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.
New Test Matrix	Deletes the current test matrix and replaces it with the selected data matrix..
Append Test Matrix	Appends the selected data matrix to the existing test matrix.
New Training Matrix	Deletes the current training matrix and replaces it with the selected data matrix..
Append Training Matrix	Appends the selected data matrix to the existing training matrix.
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.
Export Training Matrix	Export the training matrix as an external ASCII file.
Export Test Matrix	Export the 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.
Remove Inputs	Remove inputs from a neural network. Using this will require the network to be re-trained.

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.
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.
CG Tweak	Run conjugate gradient training once to fine tune the back propagation training.
Auto Save	Toggle auto saving of the model during training.

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.
Append Data Matrix	Append two data matrices.
Reserve Testing Data	Reserve part of the current data matrix for the later testing of a neural model.
Fill Missing Data	
Keep Last Value	Fills in missing values using the previous non-missing value found in the column.
Interpolate	Fills in missing values with the linear interpolation of the two adjacent values.
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.
Model Process Data With Time Lags

Reference

Dialog Boxes

Graph Options
Edit Variable Descriptors
Create Neural Model
Create Sparse Matrix
Data Import
Interrogate Model
Edit Training Parameters
Create Design
Edit Equation
Y Graph Selection
3D Surface Graph Selection
3D Scatter Graph Selection
Contour Graph Selection
Distribution Graph Selection
Graph Selection
2D Scatter Graph Selection
XY Graph Selection
XY Effect Graph Selection
Find Equation
Sparse Data Loader Options
Select Equation
Enter Number
Print Options
Best Model Search

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.

NNMODEL provides a powerful method of interrogating pre-built neural models via Windows™ DDE facility. Using the DDE interface you could write a Visual Basic program that collects data from an A/D board and writes it to a neural model. Then the neural model could automatically update hot links in an Excel spreadsheet displaying the models predictions.

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.

New Features

The following features were added to NNMODEL version 1.4

New Append Data Matrix - this function was added to facilitate appending new data records to an existing data matrix. If you find that you are getting additional data via some electronic source and it needs to be appended routinely to a master matrix and its a pain to get the variable labels into your raw ASCII file then this function can make life a little easier. To append this data, first import it into a new data matrix, DO NOT import or edit the variable labels (use the default labels) then open the master data matrix and select Append Data Matrix command. If the two matrices have exactly the same number of columns then the data is quickly added to the end of the master data matrix.

New Best Model Search Dialog - there has been a new button added to the search dialog to allow you to edit the neural parameters without exiting the search routine. The EP Button will invoke the 'Edit Parameters' dialog to allow you to make any last minute changes to the neural parameters before starting the search.

New Network Option - Circular Back Propagation options have been added to the Edit Parameters dialog. What is circular back-prop? Basically, weve added another Theta-like input to each neuron. These inputs are fed the sum of the squared values of the network inputs. CBP can decrease the training time and the network complexity when modeling some types of processes. Try these options on the VEL example in the TESTSETS sub-directory.

New Network Option - zero hidden layer neurons. This effectively removes the hidden layer from the network architecture. If youre looking for simple linear relationships this can be very fast, especially if youre using the Best Model Search routine for discovering unknown relationships in historical data.

New Training Option - keep best model during training. Some times the best model of a particular process develops somewhere between the first few seconds of training and the maximum epoch allowed. To capture this best model can be time consuming and frustrating. NNMODEL now has options to keep this intermediate model developed during the training session as the final model. How do we measure best? NNMODEL allows you to select either the mean square error or R square as the measurement. You can also select the source of the measurement as being calculated from the training matrix, the test matrix or the average of both.

New Training Option - auto save model every 10 minutes during training session. If the Auto Save Model menu item is checked then the current state of the model is automatically saved every 10 minutes or every epoch (depending on which is longer).

New Import Function - Replace test matrix. This function allows you to completely replace the test matrix. However, the importer will reject any records that are outside of the observed range of the initial training matrix.

New Import Function - Append training matrix. This function allows you to append new data to the existing test matrix. As with the previous function, the importer will reject any records that are outside of the observed range of the initial training matrix.

New Import Function - Replace training matrix. this function will allow you to completely replace the training matrix. The importer will reject any records that are outside of the observed range of the initial training matrix.

Modified / New Export Functions - Export training or test matrices. This function was been re-written so that either the training or the test matrices could be written (in ASCII format) to a file separately.

New Button - Stop Training. A new tool button has been added to the toolbar.

The button with the X over the train will now stop the current training session/
New Button - CG Tweak. A new tool button has been added to the toolbar. The button with the CG will run one iteration of the conjugate gradient weight optimization routine. This may be useful in training time series data to remove the bias that develops during back error propagation.
New Graph Options - added standard deviation lines. There has been three option buttons added to the Graph Options dialog that will plot either 1, 2 or 3 standard deviation lines on the Measured vs. Predicted graph, Measured and Predicted or Residuals graphs.
New Graph Option - added linear regression line to the Measured vs. Predicted graph.

The following features were added to NNMODEL version 1.3

1024 Columns - The 187 Column Bug has finally been fixed. The problem stemmed from a vendor supplied grid library. This library was replaced in the "Import Raw Data" dialog with another vendor's grid. This necessitated the adding of yet another DLL file to the project directory. In version 1.30 of NNMODEL the number of columns that can be created has been raised to 1024.
Append to the Test Matrix - A menu new command "File/Import/Append Test Matrix". has been added to let you add more data to your test matrix.
New Fill Missing Function - A new menu command "Data/Fill Missing/Interpolate" has been added. Previously the "Fill Missing" command only allowed you to fill the missing data with the last valid value. Now a second option allows you fill the missing values with a linear interpolated value.
Enhanced Time Lag Function - A new parameter "TimeS" has been added to the data variable descriptors. TimeS can be used to specify that when building a training matrix that this variable should be shifted back by the number of rows specified. For example, if each row represents a 10 minute scan then a TimeS of 12 will cause the training matrix loaded to include the value 120 minutes in the past from the modeled output. When building neural models the outputs will always be set to zero (in this version).
Logging In Best Model Search - The "Best Model Search" function now logs all model construction to the file nnmodel.log. This file is erased when NNMODEL is first loaded and usually contains only error conditions. The log can be viewed to see the order that inputs were included into the best model and various temporary model R squares.
Start/Stop Button - Added Start / Stop functionality to the Best Model Search. This allows you to stop a search and modify a parameter without having to re-enter the I/O grid.
Remove Inputs From Model - A new menu command "Edit/Remove Inputs" has been added. This command allows you to remove unnecessary inputs from a neural model. Many times in data mining you will add all inputs from a process and build a model then run a sensitivity analysis on those inputs (to eliminate unneeded inputs). Before this command you would have to go back to the original data matrix and build a new network. Now you can just remove the unwanted inputs. Of course you will still have to re-train the network.
Sensitivity Report Was Re-written - The sensitivity report has been completely rewritten. The sensitivity is calculated by summing the changes in the output variables caused by moving the input variables by a small amount over the entire training set. There are

three variables accumulated during the calculation. The AbsAve Sensitivity variable is the average of the absolute values of the change in the output. This value is then divided by the total amount of change for all input variables to normalize the values. The Ave Sensitivity is calculated the same as the AbsAve variable except the absolute values are not taken. If the direction of the change in the output variable is always the same then the Ave and AbsAve sensitivities will be identical. The third variable calculated is the peak sensitivity and the row in the training matrix that it occurred.

Additional Information In The Model View - The internal weights of the created model are displayed below the standard summary information. In addition, this view can now be copied to clipboard for use word processors.

Simplified Training Graph - The training progress graph has been simplified. When training a model using the standard BEP routines (without Automatic Hidden Neuron Addition) the training graph will show only the normalized sum square error of the training matrix (black) and the test matrix (red).

Additional Training Method - A conjugate gradient training method has been added. To use this method select "Conjugate Gradient" as the "Training Method" in the "Edit/Parameters" dialog screen. CG training may converge faster on large training matrices.

New Button In Create Model Dialog - A new button was added to the "Create Neural Model" dialog. The button allows you to add variables as both inputs and outputs at the same time. This can be used for creating auto-associative networks that predict the inputs from themselves. This is the first step in creating a sensor validation network.

Building a Model

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.

New Data Matrix command

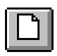
File Menu

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

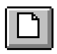
New Neural Model command

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

File Menu

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.

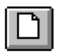
New command

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

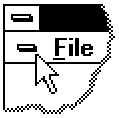
Toolbar: 
Keys: CTRL+O

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:

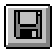


Save command

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 New 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. Also, all data imported ***MUST*** be within the initial observed range (min > new data < max). The imported data can be rejected for 4 reasons. 1) invalid range, 2) exclusions, 3) missing values and non-data record types.

Import Append Test Matrix command

File Menu

Use this command to import more test data into the neural model. It will append the new data to the current test matrix. All fields that are present in the neural model must also be present in the selected data matrix. Also, all data imported ***MUST*** be within the initial observed range (min > new data < max). The imported data can be rejected for 4 reasons. 1) invalid range, 2) exclusions, 3) missing values and non-data record types.

Import New Training Matrix command

File Menu

Use this command to import a new training matrix into the neural model. It will delete the current training matrix and reload the training matrix from a selected data matrix. All fields that are present in the neural model must also be present in the selected data matrix. Also, all data imported ***MUST*** be within the initial observed range (min > new data < max). The imported data can be rejected for 4 reasons. 1) invalid range, 2) exclusions, 3) missing values and non-data record types.

Import Append Training Matrix command

File Menu

Use this command to import more training data into the neural model. It will append the new data to the current training matrix. All fields that are present in the neural model must also be present in the selected data matrix. Also, all data imported ***MUST*** be within the initial observed range (min > new data < max). The imported data can be rejected for 4 reasons. 1) invalid range, 2) exclusions, 3) missing values and non-data record types.

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

File Menu

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 Training Matrix command

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

Export Test Matrix command

Use this command to write the test matrix 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

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

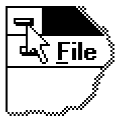
Exit command

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.

Remove Inputs

Edit Menu

Use this command to remove inputs from a neural network. This is a handy command if you normally include more inputs into your network then is really needed and then remove them after reviewing the sensitivity analysis.

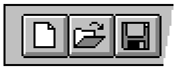
Toolbar command

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.

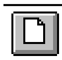
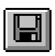
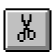
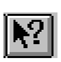
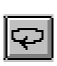
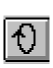
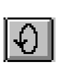
See Toolbar for help on using the toolbar.

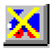
Toolbar

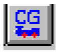


The toolbar is displayed across the top of the application window, below the menu bar.   The toolbar provides quick mouse access to many tools used in NNMODEL,

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.
	Initializes and starts training the neural network.

 Stops training the neural network.



Executes conjugate gradient training once on the neural network.

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

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 Hidden Freeze, Signif Increase and No Signif Increase might have to be tuned.

Standard Training Graph (BEP and Conjugate Gradient)

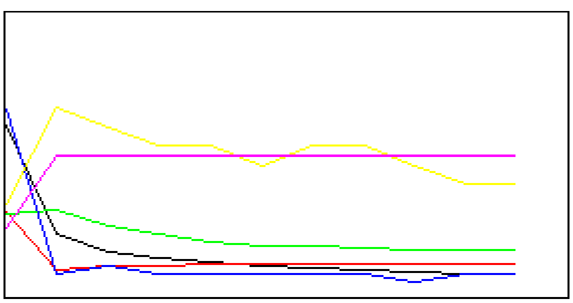
After each eon the training status graph is updated. The status graph displays two internal factors to demonstrate how well the training is proceeding. The following statistics are plotted.

Black	Sum squared error of the training matrix
Red	Sum squared error of the test matrix (if present)

Standard Training Graph with Automatic Hidden Neuron Addition

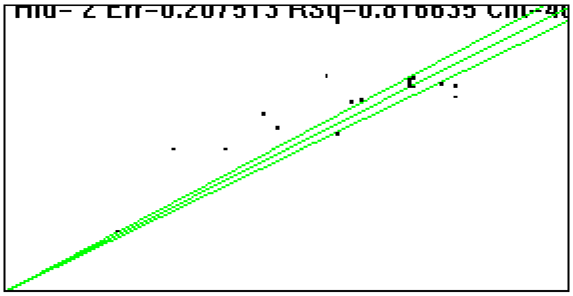
After each eon the training status graph is updated. The status graph displays six internal factors to demonstrate how well the training is proceeding. The following statistics are plotted.

Black	Sum squared error of training matrix
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
Cyan	Sum squared error of test matrix



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.

Interrogate Model command

Model Menu

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.

In addition to the aforementioned interactive mode the "SensRpt" button can generate a sensitivity report at the desired point in the input space. The "Run from DM" button will read the input values from an external data matrix, execute the model and then write back the predictions to the data matrix.

The dialog contains two grids and four buttons. The grid on the left side contains the fields necessary for entering input values. The four columns are: the "Input" field that displays the input variable name, the "Value" field were the input value can be entered and the next two fields "Minimum" and "Maximum" are for information purposes only showing the input bounds.

The grid on the right hand side contains the model output fields. The "Output" field holds the output variable's name and the "Value" field hold the latest prediction.

The following buttons are arranged along the bottom of the dialog:

"Run from DM"	Read and write values from an external data matrix.
"SensRpt"	Generates a sensitivity report.
"Finished"	Exits the Interrogate Model dialog.
"Update"	Runs the model. (same a pressing the carriage return)

Statistics Report command

Model Menu

Use this command to generate a statistical report. This report gives you the basic statistical information about your data matrix, training matrix or test matrix. The information includes the total number of rows and date of creation; plus the following specific information about each variable, number of observations, mean, standard deviation, minimum and maximum values and the sum of the squares value. If this report was generated from a neural model the also had a test matrix the statistics of the test matrix will follow the training matrix.

Following the statistical report is information from the column descriptor. This information contains the minimum and maximum values, time shifting of any column and the clipping values.

Sensitivity Report command

Model Menu

Use this command to generate a sensitivity report showing the sensitivity of the output variables to changes in the input variables. The sensitivity is calculated by summing the changes in the output variables caused by moving the input variables by a small amount over the entire training set. There are three variables accumulated during the calculation. The AbsAve Sensitivity variable is the average of the absolute values of the change in the output. This value is then divided by the total amount of change for all input variables to normalize the values. The Ave Sensitivity is calculated the same as the AbsAve variable except the absolute values are not taken. If the direction of the change in the output variable is always the same then the Ave and AbsAve sensitivities will be identical. The third variable calculated is the peak sensitivity and the row in the training matrix that it occurred.

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

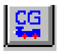
Use this command to turn off the training status graph. If this command is checked then there will be no graphical display during training.

CG Tweak command

Model Menu

Use this command to run the conjugate gradient training once. This is useful to remove the bias that develops during back propagation training on time series data.

Shortcuts

Toolbar: 

Auto Save command

Use this command to toggle the automatic saving of the model. If this option is turned on (menu checked) during training then the model is be saved every 10 minutes. If a training epoch takes longer then 10 minutes then the model will be saved after the epoch is finished training.

Basic Statistics command

Use this command to generate a statistical report. This report gives you the basic statistical information about your data matrix, training matrix or test matrix. The information includes the total number of rows and date of creation; plus the following specific information about each variable, number of observations, mean, standard deviation, minimum and maximum values and the sum of the squares value. If this report was generated from a neural model the also had a test matrix the statistics of the test matrix will follow the training matrix.

Following the statistical report is information from the column descriptor. This information contains the minimum and maximum values, time shifting of any column and the clipping values.

The following is an example of a basic statistics report.

```
        MODEL ALL.BEP
        Model Performance Statistics
        (Training Matrix) # of Points = 343
        November 19, 1996

Variable      Mean          Std Dev         Minimum         Maximum
Sum Sq
REFIN       2918.6590       2521.5565       -0.000046       9000.0000
217452055
COAR        22.731195        8.250117        8.100000       39.299999
23278.036
LWFL         1.729942        0.701114        0.660000        2.680000
168.11379
NFPGP        4.275630        3.643991        0.949000       16.910000
4541.3062
VISC        16.350729        4.319640        7.400000       32.000000
6381.4776
CSF         522.08454       149.22249       240.00000       775.00000
7615434.5

TEAR
   Measured       64.582857       24.477921       16.400000       152.39999
204915.67
   Predicted      62.501654        0.159297       62.074112       63.066227
8.678453
   Residual        2.081203       24.413081       -46.06767       89.723495
203831.49
   R Square        0.005291


------------------------------------------------------------------------

        MODEL ALL.BEP
        Model Performance Statistics
        (Test Matrix) # of Points = 332
        November 19, 1996

Variable      Mean          Std Dev         Minimum         Maximum
Sum Sq
REFIN       3503.6145       2695.2882       -0.000046       10000.000
```

```
240457562
COAR        23.642470       8.175550       8.100000       39.299999
22123.911
LWFL         1.857078       0.662284       0.660000        2.680000
145.18326
NFPGP        3.686114       3.285810       0.949000       16.910000
3573.6577
VISC        16.392169       3.885491       7.400000       32.000000
4997.1197
CSF         503.47891      157.73277      235.00000      770.00000
8235156.8

TEAR
   Measured       65.310843       24.260444       18.000000       163.19999
194816.39
   Predicted      62.513126        0.147476       62.101749       63.079483
7.198964
   Residual        2.797718       24.208675       -44.31919       100.76752
193985.83
   R Square        0.004263


-------------------------------------------------------------------------

Variable       Minimum         Maximum       TimeS     ClipHi     ClipLo
REFIN         0.000000        10000.000        0        MISSING    MISSING
COAR          8.100000        39.299999        0        MISSING    MISSING
LWFL          0.660000         2.680000        0        MISSING    MISSING
NFPGP         0.949000        16.910000        0        MISSING    MISSING
VISC          7.400000        32.000000        0        MISSING    MISSING
CSF          235.00000       775.00000         0        MISSING    MISSING

TEAR         163.19999        16.400000        0        MISSING    MISSING
```

Correlation Analysis command

Data Menu

This report displays the standard Pearson correlation coefficients for each variable in a matrix format. Each variable is printed as a column and as three rows. The first row is the Pearson correlation, the second row indicates the probability of the correlation under the null hypotheses that the statistic is zero and the third row is the number of observations where both variables are not missing.

Correlation is a measure of strength of the linear relationship between two variables. If a variable (X) is expressed as a linear function of another variable (Y) as in the equation Y = mX + B, then a correlation of 1 indicated that the variables are directly related or if the correlation is -1 then the variables are inversely related. If the correlation between two variables is 0 then each variable has no linear predictive relationship to the other. When the variables are normally distributed, then a correlation of 0 also means that they are independent of each another.

The following example shows the correlation of three variables.

Pearson Correlation Coefficients / Prob > |R| under H0: Rho = 0 / N

```
      V1          V2          V3
V1    1.00000    -0.05736     0.04086
      0.0000      0.2838      0.4454
      351         351         351
V2   -0.05736     1.00000     0.04084
      0.2838      0.0000      0.4456
      351         351         351
V3    0.04086     0.04084     1.00000
      0.4454      0.4456      0.0000
      351         351         351
```

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.

Append Data Matrix command

Data Menu

Use this command to append an identical matrix to the end of the current data matrix. If the two data matrices have the same number of columns and the new matrix has the default variable labels then the new matrix will be appended. Otherwise the two matrices will be concatenated.

Reserve Testing Data command

Data Menu

Random

Use this command to randomly place a V in the RT-type field of the data matrix. Fields with this value in the RT 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.

Sequentially

Use this command to sequentially place one or more Vs in the RT-type field of the data matrix. Fields with this value in the RT field will be used to load the test matrix rather than the training matrix. This command displays the a dialog box that allows you to select a greater than or less than operator and a beginning row number. For example if you want the last 20 rows in a one hundred row data matrix to be reserved for testing then select greater than and 80 then press OK.

Clear

Use this command to remove all Vs from the data matrix.

Fill Missing by Keeping Last Value 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.

Fill Missing by Interpolation command

Data Menu

Use this command to fill in missing values with the linear interpolation of the previous non-missing row and the next non-missing 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

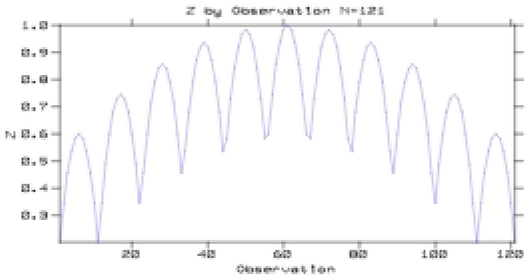
Graphs Menu

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

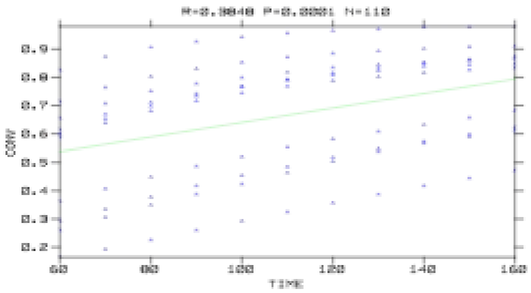
Use this command to generate a graph of the selected variable (Y axis) plotted against the row number (X axis). Any measured, input, output, predicted or residual variable can be plotted.



Scatter Plot command

Graphs Menu

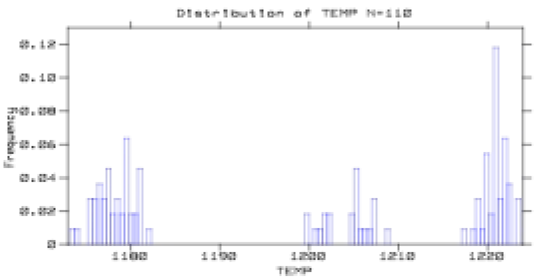
Use this command to generate an XY scatter plot of two selected variables. A symbol is placed for each XY pair. The points are not interconnected. If one or both points are missing then the pair is not plotted. Any measured, input, output or predicted variable can be plotted.



Distribution command

Graphs Menu

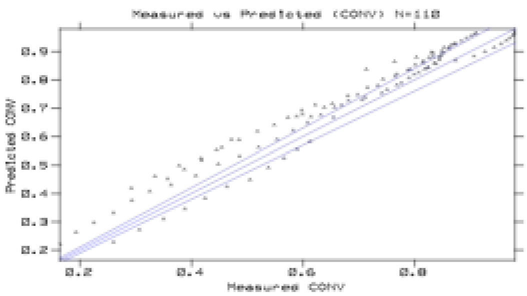
Use this command to generate a frequency distribution plot of a selected variable. The range of the variable is divided into N boxes (options). Each time a point (of M observations) falls into a box the value 1/M is added to that box. The total of all the boxes equals 1. Any measured, input, output,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



Meas vs Pred command

Graphs Menu

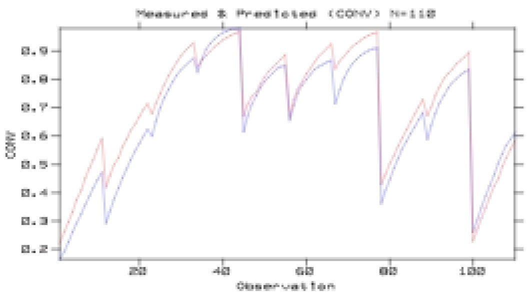
Use this command to generate a measured verses predicted graph. For every measured value a predicted value is calculated and a symbol is plotted on the graph. A line is drawn showing how a perfect model would plot. If a tolerance (Training parameters) is set to a number other than 0, then the + and -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



Meas & Pred command

Graphs Menu

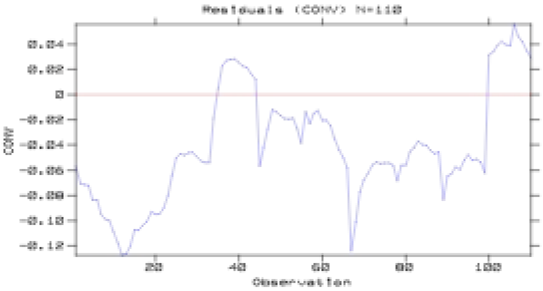
Use this command to generate a graph of the measured and predicted outputs of a model. The selected output variable is plotted against the row number (X axis). The blue line represents the measured value of the selected variable and the red line represents the predicted value. Any predicted variable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



Residuals command

Graphs Menu

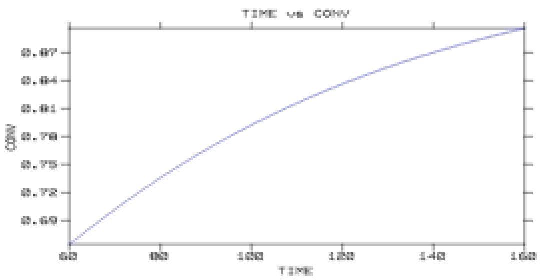
Use this command to generate a graph of the difference between measured and predicted outputs of a model. The residual calculation is plotted against the row number (X axis). The blue line represents the residuals and a red line is drawn at zero. Any predicted variable can be plotted.



XY command

Graphs Menu

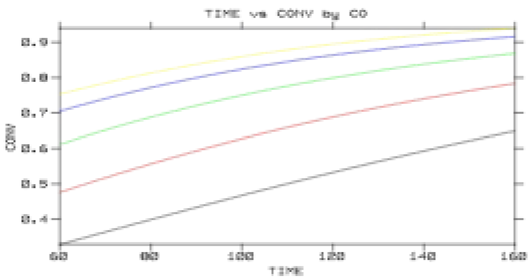
Use this command to generate a line graph showing how a model output (Y) variable changes in response to an input (X) variable. The X variable is varied from the minimum to the maximum observed value. Any combination of input and predicted variable can be plotted.



XY Effect command

Graphs Menu

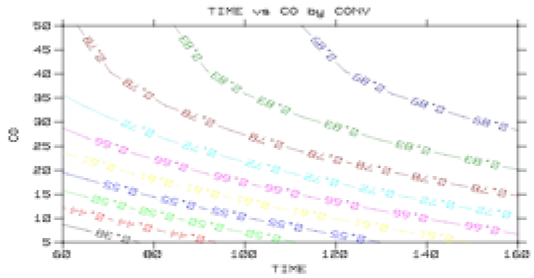
Use this command to generate a family of line graphs showing how a model output (Y) variable changes in response to an input (X) variable and an effect (E) variable. The X variable is varied from the minimum to the maximum observed value and the E-variable is varied from the minimum to maximum observed 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

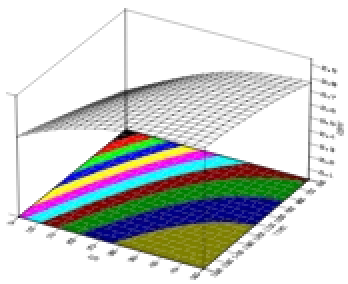
Use this command to generate a topological style graph showing how the neural output is effected by both an X and a Y (neural inputs) variables. The contours can either be drawn as a labeled contour lines or as a filled contour. The number of curves and type of contour is selectable in the contour dialog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



3D Surface command

Graphs Menu

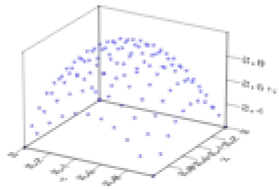
Use this command to generate a 3 dimensional surface plot of how a neural output is effected by 2 input variables (X and Y). The surface can be drawn as a black fish net or a colored fishnet or as a colored filled surface. Additionally a contour plot can be drawn under the surface. Addition options are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

Window Menu

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

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.

How To Model Process Data With Time Lags

Many manufacturing processes are time sequential in nature. For example, chemicals or materials can enter a process and pass through a maze pipes, mixers, storage tanks and reaction vessels before finally exiting, hopefully, as a more valuable product many minutes or hours later. During this time, presumably, a variety of sensors have been recording this transformation.

The problem is that each sensor may be positioned at a different point in time relative to the other sensors. To accurately model such processes these time differences must be accounted for. The method for doing this in NNMODEL is the TimsS field in the variable descriptor information. To clarify this lets run through a simple process and try to do some modeling.

The process outlined below produced some imaginary product(OUT) based on the amount of input (IN2) to the process. The hook is that the amount of OUT is dependent on the time that the input enters the process.

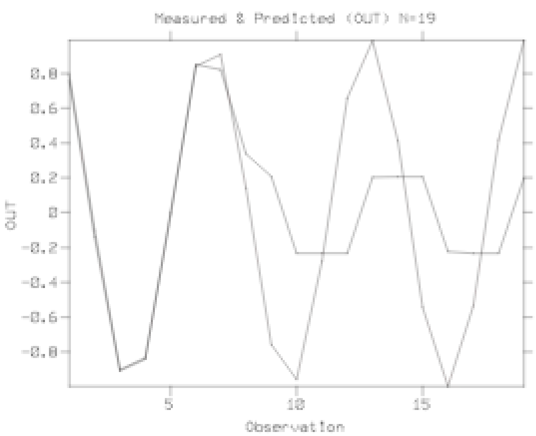
```
            +-------------+
IN2  ---->|    Delay    |-----> OUT
            +-------------+
```

The following dataset was collected from the aforementioned process. In this data matrix each row is equivalent to a one minute scan. The date and time fields were included for the purpose of demonstrating that the data is from a time series process and is not used in the neural model. The variable V1 is included from the data acquisition unit and only indicates the sequence number of the scan. Variable IN2 is the raw materials being fed into the process. The variable OUT is the product produced by the process. This is a very simplistic simulation of a non-linear process with a time lag.

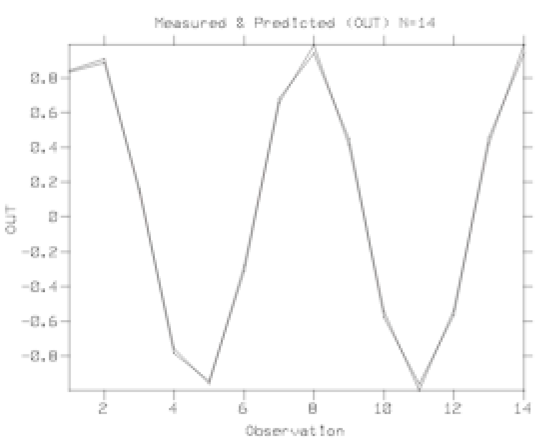
Date	Time	V1	IN2	OUT
19960703	10:00:00	1.00000	0.84147	0.75680
19960703	10:01:00	2.00000	0.90930	-0.14112
19960703	10:02:00	3.00000	0.14112	-0.90930
19960703	10:03:00	4.00000	-0.75680	-0.84147
19960703	10:04:00	5.00000	-0.95892	0.00000
19960703	10:05:00	6.00000	-0.27942	0.84147
19960703	10:06:00	7.00000	0.65699	0.90930
19960703	10:07:00	8.00000	0.98936	0.14112
19960703	10:08:00	9.00000	0.41212	-0.75680
19960703	10:09:00	10.00000	-0.54402	-0.95892
19960703	10:10:00	11.00000	-0.99999	-0.27942
1996070	10:11:0	12.0000	-	0.6569

3	0	0	0.53657	9
1996070	10:12:0	13.0000	0.42017	0.9893
3	0	0		6
1996070	10:13:0	14.0000	0.99061	0.4121
3	0	0		2
1996070	10:14:0	15.0000	0.65029	-
3	0	0		0.54402
1996070	10:15:0	16.0000	-	-
3	0	0	0.28790	0.99999
1996070	10:16:0	17.0000	-	-
3	0	0	0.96140	0.53657
1996070	10:17:0	18.0000	-	0.4201
3	0	0	0.75099	7
1996070	10:18:0	19.0000	0.14988	0.9906
3	0	0		1

When you try to predict OUT using V1 and IN2 as inputs youll find it difficult to get and R square above 0.45 even after 100,000 presentations of the training set. The following graph demonstrates how poorly the neural network is modeling the data.

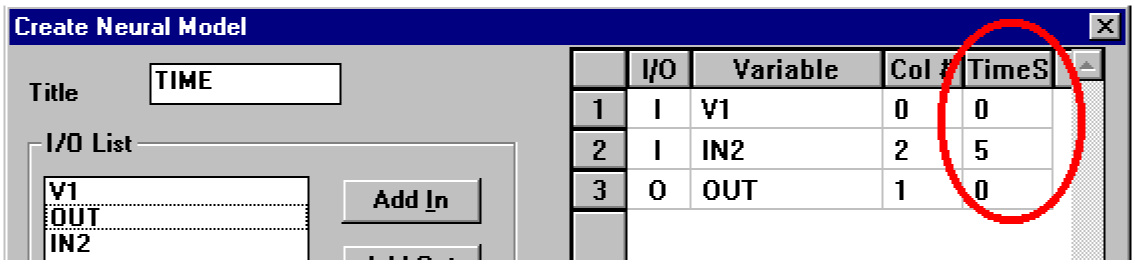


However, if the neural network is presented with the data shifted by 5 time units the resulting model fairs much better with an R square of 0.99 after only 1,000 training presentations. A sensitivity analysis shows that the V1 input isnt needed at all. The following graph demonstrates how well the model is performing.

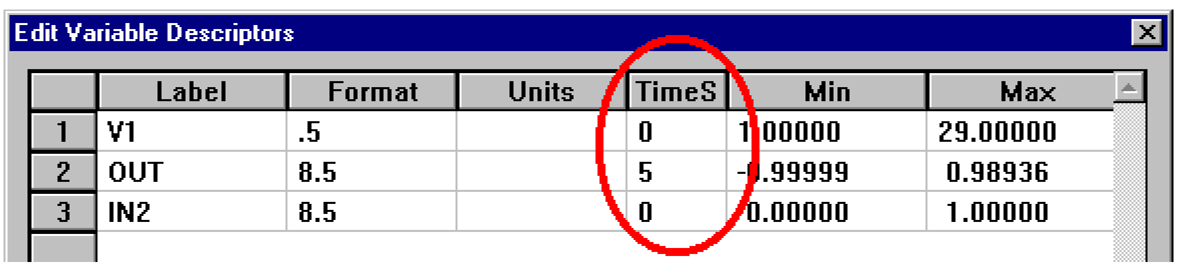


Okay, so how was this miracle achieved? To do this, the model cannot be trained with the IN2 and OUT variables taken from the same point in time. But rather, the OUT variable must be aligned with the IN2 variable measured 5 minutes in the past. Before the current version of NNMODEL you would have to create a calculated column using the LAG function. This isnt to much of a problem when working with a few input variables, but if your process has 40 inputs this could quite a bit of your time.

In order to make this task a little easier NNMODEL has been modified to allow you enter these time shifts in either of two places. When a model is created the time shift can be entered directly into the model definition. After selecting your inputs and adding them to the model definition the TimeS field is available for editing. The red bubble on the screen image below shows where this field appears in the Create Neural Model dialog with is executed by selecting the File / New Neural Model menu command.



The time shift can also be entered in the Edit Variable Descriptor dialog. Once again the red bubble indicates the new TimeS field. This dialog is activated by selecting the menu command Edit / Variable Descriptors.



DDE Interface

Dynamic data exchange (DDE) is a powerful feature of the Microsoft Windows™ operating environment. DDE provides an interface through which applications can exchange data. DDE enables applications to form hot links with DDE servers and obtain data in real time. A hot link is a direct data feed from one application (a DDE server) to another (a DDE client). As the data changes at the server, the server sends the new information to the client to be processed.

Currently, NNMODEL provides a method of interrogating pre-built neural models via Windows™ DDE facility. Thus, any Windows program that supports the DDE interface can use the models built with NNMODEL. For example, you could write a Visual Basic program that collects data from an A/D board and writes it to a neural model. Then the neural model could automatically update hot links in an Excel spreadsheet displaying the models predictions.

Functions supported by NNMODELs DDE facilities

ChanNo = DDEInitiate("NNMODEL", ModelName)	Initiates a conversation with NNMODELs DDE interface. The conversation is specific to the model passed in the ModelName field.
DDEPoke ChanNo, InputVariableName, Value	Places the value passed into the input variable of the selected model.
Result = DDERequest(ChanNo, OutputVariableName)	Retrieves the predicted value of the model.
DDEExecute ChanNo, "[ModelName]"	Causes the model named ModelName to be executed, thus forcing a prediction. (Note 1)
DDETerminate ChanNo	Dissolves the DDE communications link.
Hot Links	Excel supports hot links of the form =NNDDE|ModelName!OutputName. NNMODEL will automatically update cells with the models output has changed.

Note1: The execute command is not needed unless hot links are used. (i.e. Basic -> NNMODEL -> Excel)

NNMODEL also supports the following File-Execute commands:

DDEExecute ChanNo, "[Open (ModelName)]"	Opens and loads the model named ModelName found in the \nnmodel\dde sub-directory.
DDEExecute ChanNo, "[Close (ModelName)]"	Closes and removes the model named ModelName from the NNMODEL application.
DDEExecute ChanNo, "[CloseAll]"	Closes all open models found in the NNMODEL application.

Assumes that the conversion link was initiated with: ChanNo = DDEInitiate("NNMODEL", "File")

The following Excel Basic macros demonstrate each feature of the DDE interface.

Macro1 shows how to set the two inputs (IN1, IN2) to the XOR model and retrieve the prediction (OUT) and then places the prediction into Sheet1 Cell B4.

```
Sub Macro1()
    ChanNo = DDEInitiate("NNMODEL", "XOR")
    DDEPoke ChanNo, "IN1", Sheets("Sheet1").Range("B2")
    DDEPoke ChanNo, "IN2, Sheets("Sheet1").Range("B3")
    Sheets("Sheet1").Range("B4") = DDERequest(ChanNo, "OUT")
    DDETerminate ChanNo
End Sub
```

Macro2 shows how to load the model XOR that is stored in the sub-directory \nnmodel\dde

```
Sub Macro2()
    ChanNo = DDEInitiate("NNMODEL", "File")
    DDEExecute ChanNo, "[Open (XOR)]"
    DDETerminate ChanNo
End Sub
```

Macro3 demonstrates how to unload the XOR model from the NMODEL application

```
Sub Macro3()
    ChanNo = DDEInitiate("NNMODEL", "File")
    DDEExecute ChanNo, "[Close (XOR)]"
    DDETerminate ChanNo
End Sub
```

Macro4 shows how to unload all models form the NNMODEL application

```
Sub Macro4()
    ChanNo = DDEInitiate("NNMODEL", "File")
    DDEExecute ChanNo, "[CloseAll]"
    DDETerminate ChanNo
End Sub
```

Macro5 shows how to set the two inputs (IN1, IN2) to the XOR model and then force the model to predict and update hot links to other programs.

```
Sub Macro5()
    ChanNo = DDEInitiate("NNMODEL", "XOR")
    DDEPoke ChanNo, "IN1", 0.5
    DDEPoke ChanNo, "IN2, 0.6
    DDEExecute ChanNo, "[XOR]"
    DDETerminate ChanNo
End Sub
```

An Excel example of using the DDE interface is included in the sub-directory \nnmodel\dde. Use the following step by step procedure to run this demo:

1. Start NNMODEL and load the model xor.bep found in the \nnmodel\dde sub-directory.
2. Start Excel and load the spreadsheet named xor.xls found in the \nnmodel\dde sub-directory
3. Enter a number (between 0 and 1) into the field named IN1

4. Enter a number (between 0 and 1) into the field named IN2
5. Press the button labeled Run Xor (this executes macro1)
6. The field OUT should have been updated with the predicted value.

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.

NNLIB subroutine definitions

Returns	Name	Neural 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

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);
```

NNCALC Excel Add In

NNCalc is an Excel 5 Add-In that allows you to test or use your neural models from with-in an Excel spreadsheet. You can now build complex systems of standard mathematical equations and neural models then execute them at Excel speeds.

To build a model that NNCalc can load; first build the model as usual using NNMODEL. Then export the model as a feed-forward ENN file and copy it into the c:\nncalc sub-directory. **Note**: The current version of NNCalc ***only*** looks in the c:\nncalc sub-directory even if you have installed NNMODEL on the another hard disk.

NNCalc

When the NNCalc function is called the neural model named ModelName is looked up and it is loaded into memory (if not already present). It then parses the rest of the passed parameters and returns an error if the number of passed parameters doesnt match the number of inputs to the model. If all parameters are correct then NNCalc returns the predicted value of the network,. Otherwise, it will return an error code indicating a reason for failure.

The parameters passed to NNCalc are in the following form:

=NNCalc (ModelName, FirstInput, SecondInput, …., LastInput)

where:

ModelName	This is the file name of the neural network. No file extension or sub-directory should be specified. NNCalc assumes C:\NNCALC as the sub-directory and ENN as the extension. For example: the ModelName XOR is translated into C:\NNCALC\XOR.ENN.
FirstInput	This is the first input included into the network. The sequence order can easily be determined by viewing the input list in the Interrogate Model dialog.
SecondInput	The second input to the network.
LastInput	The final input to the network.

NNCalcM

Because NNCalc only returns the first output of a neural model (a limitation of Excel) a function is needed to get additional model outputs. NNCalcM returns the predicted values for models that have more than one output. NNCalcM does not evaluate the model (that's NNCalcs job). It simply returns the network's output value.

The parameters passed to NNCalcM are in the following form:

=NNCalcM ("ModelName", NeuralOutput, RefToNNCalc)

=NNCalcM (ModelName, FirstInput, SecondInput, …., LastInput)

where:

ModelName	This is the file name of the neural network.
NeuralOutput	The index to the neural output to be returned. The first neural output (the one returned by NNCalc) can be selected by passing a 1. The second output can be selected by a value of 2 etc.

RefToNNCalc	This is the cell reference to where NNCalc is used for the selected. For example if =NNCalc("xor2",b1,b2) is in cell G2 then to return the second output of model "xor2" add =NNCalcM("xor2",2,G2) to the cell where you want to display the second output. This "cell reference" is needed so that Excel knows to call NNCalc before calling NNCalcM.

NNCalcSet

NNCalcSet is used to turn on and off diagnostic functions. Currently there is only one diagnostic - Logging. Use the function =NNCalcSet(1) to turn on logging and =NNCalcSet(0) to turn off logging. The default value is logging off. The logging information is stored in ASCII text format and can be found in the file named C:\NNCALC.LOG.

The parameters passed to NNCalcSet are in the following form:

=NNCalcSet("LogOnOff")

where:

LogOnOff	To turn on logging pass a one and to turn off logging pass a zero.

A note about input scaling of the passed variables. The input variables are linearly scaled according to the minimum and maximum values of the initial training set. For example, the XOR models used in the demonstration spreadsheet had minimum and maximum values of 0.0 and 1.0 respectively. When the user passes values outside the range of 0.0 and 1.0 NNCalc scales them to fit within the neural operating range. In this example the XORs 0.0 becomes 0.2 and the XORs 1.0 becomes 0.8. This serves two functions: (1) the neural inputs are balanced around the neurons neural position (0.5) and (2) it scales all input to the same range, thus, input with large numbers dont overburden inputs with small numbers.

If you pass parameters that are outside of the normal operating range then these values may overwhelm the other inputs to the neural network.

There are five errors that can be generated by NNCalc. These errors are listed below:

Error codes

ErrPE	Bad parameter form. First parameter is not a string (ModelName) or any of the neural inputs parameters arent real numbers or references to real numbers.
ErrNF	Model not found. The ENN file wasnt found in c:\nncalc sub-directory
ErrIM	Too many inputs in model.
ErrBN	Bad file name. Only 1-8 alpha-numeric characters are allow in the file name. The extension .ENN is automatically appended by NNCalc.
ErrPI	Number of inputs passed doesnt match model. If the neural model calls for 3 inputs then you must pass 3 inputs.

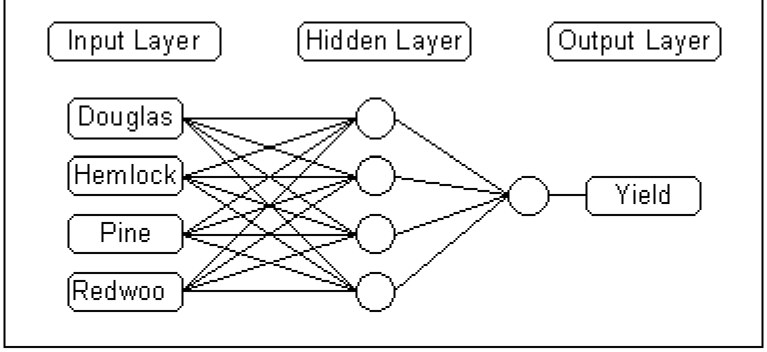
The following are examples of how to use NNCalc in an Excel spreadsheet.

=NNCalc (XOR,1,0)	This example fires the XOR model with the a 1.0 loaded into input1 and a 0.0 loaded into input2.
=NNCalc(XOR,A2,A3)	This example executes the XOR model loading input1 with the value

	found in cell A2 and input2 with the value in cell A3.
=NNCalc (COATING, B3, B4, D6, E6, 50)	This example predicts coating from the values contained in cells B3, B4, D6, E6 and 50.0 is loaded into the last input.

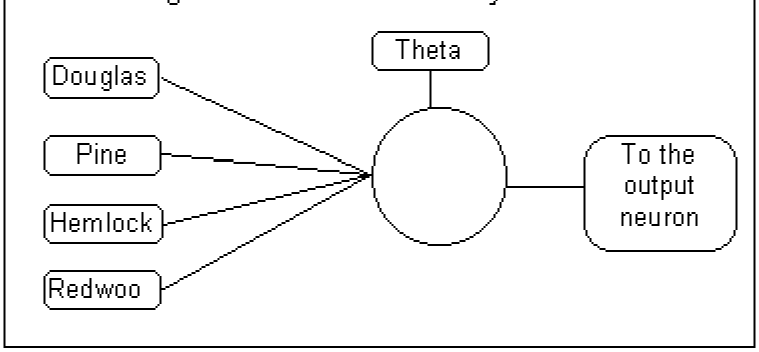
Back Error Propagation Neural Networks

Figure 1



A Back Error Propagation (BEP) neural network is constructed of many inner-connected processing elements (neurons) each executing simple nonlinear equations. The neurons are partitioned into layers, with each neuron in a layer fully connected to all neurons in the next higher layer. Hence, as demonstrated by Figure 1, all input layer neurons are connected to every hidden layer neuron and every hidden layer neuron is connected to the output neuron. Each connection has a strength or weight associated with it, and a network learns or is trained by modifying these weights. This results in the mapping of inputs to outputs via an abstract hidden layer.

Figure 2 - Details of a hidden layer neuron.



Additionally, the user can select an option that directly connects the input layer to the output layer. If this option is selected the preceding network would have four additional weights connected from the input layer to the output neuron.

Each neuron has one additional weight as an input. The Theta input allows an addition degree of freedom when trying to minimize the training error. A typical hidden layer neuron from the preceding network has 5 weights (4-inputs, 1-theta). The value of Theta is set globally (default 0.5) and Thetas weight is learned during back-propagation.

After training on a statistically designed experiment, a network can be used to predict the behavior of multivariate experimental data. Inputs or Independent Variables (IV) from the experimental database enter the network at the input layer, and outputs or Dependent Variables (DV) exit at the output layer. The input neurons contain simple scaling activation functions that translate the range of the independent variable from real world numbers to the most linear part (See Figure 2) of the neural equation (i.e. $0.2 < IV < 0.8$, centered at 0.5). The hidden and output layer neurons are nonlinear activation functions of the form:

$$o_j = \frac{1}{1 + e^{-\left(\sum_i w_{ji} o_i + w_j\right)}}$$

Where, wji denote connection weights, oi denotes output of a previous layers neuron, wj denotes the adjustable threshold for the neuron, i denotes the number of inputs to the neuron from the previous layer and j denotes the current neuron.

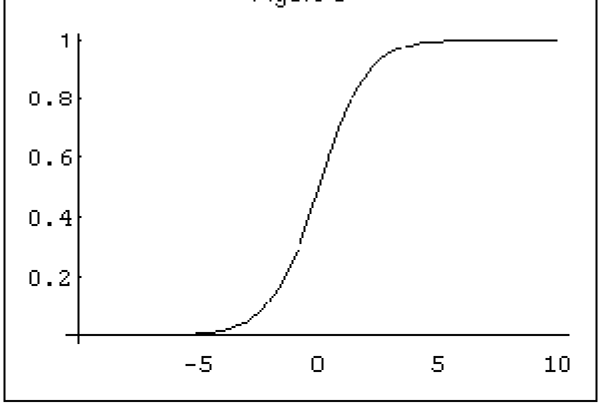


Figure 3

Training the network involves first initializing the connection weights to small random numbers. Then row by row the independent variables are presented to the input layer neurons and a dependent variable is presented to the training input of output layer. The network first calculates its own output value and then compares this with the target value presented to the output neuron. This difference is the error signal. If there is no error, then no learning takes place. Otherwise the inner-connecting weights are changed to reduce the error. Each inner-connecting weight between the output layer and the hidden layer is changed by an amount proportional to the product of an error signal and the first derivative of the nonlinear activation function as given by:

$$\frac{\partial o_j}{\partial net_j} = o_j(1 - o_j)$$

Where: netj denotes the total network error, oj denotes the output of the current neuron and j denotes the current neuron. The error signal for the hidden layer to input layer, for which there is no target value, is calculated recursively in terms of the error signals of the neurons to which it directly connects and the weights of those connections.

The presentation of the experimental data is repeated until the network is adequately trained. Adequacy can be measured as: all prediction within a tolerance band, an R square value, a time limit, the number of presentations of the experimental data or a minimum total error. Generally, if the data is from a well designed experiment the network can be trained aggressively (i.e. 2000-10000 presentations of the data). However, is the data is highly variable or from a poorly designed experiment a more cautious training session is in order (i.e. stop training at an R square or limit the training to 500-2000 presentations).

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.

Grad Color

Draw the 3D surface map using using shades of blue and red only.

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.

1 Std Dev

Display plus/minus 1 standard deviation lines on the Measured vs. Predicted and Measured and Predicted graphs.

2 Std Dev

Display plus/minus 2 standard deviation lines on the Measured vs. Predicted and Measured and Predicted graphs.

3 Std Dev

Display plus/minus 3 standard deviation lines on the Measured vs. Predicted and Measured and Predicted graphs.

Monochrome

Display graphs in shades of gray.

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.

TimeS

This parameter is used to shift its value backwards by the number of rows indicated. Use this field only on time series data to shift the time relationship of this variable in relation to the other variables. For information see the How To on time series data.

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.

Add In

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

Add Out

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

Add I/O

Add the currently selected variables in the I/O list to the model list as both inputs and outputs. Use this button to setup an auto associate the inputs to the outputs.

Invert List

Invert the current selections

Network List Grid

The current model design. This grid describes the input / output structure for the network. Each row contains fields to indicate whether the variable is to be used for an input or output, the name of the variable, the column number of the data matrix where the variable resides and finally the third field indicates the time shift for the variable. This field should only be used on time series data. The field is used to shift a variable (in time) relative to the modeled output variable. For more information on time shifting see the How To on modeling time series data.

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.

Use max/min from data matrix

Normally when a neural network is created the maximum and minimum values for the training matrix are set by the data loaded into the training matrix. If this button is checked then the maximum and minimum values of the data matrix will be used rather then the sub-set (training matrix). This option is used to extrapolate beyond the limits of the training matrixs range.

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. If this value is the result of a calculated field then the cell is automatically updated when any real input is changed.

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. The values entered into the input grid are used for the center of the sensitivity calculations.

Finished

Done Interrogating the model

Update

Update the model's prediction

Run from DM

Use an external data matrix for input/output. The data matrix must have columns for all the inputs and outputs contained in the model. The input columns must contain values that are within the maximum and minimum range of the training matrix. If there is a missing value in the input column then the row will not be predicted. The output columns may contain missing values. If the output columns contain non-missing values then they will be over written with the predictions. If the external data matrix is not open when Interrogate Model is started then the disk image will be updated and then opened, otherwise only the memory image will be updated.

Edit Training Parameters dialog box

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

Max Hid Neurons

This parameter has two modes of operation. In standard BEP learning it is the number of hidden neurons created during the initialization of the network. The number remains constant for the life of the network.

When ***Automatic Increment*** mode of training is selected, then Max Hid Neurons means the maximum number of hidden neurons that can be added to the model during the training session. In ***AI*** mode the network is initialized with one hidden layer neuron but the network is allowed to grow the number of hidden neurons to the maximum set by the Max Hid Neurons parameter during the training session.

Eon

Number of presentations of the training sets to present to the network before checking the statistics or updating the training progress graph. This parameter is usually set to 100. In standard BEP training lowering this parameter will cause the training graph to be updated more often. If training mode is selected then this parameter is automatically set to 1. This is done because ***CG*** training is very slow and waiting for a 100 epochs before updating the graph is undesirable.

In ***Automatic Increment*** training mode the Eon parameter is also used as a timing parameter to gauge when additional neurons should be added to the network. ***AI*** is very sensitive to this value so if your using ***AI*** is parameter should be set to 100.

Max Train Count

Maximum number of the times the training matrix can be presented during training. Training is automatically stopped if this number is reached.

Tolerance

Acceptable error in percent of total error. This parameter is used in graphs, as a training stop flag and as a locus in calculating ***AI*** statistics.
When tolerance is drawn in graphs a number, such as, 0.05 represents a tolerance band of +5 and -5 percent of the total error.
When the Stop Training On - Tolerance radio button is selected the training will stop when all the predictions for a epoch are less then the percent indicated by the tolerance parameter.

This parameter is also used in ***Automatic Increment*** training mode for calculating the number of predictions above and below the tolerance band. These two statistics are graphed during the training operation.

Error Tolerance

This parameter is used to stop training if the Stop Training On - Error Tolerance radio button is selected. The error tolerance is the maximum value of the Sum Square error that would be considered acceptable.

Good RSQ

This parameter is used to stop training should an acceptable R square statistic develop during the training and the Stop Training On - Good R Square radio button is selected. This parameter should be set to what the user considers a good R square.

Signif Increase

Use in calculating when not to add a new neuron. Used to indicate when there has been a significant increase in the R square statistic.

No Signif Increase

Used in calculating when to add a new neuron. Used to indicate when there has not been a significant increase in the R square statistic.

Hidden Freeze

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

Best Eon

How many eons past the best eon should training continue. Sometimes a better model will develop 50 to 100 eons after the earlier best eon. (depending on the number of epochs in an eon).

Best Fuzz

How close to zero is close enough. This value is used to determine when training really isnt improving much in the best eon routine.

Learning Rate

The learning rate for the hidden layer to output layer connections. In standard BEP training this number be used for the entire training session. If any of the incremental learning options are selected then this number represents the initial learning rate and it is reduced on the older neurons as newer neurons are added.

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 for a back error propagation learning method.

Theta

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

Random Fact

The scaling factor used when initializing the model weights. This number represents the maximum value that a random weight could attain.

InRandom Fact

The scaling factor of Gaussian noise used when training the model. Setting this value to anything other than 0.0 will cause Gaussian noise to be added to the training set during training. The larger the number the smaller the amount of noise.

Seed

The seed value to initialize the random number generator. If this value is set to -1 then the weights will be totally randomized. Any value other than -1 will cause the weights to be random in relation to each other, but the weights will always be the same random value.

CG Max Iterations

Number of iterations the conjugate gradient routine can do before returning to update graphs. Making this number larger allow the CG routine to work quicker but when the CG routine is running Windows will become less responsive.

Hidden Layer Addition - Fixed # Hid Neurons

Train using the standard BEP algorithm. The number of hidden layer neurons is set equal to the Max Hid Neurons parameter when the network is initialized.

Hidden Layer Addition - Equal Spaced Increment

Start out with one hidden layer neuron and add new hidden neuron at equally spaced intervals. Earlier neuron learning rates are reduced as additional neurons are added.

Hidden Layer Addition - Manual Increment

Start training with one hidden layer neuron and add additional neurons as directed by the user.

Hidden Layer Addition - Automatic Increment

Start training with one hidden layer neuron and use the ***Automatic Increment*** routine to statistically detect when to add a neuron to the hidden layer.

Stop Training On - Tolerance

Stop training when all predictions are within the tolerance band indicated by +/- the Tolerance parameter.

Stop Training On - Error Tolerance

Stop training when the total error is below the value specified in the Error Tolerance parameter.

Stop Training On - Good R Square

Stop when the R square of the model is at or below the value specified in the Good RSQ parameter.

Training Method - Standard BEP

Use Back Error Propagation as the training method.

Training Method - Conjugate Gradient

Use conjugate gradient optimization as the training method.

Training Method - Keep Best Eon

The turns on the Best Eon routine. This routine will keep the best model developed during the training session and restore it as the final model at the end of the session. Best can be determined by either as lowest Mean Square Error or highest R Square as selected by the Stop Training On Sum Error Tolerance or Good R Square switches. In order for Keep Best Eon training method to work a number must also be entered into the Best Eon parameter. This parameter tells the Keep Best Eon routine how far to continue training after no significant improvement has been made to the selected error measurement. No significant improvement is indicated by the current models error measurement being not better then the saved models error measurement plus the Best Fuzz parameter.

Training Method - Best Train

Use the training matrix MSE or R Square calculation to determine best model.

Training Method - Best Test

Use the test matrix MSE or R Square calculation to determine best model.

Training Method - Best Train and Test

Use the average of the training and test matrices MSE or R Square calculation to determine best model.

Circular BP Hidden Layer

Add an additional Theta like input to each of the hidden layers neurons. This input will feed the sum squares of the input values to the neuron.

Circular BP Output Layer

Add an additional Theta like input to each of the output layers neurons. This input will feed the sum squares of the input values to the neuron.

Connect Inputs to Outputs

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

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.

The third column in the I/O selector grid is the TimeS field. This field can be used (in the case of time series data) to shift the inputs backward in time. For more information see the How To on modeling time series data.

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.

RSQ/MSE

This field contains either the R square statistic or the mean square error for the last best model. The value displayed depends on which error statistic is selected in the Stop on section of the Edit Parameters dialog.

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.

EP Button

The Edit Parameters button can be used to make last minute changes to the models parameters before the training begins.

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.

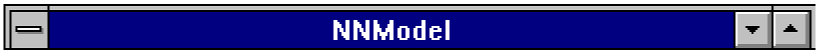
Type	Format
FLOAT	Standard formats (i.e. 1.23 -1.23 1.23E+00)
TIME	HH:MM or HH:MM:SS
DATE	YYYYMMDD

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:

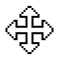
- Application Control-menu button
- Document Control-menu button
- Maximize button
- Minimize button
- Name of the application
- Name of the module
- Restore button

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.

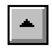
Shortcut

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.

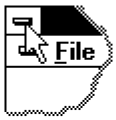
Shortcut

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).

Initialize and Start Training

Use this command to initialize and start training the neural network.

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.