

Neural Networks and Statistical Learning

Ke-Lin Du · M. N. S. Swamy

Neural Networks and Statistical Learning

Second Edition

 Springer

Ke-Lin Du
Department of Electrical
and Computer Engineering
Concordia University
Montreal, QC, Canada

Xonlink Inc.
Hangzhou, China

M. N. S. Swamy
Department of Electrical
and Computer Engineering
Concordia University
Montreal, QC, Canada

ISBN 978-1-4471-7451-6 ISBN 978-1-4471-7452-3 (eBook)
<https://doi.org/10.1007/978-1-4471-7452-3>

1st edition: © Springer-Verlag London 2014

2nd edition: © Springer-Verlag London Ltd., part of Springer Nature 2019

The author(s) has/have asserted their right(s) to be identified as the author(s) of this work in accordance with the Copyright, Designs and Patents Act 1988.

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer-Verlag London Ltd. part of Springer Nature.

The registered company address is: The Campus, 4 Crinan Street, London, N1 9XW, United Kingdom

To Falong Xing and Jie Zeng
—Ke-Lin Du

To my teachers and my students
—M. N. S. Swamy

Preface to the Second Edition

Since the publication of the first edition in December 2013, the rapid rise of deep learning and AI has resulted in a wave of research activities and numerous new results. During the past few years, there have been several breakthroughs in deep learning and AI. At the same time, research and application of big data are widespread. Machine learning has become the brains behind big data.

In such a setting, this book has become one of the best sellers of Springer books. Under the suggestion of Anthony Doyle at Springer London Ltd., we decided to publish this second edition.

In this second edition, we will add six new chapters to the first edition:

- Chapter 3 focuses on computation learning theory. Part of its content is split from Chap. 2 of the first edition.
- Chapter 18 introduces compressed sensing and sparse coding. In this approach, a datum is represented as a linear combination of basis functions, and the coefficients are assumed to be sparse.
- Chapter 19 deals with matrix completion. Recovery of a data matrix from a subset of its entries is an extension of compressed sensing and sparse approximation.
- Chapter 23 introduces the Boltzmann machine. Part of its content is split from Chap. 19 of the first edition.
- Chapter 24 describes deep learning and deep neural networks. Deep learning is the state-of-the-art approach to solving complex problems.
- Chapter 31 introduces big data, cloud computing, and Internet of Things. These topics go hand in hand. Machine learning functions as the major tool for data analytics.

We also update each chapter by including major contributions published in the past 6 years.

The authors wish to thank Jie Zeng (Richcon MC, Inc., China), Biaobiao Zhang (Yuantu Internet Technology Ltd., China), Li Yu (Zhejiang University of Technology, China), Zhijiang Xu (Zhejiang University of Technology, China), and Renwang Li (Zhejiang Sci-Tech University, China) for their help during the preparation of this second edition.

Hangzhou, China/Montreal, Canada
Montreal, Canada
May 2019

Ke-Lin Du
M. N. S. Swamy

Preface to the First Edition

The human brain, consisting of nearly 10^{11} neurons, is the center of human intelligence. Human intelligence has been simulated in various ways. Artificial intelligence (AI) pursues exact logical reasoning based on symbol manipulation. Fuzzy logics model the highly uncertain behavior of decision-making. Neural networks model the highly nonlinear infrastructure of brain networks. Evolutionary computation models the evolution of intelligence. Chaos theory models the highly nonlinear and chaotic behaviors of human intelligence.

Soft computing is an evolving collection of methodologies for the representation of ambiguity in human thinking; it exploits the tolerance for imprecision and uncertainty, approximate reasoning, and partial truth in order to achieve tractability, robustness, and low-cost solutions. The major methodologies of soft computing are fuzzy logic, neural networks, and evolutionary computation.

Conventional model-based data processing methods require experts' knowledge for the modeling of a system. Neural network methods provide a model-free, adaptive, fault-tolerant, parallel, and distributed processing solution. A neural network is a black box that directly learns the internal relations of an unknown system, without guessing functions for describing cause-and-effect relationships. The neural network approach is a basic methodology of information processing. Neural network models may be used for function approximation, classification, nonlinear mapping, associative memory, vector quantization, optimization, feature extraction, clustering, and approximate inference. Neural networks have wide applications in almost all areas of science and engineering.

Fuzzy logic provides a means for treating uncertainty and computing with words. This mimics human recognition, which skillfully copes with uncertainty. Fuzzy systems are conventionally created from explicit knowledge expressed in the form of fuzzy rules, which are designed based on experts' experience. A fuzzy system can explain its action by fuzzy rules. Neurofuzzy systems, as a synergy of fuzzy logic and neural networks, possess both learning and knowledge representation capabilities.

This book is our attempt to bring together the major advances in neural networks and machine learning, and to explain them in a statistical framework. While some mathematical details are needed, we emphasize the practical aspects of the models and methods rather than the theoretical details. To us, neural networks are merely some statistical methods that can be represented by graphs and networks. They can iteratively adjust the network parameters. As a statistical model, a neural network can learn the probability density function from the given samples, and then predict, by generalization according to the learnt statistics, outputs for new samples that are not included in the learning sample set.

The neural network approach is a general statistical computational paradigm. Neural network research solves two problems: the direct problem and the inverse problem. The direct problem employs computer and engineering techniques to model biological neural systems of the human brain. This problem is investigated by cognitive scientists and can be useful in neuropsychiatry and neurophysiology. The inverse problem simulates biological neural systems for their problem-solving capabilities for application in scientific or engineering fields. Engineering and computer scientists have conducted an extensive investigation in this area. This book concentrates mainly on the inverse problem, although the two areas often shed light on each other. The biological and psychological plausibility of the neural network models have not been seriously treated in this book, though some background material is discussed.

This book is intended to be used as a textbook for advanced undergraduate and graduate students in engineering, science, computer science, business, arts, and medicine. It is also a good reference book for scientists, researchers, and practitioners in a wide variety of fields, and assumes no previous knowledge of neural network or machine learning concepts.

This book is divided into 25 chapters and 2 appendices. It contains almost all the major neural network models and statistical learning approaches. We also give an introduction to fuzzy sets and logic, and neurofuzzy models. Hardware implementations of the models are discussed. Two chapters are dedicated to the applications of neural network and statistical learning approaches to biometrics/bioinformatics and data mining. Finally, in the appendices, some mathematical preliminaries are given, and benchmarks for validating all kinds of neural network methods and some web resources are provided.

First and foremost, we would like to thank the supporting staff from Springer London, especially Anthony Doyle and Grace Quinn for their enthusiastic and professional support throughout the period of manuscript preparation.

K.-L. Du also wishes to thank Jiabin Lu (Guangdong University of Technology, China), Jie Zeng (Richcon MC, Inc., China), Biaobiao Zhang and Hui Wang (Enjoyor, Inc., China), and many of his graduate students, including Na Shou, Shengfeng Yu, Lusha Han, Xiaolan Shen, Yuanyuan Chen, and Xiaoling Wang (Zhejiang University of Technology, China) for their consistent assistance.

In addition, we should mention at least the following names for their help: Omer Morgul (Bilkent University, Turkey), Yanwu Zhang (Monterey Bay Aquarium Research Institute, USA), Chi Sing Leung (City University of Hong Kong,

Hong Kong), M. Omair Ahmad and Jianfeng Gu (Concordia University, Canada), Li Yu, Limin Meng, Jingyu Hua, Zhijiang Xu, and Luping Fang (Zhejiang University of Technology, China), Yuxing Dai (Wenzhou University, China), and Renwang Li (Zhejiang Sci-Tech University, China). Last, but not least, we would like to thank our families for their support and understanding during the course of writing this book.

A book of this length is certain to have some errors and omissions. Feedback is welcome via email at kldu@ieee.org or swamy@encs.concordia.ca. Due to restriction on the length of this book, we have placed two appendices, namely, Mathematical preliminaries, and Benchmarks and resources, on the website of this book. MATLAB code for the worked examples is also downloadable from the website of this book.

Hangzhou, China
Montreal, Canada
April 2013

Ke-Lin Du
M. N. S. Swamy

Contents

1	Introduction	1
1.1	Major Events in Machine Learning Research	1
1.2	Neurons	4
1.2.1	McCulloch–Pitts Neuron Model	5
1.2.2	Spiking Neuron Models	7
1.3	Neural Networks	9
1.4	Neural Network Processors	13
1.5	Scope of the Book	16
	References	17
2	Fundamentals of Machine Learning	21
2.1	Learning and Inference Methods	21
2.1.1	Scientific Reasoning	22
2.1.2	Supervised, Unsupervised, and Reinforcement Learnings	24
2.1.3	Semi-supervised Learning and Active Learning	27
2.1.4	Other Learning Methods	28
2.2	Learning and Generalization	33
2.2.1	Generalization Error	34
2.2.2	Generalization by Stopping Criterion	35
2.2.3	Generalization by Regularization	36
2.2.4	Dropout	37
2.2.5	Fault Tolerance and Generalization	39
2.2.6	Sparsity Versus Stability	40
2.3	Model Selection	40
2.3.1	Cross-Validation	41
2.3.2	Complexity Criteria	43
2.4	Bias and Variance	45
2.5	Criterion Functions	47
2.6	Robust Learning	49

2.7	Neural Networks as Universal Machines	51
2.7.1	Boolean Function Approximation	51
2.7.2	Linear Separability and Nonlinear Separability	53
2.7.3	Continuous Function Approximation	55
2.7.4	Winner-Takes-All	56
	References	58
3	Elements of Computational Learning Theory	65
3.1	Introduction	65
3.2	Probably Approximately Correct (PAC) Learning	66
3.2.1	Sample Complexity	67
3.3	Vapnik–Chervonenkis Dimension	68
3.3.1	Teaching Dimension	70
3.4	Rademacher Complexity	70
3.5	Empirical Risk-Minimization Principle	72
3.5.1	Function Approximation, Regularization, and Risk Minimization	74
3.6	Fundamental Theorem of Learning Theory	75
3.7	No-Free-Lunch Theorem	76
	References	77
4	Perceptrons	81
4.1	One-Neuron Perceptron	81
4.2	Single-Layer Perceptron	82
4.3	Perceptron Learning Algorithm	83
4.4	Least Mean Squares (LMS) Algorithm	85
4.5	P-Delta Rule	88
4.6	Other Learning Algorithms	89
	References	93
5	Multilayer Perceptrons: Architecture and Error Backpropagation	97
5.1	Introduction	97
5.2	Universal Approximation	98
5.3	Backpropagation Learning Algorithm	99
5.4	Incremental Learning Versus Batch Learning	104
5.5	Activation Functions for the Output Layer	109
5.6	Optimizing Network Structure	110
5.6.1	Network Pruning Using Sensitivity Analysis	110
5.6.2	Network Pruning Using Regularization	113
5.6.3	Network Growing	115
5.7	Speeding Up Learning Process	117
5.7.1	Eliminating Premature Saturation	117
5.7.2	Adapting Learning Parameters	119

- 5.7.3 Initializing Weights 123
- 5.7.4 Adapting Activation Function 124
- 5.8 Some Improved BP Algorithms 127
 - 5.8.1 BP with Global Descent 128
 - 5.8.2 Robust BP Algorithms 129
- 5.9 Resilient Propagation (Rprop) 130
- 5.10 Spiking Neural Network Learning 132
- References 135

- 6 Multilayer Perceptrons: Other Learning Techniques 143**
 - 6.1 Introduction to Second-Order Learning Methods 143
 - 6.2 Newton’s Methods 144
 - 6.2.1 Gauss–Newton Method 145
 - 6.2.2 Levenberg–Marquardt Method 146
 - 6.3 Quasi-Newton Methods 149
 - 6.3.1 BFGS Method 150
 - 6.3.2 One-Step Secant Method 152
 - 6.4 Conjugate Gradient Methods 152
 - 6.5 Extended Kalman Filtering Methods 157
 - 6.6 Recursive Least Squares 159
 - 6.7 Natural-Gradient-Descent Method 160
 - 6.8 Other Learning Algorithms 161
 - 6.8.1 Layerwise Linear Learning 161
 - 6.9 Escaping Local Minima 162
 - 6.10 Complex-Valued MLPs and Their Learning 163
 - 6.10.1 Split Complex BP 164
 - 6.10.2 Fully Complex BP 164
 - References 168

- 7 Hopfield Networks, Simulated Annealing, and Chaotic Neural Networks 173**
 - 7.1 Hopfield Model 173
 - 7.2 Continuous-Time Hopfield Network 176
 - 7.3 Simulated Annealing 179
 - 7.4 Hopfield Networks for Optimization 182
 - 7.4.1 Combinatorial Optimization Problems 183
 - 7.4.2 Escaping Local Minima 187
 - 7.4.3 Solving Other Optimization Problems 188
 - 7.5 Chaos and Chaotic Neural Networks 189
 - 7.5.1 Chaos, Bifurcation, and Fractals 189
 - 7.5.2 Chaotic Neural Networks 190
 - 7.6 Multistate Hopfield Networks 193
 - 7.7 Cellular Neural Networks 194
 - References 197

8	Associative Memory Networks	201
8.1	Introduction	201
8.2	Hopfield Model: Storage and Retrieval	203
8.2.1	Generalized Hebbian Rule	203
8.2.2	Pseudoinverse Rule	205
8.2.3	Perceptron-Type Learning Rule	205
8.2.4	Retrieval Stage	206
8.3	Storage Capability of Hopfield Model	207
8.4	Increasing Storage Capacity	212
8.5	Multistate Hopfield Networks as Associative Memories	214
8.6	Multilayer Perceptrons as Associative Memories	215
8.7	Hamming Network	217
8.8	Bidirectional Associative Memories	219
8.9	Cohen–Grossberg Model	220
8.10	Cellular Networks as Associative Memories	221
	References	226
9	Clustering I: Basic Clustering Models and Algorithms	231
9.1	Vector Quantization	231
9.2	Competitive Learning	232
9.3	Self-Organizing Maps	234
9.3.1	Kohonen Network	235
9.3.2	Basic Self-Organizing Maps	236
9.4	Learning Vector Quantization	244
9.5	Nearest Neighbor Algorithms	246
9.6	Neural Gas	249
9.7	ART Networks	252
9.7.1	ART Models	253
9.7.2	ART 1	254
9.8	<i>C</i> -Means Clustering	256
9.9	Subtractive Clustering	259
9.10	Fuzzy Clustering	262
9.10.1	Fuzzy <i>C</i> -Means Clustering	262
9.10.2	Other Fuzzy Clustering Algorithms	265
	References	269
10	Clustering II: Topics in Clustering	275
10.1	Underutilization Problem	275
10.1.1	Competitive Learning with Conscience	275
10.1.2	Rival Penalized Competitive Learning	277
10.1.3	Soft-Competitive Learning	279
10.2	Robust Clustering	280
10.2.1	Possibilistic <i>C</i> -Means	282
10.2.2	A Unified Framework for Robust Clustering	283

- 10.3 Supervised Clustering 284
- 10.4 Clustering Using Non-Euclidean Distance Measures 285
- 10.5 Partitional, Hierarchical, and Density-Based Clustering 287
- 10.6 Hierarchical Clustering 288
 - 10.6.1 Distance Measures, Cluster Representations, and Dendrograms 288
 - 10.6.2 Minimum Spanning Tree (MST) Clustering 290
 - 10.6.3 BIRCH, CURE, CHAMELEON, and DBSCAN 292
 - 10.6.4 Hybrid Hierarchical/Partitional Clustering 295
- 10.7 Constructive Clustering Techniques 296
- 10.8 Cluster Validity 298
 - 10.8.1 Measures Based on Compactness and Separation of Clusters 299
 - 10.8.2 Measures Based on Hypervolume and Density of Clusters 300
 - 10.8.3 Crisp Silhouette and Fuzzy Silhouette 301
- 10.9 Projected Clustering 303
- 10.10 Spectral Clustering 304
- 10.11 Coclustering 305
- 10.12 Handling Qualitative Data 306
- 10.13 Bibliographical Notes 307
- References 308
- 11 Radial Basis Function Networks 315**
 - 11.1 Introduction 315
 - 11.2 RBF Network Architecture 317
 - 11.3 Universal Approximation of RBF Networks 318
 - 11.4 Formulation for RBF Network Learning 319
 - 11.5 Radial Basis Functions 320
 - 11.6 Learning RBF Centers 323
 - 11.7 Learning the Weights 325
 - 11.7.1 Least Squares Methods for Weights Learning 325
 - 11.8 RBF Network Learning Using Orthogonal Least Squares 327
 - 11.9 Supervised Learning of All Parameters 329
 - 11.9.1 Supervised Learning for General RBF Networks 329
 - 11.9.2 Supervised Learning for Gaussian RBF Networks 330
 - 11.9.3 Discussion on Supervised Learning 331
 - 11.10 Various Learning Methods 332
 - 11.11 Normalized RBF Networks 334
 - 11.12 Optimizing Network Structure 335
 - 11.12.1 Constructive Methods 335
 - 11.12.2 Resource-Allocating Networks 337
 - 11.12.3 Pruning Methods 339

- 11.13 Complex RBF Networks 339
- 11.14 A Comparison of RBF Networks and MLPs 341
- References 345
- 12 Recurrent Neural Networks 351**
 - 12.1 Introduction 351
 - 12.2 Fully Connected Recurrent Networks 353
 - 12.3 Time-Delay Neural Networks 354
 - 12.4 Backpropagation for Temporal Learning 357
 - 12.5 RBF Networks for Modeling Dynamic Systems 360
 - 12.6 Some Recurrent Models 360
 - 12.7 Reservoir Computing 362
 - References 368
- 13 Principal Component Analysis 373**
 - 13.1 Introduction 373
 - 13.1.1 Hebbian Learning Rule 374
 - 13.1.2 Oja’s Learning Rule 375
 - 13.2 PCA: Conception and Model 376
 - 13.3 Hebbian Rule-Based PCA 379
 - 13.3.1 Subspace Learning Algorithms 379
 - 13.3.2 Generalized Hebbian Algorithm 383
 - 13.4 Least Mean Squared Error-Based PCA 385
 - 13.4.1 Other Optimization-Based PCA 389
 - 13.5 Anti-Hebbian Rule-Based PCA 390
 - 13.5.1 APEX Algorithm 391
 - 13.6 Nonlinear PCA 395
 - 13.6.1 Autoassociative Network-Based Nonlinear PCA 396
 - 13.7 Minor Component Analysis 398
 - 13.7.1 Extracting the First Minor Component 398
 - 13.7.2 Self-Stabilizing Minor Component Analysis 399
 - 13.7.3 Oja-Based MCA 400
 - 13.7.4 Other Algorithms 400
 - 13.8 Constrained PCA 401
 - 13.8.1 Sparse PCA 402
 - 13.9 Localized PCA, Incremental PCA, and Supervised PCA 403
 - 13.10 Complex-Valued PCA 405
 - 13.11 Two-Dimensional PCA 406
 - 13.12 Generalized Eigenvalue Decomposition 407
 - 13.13 Singular Value Decomposition 409
 - 13.13.1 Cross-Correlation Asymmetric PCA Networks 409
 - 13.13.2 Extracting Principal Singular Components
for Nonsquare Matrices 412
 - 13.13.3 Extracting Multiple Principal Singular
Components 413

- 13.14 Factor Analysis 414
- 13.15 Canonical Correlation Analysis 415
- References 418
- 14 Nonnegative Matrix Factorization 427**
 - 14.1 Introduction 427
 - 14.2 Algorithms for NMF 429
 - 14.2.1 Multiplicative Update Algorithm and Alternating
Nonnegative Least Squares 429
 - 14.3 Other NMF Methods 432
 - 14.3.1 NMF Methods for Clustering 435
 - 14.3.2 Concept Factorization 437
 - 14.4 Nystrom Method 438
 - 14.5 CUR Decomposition 440
 - References 441
- 15 Independent Component Analysis 447**
 - 15.1 Introduction 447
 - 15.2 ICA Model 448
 - 15.3 Approaches to ICA 449
 - 15.4 Popular ICA Algorithms 451
 - 15.4.1 Infomax ICA 451
 - 15.4.2 EASI, JADE, and Natural Gradient ICA 453
 - 15.4.3 FastICA Algorithm 454
 - 15.5 ICA Networks 459
 - 15.6 Some BSS Methods 462
 - 15.6.1 Nonlinear ICA 462
 - 15.6.2 Constrained ICA 462
 - 15.6.3 Nonnegativity ICA 463
 - 15.6.4 ICA for Convolutional Mixtures 464
 - 15.6.5 Other BSS/ICA Methods 465
 - 15.7 Complex-Valued ICA 468
 - 15.8 Source Separation for Time Series 470
 - 15.9 EEG, MEG, and fMRI 472
 - References 476
- 16 Discriminant Analysis 483**
 - 16.1 Linear Discriminant Analysis 483
 - 16.2 Solving Small Sample Size Problem 487
 - 16.3 Fisherfaces 487
 - 16.4 Regularized LDA 488
 - 16.5 Uncorrelated LDA and Orthogonal LDA 490
 - 16.6 LDA/GSVD and LDA/QR 491
 - 16.7 Incremental LDA 492

16.8	Other Discriminant Methods	493
16.9	Nonlinear Discriminant Analysis	495
16.10	Two-Dimensional Discriminant Analysis	497
	References	498
17	Reinforcement Learning	503
17.1	Introduction	503
17.2	Learning Through Awards	505
17.3	Actor–Critic Model	507
17.4	Model-Free and Model-Based Reinforcement Learning	509
17.5	Learning from Demonstrations	512
17.6	Temporal-Difference Learning	513
	17.6.1 TD(λ)	514
	17.6.2 Sarsa(λ)	515
17.7	Q -Learning	516
17.8	Multiagent Reinforcement Learning	518
	17.8.1 Equilibrium-Based Multiagent Reinforcement Learning	519
	17.8.2 Learning Automata	520
	References	521
18	Compressed Sensing and Dictionary Learning	525
18.1	Introduction	525
18.2	Compressed Sensing	526
	18.2.1 Restricted Isometry Property	527
	18.2.2 Sparse Recovery	528
	18.2.3 Iterative Hard Thresholding	530
	18.2.4 Orthogonal Matching Pursuit	532
	18.2.5 Restricted Isometry Property for Signal Recovery Methods	533
	18.2.6 Tensor Compressive Sensing	535
18.3	Sparse Coding and Dictionary Learning	535
18.4	LASSO	538
18.5	Other Sparse Algorithms	540
	References	541
19	Matrix Completion	549
19.1	Introduction	549
19.2	Matrix Completion	550
	19.2.1 Minimizing the Nuclear Norm	551
	19.2.2 Matrix Factorization-Based Methods	553
	19.2.3 Theoretical Guarantees on Exact Matrix Completion	554
	19.2.4 Discrete Matrix Completion	556

- 19.3 Low-Rank Representation 557
- 19.4 Tensor Factorization and Tensor Completion 558
 - 19.4.1 Tensor Factorization 560
 - 19.4.2 Tensor Completion 561
- References 563
- 20 Kernel Methods 569**
 - 20.1 Introduction 569
 - 20.2 Kernel Functions and Representer Theorem 570
 - 20.3 Kernel PCA 572
 - 20.4 Kernel LDA 576
 - 20.5 Kernel Clustering 578
 - 20.6 Kernel Auto-associators, Kernel CCA, and Kernel ICA 579
 - 20.7 Other Kernel Methods 581
 - 20.7.1 Random Kitchen Sinks and Fastfood 583
 - 20.8 Multiple Kernel Learning 583
 - References 586
- 21 Support Vector Machines 593**
 - 21.1 Introduction 593
 - 21.2 SVM Model 594
 - 21.2.1 SVM Versus Neural Networks 597
 - 21.3 Solving the Quadratic Programming Problem 597
 - 21.3.1 Chunking 599
 - 21.3.2 Decomposition 599
 - 21.3.3 Convergence of Decomposition Methods 603
 - 21.4 Least Squares SVMs 603
 - 21.5 SVM Training Methods 606
 - 21.5.1 SVM Algorithms with Reduced Kernel Matrix 606
 - 21.5.2 ν -SVM 608
 - 21.5.3 Cutting-Plane Technique 609
 - 21.5.4 Gradient-Based Methods 610
 - 21.5.5 Training SVM in the Primal Formulation 610
 - 21.5.6 Clustering-Based SVM 612
 - 21.5.7 Other SVM Methods 613
 - 21.6 Pruning SVMs 615
 - 21.7 Multiclass SVMs 617
 - 21.8 Support Vector Regression 619
 - 21.8.1 Solving Support Vector Regression 621
 - 21.9 Support Vector Clustering 624
 - 21.10 SVMs for One-Class Classification 627
 - 21.11 Incremental SVMs 628
 - 21.12 SVMs for Active, Transductive, and Semi-supervised Learnings 630

21.12.1	SVMs for Active Learning	630
21.12.2	SVMs for Transductive or Semi-supervised Learning	630
21.13	Solving SVM with Indefinite Matrices	633
	References	635
22	Probabilistic and Bayesian Networks	645
22.1	Introduction	645
22.1.1	Classical Versus Bayesian Approach	646
22.1.2	Bayes' Theorem and Bayesian Classifiers	647
22.1.3	Graphical Models	648
22.2	Bayesian Network Model	649
22.3	Learning Bayesian Networks	652
22.3.1	Learning the Structure	653
22.3.2	Learning the Parameters	657
22.3.3	Constraint-Handling	659
22.4	Bayesian Network Inference	660
22.4.1	Belief Propagation	660
22.4.2	Factor Graphs and Belief Propagation Algorithm	663
22.5	Sampling (Monte Carlo) Methods	666
22.5.1	Gibbs Sampling	667
22.5.2	Importance Sampling	669
22.5.3	Particle Filtering	669
22.6	Variational Bayesian Methods	670
22.7	Hidden Markov Models	672
22.8	Dynamic Bayesian Networks	675
22.9	Expectation–Maximization Method	676
22.10	Mixture Models	678
22.11	Bayesian and Probabilistic Approach to Machine Learning	679
22.11.1	Probabilistic PCA	681
22.11.2	Probabilistic Clustering	682
22.11.3	Probabilistic ICA	683
22.11.4	Probabilistic Approach to SVM	685
22.11.5	Relevance Vector Machines	685
	References	689
23	Boltzmann Machines	699
23.1	Boltzmann Machines	699
23.1.1	Boltzmann Learning Algorithm	701
23.2	Restricted Boltzmann Machines	703
23.2.1	Universal Approximation	705
23.2.2	Contrastive Divergence Algorithm	706
23.2.3	Related Methods	708

- 23.3 Mean-Field-Theory Machine 709
- 23.4 Stochastic Hopfield Networks 711
- References 712
- 24 Deep Learning 717**
 - 24.1 Introduction 717
 - 24.2 Deep Neural Networks 719
 - 24.2.1 Deep Networks Versus Shallow Networks 720
 - 24.3 Deep Belief Networks 721
 - 24.3.1 Training Deep Belief Networks 722
 - 24.4 Deep Autoencoders 723
 - 24.5 Deep Convolutional Neural Networks 724
 - 24.5.1 Solving the Difficulties of Gradient Descent 725
 - 24.5.2 Implementing Deep Convolutional Neural Networks 726
 - 24.6 Deep Reinforcement Learning 729
 - 24.7 Other Deep Neural Network Methods 730
 - References 732
- 25 Combining Multiple Learners: Data Fusion and Ensemble Learning 737**
 - 25.1 Introduction 737
 - 25.1.1 Ensemble Learning Methods 738
 - 25.1.2 Aggregation 739
 - 25.2 Majority Voting 740
 - 25.3 Bagging 741
 - 25.4 Boosting 743
 - 25.4.1 AdaBoost 744
 - 25.4.2 Other Boosting Algorithms 746
 - 25.5 Random Forests 748
 - 25.5.1 AdaBoost Versus Random Forests 750
 - 25.6 Topics in Ensemble Learning 751
 - 25.6.1 Ensemble Neural Networks 751
 - 25.6.2 Diversity Versus Ensemble Accuracy 752
 - 25.6.3 Theoretical Analysis 753
 - 25.6.4 Ensembles for Streams 753
 - 25.7 Solving Multiclass Classification 754
 - 25.7.1 One-Against-All Strategy 754
 - 25.7.2 One-Against-One Strategy 755
 - 25.7.3 Error-Correcting Output Codes (ECOCs) 756
 - 25.8 Dempster–Shafer Theory of Evidence 758
 - References 762

26	Introduction to Fuzzy Sets and Logic	769
26.1	Introduction	769
26.2	Definitions and Terminologies	770
26.3	Membership Function	776
26.4	Intersection, Union and Negation	777
26.5	Fuzzy Relation and Aggregation	779
26.6	Fuzzy Implication	781
26.7	Reasoning and Fuzzy Reasoning	782
	26.7.1 <i>Modus Ponens</i> and <i>Modus Tollens</i>	783
	26.7.2 Generalized <i>Modus Ponens</i>	784
	26.7.3 Fuzzy Reasoning Methods	785
26.8	Fuzzy Inference Systems	786
	26.8.1 Fuzzy Rules and Fuzzy Interference	787
	26.8.2 Fuzzification and Defuzzification	788
26.9	Fuzzy Models	789
	26.9.1 Mamdani Model	789
	26.9.2 Takagi–Sugeno–Kang Model	790
26.10	Complex Fuzzy Logic	792
26.11	Possibility Theory	793
26.12	Case-Based Reasoning	795
26.13	Granular Computing and Ontology	795
	References	799
27	Neurofuzzy Systems	803
27.1	Introduction	803
	27.1.1 Interpretability	804
27.2	Rule Extraction from Trained Neural Networks	805
	27.2.1 Fuzzy Rules and Multilayer Perceptrons	805
	27.2.2 Fuzzy Rules and RBF Networks	806
	27.2.3 Rule Extraction from SVMs	807
	27.2.4 Rule Generation from Other Neural Networks	808
27.3	Extracting Rules from Numerical Data	809
	27.3.1 Rule Generation Based on Fuzzy Partitioning	809
	27.3.2 Other Methods	811
27.4	Synergy of Fuzzy Logic and Neural Networks	812
27.5	ANFIS Model	813
27.6	Generic Fuzzy Perceptron	819
27.7	Fuzzy SVMs	821
27.8	Other Neurofuzzy Models	822
	References	825

- 28 Neural Network Circuits and Parallel Implementations** 829
 - 28.1 Introduction 829
 - 28.2 Hardware/Software Codesign 831
 - 28.3 Topics in Digital Circuit Designs 832
 - 28.4 Circuits for Neural Networks 833
 - 28.4.1 Memristor 833
 - 28.4.2 Circuits for MLPs 835
 - 28.4.3 Circuits for RBF Networks 836
 - 28.4.4 Circuits for Clustering 837
 - 28.4.5 Circuits for SVMs 837
 - 28.4.6 Circuits for Other Neural Network Models 838
 - 28.4.7 Circuits for Fuzzy Neural Models 839
 - 28.5 Graphic Processing Unit (GPU) Implementation 840
 - 28.6 Implementation Using Systolic Algorithms 842
 - 28.7 Implementation on Parallel Computers 843
 - 28.7.1 Distributed and Parallel SVMs 845
 - References 846
- 29 Pattern Recognition for Biometrics and Bioinformatics** 853
 - 29.1 Biometrics 853
 - 29.1.1 Physiological Biometrics and Recognition 854
 - 29.1.2 Behavioral Biometrics and Recognition 857
 - 29.2 Face Detection and Recognition 858
 - 29.2.1 Face Detection 859
 - 29.2.2 Face Recognition 860
 - 29.3 Bioinformatics 862
 - 29.3.1 Microarray Technology 864
 - 29.3.2 Motif Discovery, Sequence Alignment, Protein Folding, and Coclustering 867
 - References 869
- 30 Data Mining** 871
 - 30.1 Introduction 871
 - 30.2 Document Representations for Text Categorization 872
 - 30.3 Neural Network Approach to Data Mining 874
 - 30.3.1 Classification-Based Data Mining 874
 - 30.3.2 Clustering-Based Data Mining 875
 - 30.3.3 Bayesian Network-Based Data Mining 878
 - 30.4 XML Format 879
 - 30.5 Association Mining 881
 - 30.5.1 Affective Computing 881
 - 30.6 Web Usage Mining 882
 - 30.7 Ranking Search Results 883
 - 30.7.1 Surfer Models 884

- 30.7.2 PageRank Algorithm 885
- 30.7.3 Hypertext-Induced Topic Search (HITS) 888
- 30.8 Personalized Search 889
- 30.9 Data Warehousing 891
- 30.10 Content-Based Image Retrieval 893
- 30.11 E-mail Anti-spamming 896
- References 897
- 31 Big Data, Cloud Computing, and Internet of Things 905**
 - 31.1 Big Data 905
 - 31.1.1 Introduction to Big Data 905
 - 31.1.2 MapReduce 906
 - 31.1.3 Hadoop Software Stack 910
 - 31.1.4 Other Big Data Tools 911
 - 31.1.5 NoSQL Databases 912
 - 31.2 Cloud Computing 913
 - 31.2.1 Services Models, Pricing, and Standards 914
 - 31.2.2 Virtual Machines, Data Centers, and Intercloud Connections 917
 - 31.2.3 Cloud Infrastructure Requirements 920
 - 31.3 Internet of Things 922
 - 31.3.1 Architecture of IoT 922
 - 31.3.2 Cyber-Physical System Versus IoT 924
 - 31.4 Fog/Edge Computing 927
 - 31.5 Blockchain 928
 - References 930
- Appendix A: Mathematical Preliminaries 933**
- Appendix B: Benchmarks and Resources 957**
- Index 979**

Abbreviations

A/D	Analog-to-digital
adaline	Adaptive linear element
AI	Artificial intelligence
AIC	Akaike information criterion
ALA	Adaptive learning algorithm
ANFIS	Adaptive-network-based fuzzy inference system
AOSVR	Accurate online SVR
APCA	Asymmetric PCA
APEX	Adaptive principal components extraction
API	Application programming interface
ART	Adaptive resonance theory
ASIC	Application-specific integrated circuit
ASSOM	Adaptive-subspace SOM
BAM	Bidirectional associative memory
BFGS	Broyden–Fletcher–Goldfarb–Shanno
BIC	Bayesian information criterion
BIRCH	Balanced iterative reducing and clustering using hierarchies
BP	Backpropagation
BPTT	Backpropagation through time
BSB	Brain-states-in-a-box
BSS	Blind source separation
CBIR	Content-based image retrieval
CCA	Canonical correlation analysis
CCCP	Constrained concave-convex procedure
cdf	Cumulative distribution function
CEM	Classification EM
CG	Conjugate gradient
CMAC	Cerebellar model articulation controller
COP	Combinatorial optimization problem
CORDIC	Coordinate rotation digital computer

CoSaMP	Compressive sampling matching pursuit
CPT	Conditional probability table
CPU	Central processing units
CURE	Clustering using representation
DBSCAN	Density-based spatial clustering of applications with noise
DCS	Dynamic cell structures
DCT	Discrete cosine transform
DFP	Davidon–Fletcher–Powell
DFT	Discrete Fourier Transform
ECG	Electrocardiogram
ECOC	Error-correcting output code
EEG	Electroencephalogram
EKF	Extended Kalman filtering
ELM	Extreme learning machine
EM	Expectation–maximization
ERM	Empirical risk minimization
E-step	Expectation step
ETF	Elementary transcendental function
EVD	Eigenvalue decomposition
FCM	Fuzzy <i>C</i> -means
FFT	Fast Fourier Transform
FIR	Finite impulse response
fMRI	Functional magnetic resonance imaging
FPGA	Field-programmable gate array
FSCL	Frequency-sensitive competitive learning
GAP-RBF	Growing and pruning algorithm for RBF
GCS	Growing cell structures
GHA	Generalized Hebbian algorithm
GLVQ-F	Generalized LVQ family algorithms
GNG	Growing neural gas
GSO	Gram–Schmidt orthonormal
HWO	Hidden weight optimization
HyFIS	Hybrid neural fuzzy inference system
ICA	Independent component analysis
IHT	Iterative hard thresholding
iid	Independently drawn and identically distributed
IoT	Internet of Things
KKT	Karush–Kuhn–Tucker
LASSO	Least absolute selection and shrinkage operator
LBG	Linde–Buzo–Gray
LDA	Linear discriminant analysis
LM	Levenberg–Marquardt
LMAM	LM with adaptive momentum
LMI	Linear matrix inequality
LMS	Least mean squares

LMSE	Least mean squared error
LMSER	Least mean square error reconstruction
LP	Linear programming
LS	Least squares
LSI	Latent semantic indexing
LTG	Linear threshold gate
LVQ	Learning vector quantization
MAD	Median of the absolute deviation
MAP	Maximum a posteriori
MCA	Minor component analysis
MDL	Minimum description length
MEG	Magnetoencephalogram
MFCC	Mel frequency cepstral coefficient
MIMD	Multiple instruction multiple data
MKL	Multiple kernel learning
ML	Maximum likelihood
MLP	Multilayer perceptron
MSA	Minor subspace analysis
MSE	Mean squared error
MST	Minimum spanning tree
M-step	Maximization step
NARX	Nonlinear autoregressive with exogenous input
NEFCLASS	Neurofuzzy classification
NEFCON	Neurofuzzy controller
NEFLVQ	Non-Euclidean FLVQ
NEFPROX	Neuronfuzzy function approximation
NIC	Novel information criterion
<i>k</i> -NN	<i>k</i> -nearest neighbor
NOVEL	Nonlinear optimization via external lead
OBD	Optimal brain damage
OBS	Optimal brain surgeon
OLAP	Online analytical processing
OLS	Orthogonal least squares
OMP	Orthogonal matching pursuit
OWO	Output weight optimization
PAC	Probably approximately correct
PAST	Projection approximation subspace tracking
PASTd	PAST with deflation
PCA	Principal component analysis
PCM	Possibilistic <i>C</i> -means
pdf	Probability density function
PSA	Principal subspace analysis
QP	Quadratic programming
QR-cp	QR with column pivoting
RAN	Resource-allocating network

RBF	Radial basis function
RBM	Restricted Boltzmann machine
ReLU	Rectified linear unit
RIC	Restricted isometry constant
RIP	Restricted isometry property
RLS	Recursive least squares
RPCCL	Rival penalized controlled competitive learning
RPCL	Rival penalized competitive learning
Rprop	Resilient propagation
RTRL	Real-time recurrent learning
RVM	Relevance vector machine
SDP	Semidefinite programs
SIMD	Single instruction, multiple data
SLA	Subspace learning algorithm
SMO	Sequential minimal optimization
SOM	Self-organization maps
SPMD	Single program multiple data
SRM	Structural risk minimization
SVD	Singular value decomposition
SVDD	Support vector data description
SVM	Support vector machine
SVR	Support vector regression
TDNN	Time-delay neural network
TDRL	Time-dependent recurrent learning
TLMS	Total least mean squares
TLS	Total least squares
TREAT	Trust-region-based error aggregated training
TRUST	Terminal repeller unconstrained subenergy tunneling
TSK	Takagi–Sugeno–Kang
TSP	Traveling salesman problem
VC	Vapnik–Chervonenkis
VLSI	Very large-scale integrated
WINC	Weighted information criterion
<i>k</i> -WTA	<i>k</i> -winners-take-all
WTA	Winner-takes-all
XML	eXtensible markup language