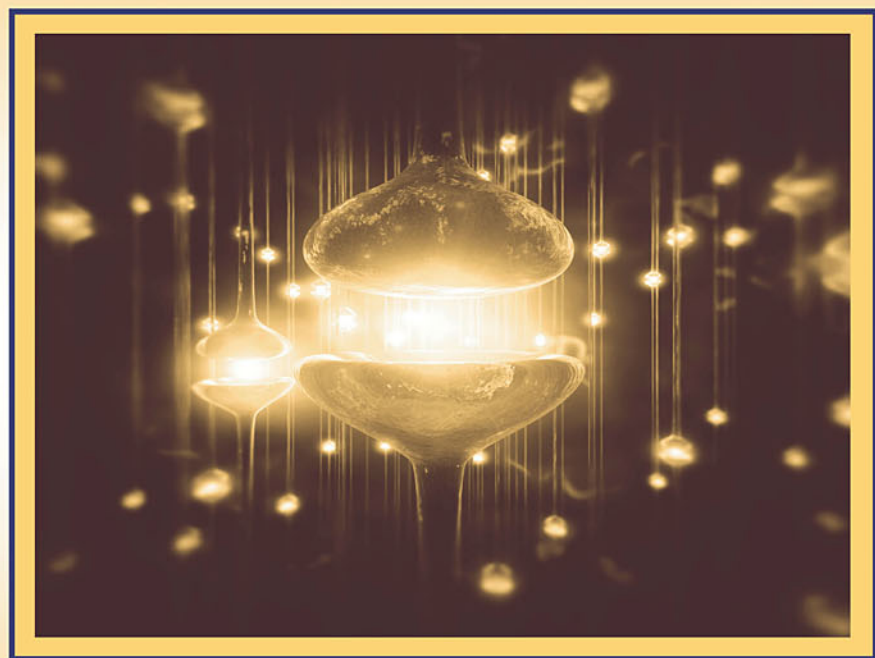


**Automation and Control
Engineering Series**

Subspace Learning of Neural Networks



**Jian Cheng Lv
Zhang Yi
Jiliu Zhou**



CRC Press
Taylor & Francis Group

Subspace Learning of Neural Networks

AUTOMATION AND CONTROL ENGINEERING

A Series of Reference Books and Textbooks

Series Editors

**FRANK L. LEWIS, Ph.D.,
Fellow IEEE, Fellow IFAC**

Professor

Automation and Robotics Research Institute
The University of Texas at Arlington

**SHUZHONG SAM GE, Ph.D.,
Fellow IEEE**

Professor

Interactive Digital Media Institute
The National University of Singapore

Subspace Learning of Neural Networks, *Jian Cheng Lv, Zhang Yi, and Jiliu Zhou*

Reliable Control and Filtering of Linear Systems with Adaptive Mechanisms,
Guang-Hong Yang and Dan Ye

Reinforcement Learning and Dynamic Programming Using Function
Approximators, *Lucian Buşoniu, Robert Babuška, Bart De Schutter,
and Damien Ernst*

Modeling and Control of Vibration in Mechanical Systems, *Chunling Du
and Lihua Xie*

Analysis and Synthesis of Fuzzy Control Systems: A Model-Based Approach,
Gang Feng

Lyapunov-Based Control of Robotic Systems, *Aman Behal, Warren Dixon,
Darren M. Dawson, and Bin Xian*

System Modeling and Control with Resource-Oriented Petri Nets, *Naiqi Wu
and MengChu Zhou*

Sliding Mode Control in Electro-Mechanical Systems, Second Edition,
Vadim Utkin, Jürgen Guldner, and Jingxin Shi

Optimal Control: Weakly Coupled Systems and Applications, *Zoran Gajić,
Myo-Taeg Lim, Dobrila Skatarić, Wu-Chung Su, and Vojislav Kecman*

Intelligent Systems: Modeling, Optimization, and Control, *Yung C. Shin
and Chengying Xu*

Optimal and Robust Estimation: With an Introduction to Stochastic Control
Theory, Second Edition, *Frank L. Lewis, Lihua Xie, and Dan Popa*

Feedback Control of Dynamic Bipedal Robot Locomotion, *Eric R. Westervelt,
Jessy W. Grizzle, Christine Chevallereau, Jun Ho Choi, and Benjamin Morris*

Intelligent Freight Transportation, *edited by Petros A. Ioannou*

Modeling and Control of Complex Systems, *edited by Petros A. Ioannou
and Andreas Pitsillides*

Wireless Ad Hoc and Sensor Networks: Protocols, Performance, and Control,
Jagannathan Saragapani

Stochastic Hybrid Systems, *edited by Christos G. Cassandras
and John Lygeros*

Hard Disk Drive: Mechatronics and Control, *Abdullah Al Mamun,
Guo Xiao Guo, and Chao Bi*

Autonomous Mobile Robots: Sensing, Control, Decision Making
and Applications, *edited by Shuzhi Sam Ge and Frank L. Lewis*

Automation and Control Engineering Series

Subspace Learning of Neural Networks

Jian Cheng Lv

*Sichuan University
Chengdu, People's Republic of China*

Zhang Yi

*Sichuan University
Chengdu, People's Republic of China*

Jiliu Zhou

*Sichuan University
Chengdu, People's Republic of China*



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2011 by Taylor and Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number-13: 978-1-4398-1536-6 (Ebook-PDF)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

Dedication

To all of our loved ones

Preface

Principal component analysis (PCA) neural networks, minor component analysis (MCA) neural networks and independent component analysis (ICA) neural networks can approximate a subspace of input data by learning. These networks inspired by biology and psychology provide a novel way for parallel online computation of a subspace. An input of these neural networks can be used at once so that they can enable fast adaptation in a nonstationary environment. Although these networks are almost linear neural models, they have found many applications, including applications relating to signal and image processing, video analysis, data mining, and pattern recognition.

The learning algorithms of these neural networks play a vital role in subspace learning. These subspace learning algorithms make these networks learn low-dimensional linear and multilinear models in a high-dimensional space, wherein specific statistical properties can be well preserved. The book will be mainly focused on the convergence analysis of these subspace learning algorithms and the ways to extend the use of these networks to fields such as biomedical signal processing, biomedical image processing, and surface fitting to name just a few.

A crucial issue of concern in a practical application is the convergence of the subspace learning algorithms of these neural networks. The convergence of these algorithms determines whether these applications can be successful. The book will analyze the convergence of these learning algorithms by mainly using discrete deterministic time (DDT) method. To guarantee their nondivergence, invariant sets of some algorithms will be obtained and global boundedness of some algorithms is studied. Then, the convergence conditions of these algorithms will be derived. Cauchy convergence principle and inequalities analysis method, and so on, will be used rigorously to prove the convergence. Furthermore, the book establishes a relationship between an SDT algorithm and the corresponding DDT algorithm by using block algorithms. This not only can overcome the shortcomings of DDT method, but also can get a good convergence and accuracy in practice. Finally, the chaotic and robust properties of some algorithms will also be studied. These results obtained lay the sound theoretical foundation of these networks and guarantee the successful applications of these algorithms in practice.

The book not only benefits the researcher of subspace learning algorithms, but also improves the quality of data mining, image processing, and signal processing. Besides its research contributions and applications, the book could

also serve as a good example for pushing the latest technologies in neural networks to some application community.

Scope and Contents of This Book

This book provides an analysis framework for convergence analysis of subspace learning algorithms of neural networks. The emphasis is on the analysis method, which can be generalized to the study of other learning algorithms. Our work builds a theoretical understanding of the convergence behavior of some subspace learning algorithms through the analysis framework. In addition, this book uses real-life examples to illustrate the performance of learning algorithms and instructs readers on how to apply them to practical applications. The book is organized as follows.

Chapter 1 provides a brief introduction to linear neural networks and subspace learning algorithms of neural networks. Some frequently used notations and preliminaries are given. Basic discussions on the methods for convergence analysis are presented which should lay the foundation for subsequent chapters.

In the following chapters, convergence of subspace learning algorithms is analyzed to lay the theoretical foundation for successful applications of these networks. In Chapter 2, the convergence of Oja's and Xu's algorithms with constant learning rates is studied in detail. The global convergence of Oja's algorithm with the adaptive learning rate is analyzed in Chapter 3. In Chapter 4, the convergence of Generalized Hebbian Algorithm (GHA) with adaptive learning rates is studied. MCA learning algorithms and the Hyvärinen-Oja's ICA learning algorithm are analyzed in Chapters 5 and 6, respectively. In Chapter 7, chaotic behaviors of subspace learning algorithms are presented.

Some problems concerning a practical application are discussed in chapters 8, 9, 10, 11, and some real-life examples are given to illustrate the performance of these subspace learning algorithms.

The contents of this book are mainly based on our research publications on this subject, which over the years have accumulated into a complete and unified coverage of the topic. It will serve as an interesting reference for post-graduates, researchers, and engineers who may be keen to use these neural networks in applications. Undoubtedly, there are other excellent works in this area, which we hope to have included in the references for the readers. We should also like to point out that at the time of this writing, many problems relating to subspace learning remained unresolved, and the book may contain personal views and conjecture of the authors that may not appeal to all sectors of readers. To this end, readers are encouraged to send us criticisms and suggestions, and we look forward to discussion and collaboration on the topic.

Acknowledgments

This book was supported in part by the National Science Foundation of China under grants 60971109 and 60970013.

Jian Cheng Lv
Zhang Yi
Jiliu Zhou

January 2010

List of Figures

1.1	A single neuron model.	2
1.2	A multineuron model.	2
1.3	Principal component direction and minor component direction in a two-dimensional data set.	4
1.4	Relationship between SDT algorithm and DDT algorithm. . .	10
1.5	The original picture of Lenna (left), and the picture divided into 4096 blocks (right).	12
2.1	Invariance of $S(2)$. Eighty trajectories starting from points in $S(2)$ remain in $S(2)$	28
2.2	A nonconvergent trajectory in $S(2)$	29
2.3	Convergence of (2.2).	30
2.4	Iteration step variations with respect to $\eta\sigma$	31
2.5	Convergence of Direction Cosine and norm of w	32
2.6	Divergence of (2.12).	45
2.7	Convergence of (2.12) with different initial vectors (left) and with different learning rates (right).	46
2.8	Convergence of direction cosine (left) and components of w (right).	47
2.9	The reconstructed image of the Lenna image.	48
2.10	Comparison of evolution rate of Xu's and Oja's algorithm with a same learning rate $\eta = 0.3$	50
3.1	Converge to an incorrect direction with $\eta(k) = \frac{1}{2^k + 1}$	52
3.2	The weights converge to the unit one (left) and the learning rates converge to 0.5 (right) with six different initial values. . .	67
3.3	Convergence of (3.7) with the distinct initial vectors.	67
3.4	Global convergence of (3.5) with small initial vectors (left) and the large ones (right).	68
3.5	Convergence of (3.5) (left) and learning rate converge to 0.98059 (right).	69
3.6	Convergence of (3.5) (left) and learning rate converge to 0.98059 (right).	70
3.7	Convergence of (3.5) (left) and learning rate converge to 0.98059 (right).	70
3.8	Convergence of (3.4) with $\xi = 0.7$ (left) and $\xi = 0.01$ (right). .	71

3.9	Convergence of (3.4) with a small initial vector (norm = 0.2495).	72
3.10	Convergence of (3.4) with a middle initial vector (norm = 2.5005).	72
3.11	Convergence of (3.4) with a large initial vector (norm = 244.7259).	73
3.12	The reconstructed image of Lenna image.	74
4.1	The trajectories of the first principal direction w_1 converge to the eigenvectors $\pm[0.9239, 0.3827]^T$ (upper) with $\xi = 0.5$, and the trajectories of the second principal direction w_2 converge to the eigenvectors $\pm[-0.3827, 0.9239]^T$ (bottom) with $\xi = 0.5$	96
4.2	The trajectories of the first principal direction w_1 converge to the eigenvectors $\pm[0.9239, 0.3827]^T$ (upper) with $\xi = 0.5$, and the trajectories of the second principal direction w_2 converge to the eigenvectors $\pm[-0.3827, 0.9239]^T$ (bottom) with $\xi = 0.5$	97
4.3	The norm evolution of four principal directions with $\xi = 0.3$ (upper) or $\xi = 0.7$ (bottom).	99
4.4	The learning rate evolution of four principal directions with $\xi = 0.3$ (upper) or $\xi = 0.7$ (bottom).	100
4.5	The direction cosine of four principal directions with $\xi = 0.3$ (upper) or $\xi = 0.7$ (bottom).	101
4.6	The norm evolution of six principal directions with $\zeta = 0.5$	102
4.7	The direction cosine of six principal directions with $\zeta = 0.5$	102
4.8	The reconstructed image (right) with SNR 53.0470.	103
5.1	Convergence of $\ w(k)\ $ in the deterministic case.	119
5.2	Convergence of Direction Cosine of $w(k)$ in the deterministic case.	120
5.3	Convergence of $\ w(k)\ $ in the stochastic case.	120
5.4	Convergence of Direction Cosine of $w(k)$ in the stochastic case.	121
6.1	The evolution of components of $\mathbf{z}$ with initial value: $\mathbf{z}(0) = [-0.3 \ -0.6 \ 0.2 \ 0.4]^T$ (upper) and $\mathbf{z}(0) = [-1 \ -0.5 \ -0.1 \ -0.2]^T$ (bottom), while $\sigma = 1$ and $Kurt(s_1) = -0.6, Kurt(s_2) = 0, Kurt(s_3) = 3, Kurt(s_4) = 0.4$	140
6.2	The evolution of components of $\mathbf{z}$ with the different kurtosis: $Kurt(s_1) = -0.6, Kurt(s_2) = 0, Kurt(s_3) = 3, Kurt(s_4) = -0.4$ (upper); $Kurt(s_1) = -0.6, Kurt(s_2) = 0, Kurt(s_3) = -2.5, Kurt(s_4) = -0.4$ (bottom), while $\sigma = -1$ and the initial value: $\mathbf{z}(0) = [-0.4 \ -0.8 \ 0.2 \ 0.4]^T$	141
6.3	The original sources s_1 and s_2 with $Kurt(s_1) = 2.6822, Kurt(s_2) = 2.4146$ (upper); two mixed and whitened signals (bottom).	143
6.4	The evolution of norm of $\mathbf{w}$ with $L = 200, \mathbf{w}(0) = [0.3 \ 0.05]^T$ (upper) and with $L = 1000, \mathbf{w}(0) = [0.03 \ 0.5]^T$ (bottom).	144

6.5	The evolution of $DirectionCosine(k)$ with $L = 200, \mathbf{w}(0) = [0.3 \ 0.05]^T$ (upper) and with $L = 1000, \mathbf{w}(0) = [0.03 \ 0.5]^T$ (bottom).	145
6.6	The extracted signal with $L = 200, \mathbf{w}(0) = [0.3 \ 0.05]^T$ (upper) and with $L = 1000, \mathbf{w}(0) = [0.03 \ 0.5]^T$ (bottom).	146
6.7	The original sources s_1, s_2, s_3 , and s_4 with $Kurt(s_1) = -1.4984, Kurt(s_2) = 0.0362, Kurt(s_3) = -1.1995 Kurt(s_4) = 2.4681$ (upper); four mixed and whitened signals (bottom). .	147
6.8	The evolution of norm of $\mathbf{w}$ (upper) and the evolution of $DirectionCosine(k)$ (bottom) of the algorithm (6.1) with the initial value $\mathbf{w}(0) = [0.6 \ 1.2 \ 0.1 \ 0.1]^T$	148
6.9	The evolution of norm of $\mathbf{w}$ (upper) and the evolution of $DirectionCosine(k)$ (bottom) of the algorithm (6.2) with the initial value $\mathbf{w}(0) = [0.6 \ 1.2 \ 0.1 \ 0.1]^T$	149
6.10	The extracted signal of (6.1) with $Kurt(s_4) = 2.4681$ (upper) and the extracted signal of (6.2) with $Kurt(s_1) = -1.4984$ (bottom).	150
6.11	Comparison of convergent rate of algorithm (6.1) with a constant learning rate $\eta = 0.05$ and a zero-approaching learning rate $\eta(k) = \frac{0.1}{2\log_2(k+1)}$. The dash-dot line denotes the evolution of the algorithm with the a zero-approaching learning rate. .	151
6.12	Comparison of convergent rate of algorithm (6.2) with a constant learning rate $\eta = 0.05$ and a zero-approaching learning rate $\eta(k) = \frac{0.1}{2\log_2(k+1)}$. The dash-dot line denotes the evolution of the algorithm with the a zero-approaching learning rate. .	151
7.1	The invariant set S_1 of the algorithm (7.1), and the divergent regions I, II , and III	160
7.2	The invariant set S_2 of the algorithm (7.2), and the divergent regions I, II , and III	160
7.3	Divergence of the algorithm (7.1) with $\eta(a + f(\ w(k)\ ^2)) = 4.01$ and $w^2(0) = 0.01 < \frac{(a+f(\ w(k)\ ^2))}{b} = 2.6733$ (left), divergence of the algorithm (7.2) with $\eta(a - f(\ w(k)\ ^2)) = 2.01$ and $w^2(0) = 0.01 < \frac{\eta(a-f(\ w(k)\ ^2))+2}{\eta b} = 2.6733$ (right).	161
7.4	The algorithm (7.1) converges to 0 with the different initial values and with $b = 0.2, \eta(a + f) = 0.56$ (left) and $\eta(a + f) = 1.4$ (right).	163
7.5	The algorithm (7.2) converges to 0 at the initial values $w(0) = 0.4$ or $w(0) = -0.4$ with the different $\eta(a - f(w^2))$ and $b = 3$ (left). The algorithm converge to $\pm \sqrt{\frac{a-f(\ w(k)\ ^2)}{b}} = \pm 0.8165$ with $b = 3, \eta = 0.4, (a - f(\ w(k)\ ^2)) = 2$ at the different initial values (right).	164
7.6	Bifurcation diagram of (7.1) with $w(0) = 0.01$	164

7.7	Bifurcation diagram of (7.2) with $w(0) = 0.01$ (left) and $w(0) = -0.01$ (right).	165
7.8	Lyapunov exponents of (7.1).	166
7.9	Lyapunov exponents of (7.2).	166
8.1	The original GHA model (left) and an improved GHA model with lateral connections (right).	171
8.2	The number of principal directions approximates the intrinsic dimensionality with $\alpha = 0.98$ (left) and the number will oscillate around the intrinsic dimensionality if $f_3(k)$ is not used (right).	178
8.3	The number of principal directions approximates the intrinsic dimensionality with $\alpha = 0.95$ (upper); the number will oscillate around the intrinsic dimensionality if $f_3(k)$ is not used (middle), and the number of principal directions approximates the intrinsic dimensionality with $\alpha = 0.98$ (bottom).	179
8.4	The Lenna image is compacted into an three-dimensional subspace with $\alpha = 0.95$ (left), and the image is compacted into an eight-dimensional subspace $\alpha = 0.98$ (right).	180
8.5	The reconstructed image with $\alpha = 0.95$ (left), and the reconstructed image with $\alpha = 0.98$ (right).	180
8.6	The original image for <i>Boat</i> (upper); the number of principal directions converging to the intrinsic dimensionality 18 with $\alpha = 0.98$ (middle), and the reconstructed image (bottom). . .	181
8.7	The original image for <i>Aerial</i> (upper); the number of principal directions converging to the intrinsic dimensionality 35 with $\alpha = 0.98$ (middle), and the reconstructed image (bottom). . .	182
8.8	The original image for <i>Plastic bubbles</i> (upper); the number of principal directions converging to the intrinsic dimensionality 37 with $\alpha = 0.98$ (middle), and the reconstructed image (bottom).	183
9.1	The original data set taken from the ellipsoid segment (9.4). .	189
9.2	The noise-disturbed data set used for surface fitting, which is obtained by adding Gaussian noise with zero means and variance $\sigma^2 = 0.3$ in the way given by (9.5).	190
9.3	The data points obtained after applying the improved Oja+'s algorithm to each 10×10 block with $\mu = 40$	190
9.4	The data points obtained after applying the improved Oja+'s algorithm to the grouped data with $\mu = 40$, that is, the final result of surface fitting by using multi-block-based MCA on the data set given by Figure 9.2.	191
9.5	The original data points taken from the saddle segment (9.6). .	192

9.6	The noise-disturbed data set used for surface fitting, which is obtained by adding Gaussian noise with zero means and variance $\sigma^2 = 0.5$ in the way given by (9.5).	193
9.7	The data points obtained after applying the improved Oja+'s algorithm to each 10×10 block with $\mu = 30$	193
9.8	The data points obtained after applying the improved Oja+'s algorithm to the grouped data with $\mu = 50$, that is, the final result of surface fitting by using multi-block-based MCA on the data set given by Figure 9.6.	194
10.1	ECG signals taken from a pregnant woman. x_1 - x_5 and x_6 - x_8 correspond to abdominal and thoracic measurements from a pregnant woman, respectively.	201
10.2	Autocorrelation of signal of x_1 of Figure 10.1.	202
10.3	Extracted FECG: y_1 is extracted by Barros's algorithm; y_2 is by our algorithm (10.6).	202
10.4	The value of the objective function (10.5). It declines with the parameters of w and b updated by the new algorithm.	202
11.1	Threshold segmentation: (a) Reference image and (b) its reference feature image with threshold 40. (c) Float image and (d) its float feature image with threshold 40.	206
11.2	Contour extraction: (a) CT image (reference image) and (b) its reference feature image. (c) MR image (float image) and (d) its float feature image.	207
11.3	MR-MR image registration: (a) Reference image and (b) its feature image with threshold 31. (c) Float image and (d) its feature image with threshold 31. (e) Comparison of contours before registration versus (f) after registration	210
11.4	CT-MR images registration: (a) Patient's head CT image and (b) its contour, centroid, principal components. (c) Patient's head MR image and (d) its contour, centroid, principal components. (e) Comparison of contours before registration versus (f) after registration	211

List of Tables

2.1	The $Norm^2$ of Initial Vector	46
3.1	The Norm of Initial Vector	66
3.2	The Average SNR	73
9.1	The solution obtained by different methods on the same noisy data set. The average values of w obtained from 20 simulations are given.	192
9.2	The solution obtained by different methods on the same noisy data set. The average values of w obtained from 20 simulations are given.	195

Contents

1	Introduction	1
1.1	Introduction	1
1.1.1	Linear Neural Networks	1
1.1.2	Subspace Learning Algorithms	3
1.1.2.1	PCA Learning Algorithms	3
1.1.2.2	MCA Learning Algorithms	5
1.1.2.3	ICA Learning Algorithms	6
1.1.3	Outline of This Book	7
1.2	Methods for Convergence Analysis	8
1.2.1	DCT Method	8
1.2.2	DDT Method	9
1.3	Relationship between SDT Algorithm and DDT Algorithm	10
1.4	Some Notations and Preliminaries	11
1.4.1	Covariance Matrix	11
1.4.2	Simulation Data	11
1.5	Conclusion	12
2	PCA Learning Algorithms with Constants Learning Rates	13
2.1	Introduction	13
2.2	Preliminaries	14
2.3	Oja's Algorithm	15
2.3.1	Oja's Algorithm and the DCT Method	16
2.3.2	DDT Formulation	16
2.3.3	Invariant Sets and Convergence Results	17
2.3.4	Simulation and Discussions	27
2.3.4.1	Illustration of Invariant Sets	27
2.3.4.2	Selection of Initial Vectors	28
2.3.4.3	Selection of Learning Rate	29
2.3.5	Conclusion	33
2.4	Xu's LMSE Learning Algorithm	33
2.4.1	Formulation and Preliminaries	33
2.4.2	Invariant Set and Ultimate Bound	34
2.4.3	Convergence Analysis	37
2.4.4	Simulations and Discussions	45
2.4.5	Conclusion	47
2.5	Comparison of Oja's Algorithm and Xu's Algorithm	47
2.6	Conclusion	49

3	PCA Learning Algorithms with Adaptive Learning Rates	51
3.1	Introduction	51
3.2	Adaptive Learning Rate	51
3.3	Oja's PCA Algorithms with Adaptive Learning Rate	53
3.4	Convergence Analysis of Oja's Algorithms with Adaptive Learning Rate	55
3.4.1	Boundedness	55
3.4.2	Global Convergence	58
3.5	Simulation and Discussion	65
3.6	Conclusion	74
4	GHA PCA Learning Algorithm	75
4.1	Introduction	75
4.2	Problem Formulation and Preliminaries	75
4.3	Convergence Analysis	78
4.3.1	Outline of Proof	78
4.3.2	Case 1: $l = 1$	79
4.3.3	Case 2: $1 < l \leq p$	84
4.4	Simulation and Discussion	95
4.4.1	Example 1	95
4.4.2	Example 2	95
4.4.3	Example 3	98
4.5	Conclusion	103
5	MCA Learning Algorithms	105
5.1	Introduction	105
5.2	A Stable MCA Algorithm	106
5.3	Dynamical Analysis	107
5.4	Simulation Results	118
5.5	Conclusion	121
6	ICA Learning Algorithms	123
6.1	Introduction	123
6.2	Preliminaries and Hyvärinen-Oja's Algorithms	124
6.3	Convergence Analysis	126
6.3.1	Invariant Sets	127
6.3.2	DDT Algorithms and Local Convergence	130
6.4	Extension of the DDT Algorithms	138
6.5	Simulation and Discussion	139
6.5.1	Example 1	139
6.5.2	Example 2	142
6.5.3	Example 3	142
6.6	Conclusion	152

7	Chaotic Behaviors Arising from Learning Algorithms	153
7.1	Introduction	153
7.2	Invariant Set and Divergence	154
7.3	Stability Analysis	161
7.4	Chaotic Behavior	163
7.5	Conclusion	167
8	Determination of the Number of Principal Directions in a Biologically Plausible PCA Model	169
8.1	Introduction	169
8.2	The PCA Model and Algorithm	170
8.2.1	The PCA Model	170
8.2.2	Algorithm Implementation	172
8.3	Properties	174
8.4	Simulations	177
8.4.1	Example 1	177
8.4.2	Example 2	178
8.5	Conclusion	184
9	Multi-Block-Based MCA for Nonlinear Surface Fitting	185
9.1	Introduction	185
9.2	MCA Method	186
9.2.1	Matrix Algebraic Approaches	186
9.2.2	Improved Oja+'s MCA Neural Network	186
9.2.3	MCA Neural Network for Nonlinear Surface Fitting	187
9.3	Multi-Block-Based MCA	188
9.4	Simulation Results	189
9.4.1	Simulation 1: Ellipsoid	189
9.4.2	Simulation 2: Saddle	192
9.5	Conclusion	195
10	A ICA Algorithm for Extracting Fetal Electrocardiogram	197
10.1	Introduction	197
10.2	Problem Formulation	198
10.3	The Proposed Algorithm	199
10.4	Simulation	200
10.5	Conclusion	203
11	Rigid Medical Image Registration Using PCA Neural Network	205
11.1	Introduction	205
11.2	Method	206
11.2.1	Feature Extraction	206
11.2.2	Computing the Rotation Angle	207
11.2.3	Computing Translations	209
11.3	Simulations	210

11.3.1 MR-MR Registration	210
11.3.2 CT-MR Registration	210
11.4 Conclusion	211
Bibliography	213

Introduction

1.1 Introduction

Subspace leaning neural networks provide a parallel online automated learning of low-dimensional models in a nonstationary environment. It is commonly known that automated learning of low-dimensional linear or multilinear models from training data has become a standard paradigm in computer vision [54, 55]. Thus, these neural networks used to learning a low-dimensional model have many important applications in computer vision, such as structure from motion, motion estimation, layer extraction, objection recognition, and object tracking [18, 19, 50, 56, 55].

Subspace learning algorithms used to update the weights of these networks pay a vital important role in applications. This book will focus on the subspace learning algorithms of principal component analysis (PCA) neural networks, minor component analysis (MCA) neural networks, and independent component analysis (ICA) neural networks. Generally, they are linear neural networks.

1.1.1 Linear Neural Networks

Inspired by biological neural networks, a simple neural model is designed to mimic its biological counterpart, the neuron. The model accepted the weighted set of input x responds with an output y , as shown in Figure 1.1. The vital effect of synapse between neurons is presented by the weights w . The mathematical model of a linear single neuron is as follows:

$$y(k) = w^T(k)x(k), (k = 0, 1, 2, \dots), \quad (1.1)$$

where $y(k)$ is the network output, the input sequence

$$\{x(k) | x(k) \in R^n (k = 0, 1, 2, \dots)\} \quad (1.2)$$

is a zero-mean stochastic process, each $w(k) \in R^n (k = 0, 1, 2, \dots)$ is a weight vector.

Consider input output relation

$$y(k) = W^T x(k), \quad (1.3)$$

where

$$y(k) = [y_1(k), y_2(k), \dots, y_m(k)]^T$$

and

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \dots & \dots & \dots & \dots \\ w_{n1} & w_{n2} & \dots & w_{nm} \end{bmatrix}.$$

This is a multineuron linear model, as shown in Figure 1.2.

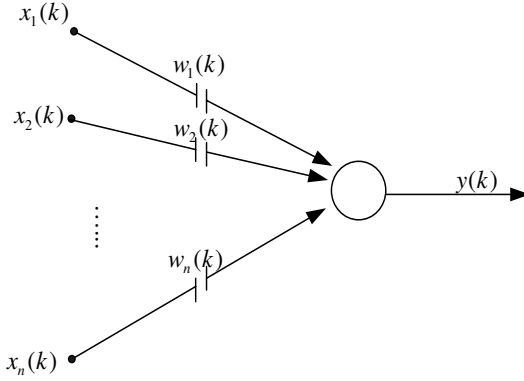


FIGURE 1.1

A single neuron model.

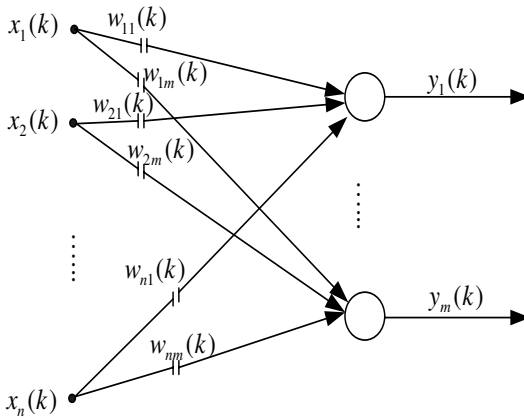


FIGURE 1.2

A multineuron model.

These linear networks have been widely studied [25, 28, 70] and used for

many fields involving signal processing [66], prediction [80], associative memory [167], function approximation [180], power system [15, 153], chemistry [196], and so on. The weight of these neural networks, which represents the strength of the connection between neurons, plays a very important role in the different applications. A variety of learning algorithms are used to update the weight so that the networks have the different applications with the corresponding weight. For instance, the adaptive linear neuron (ADALINE) network is one of the most widely used neural networks in practical applications, which was introduced by Widrow and M. Hoff in 1960 [181]. The least mean square (LMS) algorithm is used to update the weight so that ADALINE can be used as a adaptive filter [75].

In this book, these linear networks are used to extract the principal components, or minor components, or independent components from input data. It is required that these networks must approximate a low-dimensional model.

1.1.2 Subspace Learning Algorithms

Subspace learning algorithms are used to update the weights of these linear networks so that these networks intend to approximate a low-dimensional linear model. Generally, an original stochastic discrete time (SDT) algorithm is formulated as

$$w(k+1) = w(k) \pm \eta(k) \triangle w(k), \quad (1.4)$$

where $\eta(k)$ is learning rate and $\triangle w(k)$ determines the change at time k .

This book will mainly discuss the following subspace learning algorithms: PCA learning algorithms, MCA learning algorithms, and ICA learning algorithms.

1.1.2.1 PCA Learning Algorithms

Principal component analysis (PCA) is a traditional statistical technique in multivariate analysis, stemming from the early work of Pearson [141]. It is closely related to Karhunen-Loève (KL) transform, or the Hotelling transform [82]. The purpose of PCA is to reduce the dimensionality of a given data set, while retaining as much as possible of the information present in the data set.

Definition 1.1 *A vector is called the first principal component direction if the vector is along the eigenvector associated with the largest eigenvalue of the covariance matrix of a given data set, and a vector is called the second principal component direction if the vector is along the eigenvector associated with the second largest eigenvalue of the covariance matrix of a given data set, and so on.*

Definition 1.2 *Principal components are the variances that are obtained by projecting the given data onto the principal component directions.*