NNMODEL Contents

The following Help Topics are available:

For Help on Help, Press F1

File menu commands

The File menu offers the following commands:

New Data Matrix	Creates a new data matrix module
New Neural Model	Creates a new neural model module
Open	Opens an existing module.
Close	Closes an opened module.
Save	Saves an opened module using the same file name.
Save As	Saves an opened module to a specified file name.
Import	Imports either a Data Matrix or Neural Builder module.
Data From ASCII File	Imports a Data Matrix module from an external ASCII file.
Data From Clipboard	Imports a Data Matrix module from the clipboard.
Neural Model	Imports a Neural Builder module from an external ASCII file.
Test Matrix	Imports a new test matrix from an existing Data matrix module.
Export	Exports the selected module as an external ASCII file.
Data Matrix	Exports a Data Matrix module as an external ASCII file.
Full Neural Model	Export a Neural Builder module as an external ASCII file in ENN format. All internal values are written including the training and test data sets.
Feed-Forward Neural Model	Export a Neural Builder module as an external ASCII file in ENN format. No internal values, training and test data sets are written. This model cannot be re-trained.
Train/Test Matrix	Export the training and/or test matrix as an external ASCII file.
Measured vs Predicted	Export the measured vs. predicted values as an external ASCII file.
Weight Matrix	Export the weight matrix to an external ASCII file.
Diagnostic Dump	Dumps the module internals to an ASCII file of the same name as the module but with the extension DMP.
Print	Prints a module.
Print Preview	Displays the module on the screen as it would appear printed.
Print Setup	Selects a printer and printer connection.
Print Options	Selects options for printing data matrix modules
Exit	Exits NNMODEL.

Edit menu commands

The Edit menu offers the following commands:

Variable Descriptors	Allows the editing of a data or training matrix column descriptors.
Training Matrix	Allows the editing of the training and test matrices of a neural model.
Parameters	Allows the editing of a neural models training parameters.
Equations	Allows the editing of the calculated column equations.
Exclusions	Allows the editing of the exclusion equations.
Append Column	Appends a blank column at the end of the data matrix.
Append Calculated Column	Appends a calculated column to the end of the data matrix. The column equation must be present in the equation field.
Delete Column	Deletes a column in the data matrix.
Recalc	Rescans the data and recalculates scaling factors.
Copy	Copies data from the module to the clipboard.
Paste	Pastes data from the clipboard into the module.
Find	Finds the cell that matches the search criteria
Find Next	Find the next occurrence.

View menu commands

The View menu offers the following commands:

Toolbar	Shows or hides the toolbar.
StatusBar	Shows or hides the status bar.
Goto Record #	Go to the selected record number in the data matrix.
Goto End	Go to the end of the data matrix.

Model menu commands

The Model menu offers the following commands:

Initialize	Initialize the neural model.
Start Training	Start/Stop training the neural model.
Use Test Matrix	Use the test matrix for all graphs & reports.
Load Last Saved	Load the last saved version of the neural model.
Interrogate Model	Test the model interactively.
Statistics Report	Generate a statistics report.
Sensitivity Report	Generate a sensitivity report.
MVP Train Graph	During training display a measured vs. predicted graph rather that the standard plot.
No Train Graph	Dont plot the training progress graph.
Use Ext Binary File	Use the external binary file component of a very large data matrix.

Data menu commands

The Data menu offers the following commands:

Basic Statistics	Generate a basic statistics report on the current data matrix
Correlation Analysis	Generate a correlation matrix report on the current data matrix
Concatenate Datas	Combines two Data Matrices
Merge Datas by Time/Date	Combine two data matrices by the time/date stamp.
Reserve Testing Data	Reserve part of the current data matrix for the later testing of a neural model.
Fill Missing	Fills in missing values using the previous non-missing value found in the column.
Check Sequential Date	Validate that the rows are in acceding time/date sequence.
Load Sparse	Loads a sparse Data Matrix
Load Rejects	Load rejected data points from the MISSING data matrix into the current data matrix
Randomize	Scales and adds a random component to the selected columns.
Best Model Search	Search the data matrix for the best model of the selected output.

Graphs menu commands

The Graphs menu offers the following commands:

Options	Allows modifying of all user changeable graphic options.
ByRow	Generates a graph of a selected variable (Y axis) and is plotted against the row number (X axis).
Scatter Plot	Generates a XY scatter plot of two variables.
Distribution	Generates a frequency distribution plot of a variable.
Meas vs Pred	Generates a graph of the measured verses predicted outputs of a model
Meas & Pred	Generates a graph of the measured and predicted outputs of a model
Residuals	Generates a graph of the difference between measured and predicted outputs of a model.
XY	Generates a line graph showing how a model output (Y) variable changes in response to an input (X) variable.
XY Effect	Generates a family of line graphs showing how a model output (Y) variable changes in response to an input (X) variable and an effect (E) variable.
Contour	Generates a topological style graph showing how the neural output is effected by both an X and a Y (neural inputs) variables.
3D Surface	Generates a 3 dimensional surface plot of how a neural output is effected by 2 input variables (X and Y).
3D Scatter	Generates a 3 dimensional scatter plot.
ByRow Matrix	Generates a graph per variable plotted against the row number.
Scatter Matrix	Generates all combinations of XY scatter plots.
Distribution Matrix	Generates the frequency distribution graphs of all variables.

Window menu commands

The Window menu offers the following commands, which enable you to arrange multiple views of multiple modules in the application window:

Cascade	Arranges windows in an overlapped fashion.
Tile Horizontal	Arranges windows in non-overlapped tiles.
Tile Vertical	Arranges windows in non-overlapped tiles.
Arrange Icons	Arranges icons of closed windows.
Window 1, 2,	Goes to specified window.

Help menu commands

The Help menu offers the following commands, which provide you assistance with this application:

Index	Offers you an index to topics on which you can get help.
Using Help	Provides general instructions on using help.
About	Displays the version number of this application.

How to

Begin using the NNMODEL.
Create an undesigned data matrix.
Create a designed data matrix.
Import a data matrix from an ASCII file.
Import a data matrix from a spread sheet program.
Load historical data into a designed data matrix
Combine two similar data matrices.
Use calculated columns
Use the time or date fields.
Find errors in a data matrix.
Copy data, graphs and reports to other Windows based products.
Create a neural model.
Analyze a neural model.
Use a neural model in an external C program.

Reference

Dialog Boxes

Example Data Sets

The following table is a summary of the example datasets provided in the TESTSET sub-directory.

Name	Columns	Rows	Description
LOGIC	5	4	Contains a truth table for 2 input logical AND, OR and XOR operations.
ENCODE	11	8	Contains a truth table for an 8 input to 3 output binary encoder. If the inputs and outputs are reversed then the table becomes a 3 to 8 decoder.
AIR	13	108	Contains a table of number of airplane tickets sold by month for 9 years. The table is arranged 13 months wide with the first 12 months being the previous 12 months and the NEXT column being the next months number of seats. The final 12 rows are reserved for testing.
VEL	7	609	Contains the distance traveled by a projectile using different angles and initial velocity. Additional columns of angle and velocity are included with random noise added.
COATING	8	128	Contains the results of a coating experiment. Different levels of starch, latex, coating weight, bonding agent and calender pressure are visited and the effects on opacity, brightness and gloss are recorded.
SODIUM	6	220	Contains the results of a designed experiment. Different gases and mixtures were tested to see what combination of gas, time and temperature could be used to convert Na2SO4 to Na2S the most yield in the shortest time.
REDWOOD	15	72	Contains the results of a designed experiment. Different species of wood chips were tested to see if less expensive mixes could be used to make paper board while still guaranteeing a minimum strength and yield.
RING	15	507	Contains a process log of 14 sensors from a paper machine along with one laboratory measurement. The purpose of the log is to see if any process variables could be used to predict the lab variable.
SPECIES	5	2000	Contains a process log of 4 sensors along with 1 field that calculates the wood species exiting a wood digestor.
NOX	23	1340	Contains a process log of 23 sensors from a power boiler. The purpose of the log was to see if the process variables could be used to predict stack gases emitting from the boilers smoke stack.
CLO2	6	30	Contains the results of a designed experiment. Different levels of chemicals were tested to find the ideal setpoints needed to produce ClO2 most efficiently.
CLOSTAT1	9	15	Contains the results of a designed experiment. Different stream setpoints were simulated to find the most economical setpoints.
PEAK4	3	121	Contains the results of stepping angles X and Y (11 steps) from 0 to and evaluating Z = sin(X) sin(Y)
CURL	9	70	Contains the results of a designed experiment. Paper machine variables were varied to discover any major effects on paper curl.
STR4	24	1178	Contains a process log of a paper machine. The purpose of the log was to see if the process variables could be used to best predict strength properties.

To import any of the aforementioned datasets into the NNMODEL issue the **Import Data From ASCII File** command from the File menu. The files are found in the \nnmodel\testsets sub-directory. Once a raw file has been imported the data matrix can be saved in binary format and reloaded at any time using the Save or Open commands in the File menu.

NNMODEL Introduction

NNMODEL is a effective modeling tool that discovers relationships from a database of examples. It then generates a compact module that captures from simple linear to complex, non-linear relationships in a form that is readily executable. It excels at modeling systems such as fuzzy estimation, probabilities, expert experience and judgment, process sensor measurements and control strategies.

NNMODEL is easy to use because it automatically constructs mathematical models directly from your data. NNMODEL is cost-effective, enabling you to create prototype models quickly and inexpensively. NNMODEL assists you in developing accurate models in less time, even if you have limited experience with statistical regression or neural networks.

NNMODEL is designed to help you get maximum benefit from powerful modeling techniques without requiring you to learn a complicated software package or statistical language. It does not require proficiency in neural networks, artificial intelligence, advanced mathematics or statistics. Thus, you can learn how to use NM and start solving real world problems within a few hours.

The NNMODEL application is comprised of four modules: the Data Matrix Editor, Model Builder, Graphic Display and Report Display modules. The Data Matrix Editor (DME) allow you to create data matrices by either importing external data or designing a matrix and entering the data. DME has many features to help you edit, manipulate, analyze and graph your data. The Neural Builder allows you to create, train, analyze and graph neural models from a previously created data matrix. NNMODEL can export a trained model either as a static feed-forward playback module (training and test matrices not included) or as a fully dynamic BEP network. The Graphic Display module draws the selected graph in a window and allows you to copy it to the clipboard , preview or print the graph. The Report Display module is a very simple text based editor that allows you to see, edit and print a report.

Building a model with NNMODEL involves six easy steps: design a data matrix (optional), import example data, verify the data, create a neural model, analyze the performance and use the resulting model.

1 **Design a Data Matrix (Optional)**
The Data Matrix Editor allows you to create a data matrix based on a statistically designed experiment. A designed data matrix will allow you to squeeze the most information from a finite number of observations. The types of designs available are: two level, three level, simplex, star-simplex, central composite and multilevel. The designed data matrix can be created as an empty shell for later loading by the sparse data loader or fully initialized (factors set to design points and responses set to *missing*. The design can be printing with the rows in random order to aid in minimizing measurement errors. To follow a step-by-step procedure to create a designed matrix select Create Design Step

2 **Import Data**
NM builds models from binary numeric data matrices of continuous or discrete values. A data matrix must contain example sets of independent variable (inputs) and corresponding dependent variables (outputs). Scaling to and from internal neural representation is done automatically based on the observed minimum and maximum values guaranteeing maximum accuracy. You can create additional columns of data using the algebraic equation parser to use built-in functions, such as, sin, cos., tan, log, exp., lead, lag, running average or develop your own. These routines were designed to save time and money in pre-processing the data. To follow a step-by-step procedure to import a data matrix select Import Data Step

3 **Verify Data**
Once you have imported a data matrix, use Data Matrix Editor to analyze and repair (if needed) the data matrix. You can create additional columns of data, search for out-of-

spec variables, report the basic statistics (such as: number of observations, maximum, minimum, mean, standard deviation and sum of the squares), do a correlation analysis and view the data graphically using a variety of plotting routines including a thumbnail view of all the data. To follow a step-by-step procedure to validate a data matrix select Verify Data Step

4 Create Model

Once you have verified the data matrix, use NNMODEL to automatically develop a model. The NNMODEL combines a back-error propagation neural network with an advanced statistical based hidden neuron growth heuristic to outperform established methodologies on a wide range of problems while remaining statistically conservative. Unlike standard statistical regression, NNMODEL automatically develops an excellent functional model. You are not required to enumerate the functional form in advance. Because of NNMODELs advanced learning algorithm, the iterative modeling process necessary with many standard neural networks packages or statistical approaches is vastly curtailed or eliminated. To follow a step-by-step procedure to create a model select Create Model Step

5 Analyze Model

After you develop a model, use NNMODEL to analyze the performance of the model on an separate test data matrix. For each set of input values in the test matrix, residuals are calculated by comparing the test data matrix outputs to the corresponding outputs computed by the neural model. The resulting statistical and graphical displays enable you to quickly determine how well the model will perform in its final form. Basic performance statistics, sensitivity analysis reports along with many graphical analysis features round out the NNMODELs analytical tools. To follow a step-by-step procedure to analyze a model select Analyze Model Step

6 Use

You can use the models developed on-line using the interrogation feature built into NNMODEL, or export them as either trainable or non-trainable ENN files. The ENN files can be used by the supplied royalty-free C source code subroutines. This code was written in standard ANSI C and should be compatible with all major compilers. This enables you to immediately combine neural models with your own code. The playback code is compact and ideal for many real-time applications where floating point operations can be tolerated. To follow a step-by-step procedure to use a model in an external C program select Use Model Step

The Neural Builder Module

NNMODEL allows you to create, train, analyze and graph neural models from a previously created data matrix. Once a neural model has been created you can edit the training parameters to select the type of training algorithm, number of hidden neurons, additional connections, and many more internal parameters. You can edit the training and test matrices to delete/add rows or edit individual cells in the matrices.

NNMODEL has many features for analyzing, graphing and testing the performance of a trained model. For example, the analytical tools allow you to do statistical performance analysis, sensitivity analysis, model interactive interrogation, graphical plots of residuals, measured verses predicted, model output by variable, multiple variables in families of curves, 3D surface maps, contours maps, scatter and distribution plots.

Once you are satisfied with the performance of a model, you can continue to use the models results in the interactive interrogation feature or export the model either as a static feed-forward playback module (training and test matrices not included) or as a fully dynamic (retrainable) BEP network. The export routine writes the model as an ASCII file in a format that can be easily transported to many different hardware architectures.

The neural module contains the neural network structure, the data matrix used for training and an optional data matrix to be used for testing. The network structure contains the training parameters, weight matrices and other dynamic variables needed to train and playback a BEP network. The complete Neural Builder module can be saved on disk (in binary format) as a file with the extension of BEP. The neural model can also be exported to a transport ASCII file (ENN) that can be used by external programs linked with the NM C NNLIB.

A complete list of commands for the NNMODEL application is available in the Contents section.

The Data Matrix Editor Module

The Data Matrix Editor module contains the imported data stored in a binary matrix format. The descriptive information about the columns or variables is stored in column descriptor structures and descriptive information about rows or observations are stored in a row descriptor structure.

The data is presented in a spread sheet format with each variable displayed as a column and each observation displayed as a row. The first column is the row descriptor or type. This is a one character field specifying how the row is to be used (such as: training, testing, excluded, deleted, rejected or designed type). The descriptive information about the column can be viewed or edited by the menu command Edit/Variable Descriptors.

Each cell in the data matrix can be edited, columns and rows can be added or deleted. Columns and rows can be copied and pasted using the clipboard function. If a cell contains no numbers a single decimal point appears indicating the value is missing. A missing value will cause the entire row to be dropped if the column is selected in a graph or neural model.

The DME saves the data matrix data on disk in binary format with the file extension of DM.

A complete list of commands for the NNMODEL application is available in the Contents section.

The Graph Display Module

A graph can be created from either the Data Matrix Editor or Neural Builder module. At the time of creation all the information necessary to draw the graph (data, labels, options, and the type of graph) is stored in a graphic structure. The Graph Display module can replay this structure and allow you to change the options, such as, line width, character size, rotation, etc. The data, however is stored in one or more static vectors and cannot be changed.

The Graph Display module allows you to copy the graph (as a Windows metafile) to the clipboard for inclusion into other Windows based products or print it. The graphics display file can be saved to disk for later replay. GD module files are stored in binary format with the file extension of GRF.

A complete list of commands for the NNMODEL application is available in the Contents section.

The Report Display Module

Basic Statistics, Correlation Analysis, Model Performance Statistics or Sensitivity reports are written to this primitive Windows based editor. Each record is delimited by carriage return followed by a line feed. The statistical results are printed in columns separated by tabs. The editor allows you to made changes to the report, cut and paste to other Windows based products or save the results to a flat ASCII file. RD module files are stored as ASCII text with the file extension of RPT.

A complete list of commands for the NNMODEL application is available in the Contents section.

Create Design Step

Follow the next 8 steps to create a two-level designed data matrix consisting of three factors and two responses.

1 Select **New Data Matrix** command from the File menu. The Create Design dialog box will appear.

2 Enter TEST into the **Title** field. This will become the name of the data matrix when finished.

3 Change the **Design Type** to Two Level. (Note: the field that was originally labeled **# of Columns** has been changed to **# of Factors** and the field labeled **# of Rows** has been changed to **# of Responses**).

4 Change the **# of Factors** field to 3 and the **# of Responses** field to 2. (Note: **# of Runs** field has been updated to 8 and the variable descriptor grid at the bottom of the dialog has been updated to reflect the # of factors and responses).

5 Move the cursor to the first variable label in the grid and change the variables F1 to IN1, F2 to IN2, F3 to IN3, R1 to OUT1 and R2 to OUT2, Also change The minimum values of IN1, IN2 and IN3 to 0.

6 Press **OK**. The designed data matrix will be created.

7 Enter the following numbers into the OUT1 column (starting at the top). 0,1,1,0,1,0,0,1 and enter the following numbers into the OUT2 column 0,1,1,2,1,2,2,3. As you can see OUT1 is an even parity function and OUT2 is simply the sum of the inputs.

8 Press the diskette icon in the toolbox to save the changes.

Import Data Step

Follow the next 4 steps to import data from an ASCII file (CIRCLE.RAW) stored in the sub-directory \NNMODEL\TESTSETS.

1. Select **Import** menu from the File menu, then select **Data From ASCII File** command from the import sub-menu. A **File Open** dialog will appear.
2. Navigate to the TESTSETS sub-directory. A number of files will appear in the file list.
3. Select the file CIRCLE.RAW from the file list and press **OK**. The **Data Import** dialog box will appear.
4. The file CIRCLE.RAW is in standard RAW data format, therefore, no other cleaning options need be selected. Press **OK** to create the data matrix.

Verify Data Step

Use the following procedure to validate the data imported from the ASCII file BADDATA.RAW.

1. Import the file BADDATA.RAW found in the TESTSETS sub-directory.
2. Select **By Row Matrix** command from the Graphs menu. View the thumbnail graphs looking for outliers, scaling problems or patterns that might indicate clustering or cycling.
3. In this example all three variables have a point that is bad. The bad point in the ANG column was the result of an incorrect minus sign and the bad points in the X and Y columns were the result of misplaced decimal point. Find the bad points and correct them.
4. Next select the **Scatter Matrix** command from the Graphs menu. The graph plots each variable against every other variable in the data matrix. As you can see there are some obvious interdependencies in this data (sin and cosine waves are clearly present).
5. Select the **Distribution Matrix** command from the Graphs menu. As you can see the distribution is spread evenly throughout the data space.
6. Select **Correlation Analysis** from the Data menu. By viewing the Pearson Correlation Coefficients for each valid combination there does not seem to be simple linear relationship between any of the variables.

Create Model Step

Use the following procedure to create a new neural model from the data matrix CIRCLE.DM.

1 Select the **New Neural Model** command from the File menu. An open file dialog labeled **Data Matrix File** will appear.
2 Navigate to the sub-directory TESTSETS and open the data matrix named CIRCLE.DM. The **Create Neural Model** dialog will appear.
3 Select ANG and press **Add In** to add it as an input to the model. Select both X and Y and press **Add Out** to add them as outputs from model.
4 Press **OK** to create the model. The Neural Model window will appear.
5 Select the **Parameters** command from the Edit menu. The **Edit Training Parameters** dialog box will appear. Select the **Standard BEP** radio button in the Training Type group. Select **Connect Inputs to Outputs** radio button. Change the **Max Training** to 5000 and press **OK** to save the changes.
6 Select the **Initialize** command from the Model menu.
7 Select the **Start Training** command from the Model menu. The training will continue until 5000 presentations of the training matrix has been completed.
8 Press the Diskette icon in the toolbar to save the trained model.

Analyze Model Step

Use the following procedure to validate the model created by the Create Model Step.

1 Select **Open** command from the File menu. A **File Open** dialog box will appear. Change the **File Type** from Data Matrix to Model and navigate to the TESTSETS sub-directory, select CIRCLE.BEP and click on **OK**. The CIRCLE.BEP model window will appear.

2 Select **Statistics Report** from the Model menu. Study the statistics, a good model will have an R Square near 1.0 with all the residual statistics near zero. Of course this will never happen in real life (if it does you probability did something wrong).

3 Select **Meas vs Pred** command from the Graphs menu. Plot both the X and the Y variables. Notice how close to the center line the points fall. A good model will have all (or most) of the points falling near the center line. Change the **Tolerance** value in the **Options** menu to .1 and view the 10% tolerance lines.

4 Select **Meas and Pred** command from the Graphs menu. Plot both the X and the Y variables. Notice how the measured and predicted variables overlay.

5 Select **Residuals** command from the Graphs menu. Plot both the X and the Y residuals. Residuals are the difference between the predicted and measured values, also called model error. Errors can be caused by either random noise in the system or by the models inability to accurately predict the response due to missing factors or too few number of parameters (hidden neurons). Notice in this example there is definitely a sinusoidal component missing.

Use Model Step

The following procedure to interrogate the model created by the Create Model Step using an external DOS program.

1. Select **Open** command from the File menu. A **File Open** dialog box will appear. Change the **File Type** from Data Matrix to Model and navigate to the TESTSETS sub-directory, select CIRCLE.BEP and click on **OK**. The CIRCLE.BEP model window will appear.
2. Select the **Export** sub-menu from the File menu and then select **Feed-Forward Neural Model**. The **Export FeedForward Network As** dialog box will appear. Navigate to the NNLIB sub-directory, enter the name CIRCLE.ENN into the **File Name** box, and press **OK**. This will export the circle model to an ASCII file.
3. Switch to the Program Manager and open a DOS session.
4. Change directory to \NNMODEL\NNLIB
5. Execute the program ITEST.EXE and when prompted enter the filename CIRCLE. ITEST loads the model and prompts you to enter a value for the input variable ANG. Then it displays the predictions for the variables X and Y. This will continue to loop until you type in the word END.

Of course this example program is primitive and fairly useless in that the model could have been executed by using the **Interrogate Model** command in the Model menu. However, you can use this program a starting point to build up your own sophisticated multi-model simulation.

Features of Future Releases

Multi-Module Optimizer

Combine one or more neural models with algebraic equations to minimize or maximize any combination of inputs, outputs or cost functions. The optimizer utilizes a Monte Carlo started constrained conjugate gradient algorithm to minimize the objective function. The objective function can be constructed from any or all inputs or outputs along with their polarity (min or max) and their relative weight. Inputs can be constrained rectilinearly, outputs are constrained by a penalty function. Results of the optimizations can be viewed using the interactive interrogation module, graphically or by the results log.

Data Mining Utility

Allows the user to set up a historical data matrix, identify variables as factors, responses or unknown, time position (up or down stream) in time units, use full database for model or select records from the database based on goodness of fit to a multi-level design, pick the best factors for inclusion into the model based on model performance, include or exclude factors for any model based on prior knowledge, report results of search. Possible RISC based add-on processor.

Multi-Module Simulator

Combine one or more neural models with interpreted algebraic equations or pre-compiled user subroutines (user creates a DLL file). Simulator is an OLE container that can link with many graphical display modules and VBX controls. The simulator is designed using the source/sink concept. Data sources are ASCII files, OLE or DDE modules, models or equations. Sinks are reports, graphs, meters, equations or models.

Attribute Data

Automatic conversion of Attribute data to a continuous variable based on a user defined rank or conversion to discrete logical variables (1 or 0). The continuous variable simply becomes one input to the model. However, the discrete variable creates as many inputs as there are states.

Real Time Data Matrix Loader

Using the DDE interface automatically load a designed data matrix. The data matrix can be exported to be used to build a neural model. The neural model can then be used to control the process monitored by the DDE source.

OODB Linkage

Allow the data matrix to be created directly from an open OODB database such as Microsoft Access.

New Data Matrix command

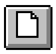
File Menu

Use this command to create a new data matrix module.   Select the type of designed experiment with the number of factors, number of responses, number of center points, etc. you want to create in the Create Design dialog box.

Other methods of creating (non designed experiment) data matrix modules are by importing directly from an external ASCII file or by importing from the Clipboard.

You can open an existing data matrix module with the Open command.

Shortcuts

Toolbar: 

Keys: CTRL+N

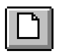
New Neural Model command

File Menu

Use this command to create a new neural model module. Select the inputs and outputs for the model in the Create Model dialog box . New neural models are created from an existing data matrix module. If a data matrix already open then the create model dialog will use it as the source of the training data. Otherwise an open file dialog will allow you to select the source. The new model dialog allows you to select the inputs, outputs and enter any exclusion equations that you want executed during the loading of the training matrix and/or test matrix.

You can open an existing neural model module with the Open command.

Shortcuts

Toolbar: 
Keys: CTRL+N

New Sparse Matrix command

Use this command to create a new data matrix for loading historical data into a sparse designed matrix.   Select the inputs and outputs for the design in the Create Sparse Matrix dialog box. The new sparse data matrix is created using the current data matrix (historical) module as a reference. The **Create Sparse Matrix** dialog allows you to select the inputs, outputs and enter the title of the new matrix.

The sparse data loader (SDL) is designed to reduce the number of observations needed to effectively train a neural model without loosing the information content of a historical dataset. In many cases the SDL will significantly reduce time necessary to train a model, while improving both the accuracy and long term stability of the predictive model. This is accomplished by storing the data in a form that is directed by a statistically designed experiment (DOX). When a designed experiment is performed careful attention is given to determining which variables are the suspected causes (independent variables or factors) and which are effects (dependent variables or responses). Also, the total amount of variability observed in the factors must be noted. This is used in calculating the center of the design, star points and vertices or factorial points.

Loading historical data using the SDL is not the same as doing a designed experiment. A DOX should be accomplished as quickly as possible, in a random order while paying careful attention to minimize the effects of other variables. The faster an experiment is completed the less chance that other unknown factors will have an effect.

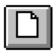
New command

File Menu

Use this command to create a new module.   Select the type of new file you want to create in the File New dialog box

You can open an existing module with the Open command.

Shortcuts

Toolbar: 

Keys: CTRL+N

File Open command

File Menu

Use this command to open an existing module in a new window. You can open multiple modules at once. Use the Window menu to switch among the multiple open modules. See Window 1, 2, ... command.

You can create new neural model modules with the New Neural Model command or new data matrix modules with the New Data Matrix command.

Shortcuts

Toolbar: 

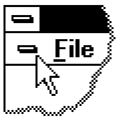
Keys: CTRL+O

Close command

File Menu

Use this command to close all windows containing the active module. NNMODEL suggests that you save changes to your module before you close it. If you close a module without saving, you lose all changes made since the last time you saved it. Before closing an untitled module, NNMODEL displays the Save As dialog box and suggests that you name and save the module.

You can also close a module by using the Close icon on the module's window, as shown below:

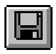


Save command

File Menu

Use this command to save the active module to its current name and directory. When you save a module for the first time, NNMODEL displays the Save As dialog box so you can name your module. If you want to change the name and directory of an existing module before you save it, choose the Save As command.

Shortcuts

Toolbar: 

Keys: CTRL+S

Save As command

File Menu

Use this command to save and name the active module.   NNMODEL displays the Save As dialog box so you can name your module.

To save a module with its existing name and directory, use the Save command.

1, 2, 3, 4 command

Use the numbers and filenames listed at the bottom of the File menu to open the last four modules you closed. Choose the number that corresponds with the module you want to open.

Import Data From Clipboard command

File Menu

Use this command to create a ***undesigned*** matrix from the contents of the clipboard. The contents of the clipboard must be in a spread sheet format (i.e. rows and columns with columns separated by tabs and rows separated by linefeeds or carriage returns).

After the selection is made the NNMODEL displays the import data dialog box so you can pre-view the conversion and make some limited editing. The scan routine looks at the first 12 records and trys to determine the separator character, record types and field types. When you are satisfied with the conversion parameters select the **Process** button to convert the data into a data matrix. The process step writes the clipboard data to an ASCII file (clipbrd.raw) then uses the same import procedures as does the import ASCII file command.

Import Data From ASCII File command

File Menu

Use this command to create a ***undesigned*** matrix from an external ASCII file. The contents of the ASCII file must be in a spread sheet format (i.e. rows and columns with columns separated by tabs, blanks or commas and rows separated by carriage returns).

After the selection is made the NNMODEL displays the import data dialog box so you can pre-view the conversion and make some limited editing. The scan routine looks at the first 12 records and trys to determine the separator character, record types and field types. When you are satisfied with the conversion parameters select the **Process** button to convert the data into a data matrix.

Import Neural Model command

File Menu

Use this command to create a new neural model module from the definitions in a ENN transport format file. This file format is used to transport a neural model between different hardware/software platforms, different versions of the NNMODEL program or to be used by an external program linked with the NNMODEL RunTime C NNLIB.

Import Test Matrix command

File Menu

Use this command to import a new test matrix into the neural model. It will delete the current test matrix (if it exists) and reload the test matrix from a selected data matrix. All fields that are present in the neural model must also be present in the selected data matrix.

Export Data Matrix command

File Menu

Use this command to export the current data matrix to an external ASCII file in the spread sheet format. This format is used to move the data to other analysis systems or transport between different versions of the NNMODEL.

Export Feed-Forward Neural Model command

File Menu

Use the command to export a (non-trainable) neural model in the ENN transport format. Only the feed-forward components are written to the file (i.e. no training or test matrices nor any of the dynamic parameters necessary to train a model. The model is frozen at the current level of training.

Export Full Neural Model command

Use this command to write the neural model to an ASCII ENN transport format file. This can later be loaded into a user written program using the NNMODEL Runtime NNLIB or a different hardware platforms.

Export Measured Vs Predicted command

File Menu

Use this command to write the current training matrix plus the predicted values as an tab delimited ASCII file. This file is intended to be used with other analysis programs. The test matrix can exported by checking the Use Test Matrix command in the Model menu.

Export Train/Test Matrix command

Use this command to write the training and test matrices to an external ASCII file in the spread sheet format.

Export Weight Matrix command

Use this command to write the neural models weight matrix to an external ASCII file in the following format.
<"title"><TAB><First Weight><TAB>...<TAB><Last Weight><CR/LF>

- Each field is separated by a TAB character.
- The title is enclosed in double quotes.
- All weights are printed in exponential notation (i.e. +1.23456E-002)
- The record ends with a carriage return and linefeed.

This format was intended to be used as an input into a second stage model. The number of weights can be calculated by:

I * H + H +O * H + O
if connect I to O enabled then add NO*NI
where I=inputs, H=hidden, O=outputs and connect I to O is a training option

Diagnostic Dump command

File Menu

Use this command to dump the contents of the CNeural, CParams and CDataMat structures in a format this is easily readable. This function is used primarily used for debugging internal problems. The filename of the diagnostic dump will be the title of the module with the file extension of DMP.

Print command

File Menu

Use this command to print a module. This command presents a Print dialog box, where you may specify the range of pages to be printed, the number of copies, the destination printer, and other printer setup options.

Shortcuts

Toolbar: 

Keys: CTRL+P

Print Preview command

File Menu

Use this command to display the active module as it would appear when printed. When you choose this command, the main window will be replaced with a print preview window in which one or two pages will be displayed in their printed format. The print preview toolbar offers you options to view either one or two pages at a time; move back and forth through the module; zoom in and out of pages; and initiate a print job.

Print Setup command

File Menu

Use this command to select a printer and a printer connection. This command presents a Print Setup dialog box, where you specify the printer and its connection.

Print Options command

File Menu

Use this command to select data matrix printing options. This command presents a Print Options where the selections can be made.

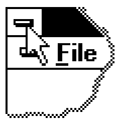
Exit command

File Menu

Use this command to end your NNMODEL session. You can also use the Close command on the application Control menu. NNMODEL prompts you to save modules with unsaved changes.

Shortcuts

Mouse: Double-click the application's Control menu button.



Keys: ALT+F4

Equations command

Use this command to add or edit the equations used to generate the calculated columns. Equations are entered in algebraic notation. Select equation parser for further information on the language and syntax

Exclusions command

Use this command to add or edit the equations used to exclude data from all numerical operations. such as graphs, statistics and building models. Select equation parser for further information on the language and syntax

Append Column command

Edit Menu

Use this command to append a new blank column to the end of the data matrix. All values in the new column are set to MISSING.

Append Calculated Column command

Edit Menu

Use this command to append a new calculated column to the end of the data matrix. The column name is selected from a list of formulas that have been entered in the equation dialog.

Variable Descriptors command

Edit Menu

Use this command to edit or view the descriptive information about the column variables. Each variable has a label, format, units, minimum, maximum, clip lo , clip hi, usage, scaletype, scale value, scale offset attributes. Only the first five can be altered.

Training Matrix command

Edit Menu

Use this command to edit both the training and test matrices in a neural model. Rows can be added or deleted or individual values changed.

Parameters command

Edit Menu

Use this command to edit or view the training parameters. These parameters control various aspects of the neural model training session. The typical types of parameters that can be changed are: # of hidden neurons, type of training, when to stop training, learning rates and momentum, tolerance, etc.

Delete Column command

Edit Menu

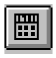
Use this command to remove a column and all it's data from the data matrix. First select the column(s) by placing the cursor in the column name (the selected column will be inverted) then select the delete column command.

Recalc command

Edit Menu

Use this command to recalculate the calculated column values, also calculated are the maximum, minimum and scaling factors.

Shortcuts

Toolbar: 

Copy command

Edit Menu

Use this command to copy selected data onto the clipboard. This command is unavailable if there is no data currently selected.

This copies the grid selection to the clipboard. The selection is converted to tab separated ASCII and placed in the clipboard as a table. The clipboard format used is CF_TEXT.

Copying data to the clipboard replaces the contents previously stored there.

Shortcuts

Toolbar: 

Keys: CTRL+C

Paste command

Edit Menu

Use this command to insert a copy of the clipboard contents at the insertion point. This command is unavailable if the clipboard is empty.

This copies the table in the clipboard (assuming CF_TEXT) into the data matrix. If necessary the data matrixs rows and columns will be extended.

Shortcuts

Toolbar: 

Keys: CTRL+V

Cut command

Edit Menu

Use this command to remove the currently selected data from the module and put it on the clipboard. This command is unavailable if there is no data currently selected.

Cutting data to the clipboard replaces the contents previously stored there.

Shortcuts

Toolbar: 
Keys: CTRL+X

Find command

Edit Menu

Use this command to search for values that meet the selected search criteria. This is useful you notice one or more points in a graph that are potential outliers. For example, if 99.9 % of the values in a column are between 0.500 and 0.800 but a few values are in the hundreds this will cause a significant error in the scaling factors. In order to remove or edit these points one must first find them. This can easily be accomplished by using the Find command. First place the cursor at the location to begin the search (column and row) then select the Find command from the Edit menu. A Find Equation dialog box will appear so that you can enter the search criteria (i.e. COLUMNNAME > 1.0). Press the OK button to begin the search. The Find command will stop at the first occurrence where the search criteria is satisfied or at the end of the data matrix (not found). The search can be continued by using the Find Next command.

Find Next command

Edit Menu

Use this command to continue the search started by the Find command. Find Next command will continue till the end of the data matrix is reached.

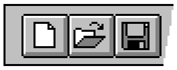
Toolbar command

View Menu

Use this command to display and hide the Toolbar, which includes buttons for some of the most common commands in NNMODEL, such as File Open.   A check mark appears next to the menu item when the Toolbar is displayed.

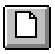
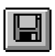
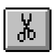
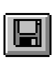
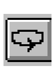
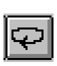
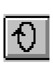
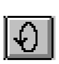
See Toolbar for help on using the toolbar.

Toolbar



The toolbar is displayed across the top of the application window, below the menu bar. The toolbar provides quick mouse access to many tools used in NNMODEL,

To hide or display the Toolbar, choose Toolbar from the View menu (ALT, V, T).

Click	To
	Open a new module.
	Open an existing module. NNMODEL displays the Open dialog box, in which you can locate and open the desired file.
	Save the active module or template with its current name. If you have not named the module, NNMODEL displays the Save As dialog box.
	Remove selected data from the module and stores it on the clipboard.
	Copy the selection to the clipboard.
	Insert the contents of the clipboard at the insertion point.
	Print the active module.
	Help.
	Context Help.
	Recalcs the active data matrix module.
	Rotates the 3D graph right.
	Rotates the 3D graph left.
	Rotates the 3D graph up.
	Rotates the 3D graph down.

Status Bar command

View Menu

Use this command to display and hide the Status Bar, which describes the action to be executed by the selected menu item or depressed toolbar button, and keyboard latch state. A check mark appears next to the menu item when the Status Bar is displayed.

See Status Bar for help on using the status bar.

Status Bar

The status bar is displayed at the bottom of the NNMODEL window.   To display or hide the status bar, use the Status Bar command in the View menu.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. This area similarly shows messages that describe the actions of toolbar buttons as you depress them, before releasing them.   If after viewing the description of the toolbar button command you wish not to execute the command, then release the mouse button while the pointer is off the toolbar button.

The right areas of the status bar indicate which of the following keys are latched down:

Indicator	**Description**
CAP	The Caps Lock key is latched down.
NUM	The Num Lock key is latched down.
SCRL	The Scroll Lock key is latched down.

Goto Record # command

View Menu

Use this command to go to a selected record number in the data matrix.

Goto End command

View Menu

Use this command to go to the end of the data matrix.

Initialize command

Model Menu

Use this command to initialize the model to the untrained state. If Automatic Increment training is enabled then the number of hidden neurons is set to 1 otherwise the number of hidden neurons is set to the value of Max Hidden. All weights are initialized to small random numbers.

Start Training command

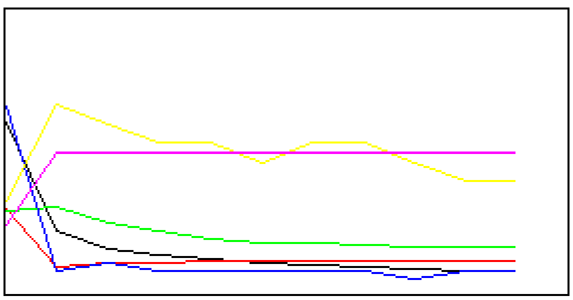
Model Menu

Use this command to start or stop the training of the neural model. If the graphing mode is enabled (default) then after ***eon*** training presentations a graph will be drawn. The graph will be updated every ***eon*** presentations until the model is trained. The update interval (***eon***) is set to 100 by default. This should not be changed because it affects the Automatic Increment training algorithm. Other Automatic Increment parameters such as hiddengrad, signinc and nosigninc might have to be tuned.

Standard Training Graph

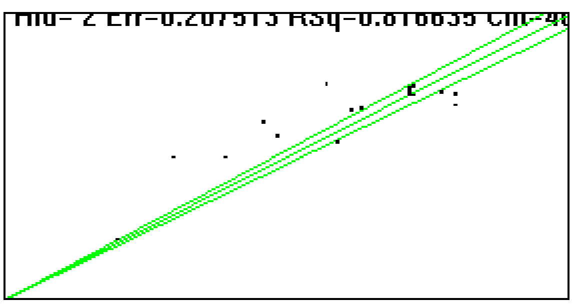
After each eon the training status graph is updated. The status graph displays six internal factors to demonstrate how well the training is proceeding. The following statistics are plotted.

Black	Sum squared error
Red	Largest negative residual
Green	Largest positive residual
Blue	Number of negative residuals > tolerance
Yellow	Number of positive residuals > tolerance
Magenta	Number of hidden units



MVP Training Graph

When the MVP Training Graph option is selected the status graph displays alternate statistics. Rather then the six internals listed above the MVP graph plots each observation on a measured verses predicted graph. Three green lines represent the zero, + tolerance and - tolerance bands. Only the output of the first modeled variable is plotted.



Use Test Matrix command

Model Menu

Use this command to switch the data source for graphs and reports from the training matrix to the test matrix. When this command is checked the test matrix is used.

Load Last Saved command

Model Menu

Use this command to load the last saved weight matrix automatically saved every autosave cycle. These values can be returned to the model in the event of a system crash.

Interrogate Model command

Use this command to interrogate a model. This command executes the interrogate model dialog box where you can enter input values and see the results of the model. Also, a sensitivity report can be created at the selected point in the data space.

Statistics Report command

Model Menu

Use this command to generate a statistic report that contains the basic statistics of the training matrix along with the model performance statistics.

Sensitivity Report command

Model Menu

Use this command to generate a sensitivity report showing the average sensitivity of the output variables to changes in the input variables. The sensitivity is based on the center of the data space.

MVP Train Graph command

Model Menu

Use this command to switch the training progress graph from the standard time series to a measured verses predicted graph.

No Train Graph command

Model Menu

Use this command to turn off the training status graph. If this command is checked then there will be no graphical display during training.

Use Ext Binary File command

Model Menu

Use the external binary file component of a very large data matrix. the current program cannot handle data matrices with more than 32k rows. The intent is to load only a small portion of a very large database (for graphing and testing) but allow the training to use the full database.

This is not fully implemented in this version of the program.

Basic Statistics command

Data Menu

Use this command to generate a statistical report containing simple statistics (number of observations, average, standard deviation, minimum and maximum values).

Correlation Analysis command

Data Menu

Use this command to generate a statistical report containing the basic statistics followed by a correlation matrix report.

Concatenate Data Matrices command

Data Menu

Use this command to combine two similar data matrices that result from the user doing a blocked experiment. The second DM will be appended to the first DM.   Fields that are not present in both data matrices will be filled with user selected values.

Merge Data Matrices by Time/Date command

Data Menu

Use this command to merge two data matrices by the time/date fields. Fields not present in both data matrices are set to missing.

Reserve Testing Data command

Data Menu

Use this command to randomly place a V in the RC-type field of the data matrix. Fields with this value in the RC field will be used to load the test matrix rather than the training matrix. This command displays the percentage of rows to be used as test rows dialog box. You can change the percentage at this point. When selecting OK the Vs will be inserted randomly. To remove all Vs from the data matrix enter a 0 in the field.

Fill Missing command

Data Menu

Use this command to fill in missing values with the value from the previous row. This is done on a column by column basis. First select the column by moving the cursor to the column and inverting the column, then select the fill missing command.

Check Sequential Data command

Data Menu

Use this command to scan the Time/Date fields to see if the timestamp is increasing as the row number increases.

Load Sparse command

Data Menu

Use this command to load a ***designed experiment*** matrix with data from another data matrix which may or may not be ***designed***. The load algorithm tries to load the data into the nearest matched design point. A metric is calculated to determine how close the data point is to the ideal design point. If there is already a row loaded at this point the metric is used to determine which row is closer to the ideal. The row that is closer to the ideal is used and the loser is either retained for testing purposes or discarded depending on user options. The load routine tries to load the matrix in the following order: center points, star points, factorial points, multilevel points and finally points are load to balance the data within the experimental space.

Load Rejects command

Data Menu

Use this command to load rejects into the current data matrix. This can only be done after a sparse load has been executed with the advisor option turned on. With the advisor option ***on*** the sparse loader will generate a missing data matrix module and display it after the load.

The advisor will make suggestions where the current data matrix is missing either center, star, factorial or multilevel points. Sometimes the suggested points cannot be run due to process restrictions or what ever reason. To avoid being asked by the advisor the next time sparse data loader is run you can put an R in the RC type field of the missing data matrix and load the missing (rejects) into the current DM. Next time the sparse loader is run the advisor will not suggest the rejected runs.

Randomize command

Data Menu

Use this command to randomize and scale the data in the selected column(s). Selecting this command will display the **Random Scales** dialog box. Change the **Multiplier** field to greater then 1 to increase and between 0 and 1 to decrease. Change the **Random** field to a number other then zero to increase the noise added to the number.

Best Model Search command

Data Menu

Use this command to search the data matrix for the best model for one or more selected output variable(s).

Options command

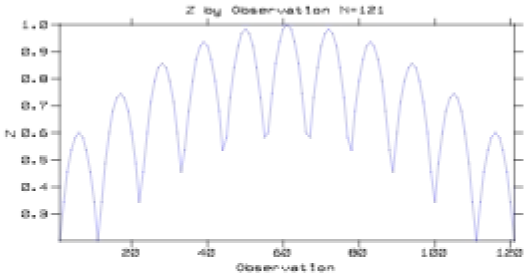
Graphs Menu

Use this command to change the basic characteristics of all graphs. Selecting this command will display the Graph Options dialog box were these selections can be made.

ByRow command

Graphs Menu

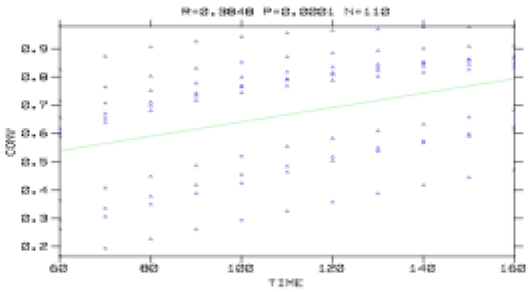
Use this command to generate a graph of the selected variable (Y axis) plotted against the row number (X axis). Any measured, input, output, predicted or residual variable can be plotted.



Scatter Plot command

Graphs Menu

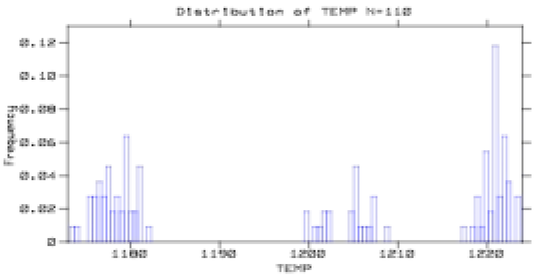
Use this command to generate an XY scatter plot of two selected variables. A symbol is placed for each XY pair. The points are not interconnected. If one or both points are missing then the pair is not plotted. Any measured, input, output or predicted variable can be plotted.



Distribution command

Graphs Menu

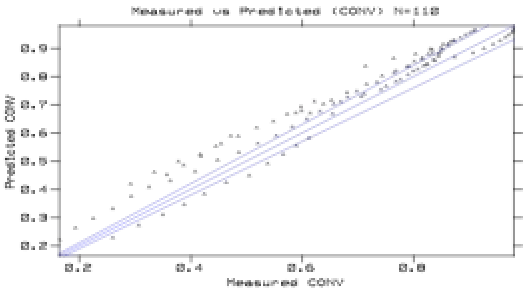
Use this command to generate a frequency distribution plot of a selected variable. The range of the variable is divided into N boxes (options). Each time a point (of M observations) falls into a box the value 1/M is added to that box. The total of all the boxes equals 1. Any measured, input, output, predicted or residual variable can be plotted.



Meas vs Pred command

Graphs Menu

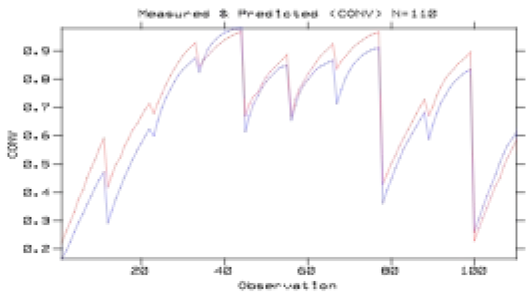
Use this command to generate a measured verses predicted graph. For every measured value a predicted value is calculated and a symbol is plotted on the graph. A line is drawn showing how a perfect model would plot. If a tolerance (Training parameters) is set to a number other than 0, then the + and - tolerance lines are also drawn. Any predicted variable can be plotted.



Meas & Pred command

Graphs Menu

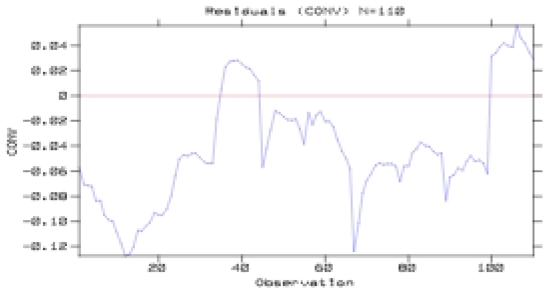
Use this command to generate a graph of the measured and predicted outputs of a model.   The selected output variable is plotted against the row number (X axis). The blue line represents the measured value of the selected variable and the red line represents the predicted value. Any predicted variable can be plotted.



Residuals command

Graphs Menu

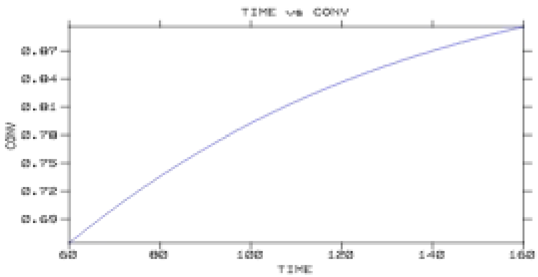
Use this command to generate a graph of the difference between measured and predicted outputs of a model. The residual calculation is plotted against the row number (X axis). The blue line represents the residuals and a red line is drawn at zero. Any predicted variable can be plotted.



XY command

Graphs Menu

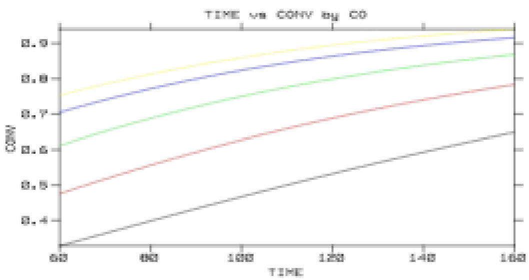
Use this command to generate a line graph showing how a model output (Y) variable changes in response to an input (X) variable. The X variable is varied from the minimum to the maximum observed value. Any combination of input and predicted variable can be plotted.



XY Effect command

Graphs Menu

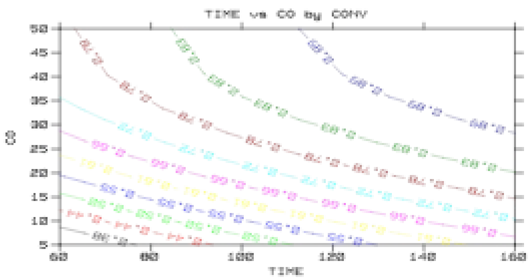
Use this command to generate a family of line graphs showing how a model output (Y) variable changes in response to an input (X) variable and an effect (E) variable. The X variable is varied from the minimum to the maximum observed value and the E-variable is varied from the minimum to maximum observed in steps of (# of curves) in the options dialog box. The lines are drawn in increasing steps of the E variable. The color order is black, red, green, blue, yellow, magenta, cyan, red, green, blue, yellow, magenta, cyan and black. Any combination of input and predicted variable can be plotted.



Contour command

Graphs Menu

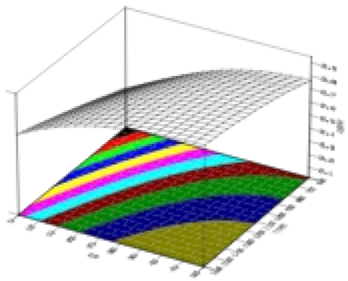
Use this command to generate a topological style graph showing how the neural output is effected by both an X and a Y (neural inputs) variables. The contours can either be drawn as a labeled contour lines or as a filled contour. The number of curves and type of contour is selectable in the contour dialog box. Any combination of input and predicted variable can be plotted.



3D Surface command

Graphs Menu

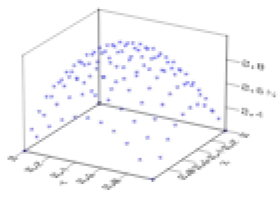
Use this command to generate a 3 dimensional surface plot of how a neural output is effected by 2 input variables (X and Y). The surface can be drawn as a black fish net or a colored fishnet or as a colored filled surface. Additionally a contour plot can be drawn under the surface. Addition options are selectable in the 3d selection dialog box. Any combination of input and predicted variable can be plotted.



3D Scatter command

Graphs Menu

Use this command to generate a 3 dimensional scatter plot. Any three measured variables can be plotted.



ByRow Matrix command

Graphs Menu

Use this command to generates a ByRow matrix plot. This command plots each variable against the row number with up to 32 variable plotted on a page.

Scatter Matrix command

Graphs Menu

Use this command to generate all combinations of XY scatter plots. The plots are drawn at postage stamp size at 32 per page. The number of plots generated is equal to the number of variables squared.

Distribution Matrix command

Graphs Menu

Use the command to generate a frequency distribution graphs of all variables. The plots are drawn 32 per page.

New Window command

Window Menu

Use this command to open a new window with the same contents as the active window. You can open multiple module windows to display different parts or views of a module at the same time. If you change the contents in one window, all other windows containing the same module reflect those changes. When you open a new window, it becomes the active window and is displayed on top of all other open windows.

Cascade command

Window Menu

Use this command to arrange multiple opened windows in an overlapped fashion.

Tile Horizontal command

Window Menu

Use this command to vertically arrange multiple opened windows in a non-overlapped fashion.

Tile Vertical command

Window Menu

Use this command to arrange multiple opened windows side by side.

Arrange Icons command

Window Menu

Use this command to arrange the icons for minimized windows at the bottom of the main window. If there is an open module window at the bottom of the main window, then some or all of the icons may not be visible because they will be underneath this module window.

1, 2, ... command

NNMODEL displays a list of currently open module windows at the bottom of the Window menu. A check mark appears in front of the module name of the active window. Choose a module from this list to make its window active.

Windows Help System

Help Menu

Use this command to evoke the Windows help system.

Index command

Help Menu

Use this command to display the opening screen of Help. From the opening screen, you can jump to step-by-step instructions for using NNMODEL and various types of reference information.

Once you open Help, you can click the Contents button whenever you want to return to the opening screen.

Using Help command

Help Menu

Use this command for instructions about using Help.

About command

Help Menu

Use this command to display the copyright notice and version number of your copy of NNMODEL.

Context Help command

Help Menu

Use the Context Help command to obtain help on some portion of NNMODEL. When you choose the Toolbar's Context Help button, the mouse pointer will change to an arrow and question mark. Then click somewhere in the NNMODEL window, such as another Toolbar button. The Help topic will be shown for the item you clicked.

Shortcut

Keys: SHIFT+F1

How to begin using NNMODEL

a step-by-step procedure

Designed Experiment or Unstructured Data

The Data Matrix Editor allows you to create a structured data matrix based on a statistically designed experiment or a totally unstructured matrix. A designed data matrix will allow you to squeeze the most information from a finite number of observations. Where as an unstructured matrix will allow you to enter or import historical data and perhaps extract some of the information buried deep within the ocean of noise.

Designed Data Matrix

In order to create a designed data matrix you must know which variables are the independent or factors and which are the dependent or responses. This imposes a type of input/output relationship onto the matrix. The types of statistical designs available are: two level, three level, simplex, star-simplex, central composite and multilevel. If the minimum and maximum levels for each variable are set prior to creation then the factor points will be initialized to the desired experimental values. Once created the design can be printing with the rows in random order to aid in minimizing experimental errors. After the experiment is completed the data can be entered using the Data Matrix Editor module.

Unstructured Data Matrix

An unstructured matrix can be created by either selecting the **New Data Matrix** command from the file menu and in the **Create Data Matrix** dialog box entering a design type of **No Design** plus the number of desired rows and columns. Then your data can be entered into the resulting data matrix. If the data already exists electronically then the matrix can created simply by importing it from the clipboard or a flat ASCII file.

Both Designed and Historical

The designed data matrix can be created from an unstructured matrix by first opening the unstructured matrix then creating an empty design shell by selecting the **New Sparse Data** option in the File menu. NNMODEL will allow you to design the matrix (picking the factors and responses and type of design) and then automatically load the new matrix from the unstructured data matrix. This loader will impose a designed matrix structure onto the unstructured historical data. However, this is not equivalent to doing a designed experiment because the "experiment" was not done in a random and timely manner. But, this method of using a structured method of selecting observations from a historical database is much better than either using all the available observations or randomly selecting a small sub-set of the data. Using all the data is impractical with very large databases due to excessive training times. Also, there is a good possibility that most of the data represents only the normal operating modes of the process causing the resulting model to lack evidence of the "big picture" and possibly develop high order internal representations of stochastic process perturbations. Using a small sample of the database, on the other hand, only addresses the problem of training time and could, in fact, cause the model to miss important evidence of causality.

Verify Data

The Data Matrix Editor contains many modules to aid you in verifying the validity of the data in the data matrix. The **Basic Statistics** command in the Data menu will generate a report containing the number of observations, maximum, minimum, mean, standard deviation and sum of the squares for each of the columns of data. The **Correlation Analysis** command in the Data menu will generate a report containing the aforementioned basic statistics plus the Pearson correlation statistic and probability comparing each variable against all others. The data can be viewed graphically by observation, as a distribution plot or as 2D and 3D scatter plots. Additionally, the by observation, scatter and distribution plots can be printed 32 to a page using the matrix plot commands in the data menu.

Reserve Testing Data

When using historical data there is a need for testing data to help validate the models. Testing data is not needed for models built from a designed experiment, but, none the less can be of some use.

Designed Data Matrix

If additional data (such as replicates) were collected for testing purposes when running a designed experiment it should be entered into a separate unstructured data matrix and imported later during model testing.

Unstructured Data Matrix

With unstructured historical data, however, some portion of the data matrix should be reserved for testing before model building using the command **Reserve Testing Data** in the Data menu. Additional test data can also be imported during the model testing phase.

Create Neural Model(s)

To build a model from a data matrix select **New Neural Model** command from the File menu. If there is not a data matrix already open then an **Open File** dialog will appear. Enter the name of the data matrix to use as the source of the training and/or test matrix. Select OK to open a data matrix and the **Create Neural Model** dialog box will appear. Select the title, inputs and outputs for the desired model and press OK and an untrained neural model will be created.

Before training the model you can modify the default training parameters by selecting the **Parameters** command from the Edit menu. This will invoke the **Edit Training Parameters** dialog box. You can modify various parameters, such as, number of hidden neurons, maximum presentations of the training matrix, type of training used, and termination parameters. When satisfied push the OK button to save the changes. The training of the model can now be started by selecting the **Start Training** command from the Model menu.

Analyze Neural Model(s)

After the model has been trained you can analyze its performance using either the training matrix or test matrix as the source dataset for the analytical tools. Basic performance statistics can be calculated by selecting the **Statistics Report** from the Model menu. This calculates the residuals (difference between the measured and predicted values) for each set of input/output values in the selected matrix. Residuals can also be displayed graphically to enable you to visually determine how well the model is performing. Both statistical and graphical analysis can be performed on the test matrix by toggling the **Use Test Matrix** command in the Model menu. Additional test matrices can be loaded and used by selecting the **Import Text Matrix** command in the File menu.

Use

After a model has been developed it can be used within the NNMODEL application either graphically or interactively. Graphically the response surface can be plotted as a three dimensional fishnet type graph, as a contour plot or as a family of curves on a two dimensional plot. Interactively, numbers can be entered into the model to see the resulting numerical result. A sensitivity analysis can be executed to explore the relative importance of each input variable.

The model can also be used externally by exporting it as either a trainable or non-trainable ENN file. These files can be read by the NNMODELs external C subroutines enabling you to immediately combine neural models with your own code.

How to create an undesigned data matrix.

An undesigned data matrix can be created by importing data from the Windows clipboard or an external ASCII file.

To create an empty data matrix for entering data manually:

1. Use the New Data Matrix command in the File menu
2. Select **No Design** for the design type (Default)
3. Enter the number of desired columns in the **# of columns** field
4. Enter the number of desired rows in the **# of rows** field
5. Re-name the automatic generated names using the Max/Min grid.
6. Enable the **Don't Create Data** option so that no data will be created (Default: Disabled).
7. Press OK to create the data matrix.

How to create a designed data matrix

A designed data matrix can only be created by using the New Data Matrix command in the File menu. Before executing this command you should give some thought to the design of the experiment (i.e. factors, responses, center point replicates, type of design and the maximum and minimum points that will be visited by each factor). A designed data matrix can be created with or without the data matrix filled. The factors are set to the design levels (as calculated) and the responses are set to missing. After the experiment is run all the responses will have to entered. The factors, however, need not be changed unless the factor levels are also measured. An empty designed data matrix can be filled using the Sparse Data Loader routine in the Data menu, but, the loader currently supports only one center point.

To create a designed data matrix for entering data manually:

1. Use the New Data Matrix command in the File menu.
2. Select the type of statistical design in the design type list box (Default: No Design).
3. Enter the number of factors(Default: 1).
4. Enter the number of responses(Default: 1).
5. Decide on the # of center point replicates to be run and enter (Default: 0).
6. Re-name the automatic generated names using the Max/Min grid.
7. Set the maximum and minimum levels for each factor (Default: 0/1).
8. If the design type is multilevel then set the # of levels for each factor (Default: 2).
9. If the design type is central composite then select type of scaling used to determine the star and factor levels (Default: Standard Composite).
10. If you are using the **User Scaled** option then set the scale factor (Default: 0.5).
11. Disable the **Don't Create Data** option so the factor columns can be initialized in the data matrix (Default: Disabled).
12. Press OK to create the designed data matrix.

    At this point you have created a designed data matrix with all the factors entered an calculated along with all the responses filled with missing values. You can now print the data matrix (standard or randomly) and run your experiments, collect the data and start entering.

To create a designed data matrix for loading the data automatically (Sparse Data Loader):

1. Use the New Data Matrix command in the File menu.
2. Select the type of statistical design in the design type list box (Default: No Design).
3. Enter the number of factors(Default: 1).
4. Enter the number of responses(Default: 1).
5. Re-name the automatic generated names using the Max/Min grid.
6. Set the maximum and minimum levels for each factor (Default: 0/1).
7. If the design type is multilevel then set the # of levels for each factor (Default: 2).
8. If the design type is **Central Composite** then select type of scaling used to determine the star and factor levels (Default: Standard Composite).
9. If you are using the **User Scaled** option then set the scale factor (Default: 0.5).
10. Enable the **Don't Create Data** option so that no data will be created (Default: Disabled).
11. Press OK to create the designed data matrix.

    At this point you have created an empty designed data matrix. The matrix is ready to be loaded using the Sparse Data Loader command in the Data menu.

How to import a data matrix from an ASCII file

Data in an external ASCII file can be imported into an undesigned data matrix by following the procedure listed below. A data matrix can also be imported from the Windows clipboard.

To import data from an ASCII file:

1. Create an ASCII file using the product of your choice.
2. Select Import / Data From ASCII File command in the File menu.
3. The Data Import dialog box will appear to allow you to pre-view the conversion. If the first 12 rows and columns don't look correct review the ASCIIfile format and re-generate the file and repeat step 2.
4. Select the **Process** button to convert the file into an undesigned data matrix.

   The Data Matrix Editor will open and display the newly created data matrix allowing you to edit or analyze the data.

How to import a data matrix from a spread sheet program

Data from a Windows based spread sheet program can be imported into an undesigned data matrix by first selecting the rows and columns in the spread sheet and coping them to the clipboard and following the procedure listed below. An external ASCII file can also be imported using the Import / Data From ASCII File command.

To import data from the clipboard:

1. Copy the desired rows and columns into the clipboard.
2. Select the Import / Data From Clipboard command in the File menu.
3. The Data Import dialog box will appear to allow you to pre-view the conversion.
4. Select the **Process** button to convert the file into an undesigned data matrix.

   The Data Matrix Editor will open and display the newly created data matrix allowing you to edit or analyze the data.

How to load historical data into a designed data matrix

To load historical data into a designed data matrix:

1. Load the historical data into an undesigned data matrix using one of the import data commands in the File menu.
2. Select **New Sparse Matrix** command in the File menu. A dialog box will appear that will allow you to choose the factors and responses from the original historical data matrix.
3. Enter the name of the new matrix in the title field.
4. Select the factors from the list of variables and press the **Add In** button.
5. Select the responses from the list of variables and press the **Add Out** button.
6. Select the **OK** button. The **Create Design** dialog box will appear.
7. Select the type of design matrix you wish to create and press **OK**. The Sparse Data Loader Options dialog box will appear.
8. Select the desired options and press **OK**. The important options that can affect the number of rows loaded are listed below.
9. The sparse loader will execute and attempt to load records from the historical data matrix into the new sparse matrix. After loading, a dialog box will be displayed showing you how many of the historical rows met the design criteria and were loaded. Press **OK** and the new data matrix will be displayed.

Option	Description
Auto Rescale	The auto rescale option uses the maximum and minimum levels from the historical data matrix. The maximum and minimum values listed in the variable descriptor of the designed data matrix are very important because they are used in calculating the design points (star, center and vertices). If the values are incompatible (between the current DM and the historical DM) the loader wont be able to correctly identify the design points.
Keep Rejects for Testing	All observations that are rejected for the designed data matrix are saved and appended to end with a record type of V. A V in the RT field indicates test data.

How to combine two similar data matrices

Many times it is necessary to combine two (or more) experiments together to generate a bigger picture model.

For example, if you wanted to combine the two matrices listed below. Notice that data matrix A has three columns (Time, H2% and Cnv) and data matrix B has three columns (Time, CO% and Cnv). To concatenate matrix B into A, one must first create a new column (CO%) then append B to the end of A. Note, this means that part of columns H2% and CO% are unknown since CO% wasnt listed in matrix A and H2% wasnt listed in matrix B. During the concatenate process dialog boxes will appear so that you can initialize these unknown sections to a constant value or leave it as ***missing***.

Data Matrix A

Time	H2%	Cnv
1	10	.2
2	20	.4
3	30	.5

Data Matrix B

Time	CO%	Cnv
1	10	.3
2	20	.5
3	30	.8

Concatenated Matrix

Time	H2%	Cnv	CO%
1	10	.2	
2	20	.4	
3	30	.5	
1		.3	10
2		.5	20
3		.8	30

The procedure for combining two data matrices is as follows:

1. Open the first data matrix by selecting the Open command in the File menu.
2. If you wish this first data matrix to remain untouched, select the Save As command from the File menu and rename it. If this is not done, the next matrix will be concatenated into the first one.
3. Select Concatenate Data Matrices from the Data menu.
4. The File Open dialog box will appear. Navigate to the desired sub-directory and select the second data matrix and press **OK**.
5. The procedure would be complete at this point if both matrices had exactly the same number of fields with identical names. If not, then you will be asked a series of questions on how to fill in the missing fields. Read the questions carefully and enter the value to be used as the filler for each section.

   When all the new columns have been created and the values filled the concatenated data matrix will appear. Scan the data to see how the unknown values have been filled.

How to use calculated columns

The calculated column feature allows you to create a new column of data that is defined by a simple algebraic equation. This can be used for user defined scaling, adding first principles equations to a neural model or creating arithmetic combinations of input variables. The procedure for adding a calculated column is as follows:

1. Select the Equations command from the edit menu.
2. Enter one or more equations to be used as columnar data.
3. Press **OK** to save the equations.
4. Select the Append Calculated Column command from the Edit menu.
5. Select the equation to append as a column.

The new column will appear as the right-most column in the data matrix. Any changes made to the equations will cause the column to be re-calculated.

caveats:

- If you delete the column equation all values in the calculated column will be set to missing.
- If you are using other columnar data in the equation there are two things to be aware of 1) if you change a cell in the row the calculated column will not automatically update, 2) dont use recurrent equations there is only one pass through the equations per row.
- The data in calculated columns that use the NORMAL or RANDOM functions will be re-generated new each time the re-calc command is executed.

How to use the time or date fields

The current version of NNMODEL provides column variable types of DATE and TIME to allow you to time merge two (or more) data matrices. The type of matrices that can be time-merged fall into three basic categories:

1 Matrices containing only TIME fields. This matrix can only contain data with time values between 00:00:00 and 23:59:59 (no more then one day). Time must be in increasing order. Format is HH:MM:SS i.e. 04:24:40 is 4 hours, 24 minutes and 40 seconds. Leading zeros are required to maintain digit placements.
2 Matrices containing only DATE fields. This matrix can contain data rows where each row represents a different day. DATE values can be from 00000000 to 99999999. Formatted as YYYYMMDD i.e. 19941012 == October 12, 1994. Only one row per day is allow. Dates must be in increasing order. Leading zeros are not required.
3 Matrices containing both TIME and DATE fields. These two fields will be combined into a 32-bit long integer representing a TIMESTAMP in seconds. The first rows DATE field is used as an offset (year-month-day). Multiple rows with the identical TIMESTAMP are not allowed in the combined matrix.

Rules for time merging two data matrices:

1 if DM1 has a TIME field then DM2 must have a TIME field.
2 if DM1 has a DATE field then DM2 must have a DATA field.
3 If DM1 has both TIME and DATE fields then DM2 must have both TIME and DATE fields.
4 Any data fields that are in DM1 (by name) cannot be in DM2 (and vice versa).

Use the following procedure to time/date merge two data matrices:

1 Open the first data matrix, select Check Sequential Date from the data matrix. If the date is not sequential fix before continuing to step 2.
2 Open the second data matrix, and check the to see if the date is sequential. Repair if needed.
3 Close the second data matrix and bring to the top the first data matrix window.
4 Select Merge Data Mats by Time/Date from the data menu. An Open File dialog box will appear. Select the filename of the second data matrix.

If there were no problems with the time and/or date fields and there were no other errors (duplicated field names) the second matrix will now be combined into the first.

How to find errors in a data matrix

The following tools are available for scanning the data matrix for obvious errors:

By Row Graph	This a simple graph of the selected variable plotted against the observation (row) number. Many obvious errors can be found by viewing this graph. See Common Data for examples.
By Row Matrix Graphs	This graph is a convenient way of generating a By Obs graphs of all the variables in the data matrix. This graph was designed to be printed and it responds rather slowly to interactive commands.
Distribution Graph	This a frequency distribution graph of the selected variable plotted against the range of observed values. This graph can show you sub-populations, effects of pinning or pocketing, normal or skewed distributions.
Distribution Matrix Graphs	This graph is a convenient way of generating the Distribution graphs of all the variables in the data matrix. This graph was also designed to be printed.
Basic Statistics Report	Use this report to validate the total range of all the variables (maximum and minimum). Verify the realistic range for each variable. The standard deviation can be helpful when comparing the variability with historical variability.

The following procedure can be used to scan the data for erroneous entries:

1 Select the By Row Matrix command from the Graphs menu. Visually inspect each individual graph for obvious outliers. A forgotten decimal point can cause a plot to look like a totally flat simply because one point caused the range to be very large. Sudden rises or drops in level, too digital looking (pocketing), never changing, pinning, gaps in data, long term shifts in levels, repeating patterns could all indicate possible problems.

2 Select the Distribution Matrix command from the Graphs menu. Visually inspect the distributions. A very tight distribution could indicate data errors. Multi-modal distributions can indicate sub-populations of data.

3 Select the Basic Statistics report from the data menu. Verify the minimum and maximum point for each variable is valid. A large standard deviation could indicate a problem.

How to copy data, graphs and reports

into other Windows   products

Data, graphs and reports can be transferred to any Windows based word processor, spread sheet or similar program by using the clipboard feature. Additionally, data can be transferred from the data matrix by exporting it to an ASCII file. A report, when saved to disk, is an ASCII file.

DATA

Copy a rectangular section to the clipboard

1 Position the cursor to the upper left hand corner of the rectangle.
2 Hold down the shift key and expand the selection until the entire section to be copied is inverted.
3 Select the Copy command from the Edit menu.
4 Use Alt-Tab to switch to the Windows program that will receive the graph.
5 Position the other programs cursor to the place were you wish the graph to be inserted.
6 Select the paste command from the other programs Edit menu.

Copy one or more entire columns to the clipboard

1 Position the cursor to the label of the first column to be copied.
2 Hold down the shift key and expand the selection until all columns to be copied are inverted.
3 Select the Copy command from the Edit menu.
4 Use Alt-Tab to switch to the Windows program that will receive the graph.
5 Position the other programs cursor to the place were you wish the graph to be inserted.
6 Select the paste command from the other programs Edit menu.

Copy one or more entire rows to the clipboard

1 Position the cursor to the label of the first row to be coped.
2 Hold down the shift key and expand the selection until all rows to be copied are inverted.
3 Select the Copy command from the Edit menu.
4 Use Alt-Tab to switch to the Windows program that will receive the graph.
5 Position the other programs cursor to the place were you wish the graph to be inserted.
6 Select the paste command from the other programs Edit menu.

Copy the entire data matrix to the clipboard

1 Position the cursor to the empty label cell above the first row and to the left of the RC field label.
2 Select the Copy command from the Edit menu.
4 Use Alt-Tab to switch to the Windows program that will receive the graph.
5 Position the other programs cursor to the place were you wish the graph to be inserted.
6 Select the paste command from the other programs Edit menu.

Copy the entire data matrix to an ASCII file

1 Select Export Data Matrix command from the File menu.
2 A Export File dialog box will appear. Select the sub-directory and filename for the ASCII file and press OK.
3 The file will be written as a TAB separated ASCII file suitable for importing into many non-windows based products.

GRAPHS

Copy the graph as a Windows metafile to the clipboard

1 Create the graph by using the command in either the Data Matrix Editor or the NNMODEL.
2 Make the graph the active window
3 Select the copy to clipboard command from the Edit menu.
4 Use Alt-Tab to switch to the Windows program that will receive the graph.
5 Position the other programs cursor to the place were you wish the graph to be inserted.
6 Select the paste command from the other programs Edit menu.

REPORTS

Copy the report as ASCII text to the clipboard

1 Create the report by using the appropriate command in either the Data Matrix Editor or the NNMOEL.
2. Make the report the active window.
3 Position the cursor to the first record to be transferred, hold down the shift key and move the selected region to the last desired record.
4 Select Copy command from the Edit menu.
5 Alt-Tab to the other application.
6 Position the cursor in the other application to where you want to insert the report.
7 Select Paste from the other programs Edit menu.

Save the report as ASCII text

1 Create the report by using the appropriate command in either the Data Matrix Editor or the NNMODEL.
2. Make the report the active window.
3 Select the Save As command from the File menu.
4 A Save As dialog box will appear. Select the sub-directory and filename for the report and press OK
5 The file will be written as a TAB separated ASCII file suitable for importing into many non-windows based products.

How to create a neural model

Assuming a data matrix has already been created

The procedure for creating a new neural model is as follows:

1 Select the Open command from the File menu.
2 An Open File dialog box will appear. Select the File Type as Data *.DM and navigate to the appropriate sub-directory. Enter the filename of the data matrix that will serve as the source for the training and optional test matrix.
3 Select the New Neural Model command from the File menu. A Create Model dialog box will appear.
4 Select from the input list all the variables to serve as inputs to the model and press **Add In**.
5 Select from the output list all the variables to be output from the model and press **Add Out**.
6 If you wish to restrict the data that will be used as training or test matrices enter the exclusion equations in the exclusions edit box.
7 Press **OK** to create the model. When the model is created the NNMODEL window will appear.
8 Select Initialize from the Model menu.
9 Select Start Training from the Model menu and wait for the beep indicating the training has been completed.
10 The model is now trained (to some extent). To save it, select the Save As command from the file menu. A File dialog will appear.
11 Select an appropriate filename and press **OK**.

By following this procedure you have created a neural model using the default parameters. The defaults work best on models that are basically linear in nature. However, this doesnt mean uncomplicated, many complex systems are built from linear combinations. Careful analysis of both the input data and the resulting model should be done before exploring other methods in the training of the model.

How to analyze a neural model

The NNMODEL application provides two basic methods for analyzing trained models. The first method is a numerical or statistical method and the second is a graphical method. Both of these methods can be used on either the training matrix or the test matrix. Also, additional test matrices can easily be imported for analyzing long term performance.

To switch between the training and test matrices select the **Use Test Matrix** command from the Model menu. When this option is checked the test matrix will be used as the source for all statistical and graphical methods. Additional test matrices can be loaded using the **Import Test Matrix** command from the File menu.

The statistical method calculates some basic statistics on the residual values. The residual is the difference between the measured output and the models prediction. Good models should have a residual mean near zero with a tight normal distribution, hence, the standard deviation, minimum, maximum and sum squares values should also approach zero. The R square statistic can be used to indicate: ***overall*** how good is the model. As the models performance increases the R2 statistic will approach one.

Statistic	Description
Residual Mean	The residual is the difference between the measured and predicted values. The closer the mean is to zero the better the model is predicting.
Residual Std Dev	The better then model predicts the closer the standard deviation will be to zero.
Residual Minimum	This is the worst case under estimated prediction. The closer this value is to zero the better.
Residual Maximum	This is the worst case over estimated prediction. The closer this value is to zero the better.
Residual Sum Sq	This value is the total sum squares of all the residuals. The lower this number the better.
R Square	The closer this number is to 1.0 the better.

Graphical methods of analyzing a models performance are:

Graph	Description
Measured vs Predicted	This graph plots for each pair of measured (X axis) and predicted (Y axis) values. The points from an ideal model would form a diagonal line with an angle of 45 degrees.
Measured & Predicted	This graph simply overlays the measured values on top of the predicted values. The measured values are plotted in blue and the predicted in red. The better the model predicts the harder it is to discriminate between the measured and predicted plots
Residuals by Observation	This graph plots the difference between the measured and predicted values by observation. A good model should yield a flat, near zero plot.
Distribution of Residuals	The distribution of the residuals can be plotted to see if they are normally distributed.

The model can be verified using data that it was not trained on by either using the data that was reserved for testing in the parent data matrix or by importing new test data from another data matrix.

To reserve a portion of the data matrix for testing select **Reserve Testing Data** from the data menu of the parent data matrix, then create the neural model(s).

To import a new testing matrix select **Import Test Matrix** from the file menu of the neural model and choose an existing data matrix file.

How to use a neural model in an external C program

A very simple test program has been provided (found in the NNLIB sub-directory) to demonstrate how to use an exported neural model in a C program. Initially the program has been written to load and execute the XOR model. To execute this program; first change directory to the NNLIB sub-directory then execute ITEST. First the program will prompt you to enter a model file name. Enter XOR and press <RETURN>. Then the program will prompt you to enter two variables (IN1 and IN2) and then display the result (OUT). This will continue in a loop until an **END** is entered in response to an input request.

The following procedure is an example of how to build an external C program that executes a feed-forward neural model:

1. Export the trained neural model using either the Export Feed-Forward Neural Model command or the Export Full Neural Model command in the File menu. Copy the exported neural model file to the sub-directory NNLIB.
2. Compile and link the example program using the DOS batch file (MKITEST.BAT). This batch file assumes you have Microsoft C and the environmental variables LIB and INCLUDE are appropriately set.
3. Execute the test program (ITEST.EXE). The ITEST program will prompt you to enter values for all inputs and then will calculate and print the outputs. To exit the program type **END** in response to an input request.

ENN File Format

All records begin with an identifier field of 3 characters. The first character identifies the parent module for the data (i.e. P=Parameters, N=Neural and D=DataMat). The next two numbers identify the record number. The record number identifies the variables within the module. The following three tables show the internal variables printed in each record. All floating point variables are printed in exponential format. When a vector is printed the C code ***for*** loop is given for clarity. When a matrix is printed the two ***for*** loops are given. Each field within the record is separated by a space character.

Parameters Module

P01	EXPORTVERSION
P02	m_TrainFlags, m_AImaxhid, m_goodness, m_autosave, m_seed, m_eon
P03	m_cnt_max, m_hiddegrad, m_errtol, m_goodrsq, m_signinc, m_nosigninc, m_alpha
P04	m_theta, m_randz, m_inrandzdiv, m_tol, m_learning_rate, m_Hlearning_rate, m_tlearning_rate, m_inoutlearn

Neural Module

N01	m_istate, m_ninputs, m_nhidden, m_noutputs, m_cnt
N02	for (i=0;i<m_ninputs;i++)
	for (j=0;j<m_nhidden;j++) m_hinputw[j][i]
N03	for (i=0;i<m_nhidden;i++) m_htheta[i]
N04	for (i=0;i<m_noutputs;i++)
	for (j=0;j<m_nhidden;j++) m_oinputw[i][j]
N05	for (i=0;i<m_noutputs;i++) m_otheta[i]
N12	for (i=0;i<m_noutputs;i++)
	for (j=0;j<m_ninputs;j++) m_iinputw[i][j]
N06	for (i=0;i<m_nhidden;i++) m_hlastvar[i]
N07	for (i=0;i<m_nhidden;i++) m_hlearn[i]
N08	for (i=0;i<m_nhidden;i++) m_htlearn[i]
N09	for (i=0;i<m_noutputs;i++) m_olastvar[i]
N10	for (i=0;i<m_noutputs;i++) m_olearn[i]
N11	for (i=0;i<m_noutputs;i++) m_otlearn[i]
N99	

Data Matrix Module

D01	m_istate, m_numcols, m_numrows, m_ninputs, m_noutputs, m_rawrows, m_rawcols, m_total
D02	m_title, m_desc, m_rawfname, m_parfname, m_creation
D03	for (i=0;i<m_numcols;i++)
	i, m_coldesc[i].flag, m_coldesc[i].fieldtype, m_coldesc[i].fscale, m_coldesc[i].foffset, m_coldesc[i].max, m_coldesc[i].min, m_coldesc[i].col_usage, m_coldesc[i].format, m_coldesc[i].vlab
D04	for (i=0;i<m_numcols;i++) m_icrossref[i]

D05	for (i=0;i<m_numcols;i++) m_ocrossref[i]
D06	for (i=0;i<m_ninputs;i++)
	i, m_icoldesc[i].flag, m_icoldesc[i].fieldtype, m_icoldesc[i].fscale, m_icoldesc[i].foffset, m_icoldesc[i].max, m_icoldesc[i].min, m_icoldesc[i].col_usage, m_icoldesc[i].format, m_icoldesc[i].vlab
D07	for (i=0;i<m_noutputs;i++)
	i, m_ocoldesc[i].flag, m_ocoldesc[i].fieldtype, m_ocoldesc[i].fscale, m_ocoldesc[i].foffset, m_ocoldesc[i].max, m_ocoldesc[i].min, m_ocoldesc[i].col_usage, m_ocoldesc[i].format, m_ocoldesc[i].vlab
M%04d	for (j=0;j<m_numrows;j++) j+1,
	for(i=0;i<m_ninputs;i++) m_iarray[i][j] for(i=0;i<m_noutputs;i++) m_oarray[i][j]
D08	m_numtests
T%04d	for (j=0;j<m_numtests;j++) j+1,
	for(i=0;i<m_ninputs;i++) m_itarray[i][j] for(i=0;i<m_noutputs;i++) m_otarray[i][j]

The following is an export of a full neural model in ENN format. The models title is LOGIC it has 2 inputs (IN1 and IN2), 1 output (XOR) and 1 hidden layer neuron. The training matrix contains 4 rows and the test matrix contains 1 row.

P00	4
P01	0 4 3 1000 15 100
P02	1000 0.750000 0.001000 0.900000 0.050000 0.005000 0.800000
P03	0.500000 0.500000 0.000000 0.050000 0.750000 1.500000 0.750000 0.100000
N01	7 2 1 1 0
N02	4.973449e-001 3.442030e-001
N03	-3.257088e-001
N04	4.103366e-001
N05	-4.918821e-001
N12	3.424634e-001 -1.800745e-001
N06	0.000000e+000
N07	1.500000e+000
N08	7.500000e-001
N09	0.000000e+000
N10	7.500000e-001
N11	7.500000e-001
N99	
D01	35031 3 4 2 1 5 5 12
D02	LOGIC Unknown C:\OOMODEL\TESTSETS\LOGIC.RAW LOGIC 791319606
D03	0 0 0 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 N %s ?

D03	1 0 0 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 N %s ?
D03	2 0 0 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 N %s ?
D04	0 1 -1
D05	4 -1 -1
D06	0 0 0 6.000000e-001 3.333330e-001 1.000000e+000 0.000000e+000 I %3.1f IN1
D06	1 0 0 6.000000e-001 3.333330e-001 1.000000e+000 0.000000e+000 I %3.1f IN2
D07	0 0 0 6.000000e-001 3.333330e-001 1.000000e+000 0.000000e+000 O %3.1f XOR
M0001	0.000000e+000 0.000000e+000 0.000000e+000
M0002	0.000000e+000 1.000000e+000 1.000000e+000
M0003	1.000000e+000 0.000000e+000 1.000000e+000
M0004	1.000000e+000 1.000000e+000 0.000000e+000
D08	1
T0001	5.000000e-001 5.000000e-001 5.000000e-001

Spread Sheet Format File Definition

The first character in each record specifies the record type. If a separator character is the first character of the record then the record is assumed to be a training record. The first character can be set to T for training record or V for the validation record. The default of the field is a T. In most cases the first two characters in the file should be blank. Each field should be separated by a tab. If a label record is present it must be before the first data record and the first character in the record should be an L followed by a separator. Units can be set by including a unit record (U). All records should end with a LINEFEED. The total number of data fields in the file must be consistent through out the file. Missing fields must be delimited by a tab character.

The first character of each row may contain any one of the following characters:

Character	Record Type
'T'	Training record
TAB	Training record
'V'	Verification & test record
'L'	Field label record
'U'	Units label record

ASCII File Import Types

The first character in each record specifies the record type. If a separator character is the first character of the record then the record is assumed to be a training record. The first character can be set to T for training record or V for the validation record. The default of the field is a T. In most cases the first two characters in the file should be blank. Each field should be separated by either a blank, tab or comma. If a label record is present it must be before the first data record and the first character in the record should be an L followed by a separator. Units can be set by including a unit record (U). If the separator is blank then one or more blanks can be used between fields to align the data (Example 4.) All records should end with a CARRIAGE-RETURN followed by a LINEFEED. The total number of data fields in the file must be consistent through out the file. Missing fields must be can only be delimited by a PERIOD in blank separated fields file or by a missing field in a TAB or COMMA separated file (Example 6.)

The first character of each record may contain any one of the following characters:

Character	Record Type
'T'	Training record
SEPARATOR	Training record
'C'	Training record Center point
'S'	Training record Star point
'F'	Training record Factorial point
'M'	Training record MultiLevel point
'X'	Training record Simplex point
'V'	Verification & test record
'D'	Deleted (dont use for training, use for scaling)
'R'	Rejected (don't suggest [DM advisor], use in scaling)
'E'	Excluded (dont use at all)
'*'	Comment record
'L'	Field label record
'U'	Units label record

NOTE: A SEPARATOR may be a BLANK, TAB or COMMA character. The separator used could be consistent in the file. Dont start using a TAB as a separator then switch to commas.

Supported field types that be automatically determined during file import:

FLOAT	Standard or exponential notation (i.e. 1.23 or +1.23E+00
TIME	HH:MM or HH:MM:SS (i.e. 22:59 or 22:59:59)
DATE	YYYYMMDD (i.e. 19840129)

Time and Date formats can be read but not automatically determined (requires manually setting the field type during import):

TIME	HHMM or HHMMSS
DATE	any one to eight digit integer number

Note: Dates and times can be loaded into the data matrix, however, they are only used for merging files by time and date (no graphing or analysis). If you want to use time or date as an input to a model or for graphical analysis then format the fields as integers or float point numbers (no colons).

Equation Parser Language

The equation parser is used for two purposes: 1) to define a variable that is was not in the original source of the data matrix (calculated column) and to exclude data during reporting, graphing or loaded a training or test matrix during neural model creation.

The equation parser is a very simple language. Basically the parser evaluates assignments in algebraic format. An assignment is simply a variable set equal to an expression. The assignment ends with a carriage return.

An expression can be composed of numbers, variables, assignments, built in functions, arithmetic operators or other expressions.

A variable can be one of the pre-assigned variables (column names assigned in the data matrix or the current row number) or a user defined variable. If the variable is user defined then the name can be from 1 to 32 characters long. The name must start with a alphabetic character and the remainder can be any combination of alphabetic letters (A-Z) or numbers (0-9).

Examples
X = 23
X = Y
X = ROW / 2
X = Y = 2
X = LOG(Y)
X = Y * 3 + Y / 4
X = Y + (2 + 4)
MYNEWVAR = MYOLDVAR ^ 2.123 + Y

The seven pre-assigned constant variables are listed below:

Constants

GAMMA	0.57721566490153
PHI	1.61803398874989
PI	3.14159265358979
DEG	57.2957795130823
E	2.71828182845904
ROW	Current observation
TOTROW	Total number of observation

The built-in common mathematical functions are listed below:

Built In Functions

Function	Description	Example
SIN	Returns the sine of A (A is in radians).	X = SIN(A)
COS	Returns the cosine of A (A is in radians).	X = COS(A)
TAN	Returns the tangent of A (A is in radians)	X = TAN(A)
ATAN	Returns the angle of X (A is in radians from -Pi/2 to +Pi/2).	A = ATAN(X)
ATAN2	Returns the angle of Y/X (A is in radians from -Pi to +Pi).	A = ATAN2(Y,X)

LOG	Returns the log to the base *e* of Y.	X = LOG(Y)
LOG10	Returns the log to the base 10 of Y.	X = LOG10(Y)
EXP	Returns the value *e* raised to the Y power.	X = EXP(Y)
SQRT	Returns the square root of Y.	X = SQRT(Y)
INT	Returns the integer value of Y.	X = INT(Y)
ABS	Returns the absolute value of Y.	X = ABS(Y)
RANDOM	Returns a uniform distribution with a max/min of R	X = RANDOM(R)
NORMAL	Return a gaussian distribution with a max/min of R	X = NORMAL(R)
RUNAVE	Returns the running average of variable X for the last N observations.	X = RUNAVE(X,N)
LAG	Returns variable X lagged by N observations.	X = LAG(X,N)
LEAD	Returns variable X Leading by N observations.	X = LEAD(X,N)
DIFLAG	Returns the difference between the current value of X and the lagged (by N obs) value.	X = DIFLAG(X,N)
XIF	Exclude data If expression evaluates to TRUE	XIF(X > Y)

X	**RUNAVE (X,4)**	**LAG(X,2)**	**DIFLAG(X,2)**	**LEAD(X,2)**
1.01000	.	.	.	3.09000
2.04000	.	.	.	4.16000
3.09000	.	1.01000	2.08000	5.25000
4.16000	2.57500	2.04000	2.12000	6.36000
5.25000	3.63500	3.09000	2.16000	7.49000
6.36000	4.71500	4.16000	2.20000	8.64000
7.49000	5.81500	5.25000	2.24000	.
8.64000	6.93500	6.36000	2.28000	.

The built-in function XIF() is the data exclusion operator. It is used to exclude rows of data based on the value of the expression. If the expression within the parenthesis evaluates to TRUE then the data in the current row is ignored during reporting, graphing or neural model creation.

The following is a list of operators:

Operators	
( )	Parentheses
- (Unary Minus)	Arithmetic Negation
^	Power
/	Division
*	Multiplication
+	Addition
-	Subtraction
>	Greater Then
>=	Greater Then or Equal To
<	Less Then
<=	Less Then or Equal To
==	Equal To
!=	Not Equal To

&&	Logical AND
||	Logical OR
!	Logical NOT

NNLIB Library Reference

NNLIB is a portable C source code library that provides the basic functions necessary to create, load from file, train, interrogate and delete neural models. Version 1.0 NNLIB supplied with this software is capable of only replaying a neural model. Version 1.1 is required to externally train a neural model.

Note: Subroutines displayed in RED are components supplied in version 1.1 of NNLIB.

NNLIB subroutine definitions

Neural

Returns	Name	Parameters	Description
NEURAL*	NCreateNeural		Creates an empty Neural object
void	NDeleteNeural	NEURAL *pN	Deletes a Neural object
float	NGetROutput	NEURAL *pN, const int neuron	Gets rescaled neural output from Neural object
float	NGetRInput	NEURAL *pN, const int neuron	Gets re-scaled neural input from Neural object
void	NSetRInput	NEURAL *pN, const int neuron, float f	Sets an input of a Neural object with scaling
char*	NGetROutputFmt	NEURAL *pN, const int neuron	Gets rescaled neural output from Neural object in ASCII format
void	Ninterrogate	NEURAL *pN,float *Ivec,float *Ovec	Interrogate Neural object
void	NFeedForward	NEURAL *pN	Executes the feed forward algorithm
int	NImportNetwork	NEURAL *pN, FILE *fd	Load a Neural object from an ENN file
NEURAL*	LoadNetwork	char *filename	Create and load a Neural object from an ENN file
void	DumpNeural	NEURAL *pN, FILE *fd	Dump a Neural object in ASCII format for diagnostics
int	NBackProp1	NEURAL *pN, int loopcnt	Executes the back propagation algorithm
void	NClearDelta	NEURAL *pN	Clears the accumulated deltas
void	NInitializeNetwork	NEURAL *pN	Inititalizes a Neural object to an un-trained state
int	NQuickTrain	NEURAL *pN, int mode,int increment	Executes the training algorithm until all selected criteria are met
void	NCGTrain	NEURAL *pN	use conjugate gradient optimization on Neural objects weight matrix

Internal Routines

float*	ReallocVector	int num, float *arptr, float init	
float**	ReallocFloats1	int first, int second, float **arptr, float ran	
float**	ReallocFloats2	int first, int second,	

		float **arptr, float ran	
void	NAddHidden	NEURAL *pN	Adds an additional hidden neuron to Neural object
void	NSetDM	NEURAL *pN, DATAMAT *pD	Sets the pointer to the DataMat object
void	NNewTrainingSet	NEURAL *pN, int t,int flag	Load the Neural objects inputs from the training matrix
int	NAI	NEURAL *pN, int flag,int a	Test for incrmental training and generate statistics
float	NCalcRsquare	NEURAL *pN	Calculate statistics
float	NCheckTol	NEURAL *pN, float *min, float *max, int *nmin, int *nmax	Calculate tolerances and training errors
void	Nfrprmn	NEURAL *pN, float *p, float ftol, int *iter, float *fret	Conjugate gradient optimization routine
float	Nbrent	NEURAL *pN,float ax, float bx, float cx, float tol,float *xmin	Used by Nfrprmn
void	Nmnbrak	NEURAL *pN, float *ax, float *bx, float *cx, float *fa, float *fb, float *fc	Used by Nfrprmn
float	Nf1dim	NEURAL *pN, float x	Used by Nfrprmn
void	Nlinmin	NEURAL *pN, float *p, float *xi, float *fret)	Used by Nfrprmn
void	NforwardDIffGradient	NEURAL *pN, float *x0, float *g	Used by Nfrprmn
void	Nrerror	char *error_text	Used by Nfrprmn
float	NErrorFunction	NEURAL *pN, float* x	Used by Nfrprmn
int	ntransl	char *cdummy	Used by import function

DataMat

Returns	Name	Parameters	Description
DATAMAT*	DCreateDataMat		Creates an empty DataMat object
void	DDeleteDataMat	DATAMAT *pD	Deletes a DataMat object
float	Drescale	DATAMAT *pD, float f,char C,int ix	Converts a scaled floating point number into user units
float	Dscale	DATAMAT *pD, float f,char C,int ix	Converts a floating point number (in user units) to internal neural units
char*	RescaleFmt	DATAMAT *pD,float f,char C,int ix	Converts a scaled floating point number into user units and converts to ASCII characters
float	DGetInputVal	DATAMAT *pD, int row, int col	Gets the value of the selected input training matrix cell
float	DGetOutputVal	DATAMAT *pD, int row, int col	Gets the value of the selected output training matrix cell

float	DGetInputTVal	DATAMAT *pD, int row, int col	Gets the value of the selected input test matrix cell
float	DGetOutputTVal	DATAMAT *pD, int row, int col	Gets the value of the selected output test matrix cell
void	DSetInputVal	DATAMAT *pD, int row, int col, float val	Sets the value of the selected input training matrix cell
void	DSetOutputVal	DATAMAT *pD, int row, int col, float val	Sets the value of the selected output training matrix cell
void	DSetInputTVal	DATAMAT *pD, int row, int col, float val	Sets the value of the selected input test matrix cell
void	DSetOutputTVal	DATAMAT *pD, int row, int col, float val	Sets the value of the selected output test matrix cell
void	DReCalcScalingFactor	COL_DESC *cold, int cols	Re-calculate the scaling factor based on the numbers in the training and test matrices

Internal Routines

int	ImportDatamat	FILE *fd, DATAMAT *pD	Imports a DataMat object from ASCII file
void	RemoveTestMatrix	DATAMAT *pD	Removes the test matrix from a DataMat object
void	RemoveMatrix	DATAMAT *pD	Removes the training matrix from a DataMat object
void	ZeroAll	DATAMAT *pD	Clears a DataMat object
void	ZeroColDesc	DATAMAT *pD, int num,COL_DESC* desc	Clears the column description
int	dtransl	char *cdummy	used by import function

Params

Returns	**Name**	**Parameters**	**Description**
Internal Routines			
PARAMS*	CreateParams		Create a Params object
int	ImportParams	FILE *fd, PARAMS *pM	Imports a Params object from ASCII file
void	DumpParams	PARAMS *pM, FILE *fd	Dump the contents of a Params object to an ASCII file

Primfunc

Returns	**Name**	**Parameters**	**Description**
Internal Routines			
float**	alloc_2d_floats	int hi1d, int hi2d	Allocate float matrix
void	free_2d_floats	float **matrix, int hi1d	Free float matrix
void	ToUpper	char *s	Convert string to upper case
int	isstring	const char *s	

int	ChkClip	float f, float hi, float lo	Check for clipping
void	dump	const char *buf,int count	Dump a buffer in ASCII Hex
void	removeleadingblanks	char *s	Remove leading blanks from a buffer
void	fixfieldsize	char *s	
void	Logit	const char* fmt, ...	Log string to NNLIN.LOG file. (similar to printf)
float	ran1	int *idum	Used by gasdev
float	gasdev	int *idum, float dzdiv	Returns a gaussian random number
int	sign	float x, float y	
float	sign_of_arg	float x	
float	Max	float a, float b	
float	random	float randz	Returns a random number

NNLIB Example Program

The following code fragment loads an existing neural model (XOR.ENN) from disk and interrogates it.

```
        NEURAL *tneural;
        FILE *fd;
        float *Ivec,*Ovec;
        int i;
        char m1[16],m2[16];

        tneural = LoadNetwork("XOR.ENN");
        if (tneural == NULL) {
                printf ("Load error\n");
                exit(1);
        }
/* Dump the neural model to disk for diagnostics */
        fd = fopen("dump","w");
        DumpNeural (tneural,fd);
        fclose (fd);

/* Create the input and output vectors for the interrogate routine */
        Ivec = (float*) malloc (sizeof(float)*tneural->m_ninputs);
        Ovec = (float*) malloc (sizeof(float)*tneural->m_noutputs);

/* Prompt the user to enter the inputs for the model */
        for (i=0;i<tneural->m_ninputs;i++) {
                sprintf (m1,tneural->m_dm->m_icoldesc[i].format,
                         tneural->m_dm->m_icoldesc[i].min);
                sprintf (m2,tneural->m_dm->m_icoldesc[i].format,
                         tneural->m_dm->m_icoldesc[i].max);
                printf ("Enter %8s (%10s > %10s ) = ",
                         tneural->m_dm->m_icoldesc[i].vlab,m1,m2);
                scanf ("%f",&Ivec[i]);
        }

/* Interrogate the model */
        NInterrogate(tneural,Ivec,Ovec);

/* Display the results *
        for (i=0;i<tneural->m_noutputs;i++) {
                sprintf (m1,tneural->m_dm->m_ocoldesc[i].format,Ovec[i]);
                printf ("\n%8s = %10s",tneural->m_dm->m_ocoldesc[i].vlab,m1);
        }

/* Free the vectors allocated my malloc */
        free (Ivec);
        free (Ovec);

/* Delete the neural model */
        NDeleteNeural(tneural);
```

Example: LOGIC Dataset

LOGIC Detailed Description
File Name - LOGIC.RAW

Description: This dataset contains a truth table of three logical operations (i.e. AND, OR and XOR). The experiment is designed to show the results of the three separate logical operations given the same inputs. The data entered into the table has been translated from the logical language into a numerical representation (i.e. 0 = FALSE and 1 = TRUE).

Column Names	**Column Description**
IN1	First input into the logical operation
IN2	Second input into the logical operation
AND	Logical AND results
OR	Logical OR results
XOR	Logical XOR results

Data Analysis Analysis is not needed due to the small size of the dataset.

Model Building It is suggested to develop 4 models with this dataset. Build a separate model for each of the logical operations and an all inclusive model. The 4 models built are:

```
AND   : IN(IN1, IN2) => OUT(AND)
OR    : IN(IN1, IN2) => OUT(OR)
XOR   : IN(IN1, IN2) => OUT(XOR)
LOGIC : IN(IN1, IN2) => OUT(AND, OR, XOR)
```

The previous notation reads: Model LOGIC has IN1 and IN2 as inputs and generates AND, OR and XOR as outputs.
After creating each model select **Initialize** and **Start Training** commands from the Model menu.

Model Analysis All four models were created and trained using the initial factory default settings for the training parameters. After training the following model statistics were reported.

Analysis of model AND

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
IN1	0.500000	0.577350	0.000000	1.000000	1.000000
IN2	0.500000	0.577350	0.000000	1.000000	1.000000
Measured	0.250000	0.500000	0.000000	1.000000	0.750000
Predicted	0.232930	0.498595	-0.126712	0.970787	0.745791
Residual	0.017070	0.081808	-0.059520	0.126712	0.020078
R Square			0.973230		

Analysis of model OR

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
IN1	0.500000	0.577350	0.000000	1.000000	1.000000
IN2	0.500000	0.577350	0.000000	1.000000	1.000000
Measured	0.750000	0.500000	0.000000	1.000000	0.750000
Predicted	0.754372	0.491654	0.019418	1.056489	0.725170
Residual	-0.004372	0.041884	-0.056489	0.034900	0.005263
R Square			0.992983		

Analysis of model XOR

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
IN1	0.500000	0.577350	0.000000	1.000000	1.000000
IN2	0.500000	0.577350	0.000000	1.000000	1.000000
Measured	0.500000	0.577350	0.000000	1.000000	1.000000
Predicted	0.500708	0.574554	-0.001126	0.999090	0.990337
Residual	-0.000708	0.004522	-0.007405	0.002535	0.000061
R Square			0.999939		

Analysis of model LOGIC

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
IN1	0.500000	0.577350	0.000000	1.000000	1.000000
IN2	0.500000	0.577350	0.000000	1.000000	1.000000
AND					
Measured	0.250000	0.500000	0.000000	1.000000	0.750000
Predicted	0.215229	0.517050	-0.197364	0.969989	0.802023
Residual	0.034771	0.119030	-0.086510	0.197364	0.042504
R Square			0.943328		
OR					
Measured	0.750000	0.500000	0.000000	1.000000	0.750000
Predicted	0.830179	0.532606	0.075665	1.322547	0.851008
Residual	-0.080179	0.175099	-0.322547	0.088393	0.091979
R Square			0.877361		
XOR					
Measured	0.500000	0.577350	0.000000	1.000000	1.000000
Predicted	0.502921	0.339727	0.247580	1.001628	0.346244
Residual	-0.002921	0.471165	-0.418133	0.655659	0.665989
R Square			0.334011		

After reviewing the above model statistics it was noted that the first three separate models predicted the output very well. However, the results of the LOGIC model showed a significant loss of accuracy (as measured by R Square) when combining the three logic functions. The all inclusive model cannot predict as well as the separate models because the default training parameters did not allow the model to build up enough internal complexity. The following table demonstrates that selecting any type of training that will raise the internal complexity will also result in better models. The highlighted model was the initial factory default parameters model shown above.

Training Type	**Count**	**Options**	**AND**	**OR**	**XOR**
AI	1000		0.943328	0.877361	0.334011
Standard 4 Hid	1000		0.894768	0.999530	0.999996
AI	1000	Connect I/O	0.991743	0.996463	0.993545
AI	5000		1.000000	1.000000	1.000000
Standard 4 Hid	1000	CG Train	0.999996	0.999999	0.999997
Equal Spaced	1000		0.999997	0.999999	1.000000

Example: ENCODE Dataset

ENCODE Detailed Description
File Name - ENCODE.RAW

Description: This dataset contains a truth table of three logical operations (i.e. AND, OR and XOR). The experiment is designed to show the results of the three separate logical operations given the same inputs. The data entered into the table has been translated from the logical language into a numerical representation (i.e. 0 = FALSE and 1 = TRUE).

Column Names	**Column Description**
IN1	Input 1 to encoder or output from decoder
IN2	Input 2 to encoder or output from decoder
IN3	Input 3 to encoder or output from decoder
IN4	Input 4 to encoder or output from decoder
IN5	Input 5 to encoder or output from decoder
IN6	Input 6 to encoder or output from decoder
IN7	Input 7 to encoder or output from decoder
IN8	Input 8 to encoder or output from decoder
OUT1	Output 1 from encoder or input to decoder
OUT2	Output 2 from encoder or input to decoder
OUT3	Output 3 from encoder or input to decoder

Data Analysis The following truth table was used as the dataset.

Encoder/Decoder Truth Table

IN1	IN2	IN3	IN4	IN5	IN6	IN7	IN8	OUT1	OUT2	OUT3
0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0
0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0
0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0

Model Building It is suggested to develop 2 models from this dataset. Build a encoder model (ENCODE) using IN1 through IN8 as inputs and OUT1 - OUT3 as outputs and build a decoder model (DECODE) using OUT1-OUT3 as inputs and IN1-IN8 as outputs. The 2 models built are:

```
ENCODE   : IN(IN1,...,IN8)   => OUT(OUT1,...,OUT3)
DECODE   : IN(OUT1,...,OUT3) => OUT(IN1,...,IN8)
```

Model Analysis Both models were created and trained using the initial factory default settings plus **Standard BEP** for the training parameters. After training the following model statistics were reported:

Model ENCODE

Predicted Outputs	R Square
OUT1	1.000000
OUT2	1.000000
OUT3	1.000000

Model DECODE

Predicted Outputs	R Square
IN1	0.903213
IN2	0.903793
IN3	0.903345
IN4	0.903565
IN5	0.903188
IN6	0.903522
IN7	0.904014
IN8	0.905515

With digital type functions it is hard to get a picture of how well these models are doing. The best way with these particular models is to interactively test them. This can be done using the **Interrogate Model** command in the Model menu.

Example: AIR Dataset

AIR Detailed Description
File Name - AIR.RAW

Description: This dataset was constructed to demonstrate how a neural model can be used to predict a time series. It contains 12 columns of the number of tickets sold during the previous twelve months followed by the number of tickets sold during the next month. The dataset was generated from the following table titled **Airline Ticket Sales 1980-1989** by re-arraigning the first 9 rows for use as a training matrix and the last row as a test matrix.

Airline Ticket Sales 1980 - 1989

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
1980	145	153	171	167	157	179	190	192	178	153	133	151
1981	155	163	183	172	162	194	221	220	204	172	144	182
1982	188	197	231	212	224	231	256	263	242	211	190	218
1983	224	234	253	235	237	286	300	313	275	249	223	253
1984	254	257	309	306	295	315	345	356	308	271	235	261
1985	267	243	303	295	304	344	394	375	338	295	261	299
1986	319	304	342	350	355	410	473	452	402	360	309	365
1987	367	362	413	408	416	487	537	527	460	397	347	399
1988	411	393	464	455	461	546	604	608	523	452	399	438
1989	439	413	468	449	473	565	641	656	527	469	401	439

The following table demonstrates how the previous table was rearranged to be used as a training matrix.

Re-arraigned Ticket Sales

M1	M2	m3	M4	M5	M6	M7	M8	M9	M10	M11	M12	NEXT
145	153	171	167	157	179	190	192	178	153	133	151	155
153	171	167	157	179	190	192	178	153	133	151	155	163
171	167	157	179	190	192	178	153	133	151	155	163	183
167	157	179	190	192	178	153	133	151	155	163	183	172
and so on...												

Column Names	**Column Description**
M1	The number of tickets sold twelve months ago
M2	The number of tickets sold eleven months ago
M3	The number of tickets sold ten months ago
M4	The number of tickets sold nine months ago
M5	The number of tickets sold eight months ago
M6	The number of tickets sold seven months ago
M7	The number of tickets sold six months ago
M8	The number of tickets sold five months ago
M9	The number of tickets sold four months ago

M10	The number of tickets sold three months ago
M11	The number of tickets sold two months ago
M12	The number of tickets sold last month
NEXT	The number of tickets that will be sold this month

Data Analysis A **By Row Matrix** graph was printed to see the monthly trend and verify that there were no gross errors in the dataset.

Model Building One model was constructed from this dataset:

```
AIR   : IN(M1,M2,...,M12) => OUT(NEXT)
```

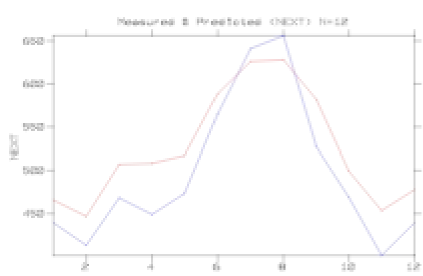
Model Analysis The model was created and trained using the initial factory default settings for the training parameters. After training the following model statistics were reported.

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
M1	277.88541	93.690969	133.00000	537.00000	833909.78
M2	280.65624	93.652983	133.00000	537.00000	833233.70
M3	283.15624	93.412425	133.00000	537.00000	828958.70
M4	286.20833	94.489759	133.00000	537.00000	848189.88
M5	289.20833	95.234110	133.00000	537.00000	861605.89
M6	292.37499	95.843983	133.00000	537.00000	872676.56
M7	296.19791	98.555103	133.00000	546.00000	922745.29
M8	300.51041	102.82453	133.00000	604.00000	1004423.9
M9	304.84375	106.88885	133.00000	608.00000	1085396.5
M10	308.43750	108.36874	133.00000	608.00000	1115659.5
M11	311.55208	108.15085	133.00000	608.00000	1111177.6
M12	314.32292	106.92875	144.00000	608.00000	1086207.0
NEXT					
Measured	317.31250	106.32475	144.00000	608.00000	1073970.6
Predicted	319.78079	105.50861	171.21496	591.72290	1057546.5
Residual	-2.468285	16.031915	-51.89569	34.427551	24417.117
R Square			0.977265		

To see how the model predicts the next twelve months select **Use Test Matrix** from the Model menu and re-run the model statistics.

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
NEXT					
Measured	495.00001	84.744527	401.00000	656.00000	78997.984
Predicted	524.93071	65.081795	446.55816	628.02099	46592.040
Residual	-29.93070	26.557598	-59.22692	27.979004	7758.3662
R Square			0.901790		

As you can see, the worst case under prediction was around 59 and the worst case over prediction was 28 seats. The following plot graphically demonstrates the result.



The command used was the **Measured and Predicted** command from the Graph menu.

Example: VEL Dataset

VEL Detailed Description
File Name - VEL.RAW

Description: This dataset was constructed to demonstrate how well a neural model can predict a trajectory. It contains the distance measurement, the angle of launch and the initial velocity. Along with the aforementioned columns the dataset also includes the aforementioned columns with noise added, plus a column of just noise so that you can experiment building neural models with noisy signals and compare them with ideal models.

Column Names	**Column Description**
ANGLE	Angle measured from horizontal
VEL	Initial velocity
RANGLE	Angle with Gaussian noise added
RVEL	Initial velocity with Gaussian noise added
NOISE	Just Gaussian noise
DIST	Distance traveled by projectile
RDIST	Distance traveled by projectile with Gaussian noise added

Data Analysis A statistics report was generated using the **Basic Statistics** command in the Data menu. This gives us an overall picture of the dataset. If correlations are of interest they can be viewed using the **Correlation Analysis** command also in the Data menu.

By viewing the data matrix it can be observed that the initial velocity was varied from 0 to 100 by 5 and the launch angle was varied from 3 to 87 by 3.

Variable	N	Mean	Std Dev	Minimum	Maximum
ANGLE	609	45.000000	25.120434	3.000000	87.000000
VEL	609	50.000000	30.301392	0.000000	100.00000
RANGLE	609	45.141823	25.438328	-5.960000	92.910004
RVEL	609	49.991724	30.296755	0.000000	102.98999
NOISE	609	-0.023645	3.387809	-10.18000	11.090000
DIST	609	61.804992	65.890137	0.000000	258.10000
RDIST	609	61.801954	65.948653	-0.040000	262.64001

Model Building Two separate models were constructed from this dataset. The first model uses simply the initial velocity and the launch angle:

```
VEL1    : IN(VEL,ANGLE) => OUT(DIST)
```

After the previous model was analyzed, and determined to be not good enough, a second model was constructed that used trigonometric functions as inputs rather than the simple angle:

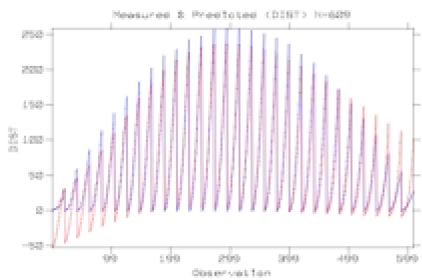
```
VEL2    : IN(VEL,SANG,CANG) => OUT(DIST)
```

Model Analysis The model was created and trained using the initial factory default settings for the training parameters plus CG. CG training was added because a trajectory is known to be trigonometric in nature and harder training is necessary. After training the following model statistics were reported.

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
ANGLE	45.000000	25.120434	3.000000	87.000000	383670.01
VEL	49.999999	30.301391	0.000000	100.00000	558249.97
DIST					
Measured	61.804992	65.890137	0.000000	258.09997	2639638.2

Predicted	58.920517	66.373302	-52.79543	236.05107	2678492.4
Residual	2.884475	15.524349	-73.25933	52.795437	146531.29
R Square			0.944488		

Although the R Square statistic is respectable, a closer examination using the **Measured and Predicted** or **Measured vs. Predicted** graphs reveal significant problems predicting the distance when the angle is near 0 or 90 degrees. The following graph demonstrates the problem.



Therefore, a second model was created using calculated columns to provide more information. Two additional columns were created to include the sine and cosine of the launch angle into the model. To do this, first add the following two equations to the equation string of the data matrix:

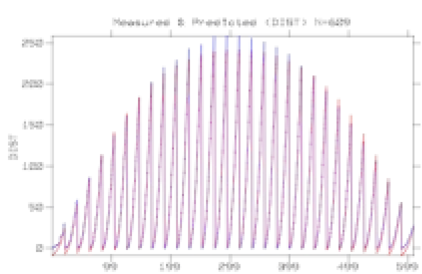
SANG = SIN (ANGLE * 2 * PI / 360)

CANG = COS (ANGLE * 2* PI / 360)

Then create the columns using the **Append Calculated Columns** command in the Edit menu. After training the following model statistics were reported.

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
VEL	49.999999	30.301391	0.000000	100.00000	558249.97
SANG	0.641180	0.298386	0.052336	0.998630	54.132656
CANG	0.641180	0.298386	0.052336	0.998630	54.132656
DIST					
Measured	61.804992	65.890137	0.000000	258.09997	2639638.2
Predicted	61.608327	65.665710	-8.983620	240.97804	2621687.1
Residual	0.196665	4.168274	-10.31471	18.355606	10563.698
R Square			0.995998		

The R Square statistic is better then the previous model and the **Measured and Predicted** or **Measured vs. Predicted** graphs reveal a significant increase in the overall accuracy.



Example: COATING Dataset

COATING Detailed Description
File Name - COATING.RAW

Description: The coating dataset contains the data from an incomplete designed experiment. This experiment was designed to determine the ideal levels of the five independent variables (STARCH, LATEX, HP91, COATWT and CPSI) necessary to maintain minimum levels of the dependent variables (BRIGHTNESS, OPAC and GLOSS). In this dataset STARCH, LATEX, HP91 and COATWT are varied to five different levels while CPSI is varied to two levels. The independent variables STARCH, LATEX, HP91 and CPSI can set to the desired target and maintained, however, COATWT cannot controlled as accurately. Therefore, the targeted COATWT value is later replaced with the measured value.

Column Names	**Column Description**
STARCH	The percentage of starch added to the coating.
LATEX	The percentage of latex added to the coating. Latex is a rubber used as a binding agent in coatings.
HP91	The percentage of HP91 added to the coating. HP91 is a plastic pigment.
COATWT	The measured amount of coating applied to the paper.
CPSI	The pressure applied by a super-calander to polish the surface of the coated paper.
BRIGHT	The measured brightness of the finished paper/coating. Brightness is the measurement of how white the surface of the piece of paper is.
OPAC	The measured opacity of the finished paper/coating. Opacity is a measurement of how opaque (impenetrable to light) a piece of paper is.
GLOSS	The measured gloss of the finished paper/coating. Gloss is a measurement of how polished the surface of a piece of paper looks.

Data Analysis A statistics report was generated using the **Basic Statistics** command in the Data menu. This gives us an overall picture of the dataset. If correlations are of interest they can be viewed using the **Correlation Analysis** command also in the Data menu.

Variable	N	Mean	Std Dev	Minimum	Maximum
STARCH	128	17.928281	5.206710	6.500000	26.000000
LATEX	128	12.720312	3.034249	6.500000	19.500000
HP91	128	5.735938	2.644463	0.000000	11.000000
COATWT	128	4.677344	0.904097	2.960000	6.380000
CPSI	128	45.500000	19.576621	26.000000	65.000000
BRIGHT	128	65.760938	1.091934	62.900002	68.400002
OPAC	128	70.977344	1.388775	67.699997	74.400002
GLOSS	128	39.965625	7.816129	25.600000	57.500000

Model Building Four models were constructed from this dataset. The first model included all dependent variables into one model:

```
COATING   : IN(STARCH, LATEX, HP91, COATWT, CPSI)
            => OUT(BRIGHT, OPAC, GLOSS)
```

The next three models were constructed to predict the dependent variables separately:

```
BRIGHT    : IN(STARCH, LATEX, HP91, COATWT, CPSI)
            => OUT(BRIGHT)
OPAC      : IN(STARCH, LATEX, HP91, COATWT, CPSI)
            => OUT(OPAC)
GLOSS     : IN(STARCH, LATEX, HP91, COATWT, CPSI)
            => OUT(GLOSS)
```

Model Analysis

The first model (COATING) was created and trained using the initial factory default settings for the training parameters plus **Standard BEP**. After training the following model statistics were reported.

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
STARCH	17.928281	5.206710	6.500000	26.000000	3442.9479
LATEX	12.720312	3.034249	6.500000	19.500000	1169.2470
HP91	5.735938	2.644464	0.000000	11.000001	888.13480
COATWT	4.677344	0.904097	2.960000	6.380000	103.80870
CPSI	45.500000	19.576621	26.000000	65.000000	48672.000
BRIGHT					
Measured	65.760938	1.091934	62.900002	68.400002	151.42473
Predicted	65.907051	1.093083	63.297192	68.363007	151.74355
Residual	-0.146113	0.368337	-1.586678	0.660271	17.230382
R Square			0.886212		
OPAC					
Measured	70.977344	1.388775	67.699997	74.400002	244.94435
Predicted	70.890938	1.287656	67.926071	74.046333	210.57338
Residual	0.086406	0.491884	-1.036316	2.010513	30.727583
R Square			0.874553		
GLOSS					
Measured	39.965625	7.816129	25.600000	57.500000	7758.6686
Predicted	39.983750	7.332670	25.655918	58.590076	6828.5421
Residual	-0.018125	2.200280	-6.198959	5.021023	614.83627
R Square			0.920755		

The next three models (BRIGHT, OPAC and GLOSS) were trained using the same training parameters as the first model. This shows that modeling the dependent variables separately can produce higher R Square models under identical conditions.

BRIGHT					
Measured	65.760938	1.091934	62.900002	68.400002	151.42473
Predicted	65.877290	1.118312	63.187656	68.545723	158.82889
Residual	-0.116352	0.321667	-1.404861	0.710152	13.140674
R Square			0.913220		
OPAC					
Measured	70.977344	1.388775	67.699997	74.400002	244.94435
Predicted	71.015069	1.306490	68.109215	74.472168	216.77825
Residual	-0.037725	0.450619	-1.255150	1.767052	25.788266
R Square			0.894718		
GLOSS					
Measured	39.965625	7.816129	25.600000	57.500000	7758.6686
Predicted	39.664502	7.135378	26.043440	55.892056	6466.0295
Residual	0.301123	2.021777	-4.901318	5.476116	519.12310
R Square			0.933091		

The performance of the first model can be increased by tweaking the training parameters. In this case **Connect IO** and **CG Training** was added to the default settings. After training the following model statistics were reported.

BRIGHT					
Measured	65.760938	1.091934	62.900002	68.400002	151.42473
Predicted	65.760366	1.048789	63.203236	68.333611	139.69475
Residual	0.000572	0.305982	-1.240593	0.699280	11.890349

R Square			0.921477		
OPAC					
Measured	70.977344	1.388775	67.699997	74.400002	244.94435
Predicted	70.978319	1.314241	67.848198	74.354477	219.35800
Residual	-0.000975	0.438371	-1.064926	1.595451	24.405517
R Square			0.900363		
GLOSS					
Measured	39.965625	7.816129	25.600000	57.500000	7758.6686
Predicted	39.904586	7.683301	24.798870	58.689686	7497.2046
Residual	0.061039	1.899198	-4.346085	4.689075	458.08291
R Square			0.940959		

The final models were exported to a system optimizer to find the answer to: What is the lowest cost coating mixture that can still meet the minimum specifications of BRIGHT, OPAC and GLOSS? In the optimizer the cost of the coating was calculated by the following equation:

```
COST = C1COATWT(C2LATEX + C3STARCH + C4HP91)
```

The solution to the problem would minimize COST while maximizing BRIGHT, OPAC and GLOSS and subject to the following constraints: BRIGHT > 71.5, OPAC > 78 and GLOSS > 48.

Optimization can not be performed in this version of the program.

Example: SODIUM Dataset

SODIUM Detailed Description
File Names - H2.RAW, CO.RAW,
COH2.RAW,MIX.RAW,COH2MIX.RAW

Description: This dataset is really made up of 5 separate datasets. It is the result of a chemical experiment to determine the best way to reduce sodium sulfate to sodium sulfide using hydrogen, carbon monoxide or a mixture of both.

The plan was to run each experiment to 160 minutes twice, however, the mixture experiment could not be run longer then 70 minutes due to a problem with the experimental apparatus. The data before sixty minutes is not of any use (all the important stuff happens from 60 to 160 minutes). Due to this problem the MIX experiment yielded only one point per run.

H2	The result of a designed experiment using only hydrogen gas as the agent and varying temperature and gas concentration.
CO	The result of a designed experiment using only carbon monoxide gas as the agent while varying temperature and gas concentration.
COH2	The result of combining both the H2 and CO datasets into one using the **Concatenate Data Matrices** command in the Data menu. The combining of these two datasets is straight forward in that the two experimental designs are similar. It involves creating a new field in both matrices and setting the missing values to zero.
MIX	The result of a designed experiment using a mixture of both hydrogen and carbon monoxide gases as the agent while varying the gas concentrations and temperatures.
COH2MIX	The combined dataset of COH2 and MIX experiments. Combining these two datasets is mechanically easy in that both matrices have the same fields. However, statistically the dataset are very different. COH2 contains experimental runs where time varies from 60 to 160 and MIX only contains the 60 minute values. It is okay to paste these datasets together as long as the consequences are understood. The MIX data will serve as reference points the model must traverse. The MIX data is very important to the model because it contains the only points where both gases are present at the same time. Other reference points could also be entered in this manner (i.e. H2 = 0, CO = 0 and CONV = 0).

Column Names	**Column Description**
TIME	Time elapsed since beginning of the run
H2	Percentage of hydrogen gas used
CO	Percentage of carbon monoxide gas used
TEMP	Temperature during the run
AVTEMP	Average temperature of run
CONV	Percentage of Na2SO4 converted

Data Analysis H2 and CO contain a central composite design varying concentration of the gas and the reaction temperature. Each run was replicated twice. The design yielded a total of 10 runs. The MIX experiment is a mixture design where the concentrations of H2 and CO are varied and the temperature is held constant at the center point. The following **Basic Statistics** reports were generated for all the datasets.

H2					
Variable	N	Mean	Std Dev	Minimum	Maximum
TIME	110	110.00000	31.767504	60.000000	160.00000
H2	110	50.000000	22.463018	25.000000	75.000000
TEMP	110	1203.8877	18.054523	1179.9599	1225.8900
AVTEMP	110	1203.7799	18.949085	1181.9000	1226.1999
CONV	110	0.837782	0.089136	0.629880	0.997350
CO					
TIME	110	110.00000	31.767504	60.000000	160.00000
CO	110	27.000000	20.241659	5.000000	50.000000
TEMP	110	1200.5162	19.210413	1173.4699	1223.4499
AVTEMP	110	1199.7699	20.422422	1174.6999	1221.9000
CONV	110	0.665540	0.210730	0.163860	0.979830
COH2					
TIME	220	110.00000	31.694892	60.000000	160.00000
CO	220	13.500000	19.672548	0.000000	50.000000
H2	220	25.000000	29.647857	0.000000	75.000000
TEMP	220	1202.2020	18.675406	1173.4699	1225.8900
AVTEMP	220	1201.7749	19.756972	1174.6999	1226.1999
CONV	220	0.751661	0.183050	0.163860	0.997350
MIX					
TIME	8	60.000000	0.000000	60.000000	60.000000
CO	8	28.125000	20.863074	0.000000	50.000000
H2	8	29.687500	28.298079	0.000000	75.000000
AVTEMP	8	1202.9000	2.988080	1199.3000	1206.9000
CONV	8	0.599325	0.252241	0.000000	0.758300
COH2MIX. DM					
TIME	229	108.03493	32.553311	60.000000	160.00000
CO	229	13.951965	19.829147	0.000000	50.000000
H2	229	25.491266	29.901225	0.000000	100.00000
TEMP	229	1202.2224	18.311378	1173.4699	1225.8900
AVTEMP	229	1201.8122	19.371543	1174.6999	1226.1999
CONV	229	0.746724	0.186770	0.000000	0.997350

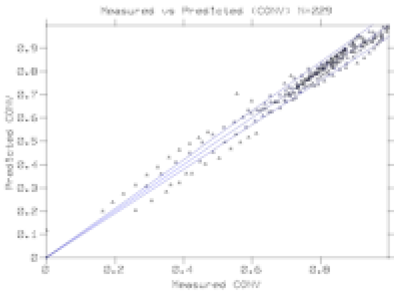
Model Building Many models were built during the course of the analysis, but only the last model is reported. The most complete model was built from the COH2MIX dataset.

```
CONV   : IN(TIME, CO, H2, TEMP)
                 => OUT(CONV)
```

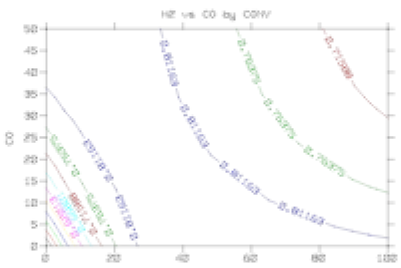
Model Analysis Model (CONV) was created and trained using the initial factory default settings for the training parameters plus **Standard BEP** and **CG Optimization** .After training the following model statistics were reported.

CONV					
Measured	0.746724	0.186770	0.000000	0.997350	7.953296
Predicted	0.746847	0.182348	0.116489	0.991212	7.581214
Residual	-0.000123	0.034309	-0.147950	0.078872	0.268383
R Square			0.966255		

A **Measured vs. Predicted** graph was generated to view how the model performed. This graph demonstrates that the model seems to predict CONV fairly well. The blue lines represent the ± 5% tolerance band.



The following contour graph was generated to demonstrate the surface of the CONV variable in relation to the concentrations of H2 and CO, given TEMP=1200 degrees and TIME=110 minutes.



Example: REDWOOD Dataset

REDWOOD Detailed Description
File Name - REDWOOD.RAW

Description:	The redwood experiment was done to see if redwood chips could be used to replace the less available Douglas fir chips in making wood pulp for container board. A designed experiment was done to set the various percentages of DFIR, HFIR, PINE, REDW and cooking temperatures. A COOK number was included in the dataset for identification purposes only. After each batch cook the pulp properties TYLD, BPH and KAPN were measured. These pulps were refined to three different levels of (REVS) and the pulp property CSF was measured. Finally paper was made from the pulp batches and the following physical measurements were made on the paper TEAR, BURST, FOLD, SCOT and PORS.

Column Names	**Column Description**
COOK	The batch number of the cook.
REVS	The number of revolutions the pulp was refined to.
DFIR	The percentage of Douglas fir chips used in the pulp.
HFIR	The percentage of Hemlock fir chips used in the pulp.
PINE	The percentage of Pine chips used in the pulp.
REDW	The percentage of Redwood chips used in the pulp.
TEMP	The temperature the chips were cooked at.
TYLD	The percentage of pulp made as a fraction of total chips (pulp test).
BPH	The pH of the cook (pulp test).
KAPN	The Kappa number (pulp test)
CSF	The freeness number. (pulp test).
BURST	The result of the burst test (paper test).
FOLD	The result of the fold test (paper test).
SCOT	The Scott Bond test (paper test).
PORS	The porosity measurement (paper test).

Data Analysis	A statistics report was generated using the **Basic Statistics** command in the Data menu. This gives us an overall picture of the dataset. If correlations are of interest they can be viewed using the **Correlation Analysis** command also in the Data menu.

Variable	N	Mean	Std Dev	Minimum	Maximum
COOK	72	252.66666	7.253945	241.00000	266.00000
REVS	72	2520.0000	2072.0106	0.000000	5040.0000
DFIR	72	0.267500	0.134675	0.080000	0.430000
HFIR	72	0.245000	0.088795	0.130000	0.340000
PINE	72	0.280000	0.068669	0.170000	0.340000
REDW	72	0.062500	0.057132	0.000000	0.130000
TEMP	72	447.00000	8.056141	439.00000	455.00000
TYLD	72	0.681931	0.034129	0.603000	0.796000
TEAR	72	27.023889	4.171943	21.440001	36.639999
BPH	69	15.636232	0.596798	14.300000	16.500000
KAPN	72	79.548611	6.466925	69.099998	96.699997
CSF	72	621.61111	42.098997	536.00000	672.00000
BURST	72	5.463750	1.255601	3.230000	6.900000
FOLD	72	2464.0694	715.55764	984.00000	4070.0000
SCOT	72	0.169958	0.070813	0.039000	0.299000
PORS	72	4.729708	2.154128	1.442000	7.824000

Model Building

4 models of unrefined pulp properties were constructed from this dataset. The pulp properties modeled are TYLD, BPH and KAPN and the only numbers to be included into the model(s) are when the REVS is equal to zero (definition of unrefined). To exclude all other rows of data except the REVS=0 add to the exclusions string the following formula:

XIF (REVS != 0)

The first model included all independent variables (except REVS) of the pulp cook into one model predicting the pulp properties:

```
PULP    : IN(DFIR, HFIR, PINE, REDW, TEMP)
            => OUT(TYLD, BPH, KAPN)
```

The next three models were constructed to predict the dependent variables separately:

```
TYLD    : IN(DFIR, HFIR, PINE, REDW, TEMP)
            => OUT(TYLD)
BPH     : IN(DFIR, HFIR, PINE, REDW, TEMP)
            => OUT(BPH)
KAPN    : IN(DFIR, HFIR, PINE, REDW, TEMP)
            => OUT(KAPN)
```

One model of refined pulp properties was created to predict CSF. This is the only pulp property (in this experiment) that varies with REVS so it is treated separately:

```
CSF     : IN(DFIR, HFIR, PINE, REDW, TEMP, REVS)
            => OUT(CSF)
```

Finally a model is constructed to predict all paper properties:

```
ALL     : IN(DFIR, HFIR, PINE, REDW, TEMP, REVS)
            => OUT(TEAR, BURST, FOLD, SCOT, PORS)
```

Model Analysis

The first model (PULP) was created and trained using the initial factory default settings for the training parameters plus **Standard BEP**, **CG Training** and **Connect IO**. After training the following model statistics were reported.

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
DFIR	0.262174	0.137112	0.080000	0.430000	0.413591
HFIR	0.250000	0.088626	0.130000	0.340000	0.172800
PINE	0.277391	0.070014	0.170000	0.340000	0.107843
REDW	0.065217	0.057672	0.000000	0.130000	0.073174
TEMP	446.65217	8.172063	439.00000	455.00000	1469.2173
TYLD					
Measured	0.681913	0.037294	0.617000	0.796000	0.030598
Predicted	0.681315	0.029041	0.634339	0.765630	0.018554
Residual	0.000598	0.021773	-0.041405	0.049940	0.010430
R Square			0.659141		
BPH					
Measured	15.634783	0.608731	14.300000	16.500000	8.152171
Predicted	15.629808	0.554882	14.611290	16.605103	6.773670
Residual	0.004975	0.267684	-0.486740	0.735995	1.576404
R Square			0.806628		
KAPN					
Measured	79.917391	6.442589	69.199997	96.699997	913.15291
Predicted	79.998469	6.075874	71.612045	92.992897	812.15743
Residual	-0.081079	2.005435	-3.490204	3.707100	88.478890
R Square			0.903106		

After viewing the rather low R Square statistic it was decided to create separate models to increase the performance. The following three models were trained using the same parameters as the previous model.

TYLD					
Measured	0.682625	0.036640	0.617000	0.796000	0.030878
Predicted	0.682765	0.035731	0.625390	0.797267	0.029364
Residual	-0.000140	0.008722	-0.019358	0.021583	0.001750
R Square			0.943335		
BPH					
Measured	15.634783	0.608731	14.300000	16.500000	8.152171
Predicted	15.633953	0.548757	14.371428	16.485111	6.624947
Residual	0.000830	0.236424	-0.522560	0.495054	1.229720
R Square			0.849154		
KAPN					
Measured	79.550000	6.552994	69.199997	96.699997	987.65988
Predicted	79.535829	6.383595	69.700935	92.437851	937.25656
Residual	0.014170	1.421623	-2.365013	4.262146	46.483297
R Square			0.952936		

A single model was constructed to predict CSF. The following model was trained using the same parameters as the first model.

CSF					
Measured	621.61111	42.098997	536.00000	672.00000	125835.11
Predicted	621.70089	41.537126	535.97607	670.44061	122498.63
Residual	-0.089779	6.975161	-13.03338	21.181396	3454.3534
R Square			0.972549		

A single model was constructed to predict all paper properties. The following model was trained using the same parameters as the first model.

TEAR					
Measured	27.023889	4.171943	21.440001	36.639999	1235.7629
Predicted	27.003455	3.946271	21.924404	35.265640	1105.6870
Residual	0.020434	1.333415	-2.842812	3.702446	126.23768
R Square			0.897846		
BURST					
Measured	5.463750	1.255601	3.230000	6.900001	111.93390
Predicted	5.464549	1.239189	3.265778	6.739976	109.02691
Residual	-0.000799	0.195209	-0.414065	0.485160	2.705563
R Square			0.975829		
FOLD					
Measured	2464.0694	715.55765	984.00006	4070.0000	36353615.
Predicted	2464.0329	639.97692	1189.0734	3518.6354	29079503.
Residual	0.036491	323.19581	-498.2634	998.62963	7416343.1
R Square			0.795994		
SCOT					
Measured	0.169958	0.070813	0.039000	0.299000	0.356031
Predicted	0.170150	0.066934	0.050595	0.260065	0.318087
Residual	-0.000192	0.023071	-0.059773	0.053572	0.037792
R Square			0.893851		
PORS					
Measured	4.729708	2.154128	1.442000	7.824000	329.45902
Predicted	4.724768	2.139457	1.539140	7.431902	324.98666
Residual	0.004940	0.242487	-0.495115	0.500623	4.174795
R Square			0.987328		

The final question. What mixture of wood chips, cooking temperature and REVS would allow us the meet the minimum paper properties while minimizing DFIR and maximizing TYLD?

subject to the following constraints:
FOLD > 2500
SCOT > 0.14
REDW > 0.10
DFIR+HFIR+PINE+REDW < 1.0

Optimization can not be performed in this version of the program.

Example: RING Dataset

RING Detailed Description
File Name - RING.RAW

Description: The RING dataset was captured during the normal operation of a paper machine. The intent of the data capture was to see if any of the standard logged process variables could be used to predict a physical property (MDRING) of the manufactured paper board. This experiment is really a fishing expedition in that no designed experiment was performed on the process variables. However, there may be enough information in the log to point to variables that have a major effect.

Column Names	**Column Description**
MDRING	Ring crush measured in machine direction
CONDWT	Basis weight measurement
AVEMO	Average moisture of the paper board measurement
SPEED	Machine speed measurement
FL1	Flow rate measurement
CS1	Consistancy measurement
FL2	Flow rate measurement
FL3	Flow rate measurement
FL4	Flow rate measurement
HP1	Horse power measurement
FL5	Flow rate measurement
FL6	Flow rate measurement
CS2	Consistancy measurement
AN1	Freeness measurement
CS3	Consistancy measurement

Data Analysis A statistics report was generated using the **Basic Statistics** command in the Data menu. This gives us an overall picture of the dataset. With this much data it is highly recommended that the data be viewed using the **By Row Matrix** command in the graph menu.

Variable	N	Mean	Std Dev	Minimum	Maximum
MDRING	507	120.27810	15.541973	75.000000	150.00000
CONDWT	507	40.058619	3.296797	32.849998	46.259998
AVEMO	507	6.249132	0.443878	4.250000	7.990000
SPEED	507	2132.8500	125.89145	1606.0000	2305.0000
FL1	507	21.466075	2.696087	13.000000	26.100000
CS1	507	3.231894	0.412474	2.500000	5.410000
FL2	507	67.242604	11.245570	35.799999	103.00000
FL3	507	8704.5956	575.66911	6849.0000	9805.0000
FL4	507	51733.443	3700.4051	10000.000	61023.000
HP1	507	1.094359	0.303432	0.500000	2.150000
FL5	507	0.064083	0.084342	0.000000	0.470000
FL6	507	42.958383	6.451481	27.700001	58.900002
CS2	507	3.379487	0.317734	3.050000	4.100000
AN1	507	684.21696	63.229086	500.00000	800.00000
CS3	507	5.599053	0.678285	2.850000	6.730000

Model Building A model was built that included all independent variables to predict the MDRING property:

```
MDRING    : IN(CONDWT, AVEMO, SPEED,FL1, CS1, FL2,
            FL3, FL4, HP1, FL5, FL6, CS2, AN1,
```

```
CS3 ) => OUT(MDRING)
```

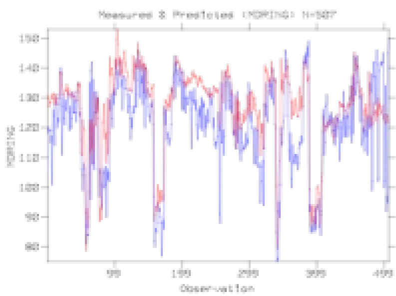
Model Analysis

The model was created and trained using the initial factory default settings for the training parameters plus **Standard BEP**. After training the following model statistics were reported.

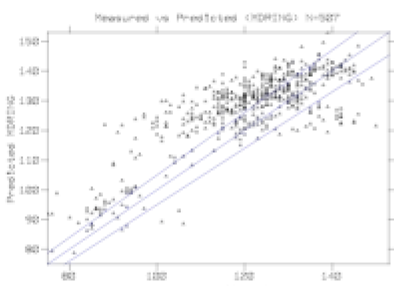
MDRING

Variable	Mean	Std Dev	Minimum	Maximum	Sum Sq
Measured	120.27810	15.541973	75.000000	150.00000	122225.78
Predicted	126.97214	13.398455	78.760902	152.96443	90836.403
Residual	-6.694042	9.308496	-33.88612	28.481262	43843.936
R Square			0.641287		

A **Measured and Predicted** graph was generated to view how the model performed as a time series. This graph demonstrates that the model seems to capture much of the variability, but there are major gaps.



A **Measured vs. Predicted** graph was also generated to demonstrate the lack of fit.



A sensitivity analysis was run to see which variables account for most of the variability of MDRING. The results are presented below.

Sensitivity Analysis of MDRING

Variable Name	Initial Setting	Percent Total
FL1	19.6	+0.13543
FL4	47204.5	+0.12555
HP1	1.33	+0.12218
CS2	3.58	-0.11213
SPEED	1955.5	-0.09160
AVEMO	6.12	-0.08170
FL5	0.24	-0.07192

CS1	3.96	-0.05456
AN1	650.0	+0.05361
CONDWT	39.56	+0.04894
CS3	4.79	-0.03340
FL6	43.3	+0.03116
FL2	69.4	-0.02752
FL3	8327.0	+0.01030

Example: SPECIES Dataset

SPECIES Detailed Description
File Name - SPECIES.RAW

Description: The species dataset was downloaded from a process control system in a paper mill. It was the result of an experiment to see if an algorithm could be developed that could predict when the wood species changed in the output of a continuous wood digestor. A continuous digestor converts wood chips into paper pulp. It is like a long pipe that you dump chips in a the top and pulp falls out at the bottom. The digestor is a hydraulic system that operates under high pressure and temperature. The inside of a digestor is a very corrosive and hence cant be well instrumented. The wood chips usually spend 3-5 hours making the trip from the top to the bottom.

Paper is made of a mixture of two species of wood (hardwood and softwood). Because the two species cook (digest) so differently they must be processed and stored separately. The ideal process would have two digestors (one for softwood and one for hardwood), however due to the expense, many mills have only one. In these mills the digestor is swung between the two species. Temperatures, chemicals, flows and cooking time vary between the two species. Pulp manufactured during this swing is called twilight pulp because it is nether hardwood or softwood. The twilight pulp must be treated as if it was hardwood thus reducing the profitability of the process. If s detector could be developed that could more exactly determine when the crossover was between the species the process would be more efficient.

The species dataset represents a 33 hour period. Each row is a one minute scan. Signal A3 was captured by an automatic sampling device that bottled the pulp. The A3 sample was then measured in a laboratory at a later time. The two questions to be answered by this experiment are 1) can the species change be detected and 2) what signals are the most important?

Column Names	**Column Description**
A1	Blow line gamma process measurement
A2	Refractivity index process measurement
A3	Softwood present calculation (laboratory test)
A4	Triple D calculation (from process measurements)
A5	Consistency process measurement

Data Analysis A statistics report was generated using the **Basic Statistics** command in the Data menu. This gives us an overall picture of the dataset. With this much data it is highly recommended that the data be viewed using the **By Row Matrix** command in the graph menu.

Variable	N	Mean	Std Dev	Minimum	Maximum
A1	2000	0.345829	0.125509	0.176045	0.715970
A2	2000	0.493294	0.166214	0.176530	0.715647
A3	2000	0.543000	0.498272	0.000000	1.000000
A4	2000	0.300089	0.150496	-0.074310	0.882878
A5	2000	0.366992	0.191050	0.136625	0.742250

Model Building Three models were constructed to predict A3 from the input variables:

```
A3a    : IN(A1, A2, A4, A5) => OUT(A3)
A3b    : IN(A1, A4, A5)     => OUT(A3)
A3c    : IN( A4, A5)        => OUT(A3)
```

Signal A2 was eliminated from model A3b because it didnt appear to be significant. Likewise signals A1 and A2 were eliminated from model A3c.

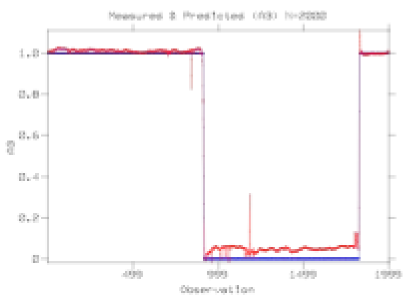
Model Analysis

The model was created and trained using the initial factory default settings for the training parameters. After training the following model statistics were reported.

A3

Measured	0.543000	0.498272	0.000000	1.000000	496.30200
Predicted	0.569940	0.478081	-0.018100	1.110421	456.89343
Residual	-0.026940	0.033566	-0.400754	0.499685	2.252166
R Square			0.995462		

A **Measured and Predicted** graph was generated to view how the model performed as a time series. This graph demonstrates that the model seems to predict A3 very well.



A sensitivity analysis was run to see if any of the variables could be eliminated from the model. The signal A2 is a candidate for elimination.

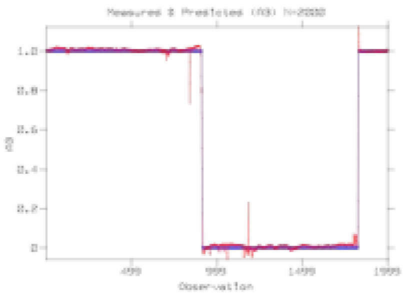
Sensitivity Analysis of A3

Variable Name	Initial Setting	Percent Total
A4	0.404284	+0.52183
A1	0.446008	-0.23331
A5	0.439438	-0.19154
A2	0.446089	+0.05332

Another model (without A2) was created to see if the performance is severely effected. As you can see from the statistics and the **Measured and Predicted** plot the performance actually increased.

A3

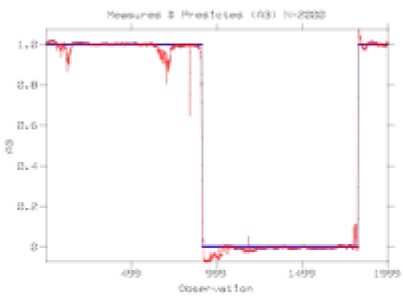
Measured	0.543000	0.498272	0.000000	1.000000	496.30200
Predicted	0.549675	0.496866	-0.059234	1.122736	493.50542
Residual	-0.006675	0.025613	-0.405061	0.493989	1.311433
R Square			0.997358		



So another sensitivity analysis was done and A1 was eliminated. As you can see the model is starting to fall apart but it is still very significant. Further attempts at reducing the number of inputs to one failed.

A3

Measured	0.543000	0.498272	0.000000	1.000000	496.30200
Predicted	0.535189	0.499161	-0.072939	1.074120	498.07506
Residual	0.007811	0.034226	-0.539843	0.389317	2.341613
R Square			0.995282		



Example: NOX Dataset

NOX Detailed Description
File Name - NOX.RAW

Description: The NOX dataset was captured during the normal operation of a power boiler. The intent of the data capture was to see if any of the standard logged process variables could be used to predict the four stack gas variables. This experiment is really a fishing expedition in that no designed experiment was performed on the process variables. However, there may be enough information in the log to point to variables that have a major effect.

Column Names	**Column Description**
AMBAIR	Ambient air temperature.
BARKFEED	Amount of bark fed into the boiler.
BARKOFP	Air pressure over bark bed.
BARKUAIR	Air pressure under bark bed.
COALUPL	Amount of coal fed to upper level.
COALMILV	Amount of coal fed to middle level.
COALLOLV	Amount of coal fed to lower level.
1LVATEMP	Level 1 flame/gas temperature.
2LVATEMP	Level 2 flame/gas temperature.
3LVATEMP	Level 3 flame/gas temperature.
GASBURN	Amount of natural gas fed into boiler (main burners).
GASIGN	Amount of natural gas fed into boiler (ignitor).
OILUPLV	Amount of fuel oil upper level.
OILMILV	Amount of fuel oil lower level.
PAIRUPLV	Primary air feed upper level.
PAIRMILV	Primary air feed middle level.
PAIRLOLV	Primary air feed lower level.
SECUPLV	Secondary air feed upper level.
SECMILV	Secondary air feed middle level.
SECLOLV	Secondary air feed lower level.
STEAMPR	Output steam pressure.
STEAMFLO	Output steam flow.
STEAMTMP	Output steam temperature.
NOX	Nitrogen oxides exhaust from stack.
O2	Free oxygen exhaust from stack.
SO2	Sulfur dioxide exhaust from stack.
OPAC	Opacity of exhaust gases from stack.

Data Analysis A statistics report was generated using the **Basic Statistics** command in the Data menu. This gives us an overall picture of the dataset. With this much data it is highly recommended that the data be viewed using the **By Row Matrix** command in the graph menu.

If you look closely at the NOX, SO2 and OPAC you will see a re-occurring blip. This was traced to a particular maintenance item done once a day.

Variable	N	Mean	Std Dev	Minimum	Maximum
AMBAIR	1340	104.44056	10.743963	85.480003	128.58999
BARKFEED	1340	84.214254	10.266984	51.849998	103.26000
BARKOFP	1340	25.602993	2.868492	-0.030000	26.530001
BARKUAIR	1340	410.85285	54.698823	27.520000	477.77999
COALUPL	1340	0.840000	2.806041	0.000000	15.620000
COALMILV	1340	0.294045	1.433840	-0.070000	10.690000

COALLOLV	1340	0.278000	1.336544	0.010000	14.230000
1LVATEMP	1340	169.54458	27.795764	100.05000	233.13000
2LVATEMP	1340	137.48353	39.120769	85.930000	227.55999
3LVATEMP	1340	113.24388	31.235535	75.570000	238.50999
GASBURN	1340	16.029052	32.871236	-0.020000	155.42999
GASIGN	1340	7.511463	2.763247	-0.040000	14.110000
OILUPLV	1340	0.602015	3.001349	-0.040000	21.070000
OILMILV	1340	0.979067	4.984428	-0.060000	32.340000
PAIRUPLV	1340	26.027418	6.590400	8.530000	35.990002
PAIRMILV	1340	13.760216	10.276768	3.860000	33.669998
PAIRLOLV	1340	9.025209	6.088600	6.140000	31.410000
SECUPLV	1340	81.394933	18.186028	54.389999	160.39999
SECMILV	1340	94.725522	20.819705	70.080002	172.58999
SECLOLV	1340	100.07086	15.736587	74.120003	137.91000
STEAMPR	1340	1638.9450	24.963765	1519.9499	1715.2800
STEAMFLO	1340	489.19210	95.430005	115.61000	694.27002
STEAMTMP	1340	1201.7549	8.199598	1130.4699	1226.1199
NOX	1340	91.407224	27.824783	32.880001	240.52000
O2	1340	9.999037	2.547359	2.930000	23.280001
SO2	1340	32.512619	39.967415	14.300000	370.76001
OPAC	1340	4.133246	1.442123	2.370000	19.900000

Model Building

Due to the large amount of data in this dataset, 80% of it was reserved for testing. The first model was constructed to predict NOX from all input variables:

```
NOX1    : IN(AMBAIR, BARKFEED, BARKOFP, BARKUAIR,
            COALUPL, COALMILV, COALLOLV, 1LVATEMP,
            2LVATEMP, 3LVATEMP, GASBURN, GASIGN,
            OILUPLV, OILMILV, PAIRUPLV, PAIRMILV,
            PAIRLOLV, SECUPLV, SECMILV, SECLOLV,
            STEAMPR, STEAMFLO, STEAMTMP)
         => OUT(NOX)
```

Model Analysis

The first model (NOX1) was created and trained using the initial factory default settings for the training parameters. After training the following model statistics were reported.

NOX1 - Training matrix statistics based on 146 observations.

Measured	93.838493	27.536246	38.169998	181.63000	109945.50
Predicted	92.675019	26.058615	46.374672	171.66412	98462.453
Residual	1.163474	9.156508	-20.98502	45.171471	12157.038
R Square			0.889427		

The model was tested using the test matrix and the following model statistics were reported.

NOX1 - Test matrix

statistics based on 1194 observations.

Measured	91.109933	27.856727	32.879997	240.52002	925764.67
Predicted	90.027941	25.187365	50.670448	173.30337	756843.17
Residual	1.081992	10.132322	-54.08673	70.059113	122478.08
R Square			0.867701		

The model performance did not collapse on the test matrix indicating that the model is probably OK. The next step is to run a sensitivity analysis on the model to see if any input variables could be removed. The following table was generated using the **Sensitivity Report** command in the Model menu.

Sensitivity Analysis of NOX

Variable Name	Initial Setting	Percent Total
COALUPL	7.81	+0.15803
COALLOLV	7.12	+0.15003
BARKOFP	13.25	-0.12880
1LVATEMP	166.59	+0.08929
AMBAIR	107.04	-0.07536
3LVATEMP	157.04	+0.05892
COALMILV	5.31	+0.05778
STEAMFLO	404.94	+0.04864
OILMILV	16.14	-0.04636
OILUPLV	10.52	+0.03677
2LVATEMP	156.75	-0.03403
PAIRLOLV	18.78	+0.02443
STEAMTMP	1178.30	+0.02078
SECMILV	121.34	-0.01325
GASIGN	7.04	-0.01165
PAIRMILV	18.77	-0.01096
SECLOLV	106.02	+0.00982
SECUPLV	107.40	-0.00617
BARKUAIR	252.65	+0.00617
PAIRUPLV	22.26	-0.00480
BARKFEED	77.56	+0.00343
GASBURN	77.71	-0.00297
STEAMPR	1617.61	-0.00160

After reviewing the previous report, it was decided that variables that had less then a 2% effect on NOX should be eliminated. The following variables were eliminated: SECMILV, GASIGN ,PAIRMILV, SECLOLV, SECUPLV, BARKUAIR, PAIRUPLV, BARKFEED, GASBURN and STEAMPR. A new model (NOX2) was then created and trained using the paired down input list.

NOX2 - Training matrix statistics

based on 146 observations.

Measured	93.838493	27.536246	38.169998	181.63000	109945.50
Predicted	92.261639	26.254764	41.677437	174.91064	99950.333
Residual	1.576855	9.132643	-22.12422	43.526802	12093.750
R Square			0.890002		

The model was tested using the test matrix and the following model statistics were reported.

NOX2 - Test matrix statistics based on 1194 observations.

Measured	91.109933	27.856727	32.879997	240.52002	925764.67
Predicted	89.746692	25.100364	46.305305	178.31256	751623.71
Residual	1.363241	10.558678	-51.73450	65.622284	133002.42
R Square			0.856332		

Example: CLO2 Dataset

CLO2 Detailed Description
File Name - CLO2.RAW

Description: The CLO2 dataset was the result of an chemical experiment to find the best operating points for ACID, TEMP, H2O2 and NaClO3 to product ClO2.

Column Names	**Column Description**
ACID	Amount of acid used in the reaction
TEMP	The temperature of the reaction
H2O2	Amount of hydrogen peroxide used in the reaction
NACLO3	Amount of sodium chlorate used in the reaction
CLO2	Amount of chlorine dioxide produced
PROD	Amount of chlorine converted relative to total available

Data Analysis The basic statistics report follows:

Variable	N	Mean	Std Dev	Minimum	Maximum
ACID	30	12.000000	2.729153	6.000000	18.000000
TEMP	30	60.000000	9.097177	40.000000	80.000000
H2O2	30	1.980333	0.991966	0.140000	4.050000
NACLO3	30	56.000000	20.943273	20.000000	100.00000
CLO2	30	3.206000	2.199261	0.160000	8.370000
PROD	30	71.833333	28.118080	3.000000	100.00000

Model Building Build a model of CLO2 and PROD using the other variables as inputs. Export the models to an optimizer to find the maximum production.

Example: CLOSTAT1 Dataset

CLOSTAT1 Detailed Description
File Name - CLOSTAT1.RAW

Description: The CLOSTAT1 dataset was the result of an chemical simulation to find the best operating points for DIL, CONS and RECY. WAT, D0CS, COSW, SOL D1CW and D1CS are process streams resulting from the simulation.

Column Names	**Column Description**
DIL	Dilution
CONS	Consistancy
RECY	Recycle water
WAT	
D0CS	
COSW	
SOL	
D1CW	
D1CS	

Data Analysis The basic statistics report follows:

Variable	N	Mean	Std Dev	Minimum	Maximum
DIL	15	5.000000	0.590399	4.000000	6.000000
CONS	15	8.794607	3.232301	3.291297	14.284348
RECY	15	0.941179	0.100738	0.778182	1.105573
WAT	15	7019.7693	1613.4971	5327.4257	11238.688
D0CS	15	767.34628	311.13754	270.94604	1285.0131
COSW	15	313.60096	56.263328	211.49833	405.66909
SOL	15	0.080857	0.019873	0.047394	0.118405
D1CW	15	204.52833	4.186309	196.89950	211.28370
D1CS	15	808.47947	18.348460	775.10199	838.51586

Model Building Build a models of WAT, D0CS, COSW, SOL, D1CW and D1CS using the other variables as inputs. Export the models to an optimizer to find ???

Example: PEAK4 Dataset

PEAK4 Detailed Description
File Name - PEAK4.RAW

Description: Contains the results of stepping angles X and Y (11 steps) from 0 to   and evaluating
$Z = \sin(X) \sin(Y)$

Column Names	**Column Description**
X	The X variable
Y	The Y variable
Z	The result of the equation

Data Analysis The basic statistics report follows:

Variable	N	Mean	Std Dev	Minimum	Maximum
X	121	0.500000	0.317543	0.000000	1.000000
Y	121	0.500000	0.317543	0.000000	1.000000
Z	121	0.680000	0.200671	0.200000	1.000000

Model Building Build a model of Z using X and Y as inputs.

Example: CURL Dataset

CURL Detailed Description
File Name - CURL.RAW

Description: The was the result of a designed experiment to find which independent variables have the most effect on paper curl.

Column Names	**Column Description**
JET	Jet to wire ratio measurement
MOIST	Moisture measured on the paper machine
DD	Dryer differential measurement (between top and bottom of the sheet)
CDPOS	Position across the paper machine (physical)
FOT	Fiber orientation angle (lab)
SCURL	Simplex curl (lab)
DCURL	Duplex curl (lab)
RCURL	Reel curl (lab)
RMOIST	Reel moisture (lab)

Data Analysis The basic statistics report follows:

Variable	N	Mean	Std Dev	Minimum	Maximum
JET	70	26.001604	17.569578	6.442120	45.548595
MOIST	70	5.591592	0.587862	4.907064	6.282612
DD	70	16.901340	8.196261	7.749969	26.027664
CDPOS	70	16.428571	10.965785	1.000000	32.000000
FOT	70	4.145139	9.372166	-18.19136	20.694777
SCURL	70	-1.858674	21.378735	-55.25967	39.023815
DCURL	70	0.183420	17.469644	-39.00546	26.024122
RCURL	70	-4.502250	11.028441	-25.98776	19.509588
RMOIST	70	6.236317	0.506186	5.269414	6.997371

Model Building Build models of FOT, SCURL, DCURL and RCURL using JET, MOIST, DD and CDPOS as inputs. Try using RMOIST (lab moisture) in place of MOIST (on-line measurement).

Example: STR4 Dataset

STR4 Detailed Description
File Name - STR4.RAW

Description:	The STR4 dataset was captured during the normal operation of a paper machine. The intent of the data capture was to see if any of the standard logged process variables could be used to predict paper strength properties. This experiment is really a fishing expedition in that no designed experiment was performed on the process variables. However, there may be enough information in the log to point to variables that have a major effect.

Column Names	**Column Description**
KSOFT	Percent softwood pulp used in furnish
KHARD	Percent hardwood pulp used in furnish
KBROKE	Percent broke pulp used in furnish
KDEINK	Percent deinked pulp used in furnish
KGRDW	Percent groundwood pulp used in furnish
STARSLD	Starch solids
SPEED	Paper machine speed
HDBXPH	Head box pH
HDBXFREE	Head box freeness
HDBXCONS	Head box consistancy
SOFTCONS	Softwood consistancy
SOFTFREE	Softwood freeness
HARDCONS	Hardwood consistancy
HARDFREE	Hardwood freeness
SBSWGT	Supered basis weight
STAF	Supered TAF (strength test)
STEARMD	Supered MD tear (strength test)
STEARCD	Supered CD tear (strength test)
RAWSTOCK	Raw stock basis weight
REELMO	Reel moisture
UBSWGT	Un-supered basis weight
COUCH	Couch vacuum
REELASH	Reel ash
LABMO	Lab moisture

Data Analysis The basic statistics report follows:

Variable	N	Mean	Std Dev	Minimum	Maximum
KSOFT	1178	35.300509	5.127401	0.000000	41.000000
KHARD	1178	11.530560	14.534154	0.000000	47.000000
KBROKE	1178	30.334465	3.890627	10.000000	40.000000
KDEINK	1178	6.057725	4.218549	0.000000	15.000000
KGRDW	1178	16.782683	14.455638	0.000000	34.000000
STARSLD	265	1.216679	0.075545	0.900000	1.600000
SPEED	1178	2254.4295	100.00711	1845.0000	2313.0000
HDBXPH	1178	7.184550	0.133124	6.900000	7.400000
HDBXFREE	1178	144.56536	73.165909	54.000000	330.00000
HDBXCONS	1178	0.584888	0.044584	0.500000	0.740000
SOFTCONS	1176	3.716556	0.206896	2.980000	4.300000
SOFTFREE	1176	501.39881	34.988070	398.00000	635.00000
HARDCONS	491	3.878411	0.268816	3.360000	4.560000
HARDFREE	491	417.72301	33.024367	351.00000	483.00000

SBSWGT	642	44.566963	7.246427	36.830002	71.330002
STAF	157	35.529618	6.042086	20.400000	54.430000
STEARMD	340	22.358529	4.784218	14.600000	45.099998
STEARCD	340	26.862941	6.009229	19.400000	56.099998
RAWSTOCK	718	30.471086	4.376988	25.950001	56.759998
REELMO	741	3.851309	0.464810	2.280000	5.420000
UBSWGT	739	45.397253	6.907419	37.099998	70.580002
COUCH	240	6.915833	1.587740	4.000000	13.900000
REELASH	197	27.450254	2.671402	22.500000	34.500000
LABMO	228	4.524561	0.654777	2.400000	6.200000

Model Building Build a model of STAF and find the variables that most effect it.

Graph Options dialog box

The following options allow you to change the way the graphs are plotted.

2 D Options

X Grid

Draws lines on every major tic that span the entire Y axis.

Y Grid

Draws lines on every major tic that span the entire X axis.

Top Axis

Draws the X axis on the top of the graph.

Right Axis

Draws the Y axis on the right side of the graph.

No X Tic Labs

Doesn't draw tic labels on the X axis.

No Y Tic Labs

Doesn't draw tic labels on the Y axis.

3D Options

No Label

No labels or tics of the 3D graph

Frame

Draws a 3D box frame around the plot.

Fill Contour

Draw a color filled contour under the 3D surface map.

Line Contour

Draw a lined contour under the 3D surface map.

Color Wire

Draw a colored surface map.

Black Wire

Draw a black write surface map.

Fill Surface

Color Fill the surface of the 3D map.

Backfill

Back fill the 3D box with gray.

Hidden

Draw the 3D surface map using hidden line removal.

Char Size

The character size of all the labels in inches.

Line Width

The line widths used to draw the variables.

Axis Width

The line width used while drawing the axis.

 # Curves

The number of curves plotted on the XY Effect graph. (1-16)

 # Bars

The number of distribution bars plotted (1-250).

Run Ave

If other then zero, plot the running average (based on the average of N rows). Only the By Row and Residual graphs can draw the running average.

Z Offset

The distance the 3D surface map is raised above the X-Y plane.

Rotation

The amount the observed is rotate from the normal. (-180 to 180 degrees)

Elevation

The amount the observer is elevated above the X-Y plane. (-90 to 90)

Distance

The distance the observer is from the X-Y axis.

Rot Inc

Number of degrees to increment/decrement 3D graphs when pressing 3D rotation buttons.

 # Contours

The number of contour lines that will be drawn (1-16).

Swait

The distance to wait before drawing another line contour label.

Cosmax

The maximum amount of contour bending allowed for labeling.

Tolerance

The tolerance band displayed on the measured verses predicted graph. This is initialized to the value of the tolerance in the models training parameters.

Tol of Full Scale

The tolerance band is equally spaced from minimum to maximum. If this radio button is not checks then the tolerance band will converge when approaching zero.

Edit Variable Descriptors dialog box

The following options allow you to change/view all of the variables descriptors. You can edit the following: variable label, format, units, minimum, maximum, clip low and clip high values. The following fields can only be viewed: Usage, flag, scale, offset and ftype.

Label

Edit the variable's name.

Format

Edit the print format. Example: "5.3" indicates 5 characters for the total field length (including sign) and print 3 digits after the decimal point.

Units

Currently not used. Reserved for units of measure descriptions.

Minimum

The smallest value observed in the matrix.

Maximum

The largest value observed in the matrix.

Clip low

Ignore values below this value.

Clip high

Ignore values above this value.

Usage

The type of field: Number, String, Time, Input or Output.

Flag

Not currently used.

Scale

Scaling factor to normalize variable.

Offset

Offset used when scaling the variable.

Ftype

Field type. 0=float, 1=string, 2=timestamp, 3=recordtype, 4=timeonly and 5=dateonly. Only type 0 fields can be used in analysis or modeling. Type 1 is currently not used. Types 2,4 and 5 are used to time/date merge files. Type 3 indicates the first field in the data matrix.

Create Neural Model dialog box

The following options allow you to design a neural model.

Title

Enter the file name of the new model.

I/O list

Select from the list of all variables that are to be inputs and outputs for the model.

Input Add

Add the currently selected variables in the I/O list to the model list as inputs.

Output Add

Add the currently selected variables in the I/O list to the model list as outputs.

Network List

The current model design.

Remove

Remove rows selected from the current model design.

OK

Create the current model.

Cancel

Abort the operation without creation

Exclusions

Enter equations to exclude data from being loaded.

Create Sparse Matrix dialog box

The following options allow you to select the factors and responses from a historical data matrix.

Title

Enter the file name of the new matrix.

I/O list

Select from the list of all variables that are to be inputs and outputs for the design.

Input Add

Add the currently selected variables in the I/O list to the design list as inputs.

Output Add

Add the currently selected variables in the I/O list to the design list as outputs.

Network List

The current design.

Remove

Remove rows selected from the current design.

OK

Create the current design.

Cancel

Abort the operation without creation

Exclusions

Not currently used.

Data Import dialog box

The options are.

Title

The filename of the new data matrix

Separator

The most prevalent field separator. Valid separators are blank, tab and comma. Manually changing this control may affect the number of fields read.

# of Columns

The Maximum number of columns with this separator

Field #

The field # of the displayed field type

Field Type

The field type of the current field number. Valid field types are FLOAT, TIME or DATE. They are automatically determined (auto field types) during the pre-scan but can be overridden by selecting another type using this control.

# of Records

Not used

File Name

The name of the ASCII file to be converted.

Screening Grid

A demonstration of how the data will convert only the first 12 records can be shown. The user can change the record type to force records to be excluded. Only the record type can be changed. Use only L for label record, U for unit record or * to exclude the record.

Scan

Rescans the file to show how the file will convert.

Process

Create the data matrix module.

Cancel

Abort the operation.

Interrogate Model dialog box

The interrogate model dialog box allows you test the predictions of a neural model by entering the desired input values.

Input Grid

The following columns are displayed for each input variable to the model.

Input

Variable name

Value

The current value

Minimum

The minimum that can be set

Maximum

The maximum that can be set

Output Grid

The following columns are displayed for each output variable from the model.

Output

The Variable name

Value

The current prediction

Control Buttons

Sensrpt

Generate a sensitivity report at the current operating point

Finished

Done Interrogating the model

Update

Update the model's prediction

Edit Training Parameters dialog box

The following options allow you to select the parameters for training the neural model.

Max Hidden

The maximum number of hidden neurons that can be added to the model during training when Automatic Increment training algorithm is selected. Otherwise the total number of neurons in the hidden layer.

Eon

Number of presentations of the training set to train before checking the statistics or updating the training progress graph. The Automatic Increment is very sensitive to this value.

Max Training

Maximum number of the times the training matrix can be presented before ending the training.

Hidden Freeze

The amount to decrease the hidden neuron's learning rate when adding a new neuron in the Automatic Increment training algorithm.

Error Tolerance

The tolerance band that all predictions must be within (total Sq. Error) to end the training.

Good RSQ

What the user considers a good enough R square stop training this model.

Sign Inc

Use in calculating when to add a new neuron (Automatic Increment).

No Sign Inc

Used in calculating when to add a new neuron (Automatic Increment)

Tolerance

Acceptable error, used in graphs and in calculating the number of points above and below (training graph) and by the Automatic Increment algorithm.

Learning Rate

The initial learning rate for the hidden layer to output layer connections.

HLearning Rate

The initial learning rate for the input layer to hidden layer connections.

TLearning Rate

The initial learning rate for the all threshold connections.

IO Learning Rate

The learning rate for the direct input layer to output layer connections.

Alpha

The momentum term.

Theta

The value feeding all threshold inputs. (default is set to 1.0)

Random Fact

The scaling factor used when initializing the model weights.

InRandom Fact

The scaling factor of Gaussian noise used when training the model.

Auto Save

Number of presentations of the training between auto saving the weights.

Seed

The seed value to initialize the random number generator.

Training Type - Standard BEP

Train using the standard BEP algorithm.

Training Type - Equal Spaced Increment

Add a new hidden neuron at an equally spaced intervals.

Training Type - Manual Increment

Add a new neuron to the hidden layer upon command by the user.

Training Type - Automatic Increment

Auto detect when to add a neuron to the hidden layer.

Stop Training On - Tolerance

Stop training when all predictions are within the tolerance band.

Stop Training On - Error Tolerance

Stop training when the total error is below the value specified.

Stop Training On - Good R Square

Stop when a good enough R square is reached.

Connect Inputs to Outputs

Connect the input neurons directly to the output neurons. This option will speed convergence if the relationships are simple.

CG Optimization after Eon

Perform conjugate gradient optimization on the weight matrix after each Eon. CG will not start until at least 500 presentations of the training matrix has been completed. This option will slow down the responses to window events (i.e. mouse) if the training matrix is large.

Set Default

Save the current settings as the default values.

Get Default

Reset the current values to the default.

OK

Done editing parameters.

Cancel

Abort without making any changes.

Create Design dialog box

The following options allow you to select the type of designed data matrix module created.

Title

Enter filename of the new data matrix.

Factors (Columns)

If the design type is **No Design** then enter the total number of columns desired in the data matrix. Otherwise enter the total of factors in the design.

Responses (Rows)

If the design type is **No Design** then enter the total number of rows desired in the data matrix. Otherwise enter the number of responses in the design.

Center Points

Enter the number of center points in the design.

Design Type

Enter the type of design The types available to choose from are: No Design, Simplex, Star-Simplex, Two-level, Three-level, Multi-level and Central Composite.

Scale Factor

Enter the scaling factor to be used when calculating the factor points in a user scaled central composite design.

# of Runs

This field displays the number of runs (experiments) needed to completely fill the designed data matrix.

Maximum # of Factors

This fields displays the maximum number of factors that the software can support for the selected type of design.

Don't Create Data

Don't create the data in the data matrix. Generally the factors in the matrix is filled with appropriate levels and the responses are set to missing. However, if this option is selected then the matrix will be empty. Later the empty matrix can be loaded using the sparse data loader.

Standard Composite

Standard rotatable design.

Face Centered

Set the levels of the star points so that they fall on the face of the hypercube.

User Scaled

Scale the factor points so that are (user scale) percent of the star points.

Variable Descriptor Grid

You can conveniently change the variable(s) minimum, maximum, labels and the number of levels (if a multilevel design was selected).

Variable	The Variable's name.
Minimum	The Minimum the variable will reach during the course of the experiment.
Maximum	The maximum the variable will reach during the course of the experiment.
Levels	The number of levels that are to be run during the course of a multilevel experiment.

OK

Choose this button to create the designed data matrix.

Cancel

Exit the dialog without creating a designed data matrix.

Edit Equation dialog box

The following options allow you to edit and check the syntax of a calculated column and data exclusion equations:

Equation Edit Window

Enter the equations in algebraic format in this window.

Parse

Press to check for syntax errors before leaving dialog.

OK

Exits the dialog and saves the changes made.

Cancel

Exits the dialog without saving changes.

Y Graph Selection dialog box

The following options allow you to select a variable to be plotted:

Y Variable List

Select one variable from the list to plot on the Y-axis. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

OK

After selecting a variable press this button to generate the graph.

Cancel

Exit the dialog without generating a graph.

3D Surface Graph Selection dialog box

The following options allow you to select the variables to be plotted:

X Variable List

Select one input variable from the list to plot on the X-axis. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Y Variable List

Select one input variable from the list to plot on the Y-axis. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Z Variable List

Select one output variable from the list to plot on the Z-axis. This variable is drawn as a surface map and (if selected) a contour map under the fishnet. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Other Variables List

Input variables not selected in the X and Y lists are set equal to the values displayed in this list box. To change the value of these variables select one in this list and edit the value in the window below this list.

Frame

Draws a 3D box frame around the plot.

No Label

Prevents the drawing of axis labels and tic marks

Fill Contour

Draw a color filled contour under the 3D surface map.

Line Contour

Draw a lined contour under the 3D surface map.

Hidden

Draw the 3D surface map using hidden line removal.

Backfill

Back fill the 3D box with gray.

Color Wire

Draw a colored surface map.

Black Wire

Draw a black write surface map.

Fill Surface

Color Fill the surface of the 3D map.

Rotation

The amount the observed is rotate from the normal. (-180 to 180 degrees.

Elevation

The amount the observer is elevated above the X-Y plane. (-90 to 90)

Distance

The distance the observer is from the X-Y axis.

Levels

The number of contour lines that will be drawn (1-16).

OK

After selecting the X, Y and Z variables press this button to generate the graph.

Cancel

Exit the dialog without generating a graph.

3D Scatter Graph Selection dialog box

The following options allow you to select the variables to be plotted:

X Variable List

Select one variable from the list to plot on the X-axis

Y Variable List

Select one variable from the list to plot on the Y-axis.

Z Variable List

Select one variable from the list to plot on the Z-axis.

OK

After selecting the variables press this button to generate the graph.

Cancel

Exit the dialog without generating a graph.

Contour Graph Selection dialog box

The following options allow you to select the variables to be plotted:

X Variable List

Select one input variable from the list to plot on the X-axis. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Y Variable List

Select one input variable from the list to plot on the Y-axis. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Z Variable List

Select one output variable from the list to use as the affect variable. This variable is incremented to generate the contours. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Other Variables List

Input variables not selected in the X and Y lists are set equal to the values displayed in this list box. To change the value of these variables select one in this list and edit the value in the window below this list.

Levels

Enter the number of levels to be plotted when the graph is drawn.

Fill Contour

Select to generate a filled contour rather than a line contour.

OK

After selecting the X, Y and Z variables press this button to generate the graph.

Cancel

Exit the dialog without generating a graph.

Distribution Graph Selection dialog box

The following options allow you to select the variable to be plotted:

Variable List

Select one variable from the list to plot as a distribution. On model graphs a prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

of Boxes

Enter the number of boxes to be used in calculating the distribution (1-250).

OK

After selecting the variable press this button to generate the graph.

Cancel

Exit the dialog without generating a graph.

Graph Selection dialog box

The following options are used to select the variables plotted by the Measured vs. Predicted, Measured & Predicted and Residuals graphs.

The following options allow you to select the variable to be plotted:

Variable List

Select one output variable from the list to plot as a distribution.

OK

After selecting the variable press this button to generate the graph.

Cancel

Exit the dialog without generating a graph.

2D Scatter Graph Selection dialog box

The following options allow you to select the variables to be plotted:

X Variable List

Select one variable from the list to plot on the X-axis. On model graphs a prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Y Variable List

Select one variable from the list to plot on the Y-axis. On model graphs a prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

OK

After selecting the X and Y variables press this button to generate the graph.

Cancel

Exit the dialog without generating a graph.

XY Graph Selection dialog box

The following options allow you to select the variables to be plotted:

X Variable List

Select one input variable from the list to plot on the X-axis. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Y Variable List

Select one output variable from the list to plot on the Y-axis. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Other Variables List

Input variables not selected in the X and Y lists are set equal to the values displayed in this list box. To change the value of these variables select one in this list and edit the value in the window below this list.

Show 1st Derivative

Display the 1st derivative curve (RED line) if the radio button is checked.

Show 2nd Derivative

Display the 2nd derivative curve (GREEN line) if the radio button is checked.

OK

After selecting the X and Y variables press this button to generate the graph.

Cancel

Exit the dialog without generating a graph.

XY Effect Graph Selection dialog box

The following options allow you to select the variables to be plotted:

X Variable List

Select one input variable from the list to plot on the X-axis. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Y Variable List

Select one output variable from the list to plot on the Y-axis. A prefix has been added to each variable name in the list to indicate the type of variable (i.e. Input, Output, Predicted or Residual).

Effect Variable List

Select one input variable from the list to use as the effect variable. This variable is incremented to generate the family of curves.

Other Variables List

Input variables not selected in the X and Y lists are set equal to the values displayed in this list box. To change the value of these variables select one in this list and edit the value in the window below this list.

# of Curves

Enter the number of curves to be plotted when the graph is drawn.

OK

After selecting the X and Y variables press this button to generate the graph.

Cancel

Exit the dialog without generating a graph.

Find Equation dialog box

The following options allow you to specify what to find:

Variable

The column or variable to search.

Operator

Select a type of arithmetic operation.

Value

Select a search value.

OK

Begin searching.

Cancel

Exit the dialog without searching.

Sparse Data Loader Options dialog box

The following options can be used when loading historical data into a designed matrix:

Auto Rescale

Check the maximums and minimums values in the current input matrix and automatically rescale the designed matrix to fit. This may change the goodness of fit for the previously loading into the designed matrix (if any).

Keep Rejects for Testing

Keep the rejected data for a test matrix. Data that doesnt meet the criteria for loading into the designed matrix is kept in a reject file. This file is then loaded at the end of the designed matrix and each row is marked as a test row.

First Load Data in Current DM

Load the data in the current data matrix first then load the source data matrix data. This will keep the current data if it is better then the data being loaded.

Advisor

After loading the historical data a search is made in the designed matrix for missing points. A missing data matrix is constructed to suggest experimental runs. These missing points can be loaded into the design with a reject flag insuring that the advisor will never ask again for the same point.

Auto Set Format

Set the format string of the current data matrix equal to the value in the source matrix.

Skip Center

Dont suggest any center points when using the advisor option.

Skip Star

Dont suggest any star points when using the advisor option.

Skip Factor

Dont suggest any factorial points when using the advisor option.

Skip Multilevel

Dont suggest any multilevel points when using the advisor option.

Max Number of Suggestions

The maximum number of suggestions to add to the MISSING data matrix module when using the advisor option.

OK

Select to load the data.

Cancel

Exit without loading.

Select Equation dialog box

The following options are available when appending a calculated column:

Equation List

Select the equation to be used for creation of th new calculated column.

OK

After selecting an equation press this button to append the new column.

Cancel

Exit the dialog without adding a column.

Enter Number dialog box

This dialog box will pop up to allow you to enter a single number. It is used in various routines, such as, **Goto Record Number** or **Make Testing Data**.

Print Options dialog box

Parameters used when printing the data matrix.

Title

Print the title of the data matrix in the page header section.

Date

Print the current date in the page header section.

Page Numbers

Print the page number in the page footer section.

Vertical Lines

Draw vertical lines separating the columns.

Horizontal Lines

Draw horizontal lines between the rows.

Print Random

Print the rows randomly. Used with a designed experiment.

Best Model Search dialog box

The best model search algorithm searches the inputs for the model that best predicts the selected output(s). The routine builds a model one input at a time in the same way a statistician might build a regression model. The routine may take many passes through the input variable list, each time building separate models and picking the model that best explains the variability.

The results of this function can be one (or more) BEP model(s) stored in the sub-directory where the parent data matrix is located. The models are named bestNN.bep. Where: NN is the column number of the selected output variable.

Input Output Selector Grid

The selector grid is where the user tells the search routine which variables are inputs, outputs and which are to be excluded. If a variable is an output type the letter O in the Use column. If the variable is not to be used type an X. If the Use the letter I then the variable must always be included as an input to the model. A blank in the Use column indicated that the variable may be included as an input.

Last Model Statistics

Displays the last best model built.

Last Model Built

This field contains the I/O configuration of the last best model built.

Last R Sq.

This field contains the R square statistic for the last best model.

Number of Inputs

This field contains the number of inputs included in the last model.

Status

This field displays the current state of the search engine.

Sub-Status

This field displays the current cycle through the input search space.

Max Training

Enter the number of passes through the training matrix to be completed before the model fit is checked. In the current version this number should be kept low (i.e. 100-200) because there is no feedback during the model building and training cycles. You cannot switch to other window programs during the execution of this function. This will be fixed in the next version.

Hidden

The number of hidden layer neurons to use for the models.

Min Increase

The minimum increase in the R sq. value to be achieved in order to continue adding input variables to the model.

Max # of Inputs

The maximum number of inputs to be included in the model.

Number of Models

The number of models that will be checked to see which is the best.

Completed

The number of models checked so far.

Estimated Time

The estimated time to complete checking all models.

Separate Models

If more that one output variable is selected and this radio button is checked then test build each output as a separate model. If unchecked all outputs are modeled at the same time.

Use R sq from Text Matrix

Use the R square statistic from the test matrix rather than the training matrix to determine goodness-of-fix.

Use Combined R sq

Use the combined R square of both the test and training matrix to determine best model. In this case the two R squares are averaged.

Use Sparse Load

Use the sparse data loaded with constructing the training matrix. This will attempt to load a designed experiment (if the data allows) matrix. This may greatly decrease the searching time for VERY large data sets.

Un-Designed Data Matrix

An undesigned data matrix is a data matrix that was created without an statistical designed experiment. The data has been entered or loaded without regard to which columns are factors (independent variables) or responses (dependent variables). This is usually the case when the source of the data is a historical database or process log. All imported data points are loaded into an undesigned matrix.

Designed Data Matrix

A designed data matrix is a data matrix that was created using the a statistically designed experiment as the organization of the data.

Factors

In an experiment, factors are the variables that cause an effects in another variables. When there are more that one factors in an experiment, each run, step or treatment is a composite of particular levels of each factor.

Responses

In an experiment, responses are the variables that factors influence.

Design of Experiments

An experiment when you do something to a process or an object to discover a response of one variable to alterations in other variables. As opposed to an observational study were you would passively observe the process and measure variables of interest. The advantages of experimentation over the passive method is that an experiment enables you to monitor the effects of the specific preparation. Furthermore, you can manipulate the conditions of the experiment to keep constant factors that are of no interest. A well designed experiment allows you to discover the effects of one variable on another. Another less apparent benefit of a designed experiment is that you can effectively study the combined effects of many factors at the same time. The interaction of several factors can result in effects that can not be demonstrated by studying each variable separately.

Print Data Matrix Randomly

To print the data matrix in a random row order select the **Print Random** option in the Print Options dialog in the File menu. To aid you in entering the data after the measurements are taken the original row number is printed in the first column along with the type of observation (i.e. Factor, Star, Center, simpleX, or Multilevel).

Field types automatically determined by Import

The following field types and formats can be automatically determined by the import file function.

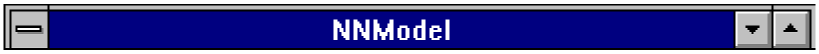
Type	Format
FLOAT	Standard formats (i.e. 1.23 -1.23 1.23E+00)
TIME	HH:MM or HH:MM:SS
DATE	YYYYMMDD

Note: Time and Date field types should be used only for file merging options, not for data analysis or modeling.

Common Data Errors

Errors in magnitude
Pinning
Noise
Trends
Sudden rises or drops in level
Pocketing or too digital looking
No signal
Gaps in data
Repeating patterns
Fudging

Title Bar



The title bar is located along the top of a window. It contains the name of the application and module.

To move the window, drag the title bar. Note: You can also move dialog boxes by dragging their title bars.

A title bar may contain the following elements:

- Application Control-menu button
- Document Control-menu button
- Maximize button
- Minimize button
- Name of the application
- Name of the module
- Restore button

Scroll bars

Displayed at the right and bottom edges of the module window. The scroll boxes inside the scroll bars indicate your vertical and horizontal location in the module. You can use the mouse to scroll to other parts of the module.

Size command (System menu)

Use this command to display a four-headed arrow so you can size the active window with the arrow keys.

After the pointer changes to the four-headed arrow:

1. Press one of the DIRECTION keys (left, right, up, or down arrow key) to move the pointer to the border you want to move.
2. Press a DIRECTION key to move the border.
3. Press ENTER when the window is the size you want.

Note: This command is unavailable if you maximize the window.

Shortcut

Mouse: Drag the size bars at the corners or edges of the window.

Move command (Control menu)

Use this command to display a four-headed arrow so you can move the active window or dialog box with the arrow keys.

Note: This command is unavailable if you maximize the window.

Shortcut

Keys: CTRL+F7

Minimize command (application Control menu)

Use this command to reduce the NNMODEL window to an icon.

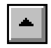
Shortcut

Mouse: Click the minimize icon  on the title bar.
Keys: ALT+F9

Maximize command (System menu)

Use this command to enlarge the active window to fill the available space.

Shortcut

Mouse: Click the maximize icon  on the title bar; or double-click the title bar.
Keys: CTRL+F10 enlarges a module window.

Next Window command (module Control menu)

Use this command to switch to the next open module window. NNMODEL determines which window is next according to the order in which you opened the windows.

Shortcut

Keys: CTRL+F6

Previous Window command (module Control menu)

Use this command to switch to the previous open module window. NNMODEL determines which window is previous according to the order in which you opened the windows.

Shortcut

Keys: SHIFT+CTRL+F6

Close command (Control menus)

Use this command to close the active window or dialog box.

Double-clicking a Control-menu box is the same as choosing the Close command.

Note: If you have multiple windows open for a single module, the Close command on the module Control menu closes only one window at a time. You can close all windows at once with the Close command on the File menu.

Shortcuts

Keys: CTRL+F4 closes a module window
ALT+F4 closes the window or dialog box

Restore command (Control menu)

Use this command to return the active window to its size and position before you chose the Maximize or Minimize command.

Switch to command (application Control menu)

Use this command to display a list of all open applications. Use this "Task List" to switch to or close an application on the list.

Shortcut

Keys: CTRL+ESC

Dialog Box Options

When you choose the Switch To command, you will be presented with a dialog box with the following options:

Task List

Select the application you want to switch to or close.

Switch To

Makes the selected application active.

End Task

Closes the selected application.

Cancel

Closes the Task List box.

Cascade

Arranges open applications so they overlap and you can see each title bar. This option does not affect applications reduced to icons.

Tile

Arranges open applications into windows that do not overlap. This option does not affect applications reduced to icons.

Arrange Icons

Arranges the icons of all minimized applications across the bottom of the screen.

Rotate 3D Graph Right

Use this command to rotate the 3D surface or scatter plot right by **Rot Inc** degrees. The rotate increment variable can be set in the graphs options dialog box (default 10).

Rotate 3D Graph Left

Use this command to rotate the 3D surface or scatter plot left by **Rot Inc** degrees. The rotate increment variable can be set in the graphs options dialog box (default 10).

Rotate 3D Graph Up

Use this command to rotate the 3D surface or scatter plot up by **Rot Inc** degrees. The rotate increment variable can be set in the graphs options dialog box (default 10).

Rotate 3D Graph Down

Use this command to rotate the 3D surface or scatter plot down by **Rot Inc** degrees. The rotate increment variable can be set in the graphs options dialog box (default 10).

No Help Available

No help is available for this area of the window.

No Help Available

No help is available for this message box.

File New dialog box

Specify the type of module you wish to create:

File Extension	Document Type
BEP	Neural Model Document
DM	Data Matrix Document
GRF	Graph Document
RPT	Report Document

File Open dialog box

The following options allow you to specify which file to open:

File Name

Type or select the filename you want to open. This box lists files with the extension you select in the List Files of Type box.

List Files of Type

Select the type of file you want to open:

BEP	Neural Model Document
DM	Data Matrix Document
GRF	Graph Document
RPT	Report Document

Drives

Select the drive in which NNMODEL stores the file that you want to open.

Directories

Select the directory in which NNMODEL stores the file that you want to open.

Network...

Choose this button to connect to a network location, assigning it a new drive letter.

File Save As dialog box

The following options allow you to specify the name and location of the file you're about to save:

File Name
Type a new filename to save a module with a different name. A filename can contain up to eight characters and an extension of up to three characters. NNMODEL adds the extension you specify in the Save File As Type box.

Drives
Select the drive in which you want to store the module.

Directories
Select the directory in which you want to store the module.

Network...
Choose this button to connect to a network location, assigning it a new drive letter.

Print dialog box

The following options allow you to specify how the module should be printed:

Printer

This is the active printer and printer connection. Choose the Setup option to change the printer and printer connection.

Setup

Displays a Print Setup dialog box, so you can select a printer and printer connection.

Print Range

Specify the pages you want to print:

All	Prints the entire module.
Selection	Prints the currently selected text.
Pages	Prints the range of pages you specify in the From and To boxes.

Copies

Specify the number of copies you want to print for the above page range.

Collate Copies

Prints copies in page number order, instead of separated multiple copies of each page.

Print Quality

Select the quality of the printing. Generally, lower quality printing takes less time to produce.

Print Progress Dialog

The Printing dialog box is shown during the time that NNMODEL is sending output to the printer. The page number indicates the progress of the printing.

To abort printing, choose Cancel.

Print Setup dialog box

The following options allow you to select the destination printer and its connection.

Printer
Select the printer you want to use. Choose the Default Printer; or choose the Specific Printer option and select one of the current installed printers shown in the box. You install printers and configure ports using the Windows Control Panel.

Orientation
Choose Portrait or Landscape.

Paper Size
Select the size of paper that the module is to be printed on.

Paper Source
Some printers offer multiple trays for different paper sources. Specify the tray here.

Options
Displays a dialog box where you can make additional choices about printing, specific to the type of printer you have selected.

Network...
Choose this button to connect to a network location, assigning it a new drive letter.

Print Preview toolbar

The print preview toolbar offers you the following options:

Print
Bring up the print dialog box, to start a print job.

Next Page
Preview the next printed page.

Prev Page
Preview the previous printed page.

One Page / Two Page
Preview one or two printed pages at a time.

Zoom In
Take a closer look at the printed page.

Zoom Out
Take a larger look at the printed page.

Close
Return from print preview to the editing window.

The NNMODEL application uses a number of dialog boxes.