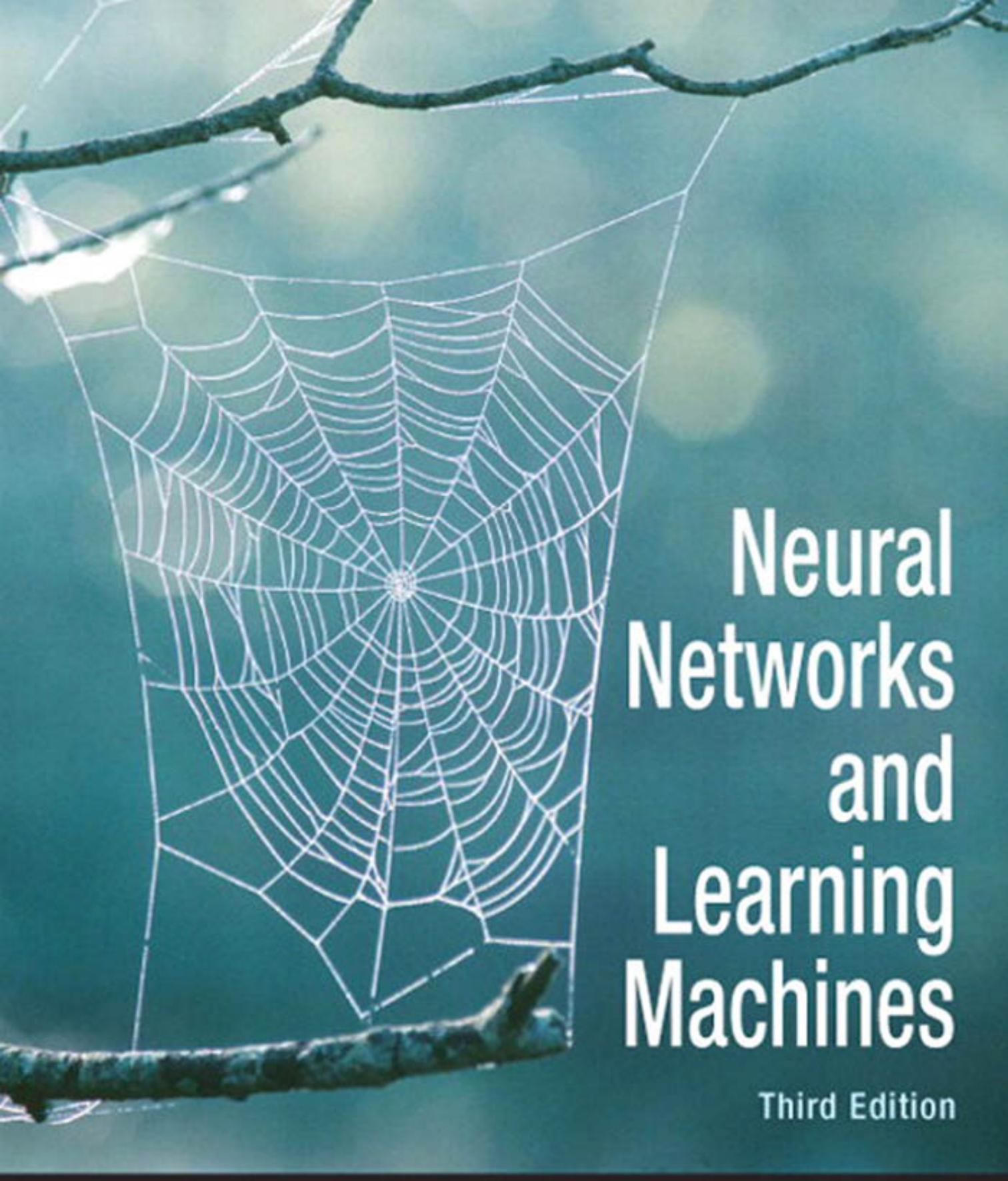

Neural Networks and Learning Machines

Third Edition

Simon Haykin

Neural Networks and Learning Machines

Third Edition

Simon Haykin
McMaster University
Hamilton, Ontario, Canada

New York Boston San Francisco
London Toronto Sydney Tokyo Singapore Madrid
Mexico City Munich Paris Cape Town Hong Kong Montreal

Library of Congress Cataloging-in-Publication Data
Haykin, Simon
Neural networks and learning machines / Simon Haykin.—3rd ed.
p. cm.
Rev. ed of: Neural networks. 2nd ed., 1999.
Includes bibliographical references and index.
ISBN-13: 978-0-13-147139-9
ISBN-10: 0-13-147139-2
1. Neural networks (Computer science) 2. Adaptive filters. I. Haykin, Simon Neural networks. II. Title.
QA76.87.H39 2008
006.3′--dc22
2008034079

Vice President and Editorial Director, ECS: Marcia J. Horton
Associate Editor: Alice Dworkin
Supervisor/Editorial Assistant: Dolores Mars
Editorial Assistant: William Opaluch
Director of Team-Based Project Management: Vince O'Brien
Senior Managing Editor: Scott Disanno
A/V Production Editor: Greg Dulles
Art Director: Jayne Conte
Cover Designer: Bruce Kenselaar
Manufacturing Manager: Alan Fischer
Manufacturing Buyer: Lisa McDowell
Marketing Manager: Tim Galligan

---

10 9 8 7 6 5 4 3 2 1
ISBN-13: 978-0-13-147139-9
ISBN-10: 0-13-147139-2

To my wife, Nancy, for her patience and tolerance,

and

to the countless researchers in neural networks for their original contributions, the many reviewers for their critical inputs, and many of my graduate students for their keen interest.

Contents

Preface

In writing this third edition of a classic book, I have been guided by the same underlying philosophy of the first edition of the book:

> *Write an up-to-date treatment of neural networks in a comprehensive, thorough, and readable manner.*

The new edition has been retitled *Neural Networks and Learning Machines*, in order to reflect two realities:

1. The perceptron, the multilayer perceptron, self-organizing maps, and neurodynamics, to name a few topics, have always been considered integral parts of neural networks, rooted in ideas inspired by the human brain.
2. Kernel methods, exemplified by support-vector machines and kernel principal-components analysis, are rooted in statistical learning theory.

Although, indeed, they share many fundamental concepts and applications, there are some subtle differences between the operations of neural networks and learning machines. The underlying subject matter is therefore much richer when they are studied together, under one umbrella, particularly so when

- ideas drawn from neural networks and machine learning are hybridized to perform improved learning tasks beyond the capability of either one operating on its own, and
- ideas inspired by the human brain lead to new perspectives wherever they are of particular importance.

Moreover, the scope of the book has been broadened to provide detailed treatments of dynamic programming and sequential state estimation, both of which have affected the study of reinforcement learning and supervised learning, respectively, in significant ways.

Organization of the Book

The book begins with an introductory chapter that is motivational, paving the way for the rest of the book which is organized into six parts as follows:

1. Chapters 1 through 4, constituting the first part of the book, follow the classical approach on supervised learning. Specifically,

- Chapter 1 describes Rosenblatt's perceptron, highlighting the perceptron convergence theorem, and the relationship between the perceptron and the Bayesian classifier operating in a Gaussian environment.
- Chapter 2 describes the method of least squares as a basis for model building. The relationship between this method and Bayesian inference for the special case of a Gaussian environment is established. This chapter also includes a discussion of the minimum description length (MDL) principle for model selection.
- Chapter 3 is devoted to the least-mean-square (LMS) algorithm and its convergence analysis. The theoretical framework of the analysis exploits two principles: Kushner's direct method and the Langevin equation (well known in nonequilibrium thermodynamics).

These three chapters, though different in conceptual terms, share a common feature: They are all based on a single computational unit. Most importantly, they provide a great deal of insight into the learning process in their own individual ways—a feature that is exploited in subsequent chapters.

Chapter 4, on the multilayer perceptron, is a generalization of Rosenblatt's perceptron. This rather long chapter covers the following topics:

- the back-propagation algorithm, its virtues and limitations, and its role as an optimum method for computing partial derivations;
- optimal annealing and adaptive control of the learning rate;
- cross-validation;
- convolutional networks, inspired by the pioneering work of Hubel and Wiesel on visual systems;
- supervised learning viewed as an optimization problem, with attention focused on conjugate-gradient methods, quasi-Newton methods, and the Marquardt–Levenberg algorithm;
- nonlinear filtering;
- last, but by no means least, a contrasting discussion of small-scale versus large-scale learning problems.

2. The next part of the book, consisting of Chapters 5 and 6, discusses kernel methods based on radial-basis function (RBF) networks.

In a way, Chapter 5 may be viewed as an insightful introduction to kernel methods. Specifically, it does the following:

- presents Cover's theorem as theoretical justification for the architectural structure of RBF networks;
- describes a relatively simple two-stage hybrid procedure for supervised learning, with stage 1 based on the idea of clustering (namely, the K-means algorithm) for computing the hidden layer, and stage 2 using the LMS or the method of least squares for computing the linear output layer of the network;
- presents kernel regression and examines its relation to RBF networks.

Chapter 6 is devoted to support vector machines (SVMs), which are commonly recognized as a method of choice for supervised learning. Basically, the SVM is a binary classifier, in the context of which the chapter covers the following topics:

- the condition for defining the maximum margin of separation between a pair of linearly separable binary classes;
- quadratic optimization for finding the optimal hyperplane when the two classes are linearly separable and when they are not;
- the SVM viewed as a kernel machine, including discussions of the kernel trick and Mercer's theorem;
- the design philosophy of SVMs;
- the ϵ-insensitive loss function and its role in the optimization of regression problems;
- the Representer Theorem, and the roles of Hilbert space and reproducing kernel Hilbert space (RKHS) in its formulation.

From this description, it is apparent that the underlying theory of support vector machines is built on a strong mathematical background—hence their computational strength as an elegant and powerful tool for supervised learning.

3. The third part of the book involves a single chapter, Chapter 7. This broadly based chapter is devoted to regularization theory, which is at the core of machine learning. The following topics are studied in detail:

- Tikhonov's classic regularization theory, which builds on the RKHS discussed in Chapter 6. This theory embodies some profound mathematical concepts: the Fréchet differential of the Tikhonov functional, the Riesz representation theorem, the Euler–Lagrange equation, Green's function, and multivariate Gaussian functions;
- generalized RBF networks and their modification for computational tractability;
- the regularized least-squares estimator, revisited in light of the Representer Theorem;
- estimation of the regularization parameter, using Wahba's concept of generalized cross-validation;
- semisupervised learning, using labeled as well as unlabeled examples;
- differentiable manifolds and their role in manifold regularization—a role that is basic to designing semisupervised learning machines;
- spectral graph theory for finding a Gaussian kernel in an RBF network used for semisupervised learning;
- a generalized Representer Theorem for dealing with semisupervised kernel machines;
- the Laplacian regularized least-squares (LapRLS) algorithm for computing the linear output layer of the RBF network; here, it should be noted that when the intrinsic regularization parameter (responsible for the unlabeled data) is reduced to zero, the algorithm is correspondingly reduced to the ordinary least-squares algorithm.

This highly theoretical chapter is of profound practical importance. First, it provides the basis for the regularization of supervised-learning machines. Second, it lays down the groundwork for designing regularized semisupervised learning machines.

4. Chapters 8 through 11 constitute the fourth part of the book, dealing with unsupervised learning. Beginning with Chapter 8, four principles of self-organization, intuitively motivated by neurobiological considerations, are presented:

(i) Hebb's postulate of learning for self-amplification;
(ii) Competition among the synapses of a single neuron or a group of neurons for limited resources;
(iii) Cooperation among the winning neuron and its neighbors;
(iv) Structural information (e.g., redundancy) contained in the input data.

The main theme of the chapter is threefold:

- Principles (i), (ii), and (iv) are applied to a single neuron, in the course of which Oja's rule for maximum eigenfiltering is derived; this is a remarkable result obtained through self-organization, which involves bottom-up as well as top-down learning. Next, the idea of maximum eigenfiltering is generalized to principal-components analysis (PCA) on the input data for the purpose of dimensionality reduction; the resulting algorithm is called the generalized Hebbian algorithm (GHA).
- Basically, PCA is a linear method, the computing power of which is therefore limited to second-order statistics. In order to deal with higher-order statistics, the kernel method is applied to PCA in a manner similar to that described in Chapter 6 on support vector machines, but with one basic difference: unlike SVM, kernel PCA is performed in an unsupervised manner.
- Unfortunately, in dealing with natural images, kernel PCA can become unmanageable in computational terms. To overcome this computational limitation, GHA and kernel PCA are hybridized into a new on-line unsupervised learning algorithm called the kernel Hebbian algorithm (KHA), which finds applications in image denoising.

The development of KHA is an outstanding example of what can be accomplished when an idea from machine learning is combined with a complementary idea rooted in neural networks, producing a new algorithm that overcomes their respective practical limitations.

Chapter 9 is devoted to self-organizing maps (SOMs), the development of which follows the principles of self-organization described in Chapter 8. The SOM is a simple algorithm in computational terms, yet highly powerful in its built-in ability to construct organized topographic maps with several useful properties:

- spatially discrete approximation of the input space, responsible for data generation;
- topological ordering, in the sense that the spatial location of a neuron in the topographic map corresponds to a particular feature in the input (data) space;
- input–output density matching;
- input-data feature selection.

The SOM has been applied extensively in practice; the construction of contextual maps and hierarchical vector quantization are presented as two illustrative examples of the SOM's computing power. What is truly amazing is that the SOM exhibits several interesting properties and solves difficult computational tasks, yet it lacks an objective function that could be optimized. To fill this gap and thereby provide the possibility of improved topographic mapping, the self-organizing map is kernelized. This is done by introducing an entropic function as the objective

function to be maximized. Here again, we see the practical benefit of hybridizing ideas rooted in neural networks with complementary kernel-theoretic ones.

Chapter 10 exploits principles rooted in Shannon's information theory as tools for unsupervised learning. This rather long chapter begins by presenting a review of Shannon's information theory, with particular attention given to the concepts of entropy, mutual information, and the Kullback–Leibler divergence (KLD). The review also includes the concept of copulas, which, unfortunately, has been largely overlooked for several decades. Most importantly, the copula provides a measure of the statistical dependence between a pair of correlated random variables. In any event, focusing on mutual information as the objective function, the chapter establishes the following principles:

- The Infomax principle, which maximizes the mutual information between the input and output data of a neural system; Infomax is closely related to redundancy reduction.
- The Imax principle, which maximizes the mutual information between the single outputs of a pair of neural systems that are driven by correlated inputs.
- The Imin principle operates in a manner similar to the Imax principle, except that the mutual information between the pair of output random variables is minimized.
- The independent-components analysis (ICA) principle, which provides a powerful tool for the blind separation of a hidden set of statistically independent source signals. Provided that certain operating conditions are satisfied, the ICA principle affords the basis for deriving procedures for recovering the original source signals from a corresponding set of observables that are linearly mixed versions of the source signals. Two specific ICA algorithms are described:

  (i) the natural-gradient learning algorithm, which, except for scaling and permutation, solves the ICA problem by minimizing the KLD between a parameterized probability density function and the corresponding factorial distribution;

  (ii) the maximum-entropy learning algorithm, which maximizes the entropy of a nonlinearly transformed version of the demixer output; this algorithm, commonly known as the Infomax algorithm for ICA, also exhibits scaling and permutation properties.

Chapter 10 also describes another important ICA algorithm, known as FastICA, which, as the name implies, is computationally fast. This algorithm maximizes a contrast function based on the concept of negentropy, which provides a measure of the non-Gaussianity of a random variable. Continuing with ICA, the chapter goes on to describe a new algorithm known as coherent ICA, the development of which rests on fusion of the Infomax and Imax principles via the use of the copula; coherent ICA is useful for extracting the envelopes of a mixture of amplitude-modulated signals. Finally, Chapter 10 introduces another concept rooted in Shannon's information theory, namely, rate distortion theory, which is used to develop the last concept in the chapter: information bottleneck. Given the joint distribution of an input vector and a (relevant) output vector, the method is formulated as a constrained

optimization problem in such a way that a tradeoff is created between two amounts of information, one pertaining to information contained in the bottleneck vector about the input and the other pertaining to information contained in the bottleneck vector about the output. The chapter then goes on to find an optimal manifold for data representation, using the information bottleneck method.

The final approach to unsupervised learning is described in Chapter 11, using stochastic methods that are rooted in statistical mechanics; the study of statistical mechanics is closely related to information theory. The chapter begins by reviewing the fundamental concepts of Helmholtz free energy and entropy (in a statistical mechanics sense), followed by the description of Markov chains. The stage is then set for describing the Metropolis algorithm for generating a Markov chain, the transition probabilities of which converge to a unique and stable distribution. The discussion of stochastic methods is completed by describing simulated annealing for global optimization, followed by Gibbs sampling, which can be used as a special form of the Metropolis algorithm. With all this background on statistical mechanics at hand, the stage is set for describing the Boltzmann machine, which, in a historical context, was the first multilayer learning machine discussed in the literature. Unfortunately, the learning process in the Boltzmann machine is very slow, particularly when the number of hidden neurons is large—hence the lack of interest in its practical use. Various methods have been proposed in the literature to overcome the limitations of the Boltzmann machine. The most successful innovation to date is the deep belief net, which distinguishes itself in the clever way in which the following two functions are combined into a powerful machine:

- generative modeling, resulting from bottom-up learning on a layer-by-layer basis and without supervision;
- inference, resulting from top-down learning.

Finally, Chapter 10 describes deterministic annealing to overcome the excessive computational requirements of simulated annealing; the only problem with deterministic annealing is that it could get trapped in a local minimum.

5. Up to this point, the focus of attention in the book has been the formulation of algorithms for supervised learning, semisupervised learning, and unsupervised learning. Chapter 12, constituting the next part of the book all by itself, addresses reinforcement learning, in which learning takes place in an on-line manner as the result of an agent (e.g., robot) interacting with its surrounding environment. In reality, however, dynamic programming lies at the core of reinforcement learning. Accordingly, the early part of Chapter 15 is devoted to an introductory treatment of Bellman's dynamic programming, which is then followed by showing that the two widely used methods of reinforcement learning: Temporal difference (TD) learning, and Q-learning can be derived as special cases of dynamic programming. Both TD-learning and Q-learning are relatively simple, on-line reinforcement learning algorithms that do not require knowledge of transition probabilities. However, their practical applications are limited to situations in which the dimensionality of the state space is of moderate size. In large-scale dynamic systems, the curse of dimensionality becomes a serious issue, making not only dynamic programming,

but also its approximate forms, TD-learning and Q-learning, computationally intractable. To overcome this serious limitation, two indirect methods of approximate dynamic programming are described:

- a linear method called the least-squares policy evaluation (LSPV) algorithm, and
- a nonlinear method using a neural network (e.g., multilayer perceptron) as a universal approximator.

6. The last part of the book, consisting of Chapters 13, 14, and 15, is devoted to the study of nonlinear feedback systems, with an emphasis on recurrent neural networks:

(i) Chapter 13 studies neurodynamics, with particular attention given to the stability problem. In this context, the direct method of Lyapunov is described. This method embodies two theorems, one dealing with stability of the system and the other dealing with asymptotic stability. At the heart of the method is a Lyapunov function, for which an energy function is usually found to be adequate. With this background theory at hand, two kinds of associative memory are described:

- the Hopfield model, the operation of which demonstrates that a complex system is capable of generating simple emergent behavior;
- the brain-state-in-a-box model, which provides a basis for clustering.

The chapter also discusses properties of chaotic processes and a regularized procedure for their dynamic reconstruction.

(ii) Chapter 14 is devoted to the Bayesian filter, which provides a unifying basis for sequential state estimation algorithms, at least in a conceptual sense. The findings of the chapter are summarized as follows:

- The classic Kalman filter for a linear Gaussian environment is derived with the use of the minimum mean-square-error criterion; in a problem at the end of the chapter, it is shown that the Kalman filter so derived is a special case of the Bayesian filter;
- square-root filtering is used to overcome the divergence phenomenon that can arise in practical applications of the Kalman filter;
- the extended Kalman filter (EKF) is used to deal with dynamic systems whose nonlinearity is of a mild sort; the Gaussian assumption is maintained;
- the direct approximate form of the Bayesian filter is exemplified by a new filter called the cubature Kalman filter (CKF); here again, the Gaussian assumption is maintained;
- indirect approximate forms of the Bayesian filter are exemplified by particle filters, the implementation of which can accommodate nonlinearity as well as non-Gaussianity.

With the essence of Kalman filtering being that of a predictor–corrector, Chapter 14 goes on to describe the possible role of "Kalman-like filtering" in certain parts of the human brain.

The final chapter of the book, Chapter 15, studies dynamically driven recurrent neural networks. The early part of the chapter discusses different structures (models) for recurrent networks and their computing power, followed by two algorithms for the training of recurrent networks:

- back propagation through time, and
- real-time recurrent learning.

Unfortunately both of these procedures, being gradient based, are likely to suffer from the so-called vanishing-gradients problem. To mitigate the problem, the use of nonlinear sequential state estimators is described at some length for the supervised training of recurrent networks in a rather novel manner. In this context, the advantages and disadvantages of the extended Kalman filter (simple, but derivative dependent) and the cubature Kalman filter (derivative free, but more complicated mathematically) as sequential state estimator for supervised learning are discussed. The emergence of adaptive behavior, unique to recurrent networks, and the potential benefit of using an adaptive critic to further enhance the capability of recurrent networks are also discussed in the chapter.

An important topic featuring prominently in different parts of the book is supervised learning and semisupervised learning applied to large-scale problems. The concluding remarks of the book assert that this topic is in its early stages of development; most importantly, a four-stage procedure is described for its future development.

Distinct Features of the Book

Over and above the broad scope and thorough treatment of the topics summarized under the organization of the book, distinctive features of the text include the following:

1. Chapters 1 through 7 and Chapter 10 include computer experiments involving the double-moon configuration for generating data for the purpose of binary classification. The experiments range from the simple case of linearly separable patterns to difficult cases of nonseparable patterns. The double-moon configuration, as a running example, is used all the way from Chapter 1 to Chapter 7, followed by Chapter 10, thereby providing an experimental means for studying and comparing the learning algorithms described in those eight chapters.
2. Computer experiments are also included in Chapter 8 on PCA, Chapter 9 on SOM and kernel SOM, and Chapter 14 on dynamic reconstruction of the Mackay–Glass attractor using the EKF and CKF algorithms.
3. Several case studies, using real-life data, are presented:
   - Chapter 7 discusses the United States Postal Service (USPS) data for semisupervised learning using the Laplacian RLS algorithm;
   - Chapter 8 examines how PCA is applied to handwritten digital data and describes the coding and denoising of images;
   - Chapter 10 treats the analysis of natural images by using sparse-sensory coding and ICA;
   - Chapter 13 presents dynamic reconstruction applied to the Lorenz attractor by using a regularized RBF network.

Chapter 15 also includes a section on the model reference adaptive control system as a case study.

4. Each chapter ends with notes and references for further study, followed by end-of-chapter problems that are designed to challenge, and therefore expand, the reader's expertise.

   The glossary at the front of the book has been expanded to include explanatory notes on the methodology used on matters dealing with matrix analysis and probability theory.

5. PowerPoint files of all the figures and tables in the book will be available to Instructors and can be found at www.prenhall.com/haykin.

6. Matlab codes for all the computer experiments in the book are available on the Website of the publisher to all those who have purchased copies of the book. These are available to students at www.pearsonhighered.com/haykin.

7. The book is accompanied by a Manual that includes the solutions to all the end-of-chapter problems as well as computer experiments.

   The manual is available from the publisher, Prentice Hall, only to instructors who use the book as the recommended volume for a course, based on the material covered in the book.

Last, but by no means least, every effort has been expended to make the book error free and, most importantly, readable.

Simon Haykin
Ancaster, Ontario

Acknowledgments

I am deeply indebted to many renowned authorities on neural networks and learning machines around the world, who have provided invaluable comments on selected parts of the book:

Dr. Sun-Ichi Amari, The RIKEN Brain Science Institute, Wako City, Japan

Dr. Susanne Becker, Department of Psychology, Neuroscience & Behaviour, McMaster University, Hamilton, Ontario, Canada

Dr. Dimitri Bertsekas, MIT, Cambridge, Massachusetts

Dr. Leon Bottou, NEC Laboratories America, Princeton, New Jersey

Dr. Simon Godsill, University of Cambridge, Cambridge, England

Dr. Geoffrey Gordon, Carnegie-Mellon University, Pittsburgh, Pennsylvania

Dr. Peter Grünwald, CWI, Amsterdam, the Netherlands

Dr. Geoffrey Hinton, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada

Dr. Timo Honkela, Helsinki University of Technology, Helsinki, Finland

Dr. Tom Hurd, Department of Mathematics and Statistics, McMaster University, Ontario, Canada.

Dr. Eugene Izhikevich, The Neurosciences Institute, San Diego, California

Dr. Juha Karhunen, Helsinki University of Technology, Helsinki, Finland

Dr. Kwang In Kim, Max-Planck-Institut für Biologische Kybernetik, Tübingen, Germany

Dr. James Lo, University of Maryland at Baltimore County, Baltimore, Maryland

Dr. Klaus Müller, University of Potsdam and Fraunhofer Institut FIRST, Berlin, Germany

Dr. Erkki Oja, Helsinki University of Technology, Helsinki, Finland

Dr. Bruno Olshausen, Redwood Center for Theoretical Neuroscience, University of California, Berkeley, California

Dr. Danil Prokhorov, Toyota Technical Center, Ann Arbor, Michigan

Dr. Kenneth Rose, Electrical and Computer Engineering, University of California, Santa Barbara, California

Dr. Bernhard Schölkopf, Max-Planck-Institut für Biologische Kybernetik, Tübingen, Germany

Dr. Vikas Sindhwani, Department of Computer Science, University of Chicago, Chicago, Illinois

Dr. Sergios Theodoridis, Department of Informatics, University of Athens, Athens, Greece

Dr. Naftali Tishby, The Hebrew University, Jerusalem, Israel

Dr. John Tsitsiklis, Massachusetts Institute of Technology, Cambridge, Massachusetts

Dr. Marc Van Hulle, Katholieke Universiteit, Leuven, Belgium

Several photographs and graphs have been reproduced in the book with permissions provided by Oxford University Press and

Dr. Anthony Bell, Redwood Center for Theoretical Neuroscience, University of California, Berkeley, California

Dr. Leon Bottou, NEC Laboratories America, Princeton, New Jersey

Dr. Juha Karhunen, Helsinki University of Technology, Helsinki, Finland

Dr. Bruno Olshausen, Redwood Center for Theoretical Neuroscience, University of California, Berkeley, California

Dr. Vikas Sindhwani, Department of Computer Science, University of Chicago, Chicago, Illinois

Dr. Naftali Tishby, The Hebrew University, Jerusalem, Israel

Dr. Marc Van Hulle, Katholieke Universiteit, Leuven, Belgium

I thank them all most sincerely.

I am grateful to my graduate students:

1. Yanbo Xue, for his tremendous effort devoted to working on nearly all the computer experiments produced in the book, and also for reading the second page proofs of the book.
2. Karl Wiklund, for proofreading the entire book and making valuable comments for improving it.
3. Haran Arasaratnam, for working on the computer experiment dealing with the Mackay–Glass attractor.
4. Andreas Wendel (Graz University of technology, Austria) while he was on leave at McMaster University, 2008.

I am grateful to Scott Disanno and Alice Dworkin of Prentice Hall for their support and hard work in the production of the book. Authorization of the use of color in the book by Marcia Horton is truly appreciated; the use of color has made a tremendous difference to the appearance of the book from cover to cover.

I am grateful to Jackie Henry of Aptara Corp. and her staff, including Donald E. Smith, Jr., the proofreader, for the production of the book. I also wish to thank Brian Baker and the copyeditor, Abigail Lin, at Write With, Inc., for their effort in copy-editing the manuscript of the book.

The tremendous effort by my Technical Coordinator, Lola Brooks, in typing several versions of the chapters in the book over the course of 12 months, almost nonstop, is gratefully acknowledged.

Last, but by no means least, I thank my wife, Nancy, for having allowed me the time and space, which I have needed over the last 12 months, almost nonstop, to complete the book in a timely fashion.

Simon Haykin

Abbreviations and Symbols

ABBREVIATIONS

AR	autoregressive
BBTT	back propagation through time
BM	Boltzmann machine
BP	back propagation
b/s	bits per second
BSB	brain-state-in-a-box
BSS	Blind source (signal) separation
cmm	correlation matrix memory
CV	cross-validation
DFA	deterministic finite-state automata
EKF	extended Kalman filter
EM	expectation-maximization
FIR	finite-duration impulse response
FM	frequency-modulated (signal)
GCV	generalized cross-validation
GHA	generalized Hebbian algorithm
GSLC	generalized sidelobe canceler
Hz	hertz
ICA	independent-components analysis
Infomax	maximum mutual information
Imax	variant of Infomax
Imin	another variant of Infomax
KSOM	kernel self-organizing map
KHA	kernel Hebbian algorithm
LMS	least-mean-square
LR	likelihood ratio

LS	Least-squares
LS-TD	Least-squares, temporal-difference
LTP	long-term potentiation
LTD	long-term depression
LR	likelihood ratio
LRT	Likelihood ratio test
MAP	Maximum a posteriori
MCA	minor-components analysis
MCMC	Markov Chan Monte Carlo
MDL	minimum description length
MIMO	multiple input–multiple output
ML	maximum likelihood
MLP	multilayer perceptron
MRC	model reference control
NARMA	nonlinear autoregressive moving average
NARX	nonlinear autoregressive with exogenous inputs
NDP	neuro-dynamic programming
NW	Nadaraya–Watson (estimator)
NWKR	Nadaraya–Watson kernal regression
OBD	optimal brain damage
OBS	optimal brain surgeon
OCR	optical character recognition
PAC	probably approximately correct
PCA	principal-components analysis
PF	Particle Filter
pdf	probability density function
pmf	probability mass function
QP	quadratic programming
RBF	radial basis function
RLS	recursive least-squares
RLS	regularized least-squares
RMLP	recurrent multilayer perceptron
RTRL	real-time recurrent learning
SIMO	single input–multiple output
SIR	sequential importance resampling
SIS	sequential important sampling
SISO	single input–single output
SNR	signal-to-noise ratio
SOM	self-organizing map
SRN	simple recurrent network (also referred to as Elman's recurrent network)

SVD	singular value decomposition
SVM	support vector machine
TD	temporal difference
TDNN	time-delay neural network
TLFN	time-lagged feedforward network
VC	Vapnik–Chervononkis (dimension)
VLSI	very-large-scale integration
XOR	exclusive OR

IMPORTANT SYMBOLS

a	action
$\mathbf{a}^T\mathbf{b}$	inner product of vectors **a** and **b**
$\mathbf{a}\mathbf{b}^T$	outer product of vectors **a** and **b**
$\binom{l}{m}$	binomial coefficient
$A \cup B$	unions of A and B
B	inverse of temperature
b_k	bias applied to neuron k
$\cos(\mathbf{a},\mathbf{b})$	cosine of the angle between vectors **a** and **b**
$c_{u,v}(u, v)$	probability density function of copula
D	depth of memory
$D_{f\|\|g}$	Kullback–Leibler divergence between probability density functions f and g
$\tilde{\mathbf{D}}$	adjoint of operator **D**
E	energy function
E_i	energy of state i in statistical mechanics
$\mathbb{E}$	statistical expectation operator
$\langle E \rangle$	average energy
exp	exponential
$\mathscr{E}_{\text{av}}$	average squared error, or sum of squared errors
$\mathscr{E}(n)$	instantaneous value of the sum of squared errors
$\mathscr{E}_{\text{total}}$	total sum of error squares
F	free energy
$\mathscr{F}^*$	subset (network) with minimum empirical risk
$\mathbf{H}$	Hessian (matrix)
$\mathbf{H}^{-1}$	inverse of Hessian **H**
i	square root of -1, also denoted by j
$\mathbf{I}$	identity matrix
$\mathbf{I}$	Fisher's information matrix
J	mean-square error
$\mathbf{J}$	Jacobian (matrix)

$\mathbf{P}^{1/2}$	square root of matrix $\mathbf{P}$
$\mathbf{P}^{T/2}$	transpose of square root of matrix $\mathbf{P}$
$\mathbf{P}n,n-1$	error covariance matrix in Kalman filter theory
k_{B}	Boltzmann constant
log	logarithm
$L(\mathbf{w})$	log-likelihood function of weight vector $\mathbf{w}$
$\mathscr{L}(\mathbf{w})$	log-likelihood function of weight vector $\mathbf{w}$ based on a single example
$\mathbf{M}_{\mathrm{c}}$	controllability matrix
$\mathbf{M}_o$	observability matrix
n	discrete time
p_i	probability of state i in statistical mechanics
p_{ij}	transition probability from state i to state j
$\mathbf{P}$	stochastic matrix
$P(e\|\mathscr{C})$	conditional probability of error e given that the input is drawn from class $\mathscr{C}$
P_{α}^{+}	probability that the visible neurons of a Boltzmann machine are in state α , given that the network is in its clamped condition (i.e., positive phase)
P_{α}^{-}	probability that the visible neurons of a Boltzmann machine are in state α, given that the network is in its free-running condition (i.e., negative phase)
$\hat{r}_x(j, k;n)]$	estimate of autocorrelation function of $\mathrm{x}_j(n)$ and $x_k(n)$
$\hat{r}_{dx}(k;n)$	estimate of cross-correlation function of $d(n)$ and $x_k(n)$
$\mathbf{R}$	correlation matrix of an input vector
t	continuous time
T	temperature
$\mathscr{T}$	training set (sample)
tr	operator denoting the trace of a matrix
var	variance operator
$V(\mathbf{x})$	Lyapunov function of state vector $\mathbf{x}$
v_j	induced local field or activation potential of neuron j
$\mathbf{w}_o$	optimum value of synaptic weight vector
w_{kj}	weight of synapse j belonging to neuron k
$\mathbf{w}^*$	optimum weight vector
$\bar{\mathbf{x}}$	equilibrium value of state vector $\mathbf{x}$
$\langle x_j \rangle$	average of state x_j in a "thermal" sense
$\hat{x}$	estimate of x, signified by the use of a caret (hat)
$\|x\|$	absolute value (magnitude) of x
x^*	complex conjugate of x, signified by asterisk as superscript
$\|\|\mathbf{x}\|\|$	Euclidean norm (length) of vector $\mathbf{x}$
$\mathbf{x}^T$	transpose of vector $\mathbf{x}$, signified by the superscript T
z^{-1}	unit-time delay operator
Z	partition function
$\delta_j(n)$	local gradient of neuron j at time n
Δw	small change applied to weight w
∇	gradient operator

∇^2	Laplacian operator
$\nabla_w J$	gradient of J with respect to w
$\boldsymbol{\nabla} \cdot \mathbf{F}$	divergence of vector $\mathbf{F}$
η	learning-rate parameter
κ	cumulant
μ	policy
θ_k	threshold applied to neuron k (i.e., negative of bias b_k)
λ	regularization parameter
λ_k	kth eigenvalue of a square matrix
$\varphi_k(\cdot)$	nonlinear activation function of neuron k
$\in$	symbol for "belongs to"
$\cup$	symbol for "union of"
$\cap$	symbol for "intersection of"
$*$	symbol for convolution
$+$	superscript symbol for pseudoinverse of a matrix
$+$	superscript symbol for updated estimate

Open and closed intervals

- The open interval (a, b) of a variable x signifies that $a < x < b$.
- The closed interval $[a, b]$ of a variable x signifies that $a \leq \mathrm{x} \leq b$.
- The closed-open interval $[a, b)$ of a variable x signifies that $a \leq x < b$; likewise for the open-closed interval $(a, b]$, $a < x \leq b$.

Minima and Maxima

- The symbol $\arg\min_{\mathbf{w}} f(\mathbf{w})$ signifies the minimum of the function $f(\mathbf{w})$ with respect to the argument vector $\mathbf{w}$.
- The symbol $\arg\max_{\mathbf{w}} f(\mathbf{w})$ signifies the maximum of the function $f(\mathbf{w})$ with respect to the argument vector $\mathbf{w}$.