

DTREG

Predictive Modeling Software

www.dtreg.com



Creating a Neural Network

Phil Sherrod
phil@philsherrod.com

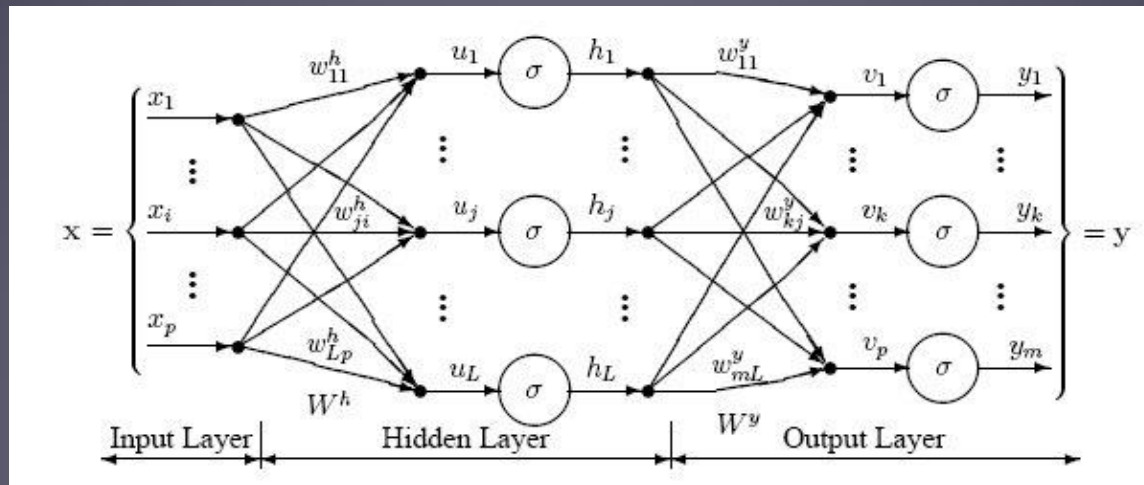
Copyright © 2014, all rights reserved
This material may not be reproduced without permission

Introduction to Neural Networks

- Neural networks are compact models that are widely used for many applications.
- DTREG supports several types of neural networks: perceptrons, probabilistic, general regression, radial basis function and cascade. This tutorial will focus on creating a traditional, perceptron network.
- Perceptron neural networks have an input layer, an output layer and one or two hidden layers. Most commonly, one hidden layer is used.
- DTREG can determine automatically the optimum number of neurons in the hidden layer.

Perceptron Neural Network Structure

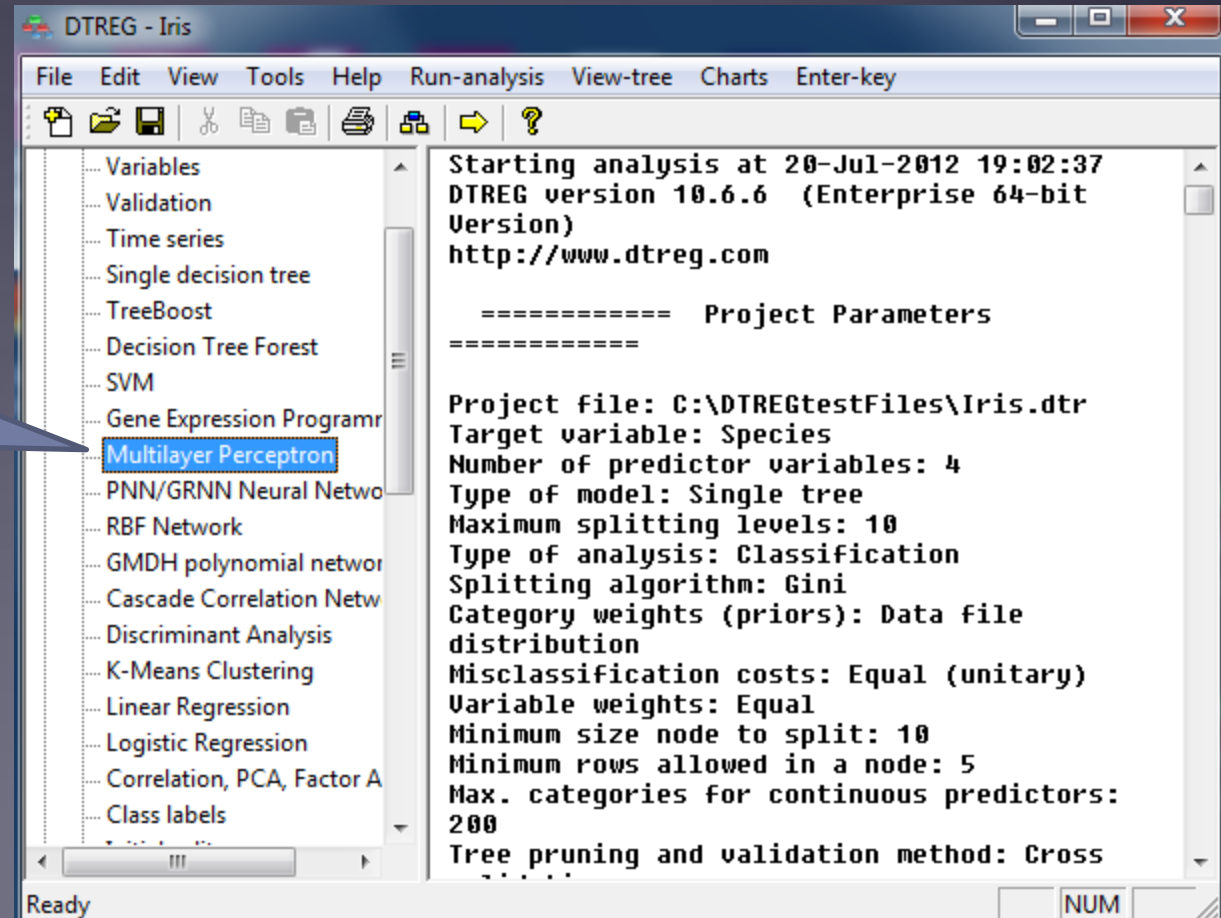
- This diagram shows an input layer on the left, a single hidden layer in the middle and an output layer on the right. The number of neurons in the hidden layer is adjustable depending on the problem.



Outline of Steps to Train a Neural Network

- Create a new project, specify the input training file, and select the target and predictor variables. See the Getting-Started tutorial for information about doing this.
- Select “Multilayer perceptron” as the type of model.
- Set training parameters or use the default.
- Train the model.
- Examine the report with model statistics

Select Multilayer Perceptron in Left Panel



Click this to open the multilayer perceptron parameter screen

Change Training Parameters if Desired

Select Multilayer Perceptron

Number of hidden layers

Do cross-validation of model

Let DTREG automatically find optimum number of neurons

Type of transfer functions

Prevent over-fitting

---- Multilayer Perceptron Neural Networks (MLP) ----

Type of model to build
Multilayer Perceptron

Number of network layers
 3 layers (1 hidden)
 4 layers (2 hidden)

Automatic hidden layer neuron selection
 Automatically optimize hidden layer 1
Min: 2 Max: 20 Step: 1
Max. steps without change: 8
% rows to use for search: 100
 Cross validate; folds: 4
 Hold-out sample %: 20
 Use training data

Model testing and validation
 No validation, use all data rows
 Use variable to select validation rows
 Random percent: 20
 Vfold cross-validation: 10
 Leave-one-out validation

Conjugate gradient parameters
Num. convergence tries: 4
Maximum iterations: 10000
Iterations without improvement: 100
Convergence tolerance: 1.000e-005
Min. improvement delta: 1.000e-006
Min. gradient: 1.000e-006
Max. minutes execution time: 0

How to handle missing values
 Don't use rows with missing values
 Replace missing values with medians
 Use surrogate variables

Options
 Compute importance of variables

Hidden layer activation function
Logistic

Output layer activation function
Logistic

Training method
 Scaled conjugate gradient (recommended)
 Traditional conjugate gradient
 Write progress report to project log

Layer 1: 6 Layer 2: 4

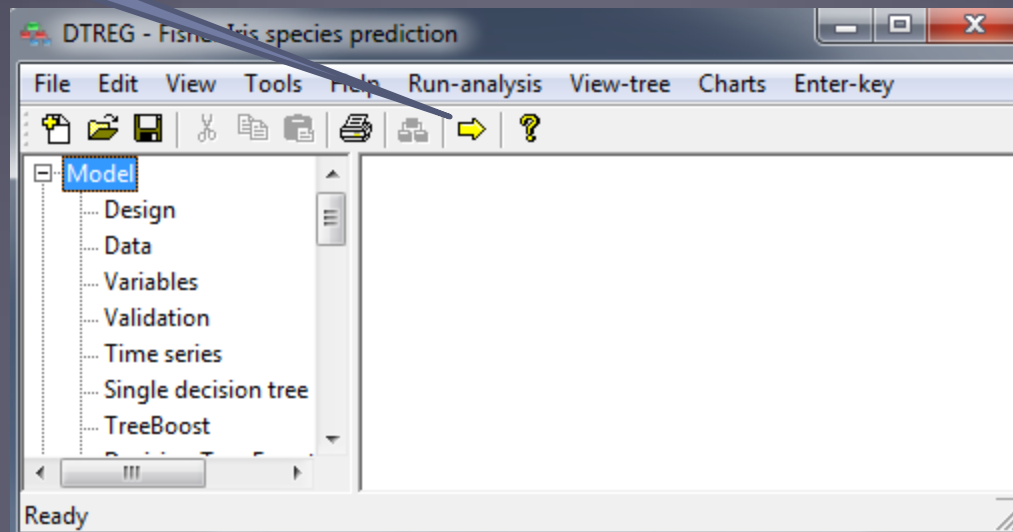
Overfitting detection & prevention
 Use test data to detect overfitting
% training rows to hold out: 20
Max. steps without change: 10

Write neuron weights to file
 Write neuron weights to file
Browse

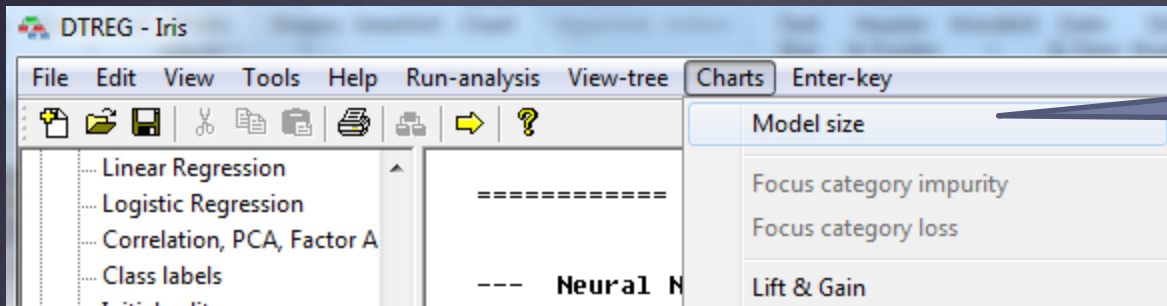
Tell DTREG to Train the Model

- Click the  icon to start training the model.

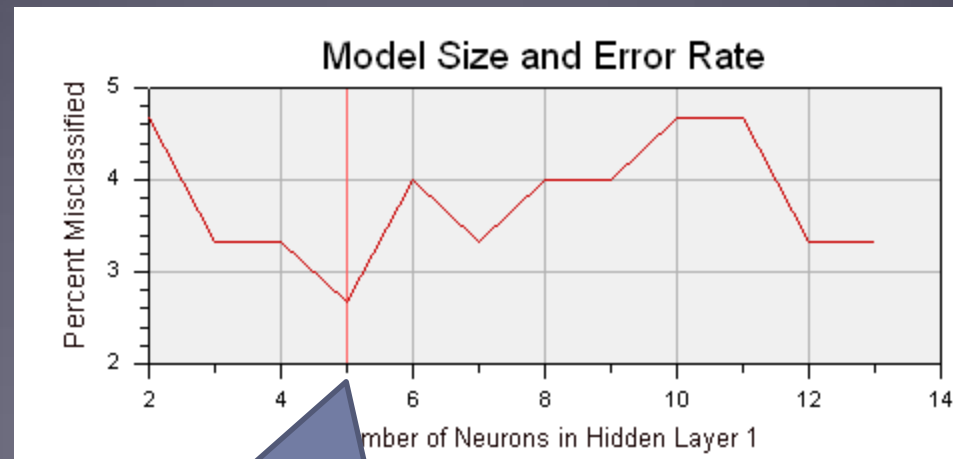
Click to start
training



View Chart of Errors vs. Number Neurons



Click Charts/Model size



Minimum error with 5 neurons in hidden layer

View Summary of Neural Network

DTREG determined that 5 neurons were optimum for the hidden layer

The screenshot shows the DTREG - Iris software interface. The main window displays the 'Neural Network Parameters' section, which includes a table of the neural network architecture. The table has five columns: Layer, Neurons, Activation, Min. Weight, and Max. Weight. The rows are: Input (4 neurons, Passthru activation), Hidden 1 (5 neurons, Logistic activation), and Output (3 neurons, Logistic activation). A callout bubble from the text above points to the 'Hidden 1' row. The software interface also shows a menu bar (File, Edit, View, Tools, etc.) and a tree view on the left side.

Layer	Neurons	Activation	Min. Weight	Max. Weight
Input	4	Passthru		
Hidden 1	5	Logistic	-4.563e+000	6.525e+000
Output	3	Logistic	-5.713e+000	4.407e+000

View Misclassification Summary Table

DTREG - Iris

File Edit View Tools Help Run-analysis View-tree Charts Enter-key

Class labels
Initial split
Category weights
Misclassification cost
Missing data
Variable weights
DTL
Score data
Translate source code
Miscellaneous

Generated tree
Analysis report
Project parameters
Input data
Variables
Categories
Neural network param
Model size
Misclassification
Confusion matrix

=====
Misclassification Tables
=====
--- Training Data ---

Category	Actual		Misclassified			
	Count	Weight	Count	Weight	Percent	Cost
Setosa	50	50	0	0	0.000	0.000
Versicolor	50	50	1	1	2.000	0.020
Virginica	50	50	1	1	2.000	0.020
Total	150	150	2	2	1.333	0.013

Overall accuracy = 98.67%

--- Validation Data ---

Category	Actual		Misclassified			
	Count	Weight	Count	Weight	Percent	Cost
Setosa	50	50	0	0	0.000	0.000
Versicolor	50	50	3	3	6.000	0.060
Virginica	50	50	1	1	2.000	0.020
Total	150	150	4	4	2.667	0.027

Overall accuracy = 97.33%

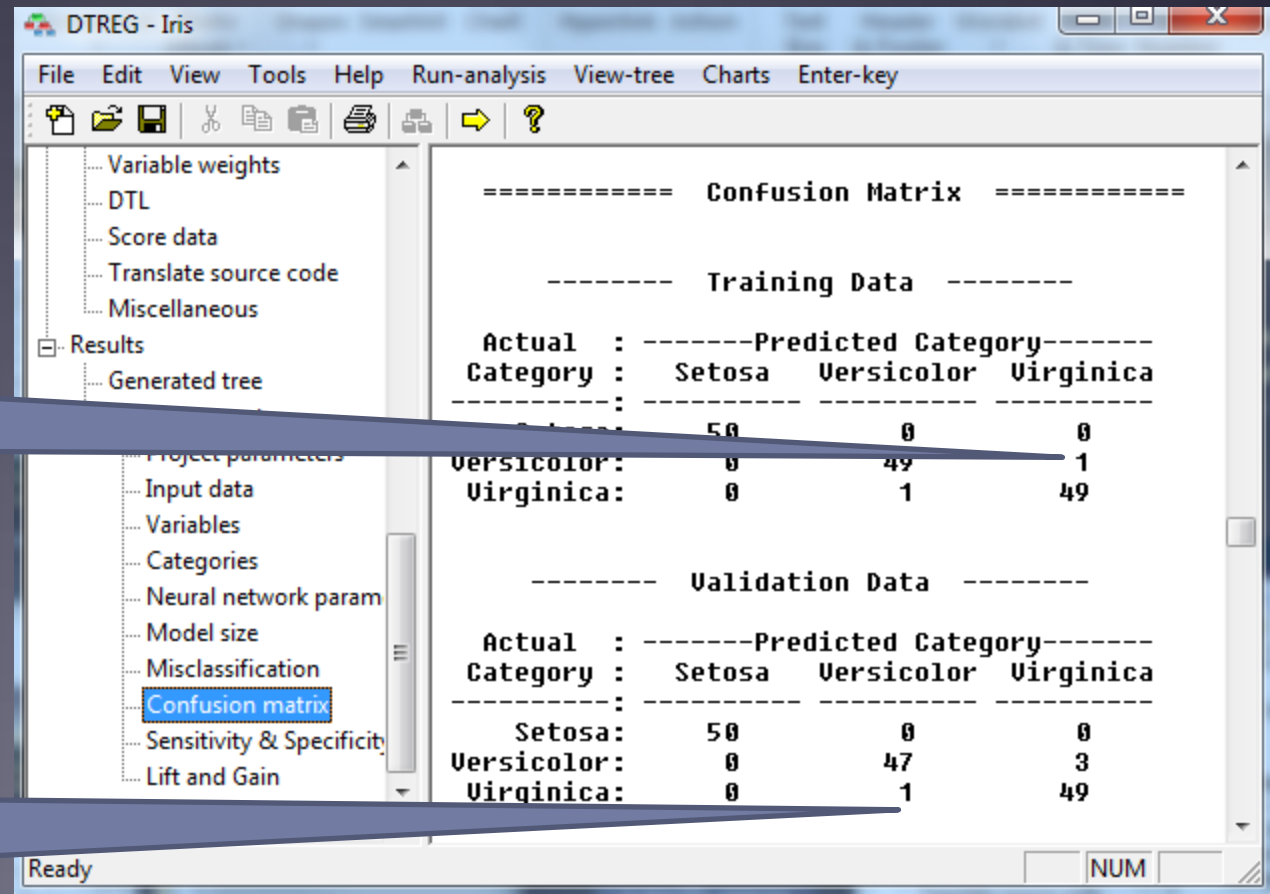
Ready NUM

1.333% error
on training
data

Click to select
mis-
classification
table

2.667%
error on
validation

Viewing the Confusion Matrix



Training data: one
Versicolor was
misclassified as
Virginica

Validation data:
one Virginica was
misclassified as
Versicolor

End of Neural Network Tutorial

- This completes the DTREG neural network training tutorial