


---

# Neural Network Experiments on Personal Computers and Workstations

---

Granino A. Korn

A Bradford Book  
The MIT Press  
Cambridge, Massachusetts  
London, England



---

# Contents

---

<b>Preface</b>	xiii
<b>1 Introduction to Neural Networks and Simulation</b>	<b>1</b>
OVERVIEW	1
NEURAL NETWORKS AND NEURON MODELS	1
1.1 Introduction	1
1.2 Biological Neurons: Simplified Anatomy	3
1.3 Biological Neurons: Data Transmission	3
1.4 A Static Neuron Model	4
1.5 Dynamic State-equation Models	6
1.6 Dynamic Neuron Models	7
NEURAL NETWORKS: DECISION MAKING AND ABSTRACTION	10
1.7 Decisions, Associative Memories, and Learning	10
1.8 Neurocomputer Applications	12
1.9 A Survey of Neural Networks: Combinations of Neural Networks	13
COMPUTER REPRESENTATION OF NEURAL NETWORKS	16
1.10 A Readable Matrix Language for Neural Networks	16
1.11 Difference Equations vs. Differential Equations	17
1.12 Floating-point, Fixed-point, Binary, and Bipolar Data Representations	18
TRUE INTERACTIVE SIMULATION REQUIRES DIRECT PROGRAM EXECUTION	18
1.13 Interactive Simulation	18
1.14 Direct-executing Simulation	19

1.15	DESIRE Interactive Simulation Systems	19
1.16	Performance: Interconnects per Second	20
	THE USER-FRIENDLY SIMULATION ENVIRONMENT	21
1.17	Hiding Advanced Features: User-written Help Screens, Menus, and Notebook Files	21
1.18	Access to Operating-System Commands	22
1.19	The Limits of DESIRE: Workstation Programs	23
<b>2</b>	<b>A Simple Language for Neural Network Models</b>	<b>25</b>
	OVERVIEW	25
	NEURON-LAYER ARRAYS AND SIMULATION RUNS	25
2.1	Vector Representation of Neuron-layer Activations, Signal Patterns, and Connection Weights	25
2.2	Interpreted Experiment-Protocol Programs and Compiled Simulation Runs	26
2.3	Simulation Time and Other Simulation Parameters	27
2.4	Resetting Initial Conditions for Repeated Simulation Runs	27
	VECTOR ASSIGNMENTS IN NEURON-LAYER MODELS	29
2.5	<b>VECTOR</b> Assignments in <b>DYNAMIC</b> Program Segments	29
2.6	<b>VECTOR</b> Expressions: Sums of Product Terms	30
2.7	Matrix-Vector Products in <b>VECTOR</b> Assignments	32
2.8	Pattern Arrays Supply Inputs for Successive Trials	32
2.9	Output-limited <b>VECTOR</b> Expressions	34
2.10	Neuron Activation Functions	34
	SIMPLE DIFFERENCE EQUATIONS SPECIFY NEURON DYNAMICS	35
2.11	<b>UPDATE</b> Accumulates a Vector State Variable	35
2.12	Other Vector Difference Equations	36
	OPERATIONS ON CONNECTION-WEIGHT MATRICES	36
2.13	<b>MATRIX</b> Assignments	36
2.14	Training Connection-weight Matrices: <b>LEARN</b> Statements	38
2.15	Other Matrix Difference Equations and Training Operations	38
2.16	Masking <b>MATRIX</b> and <b>LEARN</b> Computations	38
	DOT PRODUCTS, VECTOR NORMS, ERROR MEASURES, AND SUMS	39
2.17	<b>DOT</b> Products	39
2.18	Euclidean Norms and Error Measures	39
2.19	Computation of Sums: Taxicab Norms	40
2.20	Hamming Norm and Hamming Distance	40
2.21	Vector Normalization	41

VECTOR COMPILER AND MODEL GENERALITY	42
2.22 Use of Parentheses vs. Compiler Speed	42
2.23 Programming General <b>VECTOR</b> and <b>MATRIX</b> Expressions	42
2.24 Scalars in <b>VECTOR</b> and <b>MATRIX</b> Expressions	43
SCALAR EXPRESSIONS IN DYNAMIC PROGRAM SEGMENTS	43
2.25 Compiled Scalar Expressions and Functions	43
2.26 Recursive Scalar Assignments and Difference Equations	44
SIMULATION OUTPUT: DESIRE/NEUNET GRAPHICS AND LISTINGS	44
2.27 Time-History Output and Display Control	44
2.28 Interpreter Graphics	45
2.29 Pixel Images and <b>SHOW</b> Statements	47
2.30 CRT Screen Prints	48
2.31 "Drawing" and Saving Image Patterns for Experiments	48
<b>3 Feedforward Networks for Supervised Training</b>	51
OVERVIEW	51
IMPROVING ON PERCEPTRON LEARNING	51
3.1 Simple Perceptrons and the LMS Algorithm	51
3.2 Hidden Layers and Nonlinear Operations	55
SIMULATION OF BACKPROPAGATION NETWORKS	56
3.3 Backpropagation Networks	56
3.4 Hard Limiting vs. Sigmoid Limiting of Neuron Activations	59
3.5 Simulation of XOR Learning: the Experiment-Protocol Program	61
3.6 <b>DYNAMIC</b> Program Segment for XOR Learning	65
3.7 Parameter-editing Screens and User-written Menus	65
3.8 Results and Computing Speed	66
3.9 Simulation of 4-bit Parity Learning	68
3.10 Function-mapping Backpropagation Networks	68
3.11 Backpropagation as an Abstraction Mechanism	74
CREEPING RANDOM SEARCH AND HIGHER-ORDER CONNECTIONS	76
3.12 Creeping Random Search Optimization	76
3.13 "Functional Link" Networks: Polynomial and Orthogonal-function Expansions	79
3.14 A Self-learning Diode Function Generator	80

<b>4 Competitive Learning, Adaptive-Resonance Emulation, and Statistical Learning</b>	85
OVERVIEW	85
UNSUPERVISED PATTERN CLASSIFICATION	85
4.1 Template-matching Competition	85
4.2 Adaptive Learning of Successive Pattern Templates	87
4.3 Competition for Best Correlation	87
THE NEED FOR ADAPTIVE-RESONANCE-TYPE SCHEMES	89
4.4 Preservation vs. Adaptation of Learned Categories. Adaptive-resonance Theory	89
4.5 <b>CLEARN</b> : Simplified Computer Emulation of Adaptive Resonance	90
4.6 A Character-recognition Program	93
4.7 Computing Speed	96
SUPERVISED LEARNING: COUNTERPROPAGATION NETWORKS	97
4.8 Counterpropagation Networks	97
4.9 A Function-learning Mapping Network with Interpolation	98
RECALL WITH RANDOM INPUTS	103
4.10 Recall of Known Patterns in the Presence of Noise: Vector Quantization	103
4.11 Using Vector Quantization for Measuring Statistics of Multi-dimensional Distributions	104
STATISTICAL LEARNING	105
4.12 Adaptation to Statistical Environments: Vector Quantization and Entropy Test for Random Patterns	105
4.13 Vector Quantization as an Abstraction Mechanism	112
4.14 Random Sequences of Known Patterns	112
4.15 Unsupervised Learning of Patterns in Noise: Adaptive Detectors	114
<b>5 More Advanced Neural Network Modelling</b>	117
OVERVIEW	117
SPECIAL MODELLING TECHNIQUES	117
5.1 Sparse-Matrix Techniques	117
5.2 Declaration and Use of Subvectors	119
5.3 Histogram and Entropy Computation with <b>VECTOR</b> Functions	120
5.4 Run-termination and <b>if</b> Statements in <b>DYNAMIC</b> Segments	121
5.5 <b>DESIRE</b> Submodel Declaration, Invocation, and Nesting	122

THE VECTOR SHIFT OPERATOR AND SIMPLE ARRAY CONVOLUTIONS	123
5.6 The <b>VECTOR</b> Shift Operator	123
5.7 Shifting Time Histories into and out of Neuron Arrays	124
5.8 One-dimensional Array Convolution	125
5.9 Two-dimensional Patterns as Vectors: Simple Image Processing and Pattern Convolution	125
5.10 Simulation of a Two-dimensional Matched-filter Detector	128
CONTRAST ENHANCEMENT, PATTERN NORMALIZATION, AND BIOLOGICAL MODELS	128
5.11 Biological Models vs. Neurocomputers	128
5.12 Linear Contrast-enhancing Filters, Transformation Matrices, and Convolution	130
5.13 Contrast Enhancement (Competition) via Programmed Maximum Selection	132
5.14 Nonlinear (Shunt-multiplier) Feedforward Filters: Normalization and Pattern Matching	133
5.15 Nonlinear Lateral-inhibit Feedback and Grossberg Low-level-signal Quenching	136
5.16 More General Feedforward and Lateral Feedback Filters: Edge Enhancement	137
5.17 A Pulsed-neuron Model	140
5.18 Random time vs. Synchronous Neuron Activation	142
<b>6 Neural Networks with Feedback: Associative Memories, Stability, and Learning</b>	143
OVERVIEW	143
SOME CLASSICAL ASSOCIATIVE MEMORIES	143
6.1 Fully Connected Feedback Networks	143
6.2 Linear Hopfield-type Networks	145
6.3 Brain-State-in-a-Box and Bipolar Hopfield Networks	146
6.4 Connection-weight Training and Memory Capacity	146
6.5 Novelty Detectors	148
6.6 Simulated Annealing and Boltzmann Networks	148
6.7 Practical and Theoretical Importance of Hopfield-type Networks	149
6.8 Simulation of Hopfield-type Autoassociators	151
6.9 Simulation of Heteroassociator Networks	153
BIDIRECTIONAL ASSOCIATIVE MEMORIES	154
6.10 Bidirectional Associative Memories	154
6.11 A Database Application: McClelland's Jets-and-Sharks	157

NETWORK STABILITY AND PATTERN ASSOCIATOR CAPACITY	164
6.12 The Cohen-Grossberg Stability Theorem	164
6.13 More General BAM-connected Networks: BAM Stability	165
<b>7 More General Networks, and Combined Simulation of Neural Networks and Other Dynamic Systems</b>	<b>167</b>
OVERVIEW	167
FUNCTION-MAPPING NEURAL NETWORKS WITH RANDOM INPUTS AND NON-LEAST-SQUARES OPTIMIZATION	167
7.1 Function-mapping Networks with Random Inputs	167
7.2 Backpropagation Networks	168
7.3 Counterpropagation Networks	175
7.4 Multiple Inputs and Outputs	176
7.5 Non-least-squares Optimization of Neural Networks	177
SEQUENTIAL NEURAL NETWORKS AND MORE GENERAL FEEDBACK	180
7.6 Neural Networks as State Machines	180
7.7 Sequential Neural Networks	183
7.8 Simulation of Adaptive Transversal Filters and Predictors	183
7.9 Generation of Time-varying Patterns: Avalanche Circuit	186
7.10 More General Feedback Systems	186
MODELS INVOLVING DIFFERENTIAL EQUATIONS	188
7.11 Scalar Differential Equations	188
7.12 Gated Dipoles and Simulation of a Biological Clock	190
7.13 Vector Differential Equations and <b>STATE</b> Declarations	192
7.14 Submodels with Differential Equations	196
SIMULATIONS COMBINING NEURAL NETWORKS AND SEPARATE DIFFERENTIAL-EQUATION SYSTEMS	196
7.15 Models Involving both Differential Equations and Difference Equations: <b>OUT</b> , <b>SAMPLE</b> , and <b>step</b> Statements	196
7.16 A Simple Control-system Identification Problem	197
<b>Appendix A DESIRE/NEUNET Programs and Files</b>	<b>201</b>
HARDWARE REQUIREMENTS AND SOFTWARE INSTALLATION	201
A.1 Hardware Requirements	201
A.2 DESIRE/NEUNET Installation	201
A.3 Starting and Exiting DESIRE/NEUNET	202

DESIRE/NEUNET COMMANDS AND USER PROGRAMS	203
A.4 Command Lines and Operating System Commands	203
A.5 Loading, Listing, and Running a User Program	204
A.6 Saving and Printing Program Files	205
PROGRAM ENTRY. THE TWO DESIRE SCREEN EDITORS	206
A.7 Simple Program Entry and Editing with Line Numbers	206
A.8 Auto-line-number Mode	207
A.9 One-screen Editing	207
A.10 Full Screen Editing	208
A.11 Clearing the Computer Memory	209
FILE MANIPULATION	209
A.12 <b>old</b> and <b>save</b> Statements	209
A.13 Program Identification Codes	210
A.14 Loading ASCII Source Files	211
A.15 Deleting and Renaming Files	211
COMBINING AND CHAINING PROGRAM SEGMENTS	212
A.16 Combining Program and Library Files	212
A.17 Program Chaining	212
<b>Appendix B Interpreted Experiment-protocol Programs</b>	215
OVERVIEW	215
A SIMPLE INTERPRETER LANGUAGE	216
B.1 Interpreter Language	216
B.2 Expressions and Library Functions	216
B.3 Console, File, and Device Output	217
B.4 Console, File, and Device Input	219
B.5 Conditional Branching with <b>if</b> Statements	220
B.6 Labels, Branching, and Hot Keys	221
B.7 <b>for</b> , <b>while</b> , and <b>repeat</b> Loops	222
B.8 Subscripted Variables and Arrays	223
B.9 Data Lists, Read Assignments, and Input Statements Handle Complete Arrays	224
B.10 User-defined Functions	225
B.11 Interpreter Procedures	226
B.12 INTEGER and COMPLEX Variables and Expressions	228
B.13 Octal and Hexadecimal Integers	229
<b>Appendix C Special Features for Interactive Experiments</b>	231
C.1 Interactive Parameter Changes and Program Modifications	231
C.2 Programs with Multiple DYNAMIC Segments	233
C.3 Interactive Debugging	233



**Appendix D DESIRE/NEUNET Library Functions** 235

- D.1 Scalar Library Functions 235
- D.2 Functions Involving COMPLEX Variables 236
- D.3 Pseudorandom Noise 236
- D.4 **VECTOR** and **MATRIX** Functions 237
- D.5 Time Measurements 237
- D.6 Miscellaneous Scalar Functions 237

**References** 230

**Index** 243