# Neural Computing

## A Technology Handbook for Professional II/PLUS and NeuralWorks Explorer





NEURALWARE

需要全本请访问 www.ertongbook.com

# Neural Computing

## A Technology Handbook for Professional II/PLUS and NeuralWorks Explorer

江苏工业学院图书馆
藏书章



NeuralWare

NeuralWare welcomes your comments on this manual; please write to us or call at:

**NeuralWare, Inc.**
Technical Publications Group
202 Park West Drive
Pittsburgh, PA 15275
Phone: (412) 787-8222
FAX: (412) 787-8220
AppleLink: D6207

Macintosh and MPW are registered trademarks of Apple Computer, Inc. Think "C" is a registered trademark of Symantec Corporation.

NeuralWorks Professional II, NeuralWorks Explorer, Designer Pack, InstaNet, InstaProbe, and NeuralProbe are trademarks of NeuralWare, Inc.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form by any means, electronic, electrostatic, magnetic tape, mechanical, photocopying, recording, or otherwise, without permission in writing from the publisher.

The authors and publisher have used their best efforts in preparing this book and software. These efforts include the development, research, and testing of the programs to determine their effectiveness. The authors and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The authors and publisher shall not be liable in any way, even for incidental or consequential damages, in connection with, or arising out of the furnishing, performance, or use of these programs.

Printed in the United States of America

© NeuralWare, Inc., 1995

# Table of Contents


**Neural Computing . . . . . NC-1**

What Is Neural Computing? . . . . . NC-1
Analogy to the Brain . . . . . NC-1
Neural Networks . . . . . NC-2
Network Operation . . . . . NC-3
Layer Operations . . . . . NC-6
What Makes Neural Computing Different? . . . . . NC-6
Learning by Example . . . . . NC-6
Distributed Associative Memory . . . . . NC-7
Fault Tolerance . . . . . NC-8
Pattern Recognition . . . . . NC-8
Synthesis . . . . . NC-8
What Can Neural Computing Be Used For? . . . . . NC-9
Successful Applications . . . . . NC-9
Detection Of Explosives In Checked Airline Baggage . . NC-9
Identification Of Cloud Type Using Landsat Data . . . NC-9
Determination Of Satellite Orientation By Star Identification . . . . . NC-9
A Truck "Backer-Upper" . . . . . NC-10
Signal Processing . . . . . NC-10
A Jet Engine Diagnostic System . . . . . NC-10
Voice Recognition . . . . . NC-10
Sonar Interpretation . . . . . NC-11
Machine Vision . . . . . NC-11
Converting English Text to Phonemes . . . . . NC-11
Application Example . . . . . NC-11
Other Applications . . . . . NC-12
Future Applications . . . . . NC-12
New Technologies for Neural Computing Systems . . . . . NC-13
Software Simulators . . . . . NC-13
Hardware Accelerators . . . . . NC-14
Mark III and Mark IV . . . . . NC-14
Network Emulation Processor (NEP) . . . . . NC-14
Balboa . . . . . NC-15
Delta-II Floating Point Processor . . . . . NC-15
Silicon Chips . . . . . NC-15
Optical Processors . . . . . NC-16

## History of Neural Computing ..... NC-17

Dartmouth Summer Research Project on Artificial Intelligence . NC-17
Rosenblatt's Perceptron ..... NC-18
Widrow's Adaptive Linear Element ..... NC-19
Minsky and Papert's Perceptrons ..... NC-19
Anderson's Linear Associator ..... NC-20
Kohonen's Work ..... NC-20
Grossberg's Neurologically Based Models ..... NC-22
Hopfield and the Return to Respectability ..... NC-23
Current Research ..... NC-23
Periodicals ..... NC-24
Journals ..... NC-24
Newsletters ..... NC-25

## Adaline and Madaline ..... NC-27

The Adaline Paradigm ..... NC-28
The Adaline Processing Element ..... NC-28
Adaline Learning Algorithm ..... NC-29
Madaline (Multiple Adalines) ..... NC-31
Madaline Architecture ..... NC-31
Madaline Learning ..... NC-32
Coding of Adaline Inputs ..... NC-32
Creating an Adaline or Madaline Network ..... NC-33
Weather Forecasting with an Adaline ..... NC-34
Selected Bibliography on Adaline and Madaline ..... NC-36

## Adaptive Resonance Theory Networks (ART) . . NC-37

ART and Pattern Recognition ..... NC-38
ART 1 ..... NC-38
Operation of the ART Network ..... NC-39
Learning ..... NC-42
Confirming the Top-Down Hypothesis ..... NC-43
Gain Control ..... NC-43
The Novelty Detector and Category Size ..... NC-45
Vigilance Parameter ..... NC-47
Strengths, Limitations, and Applicability of ART ..... NC-47
Art Network Equations ..... NC-51
Creating an ART 1 Network with InstaNet ..... NC-53
ART 1 Example ..... NC-54
Loading the ART 1 Example ..... NC-54
Changing the Vigilance Factor ..... NC-56
Creating your Own Pattern File ..... NC-56
Creating the Data File ..... NC-57
Creating a Pattern ..... NC-57
Glossary for ART 1 ..... NC-57
Selected Bibliography on ART ..... NC-61

需要全本请访问 www.ertongbook.com

**Back-Propagation ..... NC-63**

The Back-Propagation Network ..... NC-63
Back-Propagating the Local Error ..... NC-65
Minimizing the Global Error ..... NC-66
The Global Error Function ..... NC-67
Summary of the Standard Back-Propagation Algorithm .. NC-67
Variations on the Standard Algorithm ..... NC-68
Momentum Term ..... NC-68
Cumulative Update of Weights ..... NC-68
Different Error Functions ..... NC-69
Softmax Activation Function ..... NC-70
Different Transfer Functions ..... NC-71
Saturation and F′ Offset ..... NC-73
Fast Back-Propagation ..... NC-73
Cascade Learning ..... NC-74
Back-propagated Value ..... NC-76
To do Cascaded Learning in NeuralWorks ..... NC-77
Applications of Back-Propagation ..... NC-78
Encoding and Data Compression ..... NC-79
Signal Processing ..... NC-79
Noise Filtering ..... NC-80
Stock Market Prediction ..... NC-81
NeuralWorks Stock Network ..... NC-81
Converting English Text to Phonemes ..... NC-82
Pattern Classification ..... NC-83
Creating a Back-Propagation Network ..... NC-84
Selected Bibliography on Back-Propagation ..... NC-84

**Bi-Directional Associative Memory (BAM) Networks ..... NC-87**

The BAM Paradigm ..... NC-87
How the BAM Learns Associations ..... NC-88
The Energy of an Association ..... NC-88
Retrieving Associations from a BAM ..... NC-89
Creating a BAM Network ..... NC-91
BAM Example: Associating Simple Patterns ..... NC-91
Selected BAM Bibliography ..... NC-93

**The Boltzmann Machine ..... NC-95**

Creating a Boltzmann Network Using InstaNet ..... NC-97
Input/Output ..... NC-99
Control Strategies ..... NC-100
Control Strategy For Boltzcmp (learn) ..... NC-100
Control Strategy For Boltzcmp (recall) ..... NC-102
Control Strategy For Boltzio (learn) ..... NC-102
Control Strategy For Boltzio (recall) ..... NC-102

Boltzmann Layer Learning and Recall Schedule ..... NC-103
Boltzmann Layer Neuro-Dynamics ..... NC-104
Transfer Function ..... NC-104
Learning Rule ..... NC-104
Example Network ..... NC-105
Selected Bibliography on Boltzmann Machines ..... NC-106

**Brain-State-in-a-Box ..... NC-109**

Linear Associator ..... NC-109
Hebbian Learning ..... NC-110
Widrow-Hoff Learning ..... NC-111
Brain-State-in-a-Box ..... NC-111
The BSB Algorithm ..... NC-112
Convergence Characteristics ..... NC-113
Applications of Brain-State-in-a-Box ..... NC-113
Creating a BSB Network ..... NC-114
Input to the Network ..... NC-115
Analyzing a Simple Database ..... NC-115
Training Patterns for Logical Inference ..... NC-116
Invoking the NeuralWorks "Gods and Men" Example . NC-117
Retraining This Example ..... NC-118
A More Complicated Database ..... NC-119
Invoking the NeuralWorks "Medicine Box" Example . NC-122
Performing a Recall ..... NC-122
Training the Network ..... NC-123
Training the Network Using Your Own Input ..... NC-123
Selected Bibliography on Brain-State-in-a-Box ..... NC-124

**Delta-Bar-Delta (DBD) Networks ..... NC-125**

Back-Propagation Algorithm ..... NC-125
DBD Algorithm ..... NC-126
EDBD Algorithm ..... NC-126
DBD Network Architecture ..... NC-127
DBD Equations ..... NC-129
EDBD Network Architecture ..... NC-132
Creating a DBD or EDBD Network with InstaNet ..... NC-135
Selected Bibliography on DBD and EDBD Networks ..... NC-135

**Digital Neural Network Architecture (DNNA) Networks ..... NC-137**

DNNA Architecture ..... NC-138
DNNA Equations ..... NC-140
Applications of DNNA ..... NC-142
Learning Rules ..... NC-142

Creating DNNA Networks ..... NC-143
Descretization of the Weights ..... NC-143
Checking the Weight Discretization ..... NC-144
Selected Bibliography on DNNA ..... NC-144

**Directed Random Search (DRS) Networks NC-145**

An Overview of DRS ..... NC-146
DRS Components ..... NC-148
Random Step ..... NC-149
Reversal Step ..... NC-149
Directed Component ..... NC-150
Self-Adjusting Variance ..... NC-151
DRS Parameters ..... NC-154
Key Parameters ..... NC-154
Creating a DRS Network with InstaNet ..... NC-154
Selected Bibliography on DRS Networks ..... NC-155

**Fuzzy ARTMAP Classification Network... NC-157**

ART 1 Revisited ..... NC-157
Fuzzy ART ..... NC-160
Complement Coding ..... NC-162
Summary ..... NC-166
Fuzzy ARTMAP Classifier ..... NC-166
Fuzzy ARTMAP Example ..... NC-167
Create a Fuzzy ARTMAP Network ..... NC-169
Selected Bibliography on Fuzzy ARTMAP ..... NC-169

**General Regression Neural Network..... NC-171**

GRNN Equations ..... NC-172
Clustering the Training Samples ..... NC-175
Clustering Procedure ..... NC-175
Feedforward Architecture ..... NC-176
Alternative Pattern Units ..... NC-177
Nonstationary Data ..... NC-179
Create a General Regression Neural Network ..... NC-179
General Regression Neural Network Example ..... NC-180
Selected Bibliography on GRRN ..... NC-180

**Hamming Networks ..... NC-181**

Network Architecture and Equations ..... NC-182
Creating a Hamming Network ..... NC-185
Hamming Network Implementation ..... NC-185
Control Strategy ..... NC-186
Learning ..... NC-187
Learning and Recall Schedule ..... NC-187

Example Hamming Network ..... NC-188
Training the Network ..... NC-188
Performing a Recall ..... NC-189
Selected Bibliography on Hamming Networks ..... NC-190

**Hopfield Networks ..... NC-191**

The Hopfield Paradigm ..... NC-191
Hopfield Learning ..... NC-193
Learning Rule ..... NC-193
Unlearning ..... NC-194
Recall ..... NC-195
Memory Capacity ..... NC-195
Energy Functions ..... NC-195
Implementing Hopfield Networks ..... NC-197
Creating a Hopfield Network ..... NC-197
The Traveling Salesperson Problem ..... NC-198
Description of the Problem ..... NC-198
A TSP Neural Network ..... NC-199
Knowledge Representation ..... NC-199
Energy Functions ..... NC-200
Free Parameters ..... NC-202
Experimental Results ..... NC-204
Newer Research ..... NC-204
Using the NeuralWorks TSP Network Generation Program . NC-206
Selected Bibliography on Hopfield Networks ..... NC-207

**The Logicon Projection Network™ ..... NC-209**

Theory ..... NC-210
The Projection ..... NC-210
Hyperspherical and Hyperplanar Decision Boundaries ... NC-212
Initialization of Weights and Thresholds ..... NC-214
First layer of Weights and Thresholds ..... NC-214
Optional Methods to Determine the Initial Settings of the Prototypes ..... NC-215
Higher Layer Weights and Output Thresholds ..... NC-215
Setting the Network Parameters ..... NC-217
Setting the Learning Rate and Momentum ..... NC-217
Setting the Projection Parameters ..... NC-217
Sample Problems ..... NC-218
The Circle and Rectangle Classification Problem ..... NC-218
Function Fitting ..... NC-220
The Double Spiral ..... NC-221
Create a Logicon Projection Network™ ..... NC-222
To Run the lpn_demo Example Network ..... NC-222
To Run the lpn_spir Example Network ..... NC-224
Using this UserIO Program with Your Own Network .. NC-225
Selected Bibliography on LPN ..... NC-226

**Learning Vector Quantization (LVQ) Networks NC-227**

LVQ Network Architecture ..... NC-227
Variants of LVQ: LVQ1 ..... NC-228
Variants of LVQ: LVQ2 ..... NC-229
Variants of LVQ: Extended LVQ ..... NC-230
Variants of LVQ: LVQ without Repulsion ..... NC-230
LVQ Equations ..... NC-230
Learning and Recall Schedule ..... NC-232
Creating an LVQ Network ..... NC-233
Selected Bibliography on LVQ Networks ..... NC-233

**Modular Neural Network ..... NC-235**

Equations ..... NC-237
Feedforward Equations ..... NC-238
Learning Equations ..... NC-238
Probabilistic Interpretation ..... NC-240
Creating an MNN in NeuralWorks ..... NC-241
To Run the modnn Demo Network ..... NC-241
Selected Bibliography on MNN ..... NC-242

**Perceptron Networks ..... NC-243**

The Perceptron Paradigm ..... NC-245
The Perceptron Network ..... NC-245
The Perceptron Processing Element ..... NC-247
Perceptron Learning ..... NC-247
The Perceptron Convergence Theorem ..... NC-248
Linear Separability and the Exclusive OR Problem ..... NC-248
Perceptron Applications ..... NC-250
Creating a Perceptron Network ..... NC-251
Perceptron Exclusive OR ..... NC-251
Selected Bibliography on Perceptrons ..... NC-252

**Probabilistic Neural Network ..... NC-253**

Creating a Probabilistic Neural Network ..... NC-257
Input and Output ..... NC-258
Control Strategy ..... NC-258
Pattern Layer Neurodynamics ..... NC-259
PNN Transfer Function ..... NC-259
PNN Output function ..... NC-260
User Kohonen Learn Function ..... NC-260
Summation Layer Neurodynamics ..... NC-261
Applying PNN ..... NC-261
Classification of Iris Species Using PNN ..... NC-262
Selected Bibliography ..... NC-263

## Radial Basis Function Networks ..... NC-265

Comparison with Back-Propagation ..... NC-267
Training an MD-RBFN Network ..... NC-270
Alternative Pattern Units ..... NC-272
Counter-Propagation Network ..... NC-274
Creating an RBFN in NeuralWorks ..... NC-274
An RBFN Example ..... NC-274
Selected Bibliography on RBFN ..... NC-275

## Recirculation Networks ..... NC-277

Recirculation Equations ..... NC-278
Convergence Properties ..... NC-280
Extensions to Recirculation Architecture ..... NC-281
Applications of Recirculation ..... NC-283
Natural Language Processing ..... NC-283
Data Compression ..... NC-285
Experimenting with the Example Network ..... NC-286
Low Pass Filtering ..... NC-286
Creating a Recirculation Network with InstaNet ..... NC-288
Implementing Recirculation Networks ..... NC-288
Architecture ..... NC-288
Transfer Function ..... NC-289
Learning Rule ..... NC-289
Learning and Recall Schedule ..... NC-290
Control Strategy ..... NC-290
Example Networks ..... NC-291
Selected Bibliography on Recirculation ..... NC-292

## Self-Organizing Maps (SOM) ..... NC-293

Self-Organization ..... NC-294
The Self-Organizing Map (SOM) Network Architecture ..... NC-296
Finding the winning PE ..... NC-298
Adjusting the Weights of Neighboring PEs ..... NC-300
Estimating the Winning Frequency of PEs ..... NC-301
Interpolation of Kohonen PEs ..... NC-302
Applications of SOM ..... NC-304
SOM Demonstration Network ..... NC-304
Creating SOM Networks ..... NC-305
Selected Bibliography on SOM ..... NC-305


需要全本请访问 www.ertongbook.com

**Spatio-Temporal Pattern Recognition .... NC-307**

SPR Network Architecture ..... NC-307
SPR Network Applications ..... NC-311
Creating an SPR Network ..... NC-311
Selected Bibliography on Spatio-Temporal Pattern Recognition . NC-312

**Neural Network Terms..... NC-313**

**Index ..... NC-319**

# Neural Computing

Welcome to the world of neural computing!

This introduction prepares you to work on actual neural models by introducing you to nerual computing. It describes what neural computing is, where it came from, what it can be used for, and what new technologies have been developed to facilitate neural computing. The next chapter provides a brief history of neural computing and the remainder of this volumne describes neural computing paradigms (models) in detail.

A glossary of neural network terms discussed throughout this volume follows those chapters.

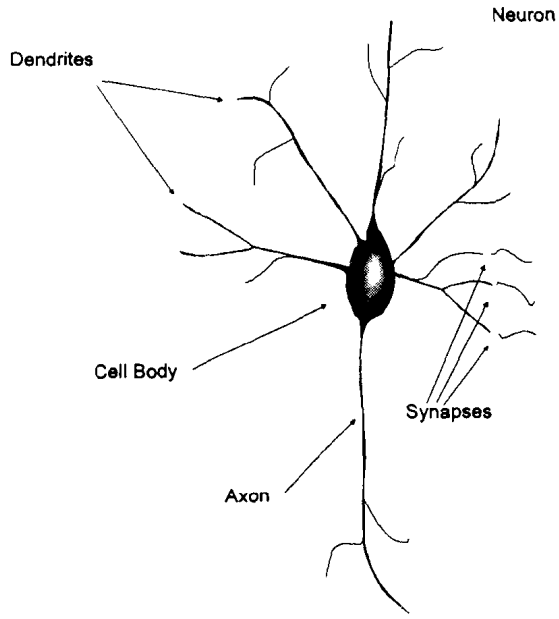
## What Is Neural Computing?

The human brain is the most complex computing device known to man. The brain's powerful thinking, remembering, and problem-solving capabilities have inspired many scientists to attempt computer modeling of its operation. One group of researchers has sought to create a computer model that matches the functionality of the brain in a very fundamental manner; the result has been neural computing.

### Analogy to the Brain

The neuron is the fundamental cellular unit of the nervous system and, in particular, the brain. Each neuron is a simple microprocessing unit which receives and combines signals from many other neurons through input processes (structures) called dendrites. If the combined signal is strong enough, it activates the firing of the neuron, which produces an output signal; the path of the output signal is along a component of a cell called the axon. This simple transfer of information is chemical in nature, but it has electrical side effects which we can measure.

The brain consists of tens of billions of neurons densely interconnected. The axon (output path) of a neuron splits up and connects to dendrites (input paths) of other neurons through a junction referred to as a synapse. The transmission across this junction is chemical in nature and the amount of signal transferred depends on the amount of chemicals (neurotransmitters) released by the axon and received by the dentrites. This synaptic efficiency (or "strength") is what is modified when the brain learns. The synapse combined with




*The basic structure of a biological neuron.*

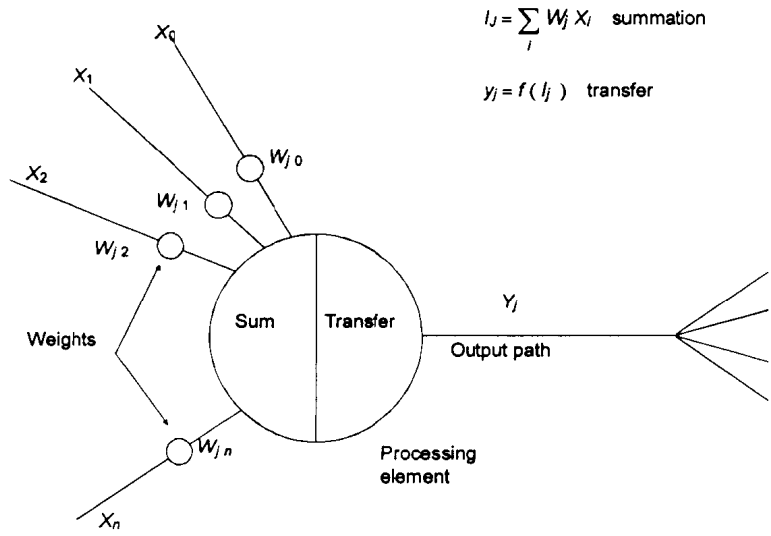
the processing of information in the neuron form the basic memory mechanism of the brain.

## Neural Networks

In an artificial neural network, the unit analogous to the biological neuron is referred to as a "processing element." A processing element (PE) has many input paths (dendrites) and combines, usually by a simple summation, the values of these input paths. The result is an internal activity level for the processing element. The combined input is then modified by a transfer function. This transfer function can be a threshold function which only passes information if the combined activity level reaches a certain level, or it can be a continuous function of the combined input. The output value of the transfer function is generally passed directly to the output path of the processing element (see descriptions of competition and asynchronous operation later in this introduction for exceptions to this).

The output path of a processing element can be connected to input paths of other processing elements through connection weights which correspond to the synaptic strength of neural connections. Since each connection has a corresponding weight, the signals on the input lines to a processing element are modified by these weights prior to being summed. Thus, the summation function is a weighted sum-

mation. In itself, this simplified model of a neuron is not very interesting; the interesting effects result from the ways neurons are interconnected.

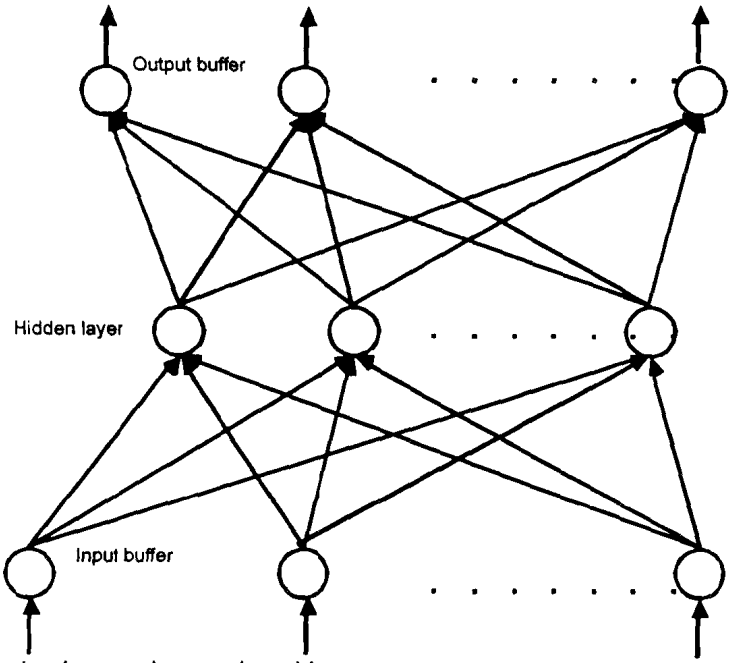



A neural network consists of many processing elements joined together in the above manner. Processing elements are usually organized into groups called layers or slabs. A typical network consists of a sequence of layers or slabs with full or random connections between successive layers. There are typically two layers with connections to the outside world: An input buffer where data is presented to the network, and an output buffer which holds the response of the network to a given input. Layers distinct from the input and output buffers are called hidden layers.

## Network Operation

There are two main phases in the operation of a network — Learning and Recall. In most networks these are distinct.

Learning is the process of adapting or modifying the connection weights in response to stimuli being presented at the input buffer and optionally the output buffer. A stimulus presented at the output buffer corresponds to a desired response to a given input; this desired response must be




*A simple neural network architecture.*

provided by a knowledgeable teacher. In such a case the learning is referred to as "supervised learning."

If the desired output is different from the input, the trained network is referred to as a hetero-associative network. If, for all training examples, the desired output vector is equal to the input vector, the trained network is called auto-associative. If no desired output is shown, the learning is called unsupervised learning.

A third kind of learning, falling between supervised and unsupervised learning, is reinforcement learning where an external teacher indicates only whether the response to an input is good or bad. In some instances, the network may only be graded after several inputs have been processed by the network.

Whatever kind of learning is used, an essential characteristic of any network is its learning rule. The learning rule specifies how weights adapt in response to a learning example. Learning may require showing a network many examples, many thousands of times, or only once. The parameters governing a learning rule may change over time as the network progresses in its learning. The long-term control of the learning parameters is referred to as a learning schedule.

Recall refers to how the network processes a stimulus presented at its input buffer and creates a response at the output buffer. Often a recall is an integral part of the learning process such as when a desired response of the network must be compared to the actual output of the network to create an error signal.

The simplest form of a network has no feedback connections from one layer to another or to itself. Such a network is called a "feedforward network." In this case, information is passed from the input buffer, through intermediate layers to the output layer, in a straightforward manner, using the summation and transfer function characteristics of the particular network. Feedforward networks are interesting because of non-linearities in the transformations. In some feedforward networks, a certain amount of feedback is used to create time-sensitivity in the network. These are called recurrent networks or feedback networks.

If there are feedback connections, information will reverberate around the network, across layers or within layers, until some convergence criterion is met. The information is then passed to the output buffer. NeuralWorks uses a control strategy to govern the sequence in which layers are processed and to govern how information is passed through the network.

Many types of networks have an energy function associated with them. Each state of the network, as defined by a particular set of processing element outputs, contribute to the energy value. The recall process iteratively modifies the processing element states so energy decreases, thus yielding a state representing a local minimum in the energy surface. For networks with feedback, a recall schedule or annealing schedule can be used to control the recall parameters as the reverberation process progresses.

Another consideration in the performance of a network is whether it operates in synchronous or asynchronous mode. Synchronous behavior means that all or select groups of processing elements in the system fire (that is, release their output values) simultaneously at each time step. Asynchronous behavior means that processing elements fire randomly and independently of any other unit in the network. In one system (the P3 system, see "Software Simulators" later in this chapter), asynchronous behavior is modified by means of an additional input line to each processing element that controls whether or not it will be updated.



需要全本请访问 www.ertongbook.com