Tutorial on
Neural Systems Modeling

Thomas J. Anastasio

Sinauer Associates Inc. Publishers
Sunderland, Massachusetts U.S.A.
CHAPTER 2 Recurrent Connections and Simple Neural Circuits 27

MATH BOX 2.1 THE GEOMETRIC DECAY IS THE DISCRETE-
TIME ANALOG OF THE EXPONENTIAL DECAY 29

MATH BOX 2.2 DISCRETE APPROXIMATION OF A
CONTINUOUS DIFFERENTIAL EQUATION MODEL OF
A SINGLE NEURON 30

2.1 The Dynamics of Two Neural Units with Feedback
in Series 30

MATLAB® BOX 2.1 This script implements a model having
two units in series, each with recurrent, excitatory self-
connections allowing the units to exert positive feedback on
themselves (the two units are leaky integrators) 32

MATH BOX 2.3 DISCRETE APPROXIMATION OF A SYSTEM
OF TWO COUPLED, CONTINUOUS DIFFERENTIAL EQUATIONS
THAT MODEL TWO INTERCONNECTED NEURONS 34

MATH BOX 2.4 EIGENMODE ANALYSIS OF DISCRETE,
LINEAR DYNAMIC SYSTEMS 35

2.2 Signal Processing in the Vestibulo-Ocular Reflex
(VOR) 38

2.3 The Parallel-Pathway Model of Velocity Storage
in the Primate VOR 40

MATLAB® BOX 2.2 The script implements the parallel-
pathway and positive-feedback models of velocity storage,
and the negative-feedback model of velocity leakage 42

2.4 The Positive-Feedback Model of Velocity Storage
in the Primate VOR 44

2.5 The Negative-Feedback Model of Velocity
Leakage in the Pigeon VOR 46

2.6 Oculomotor Neural Integration via Reciprocal
Inhibition 48

MATLAB® BOX 2.3 The script implements the two-unit model
of the integrator of the oculomotor system

2.7 Simulating the Insect-Flight Central Pattern
Generator 53

MATLAB® BOX 2.4 The script implements a linear version
of Wilson’s model of the locust-flight central pattern
generator 56

Exercises 60

References 61

CHAPTER 3 Forward and Recurrent Lateral Inhibition 65

3.1 Simulating Edge Detection in the Early Visual
System of Limulus 69

MATLAB® BOX 3.1 This function will make a connectivity
matrix by laminating shifted versions of a connectivity
profile 72

MATH BOX 3.1 FINITE (DISCRETE) DIFFERENCE
APPROXIMATIONS TO CONTINUOUS DERIVATIVES 74

MATH BOX 3.2 THE MARR FILTER 79

MATH BOX 3.3 ONE- AND TWO-DIMENSIONAL GABOR
FUNCTIONS 80

3.2 Simulating Center/Surround Receptive Fields
Using the Difference of Gaussians 81

MATLAB® BOX 3.2 This function will make a Gaussian
connectivity profile 81

3.3 Simulating Activity Bubbles and Stable Pattern
Formation 85

MATH BOX 3.4 SUMMATION NOTATION 86

MATLAB® BOX 3.3 This script implements the winners-
take-all network that has recurrent, lateral inhibitory and
central excitatory connections and nonlinear units 87

3.4 Separating Signals from Noise and Modeling
Target Selection in the Superior Colliculus 89

Exercises 94

References 95
CHAPTER 4 Covariation Learning and Auto-Associative Memory 97

MATH BOX 4.1 COVARIATION AND ITS RELATIONSHIP WITH COVARIANCE 100

4.1 The Four Hebbian Learning Rules for Neural Networks 101

MATLAB® BOX 4.1 This script implements three different methods to make connection weight matrices for recurrent auto-associative networks using the Hebb, post-synaptic, pre-synaptic, and Hopfield (covariation) rules 104

MATH BOX 4.2 THE OUTER PRODUCT OF TWO VECTORS 106

MATH BOX 4.3 MATRIX MULTIPLICATION: ACROSS THE ROW AND DOWN THE COLUMN 108

4.2 Simulating Memory Recall Using Recurrent Auto-Associator Networks 109

MATH BOX 4.4 LIAPUNOV FUNCTIONS 112

MATLAB® BOX 4.2 This function computes synchronous updates of recurrent auto-associative networks. The entire state vector is updated on each time step 113

4.3 Recalling Distinct Memories Using Negative Connections in Auto-Associators 116

4.4 Synchronous versus Asynchronous Updating in Recurrent Auto-Associators 118

MATLAB® BOX 4.3 This function computes asynchronous updates of recurrent auto-associative networks. The state of only one unit, chosen at random, is updated on each time step 120

4.5 Graceful Degradation and Simulated Forgetting 121

4.6 Simulating Storage and Recall of a Sequence of Patterns 124

MATLAB® BOX 4.4 This script makes asymmetric Hopfield connection weight matrices for sequence recall in recurrent auto-associative neural networks 124

4.7 Hebbian Learning, Recurrent Auto-Association, and Models of Hippocampus 126

Exercises 131

References 133

CHAPTER 5 Unsupervised Learning and Distributed Representations 135

MATH BOX 5.1 VECTOR NORMS, NORMALIZATION, AND INNER PRODUCTS REVISITED 141

5.1 Learning through Competition to Specialize for Specific Inputs 142

MATLAB® BOX 5.1 This script implements the Kohonen self-organizing map (SOM) algorithm 143

5.2 Training Few Output Neurons to Represent Many Input Patterns 145

MATLAB® BOX 5.2 This script can be used to visualize SOM output responses to sets of input patterns as black-and-white images 146

5.3 Simulating the Formation of Brain Maps using Cooperative Mechanisms 148

5.4 Modeling the Formation of Tonotopic Maps in the Auditory System 153

MATLAB® BOX 5.3 This script uses the Kohonen SOM algorithm to simulate the formation of a tonotopic map 156

5.5 Simulating the Development of Orientation Selectivity in Visual Cortex 158

5.6 Modeling a Possible Multisensory Map in the Superior Colliculus 164

Exercises 167

References 168
## CHAPTER 6  Supervised Learning and Non-Uniform Representations  171

6.1 Using the Classic Hebb Rule to Learn a Simple Labeled Line Response  173

MATLAB® BOX 6.1 This script trains two-layered networks of binary units to associate patterns using the Hebb rule  175

6.2 Learning a Simple Contingency Using the Covariation Rule  178

6.3 Using the Delta Rule to Learn a Complex Contingency  180

MATH BOX 6.1 THE GRADIENT OF VECTOR CALCULUS  181

MATH BOX 6.2 DERIVATION OF THE DELTA RULE LEARNING ALGORITHM  183

MATH BOX 6.3 DIFFERENTIATING THE SQUASHING FUNCTION  184

MATLAB® BOX 6.2 This script trains two-layered networks of sigmoidal units to associate patterns using the delta rule  186

6.4 Learning Interneuronal Representations using Back-Propagation  189

MATH BOX 6.4 MULTILAYERED NETWORKS OF LINEAR PROCESSING UNITS  190

MATH BOX 6.5 DERIVATION OF THE BACK-PROPAGATION LEARNING ALGORITHM  191

MATLAB® BOX 6.3 This script trains three-layered networks of sigmoidal units to associate patterns using back-propagation  194

6.5 Simulating Catastrophic Retroactive Interference in Learning  196

6.6 Simulating the Development of Non-Uniform Distributed Representations  198

6.7 Modeling Non-Uniform Distributed Representations in the Vestibular Nuclei  201

Exercises  209

References  210

## CHAPTER 7  Reinforcement Learning and Associative Conditioning  213

7.1 Learning the Labeled-Line Task via Perturbation of One Weight at a Time  216

MATLAB® BOX 7.1 This script trains two-layered networks of sigmoidal units (with only one output unit) to associate patterns using perturbation to estimate the error gradient at one weight at a time  220

7.2 Perturbing All Weights Simultaneously and the Importance of Structure  221

MATH BOX 7.1 THE BROWNIAN MOTION ALGORITHM  222

MATLAB® BOX 7.2 This script trains two-layered networks of sigmoidal units to associate patterns using perturbation to estimate the error gradient at all weights simultaneously  223

7.3 Plausible Weight Modification using Perturbative Reinforcement Learning  227

MATLAB® BOX 7.3 This script trains two-layered networks of sigmoidal units to associate patterns by perturbing all weights simultaneously  228

7.4 Reinforcement Learning and Non-Uniform Distributed Representations  230

MATLAB® BOX 7.4 This script trains three-layered networks of sigmoidal units to develop a distributed representation using parallel, perturbative reinforcement learning  232

7.5 Reinforcement in a Schema Model of Avoidance Conditioning  234

MATLAB® BOX 7.5 This script simulates avoidance conditioning as reinforcement learning  239

7.6 Exploration and Exploitation in a Model of Avoidance Conditioning  242

MATLAB® BOX 7.6 This script simulates avoidance conditioning as reinforcement learning with automatic adjustment of exploration  244

Exercises  247

References  248
CHAPTER 8  Information Transmission and Unsupervised Learning  251

8.1 Some Basic Concepts in Information Theory  253

8.2 Measuring Information Transmission through a Neural Network  257

MATLAB® BOX 8.1 This script computes input-output mutual information in a feedforward network with two stochastic, binary input units and two deterministic, binary output units  258

MATLAB® BOX 8.2 This function computes the input and output entropy and the input-output mutual information for a neural network  261

8.3 Maximizing Information Transmission in a Neural Network  265

MATH BOX 8.1 DERIVATION OF THE BELL–SEJNOWSKI INFORMAX ALGORITHM FOR A NETWORK WITH ONE INPUT UNIT AND ONE OUTPUT UNIT  266

MATLAB® BOX 8.3 This script trains a 2-by-2, feedforward network using the Bell–Sejnowski informax algorithm, and finds the input-output mutual information  268

8.4 Information Transmission and Competitive Learning in Neural Networks  276

MATLAB® BOX 8.4 This script trains a 2-by-2, feedforward network using competitive, unsupervised learning, and finds the input-output mutual information  277

8.5 Information Transmission in Self-Organized Map Networks  279

MATLAB® BOX 8.5 This script trains a self-organizing map network and finds informational measures, including estimated distortion, following training  281

MATH BOX 8.2 RATE-DISTORTION THEORY  284

8.6 Information Transmission in Stochastic Neural Networks  290

MATLAB® BOX 8.6 This script computes informational measures for a stochastic network of binary units with one input unit and arbitrarily many output units  293

Exercises  296
References  298

CHAPTER 9  Probability Estimation and Supervised Learning  299

9.1 Implementing a Simple Classifier as a Three-Layered Neural Network  302

MATLAB® BOX 9.1 This script trains a three-layered network of sigmoidal units using back-propagation to classify fish according to their lengths  303

MATH BOX 9.1 UNIVARIATE, MULTIVARIATE, AND BIVARIATE GAUSSIAN DISTRIBUTIONS  305

9.2 Predicting Rain as an Everyday Example of Probabilistic Inference  308

MATH BOX 9.2 THE DERIVATION OF BAYES' RULE  309

9.3 Implementing a Simple Classifier Using Bayes' Rule  311

MATLAB® BOX 9.2 This script computes the posterior probabilities of each of three hypothetical fish classes using Bayes' rule  313

9.4 Modeling Neural Responses to Sensory Input as Probabilistic Inference  315

MATLAB® BOX 9.3 This script computes the posterior probability of a target given sensory input of one modality (i.e., visual)  317

MATLAB® BOX 9.4 This script trains a single sigmoidal unit using the delta rule to estimate posterior target probability given sensory input of one modality (i.e., visual)  319

MATH BOX 9.3 SOLVING FOR THE INPUT AND BIAS WEIGHTS OF A SIGMOIDAL UNIT THAT COMPUTES A POSTERIOR PROBABILITY  321

9.5 Modeling Multisensory Collicular Neurons as Probability Estimators  323

MATLAB® BOX 9.5 This script computes the posterior probability of a target given sensory input of two modalities (i.e., visual and auditory)  328

MATLAB® BOX 9.6 This script trains a single sigmoidal unit using the delta rule to estimate posterior target probability given sensory input of two modalities (i.e., visual and auditory)  330

Exercises  338
References  339
## CHAPTER 10  Time Series Learning and Nonlinear Signal Processing  341

10.1  Training Connection Weights in Recurrent Neural Networks  344

MATH BOX 10.1  DERIVATION OF REAL-TIME RECURRENT BACK-PROPAGATION  345

10.2  Training a Two-Unit Network to Simulate the Oculomotor Neural Integrator  348

MATLAB® BOX 10.1  This script uses recurrent back-propagation to train a recurrent network with two linear output units to act as a leaky integrator  351

10.3  Velocity Storage in the Vestibulo-Ocular Reflex  354

10.4  Training a Network of Linear Units to Produce Velocity Storage  357

MATLAB® BOX 10.2  This script uses recurrent back-propagation to train a linear recurrent network to perform velocity storage  360

10.5  Training Networks of Nonlinear Units to Produce Velocity Storage  363

MATLAB® BOX 10.3  This script uses recurrent back-propagation to train a nonlinear recurrent network to perform velocity storage (Training requires several minutes)  364

10.6  Training a Recurrent Neural Network to Simulate Short-Term Memory  374

MATLAB® BOX 10.4  This script uses recurrent back-propagation to train a nonlinear recurrent network to simulate short-term memory (Training requires several minutes)  377

MATLAB® BOX 10.5  This script tests the ability of recurrent neural networks to simulate short-term memory  378

Exercises  383

References  384

## CHAPTER 11  Temporal-Difference Learning and Reward Prediction  387

MATLAB® BOX 11.1  This script sets up a stochastic gridworld  391

MATH BOX 11.1  FINDING STATE VALUES BY SOLVING A SET OF SIMULTANEOUS LINEAR EQUATIONS  394

11.1  Learning State Values Using Iterative Dynamic Programming  395

MATLAB® BOX 11.2  This script updates state value estimates for the stochastic gridworld using iterative dynamic programming  396

11.2  Learning State Values Using Least Mean Squares  399

MATLAB® BOX 11.3  This script updates state value estimates for the stochastic gridworld using least-mean-squares learning  401

11.3  Learning State Values Using the Method of Temporal Differences  403

MATLAB® BOX 11.4  This script updates state value estimates for the stochastic gridworld using temporal-difference learning  405

11.4  Simulating Dopamine Neuron Responses Using Temporal-Difference Learning  408

MATLAB® BOX 11.5  This script simulates the responses of midbrain dopamine neurons using temporal-difference learning  414

11.5  Temporal-Difference Learning as a Form of Supervised Learning  416

Exercises  419

References  420
CHAPTER 12  Predictor–Corrector Models and Probabilistic Inference  423

MATH BOX 12.1  THE KALMAN FILTER  425

MATLAB® BOX 12.1  This script implements a running average  426

12.1  Modeling Visual System Direction Selectivity Using Asymmetric Inhibition  428

MATLAB® BOX 12.2  This script simulates direction selectivity in the visual system  431

12.2  Modeling Visual Processing as Bottom-Up/Top-Down Probabilistic Inference  434

MATLAB® BOX 12.3  This script simulates bottom-up/top-down processing in the visual system using the joint distribution  442

MATH BOX 12.2  BELIEF PROPAGATION  446

MATLAB® BOX 12.4  This script simulates bottom-up/top-down processing in the visual system using probabilistic inference  447

12.3  A Predictor–Corrector Model of Predictive Tracking by Midbrain Neurons  450

MATH BOX 12.3  DECISION-THEORETIC AGENT DESIGN  454

MATLAB® BOX 12.5  This script sets up the target-tracking predictor-corrector model  462

MATLAB® BOX 12.6  This script implements a predictor-corrector simulation of the responses of neurons in the parabigeminal nucleus  463

12.4  Training a Sigmoidal Unit to Simulate Trajectory Prediction by Neurons  469

MATLAB® BOX 12.7  This script uses the delta rule to train a single sigmoidal unit with feedback to simulate the responses of neurons in the parabigeminal nucleus  472

Exercises  475

References  477

CHAPTER 13  Simulated Evolution and the Genetic Algorithm  481

13.1  Simulating Genes and Genetic Operators  487

13.2  Exploring a Simple Example of Simulated Genetic Evolution  488

MATLAB® BOX 13.1  THE SCHEMA THEOREM FOR THE GENETIC ALGORITHM  489

MATLAB® BOX 13.2  FINDING THE MINIMA (MAXIMA) OF FUNCTIONS  490

MATLAB® BOX 13.1  The script uses the genetic algorithm with binary genes to find the minimum of a function  491

13.3  Evolving the Sizes of Neural Networks to Improve Learning  495

MATLAB® BOX 13.2  This script uses the genetic algorithm to optimize the speed of learning by evolving the number of hidden units in three-layered, feedforward networks trained using back-propagation  499

MATLAB® BOX 13.3  This script trains a three-layered network of sigmoidal units to associate patterns using back-propagation  500

13.4  Evolving Optimal Learning Rules for Auto-Associative Memories  502

MATLAB® BOX 13.4  This script uses the genetic algorithm to optimize the Hopfield rule  504

13.5  Evolving Connectivity Profiles for Activity-Bubble Neural Networks  506

MATLAB® BOX 13.5  This script uses the genetic algorithm to optimize the connectivity profile of the activity-bubble network  506

Exercises  511

References  513