经典原版书库

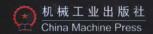# 神经计算原理

（英文版）

Principles of Neurocomputing for Science & Engineering

Fredric M. Ham

Ivica Kostanic

（美） Fredric M. Ham
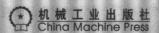Ivica Kostanic 著

Mc Graw Hill Education

神经计算原理

（英文版）

Principles of Neurocomputing for Science & Engineering

（美） Fredric M. Ham
Ivica Kostanic 著

机械工业出版社
China Machine Press

出版者的话

　　文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭橥了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

　　近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短、从业人员较少的现状下，美国等发达国家在其计算机科学发展的几十年间积淀的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

　　机械工业出版社华章图文信息有限公司较早意识到"出版要为教育服务"。自1998年开始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray等大师名家的一批经典作品，以"计算机科学丛书"为总称出版，供读者学习、研究及庋藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

　　"计算机科学丛书"的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专诚为其书的中译本作序。迄今，"计算机科学丛书"已经出版了近百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍，为进一步推广与发展打下了坚实的基础。

　　随着学科建设的初步完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都步入一个新的阶段。为此，华章公司将加大引进教材的力度，在"华章教育"的总规划之下出版三个系列的计算机教材：除"计算机科学丛书"之外，对影印版的教材，则单独开辟出"经典原版书库"；同时，引进全美通行的教学辅导书"Schaum's Outlines"系列组成"全美经典学习指导系列"。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国

家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成"专家指导委员会",为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召,为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.,Stanford,U.C. Berkeley,C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程,而且各具特色——有的出自语言设计者之手、有的历经二十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下,读者必将在计算机科学的宫殿中由登堂而入室。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑,这些因素使我们的图书有了质量的保证,但我们的目标是尽善尽美,而反馈的意见正是我们达到这一终极目标的重要帮助。教材的出版只是我们的后续服务的起点。华章公司欢迎老师和读者对我们的工作提出建议或给予指正,我们的联系方法如下:

电子邮件:hzedu@hzbook.com
联系电话:(010)68995264
联系地址:北京市西城区百万庄南街1号
邮政编码:100037

专家指导委员会

（按姓氏笔画顺序）

To Emily and Matt (FMH)
To Nikola and Radojka (IK)

ABOUT THE AUTHORS

FREDRIC M. HAM, Ph.D., is the Harris Professor of Electrical Engineering at the Florida Institute of Technology, Melbourne. He received his B.S., M.S., and Ph.D. degrees in electrical engineering from Iowa State University in 1976, 1979, and 1980, respectively. Professor Ham has written numerous research papers in the areas of signal processing, biomedical engineering, neural networks, and control systems. He is head of the Information Processing Laboratory at Florida Tech and a consultant to industry. Professor Ham is a Senior Member of the Institute of Electrical and Electronics Engineers (IEEE). His society affiliations in IEEE include control systems, biomedical engineering, signal processing, and neural networks. He is also a member of Eta Kappa Nu, Tau Beta Pi, Phi Kappa Phi, and Sigma Xi.

IVICA KOSTANIC received the B.S. degree in electrical engineering from the University of Belgrade, Yugoslavia, in 1993, and the M.S. degree in electrical engineering from the Florida Institute of Technology in 1995. He is currently pursuing his Ph.D. degree in electrical engineering at the University of Central Florida, Orlando. He is a Senior RF Engineer and a researcher at Agilent Technologies, Inc., Melbourne, Florida. His principal research interests are in the areas of wireless communication, signal processing, and practical applications of neural networks. He is a Member of the Institute of Electrical and Electronics Engineers (IEEE), and he has published several papers in the areas of wireless communication, neural networks, and nonlinear controls.

PREFACE

"One cannot learn anything so well as by experiencing it oneself."

—Albert Einstein

There are four major questions that should be answered in the preface to any textbook: (1) What is the book about? (2) Why publish another book on the particular subject, in this case artificial neural networks (or neural networks)? (3) Who is the intended audience for the book, and what are the prerequisites? (4) What is specifically contained in the book?

Question: What is the book about?

Answer: This textbook is about artificial neural networks (or neural networks). More specifically, the book covers *neurocomputing*. So the question really is, What is neurocomputing? Neurocomputing is concerned with processing information. Unlike its *programmed computing* counterpart, a neurocomputing approach to information processing first involves a *learning* process within an artificial neural network (neural network) architecture. This neural architecture *learns* or adaptively responds to inputs according to a defined learning rule. After the neural network has *learned* what it needs to *know*, the trained network can be used to perform certain tasks depending on the particular application. Neural networks have the ability to learn from their environment and adapt to it in an interactive manner similar to their biological counterparts. Neurocomputing can play an important role in solving certain problems in science and engineering that would otherwise be difficult to solve, problems such as pattern recognition, optimization, event classification, control and identification of nonlinear systems, and statistical analysis, to name a few. Therefore, this book is primarily intended for those individuals who want to understand the underlying principles of artificial neural networks for neurocomputing and want to be able to apply various neurocomputing techniques to solve problems in science and engineering.

Question: Why another book on neural networks?

Answer: The field of neural networks is vast and interdisciplinary. It has drawn interest from researchers from many different areas, and the contributions have been enormous. There are many very good books available in the area of neural networks. However, the authors felt that

there was a need for a book on neural networks that scientists and engineers could specifically identify with, that is, a book for scientists and engineers who want to apply neural networks to solve complex problems. This statement should not be misconstrued to mean this book is a smorgasbord of neural architectures with their associated training algorithms. Instead, this textbook on neural networks presents a variety of neurocomputing approaches that can be used to solve a vast number of problems in science and engineering. In almost every case, a solid mathematical foundation is presented for each neurocomputing concept along with illustrative examples to accompany the particular architecture and associated training algorithm.

Question: Who is the intended audience for the book, and what are the prerequisites?

Answer: This book is primarily for courses offered in neural networks at the graduate level; however, advanced undergraduate students could manage the material in this textbook with the proper background (the second part to this answer). Moreover, practicing engineers and scientists can also learn the material through self-study. The prerequisites for the successful study of neurocomputing by using this textbook are primarily a background in linear algebra and differential equations. It would be desirable to have knowledge in the areas of random variables and stochastic processes but it is not necessary because this material is briefly (though sufficiently) covered in Appendix A.

Question: What is specifically contained in the book?

Answer: The book is divided into two major parts, detailed below. Appendix A covers the mathematical foundation for neurocomputing.

PART I: Fundamental Neurocomputing Concepts and Selected Neural Network Architectures and Learning Rules

Overview: Part I consists of Chapters 1 through 5.

Chapter 1 introduces the reader to the basic idea of neural networks and neurocomputing. Also included is a brief history of neural networks.

Chapter 2 begins with a discussion of basic models of artificial neurons that are the building blocks of neural networks. Next is a

discussion of different types of activation functions followed by a presentation of the adaptive linear element (Adaline) and the multiple Adaline (Madaline). The least mean-square (LMS) algorithm is presented next; then the simple perceptron is detailed followed by a brief discussion of feedforward multilayer perceptrons. Some basic learning rules are covered next. Many of these learning rules are the basis for training more sophisticated neural network architectures. Chapter 2 concludes with an overview of selected data preprocessing methods. All the material in Chapter 2 should be thoroughly covered if the reader is not familiar with artificial neural networks. This chapter sets the stage for the in-depth coverage of selected neural network architectures and associated learning rules presented in Chapters 3 through 5.

Chapter 3 presents a variety of mapping neural networks, beginning with associative memories, followed by backpropagation used to train feedforward multilayer perceptrons. Next, more advanced training methods are given for backpropagation. Counterpropagation is then presented, and the chapter concludes with a presentation of radial basis function neural networks.

Chapter 4 discusses selected self-organizing neural networks. This includes the Kohonen self-organizing map (SOM) and learning vector quantization (LVQ). The chapter concludes with adaptive resonance theory (ART) neural networks; the ART1 network is presented in detail.

Chapter 5 presents recurrent neural networks along with temporal feedforward networks (which are also recurrent networks). The distinction is made between these temporal feedforward neural networks and those that are not multilayer feedforward networks. Included in this chapter are the Hopfield network, simulated annealing, the Boltzmann machine, the simple recurrent network (SRN), time-delay networks, and distributed time-lagged feedforward neural networks.

PART II: Applications of Neurocomputing

Overview: Part II consists of Chapters 6 through 10.

Chapter 6 presents selected neurocomputing approaches for solving constrained optimization problems. Neural networks for linear programming and quadratic programming problems are presented. The chapter concludes with a discussion of neural networks for nonlinear continuous constrained optimization problems. This section includes neural networks for nonlinear

programming penalty and barrier function methods. Also included are neural networks for ordinary and augmented Lagrange multiplier methods.

Chapter 7 covers structured neural network architectures and associated learning rules for solving various matrix algebra problems. A wide variety of important matrix decompositions (or factorizations) are presented along with a neurocomputing solution for each method. Neurocomputing approaches are also given for computing the pseudoinverse of a matrix, solving the algebraic Lyapunov equation, and solving the algebraic Riccati equation.

Chapter 8 covers neurocomputing approaches for solving linear algebraic equations. The various methods include a least-squares neurocomputing approach, a conjugate gradient learning rule, a generalized robust neurocomputing approach, regularization methods for ill-posed problems with ill-determined numerical rank, matrix splittings for iterative discrete-time methods, and the total least-squares problem. Also, L_∞-norm and L_1-norm neural network approaches are presented for solving linear algebraic equations.

Chapter 9 contains many neural network architectures for statistical analysis of data, including neural networks for principal-component analysis (PCA), principal-component regression (PCR), classical least squares (CLS), neural networks for nonlinear PCA and robust PCA, a neural network approach for partial least-squares regression (PLSR), and a neural network approach for robust PLSR.

Chapter 10 covers neural networks for signal processing applications, linear and nonlinear system identification, nonlinear control, and estimation. Many different examples are explained in detail. Also included is independent-component analysis (ICA) using neural networks for blind source separation. In addition, a fast ICA algorithm is presented along with an example that applies the fast ICA algorithm to separation of digital images.

Key Features of the Book

- Most training algorithms are highlighted in the book so they can be easily referenced.
- Some of these training algorithms have an accompanying MATLAB® function implementation given in the text (also highlighted). The code is relatively short and takes only a few minutes to enter into MATLAB.
- In addition, the MATLAB neural network toolbox is used extensively to experimentally illustrate certain neurocomputing concepts.

- Some of the problems have data that reside on the McGraw-Hill Higher Education website for this book and can be easily accessed. The URL for this book is: http://www.mhhe.com/engcs/electrical/ham.
- Detailed examples are presented in most sections to illustrate the neuro-computing concept covered.
- An extensive set of problems is given at the end of each chapter (except Chap. 1). Some problems require the use of MATLAB and the MATLAB neural network toolbox. In some cases the code is provided for the MATLAB function.
- Appendix A contains a comprehensive mathematical foundation for neuro-computing.

F. M. Ham
I. Kostanic

ACKNOWLEDGMENTS

Contact Information for The MathWorks:

LIST OF IMPORTANT SYMBOLS AND OPERATORS

$A > 0$	positive definite A matrix
$A \geq 0$	positive semi-definite (non-negative definite) A matrix
$A < 0$	negative definite A matrix
$A \leq 0$	negative semi-definite (non-positive definite) A matrix
$A_{:k}$	denotes the kth column of the matrix A
$A_{k:}$	denotes the kth row of the matrix A
$\tilde{A}$	complement of the set A
a_{ij}	i (row) j (column) element of the matrix A
arg	argument of a complex quantity
adj	adjoint operator
$\hat{b}_f$	calibration model
β	bias
$\mathscr{C}$	complex numbers
$\mathscr{C}^{n \times m}$	complex $n \times m$ matrices
$\mathscr{C}^{n \times n}$	complex $n \times n$ (square) matrices
$\mathscr{C}^{n \times 1}$	complex n-dimensional column vector
$\mathscr{C}^{1 \times n}$	complex n-dimensional row vector
$\mathscr{C}^{n}$	complex n-dimensional vector (row or column)
$(\mathscr{C}^{n}, \mathscr{C})$	n-dimensional complex vector space
C_X	autocovariance function
$\boldsymbol{C}_X$	covariance matrix
cof	cofactor operator
$\text{cond}_p(A)$	condition number of the matrix A
$\Delta\mathscr{E}$	energy change
diag	square matrix operator that selects the diagonal elements
δ_{ij}	Kronecker delta
det	determinant
exp	exponential
E	expectation operator
$\mathscr{E}$	energy function
E	energy function
e	exponential
$\mathscr{F}$	field
f_{bs}	binary sigmoid activation function
f_{lin}	linear activation function
f_L	logistic function

f_{hl}	hard limiter activation function
f_H	Huber's function
f_{hts}	hyperbolic tangent sigmoid activation function
f_M	M-estimator function
f_T	Talwar's function
f_s	sampling frequency, Hz
f_{shl}	symmetric hard limiter activation function
f_{sl}	saturating linear activation function
f_{ssl}	symmetric saturating linear activation function
g_{bs}	derivative of the binary sigmoid activation function
g_L	derivative of the logistic function
g_{hts}	derivative of the hyperbolic tangent sigmoid activation function
g_H	derivative of Huber's function
g_M	derivative of the M-estimator function
g_T	derivative of Talwar's function
$g \circ f$	composition of g with f
γ	forgetting factor or leakage factor
$\boldsymbol{H}$	Hessian matrix
h^O	optimal number of factors
$\boldsymbol{I}_n$	$n \times n$ identity matrix
$\boldsymbol{I}$	identity matrix of appropriate dimension
inf	infimum
J	cost function
$\boldsymbol{J}$	Jacobian matrix
k	discrete time index
kurt	kurtosis operator
L	Lyapunov function
LT	lower triangular matrix operator
$\mathscr{L}$	Lagrangian function, Laplace transform operator, or successive relaxation matrix
ℓ	learning signal
λ	eigenvalue or Lagrange multiplier
m_x	mean of a stochastic process
$\boldsymbol{m}_x$	mean vector of a vector stochastic process
MIN	fuzzy intersection
MAX	fuzzy union
min	minimum value operator
max	maximum value operator
μ	learning rate parameter
ν	nullity

Ω	nonlinear mapping
$O(\gamma^2)$	higher-order effects in γ
ω_S	sampling frequency, $\frac{rad}{sec}$
Pr	probability
$\not{p}_X$	probability density function
$\mathscr{P}_X$	probability distribution function
Ψ	loss function
Φ	regression matrix
ϕ	state transition matrix
ϕ_h	prototype memory
$\mathfrak{R}$	real numbers
$\mathfrak{R}^{n \times m}$	real $n \times m$ matrices
$\mathfrak{R}^{n \times n}$	real $n \times n$ (square) matrices
$\mathfrak{R}^{n \times 1}$	real n-dimensional column vector
$\mathfrak{R}^{1 \times n}$	real n-dimensional row vector
$\mathfrak{R}^n$	real n-dimensional vector (row or column)
$(\mathfrak{R}^n, \mathfrak{R})$	n-dimensional real vector space
$(\mathfrak{R}^n(s), \mathfrak{R}(s))$	n-dimensional rational vector space
$\mathscr{R}$	Rayleigh quotient
R_X	correlation matrix
R_x	autocorrelation function
$\mathscr{R}_x$	time autocorrelation function
ρ	matrix rank or vigilance parameter
ρ_X	correlation coefficient
S_x	power spectral density matrix
$\mathscr{S}$	probability space or sample space
sup	supremum
sgn	signum function
σ	standard deviation, singular value, spread parameter, or regularization parameter
σ^2	variance
σ_h	fixed stable point
t	continuous time
tr	trace of a matrix
trace	trace of a matrix
T	temperature
T_{binary}	threshold logic operator
T_s	sampling period
Tr	sum over all possible configurations
θ	threshold
θ	parameter vector
$V(x)$	Lyapunov function of the state vector x
var	variance operator