The NEURAL Procedure

Overview

Procedure Syntax

PROC NEURAL Statement
ARCHITECTURE Statement
CODE Statement
CONNECT Statement
CUT Statement
DECISION Statement
DELETE Statement
FREQ Statement
FREEZE Statement
HIDDEN Statement
INITIAL Statement
INPUT Statement
NETOPTIONS Statement
NLOPTIONS Statement
PERTURB Statement
PRELIM Statement
QUIT Statement
RANOPTIONS Statement
SAVE Statement
SCORE Statement
SET Statement
SHOW Statement
TARGET Statement
THAW Statement
TRAIN Statement
USE Statement
ACTIVATION FUNCTIONS
COMBINATION FUNCTIONS

Details

Examples

Example 1: Developing a Simple Multilayer Perceptron (Rings Data)
Example 2: Developing a Neural Network for a Continuous Target
Example 3: Neural Network Hill-and-Plateau Example (Surf Data)

References

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
Overview

The NEURAL procedure trains a wide variety of feedforward neural networks using proven statistical methods and numerical algorithms. Please read the chapters "Introduction to Predictive Modeling" and "Neural Network Node: Reference" before reading this chapter.

Terminology and Architectural Description

Each INPUT, HIDDEN and TARGET statement defines a "layer". For the INPUT statement, a layer is a convenient grouping of variables, serving as inputs to the network, having common values for LEVEL and STD. Similarly, for the TARGET statement, a layer is a convenient grouping of variables, serving as outputs of the network, having common values for LEVEL, STD, ACTIVATION, COMBINATION, and other characteristics. Each layer is made up of "units". A unit is synonymous with the term "neuron" in the literature. It is the smallest computational entity in the network.

The INPUT and TARGET statements require a list of variables. If the variables are interval type, there is one unit corresponding to each variable. If the variables are nominal or ordinal, there is a unit for each level of each variable. The HIDDEN statement requires a number which determines the number of units in the associated layer. This layer is a grouping of units having common values for ACTIVATION, COMBINATION, and other characteristics.

Each INPUT statement produces an input layer. Because multiple INPUT statements are allowed, a network can have multiple input layers. However, connections from multiple input layers must be parallel; there cannot be serial connections between input layers. Similarly, multiple TARGET statements generate multiple output layers. The connections to multiple output layers must be in parallel; there cannot be serial connections between output layers. Hidden layers can be connected serially or in parallel.

Running the Neural Procedure

Before running the NEURAL procedure, you must run the DMDB procedure to create a DMDB-encoded training set and a DMDB catalog entry.

A typical application of the NEURAL procedure uses the following statements:

- A PROC NEURAL statement to specify the training set, DMDB catalog, and random number seed. If you specify a 0 or negative random number seed, running the same NEURAL step repeatedly may produce different results because of different random initial weights.

- One or more INPUT statements to specify input variables.

- One or more HIDDEN statements to define hidden layers.

- One or more TARGET statements to specify target variables.

- One or more CONNECT statements to connect layers in the network.
A PRELIM statement to do preliminary training to avoid bad local optima.
A TRAIN statement to train the network.
One or more SCORE statements to create output data sets.

Two kinds of statements are used with the NEURAL procedure. All of the statements in the list above are action statements, which directly affect the network or directly produce output. There are also option statements (NETOPTIONS, NLOPTIONS, and RANOPTIONS) that set options for future use. Options specified in an action statement apply only to that statement and do not affect subsequent statements. For example, the default technique for least-squares training is Levenberg-Marquardt (TECH=LEVMAR). If you execute a TRAIN statement with the option TECH=CONGRA, conjugate gradient training will be used for that particular training run. If you then execute another TRAIN statement without a TECH= option, the technique will revert to the default value of TECH=LEVMAR. But if you submit an NLOPTIONS statement with TECH=CONGRA, conjugate gradient training will be used for all subsequent TRAIN statements until you explicitly specify a different technique.

Each layer in the network has an identifier specified by the ID= option in the INPUT, HIDDEN, or TARGET statements. An identifier can be any SAS name, but to avoid confusion, you should not use the name of a variable in the training set. Layer identifiers are used in various statements, such as CONNECT, to specify previously defined layers.

Each unit in the network has a name. For units corresponding to interval variables in input or output layers, the name of the unit is the same as the name of the variable. For units corresponding to dummy variables for categorical (nominal or ordinal) inputs or targets, the name of each unit is constructed by concatenating the name of the input or target variable with the value of the category, truncating as necessary to make the length of the name eight characters or less. For hidden units, the names are constructed by concatenating the layer ID with an integer.
The NEURAL Procedure

Procedure Syntax

PROC NEURAL <option-list>;

ARCHITECTURE architecture-name
<HIDDEN=n>
<DIRECT>;

CODE FILE=file-name
<FORMAT=format>
<RESIDUAL|NORESIDUAL>
<ERROR|NOERROR>
<GROUP=name>;

CONNECT id-list/
<RANDIST=name>
<RANLOC=number>
<RANSCALE=number>;

CUT id-list| ALL;

DECISION DECDATA=<libref.>SAS-data-set
<DECVARS=decision-variable(s)> <option(s)>;

DELETE id-list| ALL;

FREEZE weight-list /
<VALUE=number>
<EST=<libref.>SAS-data-set>;

FREQ variable(s);

HIDDEN integer /
ID=name
<ACT=keyword>
<BIAS|NOBIAS>
<COMBINE=keyword>;
INITIAL INEST=libref.SAS-data-set
OUTEST=libref.SAS-data-set
<BIAJUST=adjustment-value>
<INFAN=number>
<RANDBIAS|NORANDBIAS >
<RANDOUT|NORANDOUT >
<RANDOM=integer>
<RANDSCALE | NORANDSCALE>;

INPUT variable-list /
ID=name
<LEVEL=value>
<STD=method>;

NETOPTIONS network-option(s);

NLOPTIONS <nonlinear-options>;

PERTURB weight-list /
OUTEST=libref.SAS-data-set
DF=number
<RANDIST=name>
<RANDOM=integer>
<RANLOC=number>
<RANSCALE=number>;

PRELIM integer
INEST=libref.SAS-data-set
OUTEST=libref.SAS-data-set
<ACCELERATE=number>
<DECELERATE=number>
<LEARN=number>
<MAXLEARN=number>
<MAX | MAXMOMENTUM=number>
<MINLEARN=number >
<MOM | MOMENTUM=number>
<preiter=integer>
<pretech=name>
<pretime=number>
<RANDBIAS|NORANDBIAS >
<RANDOUT|NORANDOUT >
<RANDOM=integer>;

QUIT;
RANOPTIONS connection-list /
    <RANDIST=name>
    <RANDOM=integer>
    <RANLOC=number>
    <RANSCALE=number>;

SAVE OUTEST=<libref.>SAS-data-set
NETWORK=screen-specification;

SCORE DATA=<libref.>SAS-data-set
OUT=<libref.>SAS-data-set
OUTFIT=<libref.>SAS-data-set
<DUMMIES|NODUMMIES>
<Role=role-option>;

SET weight-list number;

SHOW weights;

TARGET variable-list /
    <ACT=keyword>
    <BIAS|NOBIAS>
    <COMBINE=keyword>
    <ERROR=keyword>
    <ID=name>
    <LEVEL=value>
    <MESTA=number>
    <MESTCON=number>
    <SIGMA=number>
    <STD=method>
    <WEIGHT=number>;

THAW weight-list;
TRAIN OUT = <libref.> SAS-data-set
OUTEST = <libref.> SAS-data-set
OUTFIT = <libref.> SAS-data-set
<ACCEL|ACCELERATE = number>
<DECEL|DECELERATE = number>
<DECAY = number>
<DUMMIES|NODUMMIES>
<ESTITER = i>
<LEARN = number>
<MAX|MAXMOMENTUM = number>
<MAXITER = integer>
<MAXLEARN = number>
<MAXTIME = number>
<MINLEARN = number>
<MOM|MOMENTUM = number>
<TECHNIQUE = name> ;

USE <libref.> SAS-data-set;
PROC NEURAL Statement

Invokes the NEURAL procedure.

PROC NEURAL <option-list>;

Required Arguments

DATA=<libref.SAS-data-set>
  Specifies the DMDB-encoded input SAS data set containing the training data.

DMDBCAT=<libref.SAS-catalog>
  Specifies the DMDB catalog.

Options

GRAPH
  Plots the objective function, validation error, and test error during training.

NETWORK=<screen-specification>
  Constructs a network according to a description that was saved by using a SAVE statement during a previous execution of the NEURAL procedure. <screen-specification> is the catalog entry that was specified in the SAVE statement.

  Default: None

RANDOM=integer
  Specifies the random number seed used in network weight initialization.

  Default: 12345

CAUTION:
The weights and predicted outputs from the network cannot be reproduced when a 0 or negative RANDOM= value is specified. When a 0 or negative value is specified, the system clock is used to generate the seed. The actual value of this seed will be unavailable, and you lose control over the initialization of weights. Different initializations will result in different final weights and predicted outputs for repeated runs of the same set of NEURAL statements and same input data sets.

STOPFILE='file path name'
  This option enables you to stop the NEURAL training when you are running a large job. Before
you invoke the NEURAL procedure, specify the file path name in the STOPFILE option. For example, STOPFILE= "c:\mydir\haltneural". Initially, this file should not exist. The NEURAL procedure checks for the existence of this file between iterations in the training process. When you want to stop the job, create the specified file, and the NEURAL procedure will halt the training at the current iteration. The file does not have to contain any contents.

TESTDATA=<libref.>SAS-data-set

   Specifies a data set used to compute the test average error during training. At selected iterations (controlled by ESTITER=), each observation in the TESTDATA= data set is read in, scored using the current network weight values, and the error computed. The average test error is then output to the OUTEST= data set.

   Note:  This requires the TESTDATA= data set to contain the inputs and target variables.

VALIDATA=<libref.>SAS-data-set

   Specifies a data set used to compute the validation average error during training. At selected iterations (controlled by ESTITER=), each observation in the VALIDATA= data set is read in, scored using the current network weight values, and the error computed. The average validation error is then output to the OUTEST= data set.

   Note:  This requires the VALIDATA= data set to contain the inputs and target variables.
The NEURAL Procedure

ARCHITECTURE Statement

Constructs a network with 0 or 1 hidden layers, sets the hidden-unit ACT= and COMBINE= options, and sets default values for various other options as described below.

Interaction: You **cannot** override the hidden-unit ACT= and COMBINE= options implied by the ARCHITECTURE statement, because these are what define the architecture. You **can** override all the other values set by ARCHITECTURE by using an INPUT, a HIDDEN, a TARGET, or a RANOPTIONS statement.

Alias: ARCH

**ARCHITECTURE** *architecture-name*

*<HIDDEN=n>*

*<DIRECT>*;

**Required Arguments**

*architecture-name*

Names the architecture you want to use to construct the network. Only one *architecture-name* from the following list can be specified:

- **GLIM** Requests a Generalized Linear Model.
- **MLP** Requests a Multilayer Perceptron.
- **ORBFEQ** Requests an Ordinary Radial Basis Function Network with Equal Widths.
- **ORBFUN** Requests an Ordinary Radial Basis Function Network with Unequal Widths.
- **NRBFEQ** Requests a Normalized Radial Basis Function Network with Equal Widths and Heights.
- **NRBFEH** Requests a Normalized Radial Basis Function Network with Equal Heights and Unequal Widths.
- **NRBFEW** Requests a Normalized Radial Basis Function Network with Unequal Heights and Equal Widths.
- **NRBFEV** Requests a Normalized Radial Basis Function Network with Equal Volume.
- **NRBFUN** Requests a Normalized Radial Basis Function Network with Unequal Heights and Unequal Widths.
**Note:** See the following two tables for INPUT, TARGET, and HIDDEN options implied by architecture name and RANOPTIONS implied by architecture name.

### INPUT, TARGET, and HIDDEN Options Implied by Architecture Name

<table>
<thead>
<tr>
<th>ARCHITECTURE NAME</th>
<th>INPUT Options</th>
<th>TARGET Options</th>
<th>HIDDEN Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLIM</td>
<td>STD=None</td>
<td>STD=None</td>
<td>No hidden layers</td>
</tr>
<tr>
<td>MLP</td>
<td></td>
<td></td>
<td>ACT=TANH, COMBINE=LINEAR</td>
</tr>
<tr>
<td>ORBFEQ</td>
<td></td>
<td>ACT=EXP</td>
<td>COMBINE=EQRADIAL</td>
</tr>
<tr>
<td>ORBFUN</td>
<td></td>
<td>ACT=EXP</td>
<td>COMBINE=EHRADIAL</td>
</tr>
<tr>
<td>NRBFEQ</td>
<td>NOBIAS</td>
<td>ACT=SOFTMAX</td>
<td>COMBINE=EQRADIAL</td>
</tr>
<tr>
<td>NRBFEH</td>
<td>NOBIAS</td>
<td>ACT=SOFTMAX</td>
<td>COMBINE=EHRADIAL</td>
</tr>
<tr>
<td>NRBFEW</td>
<td>NOBIAS</td>
<td>ACT=SOFTMAX</td>
<td>COMBINE=EWRADIAL</td>
</tr>
<tr>
<td>NRBFEV</td>
<td>NOBIAS</td>
<td>ACT=SOFTMAX</td>
<td>COMBINE=EVRADIAL</td>
</tr>
<tr>
<td>NRBFUN</td>
<td>NOBIAS</td>
<td>ACT=SOFTMAX</td>
<td>COMBINE=XRADIAL</td>
</tr>
</tbody>
</table>

The following definitions apply to the table below:

- **fan_in** specifies the fan_in of a hidden unit, that is, the number of non-bias and non-altitude weights feeding into the unit.
- **n_hidden_units** is the number of hidden units.
- **defloc** is \(2 \times \max(.01, (n\text{\_hidden\_units}^{(1/fan\_in)}))\).
ranloc is the value of the RANLOC= option.

### RANOPTIONS Implied by Architecture Name

<table>
<thead>
<tr>
<th>ARCHITECTURE NAME</th>
<th>RANOPTIONS for BIAS -&gt; HIDDEN Weights</th>
<th>RANOPTIONS for INPUT -&gt; HIDDEN Weights (not affected by early stopping)</th>
<th>RANOPTIONS for ALTITUDE -&gt; HIDDEN Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLIM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MLP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ORBFEQ</td>
<td>RANLOC=defloc RANSCALE=ranloc*.1</td>
<td>RANSCALE=1</td>
<td></td>
</tr>
<tr>
<td>ORBFUN</td>
<td>RANLOC=defloc RANSCALE=ranloc*5</td>
<td>RANSCALE=1</td>
<td></td>
</tr>
<tr>
<td>NRBFEQ</td>
<td>RANLOC=defloc RANSCALE=ranloc*.1</td>
<td>RANSCALE=1</td>
<td></td>
</tr>
<tr>
<td>NRBFEH</td>
<td>RANLOC=defloc RANSCALE=ranloc*.5</td>
<td>RANSCALE=1</td>
<td>RANLOC=1</td>
</tr>
<tr>
<td>NRBFEW</td>
<td>RANLOC=defloc RANSCALE=ranloc*.1</td>
<td>RANSCALE=1</td>
<td>RANSCALE=ranloc*.5</td>
</tr>
<tr>
<td>NRBFEV</td>
<td>RANLOC=.5<em>defloc RANSCALE=ranloc</em>.1</td>
<td>RANSCALE=1</td>
<td></td>
</tr>
<tr>
<td>NRBFUN</td>
<td>RANLOC=defloc RANSCALE=ranloc*.5</td>
<td>RANSCALE=1</td>
<td>RANLOC=1</td>
</tr>
<tr>
<td></td>
<td>RANSCALE=ranloc*.5</td>
<td></td>
<td>RANSCALE=ranloc*.5</td>
</tr>
</tbody>
</table>

### Options

**HIDDEN= n**

specifies the number of hidden units for all architectures other than GLIM.

**Default:** None
DIRECT

Requests direct connections from inputs to outputs.

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The NEURAL Procedure

CODE Statement

To score a data set based on a previously trained neural network outside of PROC NEURAL, you can specify a CODE statement to write SAS DATA step code to a file or catalog entry. This code can then be included into a DATA step that reads (using a SET statement) the data set to be scored.

<table>
<thead>
<tr>
<th>Category</th>
<th>Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.</th>
</tr>
</thead>
</table>

```sas
CODE FILE=file-name
<ERROR|NOERROR>
<FORMAT=format>
<GROUP=name>
<RESIDUAL|NORESIDUAL>;
```

**Required Arguments**

**FILE=file-name**

Specifies the filename to which the scoring code is written.

**Options**

**ERROR | NOERROR**

Specifies whether to generate code to compute the error function.

- **Default:** NOERROR

**FORMAT=format**

Specifies the format to use for the weights.

- **Default:** BEST12

**RESIDUAL | NORESIDUAL**

Specifies whether to generate code to compute residuals. If you request code for residuals and then score a data set that does not contain target values, the residuals will have missing values.

- **Default:** NORESIDUAL
A network can be specified without any CONNECT statements. However, such a network will be connected by default as follows. First all input layers are connected to the first hidden layer, then each hidden layer except the last is connected to the next hidden layer. Finally, the last hidden layer is connected to all output layers. If this particular architecture is not appropriate, use one or more CONNECT statements to explicitly define the network connections.

**CONNECT Statement**

** CONNECT id-list;**

**Required Arguments**

**id-list**

Lists the identifiers of two or more layers to connect. The identifiers must have been previously defined by the ID= option in an INPUT, a HIDDEN, or a TARGET statement. Each layer except the last is connected to the next layer in the list. Connections must be feedforward. Loops are not allowed.

For example, the following PROC NEURAL step connects the input layers to the output layer, the input layers to the hidden units, and the hidden units to the output layer.

```plaintext
title 'Fully Connected Network';
proc neural data=mydata dmdbcat=mycat;
   input a b / level= nominal id=nom;
   input x z / level= interval id=int;
   hidden 2 / id= hu;
   target y / level=interval id=tar;
   connect int tar;
   connect nom tar;
   connect int hu;
   connect nom hu;
   connect hu tar;
   train;
run;
```

*Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.*
If the weights corresponding to the connection between two layers are not contributing the predictive ability of the network, you can remove that connection and the corresponding weights by using a CUT statement.

**CUT Statement**

**Category**  Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

**CUT id-list| ALL;**

**Options**

You must specify either:

**id-list**

Specifies the identifiers of the layers to disconnect.

**ALL**

Disconnects all layers.
The NEURAL Procedure

DECISION Statement

Specifies information used for decision processing in the DECIDE, DMREG, NEURAL, and SPLIT procedures. This documentation applies to all four procedures.

Tip: The DECISION statement is required for PROC DECIDE. It is optional for DMREG, NEURAL and SPLIT procedures.

DECISION DECDATA= <libref.> SAS-data-set <DECVARS=decision-variable(s)> <option(s)>;

DECDATA= <libref.> SAS-data-set

Specifies the input data set that contains the decision matrix. The DECDATA= data set must contain the target variable.

Note: The DECDATA= data set may also contain decision variables specified by means of the DECVARS= option, and prior probability variable(s) specified by means of the PRIORVAR= option or the OLDPRIORVAR= option, or both.

The target variable is specified by means of the TARGET statement in the DECIDE, NEURAL, and SPLIT procedures or by using the MODEL statement in the DMREG procedure. If the target variable in the DATA= data set is categorical, then the target variable of the DECDATA= data set should contain the category values, and the decision variables will contain the common consequences of making those decisions for the corresponding target level. If the target variable is interval, then each decision variable will contain the value of the consequence for that decision at a point specified in the target variable. The unspecified regions of the decision function are interpolated by a piecewise linear spline.

Tip: The DECDATA= data set may be of TYPE=LOSS, PROFIT, OR REVENUE. If unspecified, TYPE=PROFIT is assumed by default. TYPE= is a data set option that should be specified when the data set is created.

DECVARS=decision-variable(s)

Specifies the decision variables in the DECDATA= data set that contain the target-specific consequences for each decision.

Default: None

COST=cost-option(s)

Specifies numeric constants that give the cost of a decision, or variables in the DATA= data set that contain the case-specific costs, or any combination of constants and variables. There must be the same number of cost constants and variables as there are decision variables in the DECVARS=
option. In the COST= option, you may not use abbreviated variable lists such as D1-D3, 
ABC--XYZ, or PQR:

| Default:          | All costs are assumed to be 0. |

**CAUTION:**

The COST= option may only be specified when the DECDATA= data set is of 
TYPE=REVENUE.

**PRIORVAR=** *variable*

Specifies the variable in the DECDATA= data set that contains the prior probabilities to use for 
making decisions.

| Tip:                     | In the DECIDE procedure, if PRIORVAR= is specified, OLDPRIORVAR= must also be specified. |

| Default:          | None |

**OLDPRIORVAR=** *variable*

Specifies the variable in the DECDATA= data set that contains the prior probabilities that were 
used when originally fitting the model.

| Tip:                     | If OLDPRIORVAR= is specified, PRIORVAR= must also be specified. |

**CAUTION:**

OLDPRIORVAR= is not allowed in PROC SPLIT.

| Default:          | None |

*Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.*
DELETE Statement

If an input or hidden layer is not contributing the predictive ability of the network, you can remove that layer with a DELETE statement. The DELETE statement also removes all associated weights.

| Category | Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement. |

```
DELETE id-list;
```

Required Argument

`id-list`

Specifies the identifiers of layers to delete.
FREEZE Statement

Normally during training, all weights are updated. If you freeze one or more weights, those weights will retain their frozen value until a corresponding THAW statement is executed. Freezing weights causes training to proceed faster and require less memory.

Category

Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

FREEZE weight-list / <VALUE=number>
<EST=<libref.>SAS-data-set>;

Required Argument

weight-list

List of weights to freeze.

Weight-list consists of 0 or more repetitions of:

wname --> wname-2 where:

wname

is a unit name, a layer identifier, BIAS, or ALTITUDE

wname-2

is a unit name or a layer identifier

Options

You can specify either VALUE= or EST= but not both. If neither option is specified, the weights are frozen to their current values.

VALUE=number

Specifies the numeric value to which weights are to be frozen.

EST=<libref.>SAS-data-set

Specifies the SAS data set containing the values to which weights are to be frozen.

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
FREQ Statement

Specifies the frequency variable for training.

Category Variable Statement - specifies variables.

FREQ variable;

Options

variable

Specifies the frequency variable. (The frequency variable can contain integer and non-integer values.)

Note: The FREQ variable is not required in the DATA= data set. The NEURAL procedure searches for the name of the FREQ variable in the DATA=, VALIDATA=, and TESTDATA= data sets. If the FREQ variable does not appear in any of these data sets, then the procedure issues a warning but continues processing. For any data set that does not contain the FREQ variable, a FREQ value of 1 is used for all observations.

Default: The default value is taken from the DMDB metadata. The FREQ statement overrides whatever is in the DMDB metadata. If a FREQ statement contains no variable name, then no FREQ variable is used.

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
**HIDDEN Statement**

You can specify as many HIDDEN statements as you want up to the limits imposed by computer memory, time, and disk space. The hidden layers can be connected in any feedforward pattern using CONNECT statements.

**Category**  
Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

```
HIDDEN integer / ID=name  
<ACT=activation-function>  
<BIAS|NOBIAS>  
<COMBINE=combination-function>;  
```

**Required Arguments**

- **integer**  
  Specifies the number of units in the hidden layer.

- **ID=name**  
  Specifies the identifier for the layer.

**Options**

- **ACT=activation-function**  
  Specifies the activation function. See [ACTIVATION FUNCTIONS](#).

  - **Default:**  
    For hidden units, the default activation function depends on the combination function and on the number of units in the layer.
    
    For COMBINE=ADD, the default is ACT=IDENTITY.
    
    For COMBINE=LINEAR or EQSLOPES, the default is ACT=TANH.
    
    For COMBINE=EHRADIAL, EQRADIAL, EVRADIAL, EWRADIAL, or XRADIAL, the default is ACT=EXP if there is only one hidden unit in the layer; otherwise the default is ACT=SOFTMAX.

- **BIAS | NOBIAS**  
  Specifies whether to use bias.
**Default:** BIAS

**COMBINE**=*combination-function*

Specifies the combination function. See [COMBINATION FUNCTIONS](#).
# Activation Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Range</th>
<th>Function (of Net Input ( t ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDENTITY</td>
<td>((-\infty, +\infty))</td>
<td>( t )</td>
</tr>
<tr>
<td>LINEAR</td>
<td>((-\infty, +\infty))</td>
<td>( t )</td>
</tr>
<tr>
<td>EXPONENTIAL</td>
<td>((0, \infty))</td>
<td>( \exp^t )</td>
</tr>
<tr>
<td>RECIPROCAL</td>
<td>((0, \infty))</td>
<td>( \frac{1}{t} )</td>
</tr>
<tr>
<td>SQUARE</td>
<td>([0, +\infty))</td>
<td>( t^2 )</td>
</tr>
<tr>
<td>LOGISTIC</td>
<td>((0, 1))</td>
<td>( \frac{1}{1+e^{-t}} )</td>
</tr>
<tr>
<td>MLOGISTIC</td>
<td>((0, 1))</td>
<td>( \frac{e^t}{\sum \text{exponentials}} )</td>
</tr>
<tr>
<td>SOFTMAX</td>
<td>((0, 1))</td>
<td>( \frac{e^t}{\sum \text{exponentials}} )</td>
</tr>
<tr>
<td>GAUSS</td>
<td>((0, 1])</td>
<td>( e^{-t^2} )</td>
</tr>
<tr>
<td>SINE</td>
<td>([0, 1])</td>
<td>( \sin(t) )</td>
</tr>
<tr>
<td>COSINE</td>
<td>([0, 1])</td>
<td>( \cos(t) )</td>
</tr>
<tr>
<td>ELLIOTT</td>
<td>((-1, 1))</td>
<td>( \frac{t}{(1+\text{abs}(t))} )</td>
</tr>
<tr>
<td>TANH</td>
<td>((-1, 1))</td>
<td>( \tanh(t) = 1 - \frac{2}{1+e^{2t}} )</td>
</tr>
<tr>
<td>ARCln</td>
<td>$(-1, 1)$</td>
<td>$\arctan(t) \times \frac{2}{\pi}$</td>
</tr>
</tbody>
</table>
A combination function combines the values received from preceding nodes into a single number called the net input. Both output and hidden layers are assigned combination functions.

The following combination functions are available.

**Add**

Adds all the incoming values without using any weights or biases.

**Linear**

Is a linear combination of the incoming values and weights.

**EQSlopes**

Is identical to the Linear combination function, except that the same connection weights are used for each unit in the layer, although different units have different biases. EQSlopes is mainly used for ordinal targets.

**EQRadial**

Is a radial basis function with equal heights and widths for all units in the layer.

**EHRadial**

Is a radial basis function with equal heights but unequal widths for all units in the layer.

**EWRadial**

Is a radial basis function with equal widths but unequal heights for all units in the layer.

**EVRadial**

Is a radial basis function with equal volumes for all units in the layer.

**XRadial**

Is a radial basis function with unequal heights and widths for all units in the layer.

The following definitions apply to the Table of Combination Functions:

**All summations**

Are divided by the net inputs indexed by $i$.

$alt_j$

The altitude of the $j$th unit

$bias_j$

The width (bias) of the $j$th unit

$bias$

A common bias shared by all units in the layer
The fan-in of the $j$th unit

The weight connecting the $i$th incoming value to the $j$th unit

The common weight for the $i$th input shared by all units in the layer

The $i$th incoming value

**Combination Functions**

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>DEFINITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>$\sum_{i} x_i$</td>
</tr>
<tr>
<td>LINear</td>
<td>$bias_j + \sum_{i} w_{ij} x_i$</td>
</tr>
<tr>
<td>EQSlopes</td>
<td>$bias_j + \sum_{i} w_i x_i$</td>
</tr>
<tr>
<td>XRA radial</td>
<td>$f \log (altb_j) - bias_{j}^{2} \sum_{i} [(w_{ij} - x_{i})^2]$</td>
</tr>
<tr>
<td>EHR radial</td>
<td>$bias_{j}^{2} [(w_{ij} - x_{i})^2]$</td>
</tr>
<tr>
<td>EVR radial</td>
<td>$f \log (bias_j) - bias_{j}^{2} [(w_{ij} - x_{i})^2]$</td>
</tr>
<tr>
<td>EWR radial</td>
<td>$f \log (altb_j) - bias^{2} [(w_{ij} - x_{i})^2]$</td>
</tr>
<tr>
<td>EQR radial</td>
<td>$-bias^{2} [(w_{ij} - x_{i})^2]$</td>
</tr>
<tr>
<td>RAD</td>
<td>defaults to EHRadial</td>
</tr>
</tbody>
</table>
The NEURAL Procedure

INITIAL Statement

After a network has been defined in terms of input, hidden and output layers, all weights and biases in the network must be given initial values before any training is performed. PROC NEURAL will by default supply appropriate random or computed values for these quantities. If you train a network without supplying an INITIAL or USE statement, the network will be initialized using the default specifications.

<table>
<thead>
<tr>
<th>Category</th>
<th>Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.</th>
</tr>
</thead>
</table>

INITIAL <BIADJUST=adjustment-value>
<INEST=<libref.>SAS-data-set>
<INFAN=number>
<OUTEST=<libref.>SAS-data-set>
<RANDBIAS|NORANDBIAS>
<RANDOUT|NORANDOUT>
<RANDOM=integer>
<RANDSCALE | NORANDSCALE>;

Options

BIADJUST= adjustment-value

Specifies how to adjust the random biases for units with a LINEAR combination function. A random bias is adjusted by multiplying by the function of the weights indicated by adjustment-value, and dividing by the scale (RANSSCALE=) of the distribution from which the random bias was drawn. adjustment value can be one of the following:

SUM

Adjusts random initial biases for the sum of the absolute connection weights leading into the unit. This value is typically used with STD=MIDRANGE for inputs and RANDIST=UNIFORM.

USS

Adjusts random initial biases for the square root of the sum of squared connection weights leading into the unit. This value is typically used with STD=STD for inputs and RANDIST=NORMAL.

NONE|NO

No bias adjustment.
Default: BIADJUST=NONE

INEST=libref.SAS-data-set

Specifies an input data set that contains some or all of the weights. Any weights in the INEST= data set that have missing values are assigned values according to the RANDOM=, RANDOUT, and RANDBIAS options, as well as the options that pertain to random number distributions that you specify in the Random statements. An INEST= data set will typically have been created by using the OUTEST= option in a SAVE or a TRAIN statement from a previous execution of the NEURAL procedure.

INFAN=number

Divide random connection weights by

\[(fan-in \text{ of unit})^\text{number}\]

where the "fan-in" of a unit is the number of other units feeding into that unit, not counting the bias or altitude.

Default: 0 for radial combination functions, otherwise .5

Range: between 0 and 1

OUTEST=libref.SAS-data-set

Specifies the output data set that contains all the initial weights.

RANDBIAS | NORANDBIAS

Specifies whether to randomize output biases.

Note: NORANDBIAS overrides whatever you specify in the RANOPTIONS statement.

Default: NORANDBIAS, which sets bias to the inverse activation function of the target mean.

RANDOM=integer

Specifies the random number seed.

Default: 0

RANDOUT | NORANDOUT

Specifies whether to randomize the output connection weights.

Note: NORANDOUT overrides whatever you specify in the RANOPTIONS statement.

Default: NORANDOUT, which sets weights to 0.

RANDSCALE | NORANDSCALE

Specifies whether to randomize target scale estimates.
<table>
<thead>
<tr>
<th>Default:</th>
<th>NORANDSCALE, which sets each scale estimate to the standard deviation of the corresponding target variable.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note:</td>
<td>NORANDSCALE overrides whatever is specified in the RANOPTIONS statement.</td>
</tr>
</tbody>
</table>
INPUT Statement

The INPUT statement allows you to group together input variables having common levels and standardizations. You can specify as many INPUT statements as you want up to the limits imposed by computer memory, time, and disk space. The input layers can be connected to hidden or output layers using CONNECT statements.

**Category**

Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

**INPUT** variable-list / ID=name

<LEVEL=value>

<STD=method>;

**Required Arguments**

*variable-list*

Specifies the input variables.

*ID=name*

Specifies the identifier for the layer.

**Options**

**LEVEL=value**

Specifies the measurement level, where *value* can be:

- NOMINAL|NOM
  - Nominal
- ORDINAL|ORD
  - Ordinal
- INTERVAL|INT
  - Interval

**Default:** Interval (for variables specified by a VAR statement in the DMDB procedure) or nominal (for variables specified by a CLASS statement in the DMDB procedure).

**STD=method**

Specifies the standardization method, where *method* is:
NONE

Variables are not altered.

STD

Variables are rescaled to have a mean of 0 and a standard deviation of 1.

RANGE

Variables are rescaled to have a minimum of 0 and a range of 1. This standardization is not recommended for input variables.

MIDRANGE

Variables are rescaled to have a midrange of 0 and a half-range of 1 (that is, a minimum of -1 and a maximum of 1).

**Default:** STD

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The NEURAL Procedure

NETOPTIONS Statement

Identifies the network options to set.

**Category**
Option Statement - does not directly affect the network, but sets options for use in subsequent action statements. The options persist until reset at a later stage in the processing.

**Alias:** NETOPTS | NETOPT

NETOPTIONS network-option(s);

**Network Options**

**DECAY=** *number*

Specifies the weight decay.

- **Range:** *number* ≥ 0
- **Default:** For the QPROP optimization technique: .0001; for all others: 0

**INVALIDTARGET=** *action*

Specifies the action taken during training if an out-of-range target value is found, where *action* can be:

- **OMITCASE | OMIT** If INVALIDTARGET = OMITCASE is specified, and an invalid target value is found in the training data set, a warning is given, the observation is not used, but the training will continue.
- **STOP** If INVALIDTARGET = STOP is specified, an error is issued, and training is terminated.

**Example:** If ERROR = GAMMA is specified in a target statement, the target values should be positive. If a zero or negative value is found for the variable(s) listed in the target statement, the observation containing that value cannot be used for training. The training either continue with the remaining valid observations or stops depending on the INVALIDTARGET= specification. Note, however, if the \( \text{\textbackslash it mean} \) over the training data set, of the target variable(s) is zero or negative, an error is issued and the training is stopped, regardless of the INVALIDTARGET= specification.
The default is INVALIDTARGET = STOP. INVALIDETARGET can be abbreviated as INVALIDTARG or INVTARG.

**OBJECT=objective-function**

Specifies the objective function where *objective-function* can be one of the following:

**DEV**

Requests deviance (for ERROR=NORMAL, this is least squares).

**LIKE**

Requests negative log-likelihood.

**MEST**

Requests M estimation.

**Default:**

Depends on the error functions that the network uses. To determine the default, examine the table below named Errors by Objective Functions. Scan the table from left to right. The default is the first column that contains a "yes" in every row corresponding to an error function used in the network. If no such column exists in the table, an error message is issued and the network cannot be trained.

<table>
<thead>
<tr>
<th>Errors by Objective Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ERRORS</strong></td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Normal</td>
</tr>
<tr>
<td>Cauchy</td>
</tr>
<tr>
<td>Logistic</td>
</tr>
<tr>
<td>Huber</td>
</tr>
<tr>
<td>Biweight</td>
</tr>
<tr>
<td>Wave</td>
</tr>
<tr>
<td>Gamma</td>
</tr>
<tr>
<td>Poisson</td>
</tr>
<tr>
<td>Bernoulli</td>
</tr>
<tr>
<td>Binomial</td>
</tr>
<tr>
<td>Entropy</td>
</tr>
</tbody>
</table>
RANDF=number

Specifies the degrees of freedom parameter for random numbers. See the following Randomization Options and Default Parameters table.

RANDIST=name

Specifies the type of distribution to be used for random initial weights and perturbations. The distributions and default parameter values are as follows:

Randomization Options and Default Parameters

<table>
<thead>
<tr>
<th>RANDIST</th>
<th>RANLOC</th>
<th>RANScale</th>
<th>DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORMAL</td>
<td>mean=0</td>
<td>std=1</td>
<td></td>
</tr>
<tr>
<td>UNIFORM</td>
<td>mean=0</td>
<td>halfrange=1</td>
<td>-</td>
</tr>
<tr>
<td>CAUCHY</td>
<td>median=0</td>
<td>scale=1</td>
<td></td>
</tr>
<tr>
<td>CHIINV</td>
<td>-</td>
<td>scale=1</td>
<td>df=1</td>
</tr>
</tbody>
</table>

Default: NORMAL

RANDOM=integer

Specifies the random number seed.

Default: 0

RANLOC=number

Specifies the location parameter for random numbers. See the above Randomization Options and Default Parameters table.

Specifies the scale parameter for random numbers. See the above Randomization Options and Default Parameters table.
The NEURAL Procedure

**NLOPTIONS Statement**

Identifies the nonlinear optimization options to set.

**Category** Option Statement - does not directly affect the network, but sets options for use in subsequent action statements. The options persist until reset at a later stage in the processing.

**NLOPTIONS** <nonlinear-options>;

**Nonlinear Options**

**ABSCONV= number**

Specifies an absolute function convergence criterion. ABSCONV= is a function of the log-likelihood for the intercept-only model.

**Default:** The default value is the negative square root of the largest double precision value.

**Range:** number > 0

**ABSFCONV= number**

Specifies an absolute function convergence criterion.

**Default:** 0

**Range:** number > 0

**ABSGCONV= number**

Specifies the absolute gradient convergence criterion.

**Default:** 1E-5

**Range:** number > 0

**ABSXCONV= number**

Specifies the absolute parameter convergence criterion.

**Default:** 0

**Range:** number > 0
DAMPSTEP= number
Specifies that the initial step size value for each line search used by the QUANEW, CONGRA, or NEWRAP optimization technique cannot be larger than the product of number and the step size value used in the former iteration.

| Default: | 2 |
| Range:   | number > 0 |

DIAHES
Forces the optimization algorithm (TRUREG, NEWRAP, or NRRIDG) to take advantage of the diagonality.

FCONV= number
Specifies a function convergence criterion.

| Default: | 1E-4 |
| Range:   | number > 0 |

FSIZE= number
Specifies the parameter of the relative function and relative gradient termination criteria.

| Default: | Not applicable. |
| Range:   | number ≥ 0 |

GCONV= number
Specifies the relative gradient convergence criterion.

| Default: | 1E-8 |
| Range:   | number > 0 |

HESCAL= 0 | 1 | 2 | 3
Specifies the scaling version of the Hessian or cross-product Jacobian matrix used in NRRIDG, TRUREG, LEVMAR, NEWRAP, or DBLDOG optimization.

| Default: | 1 - for LEVMAR minimization technique
0 - for all others |

INHESSIAN= number
Specifies how to define the initial estimate of the approximate Hessian for the quasi-Newton techniques QUANEW and DBLDOG.
Default: The default is to use a Hessian based on the initial weights as the initial estimate of the approximate Hessian. When r=0, the initial estimate of the approximate Hessian is computed from the magnitude of the initial gradient.

| Range: | number > 0 |

INSTEP= number
Specifies the initial radius of the trust region used in the TRUREG, DBLDOG, and LEVMAR algorithms.

| Default: | 1 |
| Range: | number > 0 |

LCEPS | LCEPSILON= number
Specifies the range for active constraints.

| Range: | number > 0 |

LCSINGULAR= number
Specifies the tolerance for dependent constraints.

| Range: | number > 0 |

LINESEARCH= number
Specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques.

| Default: | 2 |
| Range: | 1 ≤ number ≤ 8 |

LSPRECISION= number
Specifies the degree of accuracy that should be obtained by the second and third line-search algorithms.
**Line-Search Precision Values**

<table>
<thead>
<tr>
<th>TECHNIQUE=</th>
<th>UPDATE=</th>
<th>LSPRECISION VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUANEW</td>
<td>DBFGS, BFGS</td>
<td>0.4</td>
</tr>
<tr>
<td>QUANEW</td>
<td>DDFP, DFP</td>
<td>0.06</td>
</tr>
<tr>
<td>CONGRA</td>
<td>all</td>
<td>0.1</td>
</tr>
<tr>
<td>NEWRAP</td>
<td>no update</td>
<td>0.9</td>
</tr>
</tbody>
</table>

**MAXFUNC= number**

Specifies the maximum number of function calls in the optimization process.

**Default:** 2147483647 for all techniques

**Range:** number > 0

**MAXITER= number**

Specifies the maximum number of iterations in the optimization process.

**Default:**
- 100 for TRUREG, NRRIDG, NEWRAP, and LEVMAR
- 200 for QUANEW and DBLDOG
- 400 for CONGRA

**Range:** number > 0

**MAXSTEP= number**

Specifies the upper bound for the step length of the line-search algorithms.

**Default:** The largest double precision value

**Range:** number > 0

**MAXTIME= number**

Specifies the upper limit of CPU time for the optimization process. It is measured in seconds.
### MINITER= number

Specifies the minimum number of iterations in the optimization process.

- **Default:** 0
- **Range:** \( number \geq 0 \)

### NOPRINT

Suppresses all output. Only ERRORs, WARNINGs, and NOTEs are printed on the log file.

### PALL

Prints all optional output except the output generated by the PSTDERR, LIST, or LISTCODE option.

### PSUMMARY

Restricts the amount of default printed output to a short form of iteration history and NOTEs, WARNINGs, and ERRORs.

### PHISTORY

Prints the optimization history. If PSUMMARY or NOPRINT are not specified, then the PHISTORY option is set automatically and the iteration history is printed by default.

### RESTART= number

Specifies that the QUANEW or CONGRA algorithm is restarted with a steepest descent/ascent search direction after the number of iterations has been completed.

- **Range:** \( number \geq 1 \)
- **Default:**
  - For TECHNIQUE=CONGRA, and UPDATE= PB, restart is done automatically, so number is not used;
  - For TECHNIQUE=CONGRA, and UPDATE not = PB, number is the number of parameters.
  - For TECHNIQUE=QUANEW, number is the largest integer.

### SINGULAR= number

Specifies an absolute singularity criterion for the computation of the inertia of Hessian and cross-product Jacobian and their projected forms.

- **Default:** 1E-8
- **Range:** \( number > 0 \)
**TECHNIQUE= method**

Specifies an optimization technique, where *method* is one of the following:

**CONGRA**

Specifies the Conjugate Gradient optimization technique. This is the default when the number of parameters to be estimated is \( \geq 400 \).

**DBLDOG**

Specifies the Double-Dogleg optimization technique.

**NEWRAP**

Specifies the Newton-Raphson with Line Search optimization technique.

**NRRIDG**

Specifies the Newton-Raphson with Ridging optimization technique. This is the default when the number of parameters to be estimated is \( \leq 40 \).

**QUANEW**

Specifies the quasi-Newton optimization technique. This is the default when the number of parameters to be estimated is in the range \( 40 < n \leq 400 \).

**TRUREG**

Specifies the Trust-Region optimization technique.

**UPDATE=update-type**

Specifies an update method, where *update-type* is one of the following:

**BFGS**

For **TECHNIQUE= QUANEW**, performs the BFGS (Broyden-Fletcher-Goldfarb-Shanno) update of the Cholesky factor of the Hessian matrix.

**CD**

For **TECHNIQUE=CONGRA**, performs a conjugate descent update of Fletcher.

**DBFGS**

For **TECHNIQUE= DBLDOG** or **QUANEW**, performs the dual BFGS (Broyden-Fletcher-Goldfarb-Shanno) update of the Cholesky factor of the Hessian matrix. This is the default for **TECHNIQUE=QUANEW** and **DBLDOG**.

**DDFP**

For **TECHNIQUE= DBLDOG** or **QUANEW**, performs the dual DFP (Davidson-Fletcher-Powell) update of the Cholesky factor of the Hessian matrix.

**DFP**

For **TECHNIQUE= QUANEW**, performs the original DFP (Davidson-Fletcher-Powell) update of the inverse Hessian matrix.

**FR**

For **TECHNIQUE=CONGRA**, performs the Fletcher-Reeves update.

**PB**
For TECHNIQUE=CONGRA, performs the automatic restart update method of Powell and Beale. This is the default for TECHNIQUE= CONGRA.

PR

For TECHNIQUE=CONGRA, performs the Polak-Ribiere update.

VERSION= 1 | 2 | 3

Specifies the version of the hybrid quasi-Newton optimization technique or the version of the quasi-Newton optimization technique with nonlinear constraints.

| Default: | 2 |

XCONV= number

Specifies the relative parameter convergence criterion.

| Default: | 0 |
| Range: | number > 0 |

XSIZE= number

Specifies the number of successive iterations for which the criterion must be satisfied before the optimization process can be terminated.

| Default: | 0 |
| Range: | number > 0 |

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The NEURAL Procedure

PERTURB Statement

Perturbs weights. Perturbing weights can sometimes allow you to escape a local minimum.

**Category**
Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

**PERTURB** weight-list / **<RANDF>=number >**
<**OUTEST**=<libref.>:SAS-data-set>
<**RANDIST**=name>
<**RANDOM**=integer>
<**RANLOC**=number>
<**RANSCALE**=number>;

**Required Argument**

weight-list

List of weights to freeze.

Weight-list consists of 0 or more repetitions of:

\textit{wname} --\textit{wname-2} where:

- \textit{wname}
  - is the unit name, the layer ID, BIAS, or ALTITUDE

- \textit{wname-2}
  - is the unit name or layer ID

**Options**

**RANDF=number**

Specifies the degrees of freedom parameter for random numbers. See the Randomization Options and Default Parameters table for values.

**OUTEST=<libref.>:SAS-data-set**

Specifies the output data set containing all the weights.

Default: none

**RANDIST=**name
Specifies the type of distribution for random numbers. See the Randomization Options and Default Parameters table for values.

**RANDOM=integer**

Specifies the random number seed.

| Default: | 0 |

**RANLOC=number**

Specifies the location parameter for random numbers. See the Randomization Options and Default Parameters table for values.

**RANSCALE=number**

Specifies the scale parameter for random numbers. See the Randomization Options and Default Parameters table for values.
The NEURAL Procedure

PRELIM Statement

Performs preliminary training to reduce the risk of bad local optima. The final weights and biases in a trained network depend on the initial values. The PRELIM statement repeatedly trains a network for a small number of iterations (default 20) using different initializations. The final weights of the best trained network are then used to initialize a subsequent TRAIN statement.

| Category | Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement. |

PRELIM integer
<ACCELERATE=number>
<DECELERATE=number>
<INEST=<libref.>SAS-data-set>
<LEARN=number>
<MAXLEARN=number>
<MAX | MAXMOMENTUM=number>
<MOM | MOMENTUM=number>
<OUTEST=<libref.>SAS-data-set>
<preiter=integer>
<pretech=name>
<pretime=number>
<RANDBIAS|NORANDBIAS>
<RANDOUT|NORANDOUT>
<RANDOM=integer>;

Required Argument

ingeger
    Specifies the number of preliminary optimizations.

Options

ACCEL | ACCELERATE=number
    Specifies the rate of increase of learning for the RPROP optimization technique.

    Range: number > 1
DECEL | DECELERATE=\texttt{number}  
Specifies the rate of decrease of learning for the RPROP optimization technique.  
\textbf{Range:} \quad 0 < \texttt{number} < 1  
\textbf{Default:} \quad 0.5

INEST=\texttt{libref.SAS-data-set}  
Specifies the input data set that contains some or all weights. Any weights in the INEST= data set that have missing values are assigned values according to the RANDOM=, RANDOUT, and RANDBIAS options, as well as the options that pertain to random number distributions that you specify in the Random statements.

LEARN=\texttt{number}  
Specifies the learning rate for BPROP or the initial learning rate for QPROP and RPROP.  
\textbf{Range:} \quad \texttt{number} > 0  
\textbf{Default:} \quad 0.1

MAXLEARN=\texttt{number}  
Specifies the maximum learning rate for RPROP.  
\textbf{Range:} \quad \texttt{number} > 0  
\textbf{Default:} \quad \text{Reciprocal of the square root of the machine epsilon}

MAXMOM | MAXMOMENTUM=\texttt{number}  
Specifies the maximum momentum for BPROP.  
\textbf{Range:} \quad \texttt{number} > 0  
\textbf{Default:} \quad 1.75

MINLEARN=\texttt{number}  
Specifies the minimum learning rate for RPROP.  
\textbf{Range:} \quad \texttt{number} > 0  
\textbf{Default:} \quad \text{Square root of the machine epsilon}

MOM | MOMENTUM=\texttt{number}  
Specifies the momentum for BPROP.  
\textbf{Range:} \quad 0 \leq \texttt{number} < 1
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OUTEST=&lt;libref.&gt;SAS-data-set</td>
<td>Specifies the output data set that contains all the weights.</td>
</tr>
<tr>
<td>PREITER=integer</td>
<td>Specifies the maximum number of iterations in each preliminary optimization.</td>
</tr>
<tr>
<td></td>
<td>Default: 10</td>
</tr>
<tr>
<td>PRETECH</td>
<td>TECHNIQUE=name</td>
</tr>
<tr>
<td></td>
<td>Default: Same as TECH= in the TRAIN statement.</td>
</tr>
<tr>
<td>PRETIME=number</td>
<td>Specifies the amount of time after which training stops.</td>
</tr>
<tr>
<td>RANDBIAS</td>
<td>NORANDBIAS</td>
</tr>
<tr>
<td></td>
<td>Default: NORANDBIAS, which sets bias to the inverse activation function of the target mean.</td>
</tr>
<tr>
<td></td>
<td>[NORANDBIAS overrides whatever you specify in the RANDOM statement.]</td>
</tr>
<tr>
<td>RANDOM=integer</td>
<td>Specifies the random number seed.</td>
</tr>
<tr>
<td></td>
<td>Default: 0</td>
</tr>
<tr>
<td>RANDOUT</td>
<td>NORANDOUT</td>
</tr>
<tr>
<td></td>
<td>Default: NORANDOUT, which sets weights to 0.</td>
</tr>
<tr>
<td></td>
<td>[NORANDOUT overrides whatever you specify in the RANDOM statement.]</td>
</tr>
</tbody>
</table>

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
QUIT Statement

Stops the procedure.

**Category** Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

QUIT:

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
RANOPTIONS Statement

Specifies distribution of random initial weights.

**Category**
Option Statement - does not directly affect the network, but sets options for use in subsequent action statements. The options persist until reset at a later stage in the processing.

**RANOPTIONS** connection-list /
  <RANDF=number>
  <RANDIST=name>
  <RANDOM=integer>
  <RANLOC=number>
  <RANScale=number>;

**Required Arguments**

**Note:** When a RANOPTIONS statement is executed, the specified options are stored in all the connections listed before the slash. These options are used whenever an INITIAL or PRELIM statement is executed. If you submit two RANOPTIONS statements for the same connection, the second statement overrides all options in the first. In other words, one RANOPTIONS statement does not remember what options were specified in previous RANOPTIONS statements. To have options persist over multiple statements, use the NETOPTIONS statement.

**connection-list**
List of connections to randomize.

**Options**

**RANDF=number**
Specifies the degrees of freedom parameter for random numbers. See the Randomization Options.
RANDIST=name

Specifies the type of distribution for random numbers. See the Randomization Options and Default Parameters table for values.

Default: NORMAL

RANDOM=integer

Specifies the random number seed.

Default: 0

RANLOC=number

Specifies the location parameter for random numbers. See the Randomization Options and Default Parameters table for values.

RANScale=number

Specifies the scale parameter for random numbers. See the Randomization Options and Default Parameters table for values.
The NEURAL Procedure

SAVE Statement

Writes weights to data set or a description of the network to a catalog entry.

<table>
<thead>
<tr>
<th>Category</th>
<th>Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTE:</td>
<td>At least one option must be specified, but there is no single argument that is required.</td>
</tr>
</tbody>
</table>

**SAVE OUTEST=\<libref.\>SAS-data-set**
**NETWORK=screen-specification**;

**Options**

Specify at least one:

**NETWORK=screen-specification**

Saves the definition of the entire network. *screen-specification* is the name of a catalog entry.

**OUTEST=\<libref.\> SAS-data-set**

Saves the network weights in an output data set.

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The NEURAL Procedure

**SCORE Statement**

Creates an output data set containing predicted values and possibly other results such as residuals, classifications, decisions, and assessment values.

<table>
<thead>
<tr>
<th>Category</th>
<th>Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.</th>
</tr>
</thead>
</table>

**SCORE DATA**=<libref.>SAS-data-set
**OUT**=<libref.>SAS-data-set
**OUTFIT**=<libref.>SAS-data-set
<DUMMIES | NODUMMIES>
<ROLE=role-option>;

**Required Arguments**

**OUT**=<libref.>SAS-data-set

Specifies the output data set that contains the outputs.

**Options**

**DATA**=<libref.>SAS-data-set

Specifies the input data to be scored that contains inputs and, optionally, targets.

<table>
<thead>
<tr>
<th>Default:</th>
<th>Defaults to the training data (DATA= in the PROC statement).</th>
</tr>
</thead>
</table>

**DUMMIES | NODUMMIES**

Specifies whether to write dummy variables to the OUT= data set.

<table>
<thead>
<tr>
<th>Default:</th>
<th>NODUMMIES</th>
</tr>
</thead>
</table>

**OUTFIT**=<libref.>SAS-data-set

Specifies the output data set that contains the fit statistics.

**ROLE=role-option**

Specifies the role of the DATA= data set. **ROLE=role-option** primarily affects which fit statistics are computed and what their names and labels are. **Role-option** is one of the following:

**TRAIN**

Specifies that the DATA= data set is the training set. The data set must contain the target variable.
VALID | VALIDATION

Specifies that the DATA= data set is a validation set. The data set must contain the target variable.

TEST

Specifies that the DATA= data set is a test set. The data set must contain the target variable.

SCORE

Specifies that residuals, error functions, and fit statistics are not produced. The data set does not have to contain the target variable.

Default: TEST, except as follows:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>TRAIN</td>
<td>when the DATA= data set in the PROC statement is the same as the DATA= data set in the SCORE statement. Specifying TRAIN with any data set other than the actual training set is an error.</td>
</tr>
<tr>
<td>VALID</td>
<td>when the DATA= data set in the SCORE statement is the same as the VALIDATA= data set in the PROC statement.</td>
</tr>
</tbody>
</table>
Sets the value of the weight-list to number. The SET statement does not freeze weights, so subsequent training may change the values of the weights specified in a SET statement.

**SET Statement**

Sets the value of the weight-list to number. The SET statement does not freeze weights, so subsequent training may change the values of the weights specified in a SET statement.

<table>
<thead>
<tr>
<th>Category</th>
<th>Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.</th>
</tr>
</thead>
</table>

**SET weight-list number;**

**Required Arguments**

- **weight-list**
  - Specifies the list of weights to be affected or changed.

- **number**
  - Specifies the number to which the weight-list is set.
**SHOW Statement**

Prints information about the network.

<table>
<thead>
<tr>
<th>Category</th>
<th>Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NOTE:</strong></td>
<td>At least one option must be specified, but there is no single argument that is required.</td>
</tr>
</tbody>
</table>

**SHOWWEIGHTS STATEMENTS**

**Options**

Specify at least one:

**STATEMENTS**

Prints statements that can be used with the NEURAL procedure to reproduce the network.

**WEIGHTS**

Prints the network weights.

*Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.*
TARGET Statement

Defines an output layer.

**Category**  
Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

**Alias:**  
OUTPUT

```
TARGET | OUTPUT variable-list /
<ACT=activation-function>
<BIAS|NOBIAS >
<COMBINE=combination-function>
<ERROR=keyword>
<ID=name>
<LEVEL=value>
<MESTA=number>
<MESTCON=number>
<SIGMA=number>
<STD=method>;
```

**Required Arguments**

`variable-list`  
Specifies the target variables.

`ID=name`  
Specifies the identifier for the layer.

**Options**

`ACT=activation-function`  
Specifies the activation function. See [Activation Functions](#).

Default: Depends on the measurement level, as follows:

If \(\text{LEVEL} = \text{INTERVAL}\), then the default is IDENTITY.

If \(\text{LEVEL} = \text{ORDINAL}\) then the default is LOGISTIC.

If \(\text{LEVEL} = \text{NOMINAL}\), then the default is \text{MLOGISTIC} (For \text{Error} = \text{MBERNOULLI}, \text{MENTROPY}, \text{or MULTINOMIAL}, the only activation function allowed is \text{MLOGISTIC}).

\[
\text{BIAS | NOBIAS}
\]

Specifies whether to use bias (or not to use bias).

Default: BIAS

\[
\text{COMBINE=}\text{combination-function}
\]

Specifies the combination function. See Combination Functions.

\[
\text{ERROR=}\text{keyword}
\]

Specifies the Error function. Default is NORMAL for \(\text{LEVEL} = \text{INTERVAL}\); otherwise, default is MBERNOULLI. For more information, see the Error Functions table that follows.

### Error Functions

<table>
<thead>
<tr>
<th>KEYWORD</th>
<th>TARGET</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORmal</td>
<td>any</td>
<td>Normal distribution</td>
</tr>
<tr>
<td>CAUchy</td>
<td>any</td>
<td>Cauchy distribution</td>
</tr>
<tr>
<td>LOGistic</td>
<td>any</td>
<td>Logistic distribution</td>
</tr>
<tr>
<td>HUBer</td>
<td>any</td>
<td>Huber M estimator</td>
</tr>
<tr>
<td>BIWeight</td>
<td>any</td>
<td>Biweight M estimator</td>
</tr>
<tr>
<td>WAVe</td>
<td>any</td>
<td>Wave M estimator</td>
</tr>
<tr>
<td>Function</td>
<td>Condition</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>GAMma</td>
<td>$&gt;0$</td>
<td>Gamma distribution</td>
</tr>
<tr>
<td>POIsson</td>
<td>$\geq 0$</td>
<td>Poisson distribution</td>
</tr>
</tbody>
</table>

**Functions with no scale parameter**

<table>
<thead>
<tr>
<th>Function</th>
<th>Condition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERnoulli</td>
<td>0,1</td>
<td>Bernoulli distribution (binomial with one trial)</td>
</tr>
<tr>
<td>BINomial</td>
<td>$\geq 0$</td>
<td>Binomial distribution</td>
</tr>
<tr>
<td>ENTropy</td>
<td>0-1</td>
<td>Cross or relative entropy for independent targets</td>
</tr>
<tr>
<td>MBErnoulli</td>
<td>0,1</td>
<td>Multiple Bernoulli (multinomial with one trial)</td>
</tr>
<tr>
<td>MULtinomial</td>
<td>$\geq 0$</td>
<td>Multinomial distribution</td>
</tr>
<tr>
<td>MENtropy</td>
<td>0-1</td>
<td>Cross or relative entropy for targets that sum to 1 (Kullback-Leibler divergence)</td>
</tr>
</tbody>
</table>

**LEVEL=value**

Specifies the measurement level, where `value` can be:

- NOMINAL|NOM
  Nominal.
- ORDINAL|ORD
  Ordinal.
- INTERVAL|INT
  Interval.
<table>
<thead>
<tr>
<th>Default:</th>
<th>NOMINAL for character variables. INTERVAL for numeric variables.</th>
</tr>
</thead>
</table>

**MESTA=number**
Specifies the scale constant for M estimation.

<table>
<thead>
<tr>
<th>Default:</th>
<th>Default value is computed from MESTCON to give consistent scale estimates for normal noise.</th>
</tr>
</thead>
</table>

**MESTCON=number**
Specifies the tuning constant for M estimation.

<table>
<thead>
<tr>
<th>Default:</th>
<th>Huber: 1.5; Biweight: 9; Wave: 2.1*π</th>
</tr>
</thead>
</table>

**SIGMA=number**
Specifies the fixed value of the scale parameter.

<table>
<thead>
<tr>
<th>Default:</th>
<th>By default, SIGMA is not used; but with OBJECT=LIKE, the scale parameter is estimated.</th>
</tr>
</thead>
</table>

**STD=method**
Specifies the standardization method, where *method* is

- NONE|NO
  Variables are not altered.

- STD
  Variables are rescaled to have a mean of 0 and a standard deviation of 1.

- RANGE|RAN
  Variables are rescaled to have a minimum of 0 and a range of 1.

- MIDRANGE|MID
  Variables are rescaled to have a midrange of 0 and a half-range of 1 (that is, a minimum of -1 and a maximum of 1).

<table>
<thead>
<tr>
<th>Default:</th>
<th>NO</th>
</tr>
</thead>
</table>
THAW Statement

Thaws frozen weights.

**Category**  Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

```
THAW weight-list;
```

**Required Argument**

`weight-list`

List of weights to thaw.

Weight-list consists of 0 or more repetitions of:

```
wname --> wname-2 where:
wname
  is the unit name, the layer ID, BIAS, or ALTITUDE
wname-2
  is the unit name or the layer ID
```

**Default:** All weights are thawed.
The NEURAL Procedure

TRAIN Statement

Trains the network.

**Category**  
Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

**TRAIN OUT=**<libref.>SAS-data-set  
**OUTEST=**<libref.>SAS-data-set  
**OUTFIT=**<libref.>SAS-data-set

- `<ACCEL|ACCELERATE>=number>`
- `<DECEL|DECELERATE>=number>`
- `<DUMMIES | NODUMMIES>`
- `<ESTITER=i>`
- `<LEARN=number>`
- `<MAX|MAXMOMENTUM=number>`
- `<MAXITER=integer>`
- `<MAXLEARN=number>`
- `<MAXTIME =number>`
- `<MINLEARN=number>`
- `<MOM|MOMENTUM=number>`
- `<TECHNIQUE=name>`;

**Options**

**ACCEL | ACCELERATE=**number

Specifies the rate of increase of learning for the RPROP optimization technique.

<table>
<thead>
<tr>
<th>Range:</th>
<th>number &gt; 1</th>
</tr>
</thead>
</table>

| Default:  | 1.2 |

**DECEL | DECELERATE=**number

Specifies the rate of decrease of learning for the RPROP optimization technique.

<table>
<thead>
<tr>
<th>Range:</th>
<th>0 &lt; number &lt; 1</th>
</tr>
</thead>
</table>

| Default:  | 0.5 |
Specifies whether to write dummy variables to the OUT= data set.

**Default:** NODUMMIES

**ESTITER=i**

- \( i = 0 \)
  - Writes only initial and final weights to the OUTEST=data set.
- \( i > 0 \)
  - Writes weights after every \( i \) iterations, as well as the final weights, to the OUTEST= data set.

**Default:** 0

**LEARN=number**

Specifies the learning rate for BPROP or the initial learning rate for QPROP and RPROP.

**Range:** \( number > 0 \)

**Default:** 0.1

**MAXITER=integer**

Maximum number of iterations.

**Default:**
- 100 for TRUREG and LEVMAR
- 200 for QUANEW and DBLDOG
- 400 for CONGRA
- 1000 for BPROP, RPROP, and QPROP

**MAXLEARN=number**

Specifies the maximum learning rate for RPROP.

**Range:** \( number > 0 \)

**Default:** 1/MACSQRTEPS

**MAXMOM | MAXMOMENTUM=number**

Specifies the maximum momentum for BPROP.

**Range:** \( number > 0 \)

**Default:** 1.75

**MAXTIME=number**
Specifies the amount of time after which training stops.

**Default:** 7 days, that is, 604800 seconds

**MINLEARN=number**

Specifies the minimum learning rate for RPROP.

**Range:** \( number > 0 \)

**Default:** MACSQRTEPS

**MOM | MOMENTUM=number**

Specifies the momentum for BPROP.

**Range:** \( 0 \leq number < 1 \)

**Default:**
- For BPROP: 0.9
- For RPROP: 0.1

**OUT=<libref.>SAS-data-set**

Specifies the output data set that contains the outputs.

**OUTEST=<libref.>SAS-data-set**

Specifies the output data set that contains the network weights.

**OUTFIT=<libref.>SAS-data-set**

Specifies the output data set that contains the fit statistics.

**TECHNIQUE=name**

Specifies the optimization technique where *name* is one of the following:

- **TRUREG**
  Requests Trust region.

- **LEVMAR**
  Requests Levenberg-Marquardt.

- **DBLDOG**
  Requests Double dogleg.

- **QUANEW**
  Requests quasi-Newton.

- **CONGRA**
  Requests Conjugate gradient.

- **BPROP**
  Requests standard backprop (backpropagation), that is, a variation on an algorithm called
the generalized delta rule. In backpropagation, the difference (delta) between the output value and the target value is the error.

RPROP
Requests the RPROP algorithm.

QPROP
Requests Quickprop.

See the following table for the defaults for weight-based optimization techniques for a given value of the OBJECT= option.

**Defaults for Weight-based Optimization Techniques**

<table>
<thead>
<tr>
<th>OBJECTIVE FUNCTION</th>
<th>OPTIMIZATION TECHNIQUE</th>
<th>NUMBER OF WEIGHTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT=DEV</td>
<td>LEVMAR</td>
<td>0 to 100 weights</td>
</tr>
<tr>
<td>OBJECT=DEV</td>
<td>QUANEW</td>
<td>101 - 501 weights</td>
</tr>
<tr>
<td>OBJECT=DEV</td>
<td>CONGRA</td>
<td>501 or more weights</td>
</tr>
<tr>
<td>(All other objective functions)</td>
<td>QUANEW</td>
<td>up to 500 weights</td>
</tr>
<tr>
<td>(All other objective functions)</td>
<td>CONGRA</td>
<td>501 or more weights</td>
</tr>
</tbody>
</table>

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The NEURAL Procedure

USE Statement

Sets all weights to values from a data set.

**Category** Action Statement - affects the network or the data sets. Options set in an action statement affect only that statement.

```
USE <libref.>SAS-data-set;
```

**Required Arguments**

```<libref.>SAS-data-set```

Specifies an input data set that contains all the weights. Unlike the INITIAL statement, the USE statement does not generate any random weights, therefore the data set must contain all of the network weights and parameters.

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The NEURAL Procedure

Details

For details about neural network architecture and training, see the online Neural Network Node: Reference documentation. For an introduction to predictive modeling, see the online Predictive Modeling document. Both of these documents can be accessed by using the Help pull-down menu to select the "Open the Enterprise Miner Nodes Help" item.

The BPROP, RPROP, and QPROP Algorithms Used in PROC NEURAL

Introduction

While the standard backprop algorithm has been a very popular algorithm for training feedforward networks, performance problems have motivated numerous attempts at finding faster algorithms.

The following discussion of the implementation of the backprop (BPROP), RPROP, and QPROP algorithms in PROC NEURAL relates the details of these algorithms with the printed output resulting from the use of the PDDETAIL option. The discussion uses the algorithmic description and notation in Schiffmann, Joost, and Werner (1994) as well as the Neural Net Frequently Asked Questions (FAQ) available as a hypertext document readable by any World-Wide Web browser, such as Mosaic, under the URL: ftp://ftp.sas.com/pub/neural/FAQ.html.

There is an important distinction between "backprop" (or "back propagation of errors") and the "backpropagation algorithm".

The "back propagation of errors" is an efficient computational technique for computing the derivative of the error function with respect to the weights and biases of the network. This derivative, more commonly known as the error gradient, is needed for any first order nonlinear optimization method. The standard backpropagation algorithm is a method for updating with weights based on the gradient. It is a variation of the simple "delta rule". See ”What is backprop?” in part 2 of the FAQ for more details and references on standard backprop, RPROP, and Quickprop.

With any of the "prop" algorithms, PROC NEURAL allows detailed printing of the iterations. The PDDETAIL option in the TRAIN statement prints, for each iteration, all quantities involved in the algorithm for each weight in the network. This option should be used with caution as it can result in voluminous output. However, by restricting the number of iterations and number of non-frozen weights, a manageable amount of information is produced. The purpose of the PDDETAIL option is to allow you to follow the nonlinear optimization of the error function for each of the network weights. For any particular propagation method, as described below, all quantities used to compute an updated weight are printed.

In standard backprop, too low a learning rate makes the network learn very slowly. Too high a learning
rate makes the weights and error function diverge, so there is no learning at all. If the error function is quadratic, as in linear models, good learning rates can be computed from the Hessian matrix. If the error function has many local and global optima, as in typical feedforward neural networks with hidden units, the optimal learning rate often changes dramatically during the training process, because the Hessian also changes dramatically. Trying to train a neural network using a constant learning rate is usually a tedious process requiring much trial and error.

With batch training, there is no need to use a constant learning rate. In fact, there is no reason to use standard backprop at all, because vastly more efficient, reliable, and convenient batch training algorithms exist such as Quickprop and RPROP.

Many other variants of backprop have been invented. Most suffer from the same theoretical flaw as standard backprop: the magnitude of the change in the weights (the step size) should NOT be a function of the magnitude of the gradient. In some regions of the weight space, the gradient is small and you need a large step size; this happens when you initialize a network with small random weights. In other regions of the weight space, the gradient is small and you need a small step size; this happens when you are close to a local minimum. Likewise, a large gradient may call for either a small step or a large step. Many algorithms try to adapt the learning rate, but any algorithm that multiplies the learning rate by the gradient to compute the change in the weights is likely to produce erratic behavior when the gradient changes abruptly. The great advantage of Quickprop and RPROP is that they do not have this excessive dependence on the magnitude of the gradient. Conventional optimization algorithms use not only the gradient but also second-order derivatives or a line search (or some combination thereof) to obtain a good step size.

**Mathematical Notation**

It is helpful to establish notation so that we can relate quantities and describe algorithms.

1. \( \omega_{ij}(n) \) is the weight associated with the connection between the \( i \)th unit in the current layer and the \( j \)th unit from the previous layer. The argument \( n \) refers to iteration.

2. \( \Delta \omega_{ij}(n) \) is the update or change for weight \( \omega_{ij}(n) \). This update results in the \( n+1 \) iteration value for \( \omega_{ij} \).

3. \( \frac{\partial E(n)}{\partial \omega_{ij}} \) is the partial derivative of the error function \( E(\omega) \) with respect to the weight \( \omega_{ij} \) at the \( n \)th iteration.

4. \( y_{km}^m(x^m; \omega) \) is the \( k \)th component of the output vector for the \( m \)th case as a function of the inputs \( x^m \) and network weights \( \omega \).

5. \( t_{km}^m(x^m) \) is the \( k \)th component of the target vector for the \( m \)th case as a function of the inputs \( x^m \).

The basic algorithm in all methods is a generic update given by
The BPROP, RPROP, and QPROP algorithms differ in how $\Delta \omega_{ij}$ is computed.

**Standard Printing for the PROP Algorithms**

When the PDETAIL option is not specified, a standard table is produced displaying the iteration number, the value of the objective function at that iteration, and $l_\infty$ norm of the gradient vector $\nabla E(\omega)$.

This table is useful for overall convergence behavior. However, unlike the table produced by the PDETAIL option, no information about individual network weights is given.

In the case of sum of squared error, which results from specifying OBJECTIVE=DEV in the NETOPTS statement and ERROR=NORMAL in the TARGET statement, the error function serves as the objective function, and is given by

$$E(\omega) = \frac{1}{2} \sum_{m=1}^{M} \sum_{k=1}^{K} (y^m_k(x; \omega) - t^m_k)^2$$

Candidate network weight values $\omega^*$ that minimize the objective function $E(\omega)$ satisfy the first order condition

$$\nabla E(\omega^*) = 0$$

Hence, a natural convergence criteria is

$$\|\nabla E(\omega)\|_\infty < \varepsilon$$

for some small value $\varepsilon$. This is, in fact, convergence criteria for all prop methods. The value of $\varepsilon$ is set by the ABSGCONV= option in the NLOPTIONS statement, with a default value of $\varepsilon = 10^{-8}$. Note that the $l_\infty$ norm $\|z\|_\infty$ for some vector $z$ is simply the maximum of the absolute value of the components of $z$.

The standard table prints the following quantities:

- Iteration $n$
- Objective, $E(\omega)$, using current network weight $\omega$
- Max Abs Gradient Element, $\|\nabla E(\omega)\|_\infty$

When the PDETAIL option is specified, this standard table is still printed. Each line of the standard tables follows the detail lines for each of the weights at each iteration.

**The Standard Backprop Algorithm**

The standard backprop algorithm is a gradient descent with momentum.

At the $n$th iteration, the update is computed as

$$\Delta \omega_{ij}(n) = -\varepsilon \frac{\partial E(n)}{\partial \omega_{ij}} + \alpha \Delta \omega_{ij}(n - 1)$$

For TECH=BPROP, the PDETAIL option in the TRAIN statement results in the following quantities being printed:

- $\Delta \omega_{ij}(n - 1)$ is labeled "Previous Change"
- $\frac{\partial E(n)}{\partial \omega_{ij}}$ is labeled "Gradient"
- $\Delta \omega_{ij}(n)$ is labeled "Current Change"
- $\omega_{ij}(n)$ is labeled "Previous Weight"
- $\omega_{ij}(n + 1)$ is labeled "Current Weight"

The learning rate $\varepsilon$ and the momentum $\alpha$ are printed at the beginning of the table. These quantities are set by the LEARN= and MOMENTUM= options respectively, with the default values of $\varepsilon = 0.1$ and $\alpha = 0.9$.

**The RPROP Algorithm**

The RPROP algorithm (Riedmiller and Braun 1993) is unusual as a descent algorithm in that it does not use the magnitude of the gradient in calculating the weight update. Instead, the signs of the current and previous gradient are used to determine a step size $\Delta_{ij}(n)$ at each iteration.
To prevent oscillations and underflows, the step size $\Delta_{ij}(n)$ is bounded by

$$\Delta_{\text{min}} \leq \Delta_{ij}(n) \leq \Delta_{\text{max}}$$

The value of $\Delta_{\text{max}}$ is set by the MAXLEARN= option with a default value of $10^7$. The value of $\Delta_{\text{min}}$ is set by the MINLEARN= option with a default value of $10^{-7}$. Note that these values are substantially different from the recommendations given in Schiffmann, Joost and Werner, (1994). These new values improved stability and convergence over a wide range of problems.

For each connection weight, an initial stepsize $\Delta_{ij}(0)$ is given a small value. According to Schiffmann, Joost and Werner, (1994), results are not typically dependent on the exact value given $\Delta_{ij}(0)$. PROC NEURAL uses a default initial step size of 0.1 for all weights and is set by the LEARN= option in the TRAIN statement.

At the $n$th iteration, adjust the step size by

$$\Delta_{ij}(n) = \begin{cases} 
\Delta_{ij}(n-1) u, & \text{if } \frac{\partial E(n)}{\partial \omega_{ij}} \frac{\partial E(n-1)}{\partial \omega_{ij}} > 0 \\
\Delta_{ij}(n-1) d, & \text{if } \frac{\partial E(n)}{\partial \omega_{ij}} \frac{\partial E(n-1)}{\partial \omega_{ij}} < 0 \\
\Delta_{\text{max}}, & \text{if } \Delta_{ij}(n) > \Delta_{\text{max}} \\
\Delta_{\text{min}}, & \text{if } \Delta_{ij}(n) < \Delta_{\text{min}} 
\end{cases}$$

The factors $u$ and $d$ are the acceleration and deceleration respectively. The values of these factors are set by the ACCELERATE= and DECELERATE= options in the TRAIN statement. The default value for ACCELERATE= 1.2; for DECELERATE= the default value is 0.5.

For TECH=RPROP, the PDETAIL option in the TRAIN statement results in the following quantities being printed:

$\Delta_{ij}(n-1)$ is labeled "Previous Step Size"

$$\frac{\partial E(n-1)}{\partial \omega_{ij}}$$ is labeled "Previous Gradient"

$$\frac{\partial E(n)}{\partial \omega_{ij}}$$ is labeled "Current Gradient"
The Quickprop Algorithm

The Quickprop algorithm (Fahlman 1989) assumes that the error function behaves locally like a parabola, and uses the method of false position to find the minimum of the approximating quadratic. Variations are required to ensure the change is not uphill and to handle cases where the gradient does not change between iterations (causing the false position method to fail).

The quickprop algorithm uses a modified gradient related to the regular gradient by

\[ \frac{\partial E^* (n)}{\partial \omega_{ij}} = \frac{\partial E (n)}{\partial \omega_{ij}} + \text{decay} \cdot \omega_{ij} (n) \]

At the \( n \)th iteration, the weight update is given by

\[ \Delta \omega_{ij} (n) = -\varepsilon (n) \frac{\partial E^* (n)}{\partial \omega_{ij}} + \alpha_{ij} (n) \Delta \omega_{ij} (n - 1) \]

For initialization at \( n=1 \), we set \( \Delta \omega_{ij} (0) = 0 \), so the update step becomes a gradient descent:

\[ \Delta \omega_{ij} (1) = -\varepsilon (n) \frac{\partial E^* (1)}{\partial \omega_{ij}} \]

At the second and subsequent iterations, \( \varepsilon (n) \) and \( \alpha_{ij} (n) \) are computed as follows:
Calculation of $\epsilon (n)$ first involves evaluation $\hat{\alpha}_{ij} (n)$, the numerical estimate of the second derivative:

$$\epsilon (n) = \begin{cases} 
\epsilon_0, & \text{if } \frac{\partial E^*(n)}{\partial \omega_{ij}} \Delta \omega_{ij} (n - 1) > 0 \\
\epsilon_0, & \text{if } \Delta \omega_{ij} (n - 1) = 0 \\
0, & \text{otherwise}
\end{cases}$$

This second derivative can become large in absolute value or can signal a move "up" the gradient away from a minimum. The following modifications are applied to account for these situations.

$$\hat{\alpha}_{ij} (n) = \frac{\frac{\partial E^*(n)}{\partial \omega_{ij}}}{\frac{\partial E^*(n-1)}{\partial \omega_{ij}} - \frac{\partial E^*(n)}{\partial \omega_{ij}}}$$

The value of $\epsilon_0$ is set by the LEARN= option in the TRAIN statement, with a default value of $\epsilon_0 = 0.1$. The bound $\alpha_{\text{max}}$ is set by the MAXMOMENTUM= option in the TRAIN statement, with a default value of $\alpha_{\text{max}} = 1.75$.

For TECH=QPROP, the PDETAIL option in the TRAIN statement results in the following quantities being printed:

- $\omega_{ij} (n - 1)$ is labeled "Previous Weight"
- $\frac{\partial E(n)}{\partial \omega_{ij}}$ is labeled "Gradient"
- $\frac{\partial E^*(n)}{\partial \omega_{ij}}$ is labeled "Modified Gradient"
\[ \frac{\partial E^*(n-1)}{\partial \omega_{ij}} \] is labeled "Previous Modified Gradient"

\[ \Delta w_{ij} (n - 1) \] is labeled "Previous Change"

\[ \alpha (n) \] is labeled "Alpha"

\[ \epsilon (n) \] is labeled "Epsilon"

\[ \Delta \omega_{ij} (n) \] is labeled "Current Change"

\[ \omega_{ij} (n + 1) \] is labeled "Current Weight"

References


Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
Examples

The following examples were executed using the HP-UX version 10.20 operating system and the SAS software release 6.12TS045.

**Example 1: Developing a Simple Multilayer Perceptron (Rings Data)**

**Example 2: Developing a Neural Network for a Continuous Target**

**Example 3: Neural Network Hill-and-Plateau Example (Surf Data)**

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
Example 1: Developing a Simple Multilayer Perceptron (Rings Data)

This example demonstrates how to develop a multilayer perceptron network with three hidden units. The example training data set is named SAMPSON.DMDRING (rings data). It contains a categorical target (C= 0, 1, or 2) plus two continuous inputs (X and Y). There are 180 cases in the data set. The SAMPSON.DMSRING data set is scored using the scoring formula from the trained models.

Both data sets and the SAMPSON.DMDRING catalog are stored in the sample library.

Program

```sas
proc gplot data=sampio.dmdring;
   plot y*x=c /haxis=axis1 vaxis=axis2;
   symbol c=black i=none v=dot;
   symbol2 c=red i=none v=square;
   symbol3 c=green i=none v=triangle;
   axis1 c=black width=2.5  order=(0 to 30 by 5);
   axis2 c=black width=2.5 minor=none order=(0 to 20 by 2);
   title 'Plot of the Rings Training Data';
run;

proc neural data=sampio.dmdring
dmdbcat=sampio.dmdring
random=789;

   input x y / level=interval id=i;
   target c / id=o level=nominal;
   hidden 3 / id=h;
   prelim 5;
   train;
   score out=out outfit=fit;
   score data=sampio.dmsring out=gridout;
   title 'MLP with 3 Hidden Units';
run;
```
proc print data=fit noobs label;
  var _aic_ _ase_ _max_ _rfpe_ _misc_ _wrong_
  where _name_ = 'OVERALL';
  title2 'Fits Statistics for the Training Data Set';
run;

proc freq data=out;
  tables f_c*i_c;
  title2 'Misclassification Table';
run;

proc gplot data=out;
  plot y*x=i_c /haxis=axis1 vaxis=axis2;
  symbol c=black i=none v=dot;
  symbol2 c=black i=none v=square;
  symbol3 c=black i=none v=triangle;
  axis1 c=black width=2.5  order=(0 to 30 by 5);
  axis2 c=black width=2.5 minor=none order=(0 to 20 by 2);
  title2 'Classification Results';
run;

proc gcontour data=gridout;
  plot y*x=p_c1 / pattern ctext=black coutline=gray;
  plot y*x=p_c2 / pattern ctext=black coutline=gray;;
  plot y*x=p_c3 / pattern ctext=black coutline=gray;;
  pattern v=msolid;
  legend frame;
  title2 'Posterior Probabilities';
run;

proc gcontour data=gridout;
  plot y*x=h1 / pattern ctext=black coutline=gray;
  plot y*x=h2 / pattern ctext=black coutline=gray;;
  plot y*x=h3 / pattern ctext=black coutline=gray;;
  pattern v=msolid;
  legend frame;
  title2 'Hidden Unit Values';
run;

Output

PROC GPLOT of the Rings Training Data
Notice that the target classes are not linearly separable.

**PROC NEURAL: Preliminary Training Output**

This section lists the objective function for each preliminary iteration. The parameter estimates (weights) from the iteration number that has the smallest objective function are passed as input for final training. Because the target is nominal, the error function is set to multiple Bernoulli. Therefore, the objective function that is being minimized is the negative log-likelihood. Iteration number 4 has the smallest objective function.

The 17 initial parameter estimates are also listed in this section of the output.

**MLP with 3 Hidden Units**

<table>
<thead>
<tr>
<th>Iteration number</th>
<th>Pseudo-random number seed</th>
<th>Objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>789</td>
<td>0.35192</td>
</tr>
<tr>
<td>1</td>
<td>761237432</td>
<td>0.20067</td>
</tr>
<tr>
<td>2</td>
<td>1092694980</td>
<td>0.18602</td>
</tr>
<tr>
<td>3</td>
<td>577625332</td>
<td>0.29195</td>
</tr>
<tr>
<td>4</td>
<td>261548896</td>
<td>0.15312</td>
</tr>
</tbody>
</table>

**Parameter Estimates**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 X_H1</td>
<td>-0.399940</td>
<td>-0.00596</td>
<td>X -&gt; H1</td>
</tr>
<tr>
<td>2 Y_H1</td>
<td>-2.215319</td>
<td>0.01670</td>
<td>Y -&gt; H1</td>
</tr>
<tr>
<td>3 X_H2</td>
<td>2.570511</td>
<td>-0.03857</td>
<td>X -&gt; H2</td>
</tr>
<tr>
<td>4 Y_H2</td>
<td>0.672317</td>
<td>-0.02087</td>
<td>Y -&gt; H2</td>
</tr>
<tr>
<td>5 X_H3</td>
<td>2.589547</td>
<td>0.01907</td>
<td>X -&gt; H3</td>
</tr>
<tr>
<td>6 Y_H3</td>
<td>-1.945493</td>
<td>0.00149</td>
<td>Y -&gt; H3</td>
</tr>
<tr>
<td>7 BIAS_H1</td>
<td>2.153111</td>
<td>0.01586</td>
<td>BIAS -&gt; H1</td>
</tr>
<tr>
<td>8 BIAS_H2</td>
<td>2.276595</td>
<td>0.04635</td>
<td>BIAS -&gt; H2</td>
</tr>
<tr>
<td>9 BIAS_H3</td>
<td>-2.243021</td>
<td>0.00979</td>
<td>BIAS -&gt; H3</td>
</tr>
<tr>
<td>10 H1_C1</td>
<td>5.688000</td>
<td>0.00208</td>
<td>H1 -&gt; C1</td>
</tr>
<tr>
<td>11 H2_C1</td>
<td>5.828867</td>
<td>0.000806</td>
<td>H2 -&gt; C1</td>
</tr>
</tbody>
</table>
Levenberg-Marquardt Termination Criteria

This section lists the termination criteria for the Levenberg-Marquardt optimization.

MLP with 3 Hidden Units

Levenberg-Marquardt Optimization

*** Termination Criteria ***
Minimum Iterations .................. -1
Maximum Iterations .................. 100
Maximum Function Calls .............. 2147483647
Maximum CPU Time ................. 604800
ABSGCONV Gradient Criterion .......... 0.0000100
GCONV Gradient Criterion ......... 1E-8
GCONV2 Gradient Criterion .............. 0
ABSFCONV Function Criterion .......... 0
FCONV Function Criterion .......... 0.0001000
FCONV2 Function Criterion .......... 0
FSIZE Parameter ...................... 0
ABSXCONV Parameter Change Criterion .......... 0
XCONV Parameter Change Criterion .......... 0
XSIZE Parameter .............. 0
ABSCONV Function Criterion .......... -1.341E154

*** Other Control Parameters ***
Trust Region Initial Radius Factor .......... 1.00000
Singularity Tolerance (SINGULAR) .......... 1E-8

Levenberg-Marquardt Iteration Log

At the start of optimization, the procedure lists the number of active constraints, the current value of the objective function, the maximum gradient element, and the radius. The iteration history includes the following:

- the iteration number (iter)
- the number of iteration restarts (rest)
- the number of active constraints (act)
- the value of the objective function (optcrit)
- the difference between the adjacent objective functions (difcrit)
- the maximum of the absolute (projected) gradient components (maxgrad)
- the value of lambda
- the value of rho

The optimization results section lists information specific for the optimization technique.

MLP with 3 Hidden Units

Levenberg-Marquardt Optimization
Scaling Update of More (1978)
Number of Parameter Estimates 17

Optimization Start: Active Constraints= 0 Criterion= 0.153
### Optimization Results

Maximum Gradient Element = 0.046 Radius = 1.000

<table>
<thead>
<tr>
<th>Iter</th>
<th>rest</th>
<th>nfun</th>
<th>act</th>
<th>optcrit</th>
<th>difcrit</th>
<th>maxgrad</th>
<th>lambda</th>
<th>rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0.1007</td>
<td>0.0524</td>
<td>0.0708</td>
<td>0.0138</td>
<td>0.582</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0.0883</td>
<td>0.0124</td>
<td>0.0589</td>
<td>0.417</td>
<td>0.330</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>0.0546</td>
<td>0.0336</td>
<td>0.0493</td>
<td>0.0093</td>
<td>0.674</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>19</td>
<td>0</td>
<td>0.0255</td>
<td>0.0291</td>
<td>0.0726</td>
<td>0.00154</td>
<td>0.667</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0.0233</td>
<td>0.00227</td>
<td>0.136</td>
<td>0</td>
<td>0.0960</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>0.003321</td>
<td>0.0200</td>
<td>0.0154</td>
<td>0</td>
<td>0.897</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>0.000742</td>
<td>0.00258</td>
<td>0.00889</td>
<td>5.65E-7</td>
<td>0.816</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>0.000283</td>
<td>0.000459</td>
<td>0.00683</td>
<td>0</td>
<td>0.636</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>26</td>
<td>0</td>
<td>0.0000632</td>
<td>0.00022</td>
<td>0.00147</td>
<td>0</td>
<td>0.789</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>6.4334E-6</td>
<td>0.000057</td>
<td>0.00018</td>
<td>0</td>
<td>0.908</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td>1.1769E-6</td>
<td>5.257E-6</td>
<td>0.00006</td>
<td>0</td>
<td>0.823</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>29</td>
<td>0</td>
<td>2.6787E-7</td>
<td>9.09E-7</td>
<td>0.00002</td>
<td>0</td>
<td>0.776</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>1.2592E-7</td>
<td>1.42E-7</td>
<td>0.00002</td>
<td>0</td>
<td>0.532</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>1.6879E-8</td>
<td>1.09E-7</td>
<td>2.89E-6</td>
<td>308E-15</td>
<td>0.869</td>
</tr>
</tbody>
</table>

Optimization Results: Iterations = 14 Function Calls = 33 Jacobian Calls = 15
Active Constraints = 0 Criterion = 1.6879056E-8
Maximum Gradient Element = 2.8916E-6 Lambda = 3.08E-13 Rho = 0.8685 Radius = 26.94

NOTE: ABSGCONV convergence criterion satisfied.

### Parameter Estimates (MLP Weights)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>X_H1</td>
<td>-0.377915</td>
<td>1.2972E-6</td>
<td>X -&gt; H1</td>
</tr>
<tr>
<td>Y_H1</td>
<td>-0.665105</td>
<td>2.8916E-6</td>
<td>Y -&gt; H1</td>
</tr>
<tr>
<td>X_H2</td>
<td>1.720212</td>
<td>8.9422E-8</td>
<td>X -&gt; H2</td>
</tr>
<tr>
<td>Y_H2</td>
<td>-0.043669</td>
<td>1.63254E-7</td>
<td>Y -&gt; H2</td>
</tr>
<tr>
<td>X_H3</td>
<td>0.487788</td>
<td>-3.1918E-7</td>
<td>X -&gt; H3</td>
</tr>
<tr>
<td>Y_H3</td>
<td>-0.801489</td>
<td>-5.7188E-7</td>
<td>Y -&gt; H3</td>
</tr>
<tr>
<td>BIAS_H1</td>
<td>0.875418</td>
<td>1.87012E-6</td>
<td>BIAS -&gt; H1</td>
</tr>
<tr>
<td>BIAS_H2</td>
<td>1.340638</td>
<td>-1.1246E-7</td>
<td>BIAS -&gt; H2</td>
</tr>
<tr>
<td>BIAS_H3</td>
<td>-0.649604</td>
<td>-7.5073E-8</td>
<td>BIAS -&gt; H3</td>
</tr>
<tr>
<td>H1_C1</td>
<td>388.611382</td>
<td>-1.7823E-9</td>
<td>H1 -&gt; C1</td>
</tr>
<tr>
<td>H2_C1</td>
<td>289.584211</td>
<td>-1.6285E-9</td>
<td>H2 -&gt; C1</td>
</tr>
<tr>
<td>H3_C1</td>
<td>-309.435371</td>
<td>1.04029E-9</td>
<td>H3 -&gt; C1</td>
</tr>
<tr>
<td>H1_C2</td>
<td>177.046552</td>
<td>-1.9849E-9</td>
<td>H1 -&gt; C2</td>
</tr>
<tr>
<td>H2_C2</td>
<td>118.043485</td>
<td>1.37679E-8</td>
<td>H2 -&gt; C2</td>
</tr>
<tr>
<td>H3_C2</td>
<td>-149.150661</td>
<td>-1.1747E-8</td>
<td>H3 -&gt; C2</td>
</tr>
<tr>
<td>BIAS_C1</td>
<td>-580.555136</td>
<td>-2.0973E-9</td>
<td>BIAS -&gt; C1</td>
</tr>
<tr>
<td>BIAS_C2</td>
<td>-213.290409</td>
<td>1.39802E-8</td>
<td>BIAS -&gt; C2</td>
</tr>
</tbody>
</table>

Value of Objective Function = 1.6879056E-8

### List Report of Selected Variables in the Score OUTFIT= Data Set

The example PROC PRINT report of the OUTFIT = data set lists selected fit statistics for the training data. By default, the OUTFIT data set contains two observations. These observations are distinguished by the value of the _NAME_ variable:

- _NAME_ = 'Target Variable Name'
- _NAME_ = 'OVERALL'

Because a WHERE statement was used to select only values of _NAME_ = 'OVERALL', this report contains a single observation.
Notice that the MLP network with 3 hidden units correctly classifies all cases in the training data set.

**MLP with 3 Hidden Units**

**Fits Statistics for the Training Data Set**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>34.0000</td>
<td>3.7039E-14</td>
<td>0.0000030981</td>
<td>0.00000020177</td>
</tr>
</tbody>
</table>

**Train: Misclassification Rate.**

<table>
<thead>
<tr>
<th>Misclassification Rate.</th>
<th>Number of Wrong Classifications.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**PROC FREQ Misclassification Table for the Training Data**

**MLP with 3 Hidden Units**

**Misclassification Table**

<table>
<thead>
<tr>
<th>F_C(From: C)</th>
<th>I_C(Into: C)</th>
<th>Frequency</th>
<th>Percent</th>
<th>Row Pct</th>
<th>Col Pct</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------+--------+--------+--------+--------+--------+-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>4.44</td>
<td>4.44</td>
</tr>
<tr>
<td></td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>-------------+--------+--------+--------+--------+--------+-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>59</td>
<td>0</td>
<td>59</td>
<td>32.78</td>
<td>32.78</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>-------------+--------+--------+--------+--------+--------+-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>113</td>
<td>113</td>
<td>62.78</td>
<td>62.78</td>
</tr>
<tr>
<td></td>
<td>0.00</td>
<td>0.00</td>
<td>100.00</td>
<td>0.00</td>
<td>100.00</td>
<td></td>
</tr>
<tr>
<td>-------------+--------+--------+--------+--------+--------+-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>8</td>
<td>59</td>
<td>113</td>
<td>180</td>
<td>4.44</td>
<td>32.78</td>
</tr>
<tr>
<td></td>
<td>62.78</td>
<td>100.00</td>
<td>62.78</td>
<td>100.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PROC GPlot Plot of the Classification Results**
PROC GGONTOUR Plots of the Posterior Probabilities

The legend at the bottom of the chart identifies the target level.

MLP with 3 Hidden Units
Classification Results
PROC GPLOT creates a scatter plot of the Rings training data.

proc gplot data=sampsio.dmdring;
  plot y*x=c /haxis=axis1 vaxis=axis2;
  symbol c=black i=none v=dot;
  symbol2 c=red i=none v=square;
  symbol3 c=green i=none v=triangle;
  axis1 c=black width=2.5 order=(0 to 30 by 5);
  axis2 c=black width=2.5 minor=none order=(0 to 20 by 2);
  title 'Plot of the Rings Training Data';
run;
The PROC NEURAL statement invokes the procedure. The DATA= option identifies the training data set that is used to fit the model. The DMDBCAT= option identifies the DMDB catalog. The RANDOM= option specifies the random number seed.

```plaintext
proc neural data=sampsio.dmdring
dmdbcat=sampsio.dmdring
random=789;
```
The INPUT statement specifies an interval input layer. The LEVEL= option specifies the measurement level. The ID= option specifies an identifier for the interval input layer.

```
input x y / level=interval id=i;
```
The TARGET statement defines an output layer. The output layer computes predicted values and compares those predicted values with the value of the target variable. The ID= option specifies an identifier for the output layer. The LEVEL= option specifies the target measurement level. By default, for nominal targets the combination function is set to linear, the activation function is set to mlogistic, and the error function is set to mbernoulli.

target c / id=o level=nominal;
The HIDDEN statement defines the number of hidden units that are used to perform the internal computations. By default, the input units are connected to each hidden unit and each hidden unit is connected to the output unit. The ID= option specifies an identifier for the hidden unit.

hidden 3 / id=h;
The PRELIM statement causes the procedure to search for the best starting weights for subsequent training. The integer value of 5 specifies to use 5 preliminary runs. The weights from the seed with the smallest objective function among all runs is chosen. Preliminary training may help prevent the network from converging in a local minima.

prelim 5;
The TRAIN statement trains the network in order to find the best weights (parameter estimates) that accurately reflect the training data.
The first SCORE statement scores the training data. The OUT= option identifies the output data set that contains outputs. The OUTFIT= option identifies the output data set that contains fit statistics.

```plaintext
score out=out outfit=fit;
```
The second SCORE statement specifies the score data set that you want to score in conjunction with training.

    score data=sampsio.dmsring out=gridout;
    title 'MLP with 3 Hidden Units';
run;
PROC PRINT lists selected training fit statistics.

proc print data=fit noobs label;
  var _aic_ _ase_ _max_ _rfpe_ _misc_ _wrong_;
  where _name_ = 'OVERALL';
  title2 'Fits Statistics for the Training Data Set';
run;
PROC FREQ creates a misclassification table for the training data. The F_C variable is the actual target value for each case and the I_C variable is the target value into which the case is classified.

```sas
proc freq data=out;
  tables f_c*i_c;
  title2 'Misclassification Table';
run;
```
PROC GPLOT plots the classification results for the training data.

proc gplot data=out;
  plot y*x=i_c /haxis=axis1 vaxis=axis2;
  symbol c=black i=none v=dot;
  symbol2 c=black i=none v=square;
  symbol3 c=black i=none v=triangle;
  axis1 c=black width=2.5 order=(0 to 30 by 5);
  axis2 c=black width=2.5 minor=none order=(0 to 20 by 2);
  title2 'Classification Results';
run;
PROC GCONTOUR produces a contour plot of the posterior probabilities for the scored data set.

proc gcontour data=gridout;
  plot y*x=p_c1 / pattern ctext=black coutline=gray;
  plot y*x=p_c2 / pattern ctext=black coutline=gray;;
  plot y*x=p_c3 / pattern ctext=black coutline=gray;;
  pattern v=msolid;
  legend frame;
  title2 'Posterior Probabilities';
run;
Example 2: Developing a Neural Network for a Continuous Target

Features
- Specifying Input, Hidden, and Output Layers
- Defining Direct Connections
- Scoring Data with the Score Statement
- Outputting Fit Statistics

This example demonstrates how to develop a neural network model for a continuous target. A simple multilayer perceptron architecture is employed with one hidden unit and direct connections. The example DMDB training data set SAMSIO.DMBASE (baseball data set) contains performance measures and salary levels for regular hitters and leading substitute hitters in major league baseball for the year 1986 (Collier 1987). There is one observation per hitter. The continuous target variable is log of salary (logsalary).

Prior to fitting the neural network model, the number of original model inputs was reduced based on a preliminary stepwise PROC DMREG run. The input set from the model with the smallest SBC (Schwarz's Bayesian Criterion) is used as input to the network. The output from the PROC DMREG analysis can be found in the PROC DMREG chapter, "Example 2. Performing a Stepwise OLS Regression".

The SAMSIO.DMTBASE data set is a test data set that is scored using the scoring formula from the trained model. The SAMSIO.DMBASE and SAMSIO.DMTBASE data sets and the SAMPIO.DMDBASE catalog are stored in the sample library.

Program

```sas
proc dmreg data=samsio.dmdbase dmdbcat=samsio.dmdbase
testdata=samsio.dmtbase outest=regest;
class league division position;
model logsalar = no_atbat no_hits no_home no_runs no_rbi no_bb
   yr_major cr_atbat cr_hits cr_home cr_runs
   cr_rbi cr_bb league division position no_outs
   no_assts no_error /
   error=normal selection=stepwise
   slentry=0.25 slstay=0.25 choose=sbc;
title1 'Preliminary DMDREG Stepwise Selection';
run;

proc neural data=samsio.dmdbase
dmdbcat=samsio.dmdbase
   random=12345;
input cr_hits no_hits no_outs no_error no_bb
   / level=interval id=int;
input division / level=nominal id=nom;

hidden 1 / id=hu;
```
target logsalar /
   level=interval
   id=tar;

connect int tar;
connect nom tar;
connect int hu;
connect nom hu;
connect hu tar;

prelim 10;

train;

score data=sampsio.dmtbase outfit=netfit
   out=netout(rename=(p_logsal=predict r_logsal=residual));
   title 'NN:1 Hidden Unit, Direct Connections, and Reduced Input Set';
run;

proc print data=netfit noobs label;
   where _name_ = 'LOGSALAR';
   var _iter_ _pname_ _tmse_ _trmse_ _tmax_;
   title 'Partial Listing of the Score OUTFIT= Data Set';
run;

proc gplot data=netout;
   plot logsalar*predict / haxis=axis1 vaxis=axis2;
   symbol c=black i=none v=dot h=3 pct;
   axis1 c=black width=2.5;
   axis2 c=black width=2.5;
   title 'Diagnostic Plots for the Scored Test Baseball Data';
run;
quit;

Output
Preliminary Training Output

This section lists the objective function for each preliminary iteration run. The weights from the iteration number that has the smallest objective function are passed as input for final training. Because the target is continuous, the error function is set to normal. Therefore, the objective function that is being minimized is the least squares error. Iteration number 0 has the smallest objective function. The parameter estimates for iteration number 0 are also listed in this section of the output.

### NN: 1 Hidden Unit, Direct Connections, and Reduced Input Set

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Pseudo-random number seed</th>
<th>Objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>12345</td>
<td>0.14536</td>
</tr>
<tr>
<td>1</td>
<td>845250737</td>
<td>0.14536</td>
</tr>
<tr>
<td>2</td>
<td>111329849</td>
<td>0.28210</td>
</tr>
<tr>
<td>3</td>
<td>1696138964</td>
<td>0.18477</td>
</tr>
<tr>
<td>4</td>
<td>1038363354</td>
<td>0.14536</td>
</tr>
<tr>
<td>5</td>
<td>1071492826</td>
<td>0.18942</td>
</tr>
<tr>
<td>6</td>
<td>117568856</td>
<td>0.14536</td>
</tr>
<tr>
<td>7</td>
<td>1792608669</td>
<td>0.30494</td>
</tr>
<tr>
<td>8</td>
<td>1691324682</td>
<td>0.18331</td>
</tr>
<tr>
<td>9</td>
<td>2114796956</td>
<td>0.28605</td>
</tr>
</tbody>
</table>

### Optimization Start

#### Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CR_HITS_</td>
<td>5.240279</td>
<td>-9.8369E-7 CR_HITS -&gt; HU1</td>
</tr>
<tr>
<td>2 NO_HITS_</td>
<td>-0.575284</td>
<td>0.0000118 NO_HITS -&gt; HU1</td>
</tr>
<tr>
<td>3 NO_OUTS_</td>
<td>-0.298484</td>
<td>1.91236E-6 NO_OUTS -&gt; HU1</td>
</tr>
<tr>
<td>4 <em>DUP1</em></td>
<td>0.019049</td>
<td>0.0000127 NO_ERROR -&gt; HU1</td>
</tr>
<tr>
<td>5 NO_BB_HU</td>
<td>-0.097201</td>
<td>8.015E-6 NO_BB -&gt; HU1</td>
</tr>
<tr>
<td>6 <em>DUP2</em></td>
<td>-0.159479</td>
<td>3.44079E-6 DIVISIONEAST -&gt; HU1</td>
</tr>
<tr>
<td>7 BIAS_HU1</td>
<td>4.099012</td>
<td>1.7976E-6 BIAS -&gt; HU1</td>
</tr>
<tr>
<td>8 <em>DUP3</em></td>
<td>0.114451</td>
<td>-8.5694E-8 CR_HITS -&gt; LOGSALAR</td>
</tr>
<tr>
<td>9 <em>DUP4</em></td>
<td>0.186707</td>
<td>1.18117E-8 NO_OUTS -&gt; LOGSALAR</td>
</tr>
<tr>
<td>10 <em>DUP5</em></td>
<td>0.156401</td>
<td>2.6625E-9 NO_ERROR -&gt; LOGSALAR</td>
</tr>
<tr>
<td>11 <em>DUP6</em></td>
<td>-0.042491</td>
<td>8.96911E-8 NO_BB -&gt; LOGSALAR</td>
</tr>
<tr>
<td>12 NO_BB_LO</td>
<td>0.151510</td>
<td>4.8488E-9 NO_BB -&gt; LOGSALAR</td>
</tr>
<tr>
<td>13 <em>DUP7</em></td>
<td>0.055166</td>
<td>3.48459E-8 DIVISIONEAST -&gt; LOGSALAR</td>
</tr>
<tr>
<td>14 HU1_LOGS</td>
<td>0.839297</td>
<td>-9.8363E-8 HU1 -&gt; LOGSALAR</td>
</tr>
<tr>
<td>15 BIAS_LOG</td>
<td>5.490837</td>
<td>7.82267E-8 BIAS -&gt; LOGSALAR</td>
</tr>
</tbody>
</table>

Value of Objective Function = 0.1453574646

Levenberg-Marquardt Termination Criteria
Levenberg-Marquardt Optimization

*** Termination Criteria ***
Minimum Iterations .................. -1
Maximum Iterations .................. 100
Maximum Function Calls................ 2147483647
Maximum CPU Time .................. 604800
ABSGCONV Gradient Criterion .......... 0.0000100
GCONV Gradient Criterion ............ 1E-8
GCONV2 Gradient Criterion ........... 0
ABSFCONV Function Criterion .......... 0
FCONV Function Criterion ............ 0.0001000
FCONV2 Function Criterion .......... 0
FSIZE Parameter ..................... 0
ABSXCONV Parameter Change Criterion .. 0
XCONV Parameter Change Criterion ..... 0
XSIZE Parameter ...................... 0
ABSCONV Function Criterion .......... -1.341E154

*** Other Control Parameters ***
Trust Region Initial Radius Factor .... 1.00000
Singularity Tolerance (SINGULAR) ...... 1E-8

Levenberg-Marquardt Iteration Log

NN:1 Hidden Unit, Direct Connections, and Reduced Input Set

Levenberg-Marquardt Optimization
Scaling Update of More (1978)
Number of Parameter Estimates 15

Optimization Start: Active Constraints= 0  Criterion= 0.145
Maximum Gradient Element= 0.000 Radius= 1.000

Iter rest nfun act  optcrit  difcrit maxgrad  lambda  rho
1   0   2   0  0.1454  4.131E-9  8.81E-6   0   1.694

Optimization Results: Iterations= 1 Function Calls= 3 Jacobian Calls= 2
Active Constraints= 0  Criterion= 0.14535746
Maximum Gradient Element= 8.81033E-6 Lambda= 0 Rho= 1.694 Radius= 0.002375

NOTE:  ABSGCONV convergence criterion satisfied.

Parameter Estimates

Optimization Results
Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 CR_HITS_</td>
<td>5.242352</td>
<td>-6.7156E-7</td>
<td>CR_HITS -&gt; HU1</td>
</tr>
<tr>
<td>2 NO_HITS_</td>
<td>-0.575574</td>
<td>8.16507E-6</td>
<td>NO_HITS -&gt; HU1</td>
</tr>
<tr>
<td>3 NO_OUTS_</td>
<td>-0.298618</td>
<td>1.3324E-6</td>
<td>NO_OUTS -&gt; HU1</td>
</tr>
<tr>
<td>4 <em>DUP1</em></td>
<td>0.018999</td>
<td>8.81033E-6</td>
<td>NO_ERROR -&gt; HU1</td>
</tr>
</tbody>
</table>
List Report of Selected Variables in the Score OUTFIT= Data Set

The example PROC PRINT report of the OUTFIT= data set contains selected summary statistics from the scored training data set.

Partial Listing of the Score OUTFIT= Data Set

Test: Mean     Test: Root Mean     Test: Maximum
_ITER_    _PNAME_        Squared Error.   Squared Error.     Absolute Error.
 0      P_LOGSAL          0.15595        0.39491            1.60237

Diagnostic Plots for the Scored Test Baseball Data

Plot of the log of salary versus the predicted log of salary.
Plot of the residual values versus the predicted log of salary.

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The preliminary PROC DMREG run selects the reduced input set.

```plaintext
proc dmreg data=sampsio.dmdbase dmdbcat=sampsio.dmdbase
testdata=sampsio.dmtbase outest=regest;
class league division position;
model logsalar = no_atbat no_hits no_home no_runs no_rbi no_bb
       yr_major cr_atbat cr_hits cr_home cr_runs
       cr_rbi cr_bb league division position no_outs
       no_assts no_error /
       error=normal selection=stepwise
       slentry=0.25 slstay=0.25 choose=sbc;

title1 'Preliminary DMDREG Stepwise Selection';
run;
```
The PROC NEURAL statement invokes the procedure. The DATA= option identifies the training data set that is used to fit the model. The DMDBCAT= option identifies the training catalog. The RANDOM= option specifies the seed that is used to set the random initial weights.

```plaintext
proc neural data=sampsio.dmdbase
dmdbcat=sampsio.dmdbase
random=12345;
```
The **INPUT** statements specifies the input layers. There are separate input layers for the interval and nominal inputs. The **LEVEL=** option specifies the measurement level. The **ID=** option specifies an identifier for each input layer.

```
input cr_hits no_hits no_outs no_error no_bb
    / level=interval id=int;
input division / level=nominal id=nom;
```
The HIDDEN statement sets the number of hidden units. The ID= option specifies an identifier for the hidden layer. By default, the combination function is set to linear and the activation function is set to hyperbolic tangent.

    hidden 1 / id=hu;
The TARGET statement defines an output layer. The output layer computes predicted values and compares those predicted values with the value of the target variable (LOGSALAR). The LEVEL= option specifies the target measurement level. The ID= option specifies an identifier for the output layer. By default, the combination function is set to linear, the activation function is set to the identity, and the error function is set to normal for continuous targets.

```
target logsalar /
   level=interval
   id=tar;
```
The CONNECT statements specify how to connect the layers. The id-list specifies the identifier of two or more layers to connect. In this example, each input unit is connected to the hidden unit and to the output unit, and the hidden unit is connected to the output unit.

```
connect int tar;
connect nom tar;
connect int hu;
connect nom hu;
connect hu tar;
```
The PRELIM statement does preliminary training using 10 different sets of initial weights. The weights from the preliminary run with the smallest objective function among all runs are retained for subsequent training when using the TRAIN statement. Preliminary training may help prevent the network from converging to a bad local minima.

prelim 10;
The TRAIN statement trains the network in order to find the best weights (parameter estimates) to fit the training data. By default, the Levenberg-Marquardt optimization technique is used for small least squares networks, such as the one in this example.

```
  train;
```
The SCORE statement creates output data sets. The DATA= option specifies the data set you want to score. The OUTFIT= option creates a data set containing fit statistics.

score data=sampsio.dmtbase outfit=netfit
The OUT= option identifies the output data for predicted values and residuals. The RENAME= option renames the variables in the OUT= data set containing predicted values and residuals.

```
out=netout(rename=(p_logsal=predict r_logsal=residual));
title 'NN:1 Hidden Unit, Direct Connections, and Reduced Input Set';
run;
```
PROC PRINT lists selected variables from the OUTFIT= data set.

proc print data=netfit noobs label;
   where _name_ = 'LOGSALAR';
   var _iter_ _pname_ _tmse_ _trmse_ _tmax_;
   title 'Partial Listing of the Score OUTFIT= Data Set';
run;
PROC GLOT plots diagnostic plots for the scored data set.

proc gplot data=netout;
   plot logsalar*predict / haxis=axis1 vaxis=axis2;
      symbol c=black i=none v=dot h=3 pct;
   axis1 c=black width=2.5;
   axis2 c=black width=2.5;
   title 'Diagnostic Plots for the Scored Test Baseball Data';
   plot residual*predict / haxis=axis1 vaxis=axis2;

run;
quit;
Example 3: Neural Network Hill-and-Plateau Example (Surf Data)

This example demonstrates how to develop a neural network model for a continuous target. A multilayer perceptron architecture is employed with 3 and then 30 hidden units. The example test data set is named SAMPSIO.DMDSURF (Surf Data). It contains the interval target HIPL, and two interval inputs X1 and X2. The data set was artificially generated as a surface containing a hill and a plateau. The hill is easily learned by an RBF architecture. The plateau is easily learned by an MLP architecture.

The SAMPSIO.DMTSURF data set is a test data set that is scored using the scoring formula from the trained model. The SAMPSIO.DMDSURF and SAMPSIO.DMTSURF data sets and the SAMPIO.DMDSURF catalog are stored in the sample library.

Program: 3 Hidden Units

```
%let hidden=3;
proc neural data=sampsio.dmdsurf
dmdbcat=sampsio.dmdsurf
    random=789;
input x1 x2 / id=i;
target hipl / id=o;
    hidden &hidden / id=h;
    prelim 10;
    train maxiter=1000 outest=mlpest;
    score data=sampsio.dmtsurf out=mlpout outfit=mlpfit;
    title2 "MLP with &hidden Hidden Units";
run;
```

```
proc print data=mlpfit noobs label;
    var _tase_ _tasel_ _taseu_;
    where _name_ = 'HIPL';
    title3 'Fit Statistics for the Test Data';
run;
```

```
proc gcontour data=mlpout;
    plot x2*x1=p_hipl / pattern ctext=black coutline=gray;
    pattern v=msolid;
    legend frame;
    title3 'Predicted Values';
    footnote;
run;
```
proc g3d data=mlpout;
  plot x2*x1=p_hipl / grid side ctop=blue
caxis=green ctext=black
  zmin=-1.5 zmax=1.5;
run;

Output: 3 Hidden Units

PROC GCONTOUR Plot of the Surf Training Data

Hill & Plateau Data
MLP with 3 Hidden Units

<table>
<thead>
<tr>
<th>Iteration number</th>
<th>Pseudo-random number seed</th>
<th>Objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>789</td>
<td>0.04023</td>
</tr>
<tr>
<td>1</td>
<td>761237432</td>
<td>0.03673</td>
</tr>
<tr>
<td>2</td>
<td>1092694980</td>
<td>0.04187</td>
</tr>
<tr>
<td>3</td>
<td>577625332</td>
<td>0.05216</td>
</tr>
<tr>
<td>4</td>
<td>261548896</td>
<td>0.03593</td>
</tr>
<tr>
<td>5</td>
<td>616485149</td>
<td>0.04030</td>
</tr>
<tr>
<td>6</td>
<td>692687363</td>
<td>0.03973</td>
</tr>
<tr>
<td>7</td>
<td>1510804008</td>
<td>0.04141</td>
</tr>
<tr>
<td>8</td>
<td>1385020003</td>
<td>0.03583</td>
</tr>
<tr>
<td>9</td>
<td>1070679467</td>
<td>0.04167</td>
</tr>
</tbody>
</table>

Hill Plateau Response Surface

PROC Neural Output

Hill & Plateau Data
MLP with 3 Hidden Units
### Optimization Start

**Parameter Estimates**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1_H1</td>
<td>6.064004</td>
<td>-2.674E-6</td>
<td>X1 -&gt; H1</td>
</tr>
<tr>
<td>X2_H1</td>
<td>0.880274</td>
<td>5.2841E-6</td>
<td>X2 -&gt; H1</td>
</tr>
<tr>
<td>X1_H2</td>
<td>0.048809</td>
<td>5.3735E-7</td>
<td>X1 -&gt; H2</td>
</tr>
<tr>
<td>X2_H2</td>
<td>-4.98958</td>
<td>6.0387E-7</td>
<td>X2 -&gt; H2</td>
</tr>
<tr>
<td>X1_H3</td>
<td>-5.916343</td>
<td>-6.0486E-6</td>
<td>X1 -&gt; H3</td>
</tr>
<tr>
<td>X2_H3</td>
<td>0.730854</td>
<td>-0.0000207</td>
<td>X2 -&gt; H3</td>
</tr>
<tr>
<td>BIAS_H1</td>
<td>-3.004936</td>
<td>-0.0000105</td>
<td>BIAS -&gt; H1</td>
</tr>
<tr>
<td>BIAS_H2</td>
<td>1.791982</td>
<td>2.17127E-6</td>
<td>BIAS -&gt; H2</td>
</tr>
<tr>
<td>BIAS_H3</td>
<td>0.864474</td>
<td>-0.0000126</td>
<td>BIAS -&gt; H3</td>
</tr>
<tr>
<td>H1_HIPL</td>
<td>-0.261095</td>
<td>0.0000179</td>
<td>H1 -&gt; HIPL</td>
</tr>
<tr>
<td>H2_HIPL</td>
<td>-0.484358</td>
<td>-0.0000107</td>
<td>H2 -&gt; HIPL</td>
</tr>
<tr>
<td>H3_HIPL</td>
<td>-0.265490</td>
<td>0.0000123</td>
<td>H3 -&gt; HIPL</td>
</tr>
<tr>
<td>BIAS_HIP</td>
<td>-0.490112</td>
<td>-0.0000182</td>
<td>BIAS -&gt; HIPL</td>
</tr>
</tbody>
</table>

**Value of Objective Function** = 0.0358271766

---

### Hill & Plateau Data

**MLP with 3 Hidden Units**

**Levenberg-Marquardt Optimization**

*** Termination Criteria ***

- Minimum Iterations: -1
- Maximum Iterations: 1000
- Maximum Function Calls: 2147483647
- Maximum CPU Time: 604800
- ABSGCONV Gradient Criterion: 0.0000100
- GCONV Gradient Criterion: 1E-8
- GCONV2 Gradient Criterion: 0
- ABSFCOMP Function Criterion: 0
- FCONV Function Criterion: 0.0001000
- FCONV2 Function Criterion: 0
- FSIZE Parameter: 0
- ABSXCONV Parameter Change Criterion: 0
- XCONV Parameter Change Criterion: 0
- XSIZE Parameter: 0
- ABSFCOMP Function Criterion: -1.341E154

*** Other Control Parameters ***

- Trust Region Initial Radius Factor: 1.00000
- Singularity Tolerance (SINGULAR): 1E-8

---

**Hill & Plateau Data**

**MLP with 3 Hidden Units**

**Levenberg-Marquardt Optimization**

Scaling Update of More (1978)

Number of Parameter Estimates: 13

**Optimization Start:** Active Constraints= 0  Criterion= 0.036

Maximum Gradient Element= 0.0000  Radius= 1.000

<table>
<thead>
<tr>
<th>Iter rest nfun act</th>
<th>optcrit</th>
<th>difcrit</th>
<th>maxgrad</th>
<th>lambda</th>
<th>rho</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0.0358</td>
<td>1.989E-7</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0.0358</td>
<td>1.764E-7</td>
</tr>
</tbody>
</table>
3 0 4 0  0.0358  1.553E-7  0.00002  0  0.869
4 0 5 0  0.0358  1.44E-7  0.00002  0  1.043
5 0 6 0  0.0358  1.351E-7  0.00001  0  1.247
6 0 7 0  0.0358  1.297E-7  0.00001  0  1.415
7 0 8 0  0.0358  1.259E-7  0.00001  0  1.585
8 0 9 0  0.0358  1.235E-7  0.00001  0  1.700
9 0 10 0  0.0358  1.221E-7  0.00001  0  1.805
10 0 11 0  0.0358  1.213E-7  0.00001  0  1.864

Optimization Results: Iterations= 10 Function Calls= 12 Jacobian Calls= 11
Active Constraints= 0  Criterion= 0.035825744
Maximum Gradient Element= 0.0000127206 Lambda= 0 Rho= 1.864 Radius= 0.004365

NOTE:  FCONV convergence criterion satisfied.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 X1_H1</td>
<td>6.020084</td>
<td>-4.8867E-7 X1 -&gt; H1</td>
</tr>
<tr>
<td>2 X2_H1</td>
<td>0.823365</td>
<td>0.0000111 X2 -&gt; H1</td>
</tr>
<tr>
<td>3 X1_H2</td>
<td>0.049663</td>
<td>-3.2277E-8 X1 -&gt; H2</td>
</tr>
<tr>
<td>4 X2_H2</td>
<td>-4.986906</td>
<td>2.19755E-8 X2 -&gt; H2</td>
</tr>
<tr>
<td>5 X1_H3</td>
<td>-5.848915</td>
<td>-1.8967E-6 X1 -&gt; H3</td>
</tr>
<tr>
<td>6 X2_H3</td>
<td>0.767294</td>
<td>-0.0000127 X2 -&gt; H3</td>
</tr>
<tr>
<td>7 BIAS_H1</td>
<td>-3.013469</td>
<td>5.59999E-7 BIAS -&gt; H1</td>
</tr>
<tr>
<td>8 BIAS_H2</td>
<td>1.791192</td>
<td>1.32261E-7 BIAS -&gt; H2</td>
</tr>
<tr>
<td>9 BIAS_H3</td>
<td>0.889276</td>
<td>-3.0043E-6 BIAS -&gt; H3</td>
</tr>
<tr>
<td>10 H1_HIPL</td>
<td>-0.262306</td>
<td>4.51418E-7 H1 -&gt; HIPL</td>
</tr>
<tr>
<td>11 H2_HIPL</td>
<td>-0.484458</td>
<td>1.7993E-8 H2 -&gt; HIPL</td>
</tr>
<tr>
<td>12 H3_HIPL</td>
<td>-0.266660</td>
<td>3.35605E-7 H3 -&gt; HIPL</td>
</tr>
<tr>
<td>13 BIAS_HIP</td>
<td>-0.490183</td>
<td>-3.1205E-7 BIAS -&gt; HIPL</td>
</tr>
</tbody>
</table>

Value of Objective Function = 0.0358257445

PROC PRINT Report of Selected Fit Statistics for the Scored Test Data Set

<table>
<thead>
<tr>
<th>Test: Average Squared Error</th>
<th>Test: Lower 95% Conf.</th>
<th>Test: Upper 95% Conf.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.036830</td>
<td>0.028999</td>
<td>0.045583</td>
</tr>
</tbody>
</table>

GCONTOUR Plot of the Predicted Values
Program: 30 Hidden Units

title 'Hill & Plateau Data';
%let hidden=30;
proc neural data=sampsio.dmdsurf
   dmdbcat=sampsio.dmdsurf
   ;

random=789;
input x1 x2 / id=i;
target hipl / id=o;

hidden &hidden / id=h;
prelim 10;
train maxiter=1000 outest=mlpest2;
score data=sampsio.dmtsurf out=mlpout2 outfit=mlpfit2;
title2 "MLP with &hidden Hidden Units";
run;

proc print data=mlpfit2 noobs label;
var _tase_ _tasel_ _taseu_; 
where _name_ = 'HIPL';
title3 'Fit Statistics for the Test Data';
run;

proc gcontour data=mlpout2;
plot x2*x1=p_hipl / pattern ctext=black coutline=gray;
pattern v=msolid;
legend frame;
title3 'Predicted Values';
footnote;
run;

proc g3d data=mlpout2;
plot x2*x1=p_hipl / grid side ctop=blue caxis=green ctext=black 
zmin=-1.5 zmax=1.5;
run;

Output: 30 Hidden Units

Preliminary Iteration History

<table>
<thead>
<tr>
<th>Iteration number</th>
<th>Pseudo-random number</th>
<th>Objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>789</td>
<td>0.03400</td>
</tr>
<tr>
<td>1</td>
<td>1095609817</td>
<td>0.04237</td>
</tr>
<tr>
<td>2</td>
<td>924680074</td>
<td>0.03808</td>
</tr>
<tr>
<td>3</td>
<td>1369944093</td>
<td>0.04887</td>
</tr>
<tr>
<td>4</td>
<td>1527099570</td>
<td>0.04153</td>
</tr>
<tr>
<td>5</td>
<td>430173087</td>
<td>0.03480</td>
</tr>
<tr>
<td>6</td>
<td>739241177</td>
<td>0.03857</td>
</tr>
<tr>
<td>7</td>
<td>321367798</td>
<td>0.02955</td>
</tr>
<tr>
<td>8</td>
<td>58127801</td>
<td>0.04378</td>
</tr>
<tr>
<td>9</td>
<td>1974465768</td>
<td>0.02971</td>
</tr>
</tbody>
</table>

Optimization Start

Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 X1_H1</td>
<td>-0.467800</td>
<td>0.0000599 X1 -&gt; H1</td>
</tr>
<tr>
<td>2 X2_H1</td>
<td>-0.406536</td>
<td>0.0002672 X2 -&gt; H1</td>
</tr>
<tr>
<td>3 X1_H2</td>
<td>-1.507694</td>
<td>0.0002234 X1 -&gt; H2</td>
</tr>
<tr>
<td>Parameter</td>
<td>Estimate 1</td>
<td>Gradient 1</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>X2_H2</td>
<td>0.547571</td>
<td>-0.0003621</td>
</tr>
<tr>
<td>X1_H3</td>
<td>0.565220</td>
<td>0.0000279</td>
</tr>
<tr>
<td>X2_H3</td>
<td>0.339673</td>
<td>0.0008228</td>
</tr>
<tr>
<td>X1_H4</td>
<td>-0.832673</td>
<td>0.0000150</td>
</tr>
<tr>
<td>X2_H4</td>
<td>0.282776</td>
<td>-0.0004894</td>
</tr>
<tr>
<td>X1_H5</td>
<td>-0.129020</td>
<td>0.0000243</td>
</tr>
<tr>
<td>X2_H5</td>
<td>-0.031309</td>
<td>0.0001306</td>
</tr>
<tr>
<td>X1_H6</td>
<td>-0.060165</td>
<td>0.0002855</td>
</tr>
<tr>
<td>X2_H6</td>
<td>1.691299</td>
<td>0.0001026</td>
</tr>
<tr>
<td>X1_H7</td>
<td>0.787984</td>
<td>0.0001247</td>
</tr>
<tr>
<td>X2_H7</td>
<td>0.062126</td>
<td>0.0007617</td>
</tr>
<tr>
<td>X1_H8</td>
<td>-0.150388</td>
<td>0.0000205</td>
</tr>
<tr>
<td>X2_H8</td>
<td>-0.118346</td>
<td>0.0002827</td>
</tr>
<tr>
<td>X1_H9</td>
<td>-1.319054</td>
<td>-0.00112</td>
</tr>
<tr>
<td>X2_H9</td>
<td>-0.856195</td>
<td>0.00424</td>
</tr>
<tr>
<td>X1_H10</td>
<td>0.369538</td>
<td>-0.003333</td>
</tr>
<tr>
<td>X2_H10</td>
<td>0.149047</td>
<td>0.004931</td>
</tr>
<tr>
<td>X1_H11</td>
<td>-0.227129</td>
<td>0.0002816</td>
</tr>
<tr>
<td>X2_H11</td>
<td>-0.867869</td>
<td>0.00100</td>
</tr>
<tr>
<td>X3_H12</td>
<td>-1.208871</td>
<td>0.0000539</td>
</tr>
<tr>
<td>X2_H12</td>
<td>0.229684</td>
<td>-0.0008334</td>
</tr>
<tr>
<td>X1_H13</td>
<td>-0.774966</td>
<td>0.004976</td>
</tr>
<tr>
<td>X2_H13</td>
<td>0.479537</td>
<td>-0.00141</td>
</tr>
<tr>
<td>X1_H14</td>
<td>0.092770</td>
<td>-0.0006323</td>
</tr>
<tr>
<td>X2_H14</td>
<td>-1.080409</td>
<td>-0.0004827</td>
</tr>
<tr>
<td>X1_H15</td>
<td>-0.622859</td>
<td>0.00147</td>
</tr>
<tr>
<td>X2_H15</td>
<td>-0.772737</td>
<td>0.00172</td>
</tr>
<tr>
<td>X1_H16</td>
<td>-0.138595</td>
<td>-0.00141</td>
</tr>
<tr>
<td>X2_H16</td>
<td>0.822709</td>
<td>0.00251</td>
</tr>
<tr>
<td>X1_H17</td>
<td>-0.262409</td>
<td>0.0001737</td>
</tr>
<tr>
<td>X2_H17</td>
<td>-1.162618</td>
<td>0.0004185</td>
</tr>
<tr>
<td>X1_H18</td>
<td>0.030999</td>
<td>-0.00369</td>
</tr>
<tr>
<td>X2_H18</td>
<td>-2.728742</td>
<td>0.00228</td>
</tr>
<tr>
<td>X1_H19</td>
<td>0.021237</td>
<td>-0.00180</td>
</tr>
<tr>
<td>X2_H19</td>
<td>1.362841</td>
<td>-0.00108</td>
</tr>
<tr>
<td>X1_H20</td>
<td>-0.429323</td>
<td>2.98698E-6</td>
</tr>
<tr>
<td>X2_H20</td>
<td>0.299725</td>
<td>-2.3418E-6</td>
</tr>
<tr>
<td>X1_H21</td>
<td>0.155524</td>
<td>0.0004407</td>
</tr>
<tr>
<td>X2_H21</td>
<td>-0.563073</td>
<td>0.0002520</td>
</tr>
<tr>
<td>X1_H22</td>
<td>-0.110919</td>
<td>5.39502E-6</td>
</tr>
<tr>
<td>X2_H22</td>
<td>-0.338627</td>
<td>-8.6108E-6</td>
</tr>
<tr>
<td>X1_H23</td>
<td>1.805185</td>
<td>-0.00197</td>
</tr>
<tr>
<td>X2_H23</td>
<td>0.230073</td>
<td>-0.0007495</td>
</tr>
<tr>
<td>X1_H24</td>
<td>-1.168753</td>
<td>-0.0001878</td>
</tr>
<tr>
<td>X2_H24</td>
<td>-0.299062</td>
<td>-0.000261</td>
</tr>
<tr>
<td>X1_H25</td>
<td>0.234343</td>
<td>-0.0000412</td>
</tr>
<tr>
<td>X2_H25</td>
<td>-0.209074</td>
<td>-0.0000807</td>
</tr>
</tbody>
</table>

Optimization Start
Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate 1</th>
<th>Gradient 1</th>
<th>Label 1</th>
<th>Estimate 2</th>
<th>Gradient 2</th>
<th>Label 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1_H26</td>
<td>-0.055756</td>
<td>-0.0000101</td>
<td>X1 -&gt;  H26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2_H26</td>
<td>-0.331826</td>
<td>0.0001698</td>
<td>X2 -&gt;  H26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1_H27</td>
<td>-0.799268</td>
<td>0.0001618</td>
<td>X1 -&gt;  H27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2_H27</td>
<td>-0.195293</td>
<td>0.0005865</td>
<td>X2 -&gt;  H27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1_H28</td>
<td>-1.130014</td>
<td>0.0006003</td>
<td>X1 -&gt;  H28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2_H28</td>
<td>0.540507</td>
<td>-0.000117</td>
<td>X2 -&gt;  H28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1_H29</td>
<td>-0.808714</td>
<td>0.0000157</td>
<td>X1 -&gt;  H29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X2_H29</td>
<td>0.442020</td>
<td>-0.0003948</td>
<td>X2 -&gt;  H29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X1_H30</td>
<td>-1.075334</td>
<td>-0.0008229</td>
<td>X1 -&gt;  H30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Estimate</td>
<td>Gradient</td>
<td>Label</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------</td>
<td>------------</td>
<td>----------</td>
<td>---------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H1</td>
<td>1.652625</td>
<td>0.0002540</td>
<td>H1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H2</td>
<td>0.729833</td>
<td>0.0007927</td>
<td>H2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H3</td>
<td>-0.029132</td>
<td>0.0002571</td>
<td>H3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H4</td>
<td>-0.554034</td>
<td>0.000857</td>
<td>H4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H5</td>
<td>1.242018</td>
<td>0.000550</td>
<td>H5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H6</td>
<td>-0.429913</td>
<td>-0.00191</td>
<td>H6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H7</td>
<td>0.171418</td>
<td>-0.003334</td>
<td>H7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H8</td>
<td>0.668461</td>
<td>0.001098</td>
<td>H8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H9</td>
<td>-0.985128</td>
<td>-0.00175</td>
<td>H9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H10</td>
<td>0.362497</td>
<td>0.000239</td>
<td>H10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H11</td>
<td>-0.728020</td>
<td>-0.002738</td>
<td>H11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H12</td>
<td>0.742591</td>
<td>-0.004105</td>
<td>H12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H13</td>
<td>-1.142064</td>
<td>-0.004769</td>
<td>H13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H14</td>
<td>0.456503</td>
<td>0.001368</td>
<td>H14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H15</td>
<td>1.429261</td>
<td>0.00137</td>
<td>H15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H16</td>
<td>-1.405924</td>
<td>0.001099</td>
<td>H16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H17</td>
<td>-0.861128</td>
<td>-0.000339</td>
<td>H17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H18</td>
<td>-0.888192</td>
<td>-0.00480</td>
<td>H18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H19</td>
<td>-1.509340</td>
<td>-0.00139</td>
<td>H19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H20</td>
<td>0.431651</td>
<td>6.87045E-7</td>
<td>H20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H21</td>
<td>-0.018080</td>
<td>-0.000773</td>
<td>H21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H22</td>
<td>-0.663759</td>
<td>7.53178E-7</td>
<td>H22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H23</td>
<td>0.178667</td>
<td>0.00162</td>
<td>H23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H24</td>
<td>0.515460</td>
<td>-0.0006172</td>
<td>H24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H25</td>
<td>-0.721835</td>
<td>-0.000149</td>
<td>H25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H26</td>
<td>-0.241293</td>
<td>0.0000243</td>
<td>H26</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H27</td>
<td>0.670665</td>
<td>0.000766</td>
<td>H27</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H28</td>
<td>0.255718</td>
<td>0.0005082</td>
<td>H28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H29</td>
<td>0.195161</td>
<td>0.000364</td>
<td>H29</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BIAS_H30</td>
<td>1.509699</td>
<td>-0.00366</td>
<td>H30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H1_HIPL</td>
<td>-0.136499</td>
<td>-0.00160</td>
<td>H1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2_HIPL</td>
<td>-0.091000</td>
<td>-0.00300</td>
<td>H2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H3_HIPL</td>
<td>-0.144627</td>
<td>-0.00174</td>
<td>H3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H4_HIPL</td>
<td>0.175297</td>
<td>-0.00214</td>
<td>H4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H5_HIPL</td>
<td>-0.078282</td>
<td>-0.00190</td>
<td>H5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H6_HIPL</td>
<td>0.803260</td>
<td>-0.00174</td>
<td>H6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H7_HIPL</td>
<td>-0.132805</td>
<td>0.0004925</td>
<td>H7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H8_HIPL</td>
<td>-0.057435</td>
<td>-0.0008912</td>
<td>H8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H9_HIPL</td>
<td>-0.333394</td>
<td>0.0005940</td>
<td>H9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H10_HIPL</td>
<td>-0.092374</td>
<td>-0.004988</td>
<td>H10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H11_HIPL</td>
<td>-0.152927</td>
<td>0.00433</td>
<td>H11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H12_HIPL</td>
<td>0.117821</td>
<td>-0.00294</td>
<td>H12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H13_HIPL</td>
<td>0.206879</td>
<td>-0.00262</td>
<td>H13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H14_HIPL</td>
<td>-0.302093</td>
<td>0.00236</td>
<td>H14</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H15_HIPL</td>
<td>-0.221776</td>
<td>0.00168</td>
<td>H15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H16_HIPL</td>
<td>-0.366827</td>
<td>-0.00201</td>
<td>H16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H17_HIPL</td>
<td>-0.062601</td>
<td>0.00624</td>
<td>H17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H18_HIPL</td>
<td>0.693210</td>
<td>0.00477</td>
<td>H18</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H19_HIPL</td>
<td>-0.438978</td>
<td>-0.00270</td>
<td>H19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H20_HIPL</td>
<td>0.004246</td>
<td>-0.00201</td>
<td>H20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H21_HIPL</td>
<td>-0.195547</td>
<td>0.00236</td>
<td>H21</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H22_HIPL</td>
<td>0.004032</td>
<td>0.00203</td>
<td>H22</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H23_HIPL</td>
<td>0.307420</td>
<td>-0.00475</td>
<td>H23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H24_HIPL</td>
<td>0.391365</td>
<td>0.00300</td>
<td>H24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H25_HIPL</td>
<td>0.051031</td>
<td>0.00192</td>
<td>H25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H26_HIPL</td>
<td>-0.041873</td>
<td>0.00200</td>
<td>H26</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Optimization Start
Parameter Estimates
Hill & Plateau Data
MLP with 30 Hidden Units

Dual Quasi-Newton Optimization

*** Termination Criteria ***
Minimum Iterations ........................................... -1
Maximum Iterations ............................................ 1000
Maximum Function Calls ............................................. 2147483647
Maximum CPU Time .............................................. 604800
ABSGCONV Gradient Criterion .......................... 0.0000100
GCONV Gradient Criterion ................................. 1E-8
ABSFCNV Function Criterion ...................... 0
FCONV Function Criterion ................................. 0.0001000
FCONV2 Function Criterion ............................. 0
FSIZE Parameter .............................................. 0
ABSXCONV Parameter Change Criterion .............. 0
XCONV Parameter Change Criterion ........................ 0
XSIZE Parameter .............................................. 0
ABSCONV Function Criterion .................. -1.341E154

*** Other Control Parameters ***
Line Search Method 2: Starting Alpha .................. 1.00000
Line Search Precision LSPRECISION ................... 0.40000
DAMPSTEP Parameter for Line Search ................... 
Singularity Tolerance (SINGULAR) .................. 1E-8

Hill & Plateau Data
MLP with 30 Hidden Units

Dual Quasi-Newton Optimization
Dual Broyden - Fletcher - Goldfarb - Shanno Update (DBFGS)
Number of Parameter Estimates 121

Optimization Start: Active Constraints= 0 Criterion= 0.030
Maximum Gradient Element= 0.006

<table>
<thead>
<tr>
<th>Iter</th>
<th>rest</th>
<th>nfun</th>
<th>act</th>
<th>optcrit</th>
<th>dicfcrit</th>
<th>maxgrad</th>
<th>alpha</th>
<th>slope</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0.0295</td>
<td>0.000041</td>
<td>0.00581</td>
<td>0.00225</td>
<td>-0.0366</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>6</td>
<td>0</td>
<td>0.0294</td>
<td>0.000099</td>
<td>0.0145</td>
<td>1.949</td>
<td>-0.0001</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0.0286</td>
<td>0.000844</td>
<td>0.00638</td>
<td>2.096</td>
<td>-0.0008</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0.0285</td>
<td>0.000034</td>
<td>0.0152</td>
<td>1.000</td>
<td>-0.0002</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>11</td>
<td>0</td>
<td>0.0281</td>
<td>0.000388</td>
<td>0.0119</td>
<td>3.029</td>
<td>-0.0003</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0.0279</td>
<td>0.000244</td>
<td>0.00928</td>
<td>2.339</td>
<td>-0.0002</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>0.0274</td>
<td>0.000543</td>
<td>0.0259</td>
<td>8.913</td>
<td>-0.0001</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0.0271</td>
<td>0.000249</td>
<td>0.0174</td>
<td>1.560</td>
<td>-0.0009</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>17</td>
<td>0</td>
<td>0.0269</td>
<td>0.000243</td>
<td>0.00931</td>
<td>1.422</td>
<td>-0.0004</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>18</td>
<td>0</td>
<td>0.0266</td>
<td>0.000247</td>
<td>0.0165</td>
<td>5.537</td>
<td>-0.0001</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>0.0252</td>
<td>0.00142</td>
<td>0.0294</td>
<td>7.860</td>
<td>-0.0004</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>22</td>
<td>0</td>
<td>0.0245</td>
<td>0.000745</td>
<td>0.0104</td>
<td>1.614</td>
<td>-0.0009</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>0.0241</td>
<td>0.000382</td>
<td>0.0367</td>
<td>5.977</td>
<td>-0.0003</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>24</td>
<td>0</td>
<td>0.0235</td>
<td>0.000613</td>
<td>0.0177</td>
<td>0.737</td>
<td>-0.0013</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>25</td>
<td>0</td>
<td>0.0233</td>
<td>0.00019</td>
<td>0.0322</td>
<td>2.407</td>
<td>-0.0006</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>27</td>
<td>0</td>
<td>0.0226</td>
<td>0.00064</td>
<td>0.00916</td>
<td>1.459</td>
<td>-0.0009</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>28</td>
<td>0</td>
<td>0.0216</td>
<td>0.00104</td>
<td>0.0128</td>
<td>4.153</td>
<td>-0.0004</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>30</td>
<td>0</td>
<td>0.0212</td>
<td>0.000426</td>
<td>0.00864</td>
<td>0.744</td>
<td>-0.0009</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>0.0209</td>
<td>0.000299</td>
<td>0.0107</td>
<td>2.066</td>
<td>-0.0003</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>34</td>
<td>0</td>
<td>0.0200</td>
<td>0.00082</td>
<td>0.0182</td>
<td>3.925</td>
<td>-0.0004</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>36</td>
<td>0</td>
<td>0.0195</td>
<td>0.000538</td>
<td>0.00766</td>
<td>1.034</td>
<td>-0.0011</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>38</td>
<td>0</td>
<td>0.0193</td>
<td>0.000188</td>
<td>0.00977</td>
<td>1.467</td>
<td>-0.0002</td>
</tr>
</tbody>
</table>
******** Several Iterations Have Been Omitted ********

Hill & Plateau Data
MLP with 30 Hidden Units

********** Continuation of Iteration History **********

979  16 1597   0  0.000759  8.218E-7  0.0001   2.057 -658E-9
980  16 1598   0  0.000758  6.835E-7  0.00017  3.401 -604E-9
981  16 1599   0  0.000758  6.446E-7  0.00006  2.656 -643E-9
982  16 1600   0  0.000757  5.563E-7  0.00011  3.160 -51E-8
983  16 1601   0  0.000757  2.735E-7  0.00015  3.099 -449E-9
984  16 1602   0  0.000756  3.989E-7  0.00011  1.225 -558E-9
985  16 1603   0  0.000756  4.022E-7  0.00008  4.517 -221E-9
986  16 1604   0  0.000756  3.554E-7  0.00009  3.182 -316E-9
987  16 1605   0  0.000755  3.794E-7  0.00012  2.367 -377E-9
988  16 1606   0  0.000755  6.098E-7  0.00007  2.750 -345E-9
989  16 1608   0  0.000754  4.074E-7  0.0001   1.886 -432E-9
990  16 1609   0  0.000754  4.944E-7  0.00011  4.422 -23E-8
991  16 1610   0  0.000753  4.063E-7  0.00013  5.307 -233E-9
992  16 1611   0  0.000753  3.567E-7  0.00009  2.877 -353E-9
993  16 1612   0  0.000752  5.221E-7  0.00009  2.367 -377E-9
994  16 1613   0  0.000752  5.499E-7  0.00017  3.150 -414E-9
995  16 1614   0  0.000752  3.433E-7  0.00012  4.313 -319E-9
996  16 1615   0  0.000751  4.585E-7  0.0001   1.603 -536E-9
997  16 1616   0  0.000750  7.197E-7  0.0001   4.005 -286E-9
998  16 1618   0  0.000750  4.155E-7  0.00017  1.828 -455E-9
999  16 1619   0  0.000750  4.191E-7  0.00019  5.233 -199E-9
1000 16 1620  0  0.000749  6.838E-7  0.00006  2.696 -389E-9

Optimization Results: Iterations= 1000 Function Calls= 1621
Gradient Calls= 1310 Active Constraints= 0  Criterion= 0.00074889869
Maximum Gradient Element= 0.0000589356 Slope= -3.8861E-7

WARNING: QUANEW Optimization cannot be completed.
WARNING: QUANEW needs more than 1000 iterations or 2147483647 function

Hill & Plateau Data
MLP with 30 Hidden Units

Optimization Results
Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1_H1</td>
<td>-0.532039</td>
<td>2.3205E-6</td>
</tr>
<tr>
<td>X2_H1</td>
<td>-0.125752</td>
<td>-1.2541E-6</td>
</tr>
<tr>
<td>X1_H2</td>
<td>-3.790662</td>
<td>-0.0000317</td>
</tr>
<tr>
<td>X2_H2</td>
<td>1.651981</td>
<td>-0.0000306</td>
</tr>
<tr>
<td>X1_H3</td>
<td>1.495431</td>
<td>-0.0000166</td>
</tr>
<tr>
<td>X2_H3</td>
<td>1.222886</td>
<td>-4.1724E-7</td>
</tr>
<tr>
<td>X1_H4</td>
<td>-3.447110</td>
<td>8.67875E-6</td>
</tr>
<tr>
<td>Parameter</td>
<td>Estimate</td>
<td>Gradient</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>X2_H4</td>
<td>-0.166308</td>
<td>0.0000402</td>
</tr>
<tr>
<td>X1_H5</td>
<td>-1.365956</td>
<td>-3.7060E-6</td>
</tr>
<tr>
<td>X2_H5</td>
<td>-0.107199</td>
<td>9.34775E-7</td>
</tr>
<tr>
<td>X1_H6</td>
<td>-0.001925</td>
<td>0.0000471</td>
</tr>
<tr>
<td>X2_H6</td>
<td>2.893215</td>
<td>-0.0000366</td>
</tr>
<tr>
<td>X1_H7</td>
<td>2.017056</td>
<td>2.84015E-6</td>
</tr>
<tr>
<td>X2_H7</td>
<td>0.101604</td>
<td>9.34775E-7</td>
</tr>
<tr>
<td>X1_H8</td>
<td>-0.569829</td>
<td>9.05793E-8</td>
</tr>
<tr>
<td>X2_H8</td>
<td>-0.162453</td>
<td>-1.5387E-6</td>
</tr>
<tr>
<td>X1_H9</td>
<td>-1.840300</td>
<td>0.0000153</td>
</tr>
<tr>
<td>X2_H9</td>
<td>-2.672442</td>
<td>-2.0747E-6</td>
</tr>
<tr>
<td>X1_H10</td>
<td>1.677979</td>
<td>0.000232</td>
</tr>
<tr>
<td>X2_H10</td>
<td>0.327821</td>
<td>-0.0000115</td>
</tr>
<tr>
<td>X1_H11</td>
<td>-2.323530</td>
<td>0.0000175</td>
</tr>
<tr>
<td>X2_H11</td>
<td>-2.538369</td>
<td>0.0000162</td>
</tr>
<tr>
<td>X1_H12</td>
<td>-2.783121</td>
<td>0.0000251</td>
</tr>
<tr>
<td>X2_H12</td>
<td>0.493515</td>
<td>0.0000307</td>
</tr>
<tr>
<td>X1_H13</td>
<td>-1.702463</td>
<td>0.0000212</td>
</tr>
<tr>
<td>X2_H13</td>
<td>1.200148</td>
<td>0.0000554</td>
</tr>
<tr>
<td>X1_H14</td>
<td>1.327628</td>
<td>-9.5967E-7</td>
</tr>
<tr>
<td>X2_H14</td>
<td>-1.702246</td>
<td>-0.0000117</td>
</tr>
<tr>
<td>X1_H15</td>
<td>-1.999156</td>
<td>7.50953E-6</td>
</tr>
<tr>
<td>X2_H15</td>
<td>-2.076552</td>
<td>-6.65E-6</td>
</tr>
<tr>
<td>X1_H16</td>
<td>0.026058</td>
<td>1.94184E-4</td>
</tr>
<tr>
<td>X2_H16</td>
<td>0.795836</td>
<td>0.0000344</td>
</tr>
<tr>
<td>X1_H17</td>
<td>0.296149</td>
<td>-0.0000104</td>
</tr>
<tr>
<td>X2_H17</td>
<td>-2.901058</td>
<td>0.0000123</td>
</tr>
<tr>
<td>X1_H18</td>
<td>-1.347832</td>
<td>-0.0000242</td>
</tr>
<tr>
<td>X2_H18</td>
<td>-4.329716</td>
<td>8.39321E-6</td>
</tr>
<tr>
<td>X1_H19</td>
<td>2.261145</td>
<td>0.0000117</td>
</tr>
<tr>
<td>X2_H19</td>
<td>2.520761</td>
<td>-0.0000350</td>
</tr>
<tr>
<td>X1_H20</td>
<td>-0.829191</td>
<td>2.07643E-6</td>
</tr>
<tr>
<td>X2_H20</td>
<td>1.016295</td>
<td>-0.0000459</td>
</tr>
<tr>
<td>X1_H21</td>
<td>1.004049</td>
<td>-0.0000209</td>
</tr>
<tr>
<td>X2_H21</td>
<td>-1.005880</td>
<td>5.41321E-6</td>
</tr>
<tr>
<td>X1_H22</td>
<td>0.240370</td>
<td>0.0000339</td>
</tr>
<tr>
<td>X2_H22</td>
<td>-0.516315</td>
<td>0.0000241</td>
</tr>
<tr>
<td>X1_H23</td>
<td>3.033531</td>
<td>2.09368E-6</td>
</tr>
<tr>
<td>X2_H23</td>
<td>0.824219</td>
<td>7.6337E-6</td>
</tr>
<tr>
<td>X1_H24</td>
<td>-3.072703</td>
<td>4.63047E-6</td>
</tr>
<tr>
<td>X2_H24</td>
<td>-1.511832</td>
<td>0.0000404</td>
</tr>
<tr>
<td>X1_H25</td>
<td>1.189984</td>
<td>-3.4502E-6</td>
</tr>
<tr>
<td>X2_H25</td>
<td>-3.514446</td>
<td>0.0000424</td>
</tr>
</tbody>
</table>

Hill & Plateau Data
MLP with 30 Hidden Units
Optimization Results
Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate 1</th>
<th>Gradient 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>X2_H26</td>
<td>-0.151007</td>
<td>0.0000270</td>
</tr>
<tr>
<td>X2_H26</td>
<td>-0.521042</td>
<td>0.0000298</td>
</tr>
<tr>
<td>X1_H27</td>
<td>-2.538401</td>
<td>0.0000145</td>
</tr>
<tr>
<td>X2_H27</td>
<td>0.012546</td>
<td>-0.0000355</td>
</tr>
<tr>
<td>X1_H28</td>
<td>-3.430341</td>
<td>0.0000158</td>
</tr>
<tr>
<td>X2_H28</td>
<td>2.720247</td>
<td>-0.0000208</td>
</tr>
<tr>
<td>X1_H29</td>
<td>-0.980648</td>
<td>7.7342E-6</td>
</tr>
<tr>
<td>X2_H29</td>
<td>0.485661</td>
<td>-0.0000146</td>
</tr>
<tr>
<td>X1_H30</td>
<td>-2.265832</td>
<td>9.67792E-6</td>
</tr>
<tr>
<td>X2_H30</td>
<td>3.053364</td>
<td>-1.7164E-6</td>
</tr>
</tbody>
</table>
Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>1.621410</td>
<td>3.94302E-7</td>
<td>BIAS -&gt; H1</td>
</tr>
<tr>
<td>H2</td>
<td>1.544387</td>
<td>-0.0000555</td>
<td>BIAS -&gt; H2</td>
</tr>
<tr>
<td>H3</td>
<td>-0.534732</td>
<td>0.0000123</td>
<td>BIAS -&gt; H3</td>
</tr>
<tr>
<td>H4</td>
<td>-0.785714</td>
<td>-0.0000101</td>
<td>BIAS -&gt; H4</td>
</tr>
<tr>
<td>H5</td>
<td>0.912585</td>
<td>-0.0000115</td>
<td>BIAS -&gt; H5</td>
</tr>
<tr>
<td>H6</td>
<td>-0.768781</td>
<td>-2.2982E-6</td>
<td>BIAS -&gt; H6</td>
</tr>
<tr>
<td>H7</td>
<td>1.015622</td>
<td>-9.1444E-6</td>
<td>BIAS -&gt; H7</td>
</tr>
<tr>
<td>H8</td>
<td>0.550634</td>
<td>-1.1615E-8</td>
<td>BIAS -&gt; H8</td>
</tr>
<tr>
<td>H9</td>
<td>-2.030613</td>
<td>-0.0000519</td>
<td>BIAS -&gt; H9</td>
</tr>
<tr>
<td>H10</td>
<td>0.306667</td>
<td>-9.3574E-6</td>
<td>BIAS -&gt; H10</td>
</tr>
<tr>
<td>H11</td>
<td>-2.591558</td>
<td>0.0000337</td>
<td>BIAS -&gt; H11</td>
</tr>
<tr>
<td>H12</td>
<td>1.854141</td>
<td>0.0000213</td>
<td>BIAS -&gt; H12</td>
</tr>
<tr>
<td>H13</td>
<td>-0.378726</td>
<td>-0.0000290</td>
<td>BIAS -&gt; H13</td>
</tr>
<tr>
<td>H14</td>
<td>-0.596093</td>
<td>-0.0000340</td>
<td>BIAS -&gt; H14</td>
</tr>
<tr>
<td>H15</td>
<td>1.200441</td>
<td>-0.0000417</td>
<td>BIAS -&gt; H15</td>
</tr>
<tr>
<td>H16</td>
<td>-1.358588</td>
<td>0.0000232</td>
<td>BIAS -&gt; H16</td>
</tr>
<tr>
<td>H17</td>
<td>-2.473932</td>
<td>0.0000370</td>
<td>BIAS -&gt; H17</td>
</tr>
<tr>
<td>H18</td>
<td>-1.391321</td>
<td>0.0000284</td>
<td>BIAS -&gt; H18</td>
</tr>
<tr>
<td>H19</td>
<td>-1.537930</td>
<td>-0.0000246</td>
<td>BIAS -&gt; H19</td>
</tr>
<tr>
<td>H20</td>
<td>0.717883</td>
<td>-0.0000180</td>
<td>BIAS -&gt; H20</td>
</tr>
<tr>
<td>H21</td>
<td>0.046762</td>
<td>-0.0000347</td>
<td>BIAS -&gt; H21</td>
</tr>
<tr>
<td>H22</td>
<td>-0.062103</td>
<td>-0.0000241</td>
<td>BIAS -&gt; H22</td>
</tr>
<tr>
<td>H23</td>
<td>0.090229</td>
<td>0.0000343</td>
<td>BIAS -&gt; H23</td>
</tr>
<tr>
<td>H24</td>
<td>0.502362</td>
<td>1.47214E-6</td>
<td>BIAS -&gt; H24</td>
</tr>
<tr>
<td>H25</td>
<td>-2.014471</td>
<td>-3.7539E-6</td>
<td>BIAS -&gt; H25</td>
</tr>
<tr>
<td>H26</td>
<td>-0.448834</td>
<td>7.76322E-6</td>
<td>BIAS -&gt; H26</td>
</tr>
<tr>
<td>H27</td>
<td>2.068868</td>
<td>0.0000197</td>
<td>BIAS -&gt; H27</td>
</tr>
<tr>
<td>H28</td>
<td>1.888770</td>
<td>0.0000405</td>
<td>BIAS -&gt; H28</td>
</tr>
<tr>
<td>H29</td>
<td>0.249710</td>
<td>-6.3913E-6</td>
<td>BIAS -&gt; H29</td>
</tr>
<tr>
<td>H30</td>
<td>3.409735</td>
<td>8.33278E-6</td>
<td>BIAS -&gt; H30</td>
</tr>
<tr>
<td>H1_HIPL</td>
<td>-0.110538</td>
<td>0.0000178</td>
<td>H1 -&gt; HIPL</td>
</tr>
<tr>
<td>H2_HIPL</td>
<td>-0.473553</td>
<td>-0.0000117</td>
<td>H2 -&gt; HIPL</td>
</tr>
<tr>
<td>H3_HIPL</td>
<td>-1.150299</td>
<td>0.0000209</td>
<td>H3 -&gt; HIPL</td>
</tr>
<tr>
<td>H4_HIPL</td>
<td>0.477358</td>
<td>-4.1644E-6</td>
<td>H4 -&gt; HIPL</td>
</tr>
<tr>
<td>H5_HIPL</td>
<td>0.464599</td>
<td>0.0000156</td>
<td>H5 -&gt; HIPL</td>
</tr>
<tr>
<td>H6_HIPL</td>
<td>0.868824</td>
<td>0.0000395</td>
<td>H6 -&gt; HIPL</td>
</tr>
<tr>
<td>H7_HIPL</td>
<td>-0.305023</td>
<td>2.29377E-6</td>
<td>H7 -&gt; HIPL</td>
</tr>
<tr>
<td>H8_HIPL</td>
<td>-0.022398</td>
<td>0.0000134</td>
<td>H8 -&gt; HIPL</td>
</tr>
<tr>
<td>H9_HIPL</td>
<td>-0.971155</td>
<td>-0.0000579</td>
<td>H9 -&gt; HIPL</td>
</tr>
<tr>
<td>H10_HIPL</td>
<td>0.974106</td>
<td>-0.0000192</td>
<td>H10 -&gt; HIPL</td>
</tr>
</tbody>
</table>

Optimization Results
Parameter Estimates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Gradient</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>H11_HIPL</td>
<td>0.568802</td>
<td>-0.0000519</td>
<td>H11 -&gt; HIPL</td>
</tr>
<tr>
<td>H12_HIPL</td>
<td>1.227553</td>
<td>0.0000166</td>
<td>H12 -&gt; HIPL</td>
</tr>
<tr>
<td>H13_HIPL</td>
<td>-0.466255</td>
<td>-0.0000231</td>
<td>H13 -&gt; HIPL</td>
</tr>
<tr>
<td>H14_HIPL</td>
<td>-0.894798</td>
<td>-0.0000322</td>
<td>H14 -&gt; HIPL</td>
</tr>
<tr>
<td>H15_HIPL</td>
<td>-1.479547</td>
<td>-2.2118E-6</td>
<td>H15 -&gt; HIPL</td>
</tr>
<tr>
<td>H16_HIPL</td>
<td>0.471993</td>
<td>0.0000190</td>
<td>H16 -&gt; HIPL</td>
</tr>
<tr>
<td>H17_HIPL</td>
<td>-0.695108</td>
<td>-0.0000138</td>
<td>H17 -&gt; HIPL</td>
</tr>
<tr>
<td>H18_HIPL</td>
<td>0.411108</td>
<td>-9.7179E-7</td>
<td>H18 -&gt; HIPL</td>
</tr>
<tr>
<td>H19_HIPL</td>
<td>-0.650073</td>
<td>2.01526E-7</td>
<td>H19 -&gt; HIPL</td>
</tr>
<tr>
<td>H20_HIPL</td>
<td>-1.728201</td>
<td>0.0000190</td>
<td>H20 -&gt; HIPL</td>
</tr>
<tr>
<td>H21_HIPL</td>
<td>-0.845374</td>
<td>0.0000182</td>
<td>H21 -&gt; HIPL</td>
</tr>
<tr>
<td>H22_HIPL</td>
<td>0.907477</td>
<td>-0.0000158</td>
<td>H22 -&gt; HIPL</td>
</tr>
<tr>
<td>H23_HIPL</td>
<td>0.890942</td>
<td>0.0000194</td>
<td>H23 -&gt; HIPL</td>
</tr>
<tr>
<td>H24_HIPL</td>
<td>0.670337</td>
<td>-0.0000501</td>
<td>H24 -&gt; HIPL</td>
</tr>
<tr>
<td>H25_HIPL</td>
<td>0.524875</td>
<td>-0.0000589</td>
<td>H25 -&gt; HIPL</td>
</tr>
<tr>
<td>H26_HIPL</td>
<td>0.644267</td>
<td>-0.0000242</td>
<td>H26 -&gt; HIPL</td>
</tr>
</tbody>
</table>
Value of Objective Function = 0.0007488987

PROC PRINT Report of the Average Squared Error for the Scored Test Data Set

Hill & Plateau Data
MLP with 30 Hidden Units
Fit Statistics for the Test Data

Test:
Average
Squared
Error.

.00071717

GCONTOUR Plot of the Predicted Values

Hill & Plateau Data
MLP with 30 Hidden Units
Predicted Values
G3D Plot of the Predicted Values

Hill & Plateau Data
MLP with 30 Hidden Units
Predicted Values

Copyright 2000 by SAS Institute Inc., Cary, NC, USA. All rights reserved.
The %LET statement sets the macro variable HIDDEN to 3.

```
title 'Hill & Plateau Data';
%let hidden=3;
```
The MAXITER = option specifies the maximum number of iterations.

```
hidden &hidden / id=h;
prefim 10;
train maxiter=1000 outest=mlpest;
score data=sampsio.dmtsurf out=mlpout outfit=mlpfit;
title2 "MLP with &hidden Hidden Units";
run;
```
PROC PRINT creates a report of selected fit statistics.

proc print data=mlpfit noobs label;
  var _tase_ _tasel_ _taseu_;
  where _name_ = 'HIPL';
  title3 'Fit Statistics for the Test Data';
run;
PROC GCONTOUR creates a plot of the predicted values.

```plaintext
proc gcontour data=mlpout;
   plot x2*x1=p_hipl / pattern ctext=black coutline=gray;
   pattern v=msolid;
   legend frame;
   title3 'Predicted Values';
   footnote;
run;
```
PROC G3D creates a plot of the predicted values. Note that this network underfits badly.

```sas
proc g3d data=mlpout;
  plot x2*x1=p_hipl / grid side ctop=blue
       caxis=green ctext=black
       zmin=-1.5 zmax=1.5;
run;
```
References


