sin and Cos functions on [0, 2π], outputs of initial NN and output of trained NN. It is easy to see that in an initial stage, outputs of NN are around 0.5 for both outputs. However after training with 70% data, actual outputs of NN are roughly equal outputs of training data and have shapes of Sin and Cos functions.

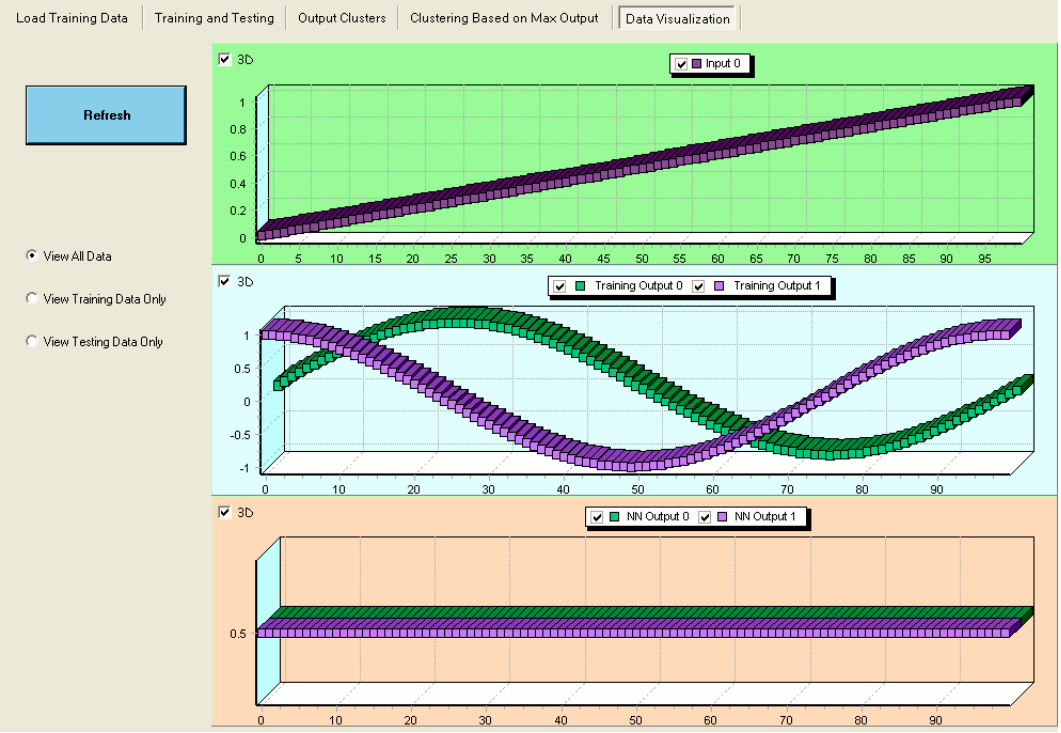
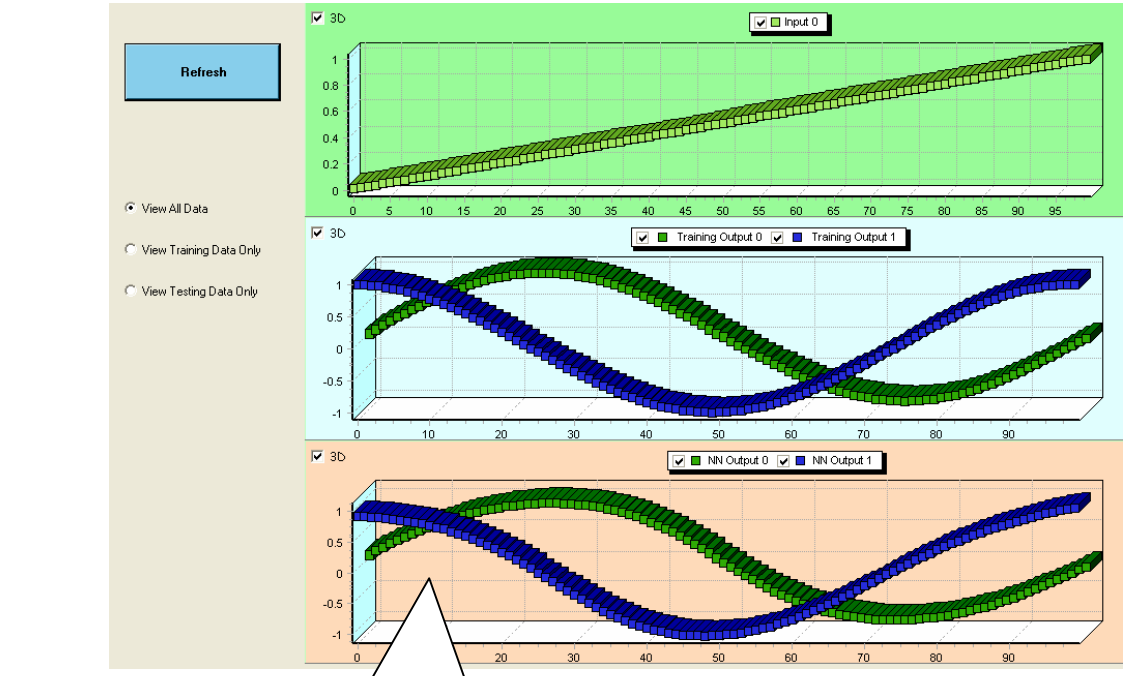


Fig 18a. Graphs of Inputs and outputs of training data, and outputs of initial NN



Actual outputs of NN roughly have shapes of the outputs of training data

Fig 18b. Graphs of Inputs and outputs of training data, and outputs of the trained NN

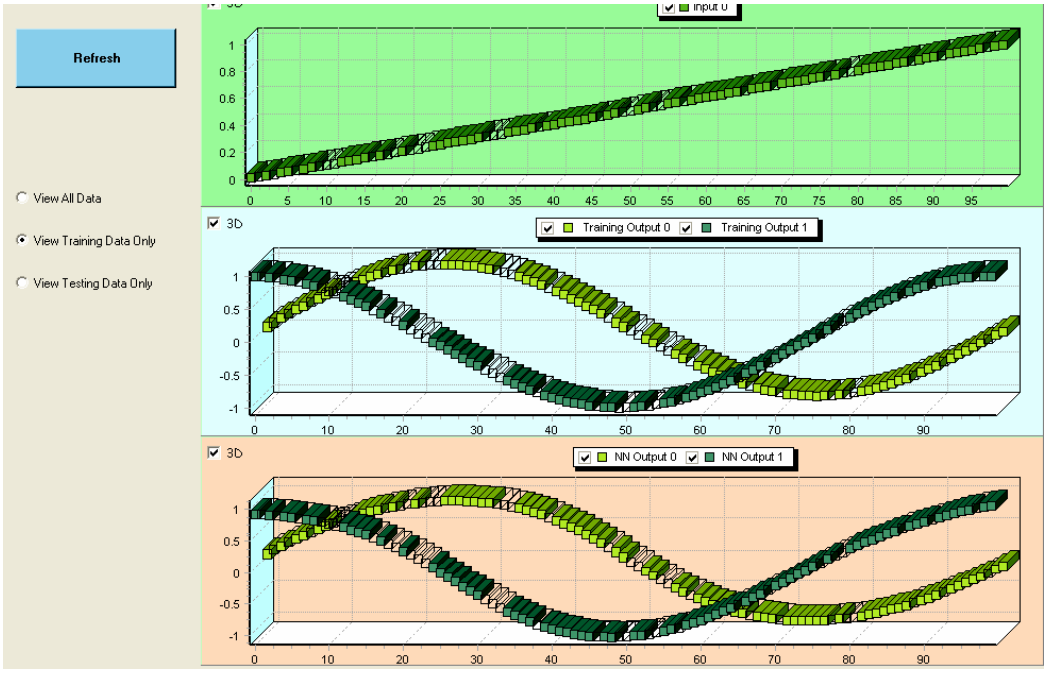


Fig. 19. View Training Data only

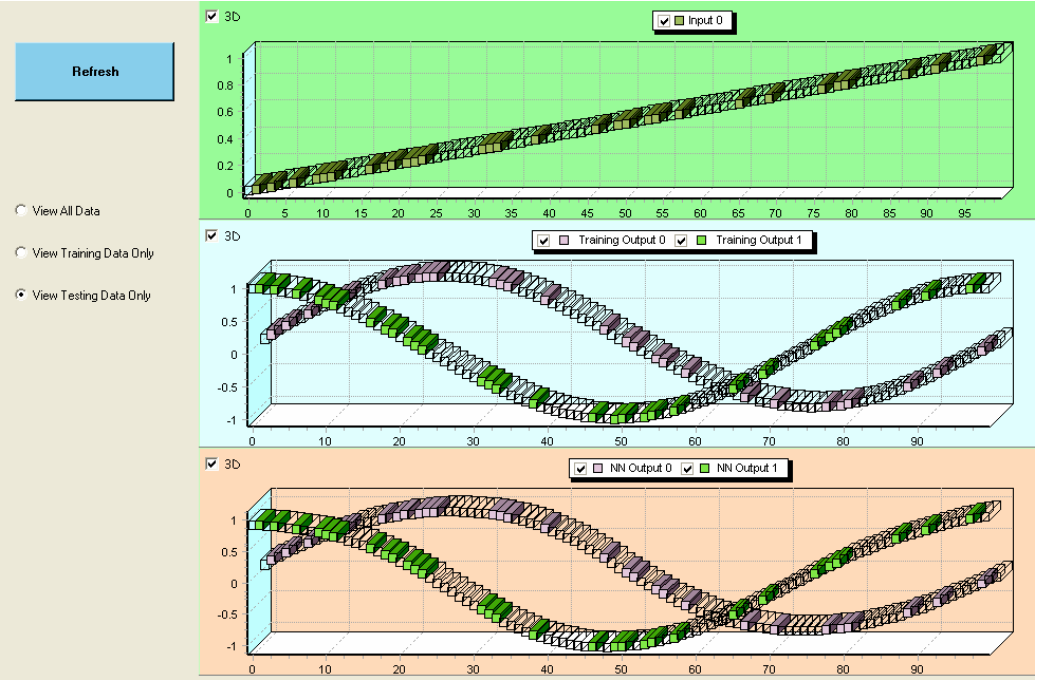


Fig. 20. View Testing Data only

3.4.6. Grouping data based on outputs of NN

You can group your data based on output values of NN. Spice-Neuro can group your data into 3 groups based on one selected output of NN, or group training data based on max output.

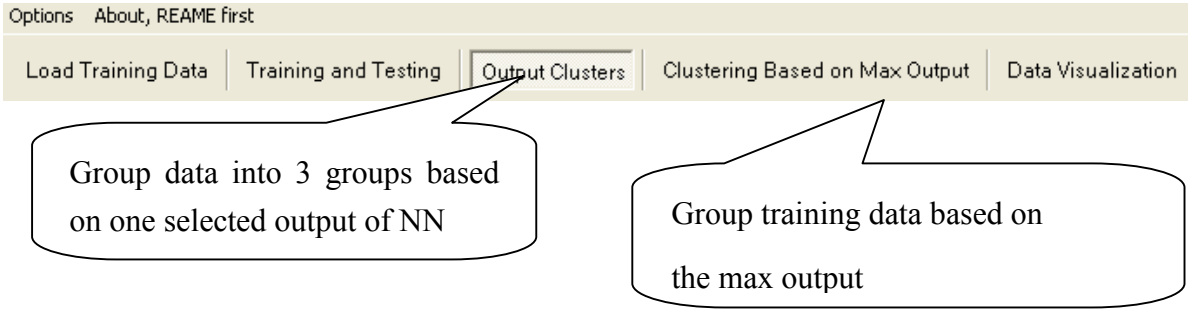


Fig. 21. Grouping Data

3.4.7. Save modeled data by NN

After training the NN, you can save modeled data by the NN. You can save modeled data with NN outputs only, or save modeled data with NN outputs and desired outputs.

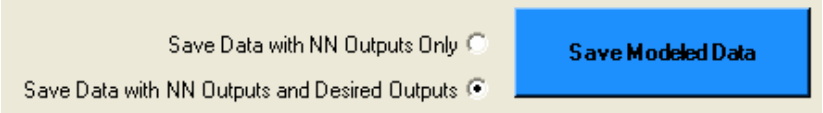


Fig. 22. Save data modeled by NN

4. Conclusions

This material briefly guides how to use Spice-Neuro, a Multi-Layer Neural Network Program. The author hopes that this software would be helpful for your study and research.

Having read this material, you may want to read neural_network_practical_use_en.pdf that illustrates classifications for face, pedestrians and car, stock price prediction, exchange forecast and other examples.

Thank you for using Spice-Neuro. If you want more functions in Spice-Neuro, please contact the author at http://spiceneuro.wordpress.com or spiceneuro AT gmail DOT com.

Thank you!