Studies in Computational Intelligence 400

Akira Hirose

Complex-Valued Neural Networks

2nd Edition

Springer

Studies in Computational Intelligence, Volume 400

Editor-in-Chief
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series can be found on our homepage: springer.com

Vol. 380. Anthony Brabazon, Michael O'Neill, and Dietmar Maringer (Eds.)
Natural Computing in Computational Finance, 2011
ISBN 978-3-642-23335-7

Vol. 381. Radosław Katarzyniak, Tzu-Fu Chiu, Chao-Fu Hong, and Ngoc Thanh Nguyen (Eds.)
Semantic Methods for Knowledge Management and Communication, 2011
ISBN 978-3-642-23417-0

Vol. 382. F.M.T. Brazier, Kees Nieuwenhuis, Gregor Pavlin, Martijn Warnier, and Costin Badica (Eds.)
Intelligent Distributed Computing V, 2011
ISBN 978-3-642-24012-6

Vol. 383. Takayuki Ito, Minjie Zhang, Valentin Robu, Shaheen Fatima, and Tokuro Matsuo (Eds.)
New Trends in Agent-Based Complex Automated Negotiations, 2012
ISBN 978-3-642-24695-1

Vol. 384. Daphna Weinshall, Jörn Anemüller, and Luc van Gool (Eds.)
Detection and Identification of Rare Audiovisual Cues, 2012
ISBN 978-3-642-24033-1

Vol. 385. Alex Graves
Supervised Sequence Labelling with Recurrent Neural Networks, 2012
ISBN 978-3-642-24796-5

Vol. 386. Marek R. Ogiela and Lakhmi C. Jain (Eds.)
Computational Intelligence Paradigms in Advanced Pattern Classification, 2012
ISBN 978-3-642-24048-5

Vol. 387. David Alejandro Pelta, Natalio Krasnogor, Dan Dumitrescu, Camelia Chira, and Rodica Lung (Eds.)
Nature Inspired Cooperative Strategies for Optimization (NICSO 2011), 2011
ISBN 978-3-642-24093-5

Vol. 388. Tiansi Dong
Recognizing Variable Environments, 2012
ISBN 978-3-642-24057-7

Vol. 389. Patricia Melin
Modular Neural Networks and Type-2 Fuzzy Systems for Pattern Recognition, 2012
ISBN 978-3-642-24138-3

Vol. 390. Robert Bembenik, Lukasz Skonieczny, Henryk Rybiński, and Marek Niezgódka (Eds.)
Intelligent Tools for Building a Scientific Information Platform, 2012
ISBN 978-3-642-24808-5

Vol. 391. Herwig Unger, Kyandoghere Kyamaky, and Janusz Kacprzyk (Eds.)
Autonomous Systems: Developments and Trends, 2012
ISBN 978-3-642-24805-4

Vol. 392. Narendra Chauhan, Machavaram Kartikeyan, and Ankush Mittal
Soft Computing Methods for Microwave and Millimeter-Wave Design Problems, 2012
ISBN 978-3-642-25562-5

Vol. 393. Hung T. Nguyen, Vladik Kreinovich, Berlin Wu, and Gang Xiang
Computing Statistics under Interval and Fuzzy Uncertainty, 2012
ISBN 978-3-642-24904-4

Vol. 394. David A. Elizondo, Agusti Solanas, and Antoni Martínez-Ballesté (Eds.)
Computational Intelligence for Privacy and Security, 2012
ISBN 978-3-642-25236-5

Vol. 395. Srikanta Patnaik and Yeon-Mo Yang (Eds.)
Soft Computing Techniques in Vision Science, 2012
ISBN 978-3-642-25506-9

Vol. 396. Marielba Zacarias and José Valente de Oliveira (Eds.)
Human-Computer Interaction: The Agency Perspective, 2012
ISBN 978-3-642-25690-5

Vol. 397. Elena Nikolaevskaya, Alexandr Khimich, and Tamara Chistyakova
Programming with Multiple Precision, 2012
ISBN 978-3-642-25672-1

Vol. 398. Fabrice Guillet, Gilbert Ritschard, and Djamel Abdelkader Zighed (Eds.)
Advances in Knowledge Discovery and Management, 2012
ISBN 978-3-642-25837-4

Vol. 399. Kurosh Madani, António Dourado Correia, Agostinho Rosa, and Joaquim Filipe (Eds.
Computational Intelligence, 2012
ISBN 978-3-642-27533-3

Vol. 400. Akira Hirose
Complex-Valued Neural Networks, 2012
ISBN 978-3-642-27631-6

Akira Hirose

Complex-Valued Neural Networks

Second Edition

Springer

Author
Prof. Akira Hirose
The University of Tokyo
Japan

Original Japanese Language Edition Published by Saiensu-sha Co., Ltd.
1-3-25, Sendagaya, Shibuya-ku, Tokyo 151-0051, Japan

ISSN 1860-949X e-ISSN 1860-9503
ISBN 978-3-642-27631-6 e-ISBN 978-3-642-27632-3
DOI 10.1007/978-3-642-27632-3
Springer Heidelberg New York Dordrecht London

Library of Congress Control Number: 2011945320

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface to the Second Edition

Five years have passed since the First Edition of this Book was published. Fortunately it obtained a large readership, and seems to have made a humble contribution to the development of the complex-valued neural-network research. The author is extremely grateful to the readers for their explicit or implicit insightful comments and continuous support. It is my great pleasure to have the occasion to publish the Second Edition.

During this period, the research on complex-valued neural networks expands largely in both the quality and the quantity. There have been many special sessions in representative conferences which cover not only mathematical sciences but also electrical and electronic engineering such as the International Joint Conference on Neural Networks (IJCNN) sponsored by IEEE Computational Intelligence Society (IEEE CIS) and International Neural Network Society (INNS), including the biyearly held World Congress on Computational Intelligence (WCCI), the International Conference on Neural Information Processing (ICONIP) promoted by Asia-Pacific Neural Network Assembly (APNNA) organized by Asia-Pacific countries' neural network societies such as Japanese Neural Network Society (JNNS), and the International Conferece on Artificial Neural Networks (ICANN) organized by the European Neural Network Society (ENNS). A project named "Practical Applications of Complex-Valued Neural Networks" has also been organized as one of the Nation-wide Cooperative Research Projects in Research Institute of Electrical Communication (RIEC), Tohoku University, in 2009 to accelerate application-oriented research in technology and society. A worldwide network, namely, the Task Force on Complex-Valued Neural Networks, has also been inaugurated in the Institute of Electrical and Electronics Engineers (IEEE) Computational Intelligence Society (CIS) Neural Network Technical Committee (NNTC) with more than 40 active members.

Recent progress is summarized as follows. The application fields spread more and more in electromagnetic-wave, lightwave and ultrasonic-wave engineering areas such as earth and environmental observation with satellite/airborne radar systems, security imaging at airports, railway stations

and other public transportation, and medical diagnosis and monitoring utilizing coherent wave phenomena. Other growing fields include adaptive image processing in frequency domain and time-sequential signal processing widely. Theories are going to extend from complex and quaternion networks to more general frameworks based on Clifford algebra.

This Second Edition is an enlarged and revised book featuring mainly the following two points. The first is the inclusion of recent trends in the overview in Section 3.7 (research history) as well as in respective chapter citation in Part II (applications), resulting in almost doubled number of references. The parametron invented in 1954 is also referred to with discussion on analogy and disparity. The second point is the addition of argument on the advantages of the complex-valued neural networks in Sections 3.1 (history of complex number), 3.2 (degree of freedom and circularity in learning), 3.4 (metric in complex domain) and some other sections to enhance the difference from real-valued neural networks.

The author is very much obliged to Prof. Janusz Kacprzyk, Editor-in-Chief, Studies in Computational Intelligence, and Dr. Thomas Ditzinger, Engineering Editor, Springer-Verlag, for their continuous help in publication. I express my sincere thanks also to Mr. Kosuke Hirase and Mr. Nobuhiko Tajima, Executive Editors, Saiensu-sha, for their kind permission and promotion.

Tokyo, Japan Akira Hirose
October 2011

Preface to the English First Edition

The original Japanese edition of this book, published by Saiensu-sha, Japan, in March 2004, has fortunately acquired a favorable reputation. I am grateful to the readers for their kind feedbacks, many of which are included in this edition. I hope this English publication attracts readers in wider areas, and evokes valuable feedbacks furthermore.

In the months after the Japanese publication, researches on the complex-valued neural networks have kept evolution in respective directions. There are some plans of special sessions in international conferences and special issues in journals. The bibliography has been slightly modified to include the special sessions and latest journal publications. On the other hand, references written in Japanese on domestic circumstances have been omitted.

Besides, Fig.1.1 has been added, and Fig.2.1 has been modified, which are related to the *Special Issue on Complex-Valued Neural Networks*, The Journal of the IEICE, 87 (6), June 2004 (in Japanese), so that even readers not having glanced the issue can obtain clear concepts. With these modifications, I expect a higher appeal in this English edition.

I am very much obliged to Dr. Thomas Ditzinger, Engineering Editor, Springer-Verlag, for his continuous help in publication.

Tokyo, Japan
April 2006

Akira Hirose

Preface to the Japanese First Edition

The studies on complex-valued neural networks have recently been evolving in various directions. The pioneering areas include electromagnetic-wave and lightwave sensing and imaging, independent component analysis in blind separation, blur restoration in image processing, and so on. Developing applications involve adaptive quantum devices, quantum computation, social systems related to periodicity and oscillation, and so forth.

This book is the first one to describe systematically the complex-valued neural networks. It is recommendable for researchers, graduate students, and undergraduate students in, for example, *electrical and electronic engineering, informatics, control engineering, mechanics, robotics, and bioengineering*. This book is useful for those who begin to study, for instance, adaptive signal processing for highly functional sensing and imaging, control in unknown and changing environment, brain-like information processing, robotics inspired by human neural systems, and interdisciplinary studies to realize comfortable society. It is also helpful to those who carry out *research and development regarding new products and services at companies*. The author wrote this book hoping in particular that the book provides them meaningful hints to make good use of neural networks in fully practical applications.

The first multiauthor book focusing on the complex-valued neural networks is "Complex-Valued Neural Networks: Theories and Applications" published by World Scientific Publishing Co. (October 2003), in which researchers most active in this field reconstruct and present their pioneering works. Besides, a foreword describes the relation of the complex-valued networks with other fields and the real world. In addition, an introductory chapter outlines this widely expanding field listing theoretical and application aspects.

On the other hand, this book, "Complex-Valued Neural Networks," describes the field systematically by a single researcher. The book emphasizes basic ideas and ways of thinking. Why do we need to consider neural networks that deal with complex numbers? What advantages do the complex-valued neural networks have? What is the origin of the advantages? In what areas do they develop principal applications? This book answers these questions

by describing details and examples, which will inspire many readers with new ideas. It may not be completely comprehensive. Instead, the author decided to make effort not to include all things relevant to this field, but to describe vividly this extensively developing field. Therefore, the fields, where the strong points of the complex-valued neural networks will potentially play important roles, extend further beyond those explained in this book.

The book consists of two parts. Part I describes basic concepts, ideas and fundamentals, while Part II presents application examples and illustrates how to use the networks in specific situations. In Part I, an application example is presented in Chapter 1, so that the readers grasp the rough idea of the complex-valued neural networks. Chapter 2 provides briefly the viewpoint of the artificial neural networks in general in the framework of information processing. Chapter 3 presents basic ideas, overview of applications, and a historic outline. Then, Chapter 4 explains constructions and dynamics of the complex-valued neural networks. In the descriptions, *conventional neural networks are first presented, and then, they are extended into the complex-valued networks. Consequently, the readers are able to understand the contents even if they do not have any knowledge on conventional neural networks.*

Therefore, *the book is recommendable as an introduction not only to the complex-valued networks, but also widely to the artificial neural networks.* The manner of description in this book places stress on the intuitional understanding of actual dynamics, practical advantages, and effective applications, rather than mathematical rigorousness. If the readers require formal mathematics further more, please refer to literature presented in each chapter.

The fields relevant to the complex-valued neural networks are enormously wide since, from a certain viewpoint, we may regard them as a superset of the conventional networks. The author is afraid that the book may includes serious errors and misunderstanding, on which the readers' corrections and suggestions are heartily welcome. Please note that the imaginary unit $\sqrt{-1}$ is expressed as i or j in accordance with the customs in respective fields.

The application examples presented in Part II have been obtained in the A. Hirose Laboratory, The University of Tokyo, with many graduate and undergraduate students belonging to the Courses in Electrical and Electronic Engineering, Computer Science, Frontier Informatics, and Interdisciplinary Science and Technology. That is, the results in Chapters 5 (land-surface segmentation) and 7 (digital elevation map) have been obtained mainly by Dr. Andriyan Bayu Suksmono, Chapter 6 (landmine visualization radar system) by Mr. Takahiro Hara, Chapters 8 (carrier-frequency-dependent associative memory) and 9 (optical phase equalizer) by Dr. Sotaro Kawata, Chapter 10 (developmental learning) by Mr. Yasufumi Asano and Mr. Toshihiko Hamano, and Chapter 11 (voice synthesis) by Mr. Keiichi Tsuda. The author expresses his sincere thanks to them and all other laboratory members and alumni. The details in the studies are available on the web pages of the Laboratory (http://www.eis.t.u-tokyo.ac.jp/).

The author is grateful also to Mr. Kosuke Hirase at Saiensu-sha, Japan, and Mr. Naoshi Takeda at Suuri-kogaku-sha, Japan, for their great help in publication.

Lastly, the author would like to ask all the readers for their cooperation in this developing field by quoting from the Preface, contributed by Dr. Nobuo Hataoka, in the *Special Issue on "Complex-Valued Neural Networks,"* The Journal of the IEICE, vol.87, No.6 (2004) p.446: "The complex-valued neural networks deal with phase information in addition. They possess remarkable ability in comparison with the conventional networks. However, they have just departed from the starting point." The author would like to make further effort to construct practically useful neural networks by obtaining the cooperation of researchers at large in the world.

Tokyo, Japan Akira Hirose
December 2004

Contents

Part II Applications: How Wide Are the Application Fields?

Part I

Basic Ideas and Fundamentals: Why Are Complex-Valued Neural Networks Inevitable?

1

Complex-Valued Neural Networks Fertilize Electronics

The complex-valued neural networks are the networks that deal with complex-valued information by using complex-valued parameters and variables. They are extremely rich in variety.[1] In this chapter, we grasp the basic ideas lying in the complex-valued neural networks by glancing over an application example. Then we also obtain a bird's-eye view of their present and prospective application fields so that we can enjoy the flavor before we go deep into the world of the complex-valued neural networks.[2]

1.1 Imitate the Brain, and Surpass the Brain

The art of the artificial neural networks is a technological framework in which we introduce and/or imitate the functions, constructions, and dynamics of the brain to realize an adaptive and useful information processing. The brain is able to manage both the pattern processing problems and the symbolic processing ones. For example, when we find a correct traveling route in a complex transportation network such as metro network in a large city, we first guess a possible route by a pattern processing and, afterward, we confirm the details and sequential consistencies. The principle and the mechanism of the brain functions are still unclear in total. However, the accumulation of physiological experiments has brought many important suggestions.

[1] Various features and applications are found, for example, in a series of special sessions in international conferences such as [1], [2], [3], [4], [5], [6], [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] and tutorials [17] [18] [19]. Details are available on the page of the Task Force on Complex-Valued Neural Networks (TF-CVNN) of the IEEE Computational Intelligence Society (CIS) Neural Network Technical Committee (NNTC) [20].

[2] This chapter is based on the article [21] (A.Hirose, Complex-valued neural networks fertilize electronics, Journal of the IEICE, 87 (6):447–449, June 2004). Chapters in a multiple-author book [22] and a Japanese review [23] are also helpful to extend the first impression of the complex-valued neural networks.

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 3–8.
springerlink.com

The biological brain, including sensory neural networks such as retinal and cochlear networks has various specific features. When it absorbs information of events occurring in the world, it reconstructs the information according to their meaning for the person. It also preserves the relation among the information meanings. The brain's manner to take in the information is roughly determined by the nature of the cells as well as the constructions of the nerve networks. Additionally the brain changes itself being influenced by the information presented by the environment. This change is the self-organization and / or learning in the neural system.

The purpose of the self-organization and learning lies in the reconstruction of information representation in the brain so that the man can utilize the information as effectively as possible. It is known that it is significantly important for the human neural network to adopt a representation suitable for the purpose assigned to respective network modules, in particular to a module located close to human-environment interface. Therefore, the network of each module also possesses a construction suitable for processing respective information specific to visual, auditory, or olfactory signals. The cerebral cortex has, however, a more homogeneous structure, i.e., the six-layer structure, so that one part of the cortex can be substituting for another part when an inability occurs. But it also self-organizes according to input signals.

The modern electronics and communications provide us with a large variety of information. It is hence expected more and more in the engineering fields to develop new systems that process a wider range of information in more adaptive and effective manners just like we do, or better than we do. In other words, we need to build systems surpassing the brain by imitating the brain. Even in such cases, the effective self-organization and learning inevitably require the information representations suitable for the purposes.

1.2 Create a "Superbrain" by Enrichment of the Information Representation

Let's consider an example. In these years, the measurement technology on the basis of the interference of waves makes remarkable progress. Assume that we have a coherent lightwave transmitter and phase-sensitive eyes, i.e., an interferometric radar function, so that we can *see* the phase of the reflected lightwave [24] as shown in Fig.1.1.

When a reflecting object approaches to us, the number of the wave (wave tops or bottoms) between the object and our eyes reduces. That is, the phase of the reflected lightwave progresses. Contrarily, when the object goes away, the phase is retarded. In this way, phase of the reflected lightwave expresses the distance between the object and us. The fluctuation is related to the unevenness and roughness of the object surface. Then, as we *see* the object coherently, our brain self-organizes in such a special manner that we can *see* the environment in a phase-sensitive way.

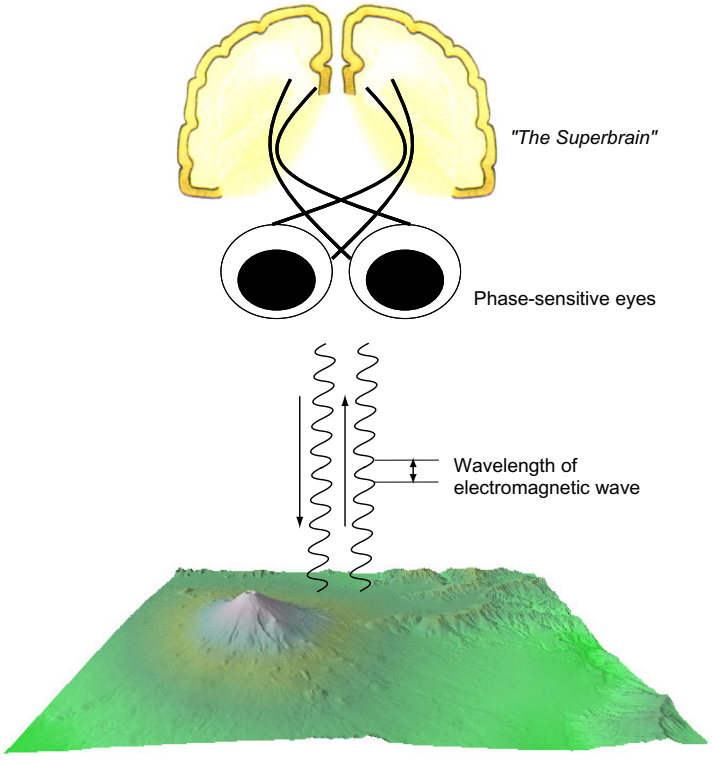


Fig. 1.1 Geographical profile acquisition using phase-sensitive eyes and a *superbrain* brought up with the special eyes. (Figure includes the data in Digital Map 50m Grid (Elevation), Geographical Survey Institute, with permission.)

That is, our brain looks the objective world adaptively on the basis of the amplitude and phase, i.e., ”complex amplitude” or ”phasor”, For example, we are on a plane and *see* Mt. Fuji and Lake Yamanaka beneath the craft. If we have phase-sensitive eyes, our brain takes in the information of the height of Mt. Fuji, the shape and roughness of its skirt, statistical unevenness of distance determined by vegetation, and the fluctuation *texture* in distance. Then the brain recognizes the state of the ground surface with a pattern processing method in complex-amplitude information space.

In Chapter 5, we present such an example of the phase-sensitive superbrain to see the region of Mt. Fuji and Lake Yamanaka. In Fig.5.2 on Page 105, you find the data of reflection observation, while Fig.5.6 on Page 110 shows the recognition results. Figure 5.6(a) was generated by a conventional neural network that sees only the intensity of the reflected wave. On the other hand, Fig.5.6(b) is the result obtained by the phase-sensitive superbrain that sees the complex amplitude. The latter figure gives an impression completely

different from that of the former one. Mt. Fuji is segmented as a mountain cluster, which is a useful result for human beings to live in the world. We have also succeeded in visualization of antipersonnel plastic landmines with this complex-amplitude superbrain brought up with the phase-sensitive eyes (Chapter 6).

One of the most important advantages of the complex-valued neural networks (CVNNs) is good compatibility with wave phenomena. The CVNNs are suitable for the processes related to complex amplitude such as the interferometric radar system mentioned above. In general, propagation and interference of electromagnetic wave is expressed by the magnitude of transmission and reflection, phase progression and retardation, superposition of fields, etc. These phenomena are expressed simply and naturally by the use of complex numbers. They are also related directly with the elemental processes in the CVNNs such as weighting at synaptic connections, i.e., multiplications in amplitude and shifts in phase, and summation of the weighted inputs.

1.3 Application Fields Expand Rapidly and Steadily

Regarding the research history of the CVNNs, we can trace back to the middle of 20th century. The details are described in Section 3.7. The first introduction of phase information in computation was made by Eiichi Goto in 1954 in his "parametron" [25] [26]. The parametron represents the values to be processed by discrete phase values of oscillation. The phase was used for a higher stability in the information representation in computing systems in 1950s. Aizenberg et al. discussed a set of multi-valued threshold logic in 1971 [27]. They suggested the representation of phase information by the shift of pulse timing. That is, an encoding based on time progression and delay was assumed.

These ideas are indeed suggestive even from the viewpoint of present research situations. The most useful applications include the above-mentioned coherent electromagnetic system, where we pay attention to amplitude and phase of electromagnetic wave. In such a system, the amplitude corresponds to energy, whereas the phase does to progression or retardation of time. The CVNNs deal with the information directly related to such existence that forms the basis of physical world.

There are many other fields in which the CVNNs provide systems with appropriate information representations. Figure 1.2 is a diagram presenting application fields. Many are related to wave phenomena, e.g., active antennas in electromagnetism, communications and measurements using waves such as radar image processing, learning electron-wave devices, quantum computation, ultrasonic imaging, analysis and synthesis in voice processing, and so on. The wavelength-dependent dynamics of optical circuit leads to adaptive optical routers in optical wavelength-division-multiplexed communications, variable optical connections, frequency-domain parallel information

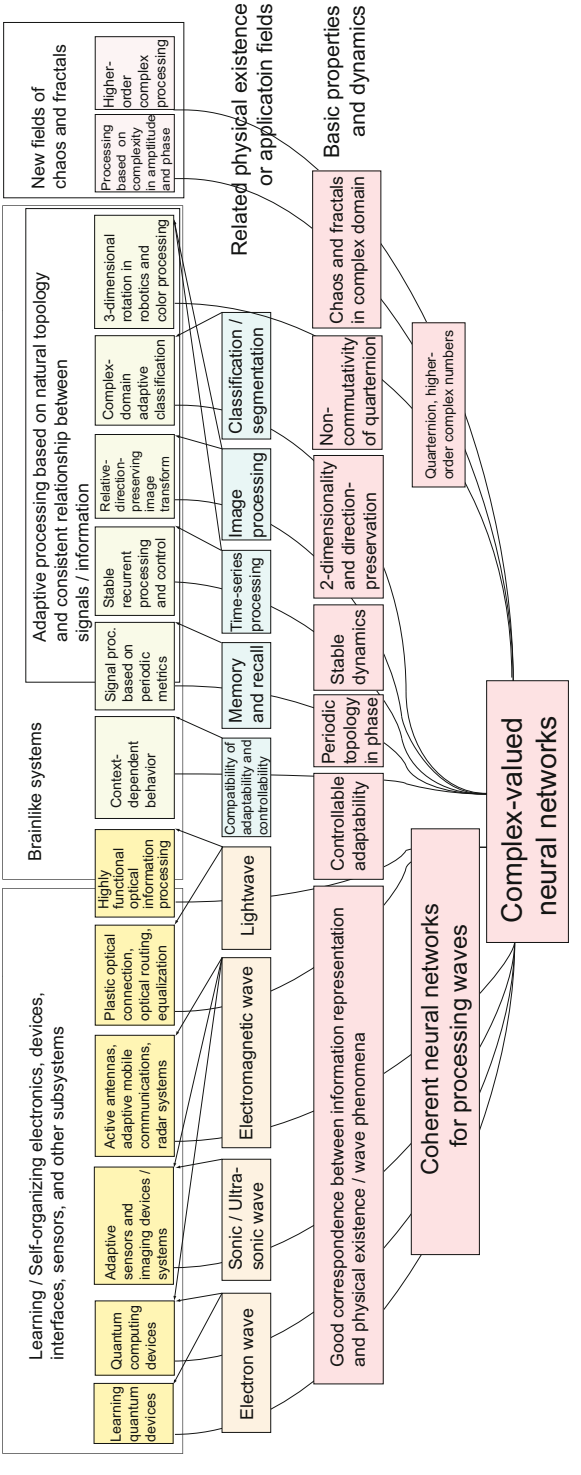


Fig. 1.2 Application fields of the complex-valued neural networks. Left-hand side: Items classified on the basis of physical existence. Right-hand side: Items classified on the basis of the features of neural dynamics. Details are explained in Chapter 3.6. (Reprinted from [21]: A.Hirose, "Complex-valued neural networks for more fertile electronics," Journal of the IEICE, 87(6):447–449, June 2004, with permission.)

processing, etc. The carrier-frequency-dependent neural behavior realizes both the adaptability and controllability in neural networks.

The compatibility of neural adaptability and controllability is closely related to context-dependent behavior and emergence of volition in neural networks and, hence, connected with so-called brainlike systems in the future. The periodicity in phase-information topology is applicable to systems that process information naturally having a cyclic structure, e.g., adaptive controller of traffic lights with periodic behavior. Such a research directly brings comfort and safety to human beings. Other topics include new development in chaos and fractals, and use of quaternion and higher-order complex numbers.

1.4 Book Organization

In this book, we present and discuss basic ideas and fundamentals of CVNNs in Part I. First, in Chapter 2, we describe the backgrounds and reasons why the art of the CVNNs becomes important more and more. Next, in Chapter 3, we present the features of CVNNs as well as in what applications they are especially useful. We also survey the history of CVNNs researches. In Chapter 4, we explain the dynamics of processes, learning, and self-organization of CVNNs. We present dynamics of conventional (real-valued) neural networks first and, afterward, we extend them to those of CVNNs. Therefore, we expect that even a beginner in conventional neural networks can easily absorb the basics of CVNNs. However, please consult the books listed in chapters, if needed, for further assistance required in understanding conventional neural networks.

In Part II, we present several examples of applications in CVNNs, proposed by the author's research group, such as an interferometric radar imaging system and an adaptive lightwave information processing system. We describe the features of self-organization and learning in these systems, and we show the effectiveness of the CVNNs that deals with phase information in waves. The framework adopted in the systems is not only useful in imaging and sensing using other wave phenomena such as sonic and ultrasonic waves, but also promising future development, e.g., in adaptive neural devices on the basis of electron wave [28],[22]. Furthermore, we explain in what manner a CVNN yields volition and developmental learning. We wish the ideas described in this book inspire the readers with new ideas more and more.

2

Neural Networks: The Characteristic Viewpoints

Before we describe complex-valued neural networks, first we review the basis of neural networks in general. The basic way of thinking is common to both the conventional (real-valued) and complex-valued neural networks.

2.1 Brain, Artificial Brain, Artificial Intelligence (AI), and Neural Networks

What is the difference between computers we use in the daily life and neural networks in their basic ideas and constructions? To know the difference must be a useful compass to navigate around the world of complex-valued neural networks.

The wonder of brain mechanism has been fascinating the human beings. How do we recognize and cope with environment? What are the mechanisms of the recognition, processing, learning, and adaptation? To begin with, what are the self and consciousness? Such questions have attracted many people. Aristotle in ancient Greek in the fourth century B.C. considered that the nature is made of four elements, i.e., fire, water, earth, and air, and they interchange with one another influenced by the sun. But living things additionally possess the *soul*, which is considered to bring us volition. That is, nutritive soul is given to plants, animals, and humans, while perceptive soul is to animals and humans, and rational soul is only to humans. A little earlier than Aristotle, Chuang-tzu in ancient China speculated on *self*. In his narrative, he was asleep on a lakeside. In his dream, he was a butterfly, and was flying around happily. But he noticed that perhaps he might be actually a butterfly, and that he might be a human being only in the butterfly's dream. What is the reality? What am I?

In the late 1930s and 1940s, researchers attempted variously and comprehensively to elucidate the excellent mechanisms of the human brain and utilize obtained knowledge to realize artificial brain. Various physiological measurements were conducted such as electroencephalography and

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 9–15.
springerlink.com

membrane-potential recording using microelectrodes. The McCulloch & Pitts proposed a simplified neuron model. Norbert Wiener founded cybernetics. D. O. Hebb presented so-called Hebbian rule, a hypothesis on the learning mechanism at synapses [29]. Furthermore, the concept of the *Turing machine* and Shannon's information theory was also developed. Roughly speaking, the aim of these researches was the realization of artificial brain.

However, after 1950s, the computers developed almost separately from the brain. The von Neumann computers, i.e., the present ordinary computers, made a great progress [30]. Hardware progress improved the speed and capacity amazingly. The principal element of computer hardware was first vacuum tube, then transistor, integrated circuit (IC), large-scale integrated circuit (LSI), and now it is very-large-scale integrated circuit (VLSI). The development in scale and function enables us to deal with a large quantity of information bits very quickly. The computers have become hence useful to human being and are widely used now. In the von Neumann computers, the function is determined by software separately from physical existence. That is, software is directed to process symbolic information expressed by *colorless* bits on the basis of logic, i.e., hard rule. In this symbolic processing, the expected process is clearly expressed by symbols and, therefore, the operation has no ambiguity. It has another advantage, i.e., the compatibility with reductionism. In other words, a problem to be solved may be reduced into a set of simpler problems, thanks to the clearness of logic. The modern society cannot go without computers even a single day.

The artificial brain function realized by such modern computers has been called artificial intelligence (AI). In AI, the principal operation is symbolic processing on the basis of discrete mathematics. Provided that a problem is expressed logically clearly, then we can construct an efficient algorithm (processing procedure) to solve it. Many various and useful algorithms have been developed hitherto. If a problem is given only in an ambiguous way, we first formalize the problem using symbolic representation. Then the computer searches an optimal action by using knowledge data and rules. An excellent example is the computer chess player "Deep Blue" that successfully beat the world chess champion in 1997. However, on the other hand, some problems have turned up. That is, most of the real-world problems cannot be formalized clearly. Rules to be used are often vague and uncertain. The search space is also too large for a computer to find an optimal action in a workable calculation time in most cases of realistically meaningful problems.

Incidentally, in parallel with the development of von Neumann computers, steady researches have been conducted to realize information processing more similar to that in the human brain, i.e., the neural networks. In these days, together with fuzzy processing and genetic algorithms, artificial neural networks are often called soft computing or natural computing.

As shown schematically in Fig. 2.1, neural networks perform pattern processing, which is complementary to symbolic processing used in ordinary AI. The pattern processing deals with pattern information, i.e., information

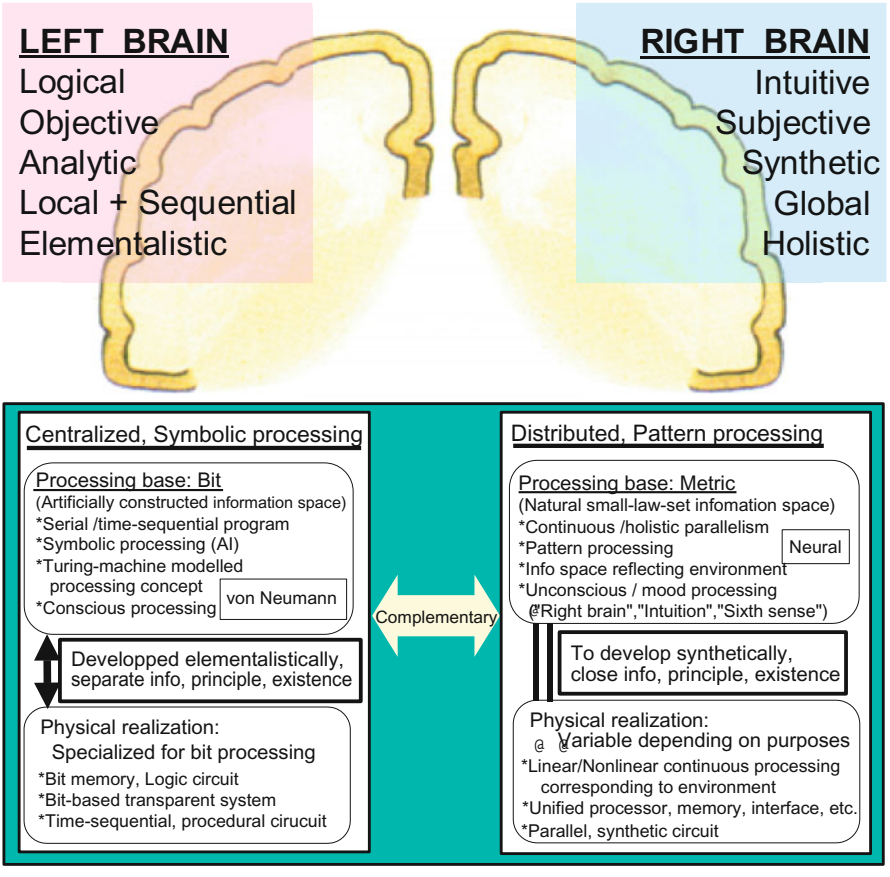


Fig. 2.1 Von Neumann information processing versus neural information processing.

expressed like a picture, holistically all at once. For example, even a baby just after the birth follows moving things with his/her eyes. This action is an unconscious reaction required for him/her to survive in this world. Such non-symbolic and logic-independent processing is frequently observed in human beings on various levels, e.g., simple reaction, complex feeling, and even the sixth sense. On the other hand, most of adults perform also logical thinking such as calculation of change. Contrarily, when we process information time-sequentially, the meaning and operation of the information is defined clearly, we call the treatment symbolic processing.

Certainly, symbolic processing is a powerful method in some application fields. However, in 1990s, it has also been recognized that the symbolic-processing system is often inflexible and inadaptable, i.e., too *hard*.

Conventional AI is increasingly considered incompatible with the human flexibility realized with learning and self-organization.

Information processing useful in the real world should be flexible enough to cope with unexpectedly and dynamically changing environment. As a consequence, such useful processing is considered inseparable from physical existence in the world. Therefore, the AI experiences a paradigm shift in these decades to introduce nonsymbolic processing framework such as neural networks. Such AI is sometimes called "the new AI" [31].

2.2 Physicality of Brain Functions

A strong point of artificial neural networks lies in the adaptability such as learning and self-organizing abilities. Therefore, the processing in the neural networks is largely influenced by the accumulation of experience, i.e., what they have felt and obtained from the environment. We notice this fact in our biological neural network or brain, in the daily life. Artificial neural networks imitate the human brain.

The brain has physicality. Assume that the brain of a person A is replaced by that of person B, while the bodies of A and B are unchanged. Then, which person is more "A-like", i.e., endowed with the property of the person A? We may say that (new A) = (brain A) + (body B) since the brain is the origin of personality. However, the brain has quite a high adaptability. It gradually adapts itself to the new body. The new A gathers information by using former-B's eyes and ears, and tackles the environment with former-B's hands and legs. Then new A should become B-like person. If it is true, the personality is attributed to body rather than brain.

An extreme case can be given as follows. Assume that Mr. A's brain is implanted into a bird, e.g., dove or wild duck. Then, in a short time, the Mr. A's brain learns how to fly with flaps of his hands (wings) in the air. There is more. Since he can fly now, he feels no fear even if he stands on the edge of a cliff. Mr. A's view of the world is completely changed, and it becomes almost a bird's view of the world. Such body-dependent property can be called the physicality of brain functions. That is to say, the brain is brought up with sense organs such as eyes, ears, glossa, nose, cutaneous sensors, etc., to sense the environment, as well as motor organs such as hands, legs, vocal cords, muscles in general, etc., to work on the environment.

The following consideration is also closely related to the physicality. "Excellent human imagination (intelligence) originated from the retardation of about 100ms in human sensorimotor organs with another retardation of about 100ms in motions of hands and legs are because of inertia [32]." The reason is as follows. When a man chases a rapidly running game, say a rabbit, we cannot catch it if we adopt feedback strategy because of unavoidable delays. The rabbit feints against approaching hands to run away. Such a steady effort

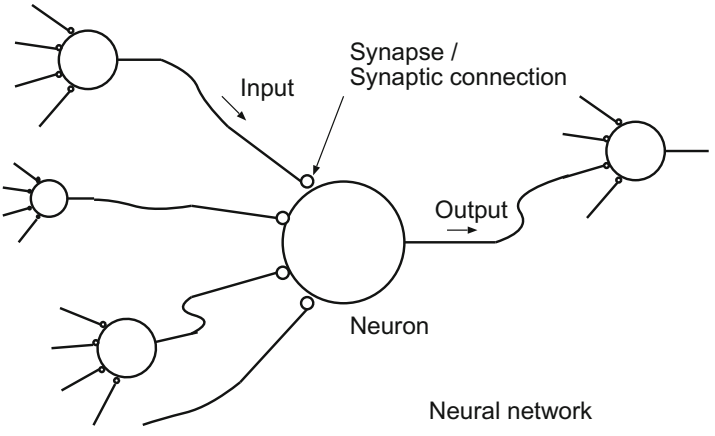


Fig. 2.2 Neural network consisting of neurons

to reduce error doesn't work well. We rather need to chase the rabbit while sensing nebulously the rabbit's action and the surroundings so that we can catch it when it turns around. A small error at normal times does not matter. Instead, we have to forefeel and predict new actions, such as turning around, to be ready such an event. This necessity made the brain develop. That is, the retardation in our sensorimotor system gave rise to the excellent brain functions.

The most important basic idea in complex-valued neural networks is an extension of the physicality mentioned above. If the properties or qualities of sensorimotor organs differ from ordinary human organs, the resulting brain should also be different from the ordinary human brain. When a certain sensor or motor is assumed to be used, there exists a neural-network architecture capable of bringing better up a brain towards expected purpose. Thereby, a brain function useful for the special purpose, different from that of human, emerges effectively and naturally. The complex-valued neural networks architecture expects such functional emergence. It should be a basic framework to yield a brain superior to ordinary one, i.e., a *Superbrain*.

2.3 Neural Networks: General Features

Figure 2.2 shows the basic structure of neural networks schematically. A neural network has all of, or some of, the following features in general.

1. Distributedness and parallelism: Many simple elements similar or identical to one another, i.e., neurons, gather and make connections to communicate, and process information distributedly in parallel.

⟶ Therefore, trouble in a single neuron does not give rise to a fatal impact on the brain function. Although a certain number of human brain cells actually die every day, we can live almost as we did before, being uninfluenced by the death of a small number of neurons.

2. Locality: Information that a neuron can sense is limited locally to input signals fed by other neurons through synaptic connections, internal state of itself, and, in some cases, the state of the neuron to which the output is connected. Connections are also often local.
   ⟶ A neuron processes signals that it can see, and it does not receive any instruction from someone who can look over the system globally. This feature is closely related to the distributedness, and with a high resistance to single trouble. However, note that biological body possesses global mechanisms such as hormone and immunity systems besides neural networks.
3. Weighted sum and activation function with nonlinearity: Input signal is weighted at the synaptic connection by a connection weight. The internal state of a neuron is simply the weighted sum of the input signals or a nonlinear transform of the weighted sum. The nonlinear function is called activation function, and usually has a saturation characteristic.
   ⟶ Accordingly, the process in a neuron is similar to that of an operation unit in historical analog computers such as weighting adders, and also of a logic gate in modern digital computers such as AND, OR, NAND, etc.
4. Plasticity: Connection weights change according to the information fed to the neuron and the internal state. Though the values may be assigned previously, they can vary gradually in the information processing operation. This plasticity of the connection weights leads to *learning* and *self-organization*.
   ⟶ The plasticity realizes the adaptability against the continuously varying environment. In contrast, a nonplastic system has to be designed with prediction of all possible situations in advance, which is impossible to cope with real-world problems.
5. Generalization: A neural network behaves expectedly not only to situations it learned, but also to unlearned situations by inferring an optimal action on the basis of previously learned events by interpolation, extrapolation, etc. It constructs its own view of the world in itself, and responds to unknown environment with its own metric (measure) provided by the world view.
   ⟶ In pattern processing, it is easy and natural to construct the view of the world from experience. The reason is that pattern representation of information is compatible with continuity and continuous metric.

In the following chapters, we present details of the processing dynamics and characteristics of conventional (real-valued) and complex-valued neural networks. Regarding conventional networks, there exist many excellent

books such as Ref.[33] and [34] by T. Kohonen on associative memories and self-organizing maps, Ref.[35] by S. Haykin in signal-processing view points, Ref.[36] by D. Marr on biological visual system, Ref.[37] by C. Mead on hardware, and so on. Please refer to them for details. This book focuses on the complex-valued neural networks by describing conventional networks briefly for easier and intuitive understanding.

3

Complex-Valued Neural Networks: Distinctive Features

Complex-valued neural networks (CVNNs) deal with information in complex domain with complex-valued parameters and variables. As explained in Section 2.2 in relation to physicality, neural functions including learning and self-organization are influenced by sensorimotor interfaces that connect the neural network with the environment. This characteristic is of great importance also in CVNNs. There exist certain situations where CVNNs are inevitably required or greatly effective. In this Chapter, we list such examples and discuss conditions, which will be helpful for readers to grasp what happens in individual dynamics of the CVNNs illustrated in Chapter 4. However, before listing situations, we first discuss the nature of complex number and its effect on the CVNNs. We look back the mathematical history to elucidate the features of complex number, in particular to confirm the importance of the phase-and-amplitude viewpoint for designing and constructing CVNNs to enhance the advantageous features. This viewpoint is essential in general to deal with waves such as electromagnetic-wave and lightwave. Then we point out that, although we may represent a complex number as an ordered pair of real numbers, CVNNs have dynamics different from that of real-valued neural networks. In short, in CVNNs, we can reduce ineffective degree of freedom in learning or self-organization to achieve better generalization characteristics. This merit is significantly useful not only for wave-related signal processing but also for general processing with frequency-domain treatment through Fourier transform. We also explain a matter specific to CVNNs, i.e., the fact that activation functions of CVNNs cannot be analytic. Additionally, at the end of this chapter, we review CVNN researches reported hitherto.

3.1 What Is a Complex Number?

3.1.1 Geometric and Intuitive Definition

In the old days history, the definition of the complex number changed gradually [38]. In the 16th century, Cardano tried to work with imaginary roots in

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 17–56.
springerlink.com

dealing with quadratic and cubic equations. Afterward, Euler used complex numbers in his calculations intuitively and correctly. It is said that by 1728 he knew the transcendental relationship $i \log i = -\pi/2$. The Euler formulae appear in his book as

$$\cos x = \frac{e^{ix} + e^{-ix}}{2} \quad \text{and} \quad \sin x = \frac{e^{ix} - e^{-ix}}{2i} \tag{3.1}$$

It is also believed that, in early 1749, Euler already had a visual concept of complex numbers as points of plane. He described a number x on a unit circle as $x = \cos g + i \sin g$ where g is an arc of the circle. In 1798, Wessel described representation of the points of a plane by complex numbers to deal with directed line segments. Argand also interpreted $\sqrt{-1}$ as a rotation through a right angle in the plane, and justified this idea on the ground that two $\sqrt{-1}$ rotations yields a reflection, i.e., -1. Gauss was in full possession of the geometrical theory by 1815. Furthermore, he proposed to call $+1$, -1, and $\sqrt{-1}$ as direct, inverse, and lateral unity, instead of positive, negative, and imaginary or "impossible" elements, to enhance the substantiality of imaginary number.

3.1.2 Definition as Ordered Pair of Real Numbers

The geometrical representation is intuitively simple and visually understandable, but may be weak in strictness. In 1835, Hamilton presented the formal definition of the complex number as an "ordered pair of real numbers," which also led to the discovery of quaternions, in his article entitled "Theory of conjugate functions, or algebra as the science of pure time." He defined addition and multiplication in such a manner that the distributive, associative, and commutative laws hold. The definition as the ordered pair of real numbers is algebraic, and can be stricter than the intuitive rotation interpretation.

At the same time, the fact that a complex number is defined by two real numbers may lead present-day neural-network researchers to consider a complex network equivalent in essence to just a real-number network that has double real input terminals and real double output neurons. However, it is not true. We can clarify the merit by focusing on the rotational function even with this definition.

Based on the definition of the complex number as an ordered pair of real numbers, we represent a complex number z as

$$z \equiv (x, y) \tag{3.2}$$

where x and y are real numbers. Then the addition and multiplication of z_1 and z_2 are defined in *complex domain* as

$$(x_1, y_1) + (x_2, y_2) \equiv (x_1 + x_2, y_1 + y_2) \tag{3.3}$$

$$(x_1, y_1) \cdot (x_2, y_2) \equiv (x_1 x_2 - y_1 y_2, x_1 y_2 + y_1 x_2) \tag{3.4}$$

As a reference, the addition and multiplication (as a step in the calculation of inner product, for example) of *two-dimensional real values* is expressed as

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2) \tag{3.5}$$
$$(x_1, y_1) \cdot (x_2, y_2) = (x_1 x_2, y_1 y_2) \tag{3.6}$$

In the comparison, the addition process is identical. Contrarily, the complex multiplication seems quite artificial, but this definition (3.4) brings the complex number with its unique function, that is, the angle rotation, as well as amplitude amplification / attenuation, which are the result of the intermixture of the real and imaginary components.

It is easily verified that the commutative, associative, and distributive laws hold. We have the unit element $(1, 0)$ and the inverse of z $(\neq 0)$ which is

$$\begin{aligned} z^{-1} &\equiv \left(\frac{x}{x^2 + y^2}, \frac{-y}{x^2 + y^2} \right) \\ &= \left(\frac{x}{|z|^2}, \frac{-y}{|z|^2} \right) \end{aligned} \tag{3.7}$$

where $|z| \equiv \sqrt{x^2 + y^2}$.

3.1.3 Real 2×2 Matrix Representation

We can also use real 2×2 matrices, instead of the ordered pairs of real numbers, to represent complex numbers [38] [39] . With every complex number $c = a + ib$, we associate the $\boldsymbol{C}$-linear transformation

$$T_c : \boldsymbol{C} \to \boldsymbol{C}, \qquad z \mapsto cz = ax - by + i(bx + ay) \tag{3.8}$$

which includes a special case of $z \to iz$ that maps 1 into i, i into -1, ..., with a rotation with right angle each. In this sense, this definition is a more precise and general version of Argand's interpretation of complex numbers. If we identify $\boldsymbol{C}$ with $\boldsymbol{R}^2$ by

$$z = x + iy \;=\; \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.9}$$

it follows that

$$\begin{aligned} T_c \begin{bmatrix} x \\ y \end{bmatrix} &= \begin{bmatrix} ax - by \\ bx + ay \end{bmatrix} \\ &= \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \end{aligned} \tag{3.10}$$

In other words, the linear transformation T_c determined by $c = a + ib$ is described by the matrix $\begin{bmatrix} a & -b \\ b & a \end{bmatrix}$. Generally a mapping represented by a 2×2

matrix is non-commutative. However, in the present case, it becomes *commutative*. By this real matrix representation, the imaginary unit i in $\boldsymbol{C}$ is given as

$$I \equiv \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad , \quad I^2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} = -E \tag{3.11}$$

In the days of Hamilton, we did not have matrices yet. Even present, it is very rare to define complex numbers in terms of real 2×2 matrices [38] (Chapter 3, §2, 5.), [39] . The introduction of complex numbers through 2×2 matrices has the advantage, over introducing them through ordered pairs of real numbers, that it is unnecessary to define an ad hoc multiplication. What is most important is that this matrix representation clearly expresses the function specific to the complex numbers. That is, the rotation and amplification or attenuation as

$$\begin{bmatrix} a & -b \\ b & a \end{bmatrix} = r \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{3.12}$$

where r and θ denote amplification / attenuation of amplitude and rotation angle applied to signals, respectively, in the multiplication calculation. On the other hand, addition is rather plain. The complex addition function is identical to that in the case of doubled-dimension real numbers.

In summary, the phase rotation and amplitude amplification / attenuation are the most important features of complex numbers. The significance is described in the following sections.

3.2 Comparison of Complex- and Real-Valued Feedforward Neural Networks

3.2.1 Function of Complex-Valued Synapse and Network Operation

In wave-related adaptive processing, we often obtain excellent performance with learning or self-organization based on the CVNNs. As already mentioned, the reason depends on situations. However, the discussion in Section 3.1 suggests that the origin lies in the complex rule of arithmetics. That is to say, the merit arises from the functions of the four fundamental rules of arithmetics of complex numbers, in particular the multiplication, rather than the representation of the complex numbers, which can be geometric, algebraic, or in matrices. Moreover, the essence of the complex numbers also lies in the characteristic multiplication function, the phase rotation, as overviewed in Section 3.1 [17] [40] [41].

Assume a task to realize a mapping that transforms an input $\boldsymbol{x^{IN}}$ to an output $\boldsymbol{x^{OUT}}$ shown in Fig.3.1. Let us consider a very simple single layer

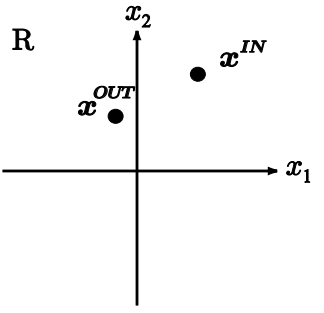


Fig. 3.1 A task to learn a mapping that maps $\boldsymbol{x}^{IN}$ to $\boldsymbol{x}^{OUT}$ [40].

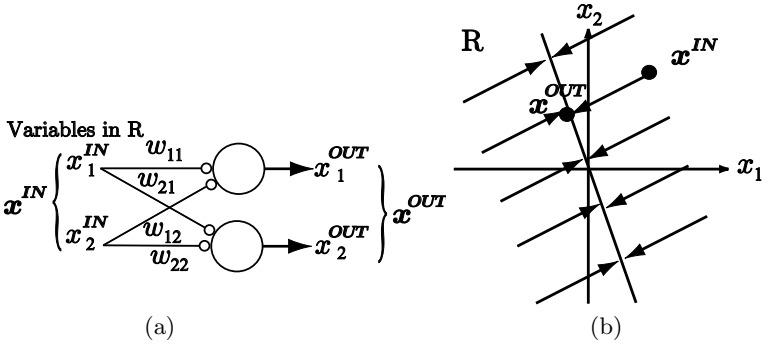
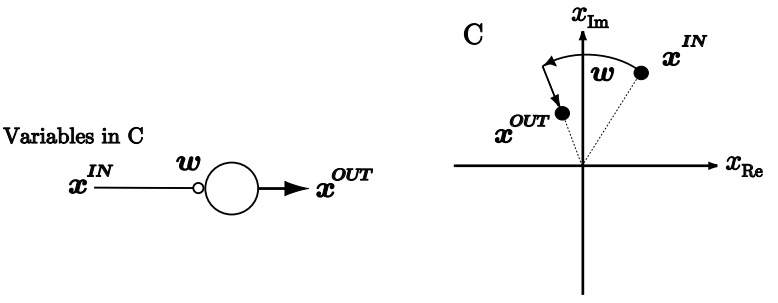


Fig. 3.2 A simple linear feedforward network to learn the task given in Fig.3.1: (a)a real-valued single-layered two-input two-output feedforward network and (b)a possible but degenerate solution that is often unuseful [40].

2-input 2-output feedforward neural network in real number as shown in Fig.3.2(a). For simplicity, we omit the possible nonlinearity at the neurons, i.e., the activation function is the identity function, where the neurons have no threshold. We train the network through supervised learning that adjusts the synaptic weights w_{ji}. Simply we have only a single teacher pair of input and output signals. Then we can describe a general input-output relationship as

$$\begin{bmatrix} x_1^{OUT} \\ x_2^{OUT} \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1^{IN} \\ x_2^{IN} \end{bmatrix} \tag{3.13}$$

We have a variety of possible mapping obtained by the learning because the number of parameters to be determined is larger than the condition, i.e., the learning task is an ill-posed problem. The functional difference emerges as the difference in the generalization characteristics. For example, learning can result in a degenerate mapping shown in Fig.3.2(b), which is often unuseful in practice.



(a) (b)

Fig. 3.3 Another simple linear feedforward network to learn the same task given in Fig.3.1: (a)complex-valued neural network seemingly identical to Fig.3.2(a), and (b)a solution obtained in this small degree-of-freedom case [40].

Next, let us consider the mapping learning task in the one-dimensional complex domain, which transforms a complex value $\boldsymbol{x}^{\boldsymbol{IN}} = \boldsymbol{z}^{\boldsymbol{IN}} = (x_1^{\boldsymbol{IN}}, x_2^{\boldsymbol{IN}})$ to another complex value $\boldsymbol{x}^{\boldsymbol{OUT}} = \boldsymbol{z}^{\boldsymbol{OUT}} = (x_1^{\boldsymbol{OUT}}, x_2^{\boldsymbol{OUT}})$. Figure 3.3(a) shows the complex-valued network, where the weight is a single complex value. The situation is expressed just like in (3.13) as

$$\begin{bmatrix} x_1^{\boldsymbol{OUT}} \\ x_2^{\boldsymbol{OUT}} \end{bmatrix} = \begin{bmatrix} |w|\cos\theta & -|w|\sin\theta \\ |w|\sin\theta & |w|\cos\theta \end{bmatrix} \begin{bmatrix} x_1^{\boldsymbol{IN}} \\ x_2^{\boldsymbol{IN}} \end{bmatrix} \tag{3.14}$$

where $\theta \equiv \arg(w)$. The degree of freedom is reduced, and the arbitrariness of the solution is also reduced. Figure 3.3(b) illustrates the result of the learning. The mapping is a combination of phase rotation and amplitude attenuation. This example is truly an extreme. The dynamics of a neural network is determined by various parameters such as network structure, input–output data dimensions, and teacher signal numbers. However, the above characteristics of phase rotation and amplitude modulation are embedded in the complex-valued network as a universal elemental process of weighting.

The essential merit of neural networks in general lies in the high degree of freedom in learning and self-organization. However, if we know *a priori* that the objective quantities include "phase" and/or "amplitude," we can reduce possibly harmful portion of the freedom by employing a complex-valued neural network, resulting in a more meaningful generalization characteristics. The "rotation" in the complex multiplication works as an elemental process at the synapse, and realizes the advantageous reduction of the degree of freedom. This feature corresponds not only to the geometrical intuitive definition of complex numbers but also to the Hamilton's definition by ordered pairs of real numbers, or the real 2×2 matrix representation.

Though we considered a small feedforward network in this section, the conclusion is applicable also to other CVNNs such as complex-valued Hebbian-rule-based network and complex correlation learning networks, where

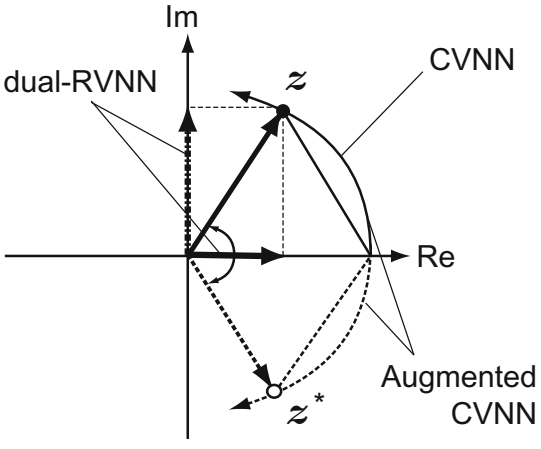


Fig. 3.4 Conceptual illustration of the relationship between bases in the respective neural networks to deal with complex signal z [41].

the weight is updated by the multiplication results. The elemental process of phase rotation and amplitude modulation results in the network behavior consistent with phase rotation and amplitude modulation in total. The nature is a great advantage when we deal with not only coherent signals and waves such as electromagnetic wave and lightwave, but also arbitrary signals with the Fourier synthesis principle, or in the frequency domain through the Fourier transform.

3.2.2 Circularity and Widely-Linear Systems

The circularity of the signals to be processed is also an important factor in CVNNs [41]. To deepen the discussion, we refer to the wide sense linear (or widely linear: WL) systems which introduce conjugate signals in addition to direct complex signals [42] [43] . WL systems well learns complex data distributed anisotropically in the complex plane, i.e., noncircular data. For example, it is useful for predict wind strength and direction by assuming the axes of the complex-number plane represent north, south, east and west, and the distance from the origin expresses the strength. Augmented complex-valued neural networks have been proposed in such a context [44] . Wind has high anisotropy in general. The augmented complex-valued networks do not lead to the reduction of the degree of freedom. The degree is the same as that of parallel real-valued networks to process real and imaginary parts of signals separately, or their variations. Accordingly the dynamics becomes similar to that of real-valued ones. A conceptual illustration is given in Fig. 3.4.

More accurately, the number of the signal-representation bases of the augmented complex networks is the same as that of the real-valued networks, and

its dynamics approaches that of real-valued neural networks. This situation is analogous to the fact that the combination of positive and negative frequency spectra generates almost real-valued signals. We can also compare the relationship to polarization of lightwave. CVNNs deal with only one of the left- or right-handed circular polarized lightwave, and are suitable for circular signal processing. Note that the signal in total can be out of complete circularity, but only each frequency component has the circularity. Since any waveform can be synthesized by sinusoidal components through Fourier synthesis, the signals that the CVNNs can deal with is not limited to completely coherent signals. In contrast, the augmented complex-valued networks deal with both the left- and right-handed circular polarized lightwave. They are more flexible because of the larger degree of freedom, which is too much for circular signals. The number of signal representation bases is the same as that of real-valued neural networks though, in the real-valued case, the bases are horizontal and vertical linear polarizations corresponding to the real and imaginary parts. The relationship is similar to that of complex-valued filtering and dual univariate real-valued filtering, which processes real and imaginary parts of signals separately in two real-valued filters in parallel [45]. Real-valued layered neural networks having various structures can be equivalently range from, so to say, dual univariate real-valued neural networks, where the real and imaginary signals are processed separately, to completely bivariate real-valued neural networks, where the signals are processed as a mixture all over the layers with a still higher flexibility originating from massive neural connections.

Consequently, complex-valued neural networks are suitable for processing analytic signals, which consist of a real component and its consistent imaginary component that has the same amplitude but 90-degree shifted phase. The details are given in Section 3.6.1. The analytic signal is essentially circular. Analytic signals exist widely in electronics, for example, at the output of heterodyne or homodyne mixers and at the output of digital signal processing using the Hilbert transform. CVNNs have a higher generalization ability to process such analytic signals appropriately.

Noncircular signals are observed in, e.g., wind information processing, but less in electronics dealing with electromagnetic wave and related time-sequential signals. The reason lies in the fact that phase does not have any meaning in its absolute value, but only in its difference from a certain reference. However, a few exceptions may exist. For example, signals in offset quadrature phase shift keying (OQPSK) modulation may include nonnegligible noncircularity generated by highly unbalanced electronics.

3.2.3 Nonlinearity That Enhances the Features of Complex-Valued Networks

The neuron nonlinearity can be another issue. We describe here the relationship between the nonlinearity and the neural dynamics mentioned above. The detail will be discussed in Section 3.3.

The complex least mean square (LMS) algorithm is the most widely-used basis of adaptive processing of complex signals [46]. The introduction of nonlinearity into the neuron activation function once seemed to have a serious problem in the differentiability in the complex domain. Liouville's theorem in complex analysis states that every entire function (holomorphic function) must be constant. It follows that, if we introduce some nonlinearity, we have to abandon the differentiability. This fact was considered to be a big problem at around 1990 because some researchers believed that the indifferentiability should lead directly to the impossibility to obtain and/or analyze the dynamics of the CVNNs.

However, the concern was found to be a trifle because neural dynamics are generally described by partial differentiation in terms of a number of variables associated with the neurons. Actually, nowadays we can determine neural dynamics in CVNNs by calculating partial differentials in terms of real and imaginary parts, or phase and amplitude. This manner is practically effective.

At the same time, it is true that we discard the conformal mapping nature of the holomorphic function. However, when we utilize a conformal mapping function, we often concentrate upon the mapping structure itself, rather than a combination with some nonlinearity. Additional nonlinearity should rather be a hindrance. Accordingly, the non-holomorphy is not a big problem again.

In complex-valued associative memories, researchers investigated the requirements necessary in the nonlinearity to determine an effective energy function [47]. As a result, roughly speaking, we have two types of possibility. One is to apply nonlinearity to real and imaginary parts separately and to combine them to yield a complex output (real–imaginary nonlinearity) [48] [49]. The other is to employ nonlinear functions for the phase and amplitude, respectively (amplitude–phase nonlinearity) [50].

In other CVNNs, we may have possibilities to employ other nonlinearity depending on the objectives, i.e., what type of processing we aim at. Even in such cases, the above-mentioned two types of nonlinearity will be the most promising candidates since we normally consider that a direct extension of the real sigmoid function works well also widely in complex domain, instead of complex sigmoid function itself.

When we deal with wave information or wave itself, the real and imaginary axes are essentially less meaningful than amplitude and phase (or phase difference) because the real and imaginary axes are determined only relatively to an arbitrarily determined phase reference. An example is the coherent detection in communications receiver, where we prepare a local oscillator (LO) with a phase-locked loop (PLL) locked to some reference to be used for demodulation, that is, extraction of real and imaginary signals. The receiver determines the real and imaginary parts, which never exist beforehand [50]. Instead, the difference of two phase values are meaningful itself, which corresponds to time course and/or position difference. In this sense, the phase difference represents certain information directly. The amplitude, orthogonal to phase, is also meaningful, signifying energy or power of the wave.

Accordingly, in this sense, the amplitude–phase nonlinearity is more suitable for wave-related processing. Part II presents various adaptive systems based on the amplitude–phase nonlinearity such as the optical learning logic circuits realizing frequency-multiplexed operation [51] [52] and the fast method to yield computer-generated hologram (CGH) for three-dimensional movies [53] [54].

3.3 Activation Functions in Neurons

3.3.1 Nonlinear Activation Functions in Real-Valued Neural Networks

In Section 2.3, we enumerated the features in artificial neurons and neural networks. A neuron weights input signals, and sums up the weighted inputs. Then a function provided in the neuron, namely activation function, yields an output signal. As the activation function, we often adopt a nonlinear function. (N.B., it can be a linear function in general.)

Figure 3.5 shows our neuron model with the input signal x_i, connection weight w_i, internal state $u \equiv \sum w_i x_i$, activation function f, and output signal y.

$$y = f(u) = f\left(\sum_{i=1}^{N} w_i x_i\right) \tag{3.15}$$

In many real-valued neural networks, f is a sigmoid function (saturation function with "S" shape) such as

$$f(u) \equiv \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}} \tag{3.16}$$

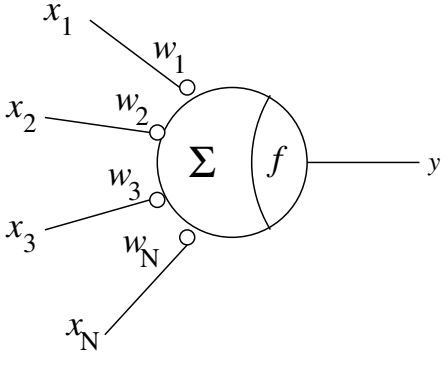


Fig. 3.5 A neuron with input signal x_i, connection weight w_i, activation function f, and output signal y.

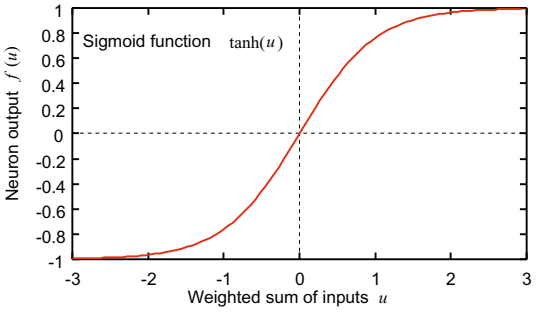


Fig. 3.6 Activation function $\tanh(u)$ widely used in real-valued neural networks.

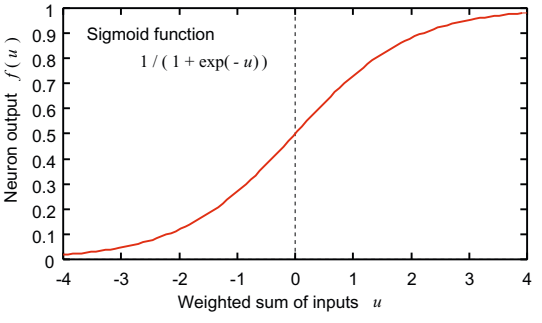


Fig. 3.7 Another activation function $1/(1+\exp(-u))$ widely used in real-valued neural networks.

The function is called hyperbolic tangent, and its shape is shown in Fig.3.6. In biological neurons, when it receives larger input signals (higher pulse frequency), the output signal (pulse frequency) becomes also higher in a saturation manner. The activation function in artificial neural networks imitates such a characteristic. The nonlinearity is effective in some tasks such as function approximation described in Chapter 4. Another activation function, similar to the above one and widely used, is

$$f(u) \equiv \frac{1}{1+\exp(-u)} \tag{3.17}$$

The shape is shown in Fig.3.7. Except for the value range of [0,1], the function is analogous with the first one. The value range can be transformed to any desirable one such as [−1,1] or [0,1] by shifting and scaling.

In any case, we often introduce a gain coefficient g and/or saturation amplitude A to modify the function as $f(u) = A\tanh(gu)$, or $f(u) = A/(1+\exp(-gu))$, to control the slope and saturation value of the sigmoid

functions. Increase in the gain g yields a steeper slope, and when $g \longrightarrow \infty$, the sigmoid function approaches the step function.

3.3.2 Problem Concerning Activation Functions in Complex-Valued Neural Networks

A problem arises when we extend the sigmoid function to a complex-valued saturation function to construct a CVNN. The problem is that such a nonlinear function *cannot* be analytic. A complex function is called *analytic* when it is differentiable at any point. A complex function is differentiable at a point u in complex plane if we can determine the limit value of $(f(u+\Delta u)-f(u))/\Delta u$ when $\Delta u \to 0$ for the function $f(u)$ of complex variable u, i.e., the limit value does not depend on the direction in which the variable approaches to the point u.

When a function considered is differentiable at a point, we also say that the function is regular at the point (regular point). Contrarily, a point at which the function is not differentialble is a singular point. Being analytic means that the function is regular at any point in the domain considered. If the activation function is an analytic function (holomorphic function in other words), we can analyze the neural dynamics such as learning, self-organization, and processing, to understand the characteristics of CVNNs, in the same manner as we investigate the dynamics of conventional neural networks.

This problem of being nonanalytic is equivalent to the Liouville's theorem. That is, if a complex function is analytic at any point and bounded, it is a constant function.

Let us consider that the variable u in (3.16) is complex. Then the function $f(u)$ is differentiable at almost any point. However, it diverges to infinity. Figure 3.8 shows the shape of the function, which is far away from the "saturation" feature. Although the complex $\tanh(u)$ in complex domain may be a natural extension of real-valued $\tanh(u)$ in the sense that the variable is extended into complex one, the meaning of the nonlinearity is completely different from something saturating. Therefore, we cannot construct useful system with the complex $\tanh(u)$.

This issue was the most serious reason that the CVNNs were considered difficult to develop before. Figure 3.9 shows complex $1/(1+\exp(-u))$ in (3.17) in complex domain. We find the same problem again.

3.3.3 Construction of CVNNs with Partial Derivatives in Complex Domain

The problem mentioned above is avoided at present as follows. When we introduce a nonlinear activation function (N.B., we may use a linear function), we do not pay attention to differentiability. It does not matter whether the

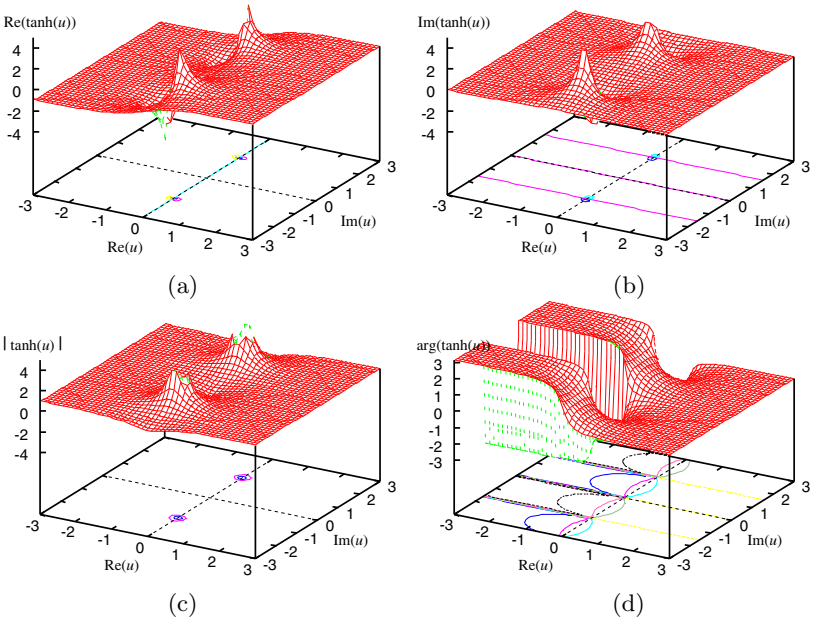


Fig. 3.8 Complex $\tanh(u)$ as a function of complex variable u: (a)real part, (b)imaginary part, (c)amplitude, and (d)phase.

function is analytic or not. Thus, we do not expect rigorous analyses similar to those in investigations in real-valued neural networks[1]. Instead, we construct the dynamics of learning and self-organization on the basis of meaningful partial derivatives in complex domain with deliberation on the properties of input and output information.

Actually, this veer of way of thinking led to constructing many useful CVNNs. Widely used are the following two activation functions. Needless

[1]However, practically speaking, we can conduct analyses almost as rigorous as those in conventional neural networks. The reason is as follows. As we see in the following sections, the dynamics of neural networks, that are parallel and distributed systems, is analyzed mostly with partial differentiation. Note that neural networks often deal with modally different variables at once. In the same way, we have no difficulty in dealing with variables different in the characteristic nature from one another, such as amplitude and phase, which we will see in the following sections. Contrarily, if we adopt an analytic one as an activation function, we can analyze the dynamics analytically. However, an analytic function cannot be a meaningfully nonlinear function, but a power series at most. Its whole shape is determined rigidly by its characteristic on a small local region in complex domain (theorem of identity), which sometimes reduces the function's attractiveness because of the hardness in resulting generalization characteristics. Note that, however, the rigidness is useful when we utilize the conformal nature.

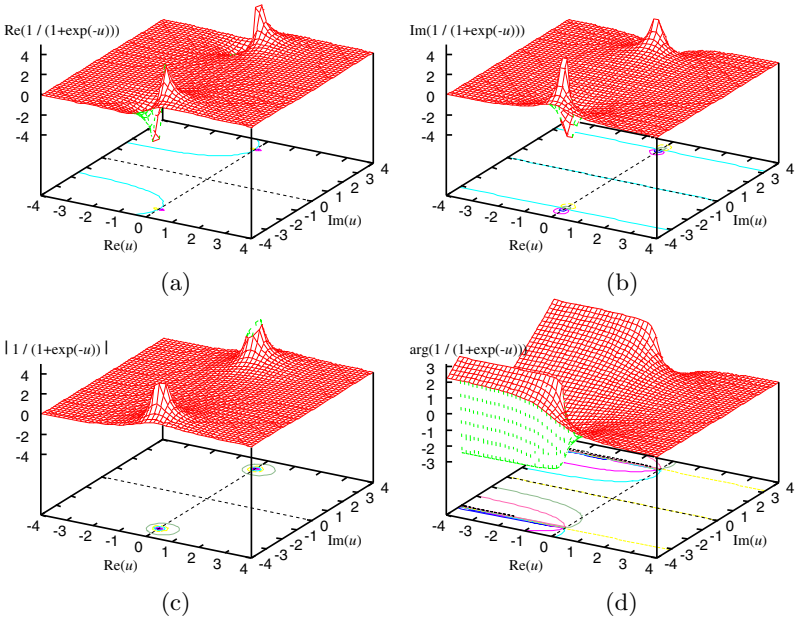


Fig. 3.9 Complex $1/(1+\exp(-u))$ as a function of complex variable u: (a)real part, (b)imaginary part, (c)amplitude, and (d)phase.

to say, activation functions are not limited to these two. Instead, we can introduce functions suitable for processing purpose, including linear ones.

It is true that the above change of thinking direction caused a coordinate dependence of the neural dynamics. In other words, the neural learning and self-organization can be variable related to the direction of the basis in partial differentiation. For example, in most applications presented in Part II adopt polar coordinate system where the bases are taken in the radial and angular directions in the complex domain. In some other cases, we may adopt Cartesian coordinate system (orthogonal line coordinate system).

Mechanics in its wide sense, including neural networks, should be independent of coordinate in general. However, the coordinate dependence of CVNN dynamics is rather favorable to construct a useful network. Consider a CVNN application related to electromagnetic wave or lightwave. Then the CVNN will deal with the wave by controlling modulators such as amplitude and phase modulators. The modulators are influential basically to energy or delay time of the wave. In robotics with periodic motion, we will also manipulate the motion's amplitude, phase, and frequency. In such a way, it is very important to clarify what properties a CVNN directly deals with in the interaction with the real world. Thus a CVNN should turn the coordinate dependence of the dynamics into an advantage, i.e., it should utilize the dependence. We will see such examples in the contrast between the

real-imaginary-type and amplitude-phase-type activation functions presented in the next section.

3.3.4 Real-Imaginary-Type Activation Function

One of the widely used activation functions is

$$f_{\mathrm{re-im}}(u) \equiv f_{\mathrm{R}}(u) + i f_{\mathrm{I}}(u) \tag{3.18}$$

$$f_{\mathrm{R}}(u) \equiv \tanh(\mathbf{Re}(u)) \tag{3.19}$$

$$f_{\mathrm{I}}(u) \equiv \tanh(\mathbf{Im}(u)) \tag{3.20}$$

where f_{R} and f_{I} are real and imaginary parts of the activation function, while $\mathbf{Re}(u)$ and $\mathbf{Im}(u)$ are real and imaginary parts of complex variable u, respectively. Here, the function tanh is a real-valued hyperbolic tangent. We call the complex function in (3.18) real-imaginary-type activation function.

Figure 3.10 shows the shape of the real-imaginary-type activation function. It has line symmetry regarding real and imaginary axes (lines of $\mathbf{Im}(u) = 0$ and $\mathbf{Re}(u) = 0$, respectively), which is shown in the shape in Figs.3.10(a) and(b), i.e., the real and imaginary parts. We can also observe characteristic changes on and around the real and imaginary axes in Figs.3.10(c)amplitude

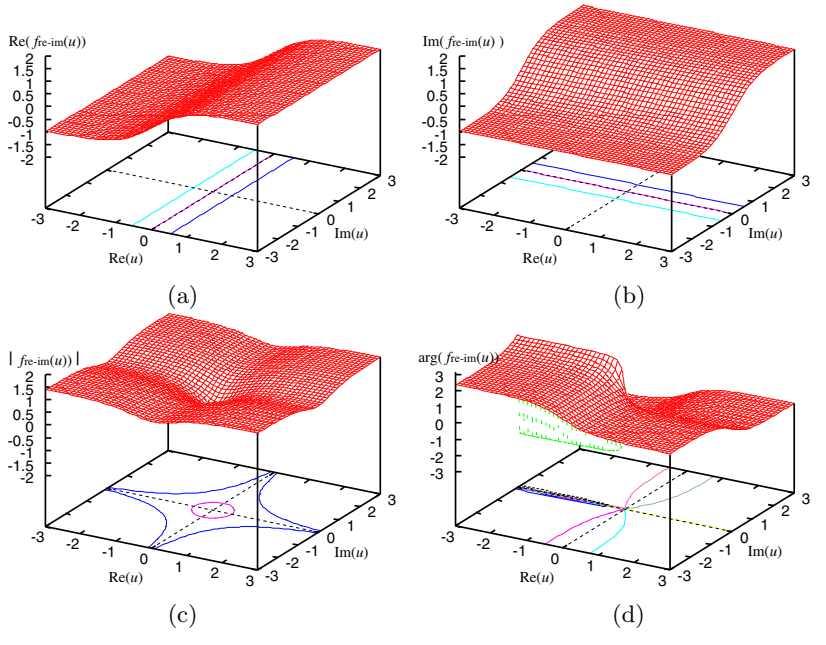


Fig. 3.10 Real-imaginary-type complex activation function: (a)real part, (b)imaginary part, (c)amplitude, and (d)phase.

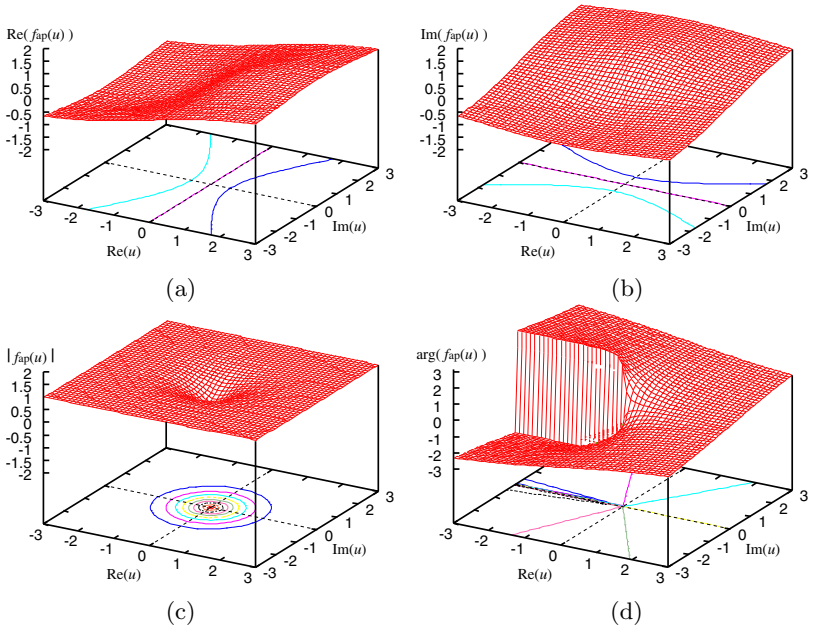


Fig. 3.11 Amplitude-phase-type activation function: (a)real part, (b)imaginary part, (c)amplitude, and (d)phase.

and (d)phase. The symmetry gives the two axes a special meaning in the neural dynamics.

Therefore, we can expect that the real-imaginary-type activation function works well when we deal with complex information that must have symmetry concerning, or a certain special meaning on, the real and imaginary axes. It is also implied that a network processing n-dimensional information with this activation function has neural dynamics slightly similar to that of a real-valued neural network processing $2n$-dimensional real-valued information because, as shown in (3.19) and (3.20), it deals with real and imaginary parts separately and independently.

3.3.5 Amplitude-Phase-Type Activation Function

Another one widely used is amplitude-phase-type activation function expressed as

$$f_{\mathrm{ap}}(u) \equiv \tanh\left(|u|\right)\ \exp\left(i\arg(u)\right) \tag{3.21}$$

The definition (3.21) means saturation in amplitude, whereas the phase is unchanged.

Figure 3.11 shows the shape. It has point symmetry concerning the origin $(0, i0)$, which is clearly observed in Figs. 3.11(c) and (d). In comparison

with the real-imaginary-type activation function, the amplitude-phase-type function is independent of the way of setting of real and imaginary axes. Therefore, it is suitable for processing information meaningful in rotation around the origin of coordinate.

Among others, it is suitable for processing waves or wave-related information. We assume that the wave amplitude corresponds to the amplitude of the complex variable in the neural network, while wave phase does to the phase of the neural variable. Then the saturation characteristic of the nonlinear function can be related to the saturation of wave energy, which is widely observed in various physical phenomena. (We deal with possible saturation-related weak nonlinearity in phase separately from this treatment.[2]) On the other hand, the wave phase rotates in accordance with the progress or delay of time. The phase is a physical entity itself, though we may observe real or imaginary part of wave in measurement. Therefore, the radial isotropy concerning coordinate origin in the amplitude-phase-type activation function is desirable in dealing with waves.

Accordingly, we use the amplitude-phase-type activation function when we deal with electromagnetic wave, lightwave, sonic wave, ultrasonic wave, quantum waves such as electron wave, and other wave-related phenomena. In electronics, we have various wave-related applications, as shown schematically in Fig.1.2 (Page 7) in Chapter 1, such as remote sensing, radar imaging, adaptive antennas and beamforming, adaptive mobile communications, lightwave information processing, lightwave routing, lighwave sensing, ultrasonic imaging and diagnosis, speech analysis and synthesis, quantum devices, quantum computing, and so forth.

The origin of the affinity of the amplitude-phase-type activation function for waves lies in the direct treatment of basic physical entities, i.e., energy and time (progress and delay) as described in Section 1.3 (application fields). This fact also implies the compatibility with information that human beings receive from the world because our sensory organs receive information as *physical* stimuli. As explained in Chapter 2 (general features of neural networks), to cope with real world problems, significantly important is the information processing inseparable from physical entities.

Additionally, the amplitude-phase-type function is useful to deal with time-sequential signals and space-sequential ones (e.g., images) in frequency domain with the help of Fourier transform or wavelet transform. This is because the frequency-domain treatment assumes that any signal can be decomposed into sinusoidal waves. Frequency-domain treatment using complex-valued neural networks is of great importance because it enables, or simplifies, various types of information processing.

[2]Note that, however, there exist exceptions. For example, when the environment is more or less related to resonance phenomena, we will utilize the relationship between amplitude and phase (Kramers-Kronig relationship). The Hilbert transform equivalent to the relations is also used to obtain so-called analytic signals in processing baseband signals as described in Section 3.6.1.

The amplitude-phase-type activation function is also suitable for applications where we find essence in point symmetry concerning origin, or in polar coordinate expression of the environment. Focusing on phase, we can say that utilizing periodic topology is also very important, which we describe in detail in Section 3.6. Besides above mentioned ones, many applications shown in Fig.1.2 in Chapter 1 are compatible with the amplitude-phase-type activation function.

3.4 Metric in Complex Domain

3.4.1 Importance of Metric: An Example in Complex-Valued Self-organizing Map

Among various neurodynamics in the complex domain, the complex-valued self-organizing maps (CSOMs), to be explained in Section 4.5, may possess less features which reflect the complex multiplication mentioned in 3.2, since most of SOMs have two sub-processes in the operation, i.e., winner determination and weight update, both of which sub-processes may consist of only addition and subtraction in its arithmetics without any multiplication that utilizes the complex nature of phase rotation.

However, the circumstances depend on the metric we use to determine the dynamics. If we employ complex inner product, instead of conventional Euclidean metric in double-dimensional real space, we can utilize the characteristics specific to complex space [55]. The general dynamics of a SOM will be explained in Section 4.5. In this section, we discuss the metric we use in feature vector space.

3.4.2 Euclidean Metric

In SOM in general, the metric most widely used to determine the *winner* neuron whose weight $\boldsymbol{w}_c$ is nearest to an input feature vector $\boldsymbol{z}$ is the Euclidean metric. Even in a complex-valued SOM (CSOM) where $\boldsymbol{z}$ and $\boldsymbol{w}$ are complex, we can express them with imaginary unit i as

$$\boldsymbol{z} \equiv \begin{bmatrix} |z_1| \exp(i\theta_1) \\ |z_2| \exp(i\theta_2) \\ \vdots \end{bmatrix} \tag{3.22}$$

$$\boldsymbol{w}_c \equiv \begin{bmatrix} |w_{c\,1}| \exp(i\psi_{c\,1}) \\ |w_{c\,2}| \exp(i\psi_{c\,2}) \\ \vdots \end{bmatrix} \tag{3.23}$$

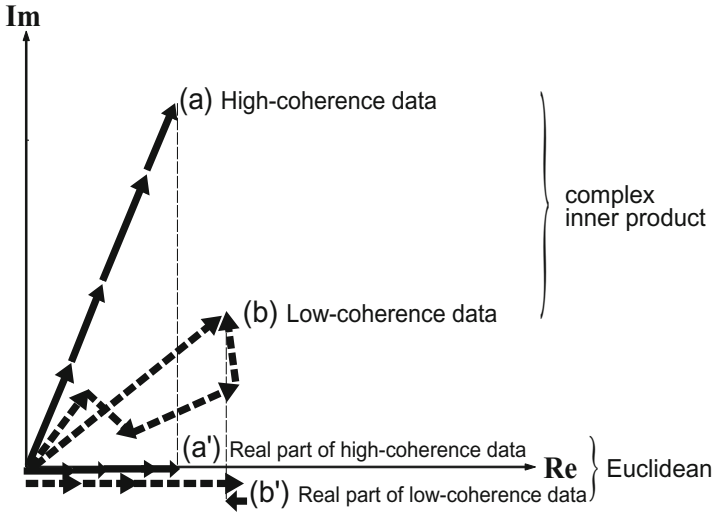


Fig. 3.12 Conceptual illustration to compare the inner product $\boldsymbol{z}^*\boldsymbol{w}_c$ and the real-part inner product $\mathbf{Re}(\boldsymbol{z}^*\boldsymbol{w}_c)$ to calculate $||\boldsymbol{z}-\boldsymbol{w}_c||^2$ [55].

The Euclidean process to choose a winner is expressed as

$$\hat{c} = \arg\min_c ||\boldsymbol{z}-\boldsymbol{w}_c|| \quad (c : \text{class index}) \tag{3.24}$$

where $\arg\min_c \cdots$ chooses a c that minimizes $\cdots$, and $||\cdot||$ denotes norm (amplitude), i.e.,

$$\begin{aligned} ||\boldsymbol{z}-\boldsymbol{w}_c||^2 &= (\boldsymbol{z}-\boldsymbol{w}_c)^* \ (\boldsymbol{z}-\boldsymbol{w}_c) \\ &= ||\boldsymbol{z}||^2 + ||\boldsymbol{w}_c||^2 - (\boldsymbol{z}^*\boldsymbol{w}_c + \boldsymbol{w}_c^*\boldsymbol{z}) \\ &= ||\boldsymbol{z}||^2 + ||\boldsymbol{w}_c||^2 - 2\mathbf{Re}(\boldsymbol{z}^*\boldsymbol{w}_c) \end{aligned} \tag{3.25}$$

Though (3.25) deals with complex numbers, this arithmetic is identical with the calculation of real-valued Euclidean distance and also of the real-valued inner product, i.e., when $\boldsymbol{x}, \boldsymbol{w}_c \in \mathbf{R}^m$,

$$\begin{aligned} ||\boldsymbol{x}-\boldsymbol{w}_c||^2 &= ||\boldsymbol{x}||^2 + ||\boldsymbol{w}_c||^2 - 2\boldsymbol{x}^T\boldsymbol{w}_c \\ &= ||\boldsymbol{x}||^2 + ||\boldsymbol{w}_c||^2 - 2\sum_i |x_i||w_{c\,i}|\cos(\psi_{c\,i} - \theta_i) \end{aligned} \tag{3.26}$$

Then, when $||\boldsymbol{z}||^2$ and $||\boldsymbol{w}_c||^2$ are almost constants, as is often the case, and/or we pay attention to phase information, the distance (3.26) is detemined by the cosine component (real part) $|z_i||w_{c\,i}|\cos(\psi_{c\,i} - \theta_i)$.

3.4.3 Complex-Valued Inner-Product Metric

Instead, we can also employ a complex inner-product metric for use in determination of a winner in the CSOM as

$$\hat{c} = \arg\max_{c} \left(\left| \frac{\boldsymbol{z}^* \ \boldsymbol{w}_c}{||\boldsymbol{z}||||\boldsymbol{w}_c||} \right| \right) \quad (c : \text{class index}) \tag{3.27}$$

This process is better understandable in equations by employing the polar representation. That is, the numerator of the complex-valued inner product (3.27) is given as

$$\boldsymbol{z}^* \ \boldsymbol{w}_c = \sum_i \left(|z_i| \exp(-i\theta_i) \right) \left(|w_{c\ i}| \exp(i\psi_{c\ i}) \right) \tag{3.28}$$

$$= \sum_i |z_i||w_{c\ i}| \exp(i(\psi_{c\ i} - \theta_i)) \tag{3.29}$$

where the summation takes into account the phase values directly, that is, the direction of the arrows [55].

In other words, the metric (3.29) takes both the cosine and sine components (real and imaginary parts) into consideration. That is, when we express the vectors as $\boldsymbol{z} = (x_1 + iy_1, x_2 + iy_2, ...)$ and $\boldsymbol{w} = (u_1 + iv_1, u_2 + iv_2, ...)$, omitting suffix c, we obtain

$$\begin{aligned} \boldsymbol{z}^* \ \boldsymbol{w} &= [x_1 - iy_1 \quad x_2 - iy_2 \quad ... \ ] \ [u_1 + iv_1 \quad u_2 + iv_2 \quad ... \ ]^T \\ &= x_1u_1 + y_1v_1 + x_2u_2 + y_2v_2 + ... \qquad \Longleftarrow \text{cos component} \\ &\quad + i\,(x_1v_1 - y_1u_1 + x_2v_2 - y_2u_2 + ...) \quad \Longleftarrow \text{sin component} \end{aligned} \tag{3.30}$$

3.4.4 Comparison between Complex-Valued Inner Product and Euclidean Distance

Figure 3.12 is a conceptual illustration to show the merit of this complex inner-product metric. In active imaging, for example (see, e.g., Chapter 6), we obtain coherent signals consisting of amplitude and phase. The feature vector is defined in complex domain. For a set of high-coherence signals, i.e., signals having similar phases, the summation to generate inner product grows straightforward as shown by arrows (a) in Fig. 3.12. Contrarily, in a low-coherence case, having random phases, the summation does not grow so much as shown by arrows (b). This effect emerges also in the Euclidean metric to some extent. However, the Euclidean metric is related only to the cosine component as shown in Fig. 3.12(a') and (b'), resulting in a partial treatment of phase directions. The evaluation results can be different from (a) and (b). The complex inner-product metric is then more sensitive to signal coherence and, therefore, enhances the distinction among various objects compared with the case of Euclidean metric described below.

In addition, the complex inner product is inherently less sensitive to the norm of signal vectors. This is simply because of the normalization. It is desirable in particular in coherent imaging systems where we often suffer from distortion in intensity caused by the mirror glaring and speckles.

3.4.5 Metric in Correlation Learning

Correlation learning, to be used widely in neural networks such as associative memories described in Section 4.3.5 and the simplest case of filtering based on Markov random field model in Section 4.6, also possess the same feature of the complex-valued learning. The correlation learning embeds the correlation between output signals $\boldsymbol{z}_s$ and input signals $\boldsymbol{z}_t$ in synaptic weights $\boldsymbol{w}$. For simplicity of expression, we consider one of the output signals z_s out of $\boldsymbol{z}_s$. As shown in detail in Sections 4.3.5 and 4.6, the learning dynamics is expressed as

$$\tau \frac{d\boldsymbol{w}}{dt} = -\boldsymbol{w} + z_s \ \boldsymbol{z}_t^* \tag{3.31}$$

where τ is learning time constant in time t domain. Various pairs of input $\boldsymbol{z}_t$ and output z_s teacher signals are presented to the network for the training. The correlation is accumulated into $\boldsymbol{w}$, converging at

$$\boldsymbol{w} \quad \longrightarrow \quad K < z_s \ \boldsymbol{z}_t^* > \tag{3.32}$$

where K is a real constant.

Here we express the teacher signal pairs in real and imaginary parts as

$$z_s = x_s + jy_s \tag{3.33}$$

$$\boldsymbol{z}_t = [x_{t1} + jy_{t1}, x_{t2} + jy_{t2}, \cdots, x_{tN} + jy_{tN}]^T \tag{3.34}$$

where j and N are imaginary unit and the input terminal number. Then the product in the correlation in (3.32) is rewritten as

$$\begin{aligned} z_s \ \boldsymbol{z}_t^* = [\ &(x_s x_{t1} + y_s y_{t1}) + j(y_s x_{t1} - x_s y_{t1}), (x_s x_{t2} + y_s y_{t2}) + j(y_s x_{t2} - x_s y_{t2}), \\ &\cdots, (x_s x_{tN} + y_s y_{tN}) + j(y_s x_{tN} - x_s y_{tN})]^T \end{aligned} \tag{3.35}$$

The real and imaginary parts mix with each other. The meaning becomes obvious when we express the pixel values in amplitude and phase as

$$z_s = r_s e^{i\theta_s} \tag{3.36}$$

$$\boldsymbol{z}_t = [r_{t1} e^{jy_{t1}}, r_{t2} e^{jy_{t2}}, \cdots, r_{tN} e^{jy_{tN}}]^T \tag{3.37}$$

and rewrite (3.35) as

$$z_s \ \boldsymbol{z}_t^* = [\ r_s r_{t1} e^{j(\theta_s - \theta_{t1})}, r_s r_{t2} e^{j(\theta_s - \theta_{t2})}, \ \cdots, r_s r_{tN} e^{j(\theta_s - \theta_{tN})}]^T \tag{3.38}$$

The product yields the phase difference as well as the amplitude product, which is compatible with the signal circularity.

On the contrary, if we regard the neural network as a real-valued network having double input terminals and two output neurons corresponding to real and imaginary parts, the dynamics for double-dimensional real signals z_s and $\boldsymbol{z}_t$ are expressed as

$$z_s = [x_s, y_s] \tag{3.39}$$

$$\boldsymbol{z}_t = [x_{t1}, y_{t1}, x_{t2}, y_{t2}, \cdots, x_{tN}, y_{tN}]^T \tag{3.40}$$

and the product as a step to calculate correlation becomes

$$z_s\ \boldsymbol{z}_t = [\ x_s x_{t1}\ ,\ \ y_s y_{t1}\ \ ,\ \ x_s x_{t2}\ ,\ \ y_s y_{t2},\ \cdots\ ,\ x_s x_{tN}\ ,\ \ y_s y_{tN}]^T \tag{3.41}$$

We can find that the product (3.41) is different from (3.35) or (3.38). That is, the dynamics of the real-valued network is completely different from that of the complex-valued one. The difference originates from the very basic arithmetic operation, and is therefore very fundamental. This property may also be called circularity as one of the characteristics of the complex-valued neural network. The circularity is one of the most essential features of the complex-valued neural networks.

3.5 What Is the Sense of Complex-Valued Information and Its Processing?

Utilizing complex numbers is one of the bottom lines in modern science and technology.[3] However, "imaginary" was a great misnomer. "Imaginary number" implies unreality and powerlessness. We have to sometimes inquire about to CVNNs as follows. What is the difference between a CVNN dealing with n-dimensional information and a real-valued network processing $2n$-dimensional one? Do we observe something imaginary in brain measurement using microelectrode?

However, the application fields of CVNNs have been extending more and more [21],[56]. In CVNNs, the flexibility in learning and self-organization is restricted rather than that in double-dimensional real-valued neural networks. As we discussed in Section 3.2.1, the restriction is brought by the four fundamental rules of arithmetic in complex numbers. Such fundamental rules in processing often work well in solving real world problems. This feature is one of the most useful advantages in CVNNs.

Typical applications include the treatment of wave-related phenomena such as sonic wave, lightwave, and electromagnetic wave. Let us consider a broadcasting system, namely, a radio or television system.

[3] The beginning of this section is a retouch of the Article [22].

A sender transmits an electromagnetic wave having a carrier frequency of f, and a receiver catches the wave. The wave needs to be modulated so that it carries information. There are various modulations, but they are fundamentally categorized into amplitude modulation, phase modulation, and frequency modulation, which change amplitude, phase and frequency of the carrier wave, respectively. That is, information is carried on amplitude a, phase θ, or frequency f that is time-differential of phase. These facts are common to both analog and digital communications.

The receiver decides what information is sent by the sender by examining the amplitude a and phase θ (and sometimes frequency f as well). A square-law detector detects squared amplitude a^2, which is equivalent to energy, while an envelope detector detects amplitude a. However, the phase cannot be detected in such a simple manner. Instead, we use homodyne (or heterodyne) detection as follows. In the receiver, we prepare a local oscillator of the same frequency f, and multiply the received wave by the output wave of the oscillator. The multiplying process is called mixing. Then, thanks to the angle sum and difference identities of trigonometric functions, the mixing yields a subtracted-frequency signal ($f - f = 0$) and an added-frequency one ($f + f = 2f$). We extract the former, i.e., the 0-frequency signal, with a filter. It is called baseband signal containing cos and sin components corresponding to the amplitude and phase of the transmitted signal. (The detail of the technique is explained in the next section.) In this process, we need both the cos and sin components to determine the phase value. In other words, we observe $a\cos\theta + ia\sin\theta = ae^{i\theta}$.

There are various types of devices that realizes modulations at the sender such as amplitude modulators and phase modulators. An amplitude modulator modulates the wave power, while a phase modulator put the wave forward or backward in time. Each modulator interacts with elemental entity in physics. Therefore, in adaptive signal detection and noise elimination, neural processing that deals with amplitude and/or phase is expected to be capable of learning or self-organizing efficiently by excluding unnatural or needless adaptive behavior. The above consideration leads to a neural network whose synaptic weights are composed of amplitude and phase modulations. That is, the network deals with phasor (complex amplitude) by amplifying / attenuating the amplitude and rotating the phase.

At the early proposal stage of CVNNs, the effectiveness was sometimes suspected, i.e., "why do we need deal with complex numbers?" However, in these years, we have various fields where CVNNs are effectively used, in particular in wave-related processing. They are even necessary in some areas such as independent component analysis in frequency domain. Application fields are explained in detail in the next section.

Considering the above situation, together with the dismissal of differentiability, we may say that CVNNs pay more attention to complex number as complex amplitude, i.e., phasor, than that as two-dimensional plane number.

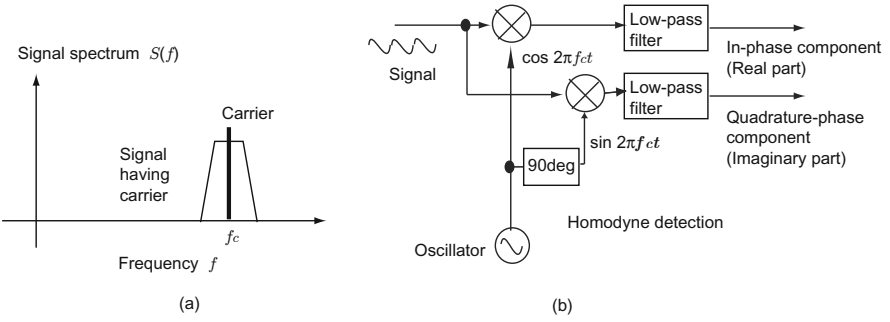


Fig. 3.13 (a)Spectrum of non-baseband signal (modulated wave) and (b)extraction of in-phase (real) and quadrature-phase (imaginary) components of the signal by homodyne detection.

3.6 In What Fields Are CVNNs Effective?

Complex-valued neural networks (CVNNs) are useful in particular in the following fields. Referring to the diagram showing applications in Fig.1.2 (Page 7) in Chapter 1, we discuss features in dynamics in the respective fields.

3.6.1 Electromagnetic and Optical Waves, Electrical Signals in Analog and Digital Circuits

Non-baseband signals (modulated carrier waves)

Figure 3.13(a) shows the spectrum of a modulated carrier waves (non-baseband signals) such as electromagnetic-wave signals in radio and television broadcasting and mobile communications, as well as lightwave signals in optical communications. The spectrum ranges around its carrier frequency f_c. The signal is carried on the carrier wave modulated by a modulator. A technique called homodyne detection extracts the real and imaginary parts of the signal as follows [57].

Figure 3.13(b) shows the schematic diagram of the homodyne detection. We prepare a local oscillator that generates two waves orthogonal to each other (cos and sin) of the same frequency f_c, and multiply (mix) divided incoming signals by them, respectively. Low-pass filters extract only the baseband signals (subtracted-frequency component), i.e., in-phase (real) component obtained through the cos path and quadrature-phase (imaginary) component obtained through the sin path. The phase of the local oscillator works as the phase reference that determines real and imaginary axes in information. Note that this scheme functions well for various signals if the relative bandwidth (= (bandwidth) / (carrier frequency)) is small enough, even for a carrier suppressed signals such as binary-shift-keying signals.

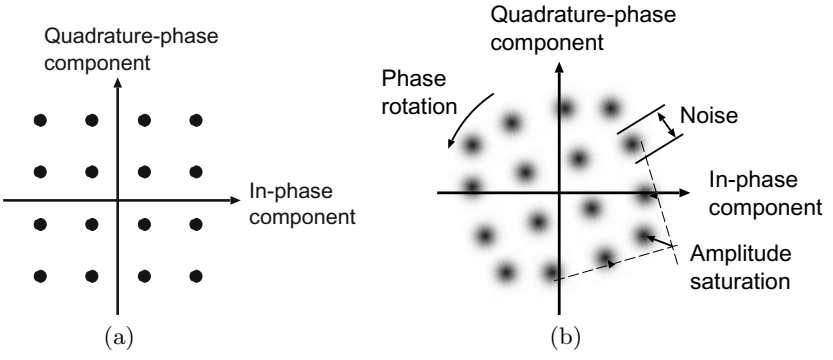


Fig. 3.14 Signal constellation in complex plane in 16-QAM digital communications. (a)Ideal constellation and (b)received one affected by noise, phase rotation, and saturation.

The mixing is effective also for lightwave detection. When a photodiode detects a lightwave, it yields a photocurrent proportional to optical power, i.e., the squared amplitude. Therefore, we can obtain a baseband signal simply by superimposing a local-oscillator lightwave on the incoming signal lightwave at a halfmirror, and detecting the mixture with a photodiode [58]. The mechanism is the same as that of Young's interference experiment where two interfering lightwaves have an identical carrier frequency so that the frequency difference is 0.

Note that, in most cases, we can practically neglect harmonic waves that we may observe in nonlinear systems dealing with waves. As an example, let us consider the so-called 16-quadrature-amplitude modulation (16-QAM) in wireless or lightwave communications. Figure 3.14(a) shows an ideal signal constellation in the complex plane. When a receiver detects the signal, the constellation is affected by random noise, phase rotation (Doppler effect), and possible harmonic waves originating from, for example, the nonlinearity in the output stage in a transmitter. However, if the receiver is equipped with a filters to eliminate the harmonics, the constellation does not include the harmonics, but instead, a simple amplitude saturation just like shown in Fig.3.14(b) [59]. Such a simple saturation is useful to realize an activation function in neural networks. Similar nonlinearity is observed in optical-fiber amplifiers and various saturation phenomena.

Baseband signals

Incidentally, we often need to deal with baseband signals also, i.e., time-sequential signals of $f_c = 0$, with CVNNs. Even in such baseband cases, neural adaptive processing in complex domain is effective, as mentioned in Chapter 1. We can adaptively deal with most of baseband signals consistently

in total by assuming that they have imaginary components obtained in the following manner.

In general, a real-world time-sequential signal has causality. The real and imaginary parts of the Fourier-transformed signal are, hence, not independent of each other. That is, we can construct an imaginary signal consistent with the real one as the Hilbert transform pair. The generated signal is called *analytic signal*. Note that the automatic generation of the imaginary information does not mean the insignificance of the complex treatment. Instead, by taking the imaginary part into account, we can realize a neural learning and self-organization more appropriately without unnecessary search in information space than we might do when we consider, as it were, only the real-part projection of what the signal should be [60]. That is, even though all information is carried by the real part, we can make a neural network learn or self-organize in a better way by a preprocessing that transforms a real-valued input signal into a consistent complex-valued one.

Complex-valued signals are generated as follows. Let us consider a signal whose frequency spectrum is shown in Fig.3.15(a). A real-valued signal in time domain has an even spectrum with a symmetry line at $f = 0$. (More generally speaking, $S(f) = (S(-f))^*$ where * means complex conjugate.), that is, if the spectrum component at f is the same as that at$-f$, the real part of the signal is doubled, while the imaginary part is cancelled out, resulting in a real-valued signal. Therefore, if we extract only the positive f component of the spectrum and inversely Fourier transforms it, we will obtain a complex-valued baseband signal (complex amplitude, or phasor) that has a imaginary part naturally consistent with the real part.

A procedure to generate the complex-valued baseband signal is explained as follows. First, we conduct the Hilbert transform on a real-valued baseband signal $s(t)$. The result is $s_{\mathrm{H}}(t) \equiv Hs(t)$. The transform is expressed in continuous or discrete time (unit time delay is τ_0) as

$$s_{\mathrm{H}}(t) = \frac{1}{\pi}\int_{-\infty}^{\infty}\frac{s(t-\tau)}{\tau}d\tau \qquad \text{(continuous time)} \tag{3.42}$$

$$= \frac{1}{\pi}\sum_{k\neq 0}\frac{s(t-k\tau_0)}{k\tau_0}\tau_0 \qquad \text{(discrete time)} \tag{3.43}$$

A little arithmetic reveals that the characteristic function of the Hilbert transform $H(f)$ rotates the spectrum in positive frequency by $-\pi/2$ while that in negative frequency by $+\pi/2$, i.e.,

$$H(f) = \begin{cases} j \ (= e^{j\pi/2}) & (f < 0) \\ 0 & (f = 0) \\ -j \ (= e^{-j\pi/2}) & (f > 0) \end{cases} \tag{3.44}$$

where $j \equiv \sqrt{-1}$ here. As a result, we obtain the spectrum shown in Fig.3.15(b). Then, $(1 + jH)$ transform generates the spectrum in Fig.3.15(c).

The multiplication of j means a rotation by $\pi/2$, which is easily realized in digital processing by alternating the real and imaginary parts with signs taken into account. Then the inverse Fourier transform generates a complex-valued baseband signal.

Note that, in a practical system, the above process including the Hilbert transform is realized in time domain with a digital filter as shown in Fig.3.15(d). Since baseband signals are mostly composed of low frequency components, they are easily dealt with by analog / digital converters (A/D converters) and digital signal processors. In such a case, we construct a filter of certain steps, instead of ideal one of infinite steps. We also need a delay of half number of total steps for the real-part signal flow because the Hilbert transform requires the delay.

3.6.2 Electron Wave

On the left-hand side in Fig.1.2 (Page 7) in Chapter 1, we find various applications based on physical entities such as electromagnetic wave and lightwave. Next to them, we can also find quantum waves such as electron wave. When we observe microscopic particles such as electrons, we find significant quantum nature. It is a big issue in science and technology to utilize the quantum nature, in particular in the modern nanotechnology.

Quantum nature often means the wave nature of particles, i.e., quantum wave. That is, in the case of electron, we deal with the electron wave. In this sense, electron wave is somewhat analogous to electromagnetic wave. For example, single-electron devices, where an electron is apparently regarded as a particle, are also considered based on the wave nature, i.e., the standing electron wave. Note that, at the same time, we have to pay attention to differences between electromagnetic wave (photon) and electron wave in relation to the electric charge and the rest mass.

Conventional direction of development of electron devices has been very simple and clear, i.e., higher speed, lower operation power and voltage, and larger-scale integration of transistors such as field-effect transistors (FETs). The criteria have always been valid because we have been sure that we can solve more problems if we can deal with larger capacity of bit information quickly. However, as we see in Section 2.1 (comparison between AI and neural networks) and Fig.2.1, it turned out that we have various problems difficult to formalize, to reduce into elements, and therefore, to solve by conventional approaches. The maturity of the system-on-chip technology to integrate complicated system on a single chip also accelerates new approaches to consider device physics and system functions altogether.

Consequently, it is expected that the researches on highly functional systems including neural networks develop into device level investigations more and more. It becomes hence significantly important to construct a neural framework where we deal with wave aspects of electrons as quantum waves from adaptive-system viewpoints. We have the possibility to develop quantum

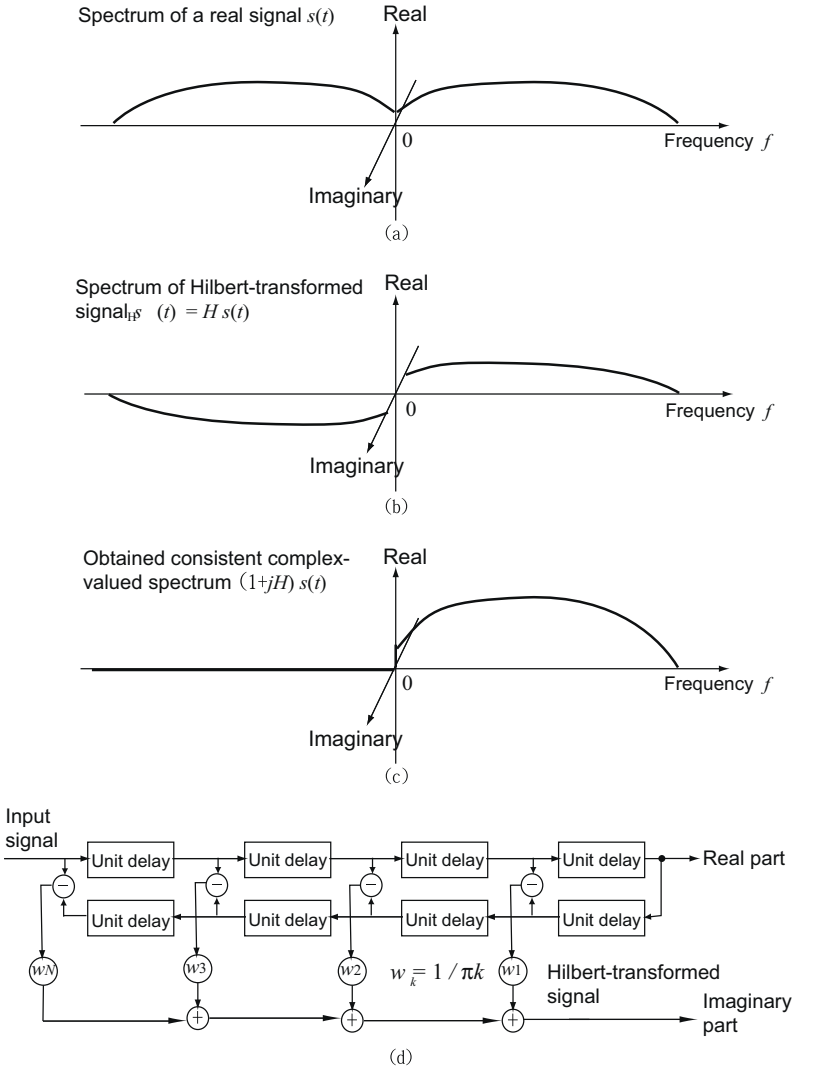


Fig. 3.15 Spectrums of (a)real-valued baseband signal $s(t)$, (b)Hilbert-transformed real-valued signal $Hs(t)$, and (c)complex-valued signal to be obtained $(1+jH)s(t)$, and (d)digital circuit or program that generates the complex signal.

devices that possess adaptability at the microscopic level of device physics. The theory of the CVNNs can be a framework exactly suitable for such investigations. Quantum-wave neural theories and quantum neural devices is increasingly developing such interdisciplinary fields.

3.6.3 Superconductors

Superconductivity is a quantum effect having some parallels with electron wave. Based on the same reason that electron wave has, adaptive superconductive devices will be developed in the future. In comparison with electron wave, we may utilize its macroscopic nature and ultralow power consumption. This field is also a hopeful realm. Interactions of Josephson junction with electromagnetic wave will also be a key phenomenon.

3.6.4 Quantum Computation

Quantum computation is another field to which the theories in the CVNNs are effectively applicable since the quantum states are represented by complex amplitude. It is a very natural and prospective approach to pay attention to the complex states and operators as well as the parallel and distributed operations to find analogies with the theory of CVNNs. References in Section 3.7 present recent development.

3.6.5 Sonic and Ultrasonic Waves

In Fig.1.2(Page 7) in Chapter 1, we also find sonic and ultrasonic waves. In these applications, we can expect the same adaptability as that in electromagnetic-wave and lightwave applications. When we process sonic and ultrasonic waves electronically, we generate or detect them with transducers such as speakers and microphones. Then we treat electrical signals, that reflect the sonic and ultrasonic properties governing propagation, reflection, diffraction, and other sound-wave phenomena, in the same manner as those in dealing with electric signals in Section 3.6.1.

Medicine is one of the most important fields related to ultrasonic waves. Ultrasonic diagnosis will become more useful if we can generate ultrasonic beams more adaptively (beamforming), or can process obtained image with learning, on the basis of CVNNs. To improve reflection images, we will be able to realize an automatic adaptive feedback for a vibrator array by being based on obtained image information such as independence in pixelwise information. We may call such a system a superbrain that can *see* or *hear* ultrasonic complex amplitude.

We have another direction, i.e., realization of CVNNs using sonic and ultrasonic waves. In such an approach, we can expect a larger variety in constructing neural systems since we have physically various possibilities in wavelength and velocity than those in electromagnetic waves. It is also attractive to provide ultrasonic wave itself with adaptability. For example, there exists a method to inform solely a person standing at the edge of a platform at a railway station by focusing to the person an ultrasonic beam modulated by vocal wave. This caution system utilizes nonlinearity at our outer

ears that extracts the vocal information out of the modulated ultrasonic carrier wave. Adaptive and soft CVNN processing will realize more variously shaping beamforming required for more complex arrangement of persons and objects, which will lead to information delivery with a more complicated individualization.

3.6.6 Compatibility of Controllability and Adaptability

We find application fields related to CVNN dynamics on the right-hand side in Fig.1.2 (Page 7) in Chapter 1. At the center of the figure, we have a box showing compatibility of controllability and adaptability. The compatibility is realized by modulating carrier frequency in a CVNN to change its behavior, i.e., processing, learning, and self-organization. System examples are shown in Chapter 8 (Lightwave associative memory that has carrier-frequency dependent behavior), Chapter 9 (Lightwave phase equalizer), and Chapter 10 (Developmental learning in robotics).

Neural networks possess high adaptability. Therefore, it is sometimes difficult for users to control the network behavior. However, in coherent neural networks, we can use the carrier frequency as a key to change the behavioral mode after the network conducted learning in a carrier-frequency dependent manner. A carrier-frequency dependent self-organization is also possible if the network has an appropriate feedback mechanism including frequency-determining network in the self-organization, in which the network finds the most suitable key value self-organizingly by itself. Such a self-organization is related to volition, which is indispensable for emergence of context-dependent behavior and so-called brainlike information-processing systems in the future.

Moreover, if we conduct processing at multiple behavioral modes in parallel by assigning multiple carrier frequencies, we can realize a carrier-frequency-domain parallelism in neural processing. Because one of the structural features of neural networks in general is parallelism, the above frequency-domain parallelism is attractive in relation not only to neurodynamics but also to hardware implementation utilizing vast optical frequency bandwidth.

3.6.7 Periodic Topology and Metric

In CVNNs, phase information has a periodic topology. We can sometimes utilize this nature. Conventional neural networks employing ordinary sigmoid function or radial basis function (RBF) are not good at dealing with information intrinsically possessing a periodic structure. Previously, researchers proposed nonmonotonic activation-function neurons employing, e.g., trigonometric activation functions (cos or sin) as the activation functions, to deal with periodic information. In this context, CVNNs are the networks that address this issue head-on. We have many technical areas that include periodically structured information.

Even in physiology, periodic information structure is related to various observations. On human visual cortex, we have a neuron that fires for a vertical-line stimulus. We also have a neuron that fires for slightly slant (e.g., in 10 degrees) line stimulus. Another neuron fires for a little more slant stimulus. The neurons are aligned on a curve in the order of the degree of slant as shown in Fig.3.16. That is, the order of what the stimulus means to human being determines the mapping of the corresponding neurons, though they are actually on a meandering curve. The brain develops the slant information as the placement of corresponding neurons, i.e., as the topology and metric of the neuron mapping. In the following neural processing realized with many local connections, the contiguity plays an important role. The human beings have a higher resolution in particular at around vertical and horizontal lines because these angles have important meanings for us to live in the real world, and also because we are familiar with such angles in our lives. On the other

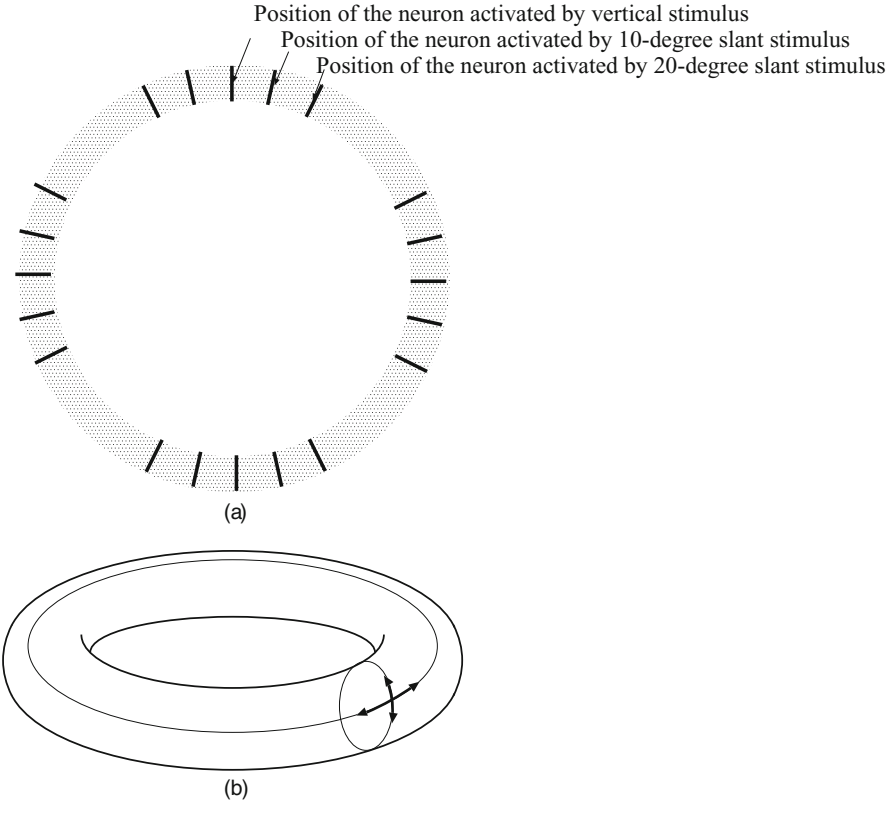


Fig. 3.16 (a)Schematic illustration of mapping of slant-bar information on the visual cortex and (b)a torus schematically expressing multidimensional circular mapping.

hand, we have lower resolution at some other angles and, at around such angles, neurons are assigned with a lower density.

When we intend to realize a mapping with a high resolution homogeneously in all the angles in engineering systems, the fragmentary arcs will be connected with one another to form a circular structure. In a task where such circular structures will be constructed in n-dimensional space, it must be helpful to prepare some arrangement, within the basic neural structure, favorable for constructing n-dimensional torus. For this purpose, we can utilize the periodicity in phase. (In other words, the one-dimensional torus group is isomorphic to multiplicative group of all complex numbers with absolute value of unity, and the n-times direct product of the one-dimensional torus with itself is geometrically an n-torus.) Such an approach can be named the preparation of "a superbrain cortex having n-dimensional torus structure."

3.6.8 Direct Use of Polar Coordinates

We can utilize the coordinate dependence of the neural dynamics, in particular, the dependence on the polar coordinate instead of Cartesian one. We can also enjoy the ease in numerical treatment. For example, the fovea centralis on the retina has a high resolution to gaze at a still object, whereas the fringes have lower one to catch something moving. The functions of the retina and following networks depend on the position of the visual field, i.e., at the center or on the edge. The polar coordinate system is suitable for realizing an adaptability having a smooth changeover from the central function to the edge one basically continuously.

The polar coordinate is advantageous also to us to deal with images obtained by omnidirectional cameras without deforming it into a rectangular frame. The computational time in deformation deteriorates the real-time processing ability. In particular, when we use a retina-like imaging device having a higher resolution at the center in the future, the deformation will be a serious detour. A direct treatment using the polar coordinate is ideal and effective. Incidentally, if we feed the omnidirectional image directly to neural networks, the network will self-organize in such a way that it constructs a self-centered view of the world, which may be similar to the worldview of the human beings and useful to realize a humanlike robot. That is, we can bring up a "brain that self-organizes with a polar-coordinate visual field."

3.6.9 High Stability in Recurrent Dynamics

Recurrent neural networks, that have feedback loops in the structure, realize time-sequential operation. Recurrent CVNNs have a high stability in general in dealing with time-sequential signals such as movies, i.e., gradually changing images.

When we employ a real-valued recurrent neural network, we often experience breakdowns into chaotic behavior. However, if we try with a CVNN,

we can obtain more stable behavior [61]. The advantage stems from the fact that a CVNN can deal with signals as a weighted summation of various phasors, i.e., by synthesizing revolutions of complex vectors of various amplitude, phase, and frequency. This feature results in the stability in dynamics in, e.g., the analysis and synthesis of time-sequential patterns, such as memory and recall of music melodies [62].

3.6.10 Preservation of Relative Directions and Segmentation Boundaries in Two-Dimensional Information Transform

If we use a linear or holomorphic activation function, we can utilize the conformal nature in total neural transform. When we equate the complex plane with two-dimensional information, such as an image to be transformed through a CVNN, the relative direction of two vectors in the input image is preserved in the output one. Therefore, in clustering or classification tasks, the direction relationship of crossing two boundary curves is also preserved [63]. This nature is approximately available in networks with weak nonlinear activation functions including real-imaginary-type and amplitude-phase-type ones.

3.6.11 Chaos and Fractals in Complex Domain

Chaos in complex domain possesses characteristic nature [64]. Chaotic phenomena have been analyzed in these years, and may be connected to useful applications in the future.

Fractals in complex domain are also interesting. Complex transforms remind us of the Mandelbrot set obtained by iterative complex transform. In the construction of digital elevation map (DEM) (See Chapter 7), we observe earth surface with interferometric radars, and conduct the phase unwrapping process. Though the unwrapping process sometimes yields distortion, we can reduce the distortion by use of fractal parameters [65]. Such practical usages of fractals will be developed increasingly.

3.6.12 Quaternion and Other Higher-Order Complex Numbers

The complex number is directly related to two-dimensional plane, or rather rotations in two-dimensional space. Similarly, the quaternion is parallel to three-dimensional space and rotations thereby. Rotation in three- or higher-dimensional space is noncommutative. That is, when the order of sequential rotation processes having various rotational-axis directions is changed, the process in total is also changed. Quaternion has this nature intrinsically. Application fields include three-dimensional control in robotics, adaptive

computer graphics such as virtual realities, and generation of images in adaptive three-dimensional color space. It is also useful in control of polarization of lightwave and electromagnetic wave. Polarization is represented on the Poincaré sphere. Control of polarization on the Poincaré sphere means three-dimensional rotation. The combination of phase rotation by ordinary CVNNs and the polarization manipulation by quaternion neural networks leads to a variety of adaptive polarization control.

Further higher-order complex numbers are also known for their characteristic natures. Octernion does not satisfy the commutativity law nor the associativity law. If these laws hold, we have a larger degree of operations (order of operation, etc.). However, higher-order complex numbers gradually looses the freedom and increases restriction. This nature is completely different from that of increase in the dimension of real-number space. To utilize higher-order complex numbers, it is desirable to understand the nature of the objective situations so that we can adopt appropriate features of the complex numbers to make good use of the restrictions.

3.7 Investigations in Complex-Valued Neural Networks

3.7.1 History

Researches on CVNNs date back to the middle of the 20th century. Introduction of phase-based information representation was introduced by Eiichi Goto of Japan in 1954 in his invention of "parametron" [25] [26] . Valve computers and memories at that time had a short lifetime and unstable operation. He aimed at more stable physical representation of values by employing multi-stable phase locking in oscillation. Numbers were expressed binary, decimal, or in the N-based manner by discrete phase values of integral multiple of $2\pi/N$ as $\exp(\sqrt{-1}\ 2n\pi/N)$ where n ($0 \leq n < N$) is an integer. Along with elemental researches on parametron counter circuits [66] , theoretical analysis on frequency division in feedback circuits [67] , phase memory devices [68] , design of decimal arithmetic circuits [69] and others, he and Hidetosi Takahasi succeeded in constructing generic computers based on inherently multi-valued logic circuits utilizing multi-stable oscillation and convergence at discrete multiple phase values [70] [71] , though in many cases they adopted binary representation. They realized general-purpose digital computers that possess higher reliability, quicker operation, and lower cost than vacuum-tube computers at that time. The series of the researches had big propagation effects in many fields. Afterwards, however, transistor computers have come to dominate all over the world. In general-use digital computers, the information representation itself has only small influence on the dynamics of information processing because the processing algorithm is determined in discrete *logic*, instead of any *dynamics*. This is a point of great difference from that of neural networks. In the CVNN case, the use of phase (or complex-amplitude) in

information representation plays a significantly important and essential role as shown in Sections 3.1 to 3.6.

In 1971, Aizenberg et al. in Soviet Union (present Russia) reported an idea to extend binary output values of "0" and "1" to multiple values on the unit circle in the complex plane [27]. They also proposed an implementation on the basis of pulse position modulation (PPM) where they coded the multiple phase values into the time shift of the "1" pulse in a unit period. This time-shift idea is compatible with the phase shift in the parametron and modern wave applications, suggesting the universality of close relationship between time and phase.

In the above two cases, the multiple-valued output signals are placed on discrete points on the unit circle. Such a class of CVNNs is sometimes called the phasor neural networks where the amplitude is fixed at unity, though the word phasor does not always mean a fixed-amplitude vector in complex plane. Aizenberg's group published a book on the multiple-valued networks in 2000 [72]. (One of the authors of the book, Naum Aizenberg, is the father of another, Igor Aizenberg. Though Naum already passed away, Igor is developing the work.)

In the adaptive signal processing in radar systems, communications, and other applications, on the other hand, we inevitably need to deal with complex signals. In such linear processing, Widrow et al. in the U.S.A. presented the complex least mean square (LMS, the steepest descent method with squared error) [46]. Since it is linear, the dynamics is clear. The most basic treatment is that *we substitute the hermitian conjugate (conjugate transpose) for the transpose of vectors and matrices in real-valued processing.* This operation is widely found in science and technology such as quantum mechanics.

In 1988, Noest in the Nederland proposed multiple-valued (phasor) associative memory [73] and obtained the memory capacity [74]. It was found that, under a certain condition, the normalized capacity of the phasor associative memory is calculated as $\pi/4$, which is a little larger than $2/\pi$ of a binary associative memory.

Afterward, in late 1991 and 1992, many ideas were presented on the steepest descent method and backpropagation learning, both of which are important learning process explained in Chapter 4, intensively and independently. In 1991, Leung & Haykin considered a CVNN whose activation function is expressed as $y_j = 1/(1 + \exp(-\sum w_{ji}x_i))$ with complex variables, as presented in (3.17). They analyzed the dynamics with partial derivatives in real and imaginary parts [48].

In 1992, Benvenuto & Piazza considered a CVNN whose activation function is expressed separately in real and imaginary parts as $f(z) = \text{sigmoid}(\mathbf{Re}(z)) + i\ \text{sigmoid}(\mathbf{Im}(z))$, namely, the real-imaginary-type activation function [49]. This can be one of the simple extensions of a real sigmoid activation function. Its advantage is that we can also obtain a backpropagation algorithm as a simple extension of that in the real-valued networks.

Paper by Nitta [75], an extension of his Japanese one [76], also considered a CVNN with the real-imaginary-type activation function and analyzed the dynamics by dealing with the real and imaginary parts separately. He assumed a task to transform two-dimensional images composed of points in complex plane to discuss the two-dimensional generalization ability.

There exist other papers introducing a real-imaginary-type activation function to consider steepest descent methods and backpropagation algorithms with independent real and imaginary parts. For example, the purpose of the papers by Kim & Guest [77] and Birx & Pipenberg [78] lay in optical implementation. However, they analyzed the dynamics with the real and imaginary parts only separately, though the separate treatment makes most analyses simple.

In such a situation, the paper by Georgiou & Kutsougeras in 1992 [79] discussed what class of activation functions is meaningful, after some real-imaginary-type theoretical analysis is given in the beginning. Assuming an analog circuit implementation, they proposed a function whose amplitude is saturating, while the phase is unchanged, i.e., $f(z) = \frac{z}{c+|z|/r}$ where c and r are parameters determining the saturation characteristic. This function is basically equivalent to the amplitude-phase-type activation function shown in (3.21). Consideration on signals or waves in electronic circuits reasonably leads us to an activation function of this type.

In the same year, Hirose proposed an associative memory employing an amplitude-phase-type activation function, independent of Goergiou's group, and demonstrfated smooth recall in the complex domain [80]. This proposal also assumed physical implementations such as quantum devices in the future.

In 1992 again, Hirose proposed a steepest descent method and a backpropagation algorithm available for CVNNs employing the amplitude-phase-type activation function [81], and demonstrated their effectiveness. They were derived on the basis of partial derivatives in amplitude and phase directions, which are compatible with this activation function. The backpropagation has another characteristic feature. That is, it does not make error signals backpropagate but, instead, makes teacher signals themselves backpropagate through the neuron layers. This feature matches implementations with lightwave and electromagnetc wave (See Chapter 4).

The year 1992 was fruitful year in CVNN researches. Takeda & Kishigami reported the theory and experiments on a complex-valued lightwave associative memory [82]. They noticed the fact that the mathematical expression of the lightwave field in a phase-conjugate-mirror resonator is identical with that of a complex-valued associative memory. Their success in the experiments was epoch-making indeed. To construct a theory realistically effective, it is important to develop it in conformity with physical reality. They also succeeded in presenting an excellent example that a basic physical phenomenon realizes a CVNN.

Following above ideas related to the amplitude-phase-type CVNNs, Hirose's group published several papers on coherent lightwave neural networks,

including a paper describing physical implementation with homodyne detection and the details of the amplitude-phase-type backpropagation [50], and another one presenting numerical analysis on the generalization characteristics in the frequency domain [83] [84] [85].

Moreover, they proposed the realization of carrier-frequency-dependent behavior (processing, learning, self-organization, etc.) in coherent electromagnetic-wave or lightwave CVNNs [83] [86]. This idea leads not only to the control of neural behavior by modulating the carrier frequency, but also to emergence of volition in combination with a feedback of output signals to controllers (See Chapter 10). Furthermore, it realizes frequency-domain multiplexing (i.e., frequency-domain parallelism) in the vast optical frequency band (Chapters 8 and 9). These advantages originate from the precise frequency-dependent manipulation of signals by dealing with the phase information in the CVNNs.

3.7.2 Recent Progress

Theories

Theoretical aspects have been variously discussed from diverse viewpoints with a constellation of ideas. Hanna & Mandic analyzed the dynamics of steepest descent learning in a single-layered filter having an activation function of $1/(1 + \exp(-\sum w_i x_i)$, or of the amplitude-phase type [87]. They also investigated the resulting outputs of the filtering process [88]. Hanna & Mandic [89] and Mandic & Chambers [90] discussed the effect of data reusing in learning in CVNNs. Besides, many papers reported analysis results on learning algorithms such as Casasent & Natarajan [91], Nitta [92] [93], Goh & Mandic [94] [95] [96], Fiori [97], and Xu et al. [98]. We can also find an error evaluation in perceptrons by Yang et al. [99] and reduction of hidden-layer neuron number by Kobayashi [100]. Learning property in multiple-valued networks has also been investigated by Cao et al. [101] and Gao et al. [102].

Characteristics of activation functions were discussed in, for example, Kim & Adali [103], Kim & Adali [104], Jankowski et al. [105]. Regarding complex-valued associative memories, there are various reports on energy functions and dynamics: Kuroe et al. [106], Hirose [107], Nemoto & Kubono [108], Lee & Wang [109], Aoki & Kosugi [110], Lee [111] [112] [113], Prashanth [114], Takahashi [115], Müezzinoğlu et al. [116] , as well as detailed discussion on stability of point attractors: Agu et al. [117] and Hirose [118]. Dynamics of CVNNs with real-imaginary-type activation functions were analyzed when they are used for complex-plane transform in Nitta [119] [63] [120]. Principal component analysis (PCA) and independent component analysis (ICA) have also been investigated from various points of view by Zhang & Ma [121], Sawada et al. [122], Makino et al. [123], Rattan & Hsieh [124], Uncini & Piazza [125], Anemuller et al. [126], Li & Adali [127] [128] and Novey & Adali [129]. Oscillatory networks are also closely related to complex-valued networks, which are discussed by Burwick [130] [131] .

Quaternion and Clifford neural networks were proposed and analyzed by Pearson & Bisset [132], Arena et al. [133], Isokawa et al. [134], Kusamichi et al. [135], Buchholz [136] [137], Byro-Corrochano & Arana-Daniel [138] and many other research groups.

Applications

There are many applications in microwave and millimeter-wave fields such as CVNN-based adaptive designing of patch antennas:

Mishra & Patnaik [139], Du et al. [140], Mishra & Patnaik [141], estimation of DoA (direction of arrival) using CVNNs proposed by Yang et al. [142], adaptive beamforming in array antennas by Suksmono & Hirose [143] [144], Yamaki & Hirose [145] [146], Chang et al. [147] , and adaptive processing of airborne and satellite interferometric radar images for globe observation [148], or ground penetrating radars (GPRs) for landmine detection Hara & Hirose [149] [150], Masuyama & Hirose [151], Masuyama et al. [152], Nakano & Hirose [153] [154] [143]. Some of the following chapters are dedicated to such radar image processing. Such ideas are useful also in ultrasonic imaging reported by Nishino & Hirose [155].

In neurophysiological analysis, the dynamics of complex-valued Nagumo-Sato model to represent time-sequential neuron activities and emerging chaotic behavior were analyzed and discussed by Nemoto & Saito [64]. In bioinformatics and related image processing, the gene expression was analyzed for classification of the stages of gene expression by using CVNNs by Aizenberg et al. [156]. Handayani et al. [157] proposed an adaptive image segmentation method based on so-called "snake," which is a dynamic boundary, in the complex domain for segmentation of vessels in magnetic resonance images (MRI).

In associative memories, various beautiful manners were proposed to deal with data transformed into frequency domain such as Aoki et al. [158], Aizenberg & Butakoff [159], Aizenberg et al. [160]and Tanaka & Aihara [161].

In communications, a specific dynamics was designed by Miyajima et al. [162] where a phasor neural network has an attractor at the coordinate origin as well as other attractors on the unit circle for applications to multiple-access communications. We can regard CVNNs as a superset of complex-valued filters [163]. Adaptive equalizers and filters have been proposed in many papers such as You & Hong [164], Gan et al. [165], Hirose & Nagashima [59], Wang et al. [166], Jianping et al [167], Park & Jeong [168], Kawamoto & Inouye [169], Koike & and Noda [170], Deng & Yang [171], and Chen et al. [172].

We have large number of CVNN papers related to time-sequential processing such as Chakravarthy & Ghosh [173] where they discussed embedment of pattern sequences in oscillations. CVNNs have a more stable dynamic behavior in comparison with those in real-valued ones when they construct recurrent neural networks such as time-sequential associative memories [174], [175],

[61]. The dynamic stability is widely applicable in other areas. For example, melody (music) memorization and extraction was reported by Kinouchi & Hagiwara [62]. Other methods to learn for time-sequential prediction were also proposed in, e.g., Rajagopal & Kak [176] showing a quick learning algorithm.

One of the recent novel applications is the traffic-dependent optimal control of traffic signals connected by roads mutually and complicatedly, proposed and analyzed by Nishikawa & Kuroe [177]. They considered the learning dynamics of large number of oscillators (traffic signals) connected by roads to make cars flow better. CVNNs were also applied to calculate inverse matrices by Song & Yam [178], which can be a feedback to mathematics. Regarding higher-order complex numbers, we have a quaternion neural-network application, for example, in learning in three-dimensional RGB space to transform color images proposed by Isokawa et al. [134].

A variety of CVNN hardware has also been proposed using lightwave [51], electromagnetic wave, in particular high-frequency electric signal wave, electron wave, and sonic or ultrasonic wave [155]. Since a high-frequency processing requires high-speed operation, analog neural networks are desirable rather than digital or pulsed neural networks. Though analog networks can deal with signals speedily, the low precision has usually been a big problem. However, recent progress in circuit components is overcoming the drawback [179].

Quantum neural networks form another new field with various proposals such as Kinjo et al. [180], Sato et al. [181], Kouda et al. [182], Kinjo et al. [183] and Nakamiya et al. [184].

Books

Besides the first edition of this present book

- "Complex-Valued Neural Networks" by Hirose (2006, Springer, book review by Georgiou [185]) (Japanese Edition, 2005, Saiensu-sha) [186],

there are some books on CVNNs or CVNN-related topics:

- "Multi-Valued and Universal Binary Neurons – Theory, Learning and Applications –" by I.Aizenberg, N.Aizenberg & J.Vandewalle (2000, Kluwer Academic Publishers) [72],
- "Complex Valued Nonlinear Adaptive Filters – Noncircularity, Widely Linear and Neural Models" by Mandic & Goh (2009, Wiley) [43],
- "Complex-Valued Neural Networks with Multi-Valued Neurons" by Igor Aizenberg (Springer, 2011, in the Series of Studies in Computational Intelligence) [187],

and multi-author books:

- "Complex-Valued Neural Networks: Theories and Applications" edited by Hirose (2003, World Scientific, book review by Aizenberg [188]) [56], and
- "Complex-Valued Neural Networks: Utilizing High–Dimensional Prarameters" edited by Nitta (2009, Information Science Publishing) [189].
- "Complex-Valued Neural Networks: Advances and Applications" edited by Hirose (2012, The IEEE Press / Wiley) [190].

4

Constructions and Dynamics of Neural Networks

In this chapter, we present constructions of neural networks and their dynamics in processing, learning, and self-organization. Biological neural networks, in particular, most of interface networks, evolved in such a specific manner that each network fits its particular purpose such as seeing, listening, or speaking. Artificial neural networks also possess various constructions dependent on purposes. Therefore, in this chapter, we investigate the constructions and dynamics in individual networks according to purposes. However, there exits a common dynamics in their microscopic mechanisms of learning and self-organization, namely, the Hebbian rule in the broad sense of the word. First, we consider the Hebbian rule. Then we go on to various constructions and dynamics in networks. You do not need any detailed background in advance since we begin with conventional (real-valued) neural networks and, afterward, we extend them into complex-valued neural networks (CVNNs).

4.1 Processing, Learning, and Self-organization

4.1.1 Pulse-Density Signal Representation

In biological neural networks, electric pulses, called action potentials, pass through the network and carry information among neurons. Transmission of an action potential is called *firing*. Once the neurons at sensory organs such as eyes and ears fire, the action potentials are transmitted to sensory neurons, processed in various ways, and sent with other internally generated pulses to motor neurons that activate motor organs such as hands, legs, and vocal cords so that we can move and speak.

We construct artificial neural networks by modeling the biological ones. The most widely used modeling technique is to deal with pulse density in time instead of pulse itself. That is, we consider that the principal variable in the network is the pulse frequency, i.e., how many pulses pass the network

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 57–100.
springerlink.com

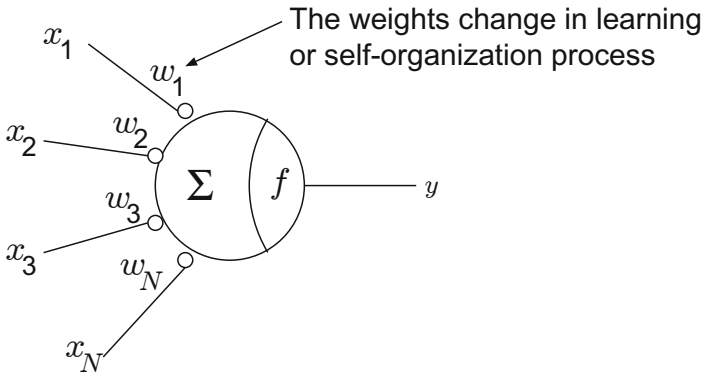


Fig. 4.1 Learning and self-organization are attributed to the changes of the weights placed at the input of the neuron.

in unit time. We consider the pulse density as the signal in neural networks. This type of neuron model is called a McCulloch-Pitts neuron.

Figure 4.1 shows input and output signals. By normalizing the pulse density in time appropriately, we can consider that input and output signals, x_i and y, respectively, are regarded as real number in a range of [0,1]. We can also modify the range to [−1,1] for symmetry. They are basically identical, which we already examined in relation to the activation functions expressed by (3.16) and (3.17) in Chapter 3. We assume that the input signal values are also in the range of [−1,1]. We introduce a real-valued activation function $f(u) = \tanh(u)$ shown in (3.16). The weights are also real numbers in real-valued networks.

Here we remember the features in neural networks listed in Section 2.3.

1. Distributedness and parallelism: Many simple elements similar or identical to one another, namely, neurons, gather and make connections to let pulse-density information pass in parallel.
2. Locality: Information that a neuron can sense is limited to input signals fed from other neurons through connections, internal state of the neuron itself, and, in some cases, the state of the neuron to which the output is connected. Connections are also often local.
3. Weighted sum and activation function with nonlinearity: Input signals are weighted at the connections. The internal state of a neuron is simply the weighted sum of the input signals or a nonlinear transform of the weighted sum. The nonlinear function is called activation function.
   Note that, in this Book, we do not deal explicitly with the so-called threshold in the model neuron. Contrarily, model neurons in not a few books possess a threshold so that (internal state) = (weighted sum of inputs) − (threshold). When we consider an analogy between neurons and logic gates such as AND and OR, the threshold gives us useful insights,

since AND is realized with a high threshold in the neuron, while OR is with a low one. But it has no relation to the basis of neuron dynamics, which you will find in the following sections. If you need a threshold for some reason, you can put an additional input, x_0, per neuron and feed it with -1 constantly. Then the weight w_0 works just like the threshold.

4. Plasticity: Weights change according to the information fed to the neuron and the internal state. They can change even in the processing of tasks. The change is called learning or self-organization.
5. Generalization: A neural network behaves expectedly not only to situations it learned, but also to unlearned ones by inferring an optimal action on the basis of previously learned events. It constructs its own view of the world, i.e., its own metric (measure), to estimate the optimal reaction to unforeseen environment.

4.1.2 Neural Dynamics

"Behavior" of a neural network has the following two aspects.

1. Processing tasks. A neural network reacts to signals, presented by the environment, by processing the presented input information in a manner meaningful to the network or network users.
2. Learning or Self-organization. The network changes itself to process information meaningfully. The change in processing is mostly realized as the change in weight values, but sometimes also as the change in connection routes, i.e., network topology.

We call the details of the behavior the neurodynamics. Section 4.2 and the following sections present several examples of neurodynamics.

4.1.3 Task Processing

"A neural network processes a task" means that the network yields output signals appropriately to input signals. In a real-valued neural network, both the input and output signals are real numbers.

The signals can be temporally changing. Then the network processes time-sequential signals. In digital logic circuits, we have combinational circuits, such as AND and OR, and sequential circuits having memory circuit to deal with time sequence. Neural networks can also work as both of them. However, we seldom prepare explicit memories. Instead, we utilize the latency, or delay, in the reaction of neurons. Though conventional computers are equipped with processors and memories separately, neural networks possess them in a harmonious whole.

4.1.4 Learning and Self-organization

Learning is the changes of the neural network, made by the network itself, in such a way that the resultant processing behavior becomes in accordance with the wishes of network users, e.g., "memorize this relationship" or "behave in this way." One can wish the behavior in full detail (supervised learning) or vaguely by presenting only the final goal to the network (unsupervised learning). One may give an evaluation to the network for every learning trial (reinforcement learning).

Self-organization is the change of the neural network, made by the network itself again, to become self-consistent through interactions with environment. Users find something useful in the result or the resultant behavior.

4.1.5 Changes in Connection Weights

Though both the learning and self-organization are explained as the changes mentioned above from a macroscopic viewpoint, they are also gradual changes of neural-connection weights microscopically. Figure 4.1 shows the changeable weights w_i for inputs x_i. The capability of the connection change is called plasticity. The change does not necessarily require explicit target values. Instead, the weights can approach appropriate values by taking the input and output values into consideration. When desirable input and output values are given explicitly from the environment, they are called teacher signals.

There are various procedures in the weight changes. For example, the network may be provided with all teacher signals required, and change the connections starting from certain initial values. This learning can be regarded as a batch process. If the environment is changed, the network repeats to learn.

In other cases, however, the network may learn or self-organize during task processing. It is, so to speak, an on-the-job training. The learning and processing occur in unison. In general, the task processing works quickly, while the learning progresses slowly. Besides, learning may occur at intervals between task processing.

4.2 Hebbian Rule

The Hebbian rule is considered to be the most basic principle of learning. It is a hypothesis proposed by D. O. Hebb, a psychophysiologist in Canada / U.S.A., in 1949, related to biological neural networks [29], namely, "When an axon of cell A is near enough to excite cell B and repeatedly or consistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."

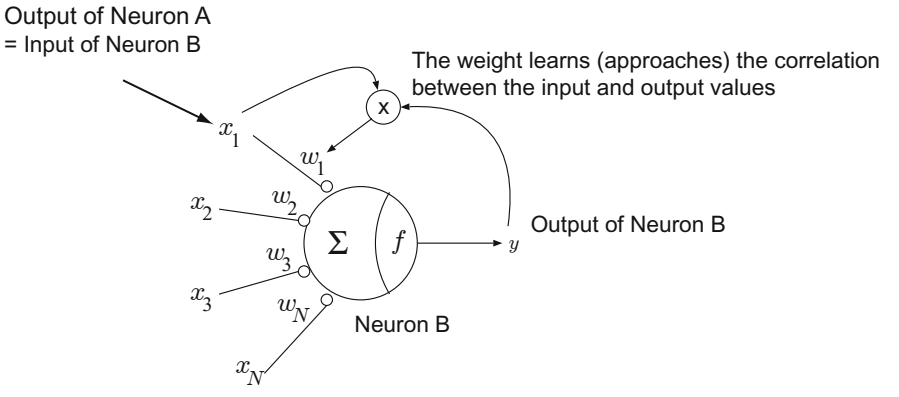


Fig. 4.2 Changes in connection weights according to the Hebbian rule.

Recent progress in physiological experiments revealed the correctness of the assumption.

With the neuron model described in the previous section, the rule is expressed as the temporal change of the connection weight w_i as

$$\tau \frac{dw_i}{dt} = -w_i + yx_i \tag{4.1}$$

where τ is the time constant of the change, and y and x_i denote output and input signals, respectively, as shown in Fig.4.2.

When the input signal of Neuron B (output signal of Neuron A) x_i is almost unity, and the output signal of Neuron B y is also near to unity, the weight w_i increases. In a [−1,1]-neuron's network, the weight w_i also increases even if both the input and output are near to -1, which is consistent with the meaning of the hypothesis. The steady state solution is derived by assuming $dw_i/dt = 0$ as

$$w_i = < yx_i > \tag{4.2}$$

where $< \cdot >$ denotes expectation. That is, when various input signals $x_i(t)$ are fed to the neuron one after another, the mean product $< yx_i >$, i.e., the correlation, is stored in the weight w_i.

We can implement the Hebbian rule as software by discretizing (4.1) as

$$w_i(t+1) = (1-K)w_i(t) + K\ y\ x_i \tag{4.3}$$

where t is discrete time, and K $(0 < K < 1)$ is the constant determining the changing speed. In a steady state where the fraction $\Delta w_i(t) \equiv w_i(t+1) - w_i(t)$ that changes the weight w_i is zero, the weight is $< yx_i >$ again.

Then, in what way the complex-valued Hebbian rule, the complex version of the Hebbian rule, can be expressed? When we consider the correlation

between two complex variables, we multiply one variable with the complex conjugate of the other one. Therefore, we obtain the follows consistent with the real valued one.

$$\tau \frac{dw_i}{dt} = -w_i + y\ (x_i)^* \qquad \text{(continuous time)} \tag{4.4}$$

$$w_i(t+1) = (1-K)w_i(t) + K\ y\ (x_i)^* \qquad \text{(discrete time)} \tag{4.5}$$

where $(\cdot)^*$ means complex conjugate. The weight w_i is, so to speak, an operator to operate on the input x_i to yield the output y. Therefore, we adopt the correlation yx_i^* where the latter variable x_i (input) is conjugated. The operation is clarified in a matrix expression. We express the input signals as a vertical vector, while the hermitian conjugate (conjugate transpose) as a horizontal vector.

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \qquad \boldsymbol{x}^* = [(x_1)^*\ (x_2)^*\ \cdots\ (x_N)^*] \tag{4.6}$$

The weight $\boldsymbol{w}$ is expressed as a horizontal vector to generate an output y (presently a scalar) as the inner product of the weight $\boldsymbol{w}$ itself and the input signals $\boldsymbol{x}$.

$$\boldsymbol{w} = [w_1\ w_2\ \cdots\ w_N] \tag{4.7}$$

The Hebbian rule, (4.4) and (4.5), is then expressed as a horizontal-vector equations as

$$\tau \frac{d}{dt}\boldsymbol{w} = -\boldsymbol{w} + y\ \boldsymbol{x}^* \qquad \text{(continuous time)} \tag{4.8}$$

$$\boldsymbol{w}(t+1) = (1-K)\boldsymbol{w} + K\ y\ \boldsymbol{x}^* \qquad \text{(discrete time)} \tag{4.9}$$

Incidentally, the conjugating operation in lightwave and electromagnetic-wave physics is related to, for example, the phase-conjugation mirrors. It realizes the reverse of time, or the reverse of propagation direction. We will utilize the relationship between the conjugation and the reverse of time when we develop a so-called backpropagation learning process in Section 4.4 (Function approximation).

Moreover, the complex-valued Hebbian rule corresponds directly to the recording process in holography. That is, the memory of correlation (4.8) is optically realized simply by using a hologram. Frequency-domain multiplexing in complex-valued Hebbian-rule learning was also reported [191], [192].

To summarize the Hebbian rule, it is the most fundamental weight-changing rule. The real-valued Hebbian rule is expressed by (4.1) and (4.3), while the complex-valued Hebbian rule is expressed by (4.4) and (4.5), or (4.8) and (4.9). In any case, the Hebbian rule makes the connection weight

approach to the input-output correlation. We will find in the following section that the Hebbian rule is applicable in correlation learning when teacher signals are given to the network in a bunch or time-sequentially. In Section 4.4.6, it is also useful in function approximation in supervised learning.

In the following sections, we present variously functional neural networks using the Hebbian rule. The behavior and functional operations will be explained in a plain manner, for example, by showing how a set of input signals is processed in the network, so that the readers can grasp some hints to apply the dynamics in various application fields. If one needs more rigorous mathematics, please refer to literature listed in Chapter 2.3.

4.3 Associative Memory

4.3.1 Function: Memory and Recall of Pattern Information

Associative memories typify the spirit of neural networks. They are sometimes called Hopfield networks. The function is the memory and recall of pattern information. That is, an associative memory memorizes a set of patterns. Then, it chooses one of the memorized patterns nearest to an input pattern fed, and yields it as the output pattern. Though the input is generally noisy and ambiguous, the associative memory estimates the pattern most similar to the input.

4.3.2 Network Construction and Processing Dynamics

Figure 4.3 shows the network construction. All the neurons are connected with one another. That is, it is a fully-connected neural network. Once an input signal pattern is fed to the network, the output pattern is fed again to the neural inputs iteratively. This structure is called *recurrent*. The hippocampus in the human brain, that plays an important role in memorizing information, possess a clear recurrent structure. The associative memory was modeled after the hippocampus.

The input signal is an N-dimensional vector $\boldsymbol{x}$. The output is also an N-dimensional vector, and the values of its elements are within $[-1, 1]$. We assume that the element values of the vectors to be recalled are ± 1. We adopt $f(u) = \tanh(u)$ as the activation function saturating at ± 1. In such an associative memory, we usually obtain positive or negative unity as the final output values, though we may have halfway values in the recall process. The patterns to be memorized $\boldsymbol{s}_\mu$ are also N-dimensional vectors whose elements are ± 1.

We express input, output, and memorized vectors, $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{s}_\mu$, respectively, as vertical vectors as

$$\boldsymbol{x} \equiv \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}, \quad \boldsymbol{y} \equiv \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \quad \boldsymbol{s}_\mu \equiv \begin{bmatrix} s_{1\mu} \\ s_{2\mu} \\ \vdots \\ s_{N\mu} \end{bmatrix} \tag{4.10}$$

For example, when we deal with an image whose pixel number is N, the simplest way to map the pixel values to the input vector is to make the pixels align to form an N-dimensional vector.

The output $\boldsymbol{y}$ is also a vector, and then the connection-weight vector explained in the Hebbian-rule section is extended to a matrix. The connection-weight matrix $\mathbf{W} \equiv [w_{ji}]$ is determined, or formed through learning, as follows.

When all the N-dimensional vectors to be memorized $\boldsymbol{s}_\mu$ can be previously known, we can determine the weight matrix at once. That is, the weight matrix should be the autocorrelation matrix of $\boldsymbol{s}_\mu$ where the subscript $\mu = 1, 2, \cdots, \overline{\mu}$ is given to each vector to be memorized.

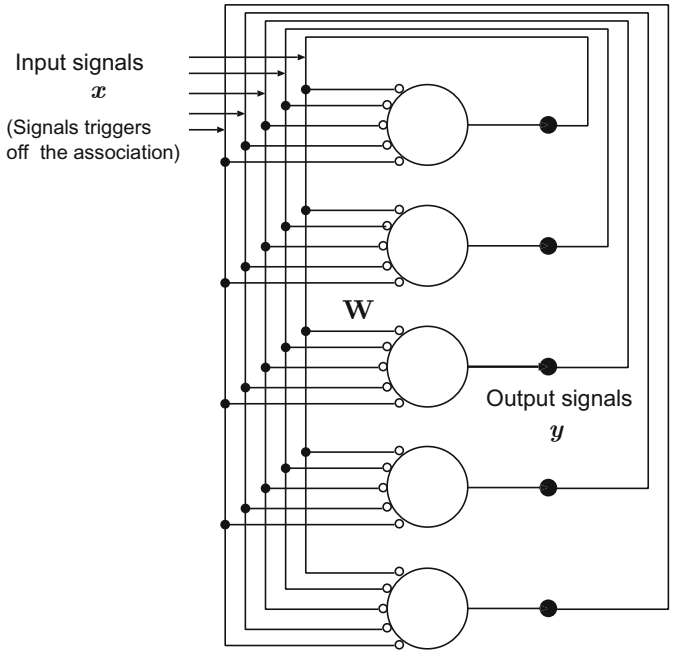


Fig. 4.3 Construction of associative memory network.

$$\mathbf{W} \equiv \left[\begin{array}{c|c|c|c} & & & \\ \boldsymbol{s}_1 & \boldsymbol{s}_2 & \cdots & \boldsymbol{s}_{\overline{\mu}} \\ & & & \end{array}\right] \left[\begin{array}{c} (\boldsymbol{s}_1)^{\mathrm{T}} \\ \hline (\boldsymbol{s}_2)^{\mathrm{T}} \\ \hline \vdots \\ \hline (\boldsymbol{s}_{\overline{\mu}})^{\mathrm{T}} \end{array}\right]$$
$$= \mathbf{S}\mathbf{S}^{\mathrm{T}} \tag{4.11}$$

where $\mathbf{S}$ denotes horizontally aligned vertical vectors $\boldsymbol{s}_\mu$, and $(\cdot)^{\mathrm{T}}$ means a transposed vector or transposed matrix.

$$\mathbf{S} \equiv \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1\overline{\mu}} \\ s_{21} & s_{22} & \cdots & s_{2\overline{\mu}} \\ \vdots & \vdots & & \vdots \\ s_{N1} & s_{N2} & \cdots & s_{N\overline{\mu}} \end{bmatrix}, \quad \mathbf{S}^{\mathrm{T}} \equiv \begin{bmatrix} s_{11} & s_{21} & \cdots & s_{N1} \\ s_{12} & s_{22} & \cdots & s_{N2} \\ \vdots & \vdots & & \vdots \\ s_{1\overline{\mu}} & s_{2\overline{\mu}} & \cdots & s_{N\overline{\mu}} \end{bmatrix} \tag{4.12}$$

The weight matrix $\mathbf{W}$ becomes symmetric.

The task processing, namely, recall, is conducted as follows. First, an N-dimensional input vector $\boldsymbol{x}$ is fed to the network as a trigger to start the recall at (discrete) time $t = 0$. The input propagates through the weights and the neuron, and result in an output $\boldsymbol{y}(t = 1)$ with a unit-time delay. The output is fed again to the neural inputs recurrently and transformed to a new output $\boldsymbol{y}(t = 2)$. After a single or several times of iteration, the output vector $\boldsymbol{y}(t)$ converges at a certain vector, which is often the vector nearest to the input $\boldsymbol{x}$ among the memorized ones $\boldsymbol{s}_\mu$. The "nearest vector" usually means the vector that yields a largest inner product $(\boldsymbol{s}_\mu)^{\mathrm{T}}\boldsymbol{x}$ with the input vector. That is,

$$\begin{aligned} \boldsymbol{y}(1) &= f(\mathbf{W}\boldsymbol{x}) \\ \boldsymbol{y}(2) &= f(\mathbf{W}\boldsymbol{y}(1)) \\ \boldsymbol{y}(3) &= f(\mathbf{W}\boldsymbol{y}(2)) \\ &\vdots \\ \boldsymbol{y}(t) &\longrightarrow \boldsymbol{s}_\mu \text{ nearest to the input vector } \boldsymbol{x} \end{aligned} \tag{4.13}$$

where $f(\boldsymbol{u}) = \tanh(\boldsymbol{u})$ is the activation function, for output values of $[-1, 1]$, working on every element of the vector $\boldsymbol{u}$. The output $\boldsymbol{y}(t)$ converges typically with a single or several iterations. In other words, the recall is completed.

Why it is capable of recalling the vector nearest to the input? To understand the dynamics, we trace the changes in the signal vector. Assume that the input vector $\boldsymbol{x}$ is near to, say, $\boldsymbol{s}_2$. Then we can express $\boldsymbol{x}$ using $\boldsymbol{s}_2$ and noise $\boldsymbol{n}$ as

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{s}_2 + \boldsymbol{n} \\ &= \begin{bmatrix} s_{12} \\ s_{22} \\ \cdots \\ s_{N2} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \cdots \\ n_N \end{bmatrix} = \begin{bmatrix} s_{12} + n_1 \\ s_{22} + n_2 \\ \cdots \\ s_{N2} + n_N \end{bmatrix}\end{aligned} \tag{4.14}$$

Then the product of the weight matrix and the input is calculated as

$$\begin{aligned}\mathbf{W}\boldsymbol{x} &= \mathbf{W}(\boldsymbol{s}_2 + \boldsymbol{n}) \\ &= \mathbf{S}\mathbf{S}^{\mathrm{T}}(\boldsymbol{s}_2 + \boldsymbol{n}) \\ &= \left[\begin{array}{c|c|c|c} & & & \\ \boldsymbol{s}_1 & \boldsymbol{s}_2 & \cdots & \boldsymbol{s}_{\overline{\mu}} \\ & & & \end{array}\right] \left[\begin{array}{c} (\boldsymbol{s}_1)^{\mathrm{T}} \\ \hline (\boldsymbol{s}_2)^{\mathrm{T}} \\ \hline \vdots \\ \hline (\boldsymbol{s}_{\overline{\mu}})^{\mathrm{T}} \end{array}\right] (\boldsymbol{s}_2 + \boldsymbol{n})\end{aligned} \tag{4.15}$$

We pay attention at the end of (4.15). If the memorized vectors are chosen at random, they are approximately orthogonal to each other. That is, their inner product should almost be $(\boldsymbol{s}_\mu)^{\mathrm{T}}\boldsymbol{s}_\nu \simeq 0$. The noise, on the other hand, is independent of the memorized vectors, resulting also in $(\boldsymbol{s}_\mu)^{\mathrm{T}}\boldsymbol{n} \simeq 0$. Therefore, (4.15) is calculated as

$$\begin{aligned}&\left[\begin{array}{c} (\boldsymbol{s}_1)^{\mathrm{T}} \\ \hline (\boldsymbol{s}_2)^{\mathrm{T}} \\ \hline \vdots \\ \hline (\boldsymbol{s}_{\overline{\mu}})^{\mathrm{T}} \end{array}\right] (\boldsymbol{s}_2 + \boldsymbol{n}) \\ &= \begin{bmatrix} (\boldsymbol{s}_1)^{\mathrm{T}}\boldsymbol{s}_2 + (\boldsymbol{s}_1)^{\mathrm{T}}\boldsymbol{n} \\ (\boldsymbol{s}_2)^{\mathrm{T}}\boldsymbol{s}_2 + (\boldsymbol{s}_2)^{\mathrm{T}}\boldsymbol{n} \\ \vdots \\ (\boldsymbol{s}_{\overline{\mu}})^{\mathrm{T}}\boldsymbol{s}_2 + (\boldsymbol{s}_{\overline{\mu}})^{\mathrm{T}}\boldsymbol{n} \end{bmatrix}\end{aligned} \tag{4.16}$$

$$= \begin{bmatrix} 0+0 \\ (\boldsymbol{s}_2)^{\mathrm{T}}\boldsymbol{s}_2 + 0 \\ \vdots \\ 0+0 \end{bmatrix} = \begin{bmatrix} 0 \\ |\boldsymbol{s}_2|^2 \\ \vdots \\ 0 \end{bmatrix} \tag{4.17}$$

In consequence, we obtain the product of the weight and the input signal as

$$\begin{aligned}\mathbf{W}\boldsymbol{x} &= \left[\begin{array}{c|c|c|c} & & & \\ \boldsymbol{s}_1 & \boldsymbol{s}_2 & \cdots & \boldsymbol{s}_{\overline{\mu}} \\ & & & \end{array}\right] \begin{bmatrix} 0 \\ |\boldsymbol{s}_2|^2 \\ \vdots \\ 0 \end{bmatrix} \\ &= |\boldsymbol{s}_2|^2 \boldsymbol{s}_2 \end{aligned} \tag{4.18}$$

That is, the output $\boldsymbol{y}(1)$ becomes

$$\boldsymbol{y}(1) = f(|\boldsymbol{s}_2|^2 \boldsymbol{s}_2) \tag{4.19}$$

The activation function transforms $|\boldsymbol{s}_2|^2 \boldsymbol{s}_2$ with a saturation characteristic that binds the output in the range of $[-1, 1]$. If $|\boldsymbol{s}_2|^2$ is sufficiently larger than unity, we obtain

$$\boldsymbol{y}(1) = \boldsymbol{s}_2 \tag{4.20}$$

In this manner, the associative memory recalls $\boldsymbol{s}_2$. In practice, the product of memorized vectors $(\boldsymbol{s}_\mu)^{\mathrm{T}} \boldsymbol{s}_\nu$ and that of memorized and noise vectors $(\boldsymbol{s}_\mu)^{\mathrm{T}} \boldsymbol{n}$ may not completely be zero. However, the residual difference should statistically be very near to zero. Therefore, we can expect that the residue is removed with a few or several iterations of the above process, and that the network recalls $\boldsymbol{s}_2$.

Note that, when the number of memorized vectors $\overline{\mu}$ becomes large in comparison with the vector length (neuron number) N, the influence of the noise is enhanced and significantly influential to the recalling dynamics as mentioned later. It is known in statistics that the capacity of the associative memory (the maximum number of memorizable vectors $\overline{\mu}$ normalized by the neuron number N) is approximately $\overline{\mu}/N = 0.15$.

4.3.3 Energy Function

We can interpret the recall operation as a process to minimize a certain energy function. We define the energy E as

$$E(\boldsymbol{y}; \mathbf{W}) \equiv -\frac{1}{2} \boldsymbol{y}^{\mathrm{T}} \mathbf{W} \boldsymbol{y} \tag{4.21}$$

It is shown that the energy (4.21) decreases along with the progress of the recall $\boldsymbol{y} \longrightarrow \boldsymbol{s}_\mu$, and becomes a relative minimum at the end. The energy is also called Lyapunov function. It is a quadratic form of $\boldsymbol{y}$, and $\mathbf{W}$ is the metric matrix of the energy.

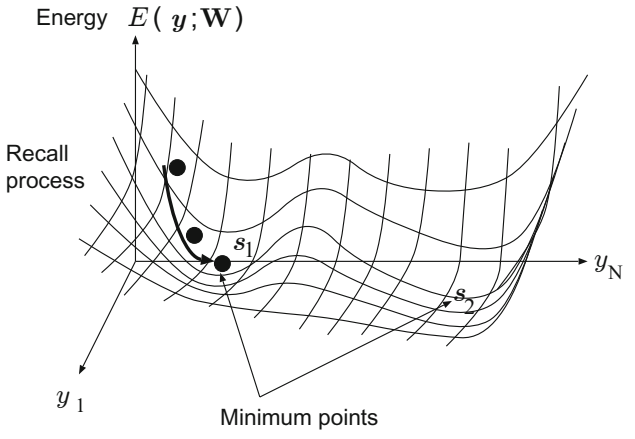


Fig. 4.4 Schematic illustration of the energy function $E(\boldsymbol{y};\mathbf{W})$ determined by the weight matrix of the associative memory $\mathbf{W}$ against signal vector $\boldsymbol{y}$.

We can understand the decrease of the energy intuitively as follows. The energy function (4.21) is decomposed as $-\frac{1}{2}\sum_\mu \left|(\boldsymbol{s}_\mu)^{\mathrm{T}}\boldsymbol{y}\right|^2$, just like we did in (4.15). It is the summation of the squared products of the memorized vectors and the signal where, as we considered in the calculation in (4.17), all the products are approximately zero except for that with the memorized vector nearest to the signal ($\boldsymbol{s}_2$ in the previous case). Therefore, as the recall proceeds and the signal $\boldsymbol{y}$ approaches to $\boldsymbol{s}_2$, the product of $\boldsymbol{s}_2$ and $\boldsymbol{y}$ increases, while other ones stay around zero. Therefore, the summation grows in total, and the negative of the summation is the energy.

We also find that, since the weight matrix $\mathbf{W}$ is symmetric, the energy variation per iteration must be negative as shown below, and that the movement of $\boldsymbol{y}$ stops at a relative minimum. For example, assume that an element y_k of the signal vector $\boldsymbol{y}$ changes as $y_k(t+1) = y_k(t) + \Delta y_k$. Then, by using the relationship $\sum_i w_{ki}y_i = \sum_j y_j w_{jk} = u_k$, we can express the energy variation as

$$\begin{aligned}\Delta E &= -\frac{1}{2}(\boldsymbol{y}+\Delta\boldsymbol{y})^{\mathrm{T}}\mathbf{W}(\boldsymbol{y}+\Delta\boldsymbol{y}) - \left(-\frac{1}{2}\boldsymbol{y}^{\mathrm{T}}\mathbf{W}\boldsymbol{y}\right)\\ &= -\frac{1}{2}\left(\sum_i \Delta y_k\ w_{ki}y_i + \sum_j y_j w_{jk}\ \Delta y_k + w_{kk}(\Delta y_k)^2\right)\\ &= -\Delta y_k\ u_k(t) - \frac{1}{2}w_{kk}\ (\Delta y_k)^2 \end{aligned} \tag{4.22}$$

Therefore, if the activation function has a steep rise against input signal value, $u_k(t) > 0$ leads to $y_k(t+1) \sim 1$ which implies $\Delta y_k > 0$. Then the first term of the right-hand side of (4.22) should be negative. Contrarily, if

$u_k(t) < 0$, then $y_k(t+1) \sim -1$ and $\Delta y_k < 0$, resulting in the term being negative again. On the other hand, the second term, $w_{kk}\Delta y_k^2$, is negligible if the diagonal components are not large.

Figure 4.4 shows a schematic illustration of the energy landscape. The energy has relative minima at memorized points. This landscape is the "world" that is constructed in the associative-memory network and represented by the metric matrix, namely the weight matrix. If the memorized vectors are not orthogonal but close to one another, the energy minima are also placed near to one another and gives rise to incorrect recall. As the number of memorized vectors increases, the landscape does not stay simple any more, but becomes complex with many folds and local minima in which the signal vector is trapped and incapable of reaching the correct goal. That is, the recall fails.

The matrix construction (4.11) shows that the diagonal components of $\mathbf{W}$ consists of nonnegative numbers such as $w_{11} = s_{11}^2 + s_{12}^2 + \cdots + s_{1\overline{\mu}}^2$. They are often large in particular when the memory has many memorized vectors. Then we may not be able to neglect the second term of the right-hand side of (4.22). The term yields fluctuation in the energy and local minima. Therefore, to avoid such inconvenience, we sometimes force the diagonal components be zero, i.e., $w_{ii} = 0$, after the matrix construction by (4.11). The process removes direct self-feedback connections in the network. Even in this case, the dynamics of the recall process is principally the same as that mentioned above.

4.3.4 Use of Generalized Inverse Matrix

Do we have any good measures to overcome the harm of nonorthogonality of the memorized vectors, i.e., non-zero inner products $(\boldsymbol{s}_\mu)^{\mathrm{T}}\boldsymbol{s}_\nu$? We can improve the weight matrix in such a way that the recall dynamics works well. That is, we use a generalized inverse matrix to change the metric of energy (metric matrix) by substituting a set of new bases of the metric for the old one. The introduction of the generalized inverse matrix changes the understanding of the world in the network so that it feels as if the memorized vectors were orthogonal to one another. We determine $\mathbf{W}$ as

$$\mathbf{W} \equiv \mathbf{S}(\mathbf{S}^{\mathrm{T}}\mathbf{S})^{-1}\mathbf{S}^{\mathrm{T}} \tag{4.23}$$

where $(\mathbf{S}^{\mathrm{T}}\mathbf{S})^{-1}\mathbf{S}^{\mathrm{T}}$ is a generalized inverse matrix called Moor-Penrose pseudo inverse.

4.3.5 Weight Learning by Sequential Correlation Learning

In the previous section, we constructed the weight matrix in a batch process. However, we can also make the network learn the vectors to be memorized by presenting them one after another to the network. This learning is called correlation learning since the network learns the correlation between presented

input and output vectors. It is one of the supervised learning methods. With this learning, the network learns the environment gradually by adapting even to a slowly changing or shading world. In this sense, the correlation learning is more flexible than the batch method.

We present one of the vectors $\boldsymbol{s}_\mu$ $(\mu = 1, 2, \cdots, \overline{\mu})$ to the input and output of the network simultaneously. Then we present the next one, and we continue the presentation for all the vectors to be memorized repetitively. To memorize the correlation, the weight $\mathbf{W}$ gradually changes as

$$\tau \frac{dw_{ji}}{dt} = -w_{ji} + s_{j\mu} s_{i\mu} \tag{4.24}$$

where τ is time constant of learning. The corresponding matrix expression is

$$\tau \frac{d\mathbf{W}}{dt} = -\mathbf{W} + \boldsymbol{s}_\mu (\boldsymbol{s}_\mu)^{\mathrm{T}} \tag{4.25}$$

The formal expression is the same as (4.1). In this sense, the correlation learning is an application of the Hebbian rule. As mentioned in Section 4.2, the Hebbian rule is fundamentally a rule to memorize correlation. Therefore, the learning of the autocorrelation explained here has the same formal expression. The discrete expression is also the same as that of the discrete Hebbian rule (4.3), i.e.,

$$\mathbf{W}(t+1) = (1-K)\mathbf{W}(t) + K\ \boldsymbol{s}_\mu\ (\boldsymbol{s}_\mu)^{\mathrm{T}} \tag{4.26}$$

where time t can be differently scaled from the processing time in combination with the time constant K. Usually we choose the time constant τ (or K) in such a way that the learning process proceeds much more slowly than the recall process does.

4.3.6 Complex-Valued Associative Memory

Now we obtain a complex-valued associative memory on the basis of the real-valued one mentioned above. First, we consider that the input vector, output vector, and weight matrix are complex-valued, i.e., $x_i \in \boldsymbol{C}$, $y_i \in \boldsymbol{C}$, and $w_{ji} \in \boldsymbol{C}$ where $\boldsymbol{C}$ is the complex number set. When we adopt a saturation-characteristic activation function, we can consider that the elements of input and output vectors are limited as $|x_i| \leq 1$ and $|y_j| \leq 1$.

The task that a complex-valued associative memory processes is the same as that of a real-valued memory except for the fact that the input and output vectors are complex-valued vectors. According to the discussion in Section 3.3, we adopt the amplitude-phase-type activation function $f(u)$ expressed as

$$f_{\mathrm{ap}}(u) = \tanh(|u|)\ \exp(i \arg(u)) \tag{4.27}$$

Therefore, the elements of finally recalled vectors $\boldsymbol{y}$ become $|y_j| = 1$. That is, $y_j = e^{i\theta_j}$. The elements of memorized vectors $\boldsymbol{s}$ are also $|s_{i\mu}| = 1$, i.e., $s_{i\mu} = e^{i\theta_i}$.

In a batch construction of the weight matrix, we determine it in the same manner as we did in (4.11) as

$$\begin{aligned} \mathbf{W} &\equiv \left[\begin{array}{c|c|c|c} & & & \\ \boldsymbol{s}_1 & \boldsymbol{s}_2 & \cdots & \boldsymbol{s}_{\overline{\mu}} \\ & & & \end{array}\right] \left[\begin{array}{c} (\boldsymbol{s}_1)^* \\ \hline (\boldsymbol{s}_2)^* \\ \hline \vdots \\ \hline (\boldsymbol{s}_{\overline{\mu}})^* \end{array}\right] \\ &\equiv \mathbf{S}\mathbf{S}^* \end{aligned} \tag{4.28}$$

where $(\cdot)^*$ denotes conjugate transpose vector or conjugate transpose matrix (hermitian conjugate). Note that *a transpose in real-valued neural network should be a conjugate transpose in complex-valued one. This is one of the most basic operations to construct complex-valued networks.* Such an operation is quite universal and seen in various fields such as quantum dynamics. This operation also means time reversal or space reversal. When we consider the correlation between two quantities (or a squared value of a quantity), we take the product of one value and the conjugate transpose of the other one (or the identical one).

The energy correpsonding to (4.21) is defined as follows. The metric matrix $\mathbf{W}$ becomes hermitian, which yields a real-valued energy.

$$E(\boldsymbol{y}; \mathbf{W}) = -\frac{1}{2}\boldsymbol{y}^*\mathbf{W}\boldsymbol{y} \tag{4.29}$$

When we use a generalized inverse matrix, we again replace the transpose in (4.23) with conjugate transpose as

$$\mathbf{W} \equiv \mathbf{S}(\mathbf{S}^* \ \mathbf{S})^{-1}\mathbf{S}^* \tag{4.30}$$

The complex-valued correlation learning is also formally identical with the complex-valued Hebbian rule. That is, we modify (4.9) as

$$\tau\frac{d}{dt}\mathbf{W} = -\mathbf{W} + \boldsymbol{s}_\mu \ (\boldsymbol{s}_\mu)^* \qquad \text{(continuous time)} \tag{4.31}$$

$$\mathbf{W}(t+1) = (1-K)\mathbf{W} + K \ \boldsymbol{s}_\mu \ (\boldsymbol{s}_\mu)^* \qquad \text{(discrete time)} \tag{4.32}$$

We present $\boldsymbol{s}_\mu$ one after another to make the complex-valued neural network learn correlation.

The above is the construction, processing, and learning in complex-valued associative memories.

4.3.7 Amplitude–Phase Expression of Hebbian Rule

The complex-valued Hebbian rule, or precisely speaking, the complex-valued correlation learning, is expressed by (4.31) or (4.32). When we implement the learning as software or a physical setup dealing with lightwave or electromagnetic wave, we vary w_{ji} on the complex plane by designating the span and direction of the small shift on the plane. The simplest way is to change the real and imaginary parts. That is, every time when a vector to be memorized $\boldsymbol{s}_\mu = [s_{i\mu}]$ is presented to the network, we vary the real and imaginary parts of the weights $\mathbf{W} = [w_{ji}]$ as

$$\begin{aligned}\tau\frac{d}{dt}\mathbf{Re}(w_{ji}) &= -\mathbf{Re}(w_{ji}) + \mathbf{Re}(s_{j\mu}\ (s_{i\mu})^*)\\ \tau\frac{d}{dt}\mathbf{Im}(w_{ji}) &= -\mathbf{Im}(w_{ji}) + \mathbf{Im}(s_{j\mu}\ (s_{i\mu})^*)\end{aligned}$$

(continuous time) (4.33)

$$\begin{aligned}\mathbf{Re}(w_{ji})(t+1) &= (1-K)\mathbf{Re}(w_{ji}) + K\mathbf{Re}(\ s_{j\mu}\ (s_{i\mu})^*)\\ \mathbf{Im}(w_{ji})(t+1) &= (1-K)\mathbf{Im}(w_{ji}) + K\mathbf{Im}(\ s_{j\mu}\ (s_{i\mu})^*)\end{aligned}$$

(discrete time) (4.34)

Or we may rather vary the amplitude and phase of the weights, which matches the amplitude-phase-type activation function. Moreover, when we construct a lightwave or electromagnetic-wave system, we actually utilize amplitude modulators and phase modulators. The changing rule of the amplitude and phase values, $|w_{ji}|$ and θ_{ji}, is obtained for the weights $w_{ji} \equiv |w_{ji}|e^{i\theta_{ji}}$ and vectors $s_i \equiv |s_i|e^{i\theta_i}$ (μ is omitted here) as

$$\begin{aligned}\tau\frac{d}{dt}w_{ji} &= -w_{ji} + s_j(s_i)^*\\ \tau\frac{d}{dt}\left(|w_{ji}|e^{i\theta_{ji}}\right) &= -|w_{ji}|e^{i\theta_{ji}} + |s_j|e^{i\theta_j}\ |s_i|e^{-i\theta_i}\\ &= |w_{ji}|e^{i\theta_{ji}}\ \left(-1 + \frac{|s_j||s_i|}{|w_{ji}|}e^{i(\theta_j-\theta_i-\theta_{ji})}\right)\\ &= |w_{ji}|e^{i\theta_{ji}}\ \Bigg(\underbrace{-1 + \frac{|s_j||s_i|}{|w_{ji}|}\cos(\theta_j-\theta_i-\theta_{ji})}_{\text{change in amplitude direction } \Delta(|w_{ji}|)}\\ &\qquad + \underbrace{i\frac{|s_j||s_i|}{|w_{ji}|}\sin(\theta_j-\theta_i-\theta_{ji})}_{\text{change in phase direction } \Delta\arg(w_{ji})}\Bigg)\end{aligned} \quad (4.35)$$

Accordingly, considering the relationship among the vectors shown in Fig.4.5, we find that the real and imaginary parts in the curly brackets correspond

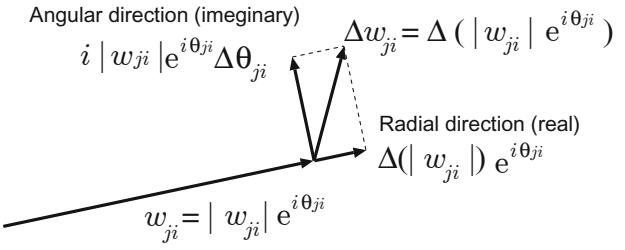


Fig. 4.5 Relationship among vectors expressing the changes in the amplitude and phase of the weight in the complex-valued Hebbian rule.

to the amplitude and phase variations, respectively. That is, the amplitude change is written as

$$\tau\frac{d|w_{ji}|}{dt} = |w_{ji}|\left(-1+\frac{|s_j||s_i|}{|w_{ji}|}\cos(\theta_j-\theta_i-\theta_{ji})\right)$$

Therefore, we obtain

$$\tau\frac{d|w_{ji}|}{dt} = -|w_{ji}|+|s_j||s_i|\cos(\theta_j-\theta_i-\theta_{ji}) \tag{4.36}$$

The phase change is also obtained as

$$\tau\frac{d\theta_{ji}}{dt} = \frac{|s_j||s_i|}{|w_{ji}|}\sin(\theta_j-\theta_i-\theta_{ji}) \tag{4.37}$$

These changes in amplitude, $|w_{ji}|$, and phase, θ_{ji}, realize the complex-valued Hebbian learning. These expressions are compatible with the amplitude-phase-type activation function. The discrete-time version is expressed just like (4.5) with a real learning constant K $(0 < K < 1)$ as

$$|w_{ji}|(t+1) = (1-K)|w_{ji}|(t)+K|s_j||s_i|\cos(\theta_j-\theta_i-\theta_{ji}) \tag{4.38}$$

$$\theta_{ji}(t+1) = K\frac{|s_j||s_i|}{|w_{ji}|}\sin(\theta_j-\theta_i-\theta_{ji}) \tag{4.39}$$

4.3.8 Lightwave Neural Networks and Carrier-Frequency-Dependent Learning

Among various applications presented in Part II, we will deal with lightwave neural networks in Chapters 8 and 9. They treat information utilizing coherence of lightwave emitted by lasers. We call this class of complex-valued neural networks *coherent neural networks*.

Let us consider an optical coherent neural network. The phase value of a propagating lightwave at a point is determined by optical carrier frequency and optical path length, i.e., the delay time of propagation. This fact enables

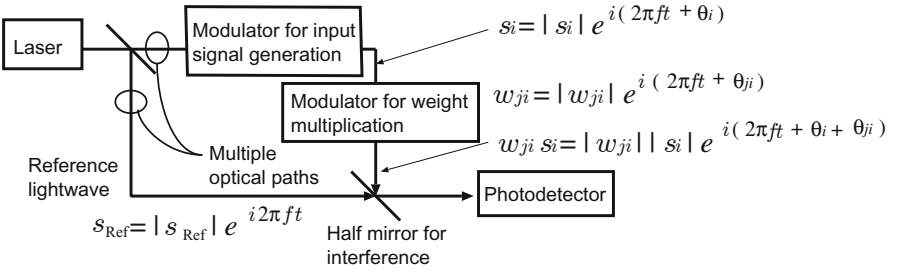


Fig. 4.6 A lightwave neural network forming a self-homodyne optical circuit.

us to realize carrier-frequency-dependent learning, self-organization, and task processing in the coherent neural network. The carrier frequency is identical with the oscillation frequency of the laser. When we use a semiconductor laser, the oscillation frequency can easily be modulated by the change in the injection current (frequency modulation: FM). We can modulate, so to speak, the *behavior* of the neural network by changing the carrier frequency as follows.

The carrier frequency is f. (Please do not confuse f here and the activation function.) We consider an optical self-homodyne circuit shown in Fig.4.6. It is also called Mach-Zehnder interferometer that is the most basic interferometer for two lightwaves generated at an identical light source. The modulators in Fig.4.6 modulates the phase (or, actually, the delay time) and the amplitude of the lightwave. We express the signal s_i, weight w_{ji}, and reference wave s_{Ref} as follows, respectively, by taking the phase of s_{Ref} as the phase reference.

$$s_i = |s_i| \exp(i(2\pi ft + \theta_i)) \tag{4.40}$$

$$w_{ji} = |w_{ji}| \exp(i(2\pi ft + \theta_{ji})) \tag{4.41}$$

$$s_{\mathrm{Ref}} = |s_{\mathrm{Ref}}| \exp(i(2\pi ft)) \tag{4.42}$$

The signal s_i, generated by the modulator for input signal generation, is modulated again by the modulator for weight multiplication to become

$$w_{ji} s_i = |w_{ji}||s_i| \exp(i(2\pi ft + \theta_i + \theta_{ji})) \tag{4.43}$$

Then it is superimposed on the reference wave for the homodyning at the half mirror as

$$|w_{ji}||s_i| \exp(i(2\pi ft + \theta_i + \theta_{ji}) + |s_{\mathrm{Ref}}| \exp(i(2\pi ft)) \tag{4.44}$$

By detecting the mixed lightwave with a square-law detector, we obtain a detection current I as

$$\begin{aligned} I &\propto \left| |w_{ji}||s_i| \exp(i(2\pi ft + \theta_i + \theta_{ji})) + |s_{\mathrm{Ref}}| \exp(i(2\pi ft)) \right|^2 \\ &= |w_{ji}|^2 |s_i|^2 + |s_{\mathrm{Ref}}|^2 + 2|w_{ji}||s_i||s_{\mathrm{Ref}}| \cos(\theta_i + \theta_{ji}) \end{aligned} \tag{4.45}$$

If the optical power of the reference lightwave is sufficiently large, $|s_{\mathrm{Ref}}| \gg |s_i||w_{ji}|$, then the *dc* component, $|w_{ji}|^2 |s_i|^2 + |s_{\mathrm{Ref}}|^2$, is almost unchanged

even if the signal or the weight changes. By subtracting the *dc* component, we obtain the following signal current as the resulting information of the interference.

$$I \propto |w_{ji}||s_i| \cos(\theta_i + \theta_{ji}) \tag{4.46}$$

Consequently, the optical circuit produces the product of the signal and the weight quite simply.

Therefore, we can construct a lightwave neural network by preparing optical circuits mentioned above in parallel. For instance, when we build an associative memory, we present signal vectors to be memorized $\boldsymbol{s}$ at the inputs and outputs so that the network learns the correlation. Then the learning dynamics of the absolute value of the weight $|w_{ji}|$ is expressed directly by the amplitude-phase-type Hebbian rule (4.36) or by the discrete version (4.38). That of the phase value, in (4.37) or (4.39), requires a modification because an actual modulator modulates the delay time, τ_{ji} (which is not the learning time constant τ), instead of the phase value itself. We rewrite the equations by using the relationship among the phase θ_{ji}, carrier frequency f, and delay time τ_{ji}, namely $\theta_{ji} = 2\pi f \tau_{ji}$, as

$$\tau \frac{d|w_{ji}|}{dt} = -|w_{ji}| + |s_i||s_j| \cos(\theta_j - \theta_i - 2\pi f \tau_{ji}) \tag{4.47}$$

$$\tau \frac{d\tau_{ji}}{dt} = \frac{1}{2\pi f} \frac{|s_i||s_j|}{|w_{ji}|} \sin(\theta_j - \theta_i - 2\pi f \tau_{ji}) \tag{4.48}$$

The discrete version is obtained as

$$|w_{ji}|(t+1) = (1-K)|w_{ji}|(t) + K|s_i||s_j| \cos(\theta_j - \theta_i - 2\pi f \tau_{ji}) \tag{4.49}$$

$$\tau_{ji}(t+1) = \tau_{ji}(t) + \frac{K}{2\pi f} \frac{|s_i||s_j|}{|w_{ji}|} \sin(\theta_j - \theta_i - 2\pi f \tau_{ji}) \tag{4.50}$$

Please pay attention to the following features in the above learning. The arguments of cos and sin include the product of the carrier frequency f and delay time τ_{ji}. The typical optical-carrier frequency is very high, say, 400THz for GaAs diode lasers. Therefore, the delay time is much influential to the phase value. The change in f also causes the variation in the phase value. Available frequency range of modulation is, e.g., a few THz, which is much smaller than 400THz but very large as an information-processing bandwidth. On the other hand, though τ_{ji} includes f also in the denominator as $1/(2\pi f)$, the change in f in this term is relatively so small that it has no distinct effect on the learning behavior.

Chapters 8 and 9 present lightwave neural networks whose behavior can be modulated by the carrier-frequency modulation based on the frequency-dependent learning mentioned above. A bicycle-riding robot shown in Chapter 10 is also based on the identical principle. Other applications include microwave or millimeter-wave information processing in, for example, active antennas and radar systems.

4.4 Function Approximation

4.4.1 Function: Generation of Desirable Outputs for Given Inputs

Function approximation is a task to generate desirable output signals $\boldsymbol{y}$ for given input signals $\boldsymbol{x}$. In particular, when a set of limited number of input-output pairs $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ are presented to the network, the network is expected, in a function-approximation task, to yield appropriate output signals even for unlearned signals with interpolation and extrapolation. Such an ability to meaningfully correspond even to inputs other than teachers is called *generalization*. The quality of generalization is the *generalization characteristics*. If a generalization characteristic meets with a processing purpose, it is a good generalization characteristic. Obtaining a good generalization characteristic leads to a skillful prediction and estimation against unknown environment. Prediction and estimation are difficult problems for symbolic processors. Contrarily, neural networks are good at solving them since they construct their landscape expressing the world.

4.4.2 Network Construction and Processing Dynamics

Figure 4.7 shows the network construction. Neurons and connections form a single or multiple layers. Such a network is called layered neural network. If a network consists of single layers of input terminals, connections, and output neurons, it is a single-layered neural network. A network having L pairs of connection and neuron layers, with a single input layer, is called a L-layered neural network.

The input-terminal layer is called the input layer, while the output-neuron layer is called the output layer. If the network has a single or multiple layers of connection- and neuron-layer pairs besides the output layer, the layers are called the hidden layers.

A network having an input layer (input terminals), a hidden layer, and an output layer is called a two-layered neural network.[1] It is known that, in the case of real-valued neural networks, any logic function can be realized with a network having two layers at least (i.e., single hidden layer at least) if the number of hidden-layer neurons is sufficiently large. The reason is as follows. A logic function determines the output as "1" if the weighted sum is larger than a threshold, whereas, if not, the network generates "0". A neural network having a steep activation function, such as a step function, works as a logic function. A high threshold level realizes AND operation, whereas low threshold level does an OR. A single neuron corresponds to a logic operation

[1] Formerly, we called it a three-layered network by taking the input layer into account.

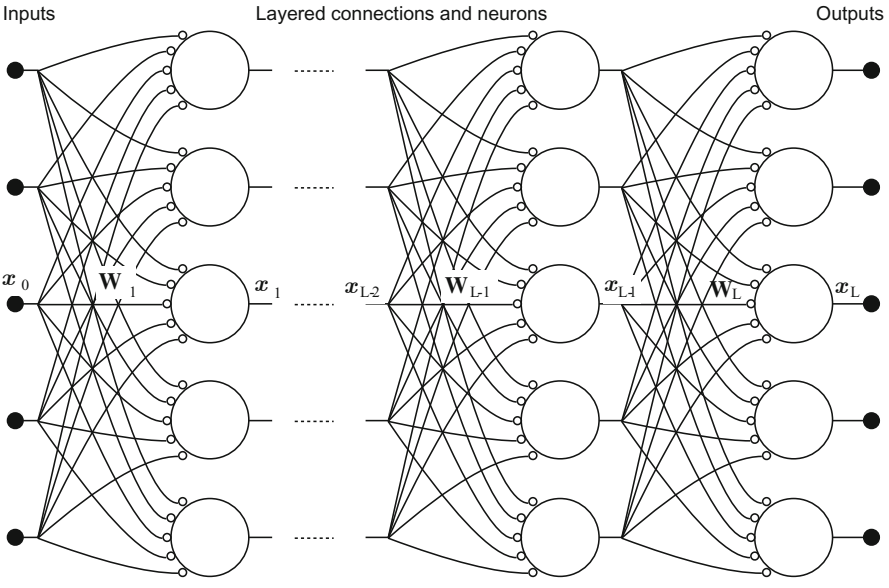


Fig. 4.7 Construction of layered neural network.

such as AND or OR, which divides input information space simply into "1"- and "0"-output regions (linear separation). If the hidden layer is capable of preparing sufficiently various ways of linear separation (i.e., if the hidden layer has a sufficiently large number of neurons), the output neurons are able to synthesize any desirable partition of the space by the following weighting and summing operation at the output layer. That is to say, the hidden layer provides freedom in the way of linking of "1" or "0" regions (topology).

The connection weights vary in learning. The historically well-known *perceptron* is a two-layer neural network whose weights in the hidden layer (input terminals – hidden neurons) are fixed at certain values at random, while the weights at the output layer (hidden neurons – output neurons) are variable in learning. If the network has a large number of hidden neurons, it is expected that the hidden layer has prepared various linear-separation operation. Then the output layer can select needed hidden outputs as its inputs that are sufficiently useful to generate desirable output characteristic. However, because the perceptron does not possess the mean to make the hidden layer learn, it requires an extremely large number of hidden neurons. One of the learning methods in hidden layers is the so-called backpropagation learning mentioned below.

The task processing dynamics in a layered neural network is explained generally as follows. Input signals $\boldsymbol{x}_0$ is fed to the input terminals, weighted by connection weights $\mathbf{W}_1$, and summed. The weighted sum is fed to the activation function f, and an output of the neuron $\boldsymbol{x}_1$ is detemined.

$$\boldsymbol{x}_1 = f(\mathbf{W}_1 \boldsymbol{x}_0) \tag{4.51}$$

By repeating this operation, we obtain output signals at the output layer.

$$\begin{aligned} \boldsymbol{x}_2 &= f(\mathbf{W}_2 \boldsymbol{x}_1) \\ \boldsymbol{x}_3 &= f(\mathbf{W}_3 \boldsymbol{x}_2) \\ &\vdots \\ \boldsymbol{x}_l &= f(\mathbf{W}_l \boldsymbol{x}_{l-1}) \\ &\vdots \\ \boldsymbol{x}_L &= f(\mathbf{W}_L \boldsymbol{x}_{L-1}) \end{aligned} \tag{4.52}$$

This is the task-processing operation in a layered neural network.

4.4.3 Learning by Steepest Descent Method

The steepest descent method is an optimization method in multiple parameter systems to use partial derivatives to determine the direction of parameter variation. Let us consider a problem in which we want to minimize an error function E expressed with parameters $\boldsymbol{w} \equiv [w_i]$. The rate of change of E in terms of w_i is derived as the partial derivative $\partial E/\partial w_i$. Then, the vector $\partial E/\partial \boldsymbol{w} \equiv [\partial E/\partial w_i]$ points to the direction in which the error function E increases in the parameter space $\boldsymbol{w}$. Therefore, the opposite vector $-\partial E/\partial \boldsymbol{w}$ indicates the path in which the error E reduces most efficiently. That is, we can decrease E by changing the parameter $\boldsymbol{w}$ in the direction of $-\partial E/\partial \boldsymbol{w}$. This method is called the steepest descent method.

The steepest descent method realizes the learning in weights at a layer (l-th layer), or those in a single-layered network, as follows. We prepare a set of input and output teacher pairs $(\hat{\boldsymbol{x}}_{l-1}, \hat{\boldsymbol{x}}_l)$. We define an error function E_l as the squared difference between output teachers $\hat{\boldsymbol{x}}_l$ and temporary outputs $\boldsymbol{x}_l(\hat{\boldsymbol{x}}_{l-1}; \mathbf{W}_l)$ determined for the input teachers $\hat{\boldsymbol{x}}_{l-1}$. Since, usually, we have multiple teacher signal pairs $(\hat{\boldsymbol{x}}_{l-1\ \mu}, \hat{\boldsymbol{x}}_{l\ \mu})$, the error function E_l is defined as

$$E_l \equiv \frac{1}{2} \sum_{\mu} (\boldsymbol{x}_{l\ \mu} - \hat{\boldsymbol{x}}_{l\ \mu})^2 \equiv \frac{1}{2} (\boldsymbol{x}_l - \hat{\boldsymbol{x}}_l)^2 \tag{4.53}$$

where μ is the index for each pair. However, in the most right-hand side in (4.53), we omit μ only for simplicity in expression.

We change the connection weights $\mathbf{W}_l$ in such a way that we decrease the error E_l. In general, the values $\partial E_l/\partial w_{jil}$ show the rates of changes of E_l in terms of w_{jil}. That is, a set of the values $\partial E_l/\partial w_{jil}$ points to the direction in which E_l increases in $[w_{jil}]$ space. Therefore, we change $\mathbf{W}_l$ in the opposite direction, namely, $-\partial E_l/\partial \mathbf{W}_l$. A weight w_{jil} should be changed as

$$\begin{aligned}\tau\frac{dw_{jil}}{dt} &= -\frac{\partial E_l}{\partial w_{jil}} \\ &= -\frac{\partial}{\partial w_{jil}}\left(\frac{1}{2}(\boldsymbol{x}_l - \hat{\boldsymbol{x}}_l)^2\right) \\ &= -(x_{jl} - \hat{x}_{jl})\frac{\partial x_{jl}}{\partial w_{jil}} \\ &= -(x_{jl} - \hat{x}_{jl})f'(\sum_i w_{jil}\ x_{i\ l-1})x_{i\ l-1} \end{aligned} \tag{4.54}$$

where τ and f' denote learning time constant and derivative function of f, respectively. The derivative is calculated for $f(u) = \tanh(u)$ as

$$f'(u) = (\tanh(u))' = \mathrm{sech}^2(u) = 1 - \tanh^2(u) = 1 - (f(u))^2 \tag{4.55}$$

We derive

$$\tau\frac{dw_{jil}}{dt} = -(x_{jl} - \hat{x}_{jl})(1 - x_{jl}^2)x_{i\ l-1} \tag{4.56}$$

The discrete-time version is expressed with a learning time constant K as

$$w_{jil}(t+1) = w_{jil}(t) - K(x_{jl} - \hat{x}_{jl})(1 - x_{jl}^2)x_{i\ l-1} \tag{4.57}$$

In this manner, we can realize the learning if we know the output error $(x_{jl} - \hat{x}_{jl})$, temporary outputs x_{jl}, and temporary inputs $x_{i\ l-1}$. The neurons also know these values locally. Therefore, the learning is performed locally and distributedly.

In practice, we have multiple teacher signal pairs $(\hat{\boldsymbol{x}}_{l-1\ \mu}, \hat{\boldsymbol{x}}_{l\ \mu})$ as noted in relation to (4.53). We present teacher pairs to the network one after another, and reduce gradually the total error $(1/2)\sum_\mu(\boldsymbol{x}_{l\ \mu} - \hat{\boldsymbol{x}}_{l\ \mu})^2$.

4.4.4 Backpropagation Learning

The backpropagation (BP) learning is a supervised learning method to realize the learning in all hidden layers by making output-teacher information propagate backward to the preceding layers. In particular, in error backpropagation learning, the difference between the teacher signals and the temporary output signals propagates layer by layer. Usually, "backpropagation" or "BP" refers to this "error backpropagation."

In the steepest descent method, we obtained the rule to change the weights in l-th layer by presenting a set of teacher signals at the layer's outputs. Then, how can we change the weights in $(l-1)$-th layer? First, let us see the task-processing operation related to these layers.

$$\begin{aligned}\boldsymbol{x}_{l-1} &= f(\mathbf{W}_{l-1}\boldsymbol{x}_{l-2}) \\ \boldsymbol{x}_l &= f(\mathbf{W}_l\boldsymbol{x}_{l-1}) = f(\mathbf{W}_l f(\mathbf{W}_{l-1}\boldsymbol{x}_{l-2})) \end{aligned} \tag{4.58}$$

Therefore, we can extract the influence of the change in a certain weight $w_{ji\ l-1}$ in the preceding weights $\mathbf{W}_{l-1}$ on the $(l-1)$-th layer's outputs $\boldsymbol{x}_l$ as

$$x_{kl} = f(\sum_j (w_{kjl} \underbrace{f(\sum_i w_{ji\ l-1} x_{i\ l-2})}_{x_{j\ l-1}})) \qquad \text{(for every } k\text{)} \tag{4.59}$$

where $(\cdot)_{kj}$ are subscript for l-th layer's weights. The vector expression is obtained as

$$\boldsymbol{x}_l = f(\ \boldsymbol{w}_{jl} \underbrace{f(\sum_i w_{ji\ l-1} x_{i\ l-2})}_{x_{j\ l-1}}\ ) \tag{4.60}$$

where $\boldsymbol{w}_{jl}$ is the j-th column elements in $\mathbf{W}_l = [w_{kjl}]$, forming a vertical vector $\boldsymbol{w}_{jl} \equiv [w_{kjl}]$. Therefore, the steepest descent process changes an element $w_{ji\ l-1}$ in $\mathbf{W}_{l-1} = [w_{ji\ l-1}]$ as

$$\begin{aligned}
\tau \frac{dw_{ji\ l-1}}{dt} &= -\frac{\partial E_l}{\partial w_{ji\ l-1}} \\
&= -\left(\sum_k \frac{\partial E_l}{\partial x_{kl}} \frac{\partial x_{kl}}{\partial x_{j\ l-1}} \right) \frac{\partial x_{j\ l-1}}{\partial w_{ji\ l-1}} \\
&= -\left(\sum_k \left\{ \frac{\partial \{(x_{kl} - \hat{x}_{kl})^2/2\}}{\partial x_{kl}} \frac{\partial \overbrace{f(\sum_j w_{kjl} x_{j\ l-1})}^{x_{kl}}}{\partial x_{j\ l-1}} \right\} \right) \\
&\qquad \times \left(\frac{\partial \overbrace{f(\sum_i w_{ji\ l-1} x_{i\ l-2})}^{x_{j\ l-1}}}{\partial w_{ji\ l-1}} \right) \\
&= -\Big(\sum_k (x_{kl} - \hat{x}_{kl})(1 - x_{kl}^2) w_{kjl} \Big) (1 - x_{j\ l-1}^2) x_{i\ l-2}
\end{aligned} \tag{4.61}$$

The treatment in the last step in (4.61) is identical with that in the steepest descent method. We can consider this process as a backward propagation of the error $(x_{kl} - \hat{x}_{kl})$. Iteration of this process realizes learning in all the layers.

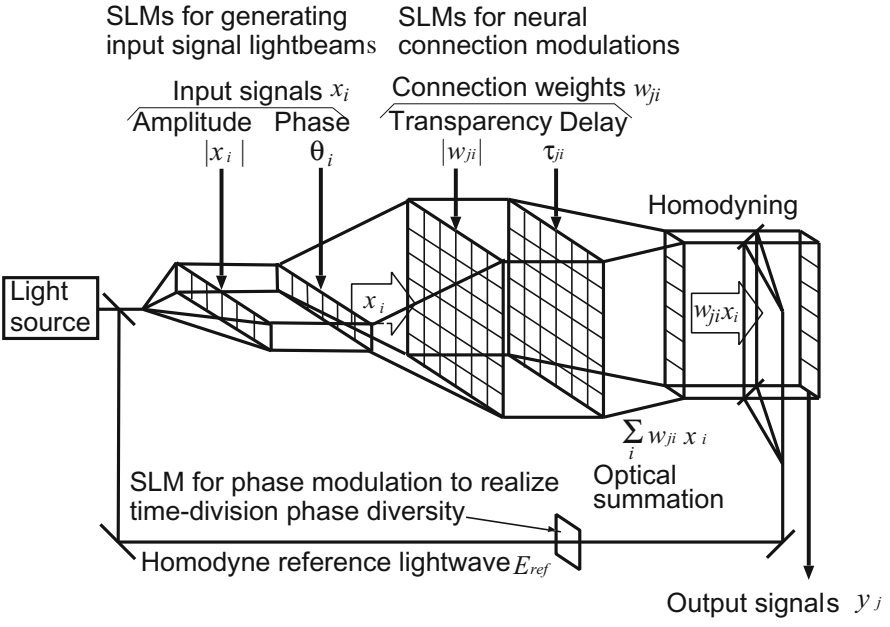


Fig. 4.8 Schematic illustration of single-layered coherent lightwave neural network. (Detail in coherent optical summation is omitted.)

The discrete-time version of (4.61) is expressed with learning time constant K as

$$w_{ji\ l-1}(t+1) = w_{ji\ l-1}(t) - K\Big(\sum_k (x_{kl} - \hat{x}_{kl})(1 - x_{kl}^2) w_{kjl}\Big) \times (1 - x_{j\ l-1}^2) x_{i\ l-2} \tag{4.62}$$

4.4.5 *Learning by Complex-Valued Steepest Descent Method*

Next, we consider the complex-valued steepest descent method [2] in a single-layered coherent neural network shown in Fig.4.8. The laser beam is divided into halves. One is used as the signal light and the other is the reference for homodyning. The signal light is modulated in the amplitude and phase to represent input signals $\boldsymbol{x} = [x_i]$ physically as

$$x_i = |x_i|\ e^{i\theta_i} \tag{4.63}$$

[2] The complex-valued BP method that is compatible rather with the real-imaginary-type activation functions is derived in a similar manner as that in the real-valued BP. See, for example, [48], [79], etc.

Then the signal lightwaves are modulated again in the amplitude and phase corresponding to the multiplication by the connection weights $\mathbf{W} = [w_{ji}]$. The weights are composed of transparency $|w_{ji}|$ and time delay $|\tau_{ji}|$ as described in Section 4.3.8. For the meantime, θ_{ji} represents the phase shift caused by the time delay.

$$w_{ji} = |w_{ji}|\ e^{i2\pi f \tau_{ji}} = |w_{ji}|\ e^{i\theta_{ji}} \tag{4.64}$$

The modulated signals are expressed as the products of the signals and the weights, and coherently summed in, e.g., a star coupler, to be the internal states of the neurons $\boldsymbol{u} = [u_j]$ written as

$$u_j = \sum_i w_{ji} x_i = \sum_i |w_{ji}||x_i|\ e^{i(\theta_{ji}+\theta_i)} \tag{4.65}$$

According to the discussion in Section 3.3.5, we adopt an amplitude-phase-type activation function f_{ap}. The function presents saturation in amplitude, but no change in phase. Then the outputs $\boldsymbol{y} = [y_j]$ are obtained as

$$\begin{aligned} y_j &= f_{\mathrm{ap}}(u_j) \\ &= \tanh\left(|u_j|\right)\ \exp\left(i \arg\left(u_j\right)\right) \end{aligned} \tag{4.66}$$

Let us consider how to realize steepest descent learning by presenting a set of teacher signals to the system mentioned above. As we did in the previous section, we define the energy function E as the sum of the squared difference between the teacher signals $\hat{y_j}$ and the temporary output signals y_j as

$$\begin{aligned} E_l &\equiv \frac{1}{2}|\boldsymbol{y} - \hat{\boldsymbol{y}}|^2 \\ &= \frac{1}{2}\sum_j |y_j - \hat{y_j}|^2 \\ &= \frac{1}{2}\sum_j \Big(\tanh^2(|u_j|) + \tanh^2(|\hat{u}_j|) \\ &\qquad - 2\tanh(|u_j|)\tanh(|\hat{u}_j|)\cos(\theta_j - \hat{\theta}_j)\Big) \end{aligned} \tag{4.67}$$

where $\hat{\boldsymbol{u}} = [|\hat{u}_j| e^{i\hat{\theta}_j}]$ is the equivalent internal state corresponding to the teacher signal $\hat{\boldsymbol{y}}$.

The changes in amplitude and phase of the internal state, $|u_j|$ and θ_j, respectively, are orthogonal to each other. Therefore, we obtain an amplitude-phase-type steepest descent change by summing separately-obtained partial derivatives in terms of w_{ji} in these orthogonal directions. The derivative of $\tanh(|u|)$ is $(1 - \tanh^2(|u|))$, as we confirmed in the previous section, while the derivative of the teacher signal is 0. Then we have

$$\frac{\partial E}{\partial(|u_j|)} = \frac{1}{2}\Big(2\tanh(|u_j|)(1-\tanh^2(|u_j|)) - 2(1-\tanh^2(|u_j|))\tanh(|\hat{u}_j|)\cos(\theta_j-\hat{\theta}_j)\Big)$$
$$= (1-|y_j|^2)(|y_j| - |\hat{y}_j|\cos(\theta_j-\hat{\theta}_j)) \tag{4.68}$$
$$\frac{\partial E}{\partial \theta_j} = |y_j||\hat{y}_j|\sin(\theta_j-\hat{\theta}_j) \tag{4.69}$$

How the amplitude and phase of the weighted sum u_j varies when a certain weight $w_{ji'}$ changes? Assume that the weight changes with two fractions, $\Delta(w^{\mathrm{a}}_{ji'})$ and $\Delta(w^{\mathrm{p}}_{ji'})$, in the complex plane as

$$w_{ji'} \longrightarrow w_{ji'} + (\Delta(w^{\mathrm{a}}_{ji'}) + i\Delta(w^{\mathrm{p}}_{ji'}))e^{i(\theta_j-\theta_{i'})} \tag{4.70}$$

Defined in this way, the fractions, $\Delta(w^{\mathrm{a}}_{ji'})$ and $\Delta(w^{\mathrm{p}}_{ji'})$, are related directly to the amplitude $|u_j|$ and the phase θ_j of the internal state u_j, respectively. In other words, we multiply $w_{ji'}$ by the input $x_{i'} \equiv |x_{i'}|e^{i\theta_{i'}}$ to see the change in u_j, i.e., $\Delta(|u_j|e^{i\theta_j})$, as follows.

$$\Delta(|u_j|e^{i\theta_j}) = (\Delta w_{ji'})x_{i'} = (\Delta(w^{\mathrm{a}}_{ji'}) + i\Delta(w^{\mathrm{p}}_{ji'}))|x_{i'}|e^{i\theta_j} \tag{4.71}$$

That is, the phase $(\theta_j - \theta_{i'})$ in (4.70) is multiplied by the input $\theta_{i'}$ to become θ_j in (4.71), being oriented in the same direction as u_j.

Accordingly, we can obtain the two fractions, $\Delta(w^{\mathrm{a}}_{ji'})$ and $\Delta(w^{\mathrm{p}}_{ji'})$, separately and independently. For general $i = i'$, we have

$$\frac{d(|u_j|)}{d(w^{\mathrm{a}}_{ji})} = |x_i| \tag{4.72}$$
$$\frac{d\theta_j}{d(w^{\mathrm{p}}_{ji})} = \frac{|x_i|}{|u_j|} \tag{4.73}$$

Putting them all together, we obtain the partial derivatives in terms of w^{a}_{ji} and w^{p}_{ji} as

$$\frac{\partial E}{\partial w^{\mathrm{a}}_{ji}} = \frac{\partial E}{\partial(|u_j|)}\frac{d(|u_j|)}{dw^{\mathrm{a}}_{ji}}$$
$$= (1-|y_j|^2)(|y_j| - |\hat{y}_j|\cos(\theta_j-\hat{\theta}_j))|x_i| \tag{4.74}$$
$$\frac{\partial E}{\partial w^{\mathrm{p}}_{ji}} = \frac{\partial E}{\partial \theta_j}\frac{d\theta_j}{dw^{\mathrm{p}}_{ji}}$$
$$= |y_j||\hat{y}_j|\sin(\theta_j-\hat{\theta}_j)\frac{|x_i|}{|u_j|} \tag{4.75}$$

Incidentally, the fractions, $\Delta(w^{\mathrm{a}}_{ji})$ and $\Delta(w^{\mathrm{p}}_{ji})$, and the amplitude and phase of the weight, $|w_{ji}|$ and θ_{ji}, have the relationship of (4.70), i.e.,

$$|w_{ji'}|e^{i\theta_{ji'}} \longrightarrow |w_{ji'}|e^{i\theta_{ji'}} + (\Delta(w^{\mathrm{a}}_{ji'}) + i\Delta(w^{\mathrm{p}}_{ji'}))e^{i(\theta_j - \theta_{i'})} \tag{4.76}$$

By dividing the both sides of (4.76) by $e^{i\theta_{ji'}}$, which corresponds to rotation in the complex plane, we derive a relationship, for general $i = i'$, expressed as

$$\begin{bmatrix} \Delta(|w_{ji}|) \\ |w_{ji}|\Delta\theta_{ji} \end{bmatrix} = \begin{bmatrix} \cos\theta^{\mathrm{rot}}_{ji} & -\sin\theta^{\mathrm{rot}}_{ji} \\ \sin\theta^{ji}_{\mathrm{rot}} & \cos\theta^{\mathrm{rot}}_{ji} \end{bmatrix} \begin{bmatrix} \Delta(w^{\mathrm{a}}_{ji}) \\ \Delta(w^{\mathrm{p}}_{ji}) \end{bmatrix} \tag{4.77}$$

where

$$\theta^{\mathrm{rot}}_{ji} \equiv \theta_j - \theta_i - \theta_{ji} \tag{4.78}$$

The explicit expression for $\Delta(w^{\mathrm{a}}_{ji})$ and $\Delta(w^{\mathrm{p}}_{ji})$ is written as

$$\begin{bmatrix} \Delta(w^{\mathrm{a}}_{ji}) \\ \Delta(w^{\mathrm{p}}_{ji}) \end{bmatrix} = \begin{bmatrix} \cos\theta^{\mathrm{rot}}_{ji} & \sin\theta^{\mathrm{rot}}_{ji} \\ -\sin\theta^{\mathrm{rot}}_{ji} & \cos\theta^{\mathrm{rot}}_{ji} \end{bmatrix} \begin{bmatrix} \Delta(|w_{ji}|) \\ |w_{ji}|\Delta\theta_{ji} \end{bmatrix} \tag{4.79}$$

which leads to

$$\begin{aligned} \frac{\partial E}{\partial(|w_{ji}|)} &= \frac{\partial E}{\partial w^{\mathrm{a}}_{ji}} \frac{\partial w^{\mathrm{a}}_{ji}}{\partial(|w_{ji}|)} + \frac{\partial E}{\partial w^{\mathrm{p}}_{ji}} \frac{\partial w^{\mathrm{p}}_{ji}}{\partial(|w_{ji}|)} \\ &= \frac{\partial E}{\partial w^{\mathrm{a}}_{ji}} \cos\theta^{\mathrm{rot}}_{ji} - \frac{\partial E}{\partial w^{\mathrm{p}}_{ji}} \sin\theta^{\mathrm{rot}}_{ji} \end{aligned} \tag{4.80}$$

$$\begin{aligned} \frac{\partial E}{\partial \theta_{ji}} &= \frac{\partial E}{\partial w^{\mathrm{a}}_{ji}} \frac{\partial w^{\mathrm{a}}_{ji}}{\partial \theta_{ji}} + \frac{\partial E}{\partial w^{\mathrm{p}}_{ji}} \frac{\partial w^{\mathrm{p}}_{ji}}{\partial \theta_{ji}} \\ &= \frac{\partial E}{\partial w^{\mathrm{a}}_{ji}} |w_{ji}| \sin\theta^{\mathrm{rot}}_{ji} + \frac{\partial E}{\partial w^{\mathrm{p}}_{ji}} |w_{ji}| \cos\theta^{\mathrm{rot}}_{ji} \end{aligned} \tag{4.81}$$

Therefore, the rule of the changes is obtained as

$$\begin{aligned} \tau \frac{d(|w_{ji}|)}{dt} &= -\frac{\partial E}{\partial(|w_{ji}|)} \\ &= -\left(\frac{\partial E}{\partial w^{\mathrm{a}}_{ji}} \cos\theta^{\mathrm{rot}}_{ji} - \frac{\partial E}{\partial w^{\mathrm{p}}_{ji}} \sin\theta^{\mathrm{rot}}_{ji} \right) \end{aligned} \tag{4.82}$$

$$\begin{aligned} \tau \frac{d\theta_{ji}}{dt} &= -\frac{1}{|w_{ji}|} \frac{\partial E}{\partial \theta_{ji}} \\ &= -\left(\frac{\partial E}{\partial w^{\mathrm{a}}_{ji}} \sin\theta^{\mathrm{rot}}_{ji} + \frac{\partial E}{\partial w^{\mathrm{p}}_{ji}} \cos\theta^{\mathrm{rot}}_{ji} \right) \end{aligned} \tag{4.83}$$

where the partial derivatives in the right-hand sides are given in (4.74) and (4.75). Finally, the learning rule for the amplitude $|w_{ji}|$ and the phase θ_{ji} of the weight is expressed as

$$\tau\frac{d(|w_{ji}|)}{dt} = -\frac{\partial E}{\partial(|w_{ji}|)}$$
$$= -\Big((1-|y_j|^2)(|y_j| - |\hat{y}_j|\cos(\theta_j - \hat{\theta}_j))|x_i|\cos\theta_{ji}^{\mathrm{rot}}$$
$$- |y_j||\hat{y}_j|\sin(\theta_j - \hat{\theta}_j)\frac{|x_i|}{|u_j|}\sin\theta_{ji}^{\mathrm{rot}}\Big) \tag{4.84}$$

$$\tau\frac{d\theta_{ji}}{dt} = -\frac{1}{|w_{ji}|}\frac{\partial E}{\partial\theta_{ji}}$$
$$= -\Big((1-|y_j|^2)(|y_j| - |\hat{y}_j|\cos(\theta_j - \hat{\theta}_j))|x_i|\sin\theta_{ji}^{\mathrm{rot}}$$
$$+ |y_j||\hat{y}_j|\sin(\theta_j - \hat{\theta}_j)\frac{|x_i|}{|u_j|}\cos\theta_{ji}^{\mathrm{rot}}\Big) \tag{4.85}$$

Though the expressions are slightly complicated thanks to the rotation operation, the learning is found distributed because the learning rule uses only locally obtainable variables, which is what a neural network should do. The rotation angle $\theta_{ji}^{\mathrm{rot}}$ defined in (4.78) is the difference between temporary output phase, θ_j, and the phase of the input we focus on, which is weighted but *not yet* summed, $\theta_i + \theta_{ji}$. In other words, it is the difference between the phase of summation result and the phase of each weighted input before summation, which is theoretically obtainable as the interference fringe of these waves. Hence, the phase information is locally available, and its meaning is clear.

We also find that the partial derivatives in (4.74) and (4.75) are docile and similar to the expressions of the complex-valued Hebbian rule. We can grasp the meaning as follows. In these equations, the value $\theta_j - \hat{\theta}_j$ is obtained as the difference of the temporary output phase θ_j and the teacher phase $\hat{\theta}_j$. The difference is, as described in the next section, available as the interference fringe of the teacher and output signals when we make the *teacher signal itself* propagate backward in the layered neural network, *instead of the error signal*. The above rotation angle $\theta_{ji}^{\mathrm{rot}}$ is also obtainable as a fringe if the teacher wave propagates backward.

The discrete-time version of the complex-valued steepest descent method is obtained in the same manner as that of the complex-valued Hebbian rule by rewriting (4.84) and (4.85) as

$$|w_{ji}|(t+1) = |w_{ji}|(t)$$
$$- K\Big((1-|y_j|^2)(|y_j| - |\hat{y}_j|\cos(\theta_j - \hat{\theta}_j))|x_i|\cos\theta_{ji}^{\mathrm{rot}}$$
$$- |y_j||\hat{y}_j|\sin(\theta_j - \hat{\theta}_j)\frac{|x_i|}{|u_j|}\sin\theta_{ji}^{\mathrm{rot}}\Big) \tag{4.86}$$

$$\theta_{ji}(t+1) = \theta_{ji}(t)$$
$$- K\Big((1-|y_j|^2)(|y_j| - |\hat{y}_j|\cos(\theta_j - \hat{\theta}_j))|x_i|\sin\theta_{ji}^{\mathrm{rot}}$$
$$+ |y_j||\hat{y}_j|\sin(\theta_j - \hat{\theta}_j)\frac{|x_i|}{|u_j|}\cos\theta_{ji}^{\mathrm{rot}}\Big) \tag{4.87}$$

where K is learning time constant. We can also obtain a carrier-frequency-dependent dynamics, as we did for the complex-valued Hebbian rule. The amplitude equation is identical with (4.86), whereas the time delay τ_{ji} is related to the phase θ_{ji} as $\theta_{ji} = 2\pi f \tau_{ji}$, i.e.,

$$\begin{aligned}\tau_{ji}(t+1) = \tau_{ji}(t) \\ - \frac{K}{2\pi f}\Big((1-|y_j|^2)(|y_j| - |\hat{y}_j|\cos(\theta_j - \hat{\theta}_j))|x_i|\sin\theta_{ji}^{\mathrm{rot}} \\ + |y_j||\hat{y}_j|\sin(\theta_j - \hat{\theta}_j)\frac{|x_i|}{|u_j|}\cos\theta_{ji}^{\mathrm{rot}}\Big)\end{aligned} \tag{4.88}$$

The rotation angle $\theta_{ji}^{\mathrm{rot}}$ in (4.78) is given as

$$\theta_{ji}^{\mathrm{rot}} \equiv \theta_j - \theta_i - 2\pi f \tau_{ji} \tag{4.89}$$

4.4.6 Function Approximation by Use of Complex-Valued Hebbian Rule

In a phasor neural network whose output amplitude is always unity, the meaningful signal is only the phase. In this case, we can utilize the complex-valued supervised Hebbian rule (complex-valued correlation learning) described in Section 4.3.7 to realize function approximation of the phase value. Though the method is rougher than the steepest descent method, it works well in many applications.

The complex-valued Hebbian rule in (4.36) and (4.37) can be rewritten for the present function-approximation problem as

$$\tau\frac{d|w_{ji}|}{dt} = -|w_{ji}| + |\hat{y}_j||x_i|\cos(\hat{\theta}_j - \theta_i - \theta_{ji}) \tag{4.90}$$

$$\tau\frac{d\theta_{ji}}{dt} = \frac{|\hat{y}_j||x_i|}{|w_{ji}|}\sin(\hat{\theta}_j - \theta_i - \theta_{ji}) \tag{4.91}$$

where $x_i \equiv |x_i|e^{i\theta_i}$ is input signal, $w_{ji} \equiv |w_{ji}|e^{i\theta_{ji}}$ is weight, and $\hat{y}_j \equiv |\hat{y}_j|e^{i\hat{\theta}_j}$ is output teacher. Equation (4.91) means that there exists an ideally stable state of phase at

$$\hat{\theta}_j = \theta_{ji} + \theta_i \tag{4.92}$$

Namely, the input $|x_i|e^{i\theta_i}$ is multiplied by the weight $|w_{ji}|e^{i\theta_{ji}}$, and directed to the direction (phase) of the teacher signal $|\hat{y}_j|e^{i\hat{\theta}_j}$. This rotation occurs for all i. After appropriate learning in phase, all the weighted inputs $w_{ji}x_i$ point to the teacher's direction $\hat{\theta}_j$. The summation also points to the same direction. Even if the learning does not converge, each $w_{ji}x_i$ will be expected to point to approximately the same direction as $\hat{\theta}_j$. In this sense, the phase value $\hat{\theta}_j$ can be learned properly according to (4.91).

The above learning method is applicable, not in a few cases, even to function approximations with general complex-valued neural networks where the signal amplitude is variable, when we can control the amplitude appropriately by some means. We have several ways to control the amplitude. Among others, if all the signals point to a certain direction in the complex plane, we will be able to use the same learning method as that in the real-valued neural networks. In other words, we may construct a roughly effective complex-valued function-approximation dynamics by substituting the real-valued steepest descent learning expressed by (4.56) for the amplitude equation of the complex-valued Hebbian rule.

$$\tau \frac{dw_{ji}}{dt} = -(|y_j| - |\hat{y}_j|)(1 - |y_j|^2)|x_i| \tag{4.93}$$

$$\tau \frac{d\theta_{ji}}{dt} = \frac{|\hat{y}_j||x_i|}{|w_{ji}|} \sin(\hat{\theta}_j - \theta_i - \theta_{ji}) \tag{4.94}$$

This method sometimes increases local minima in learning because it is not a genuine steepest descent method. It is, instead, a very rough method to deal with the amplitude and phase in completely different manners, though each equation is reasonable. When we have many inharmonious teacher signals, the phase values may not converge and, in such a case, the learning in amplitude will stop at a pseudo solution. The success depends on the nature of the problem to be solved. Practically, this method functions in many cases where the amplitude and phase are related to each other only weakly, and when the teacher signals have less inharmoniousness.

The discrete-time expression is given as

$$w_{ji}(t+1) = w_{ji}(t) - K(|y_j| - |\hat{y}_j|)(1 - |y_j|^2)|x_i| \tag{4.95}$$

$$\theta_{ji}(t+1) = \theta_{ji}(t) + K\frac{|\hat{y}_j||x_i|}{|w_{ji}|} \sin(\hat{\theta}_j - \theta_i - \theta_{ji}) \tag{4.96}$$

In addition, the continuous- and discrete-time expressions of delay-time learning in coherent neural networks is obtained with the relation of $\theta_{ji} = 2\pi f \tau_{ji}$ as follows.

$$\tau \frac{d\tau_{ji}}{dt} = \frac{1}{2\pi f}\frac{|\hat{y}_j||x_i|}{|w_{ji}|} \sin(\hat{\theta}_j - \theta_i - \theta_{ji}) \tag{4.97}$$

$$\tau_{ji}(t+1) = \tau_{ji}(t) + \frac{K}{2\pi f}\frac{|\hat{y}_j||x_i|}{|w_{ji}|} \sin(\hat{\theta}_j - \theta_i - 2\pi f \tau_{ji}) \tag{4.98}$$

4.4.7 Backpropagation Learning by Backward Propagation of Teacher Signals instead of Errors

Figure 4.9 is a schematic illustration of a multiple-layered lightwave coherent neural network where multiple modules similar to the network in Fig.4.8 are

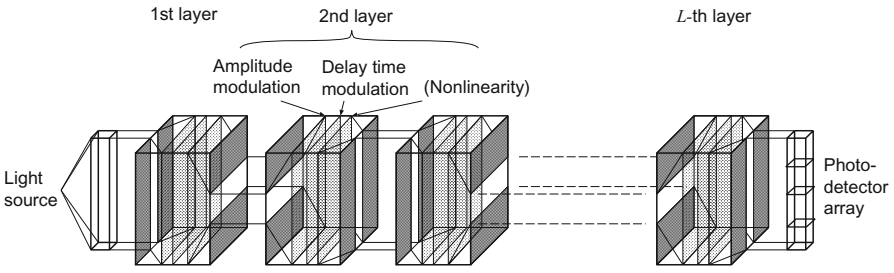


Fig. 4.9 Schematic illustration of multiple-layered coherent lightwave neural network.

stacked. We can obtain error-backpropagation learning dynamics by calculating the steepest descent operations over two adjacent modules as we did in the real-valued backpropagation section. However, the calculation and the obtained dynamics are a little more complicated than those for real-valued ones.

Moreover, in the error-backpropagation learning in such a network, we have a problem in the physical picture. That is to say, in a system based on lightwave or electromagnetic wave, it is difficult to make "error" wave propagate (backward). It is unnatural. (Even in a real-valued neural-network case, it seems also unnatural that "error" voltage, for example, backpropagates in the system.) Though we may mathematically be able to express the propagation of a difference of amplitude or phase (or real part or imaginary part), we cannot obtain a clear physical picture representing such a phenomenon. If such propagation of "error" is hard to illustrate, the dynamics should require additional global information-transmission connections, resulting in the failure in distributed learning with local interactions expected in neural networks.

On the other hand, the meaning of the complex-valued Hebbian rule (or correlation learning) is quite clear. The Hebbian rule accumulates the correlation between input and output signals, where we take the complex conjugate of the output signals in the complex-valued neural network. The complex conjugate means reversal of time, i.e., reversal of propagation direction just like what we would see if a movie film ran in reverse. In this sense, we can say that the complex-valued Hebbian rule corresponds to a physical picture in which forward-propagating processing waves interfere with backward-propagating teacher waves to produce standing waves. We observe the amplitude and phase at learning points to obtain information needed for the learning. This viewpoint is insightful, and is directly applicable to the use of holography in optical neural networks. [191], [192].

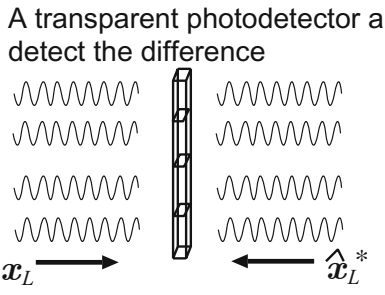


Fig. 4.10 Detection of the difference, $|\boldsymbol{x}_L - \hat{\boldsymbol{x}}_L^*|^2$, by detecting two counter-propagating lightwaves with a square-law detector.

In this context, the teacher-signal backpropagation learning was proposed [193],[83],[81]. In this learning method, the phase-conjugate waves of teacher signals propagate from the output layer (the L-th layer) backward through the layers toward the input terminals to deliver teacher information to neurons in hidden layers. The neurons perform the learning locally by using the delivered information sensible for themselves. In each layer, neurons employ the steepest descent method, or they may adopt the easy method based on complex-valued Hebbian rule mentioned in the previous section. Please refer to literature for details cited above. In this section, we present the underlying basic idea briefly.

Assume that a set of output teacher signals $\hat{\boldsymbol{x}}_L$ are presented at the output (L-th) layer. We want to reduce the difference between temporary outputs $\boldsymbol{x}_L$ and $\hat{\boldsymbol{x}}_L$. If the system is based on lightwave, and if the output lightbeams $\boldsymbol{x}_L$ propagate outward through the output layer, it is natural that the teacher signal lightbeams are also given as outward propagating lightwave. In Fig.4.9, the lightbeams propagate rightward.

The error function was defined as

$$\begin{aligned} E &= \frac{1}{2}\,|\boldsymbol{x}_L - \hat{\boldsymbol{x}}_L|^2 \\ &= \frac{1}{2}\sum_k \Big(|x_{kL}|^2 + |\hat{x}_{kL}|^2 - 2|x_{kL}||\hat{x}_{kL}|\cos(\theta_{kL} - \hat{\theta}_{kL})\Big) \end{aligned} \tag{4.99}$$

where suffix k is given to each element in $\boldsymbol{x}_L$ and $\hat{\boldsymbol{x}}_L$. Equation (4.99) means the detection of the difference between the signals carried by two lightbeams (one of them should be conjugate) by superimposing and detecting them with square-law detectors. We can also consider an alternative as follows. It means that we detect standing waves generated by backward-propagating conjugate waves $\hat{\boldsymbol{x}}_L^*$ and forward-propagating output waves $\boldsymbol{x}_L$ at the output layer with thin and transparent square-low detectors as illustrated in Fig.4.10.

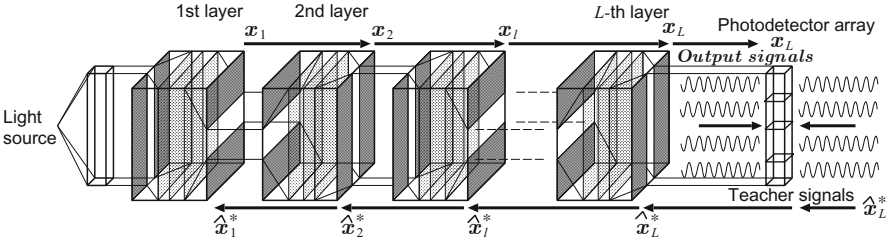


Fig. 4.11 Teacher signals $\hat{\boldsymbol{x}}_L^*$ propagating backward in a layered neural network.

Then we reach the following pictures. We make the teacher signals at the output layer $\hat{\boldsymbol{x}}_L$ backpropagate through the layered network as $\hat{\boldsymbol{x}}_{l-1} \equiv \mathbf{W}^{-1} f^{-1}(\hat{\boldsymbol{x}}_l)$, or more simply as shown in Fig. 4.11, $\hat{\boldsymbol{x}}_{l-1}^* = f(\hat{\boldsymbol{x}}_l^* \mathbf{V}_l)$ where $v_{jil} \equiv w_{jil}/|w_{jil}|^2$. A more simpler one is the backward propagation, using the optical configuration as it is, as $\hat{\boldsymbol{x}}_{l-1}^* = f(\hat{\boldsymbol{x}}_l^* \mathbf{W}_l)$. We detect the errors with thin and transparent detector arrays placed at output neurons in each layer. Even with such simplified dynamics, the learning can be conducted successfully. The success is attributed mainly to the fact that the meaning of the layering of the network lies in the reconstruction of topology (as it was in the perceptron), not in the details of metric, and that it is insignificant to manipulate the metric finely in hidden layers. In this way, the backpropagation learning is able to possess a meaningful physical picture if we make the teacher signals per se, instead of the errors, propagate backward penetrating the layers.

4.5 Adaptive Clustering and Visualization of Multidimensional Information

4.5.1 Function: Vector Quantization and Visualization

Self-organizing map (SOM) is used for adaptive clustering (vector quantization) and visualization of high-dimensional information by adaptively projecting signals onto two- or low-dimensional space [33] [34] .

For example, a SOM conducts the following adaptive clustering. For an input vector, the SOM examines which class cluster the input is nearest to, and classifies the input into the nearest class. To determine the distance between the input and a cluster, the SOM measures the distance, with some metric, between the input vector and the reference vector that represents the cluster of the class we focus on. A reference vector is prepared for each class assumed in the SOM. At the same time, the SOM modifies the position of the reference vector, into which the input is classified, slightly in the direction of the input vector.

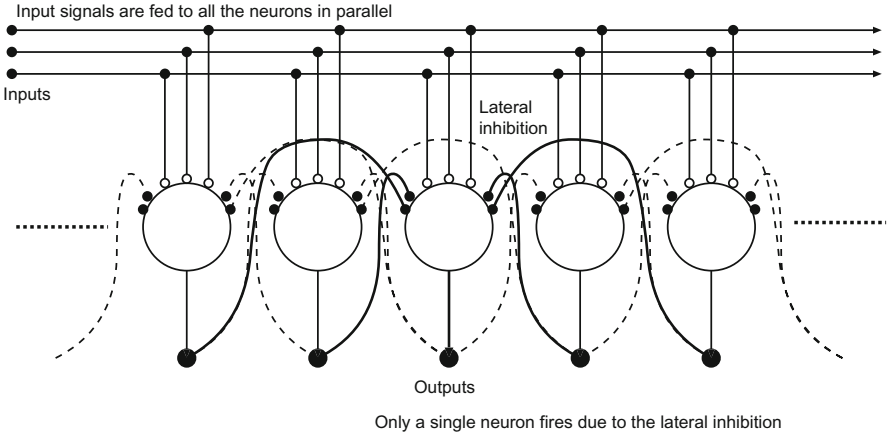


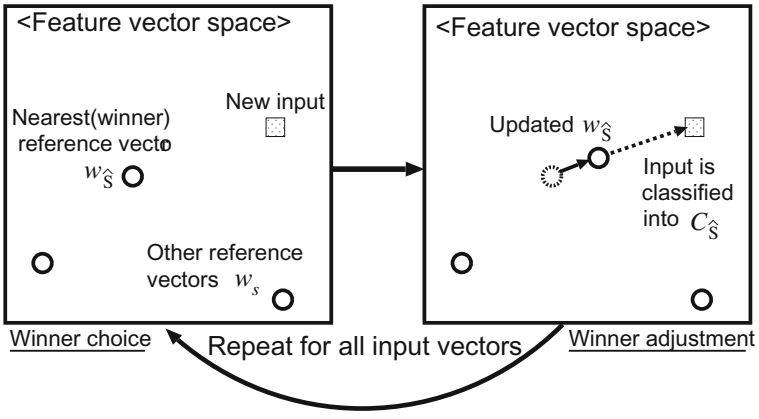
Fig. 4.12 Construction of the neural network functioning as a self-organizing map (SOM). Closed input circles are inhibitory connections, while open ones are ordinary excitatory connections.

By repeating the treatment mentioned above, the SOM finally arranges the positions of the reference vectors in the information space in such a way that their positions reflect the distribution of the input vectors fed sequentially to the SOM. That is, the SOM places many reference vectors at regions of high densities of input vectors to realize a fine classification, while it does only a small number of reference vectors at low input-vector-density regions for rough classification. By placing the reference vectors according to the input distribution, the SOM realizes an adaptive classification.

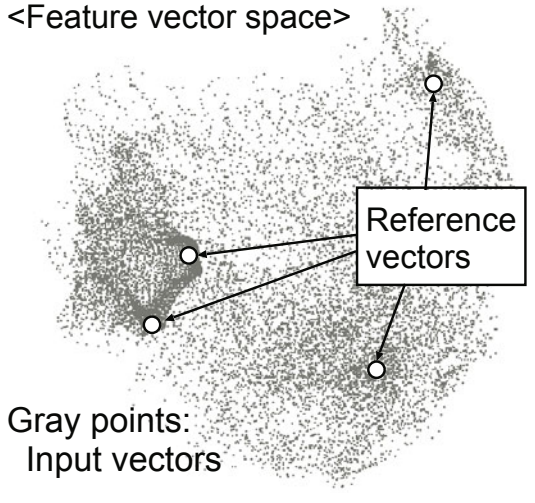
4.5.2 Network Construction, Processing, and Self-organization

Figure 4.12 shows the construction of the SOM neural network. Let us consider that each neuron represents a class. The neuron has a set of connection weights (weight vector), which are identical with the elements of the reference vector of the class, of which the neuron is in charge. When an input vector is fed to the network, it is delivered to all the neurons. Accordingly, the neuron that has the weight vector nearest to the input vector is most likely to fire.

We also assume that every neuron has strongly inhibitory output connections (negative connections that suppress firing) to all neurons other than itself. As a result, once one of the neurons fires, the output propagates to all other neurons as an inhibitory signal to block firing. This mechanism is called lateral inhibition, and is often found in the brain. Because of the lateral inhibition, only a single neuron, if any, fires at once. This process is named the winner-take-all (WTA) process.



(a)



(b)

Fig. 4.13 Schematic diagram of the classification in SOM: (a) Input vectors are fed to the SOM one after another. Each vector is classified into the nearest class, and the reference vector of the class is updated. (b) After sufficient iterations, all the reference vectors are placed appropriately reflecting the input-vector density, resulting in adaptive classification of the input vectors.

With the above mentioned mechanism, a single neuron having the weight vector nearest to the input vector fires. The class represented by the firing neuron is the class into which the input vector is classified. The weight vector of the firing neuron is updated.

The SOM performs the update of the weight vector, which is identical with the reference vector representing the class of the neuron, as follows. We consider a case in which the number of classes (neurons) is fixed. The class

number is determined according to the nature of application and characteristics of input vectors. It may also be increased or decreased adaptively.

First, we distribute the reference vectors (connection-weight vectors) in the information space at random. Then we present input vectors to the SOM one after another. As shown in Fig.4.13(a), the SOM finds the reference vector nearest to the input, and classifies the input into the class represented by the reference. Simultaneously, the SOM makes the reference vector slightly approach the input vector. By repeating this treatment, the SOM arranges more reference vectors in the regions of higher input-vector density, resulting in an adaptive classification of input vectors as shown in Fig.4.13(b).

The procedure of the input-vector classification and the reference-vector update is summarized as follows.

Adaptive input-vector classification and reference-vector update in SOM:

1. Initialization of reference vectors:
   Initialize reference vectors $\boldsymbol{w}_s(t=0)$ (s=1,2,3,...) at random.
2. Reference-vector update and input-vector classification:
   Repeat a) and b) described below for every input vector $\boldsymbol{x}$.
   a) Find the winner class $C_{\hat{s}}$ to which $\boldsymbol{x}$ should belong, and label $\boldsymbol{x}$ as a member of class $C_{\hat{s}}$.

$$\boldsymbol{x} \in C_{\hat{s}} \quad if \quad \|\boldsymbol{x} - \boldsymbol{w}_{\hat{s}}(t)\| = \min_s \{\|\boldsymbol{x} - \boldsymbol{w}_s(t)\|\} \tag{4.100}$$

   where $C_{\hat{s}}$ denotes class, $\|.\|$ means some distance such as Euclidean distance.
   b) Update the winner reference vector.

$$\boldsymbol{w}_s(t+1) = \begin{cases} \boldsymbol{w}_s(t) + \alpha[\boldsymbol{x} - \boldsymbol{w}_s(t)] & , C_s = C_{\hat{s}}(t) \\ \boldsymbol{w}_s(t) & , C_s \neq C_{\hat{s}}(t) \end{cases} \tag{4.101}$$

   where α ($0 < \alpha < 1$) is self-organization parameter.

This is the simplest SOM dynamics called the k-mean algorithm. We often use a variable α that is large at the beginning and gets smaller as the self-organization proceeds.

On the other hand, for the purpose of visualization of multidimensional information, we prepare $\boldsymbol{w}_s$ two-dimensionally on a grid, and update not only the winner but also its neighbors weakly. This situation is analogous to the movement of points on a thin rubber. That is, the movement of the winner causes additional movement of neighbors on the elastic film. This is more general dynamics of SOM. In this case, we have a neural-network picture in which a neuron communicates with neighbors.

As shown in the above explanation, a SOM is a self-organizing neural network, i.e., it changes itself according to input signals. The self-organization proceeds along with task processing, e.g., the adaptive classification in the

above case. If we regard the self-organization as a learning, it is unsupervised learning.

Recently, a general SOM dynamics to realize an adaptive classification of migrating signals was proposed, and results of theoretical analysis and experiments were reported [59]. In the paper, they dealt with the adaptive demodulation problem in terms of communication signals in the complex plane affected by noise, nonlinearity, and Doppler effect causing migration of the signals. The method is applicable not only to complex-valued signal processing but also to applications in wide areas. This type of SOM is called predictive SOM (P-SOM) because the P-SOM predicts the future signal points in the SOM operation. Since phase-shift phenomena have an endless signal movement in the space with circular topology, the P-SOM is useful in particular in the complex-valued signal processing related to phase information.

4.5.3 Complex-Valued Self-organizing Map: CSOM

There exists the complex-valued self-organizing map (CSOM). To determine the distance between input vectors and reference ones, the CSOM employs a metric suitable for complex numbers. The most basic metric to measure the distance is to evaluate the absolute value of the difference of the two vectors, i.e., the scalar product of the difference vector and its conjugate transpose. In the case, where the absolute values of vectors are not so significant, we may use the inner product, i.e., the scalar product of one vector and the conjugate transpose of the other one (normalized by the absolute values of the two vectors). For example, the larger the real part of the inner product, the nearer we consider the two vectors.

These treatments are very simple in which we assume complex-valued space without warp. However, it is sufficient to include the nature of complex arithmetic such as periodicity in phase. Application examples are presented in Chapter 5 (adaptive classification of landsurface) and in Chapter 6 (adaptive radar system to visualize antipersonnel plastic landmines).

4.6 Markov Random Field Estimation

4.6.1 Function: Signal Estimation from Neighbors

Natural data such as elevation maps expressing earth landscape are well modeled upon the Markov random field (MRF) model. That is to say, we expect that the probability $P(x_s)$ that a variable defined at a position s in space and time takes a value of x_s is determined by the values in the vicinity. As an example, we consider an image having a size of $M \times N$ pixels, in which N_s denotes the vicinity including neighbors of a point s. The probability function $P(x_s)$ is expressed by the values of the neighbors as

$$P\Big(x_s|x_1, x_2, \cdots, x_{s-1}, x_{s+1}, \cdots, x_{M\times N}\Big) = P\Big(x_s|x_t; t \in N_s\Big) \qquad (4.102)$$

In the natural world, most variables have some relationship with its neighbors more or less, and changes continuously in space and time. The situation is compatible with the basic idea of neural networks. We can expect that neural networks play important roles in the problems related to the MRF model.

If the system under consideration is linear in total, the basis of processing with the MRF model is the same as that of so-called adaptive filters. However, even in such a case, "neural networks" will convey a more suitable meaning of the processing, rather than "filters," as the degree of parallelism increases.

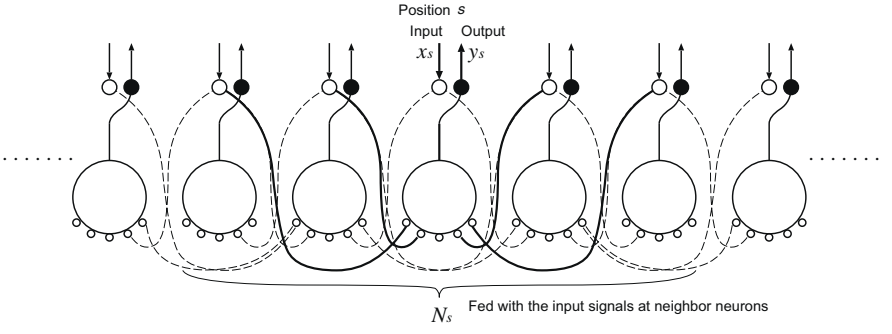


Fig. 4.14 Construction of a MRF-estimating neural network.

4.6.2 Network Construction and Processing Dynamics

Figure 4.14 shows the network construction. Though the illustration shows one-dimensional network for simplicity in display, the construction can be n-dimensional in general according to the data treated. For instance, a network dealing with two-dimensional data such as landscape maps, as mentioned in Chapter 7 (noise elimination in interferogram), possesses two-dimensional structure.

Let us consider an image. A neuron is assigned to every pixel, and each pixel has an input and an output terminals. As an example of processing, assume that we want to fill a defective pixel at position s. The neuron at s gathers input signals x_t at neighbors $t \in N_s$ to generate an estimate of the value as an output y_s. In the case that a noisy pixel value x_s is temporarily given to the pixel, the network can also compare the value with an estimated one y_s and calculate an energy to evaluate the appropriateness of the temporary input pixel value. When we do not consider any nonlinearity, a probability density $P(x_s)$ that the pixel has a value of x_s can be constructed with a set of connection weights $\boldsymbol{w} \equiv [w_{st}]$ as

$$P(x_s) = \frac{1}{Z} e^{-E(x_s)} \tag{4.103}$$

$$E(x_s) = \frac{1}{2\sigma^2} \left| x_s - \sum_{t \in N_s} w_{st}^{\mathrm{T}} x_t \right|^2 \tag{4.104}$$

where Z is partition function (coefficient to normalize the probability), σ^2 is the variance of pixel values, and E means energy. In the equations, w_{st} and σ^2 are called MRF parameters.

In a complex-valued neural network, the relationship (4.103) and (4.104) are expressed for a complex pixel value z_s as [194]

$$P(z_s) = \frac{1}{Z} e^{-E(z_s)} \tag{4.105}$$

$$E(z_s) = \frac{1}{2\sigma^2} \left| z_s - \sum_{t \in N_s} w_{st}^{*} z_t \right|^2 \tag{4.106}$$

Applications of the statistics and the processing are presented in Chapter 5 (clustering of landsurface), Chapter 6 (visualization of plastic landmines), and Chapter 7 (generation of digital elevation maps by eliminating noise).

4.6.3 Learning Correlations between Signals at a Pixel and Its Neighbors

A neural connection learns the correlation between two signals by following the Hebbian rule. For given values at pixels s and t, the neuron accumulates the correlation between the values at s and t in the connection weight w_{st}. In an image, or in a part of image, modeled upon the MRF, the correlation value is one of the MRF parameters reflecting a statistical feature of (the part of) the image. When a series of images are presented sequentially to the network, and if the statistical feature is unchanged at the two pixels, a correlation is gradually stored in the weight. An estimate of the MRF parameter is obtained as $\hat{\boldsymbol{w}}$.

Alternatively, if the statistical feature is uniform over a local area in the image, a neuron will obtain an estimate of the correlation between pixels s ant t in $\hat{\boldsymbol{w}} = [\hat{w}_{st}]$ by scanning the positions of s and t in the local area with a fixed relative location. Note that, in practice, we often replace the simple correlation matrix resulting in the Hebbian laerning by an expression using the generalized inverse matrix mentioned in Section 4.3.4 (associative memory). We obtain

$$\hat{\boldsymbol{w}}^* \equiv \left[\sum_{s \in L \times L} z_s \boldsymbol{q}_s^* \right] \left[\sum_{s \in L \times L} \boldsymbol{q}_s^* \boldsymbol{q}_s \right]^{-1} \tag{4.107}$$

$$\hat{\sigma}^2 \equiv \frac{1}{L^2} \sum_{s \in L \times L} |z_s - \hat{\boldsymbol{w}}^* \boldsymbol{q}_s|^2 \tag{4.108}$$

$$\boldsymbol{q}_s \equiv (\text{vertical vector consisting of } s\text{-neighbor pixel values}) \tag{4.109}$$

where $L \times L$ is the number of the pixels in the local area.

In the application to generate a digital elevation map in Chapter 7, we deal with radar interferogram obtained in landsurface observation. We restore a pixel value, which we consider is strongly affected by noise (resulting in so-called phase singular points), by estimating a noiseless value using its neighbor pixels. Then we remove the singular points, and obtain a high-quality digital elevation map with a low calculation cost.

4.7 Principal Component Analysis

4.7.1 Function: Extraction of Principal Information in Statistical Data

Principal component analysis (PCA) is a method to analyze multidimensional statistical data by reducing the data dimension with only little loss of information. In the PCA, we reconstruct input signals by generating their linear combinations, and present the first principal signal component (largest-energy component), secondly principal signal component (secondly-largest-energy component), and so on.

The PCA is widely used in multivariate analyses, and is often implemented in numerical-analysis software. In ordinary cases, we obtain all data at the beginning, and construct correlation coefficient matrices and variance matrices to conduct the PCA. In sum, we do it in a batch. On the other hand, in a PCA neural network presented in this section, the network self-organizes gradually in an on-the-job manner to become a PCA network, though the principle of the PCA processing is identical. Because of this adaptability, PCA neural networks can follow input signals even when their statistical property changes slowly.

The PCA is used extensively in processing waves such as speech and electromagnetic wave. Therefore, we naturally deal with complex-valued data. That is, complex-valued neural networks play an important role also in PCA.

4.7.2 Network Construction and Dynamics of Task Processing and Self-organization

Figure 4.15 shows the network construction for a complex-valued PCA. Since the construction is very simple, we skip the detailed description on real-valued networks. In a real-valued one, we substitute transpose $(\cdot)^{\mathrm{T}}$ for conjugate transpose $(\cdot)^*$ in a complex-valued network described below.

An input signal vector $\boldsymbol{x}$ is fed to the first neuron. We do not need a nonlinear activation function. The input is weighted and summed as

$$y_1 = \boldsymbol{w}_1^* \boldsymbol{x} \tag{4.110}$$

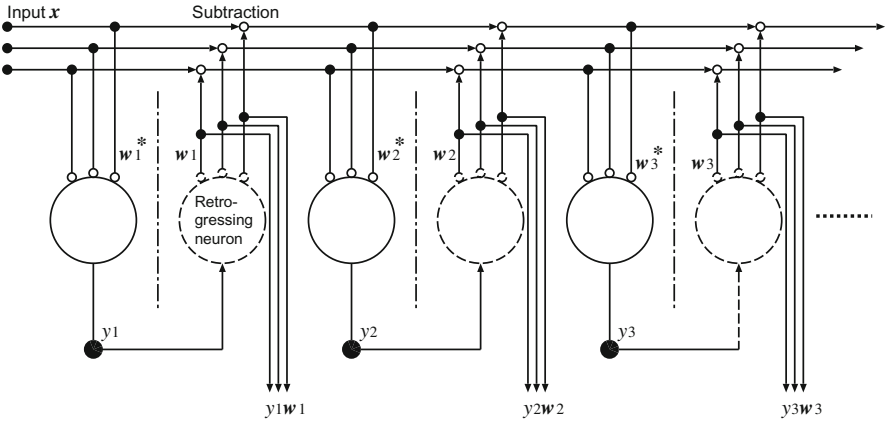


Fig. 4.15 Construction of a neural network for principal component analysis (PCA).

At the same time, the weight $\boldsymbol{w}_1$ self-organizes by following the complex-valued Hebbian rule as

$$\boldsymbol{w}_1 \longleftarrow \boldsymbol{w}_1 + K y_1^* \boldsymbol{x} \tag{4.111}$$

where K is a constant determining the speed of self-organization.

The input signals are, for example, sound waveforms obtained by N microphones placed against multiple sound sources. The analog waveforms are A/D converted and transformed into complex-valued data with the help of Hilbert transform described in Section 3.6.1 (preprocessing for baseband signals). The signals form time-sequential vectors having a length of N. Alternatively, just like bats, we may transmit ultrasonic waves all around, and detect reflections with N sensors by homodyning and A/D conversion to obtain time-sequential complex-valued N-long signal vectors.

We consider a set of zero-mean stationary-statistic signals, or signals having statistic characteristics migrating sufficiently slowly. As we present the signal vectors to the neural network time-sequentially, the weight $\boldsymbol{w}_1$ is gradually directed to the principal component direction of the series of the signals $\boldsymbol{x}$ in N-dimensional information space. The principal component is a linear combination of the signal-vector elements whose direction in the scatter diagram has the largest variance, i.e., the largest power. This is because the signals have the largest amplitude in this direction, and the weight $\boldsymbol{w}_1$ is dragged into the direction of $y_1^* \boldsymbol{x}$, namely, the direction of $\boldsymbol{x}$ since y_1^* is scalar. The variation causes a positive feedback and enhances the influence of the component in this direction.

However, the simple Hebbian rule (4.111) makes the weight $\boldsymbol{w}_1$ increase infinitely. To avoid this divergence, we modify the dynamics as

$$\begin{aligned} \boldsymbol{w}_1 &\longleftarrow \boldsymbol{w}_1 + Ky_1^*\boldsymbol{x} - Ky_1^*y_1\boldsymbol{w}_1 \\ &= \boldsymbol{w}_1 + Ky_1^*(\boldsymbol{x} - y_1\boldsymbol{w}_1) \end{aligned} \tag{4.112}$$

Then, $\boldsymbol{w}_1$ is directed to the principal-component direction without the divergence. Simultaneously, every momentary $y_1\boldsymbol{w}_1$ for $y_1 \equiv \boldsymbol{w}_1^*\boldsymbol{x}$ represents the principal component included in $\boldsymbol{x}$. Therefore, we can "hear" the principal-component signal by D/A-converting $\boldsymbol{w}_1y_1$. Thus the principal component is extracted by the on-the-job self-organization on the bases of the Hebbian rule.

Next, we extract the secondary-, third-, etc., principal components by using so-called the deflation method. As shown in Fig.4.15, we subtract the first principal component from the input $\boldsymbol{x}$, and we apply the identical extracting process to the residual signals. The second neuron will extract the largest component in the residue, namely, the secondary-principal component. We repeat the process to extract n-th principal components sequentially.

The final representation of the dynamics including the deflation method is obtained as

$$\boldsymbol{w}_j \longleftarrow \boldsymbol{w}_j + Ky_j^* \left(\boldsymbol{x} - y_j\boldsymbol{w}_j - \sum_{i<j} y_i\boldsymbol{w}_i \right) \tag{4.113}$$

In this way, we can extract some principal components from time-sequential sound signals $\boldsymbol{x}$, and we can hear $y_1\boldsymbol{w}_1$, $y_2\boldsymbol{w}_2$, $y_3\boldsymbol{w}_3$, and so on, separately from one another.

The complex-valued PCA was reported, for example, in application to an adaptive sonar (sound navigation and ranging) system [121] to separate sounds obtained in the sea. Mixture of signals often occurs in relation with propagations of waves such as sound and electromagnetic wave. In such wave-related fields, the signal treatment in complex domain possesses inevitably great advantage. The complex-valued neural networks function effectively and fruitfully.

4.8 Independent Component Analysis

Independent component analysis (ICA) is another field in which the complex-valued neural networks are expected to work effectively, because we frequently deal with waves just like we do in PCA. The processing function and the network construction of the ICA are similar to those of PCA, respectively. However, in ICA, we extract independent signal components by evaluating the independence in the extracted components. In practice, we pay attention to the fact that, if sequences of signals $\boldsymbol{y} = [y_j]$ are independent of one another, signal sequences $f(\boldsymbol{y})$, derived from $\boldsymbol{y}$ with some nonlinear function $f(\cdot)$, should also be independent of one another. Accordingly, in a matrix representation of neural weights, we update the weights as, for example,

$\mathbf{W} \longleftarrow \mathbf{W} + K(\mathbf{I} - f(\boldsymbol{y})\boldsymbol{y}^*)\mathbf{W}$, where $\mathbf{I}$ is the identity matrix and K is self-organization constant.

Detailed descriptions of the ICA is given in literature, for example, [195] and [196]. As mentioned there, in the extraction of independent components, we have two approaches to represent a mixing matrix, i.e., representation in the time domain and that in the frequency domain. When we adopt a frequency-domain representation, we deal with complex spectra obtained as short-time Fourier transform of time-domain signals. Thereby, at least, we need a complex-valued neural network.

Sawada et al. [122] reported that, when we conduct frequency-domain blind separation (separation of mixed signals without advance knowledge) of speech, polar-coordinate-type (amplitude-phase-type) nonlinear functions result in better separation than rectangular-coordinate (real-imaginary-type) nonlinear functions do. This is because there is no special meaningful reference to measure the spectrum phase (setting of real and imaginary axes) since the window in the short-time Fourier transform cannot be synchronized with the speech signals in any sense. This condition holds in most cases when we deal with waves. Principal attributes of waves lie in amplitude and phase. Contrarily, real and imaginary parts cannot be essential entities because they depend on a phase reference introduced in observation.

Part II

Applications: How Wide Are the Application Fields?

5

Land-Surface Classification with Unevenness and Reflectance Taken into Consideration

In this chapter, we describe an adaptive system to classify land surface by taking unevenness and reflectance into consideration. We deal with interferograms on the basis of the complex-valued Markov random field (CMRF) model in statistics. We generate an adaptively segmented map in terms of the complex-valued texture of land-surface reflection by using the complex-valued self-organizing map (CSOM) that processes CMRF-based feature vectors.

5.1 Interferometric Radar

Figure 5.1 illustrates airborne or satellite radar observation to detect reflection from the earth's surface. By employing phase-sensitive electronics, we can obtain not only the amplitude but also the phase of the reflected electromagnetic wave. As a result, we acquire an image having complex-valued pixel values. These types of radars are called interferometric radars. We can detect the phase value of a signal by mixing the signal wave with a reference wave to observe their interference. For simplicity, in Fig.5.1, we show a system in which we transmit one of the waves, capture the reflection, and mix it with the other wave to obtain the phase[1].

Roughly speaking, the amplitude represents the reflectance since it gives the power of the reflected wave. On the other hand, the phase represents the distance. That is, when the reflecting object approaches, the phase is advanced since the number of waves existing in the propagation path between the antenna and the object is reduced. Contrarily, when the object recedes,

[1] In actual airborne or satellite interferometric radars, we prepare two antennas or navigation routes, and we look aside to obtain the phase difference of the two electromagnetic waves having slightly different off-nadia angles (angle between vertical line and the radio beam). Therefore, the wavelength of the transmitted wave does not correspond to 2π in the phase value in the phase image. However, we consider the system shown in Fig.5.1 for simplicity.

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 103–111.
springerlink.com

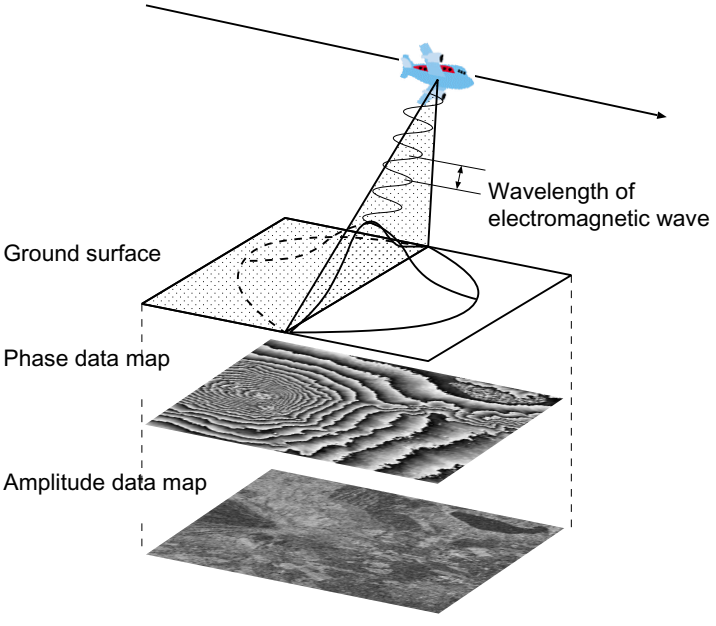


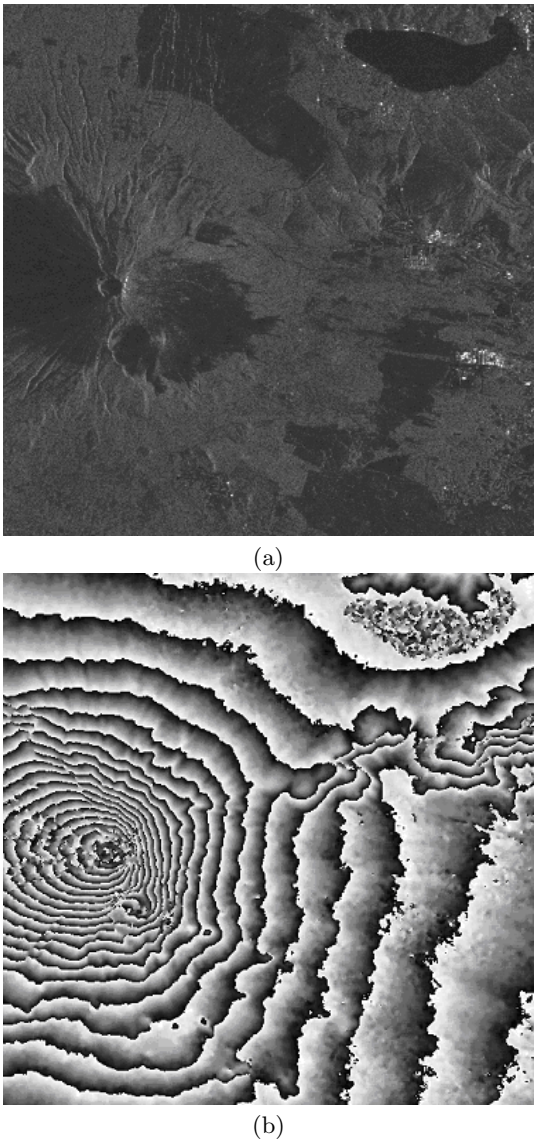
Fig. 5.1 Acquisition of land-surface information with an interferometric radar system.

the phase is retarded. Therefore, the phase represents the distance to the object, though the phase is expressed as modulo 2π. Then the phase fluctuation corresponds to surface unevenness. Slopes and fine fluctuation also change the reflection amplitude because it changes the reflection direction, or causes scattering.

In this chapter, we present a neural system that generates highly useful land-surface classification maps[197],[198]. It evaluates the texture in complex-valued reflection images that conveys reflectance and unevenness information. With this system, we can extract not only forests, deserts, lakes, and other regions having specific reflectance, but also mountain areas, ridges, spurs, rock fields, and so on, reflecting characteristic unevenness. The system is based on the complex-valued self-organizing map (CSOM) described in Section 4.5.

5.2 CMRF Model

Figure 5.2 shows an example of images obtained by an interferometric synthetic-aperture radar (InSAR) observing at around Mount Fuji. Figure 5.2(a) shows the amplitude, while Fig.5.2(b) gives the phase in modulo 2π, both in gray scale. (These original data were provided by courtesy of Dr. Masanobu Shimada of NASDA, which is presently JAXA.)



(a)

(b)

Fig. 5.2 (a)Amplitude and (b)phase of land-surface reflection obtained by an InSAR system. Reprinted from Fig.3 in [197]: Andriyan Bayu Suksmono and Akira Hirose, Adaptive complex-amplitude texture classifier that deals with both height and reflectance for interferometric SAR images, IEICE Trans. on Electron., E83-C (12):1912–1916, 2000, with permission from IEICE.

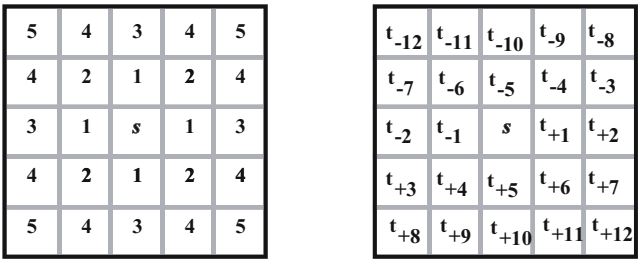


Fig. 5.3 Pixel at position s to which we pay attention and its neighbors in the vicinity N_s where the neurons are (a)labeled by distance from s and (b)labeled sequentially as t_i.

Conventional adaptive segmentation systems have utilized the texture in amplitude only. In this chapter, we use the phase information as well to generate more useful segmentation maps.

We evaluate local complex-valued texture quantitatively, but as simply as possible, to classify local areas and segment the image. We consider the complex-valued Markov random field (CMRF) model. In the present process, we introduce a noncausal CMRF model, i.e., unlike the time-sequential one, having no cause and result directions. Such a model is usually suitable for images.

We deal with complex-valued images based on the noncausal CMRF model as follows. Figure 5.3 shows the assignment of pixels. The value of the pixel at position s is $z_s \in \boldsymbol{C}$. Since an observed actual image is a part of nature, we consider that it has the Markovianity. That is to say, the pixel value z_s has some relationship statistically with the values of neighbors. The nearest pixels labeled as "1" must have a strong relationship, while far pixels labeled as "4" or "5" will have a weaker one. Moreover, when the statistical characteristic is uniform in a certain area, we can expect almost identical relationship statistically even if we pay attention to another pixel in the area.

Figure 5.3(b) shows a vicinity of the pixel at position s, N_s, with local neighbors labeled as t_i. Let us consider the probability distribution $P(z_s)$ that the pixel s has a value z_s in statistics. Then we have

$$P(z_s|\text{ values of pixels in the image except for } s\ ) = P(z_s|z_{t_i} \in N_s) \quad (5.1)$$

In other words, the probability distribution is determined by the neighbors, and is unchanged within the area having a uniform statistics. The probability distribution represents a set of features of the area in the image. This is the basic idea of the CMRF.

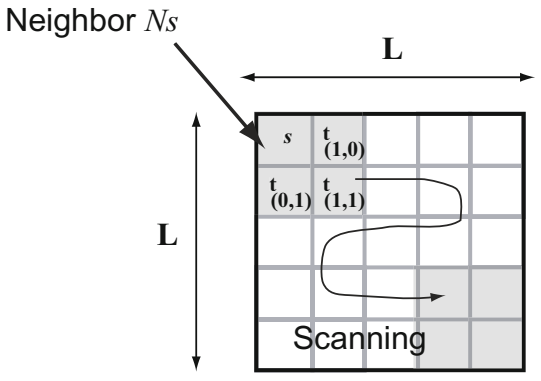


Fig. 5.4 Neighbors of point s, $t_{+1} \equiv t_{(1,0)}$, $t_{+5} \equiv t_{(0,1)}$, and $t_{+6} \equiv t_{(1,1)}$, sweep a local area having a size of $L \times L$ to gather the statistical features.

5.3 CMRF Model and Complex-Valued Hebbian learning Rule

Incidentally, (5.1) can be interpreted as follows. For statistically uniform data, the correlation between a pixel value z_s and a neighbor's values z_t is unchanged. Therefore, as mentioned in Section 4.6 (MRF estimation), we can obtain correlations between pixels by assigning a neuron to each pixel, and by making the neural connections learn the correlations with the complex-valued Hebbian rule as $< z_s(z_t)^* >$.

We assign neurons to the pixels one to one. The neurons are connected with the neighbors, and each neural connection learns the correlation between the input signals as $< z_s(z_t)^* >$. As shown in Fig.5.3, let us consider a small neighbor consisting of three pixels, $t_{+1} \equiv t_{(1,0)}$, $t_{+5} \equiv t_{(0,1)}$, and $t_{+6} \equiv t_{(1,1)}$. When the neurons *see* various, but statistically identical, images, the correlations, $< z_s(z_{t_{(1,0)}})^* >$, $< z_s(z_{t_{(0,1)}})^* >$, and $< z_s(z_{t_{(1,1)}})^* >$, will converge at certain values, respectively.

Alternatively, we can replace the temporal accumulation by a spatial one. Let us consider a local area having a size of $L \times L$ in which the statistics is uniform. Neurons in the area communicate to one another to accumulate statistical characteristics spatially. We may have a picture that the point s and the neighbors $t_{(0,1)}$, $t_{(1,0)}$, and $t_{(1,1)}$ sweep the local area with their relative location fixed. Then the neural connections memorize the following correlations

$$K(\xi, \eta) = \frac{1}{L^2} \sum_{i'=0}^{L-1} \sum_{j'=0}^{L-1} (z(i', j'))^* \ z(i' - \xi, j' - \eta) \tag{5.2}$$

where $(\xi, \eta) = \{(1,0), (0,1), (1,1)\}$.

We construct the network to gather the statistical features in the local area as described above. Other neurons, that see another area having another statistics, will memorize the correlations specific to that area. Then we can segment the image based on the correlation values accumulated in the neural connections. In the present case, the pixel values are represented by complex numbers. Therefore, the correlations reflect both the changes in reflectance (included mainly in amplitude) and the unevenness (in phase), i.e., complex texture in total. We conduct the segmentation based on the complex texture.

5.4 Construction of CSOM Image Classification System

We classify local areas into classes based on statistics adaptively by using the complex-valued self-organizing map (CSOM) mentioned in Section 4.5. Resultantly, we segment the land-surface by taking the reflectance and unevenness into consideration. In general, a SOM is widely used in adaptive vector quantization. In the present system, we expect that the CSOM also quantizes the texture-based features and segments the image into classes adaptively.

Figure 5.5 shows the construction of the CSOM-based radar system to segment a complex-valued image into classes adaptively by paying attention to the complex texture [197]. The expected function is the segmentation of landscape into, for example, Mount Fuji and Lake Yamanaka as if we had phase-sensitive eyes as mentioned in Chapter 1.

We place a local window block having a size of $L \times L$, and scan the image with this block. First, we unwrap the phase values $z(i,j)$ in the block (see Chapter 7) in a simple way. Then we obtain statistical properties such as mean M and covariance $K(\xi,\eta)$ to construct a feature vector $\boldsymbol{K}$ to be fed to the CSOM as

$$\boldsymbol{K} \equiv [M, K(0,0), K(0,1), K(1,0), K(1,1)] \tag{5.3}$$

$$M = \frac{1}{L^2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} z(i,j) \tag{5.4}$$

$$K(\xi,\eta) = \frac{1}{L^2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} (z(i,j))^* \ z(i+\xi, j+\eta) \tag{5.5}$$

where $(\cdot)^*$ means complex conjugate. Consequently, the covariance $K(\xi,\eta)$ includes phase differences, reflecting the height variation, unevenness, and their texture. Therefore, the system can estimate whether an amplitude change is caused by a change in reflectance, or by a change in unevenness.

Moreover, we can introduce some concepts such as “mountain” and “valley” as new indices for classification. In this case, the directions of slopes are

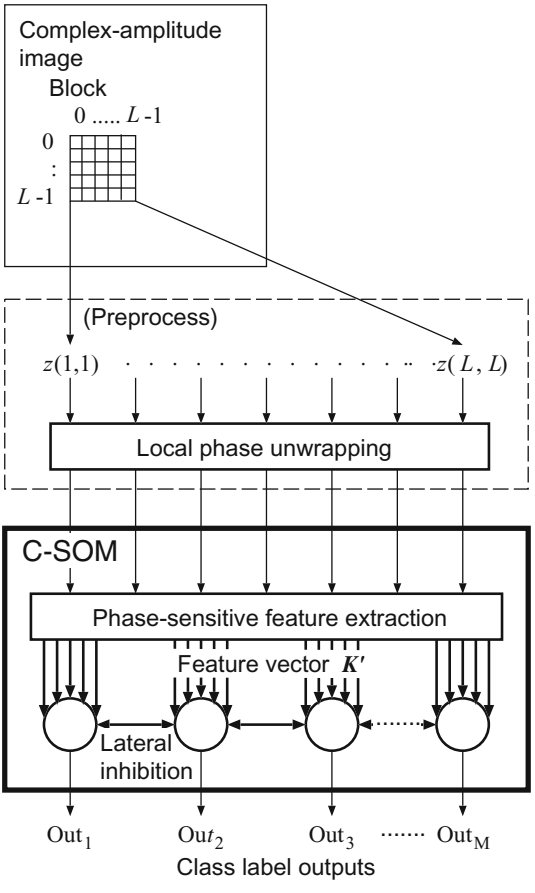


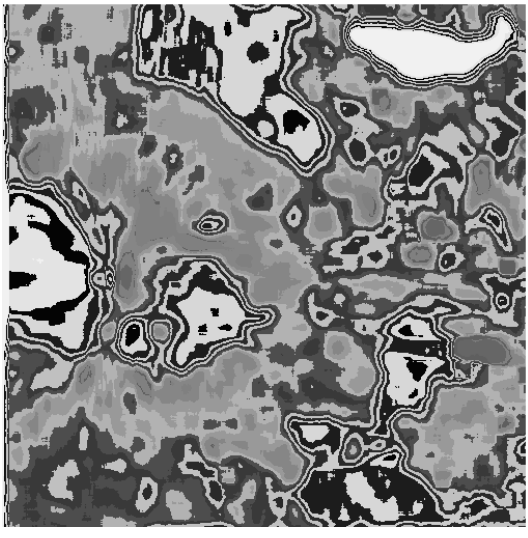
Fig. 5.5 Construction of the adaptive interferometric-radar-image segmentation system based on complex-valued self-organizing-map (CSOM). Reprinted from Fig.2 in [197] in the caption of Fig.5.2 with permission from IEICE.

not so important, or should rather be suppressed in the classification process to make a mountain area "mountain." For this reason, we slightly modify $\boldsymbol{K}$ into $\boldsymbol{K}'$ in such a way that the covariance is insensitive to the positive and negative of the phase differences as

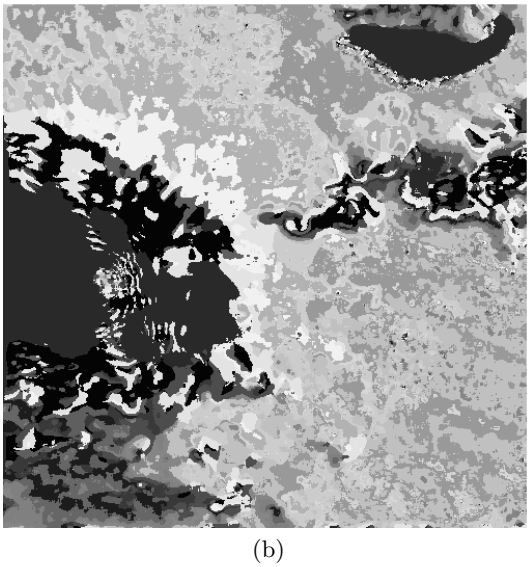
$$\boldsymbol{K}' \equiv [M, K(0,0), K'(0,1), K'(1,0), K'(1,1)] \tag{5.6}$$

$$K'(\xi,\eta) = |K(\xi,\eta)| \ e^{j|\varphi(\xi,\eta)|} \tag{5.7}$$

where we write $K \equiv |K(\xi,\eta)| \ e^{j\varphi(\xi,\eta)}$. We adopt such fine customization in the feature vector according to the purposes.



(a)



(b)

Fig. 5.6 Adaptive segmentation results for the InSAR land image generated by (a)a conventional SOM system and (b)the proposed CSOM system. Reprinted from Figs.5 and 6 in [197] in the caption of Fig.5.2 with permission from IEICE.

5.5 Generation of Land-Surface Classification Map

Figure 5.2 is a radar-image example obtained at an area around Mount Fuji and Lake Yamanaka, Japan. Figure 5.2(a) shows amplitude, while Fig.5.2(b)

shows phase, both in gray scale. We find that the reflection is very small at Lake Yamanaka and relatively small at forests and rocky areas. On the other hand, in the phase map, though the contours are apparently similar to the contours that should be found in an elevation map, the phase at the Lake is unnaturally turbulent because the low reflectance relatively emphasizes measurement noise.

Figure 5.6 shows the segmentation results (a)generated by a conventional (real-valued) SOM system for the amplitude image, and (b)generated by the proposed CSOM system for the complex-amplitude image. In 5.6(a), we find that Lake Yamanaka, forests, and rocky areas are segmented from others. On the other hand, in Fig.5.6(b), we find that the mass of Mount Fuji and the mountain ridge near Lake Yamanaka are also segmented additionally to those above, showing the fine folds of the skirt of Mount Fuji. In this way, we can generate a more useful adaptively segmented map by incorporating phase information into the segmentation with the concept of the phase-sensitive superbrain.

5.6 Summary

In this chapter, we described the usefulness of the complex-valued neural network to segment adaptively the land surface. This method is now going to be applied to wide areas such as inspections in factories. Complex-amplitude signals are of wide use in high-resolution ranging systems. They are also important in imaging permittivity distribution, e.g., with phase-contrast microscope. In such applications, the above-mentioned ideas are quite useful to realize adaptive processing.

6

Adaptive Radar System to Visualize Antipersonnel Plastic Landmines

We extend the idea in the adaptive classification described in Chapter 5 by introducing frequency-domain information to visualize plastic landmines buried shallowly underground. Antipersonnel landmines, in particular plastic ones, use so slight metal that it is difficult to detect them with metal detectors because many shots and metal fragments are scattered under battlefields. The shallowness also causes serious surface-reflection noise. We construct a phase-sensitive millimeter-wave / microwave front-end to observe ground reflection in spatial and frequency domains, and feed the data to a complex-valued self-organizing map (CSOM). The CSOM visualizes plastic landmines by segmenting the reflection image adaptively.

6.1 Ground Penetrating Radars

Ground penetrating radars (GPRs) are widely used in many fields such as buried-object detection, ruin explorations, and groundwater surveillance. In landmine detection, we also expect to apply them to nonmetal landmine detections, and many researches have been done in a long while. However, the detection of antipersonnel plastic landmines is mostly still difficult in practice because of the small target size, low reflectance, and relatively large land-surface reflection when they are buried shallowly underground. At the same time, insufficiency of demining professionals and high danger of operation augments the demand for support and automatization of the demining operation.

The complex-valued self-organizing map (CSOM) is highly effective in such a task. When a part of electromagnetic wave penetrates ground surface and landmine itself, we want to measure the range distribution (and texture) of reflectance. Then, by applying the CSOM to processing data obtained at multiple frequencies, we can expect a successful adaptive segmentation of three-dimensional data, i.e., in the propagation direction and the transversal two-dimensional space. Because the inverse Fourier transform of the

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 113–122.
springerlink.com

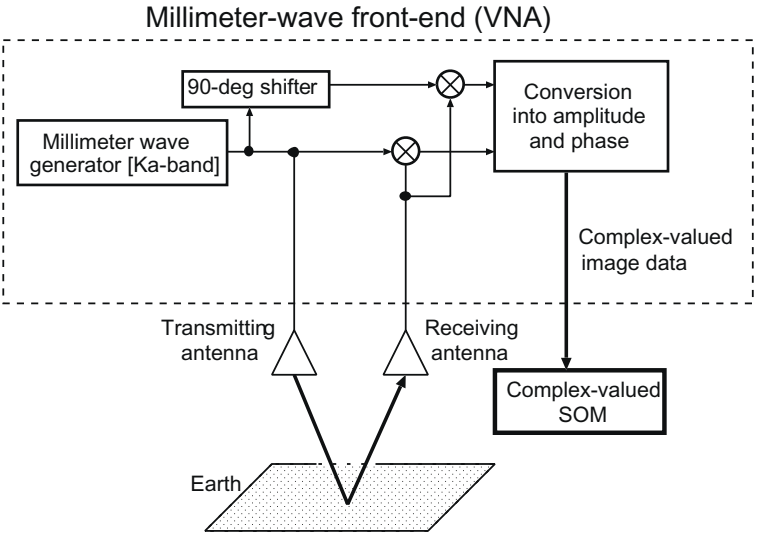


Fig. 6.1 Adaptive plastic-landmine visualization system consisting of millimeter-wave / microwave antennas, front-end, and CSOM processing unit [149][2].

frequency-domain data gives the time-domain reflection of an electromagnetic-wave pulse, the above processing is equivalent to classification of reflection texture in three dimensions.

In the system presented in this chapter, we observe the amplitude and phase of reflection at multiple frequencies to acquire complex-amplitude images. We visualize landmines adaptively by classifying the reflection texture in three dimensions, i.e., frequency + two-dimensional-space dimensions, by utilizing frequency-domain information in the CSOM [199] [149] . We also compare the result obtained by the CSOM with that by a conventional real-valued SOM dealing with amplitude texture only.

6.2 Construction of CSOM Plastic Landmine Visualization System Dealing with Frequency- and Space-Domain Texture

Figure 6.1 shows the schematic construction of the system focusing on the antennas and the front-end. A vector network analyzer (VNA) is used for

[2] Figs.6.1, 6.3, 6.4, 6.5 and Table 6.1 are reprinted from [149]: *Neural Networks*, vol.17, No.8–9, Takahiro Hara and Akira Hirose, "Plastic mine detecting radar system using complex-valued self-organizing map that deals with multiple-frequency interferometric images," pp.1201–1210, Copyright (2004), with permission from Elsevier. Japanese review [200] is also helpful for further understanding of the background and the technology.

homodyne detection of the received reflection. We obtain the amplitude and phase information simultaneously. The transmitter and receiver are rectangular horn antennas. The frequency range is 30−40GHz. The wider the frequency bandwidth is (which requires a higher center frequency), the higher resolution we can enjoy. However, a too high frequency is not suitable for GPRs because it will be so largely absorbed by the soil that we cannot see underground. We prepare a 70×70cm corrugated-cardboard box with a thin plastic sheet paved inside, and put ordinary Tokyo soil, containing stones and miscellaneous, and a target object in it. We move the pair of antennas facing to the soil horizontally, and obtain complex-valued data at multiple frequency points. The system is totally controlled by a personal computer (PC), and the obtained data is stored also in the PC, where the CSOM classifies pixels in the images by conducting the adaptive segmentation.

6.3 Adaptive Signal Processing in CSOM

As described in the previous chapter, first we prepare a small window block at around a pixel we focus on. In the block, we generate a feature vector reflecting the stochastic properties contained in the pixel values (texture). We sweep the whole image with the block, calculating a feature vector in the block at each position. Then we classify the obtained feature vectors adaptively in the CSOM in such a manner that the classification reflects the feature-vector distribution in the information space. Finally, we segment the image by labeling each pixel with the class into which the block belongs.

In this system, we observe reflection at multiple frequency points. The Fourier-transform operation reveals that the frequency-domain data contains information essentially identical to that of the time-domain data, i.e., the depth-direction reflection information. It is possible that, first, we Fourier transform the frequency-domain data inversely into time-domain one. In the present adaptive system, however, we deal with the frequency-domain data directly without the linear transform as follows.

6.3.1 Feature Vector Extraction by Paying Attention to Frequency Domain Information

Figure 6.2 shows the construction of the adaptive three-dimensional radar-image segmentation system consisting of two modules, namely, a complex-valued feature extractor and a CSOM in its narrow sense. We prepare a window block having a size of $L \times L$ on a set of input images measured at N_f frequency points. The feature extractor calculates a feature vector expressing stochastic properties of the pixel values in the block. As we did in the previous chapter, we adopt the complex-valued mean and covariances as the feature vector elements. We shift the block pixel by pixel, and calculate a local feature vector for the block at each position. Finally, the block finishes sweeping the

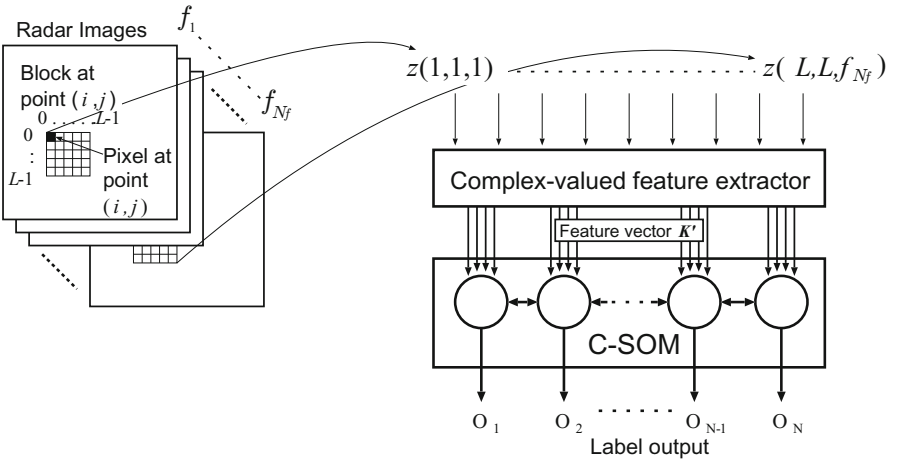


Fig. 6.2 Construction of the adaptive three-dimensional radar-image segmentation system [201][4].

whole image set. We feed the feature vectors sequentially to the CSOM, which determines adaptively which class each vector should belong to.

The procedure is described as follows. We consider the mean M and the covariances $K(\xi, \eta, f_\zeta)$ in each block as the elements of the feature vector $\boldsymbol{K}$. We determine the mean M and a covariance K in an $L \times L$ block for complex-valued pixel data $z(i, j, f)$ as

$$M = \frac{1}{L^2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} z(i, j, f_\mathrm{b}) \tag{6.1}$$

$$K(\xi, \eta, f_\zeta) = \frac{1}{L^2} \sum_{i=0}^{L-1} \sum_{j=0}^{L-1} z(i, j, f_\mathrm{b})\ z^*(i + \xi, j + \eta, f_\mathrm{b} + f_\zeta) \tag{6.2}$$

where f_b is a basis frequency determined arbitrary.

The total number of the covariances $K(\xi, \eta, f_\zeta)$ is enormous because of the possible combinations of the variables ξ, η, and f_ζ. To avoid this explosive expansion, we consider approximately that the spatial- and frequency-domain data are qualitatively orthogonal to each other. Then, the covariance elements have two parts, namely, the covariances in spatial domain $\boldsymbol{K}_s$ at a basis frequency f_b, and those in frequency domain $\boldsymbol{K}_f$. Finally, we take into account only the four spatial-domain values (mean M, variance $K(0, 0, 0)$

[4] Fig.6.2 is reprinted from [201]: *System and Human Science – For Safety, Security and Dependability* (T.Arai, S.Yamamoto, K.Makino (eds.)), Akira Hirose and Takahiro Hara, "Complex-valued self-organizing map: A framework of adaptive processing for multiple-frequency millimeter-wave interferometric imaging systems," pp.299–308, Copyright (2004), with permission from Elsevier.

(i.e., energy), covariances $K(0,1,0)$, $K(1,0,0)$, and $K(1,1,0)$) and a limited number of frequency-domain ones (covariances $K(0,0,f_\zeta)$). That is,

$$\boldsymbol{K} \equiv [\boldsymbol{K}_s, \boldsymbol{K}_f] \tag{6.3}$$

$$\boldsymbol{K}_s \equiv [M, K(0,0,0), K(0,1,0), K(1,0,0), K(1,1,0)] \tag{6.4}$$

$$\boldsymbol{K}_f \equiv [K(0,0,f_1), K(0,0,f_2), \cdots, K(0,0,f_{N_f})] \tag{6.5}$$

In (6.4) and (6.5), $K(0,0,0)$ is always a real number, while others are complex numbers.

With this feature vector, we expect qualitatively an adaptive segmentation of the image as follows. First, we consider the relationship in the pixel values in the frequency domain. If, at a local spot or a point, the soil including objects has some specific reflection at certain depths differently from its surrounding area, the frequency-domain-correlation elements are expected to express certain values specific to the frequency-domain texture at the point. The peculiarity assigns the feature vector $\boldsymbol{K}$ to a special location in the feature-vector information space. Then the points containing the identical feature are clustered in a single class.

Next, we consider the pixel-value relationship in the spatial domain. Remember the process described in Chapter 5 (clustering Mount Fuji, etc.). When we *see* the phase image, we regard the mountain as a mass. We deal with only the steepness of slopes without slope directions. Such treatment probably works well also in the present landmine visualization. Therefore, we adopt a direction-insensitive slope variables as we did in Chapter 5. When we write the spatial feature-vector elements $K(\xi,\eta,0) \in \boldsymbol{K}_s$ in polar coordinate as

$$K(\xi,\eta,0) = |K(\xi,\eta,0)|\, e^{j\varphi(\xi,\eta,0)} \tag{6.6}$$

the phase value φ in (6.6) represents the slope information. Therefore, we modify K_s to adopt a new one as

$$K'(\xi,\eta,0) = |K(\xi,\eta,0)|\, e^{j|\varphi(\xi,\eta,0)|} \tag{6.7}$$

By using the K'_s, we define a new feature vector $\boldsymbol{K}'$ as

$$\boldsymbol{K}' \equiv [\boldsymbol{K}'_s, \boldsymbol{K}_f] \tag{6.8}$$

We use $\boldsymbol{K}'$ as the feature vector to be fed into the CSOM.

6.3.2 Dynamics of CSOM Classification

The CSOM classifies the feature vectors according to the dynamics described in Chapter 4. In the present case, we assign a window block to every pixel position, and calculate the statistic features in the block to determine the

feature vector $\boldsymbol{K}'$. The number of the vectors fed to the CSOM is the pixel number in total (see Section 4.5). The CSOM classifies the vector determined for every pixel position into a class adaptively. By coloring the pixel according to the class into which the vector is classified, the set of the input images for an observation is segmented in the manner that reflects the result of the adaptive classification. Then we find spatial clusters in the segmented images.

In buried-object detection like the present case, the depth information plays a significantly important role. Though a pulse-radar system usually possesses depth information in the time-domain data, the multiple-frequency continuous-wave (CW) interferometric radar system has this information in the frequency-domain data. The Fourier transform reveals substantial equivalence between the frequency-domain data and the time-domain one. In our system, however, we do not employ the Fourier transform, which is a linear processing, but, instead, apply the nonlinear and adaptive CSOM processing directly to the frequency-domain feature-vector information. In this process, we expect the follows. We have a plastic object having a certain thickness at a certain depth. Then we observe specific signals such as frequency-dependent characteristic reflection and resonance. Such signals make the feature vector distribution inhomogeneous and, instead, specific to the object in the feature-vector space. The CSOM segments the distribution into a number of classes adaptively.

6.4 Visualization of Antipersonnel Plastic Landmines

6.4.1 Measurement Parameters

As shown in Fig.6.1, we place antennas facing to the land surface. We bury a mock plastic landmine called TYPE 72 whose diameter and height are 78mm and 40mm, respectively. It is filled with a substance having the same permittivity as that of explosive. Parameters in electromagnetic-wave observation and CSOM processing are shown in Table 6.1.

Table 6.1 Parameters in measurement and CSOM processing [149].

Start frequency	$f_{\min}$	30.0GHz
Stop frequency	$f_{\max}$	40.0GHz
Frequency point number	N_f	81
Frequency interval	Δf	125MHz
Scanning area	$X \times Y$	381.0mm × 381.0mm
Sampling point number	$N_x \times N_y$	128 × 128
Sampling interval	ΔX (ΔY)	3.0mm
Clustering class number	$s_{\max}$	16

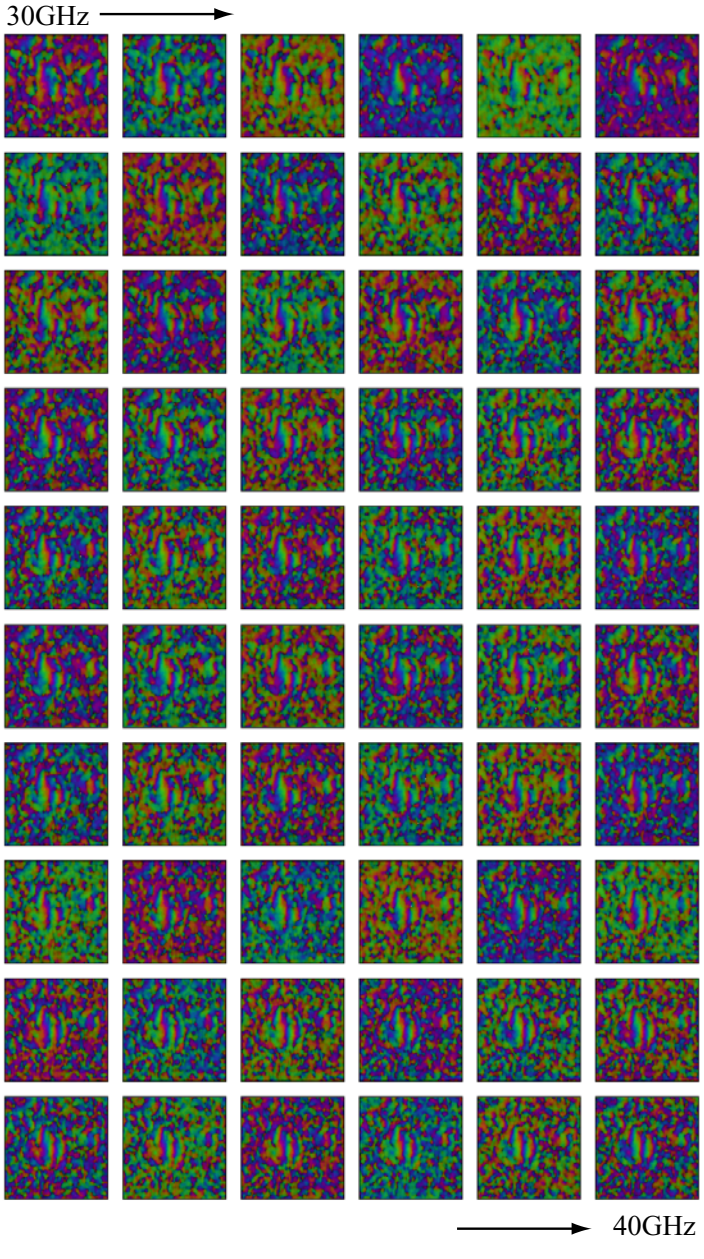


Fig. 6.3 Complex-amplitude images obtained in multiple-frequency observation of a plastic landmine buried shallowly underground where brightness shows intensity and hue shows phase [149].

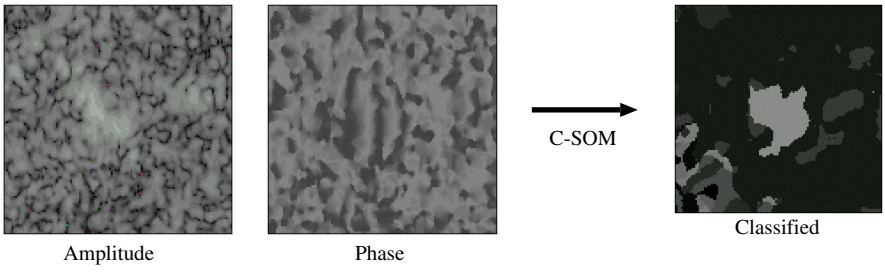


Fig. 6.4 Adaptive visualization result for a plastic landmine buried shallowly underground [149].

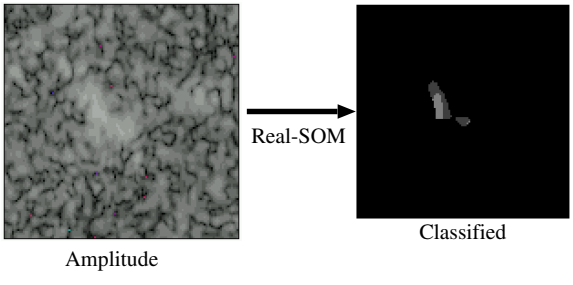


Fig. 6.5 Adaptive visualization result for a plastic landmine buried shallowly underground with a conventional (real-valued) SOM [149].

6.4.2 Results of Observation and Classification

First, we examine the raw data obtained in the observation. Figure 6.3 shows the phase images in grayscale observed at multiple frequency points for the mock plastic landmine buried shallowly underground at about 1cm depth. Black corresponds to $-\pi$ phase angle, while white means π. The frequency is 30GHz in the top-left image, whereas it is 40GHz in the bottom-right one, and the frequency is changed stepwise with a constant interval. When we see each image individually, we cannot find anything hidden. That is, even human brain cannot detect a landmine.

Actually, a round plastic landmine is buried at the center. When we know this fact, and when we examine all the images in total, not a few readers probably find that we can construct something round intuitively in our mind. With our volition, we can *see* something undetected when we watch a single image. However, we have still difficulty in *seeing* it.

If we analyze our mind, we may say the follows. First, the central area presents more-orderly changes in the space than the surrounding areas. The soil-and-stone areas have more random phase changes. In the frequency domain, we can also find more-orderly changes in the central area, though the

frequency-domain changes are little more difficult to notice. The role of the CSOM is to extract such differences in the texture to visualize the object.

In contrast, the amplitude images observed at multiple-frequency points are similar to each other. As an example, we show an amplitude image on the left-hand side in Fig.6.4 in grayscale. White means a high power, while black means a low one. In Fig.6.4, the next image is the phase image. We can find in the amplitude image that the power is a little high at the center. However, the shape is not round. Moreover, in multiple observations, we find similar high-power reflection even for a buried metal bolt. Therefore, it is difficult to find plastic landmines only with the amplitude observation.

The right-hand-side image in Fig.6.4 is the result of the CSOM segmentation. We clearly find something round at the center. The CSOM system visualizes the plastic landmine buried shallowly underground by performing an effective segmentation in the observed images successfully.

On the other hand, Fig.6.5 shows a result when we employ a conventional (real-valued) SOM for segmentation of the amplitude image. Though we can find something small at the center, the visualization quality is much lower than that of the CSOM in comparison with the result in Fig.6.4. We cannot say that the landmine area is segmented with the conventional SOM.

The comparison between the results of the real-valued SOM and the CSOM, we find that the round shape of the landmine has been brought about by the phase information. That is to say, the CSOM has successfully performed the segmentation that we could do, in Fig.6.3, when we examine the spatial and frequency-domain phase data in total, and when we know the correct answer. The CSOM has worked as a phase-sensitive superbrain.

6.4.3 Performance Evaluation by Visualization Rate

We repeated experiments to estimate the success rate in visualization. With our present CSOM system, the rate is about 70%, while it is about 10% for a real-valued SOM, which suggests the effectiveness of the CSOM processing. The value of 70% is comparable to the rate that a metal detector finds metal landmines. In this sense, at least, the present system has a sufficient ability in practical use.

6.5 Summary

In this chapter, we have presented a CSOM system to visualize plastic landmines buried shallowly underground by dealing with complex-amplitude images obtained at multiple-frequency points. By comparing the result with that of a real-valued SOM, we have discussed in what way the CSOM realizes a successful visualization.

It is urgent to detect and remove plastic landmines in the world. Needless to say, we have to develop useful techniques to realize efficient detections

of plastic landmines. Complex-valued neural networks provide the human beings with solutions in such problems.

In the first decade of 2000s, the system presented here has been modified and improved into a series of small portable visualization systems employing array antennas for quick acquisition of scattering / reflection of electromagnetic wave. They have been tested in the field of Cambodia, for example, for further improvement for practical use in the near future [151] [152] [154].

The neural processing in two-dimensional space $\times$ frequency-domain (or time-domain) data realizes an adaptive processing of three-dimensional spatial information. The importance of such three-dimensional adaptive processing will increase more and more in many fields related to millimeter wave and microwave systems such as intelligent transport systems (ITS) and multiple-input multiple-output (MIMO) systems where we use multiple antennas in transmission and detection in wireless communications.

7

Removal of Phase Singular Points to Create Digital Elevation Map

The phase map obtained in observation of Mount Fuji presented in Chapter 5 shows the reflection phase modulo 2π. Therefore, the fringe curves are contours showing the geography in the observation area. Given one knows the fact, one can imagine the landscape of Mount Fuji to some extent. Computers can perform a similar processing to yield a height map, which we call a digital elevation map (or digital elevation model: DEM). The process to unwrap the phase image wrapped within $-\pi$ to π is the *phase unwrapping*. The phase unwrapping is, however, known as a difficult process for conventional computers because of the existence of phase singular points. In this chapter, first we explain the singular points, which is a serious noise induced in the interferometric observation. Then we present a method to remove the singular points effectively by using a complex-valued neural network, and generate a high-quality DEM with a smaller calculation cost.

7.1 Phase Unwrapping

The phase unwrapping is explained as follows. As we discussed in Chapter 5, the phase map obtained in airborne / satellite observation using interferometric radars possesses information equivalent to the height of land surface. However, the phase values are *wrapped* into $-\pi$ to π. By unwrapping them using computers, we can obtain so-called digital elevation map (DEM). This process is the phase unwrapping.

However, the phase unwrapping is known as a difficult task in conventional computing. Figure 7.1 illustrates, a little more microscopically, the amplitude and phase observed in a similar observation [202]. In the phase image, we can find some points where the phase values have rotational components. That is to say, when we examine the phase value around the point, it increases in 2π unawares, or decreases in -2π. We call the point the *singular point* (SP). The main origin is interference noise. We can also find in the amplitude

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 123–131.
springerlink.com

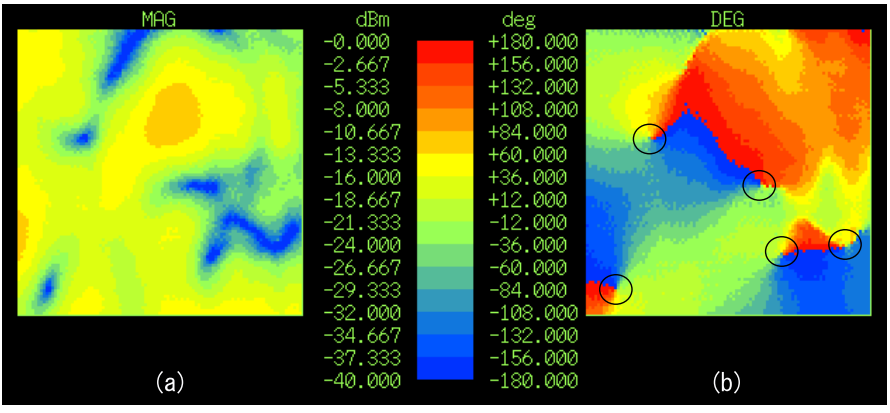


Fig. 7.1 Typical (a)amplitude and (b)phase images obtained by interferometric radars. In the phase image, we can find five phase singular points indicated by open circles. (Reprinted from Fig.3 in [202] with permission from Springer-Verlag.)

image that the points corresponding to the SPs have abrupt small values. The phenomenon is very natural since, in physical consideration, a point having indefinite phase value should not have any non-zero amplitude.

Because the phase value represents height of land surface, the phase field should intrinsically be conservative. Nonetheless, the field cannot be conservative actually. Then we cannot determine the height. For example, as we see later, the phase map shown in Fig.7.6(a) (Page 130) contains SPs of over 600.

However, we have a conventional method called the branch-cut method, so that we manage to determine, or more accurately, to estimate the height. In the branch-cut method, fundamentally we determine the height of a pixel by examining the phase shift between neighboring pixels, pixel by pixel, by incorporating 2π jump when we consider it appropriate. However, in this process, we place lines connecting SPs, and we determine the height without crossing the SP-connecting lines. In this way, regardless, we determine the elevation, though we have 2π cliffs on the SP-connecting lines because the line functions as a barrier of the spreading of the 2π inconsistency. Therefore, the total length of the lines should be as small as possible. When we connect all the SPs, the lines appear to be a branch in total. Hence the name branch-cut. The shorter the total length is, the better the DEM quality becomes. However, such a search problem with combinatorial explosion to find the shortest-line pairs is incompatible with ordinary computing. The computational cost grows seriously rapidly as the number of SPs increases.

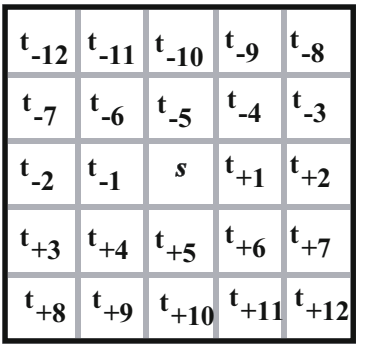


Fig. 7.2 Point s and its neighboring points t_i in the neighbor's area N_s.

7.2 Noise Reduction with a Complex-Valued Cellular Neural Network

Then, how can the human brain easily estimate the shape of Mount Fuji? Human beings first estimate the shape roughly and globally. Afterwards, if we find an inconsistent point in a precise examination of the phase map, we will presume a correct pixel value in a pattern processing by observing the surrounding pixels, and will remove the SP.

In this chapter, we present a neural network that performs a task such as the "phase-familiar brain" does. It is a superbrain named complex Markov-random-field (CMRF) estimating neural network (See Section 4.6) which is constructed as a complex-valued cellular neural network. A cellular neural network has neurons at lattice points, and a neuron has connections only locally. A two-dimensional cellular network is suitable for image processing and compatible with integration on a chip.

In phase unwrapping, we may use only phase data in general. However, the present complex-valued network deals with not only the phase but also the amplitude, as complex amplitude consistently, of the image data because, in many cases, they are related closely to each other. As we mentioned above, they are correlated strongly in particular at around an SP. We utilize the amplitude information effectively.

The network estimates a correct pixel value at a detected SP by using the values around the SP, and modifies the pixel value based on the estimate. This process reduces the number of SPs considerably, lightens the calculation cost required in making a DEM, and improves the DEM quality [194], [203], [198].

In the same manner as we did in Chapter 5, we label the pixels as shown in Fig.7.2. We adopt the CMRF model. Remember that, when we consider a probability $P(z_s)$ that the pixel s has a value of z_s

$$P(z_s| \text{ values of pixels in the image except for } s \ ) = P(z_s|z_{t_i} \in N_s) \quad (7.1)$$

where $t_i \in N_s$ is neighbors, the probability can be assumed constant in an area with a certain extension and, consequently, it represents a nature of the image statistically.

Accordingly, the value at the pixel s is estimated statistically by the neighbors. In the conventional real-valued MRF model, the probability P_s that the pixel s takes the value of x_s is estimated by the neighbor values x_t as

$$P(x_s) = \frac{1}{Z} e^{-E(x_s)} \tag{7.2}$$

$$E(x_s) = \frac{1}{2\sigma^2} \left(x_s - \sum_{t \in N_s} \Lambda_{st}^{\mathrm{T}} x_t \right)^2 \tag{7.3}$$

where Z is the partition function for normalization, Λ_{st} ($= w_{st}$ in Section 4.6) is correlations in a generalized-inverse-matrix expression, σ^2 is variance reflecting the degree of fluctuation, and $E(x_s)$ is energy. The parameters Λ_{st} and σ^2 are called the MRF parameters.

We extend the formula into a complex one to extract statistic features and to estimate pixel values of complex-valued images. We derive the complex-valued version for a complex pixel value z_s, by considering an analogy with the real-valued one given above and substituting conjugate transpose for simple transpose, as [194]

$$P(z_s) = \frac{1}{Z} e^{-E(z_s)} \tag{7.4}$$

$$E(z_s) = \frac{1}{2\sigma^2} \left| z_s - \sum_{t \in N_s} \Lambda_{st}^{*} z_t \right|^2 . \tag{7.5}$$

However, in practice, we have to estimate the CMRF parameters Λ_{st}^{*} based on the data in a local area. By considering averages within an $L \times L$ local observation area, we estimate the CMRF parameters as

$$\hat{\boldsymbol{\Lambda}}^{*} \equiv \left[\sum_{s \in L \times L} z_s \boldsymbol{q}_s^{*} \right] \left[\sum_{s \in L \times L} \boldsymbol{q}_s^{*} \boldsymbol{q}_s \right]^{-1} \tag{7.6}$$

$$\hat{\sigma}^2 \equiv \frac{1}{L^2} \sum_{s \in L \times L} \left| z_s - \hat{\boldsymbol{\Lambda}}^{*} \boldsymbol{q}_s \right|^2 \tag{7.7}$$

$$\boldsymbol{q}_s \equiv \begin{bmatrix} z_{t_{-12}} \\ z_{t_{-11}} \\ \vdots \\ z_{t_{+11}} \\ z_{t_{+12}} \end{bmatrix} \tag{7.8}$$

Note that, in the above expression, we omit the suffix st of $\hat{\boldsymbol{\Lambda}}^{*}$ since we estimate the parameters finally for arbitrary s by scanning the local area by shifting s and N_s.

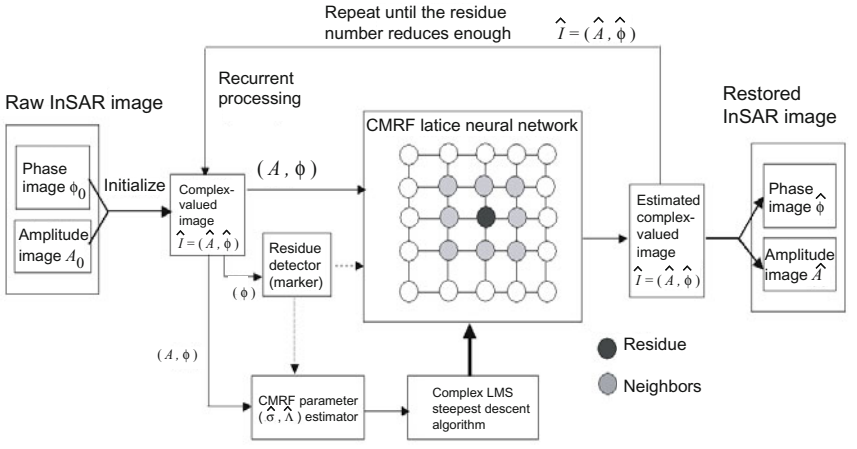


Fig. 7.3 Construction of the complex-valued cellular neural network to remove singular points. (Reprinted from Fig.4 in [194]: Andriyan Bayu Suksmono and Akira Hirose: Adaptive noise reduction of InSAR image based on complex-valued MRF model and its application to phase unwrapping problem, *IEEE Trans. on Geoscience and Remote Sensing,* 40(3):699–709 (followed by publisher's errata on Fig.12), 2002 ((C) 2002 *IEEE*) with permission from the IEEE.)

In (7.6), we obtain the estimate of z_s by using $\hat{\mathbf{\Lambda}}^*$ obtained as a generalized-inverse matrix described in Chapter 4. However, fundamentally we can estimate the parameters $\hat{\mathbf{\Lambda}}^*$ by employing the simple complex-valued Hebbian rule. That is to say, we can do it with the neural Hebbian (correlation) learning as $\hat{\mathbf{\Lambda}}^* = \sum z_s \boldsymbol{q}_s^*$, which has the same nature as the CMRF parameters expressed by (7.6). In other words, we can explain the process to estimate the CMRF parameters as a self-organization in the cellular neural network where a neuron is assigned to every pixel, and neural connections between neurons learn the correlation between the pixel values as $< z_s(z_t)^* >$. When the neighbor window N_s scans the local area in which the statistic property is unchanged, the network self-organizes to realize the estimation of the CMRF parameters similar to those expressed in (7.6).

7.3 System Construction

Figure 7.3 shows the system construction. A complex-valued reflection image $\boldsymbol{I}$ is input from the left-hand side. In the reflection observation, a homodyne circuit yields in-phase (real-part) and quadrature-phase (imaginary-part) components. Thereby, we regard the phase of the local oscillator output as the phase reference. If we use a vector network analyzer, we may obtain

directly the phase and amplitude of the reflection. In any case, by scanning the beam direction, we obtain a complex-valued reflection image $\boldsymbol{I}$.

We scan the image $\boldsymbol{I}$ and detect SPs. At the same time, we estimate the CMRF parameters $(\hat{\rho}^2, \hat{\Lambda})$ by using the data in the area containing no SPs. Then, as shown in the large rectangle at the center in Fig.7.3, we update the pixel value at the SPs. That is, the neuron at an SP connected locally to other neurons estimate the proper pixel value based on the CMRF parameters $(\hat{\rho}^2, \hat{\Lambda})$ and the surrounding pixel values. We modify the pixel value in such a manner that the energy expressed in (7.5) is reduced. We can employ a deterministic method [194], or a probabilistic hill-climbing one [203].

We update the pixel values at all the SPs to obtain a first estimate image $\hat{\boldsymbol{I}}$. We can expect that the number of the SPs in the estimate is less than that in the initial image. By repeating the above process, we generate a final estimate image $\hat{\boldsymbol{I}}$.

7.4 Dynamics of Singular-Point Reduction

Figure 7.4 shows the detail of the SP reduction. In Fig.7.4(a), we find observed amplitude and phase images with a corresponding SP map from the left-hand side to the right. The values of the amplitude and phase images are expressed in grayscale. There are two types of SPs, i.e., counterclockwise increasing SPs (positive SPs) and clockwise increasing SPs (negative SPs). They are

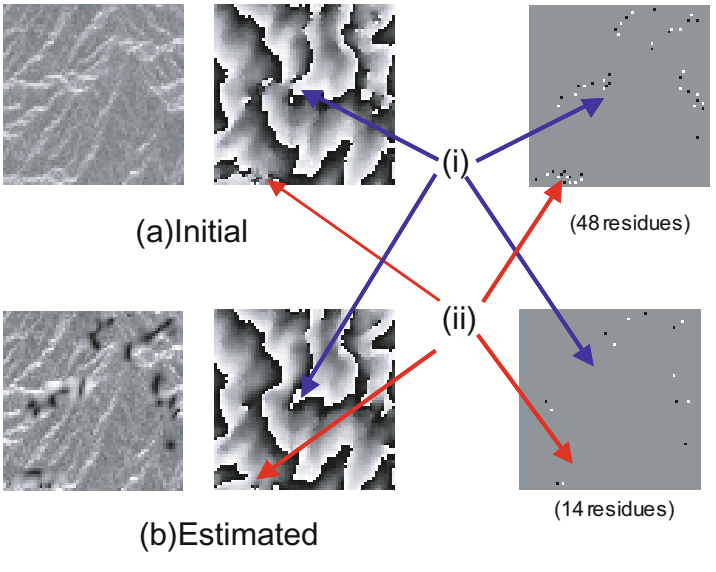


Fig. 7.4 Close-up of amplitude and phase images with a corresponding SP map from the left-hand side to the right. (Reprinted from Fig.6 in [194] in figure caption of Fig.7.3 ((C) 2002 *IEEE* with permission, IEEE.)

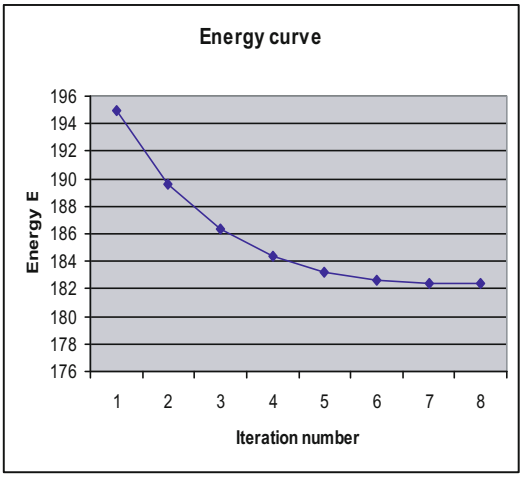


Fig. 7.5 Energy versus iteration number. (Reprinted from Fig.7 in [194] in figure caption of Fig.7.3 ((C) 2002 *IEEE* with permission, IEEE.)

expressed by white and black dots, respectively, in the SP map, where we have 48 SPs in this case in total. On the other hand, Fig.7.4(b) shows the processing result. We find that the number of the SPs is reduced to 14, and that the phase image has almost the same contours as those in the observed phase image in (a). In a close observation at the areas indicated by (i) and (ii), it is clear that the proposed neural network removes the SPs and, at the same time, the global elevation information is preserved. In the amplitude image in (b), the amplitude values around the SPs are modified to approximately zero in a wider area so that, as we considered in Section 7.1, the physical consistency is maintained explicitly around the SPs where the phase values cannot be determined uniquely.

Figure 7.5 presents the energy value defined in (7.5) versus the number of iteration of the neural processing. It decreases monotonically, which suggests a successful removal of the SPs.

7.5 DEM Quality and Calculation Cost

Figure 7.6 shows the results for an observation at the Mount Fuji area [194]. Figure 7.6(a) is the case where we apply the branch-cut method directly to the observed phase image. There we find the phase image, corresponding minimum-spanning-tree branch-cut, and obtained DEM. The DEM is expressed in color, where purple means undefined height because of closed branch-cut.

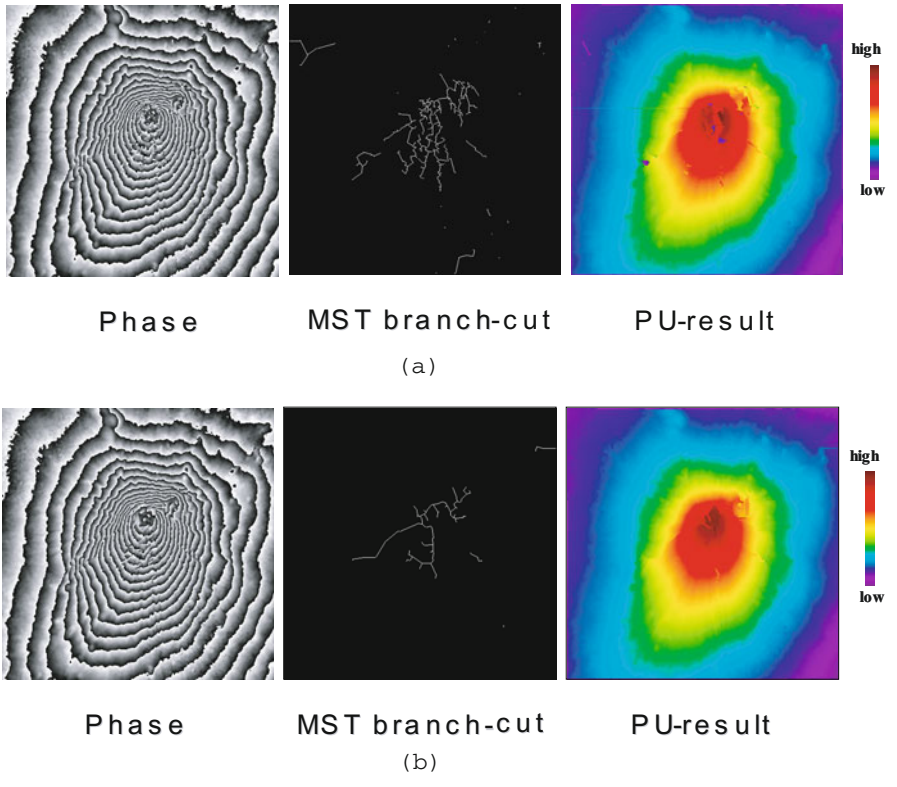


Fig. 7.6 Results for Mount Fuji observation: (a)Observed phase image, minimum-spanning-tree branch-cut, and resulting DEM obtained directly from the observed phase image without SP reduction, and (b)phase image after SP reduction, corresponding minimum-spanning-tree branch-cut, and resulting DEM obtained for the SP-reduced complex-amplitude image. (Reprinted from Figs.10, 12, 13, and 14 in [194] in figure caption of Fig.7.3 ((C) 2002 *IEEE* with permission, IEEE.)

On the other hand, Fig.7.6(b) shows the results when we conducted the SP reduction with the complex-valued cellular neural network. Though we cannot find difference between the phase images in (a) and (b) at a glance, the SP number in (a) is 611, while that in (b) has been reduced to just 100. That is to say, the neural processing realizes a great SP reduction without global landscape modification. Consequently, the total length of the branch-cut in (b) has become so short that the resulting DEM contains only small unnatural areas. The calculation time required for the branch-cut phase-unwrapping process has also been reduced to approximately 1/8.5.

7.6 Summary

In this chapter, we presented a complex-valued cellular neural network to estimate pixel values based on the complex-valued Markov random field (CMRF) model. At around singular points (SPs), the network estimates correct complex-amplitude reflection values, and reduces the number of the SPs. As a result, the calculation cost in the branch-cut phase unwrapping (PU) is drastically reduced. The quality of the obtained DEM is also greatly improved. The present estimation technique based on statistics is useful when the statistical property is considered almost uniform in a local area. In the Mount Fuji case, the statistics changes in the image in total. However, by preparing a local area smaller than the spatial changes of the statistic characteristics, we have been successful in applying the present method very effectively. Note that it is quite important that we deal with the phase together with the amplitude, i.e., the complex-amplitude, even if we are finally interested only in phase information.

The method presented here has advanced into an improvement in the works of Yamaki & Hirose [146] in the first years of 2000s, where they deal with the four pixels constructing a singular point simultaneously, and also lead to new unwrapping techniques such as the "progressive least-square unwrapping method" (Suksmono & Hirose) [144] and the so-called "singularity spreading phase unwrapping (SSPU) method" (Yamaki & Hirose) [145]. Their practical applicability has been evaluated by, e.g., Japan Aerospace Exploration Agency (JAXA) for the near-future use in the world.

8

Lightwave Associative Memory That Memorizes and Recalls Information Depending on Optical-Carrier Frequency

This chapter presents a lightwave neural network that learns behavior depending on optical frequency. In general, a neural network learns or self-organizes adaptively to environment. Because of this adaptability, we sometimes have difficulty in controlling the neural network at will. The lightwave network to be presented here learns a certain processing at a certain optical-carrier frequency, and another processing at another frequency, to possess both the adaptability and the controllability. From a different viewpoint, we can consider this compatibility as a mean of multiplexing of behavior in the optical frequency domain. The frequency-domain multiplexing provides the neural network, which is characterized by distributedness and parallelism, with a new dimension of massive parallelism utilizing the vast optical frequency bandwidth. In this sense, the network presents a novel direction in optical information processing hardware.

8.1 Utilization of Wide Frequency Bandwidth in Optical Neural Networks

The present state-of-the-art optical fiber communications involves frequency-domain multiplexing (FDM) in the optical frequency domain. The communications frequency (so-called 1.55μm-wavelength band) is first divided into L band (186.25‘190.95THz) and C band (191.60‘196.55THz). Then each band is further divided into channels of 100 or 200GHz bandwidth just like the channels in television and radio broadcasting systems. Each channel carries information independent of the information conveyed by other channels.

If we utilize the same technique in optical information processing, we can realize a massive parallelism in the vast optical frequency domain. Simultaneously, the fact that early-days' telephone-switching systems were equivalent to electronic computers suggests us that the realization of an optical FDM information processor should also be equivalent to the construction of an adaptive and intelligent all-optical FDM packet switch.

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 133–142.
springerlink.com

Parallelism and distributedness are the most specific characteristics in the construction and dynamics of neural networks. Optical circuits have plasticity and three dimensionality in connections. At these points, we can say that optical circuits possess high spatial parallelism. In particular in short-wavelength circuits, the spatial parallelism can be enhanced. Therefore, conventional optical neural networks emphasized the advantage in the spatial parallelism.

On the other hand, electronic neural hardware has also been investigated for a long time. Owing to high locality of electrons, electronic circuits have been greatly minimized and integrated, and are nowadays integrated even three dimensionally. Consequently, the spatial density of electronic hardware is generally much higher than that of optical hardware.

Ordinary optical circuits don't work well in space smaller than its wavelength. Moreover, optical circuits have the so-called telescope-effect problem. That is, lightwave generally requires such a long propagation length that an optical circuit cannot be densely integrated in this direction. To solve such problems, we need to go further into optical nanotechnology, for instance, techniques to practically utilize evanescent light.

However, the operation speed of electron devices is inevitably restricted by the electron charge. Contrarily, lightwave has an extremely higher operation speed (= wider bandwidth). The ultrawide bandwidth is a big advantage in optical circuits and should be explored much more. Additionally, conventional neural optical circuits have utilized only the power of lightwave ($\approx$ intensity and amplitude) among other parameters such as phase, frequency, and polarization. If we pay attention to frequency, we realize the vast optical frequency bandwidth mentioned above, which is a crucial point to be investigated in neural networks. To manipulate frequency precisely with high resolution, we have to deal with phase of lightwave. For this purpose, the complex-valued neural networks are essential and indispensable.

This chapter describes a coherent optical neural network that has carrier-frequency-dependent behavior by utilizing the wide optical bandwidth. Carrier frequency is the frequency of the pure sinusoidal wave without information modulation. When we use a semiconductor laser, we can easily modulate the carrier frequency by changing the injection current fed to the laser diode. This type of frequency modulation (FM) is called the direct frequency modulation. In this chapter, we construct a coherent lightwave associative memory that memorizes and recalls patterns dependent on the value of the optical-carrier frequency. Neural connection weights are determined by use of the complex-valued Hebbian rule expressed in a frequency-dependent form. Optical experiments demonstrate carrier-frequency-dependent memory and recall [204],[192].

The system to be presented here follows the preceding construction of a coherent lightwave associative memory that memorizes complex-valued vectors reported in Ref. [205]. Coherent neural networks inherently possess

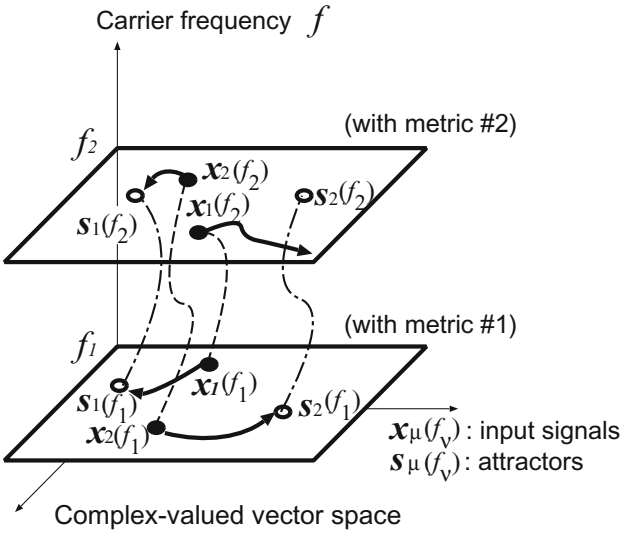


Fig. 8.1 Conceptual illustration of the carrier-frequency-dependent behavior in the associative memory system. (Reprinted from Fig.1 in [204]: Sotaro Kawata and Akira Hirose: Coherent lightwave associative memory system that posesesses a carrier-frequency-controlled behavior, *Opt. Eng.,* 42(9):2670–2675, 2003, with permission, SPIE.)

carrier-frequency-dependent behavior as mentioned in Section 4.3.8. The present system is based on this nature.

The first optical experiment of complex-valued associative memory was reported by Takeda & Kishigami in 1992 [82]. They constructed the epoch-making lightwave associative memory based on the mathematical analogy between the electromagnetic field in optical resonators formed with phase-conjugate mirrors and the dynamics of hermitian associative memories. The underlying idea is applicable to various situations where we deal with physical waves. However, the possible neural functions are limited within associative memories. On the other hand, the system to be presented here is not based on phase-conjugate physics, but using optical modulators having more flexible operation as an optical device. Therefore, in this sense, the bases described below have a higher potential to construct neural networks with a variety of functions.

Besides, we have several reports on the utilization of the optical-frequency domain so far, for instance, an FDM memory using volume hologram [206], a winner-take-all network based on the nonlinearity in semiconductor laser oscillation with external optical feedback and the spectral hole burning [207], and numerical experiments showing various generalization characteristics in the frequency domain realized by learning and self-organization [83].

8.2 Optical-Carrier-Frequency Dependent Associative Memory: The Dynamics

Figure 8.1 is an illustration presenting the concept of the optical-carrier-frequency-dependent behavior in the associative memory. The carrier frequency means the frequency of the lightwave that conveys information, and is identical to the oscillation frequency of the semiconductor laser used.

If the optical carrier frequency is f_1, the associative memory system has an information metric corresponding to f_1. When the memory is fed with an input vector $\boldsymbol{x}_1(f_1)$, it recalls a memorized vector $\boldsymbol{s}_1(f_1)$ that is nearest to the input in the metric system determined by f_1. For a different input vector $\boldsymbol{x}_2(f_1)$, it recalls another one, $\boldsymbol{s}_2(f_1)$, nearest to $\boldsymbol{x}_2(f_1)$. However, on condition that the system has a carrier frequency of f_2, the metric is changed, giving the memory a different worldview. That is, for instance, it recalls $\boldsymbol{s}_1(f_2)$ for an input $\boldsymbol{x}_2(f_2)$ since $\boldsymbol{x}_2(f_2)$ is near to $\boldsymbol{s}_1(f_2)$ in this metric. An input $\boldsymbol{x}_1(f_2)$ may be too far from any memorized vectors, and yields none of them.

The above recalling story is just a possible example. The recall behavior is determined by learning that is dependent on the carrier frequency. The detail is given below.

8.2.1 Recalling Process

Given the amplitude is fixed at unity without amplitude modulation, the neurodynamics to recall a memorized vector is expressed as follows. As mentioned in Chapter 4, the amplitude can be variable in general. However, in the present system, it is constant for simplicity.

$$\boldsymbol{x}(d+1) = A\left(|\mathbf{W}\boldsymbol{x}(d)|\right)\exp(i\arg\{\mathbf{W}\boldsymbol{x}(d)\}) \tag{8.1}$$

$$x_i = |x_i|\exp(i\alpha_i) \tag{8.2}$$

$$w_{ji} = |w_{ji}|\exp(i2\pi f\tau_{ji}) \tag{8.3}$$

where $A(\cdot)$ and $\mathbf{W} = [w_{ji}]$ are amplitude nonlinear function and connection weights, respectively, and d is discrete time, f is carrier frequency, α_i is input signal phase, and τ_{ji} is delay time of the connections.

We realize phase modulation by using a phase-modulation-type (more precisely, delay-time modulation type) spatial light modulator (SLM), which is named parallel-aligned liquid-crystal spatial light modulator (PAL-SLM). In the experiment below, we use two SLMs, i.e., one (SLM#1) is to generate an input vector, and the other (SLM#2) works as connection weights. The details will be given later in Fig.8.2. In the present system, we modulate only the phase (or actually, delay time), and keep the amplitude unchanged.

8.2.2 Memorizing Process

The learning dynamics to realize an associative-memory network is explained as follows. We apply sequential correlation learning to the weights

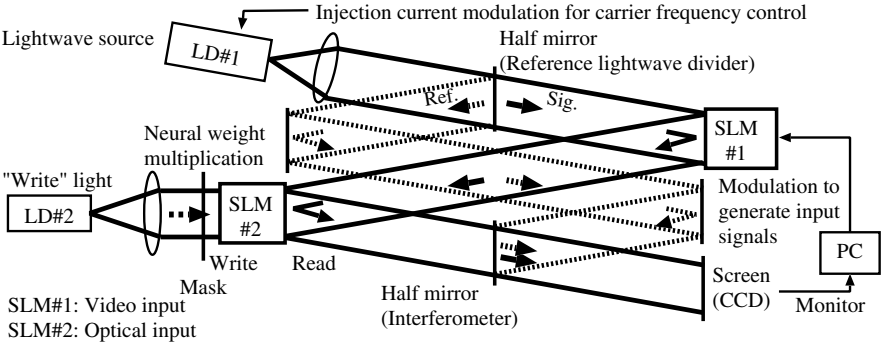


Fig. 8.2 Construction of the lightwave associative memory whose behavior is controllable by carrier-frequency modulation. (Reprinted from Fig.2 in [204] in figure caption of Fig.8.1 with permission.)

$w_{ji} = |w_{ji}| \exp(i2\pi f_\nu \tau_{ji})$. That is to say, we present the vector to be memorized ($\boldsymbol{s}_{\mu,\nu}$) to the output of the network instead of $\boldsymbol{y}$ in the ordinary Hebbian learning, while the conjugate $(\boldsymbol{s}_{\mu,\nu})^*$ to the input as $\boldsymbol{x}$, to make the network learn their correlation one by one. The process is, so to speak, a supervised complex-valued Hebbian learning. The updates of the amplitude and delay, $|w_{ji}|$ and τ_{ji}, of the weight w_{ji} is conducted in the manner described in Section 4.2 as

$$\tau \frac{d|w_{ji}|}{dt} = -|w_{ji}| + |y_j||x_i| \cos(\beta_j - \alpha_i - 2\pi f \tau_{ji}) \tag{8.4}$$

$$\tau \frac{d\tau_{ji}}{dt} = \frac{1}{2\pi f} \frac{|y_j||x_i|}{|w_{ji}|} \sin(\beta_j - \alpha_i - 2\pi f \tau_{ji}) \tag{8.5}$$

where $x_i = |x_i| \exp(i\alpha_i)$ and $y_j = |y_j| \exp(i\beta_j)$ are input and output signals, respectively, and τ is learning time constant [208].

In the case that all the memorized vectors are given at once, we can construct the weight matrix directly as the autocorrelation matrix.

8.3 Optical Setup

Figure 8.2 shows the system construction. We can control the carrier frequency f by choosing appropriate injection current of the light source (semiconductor laser: LD#1). Emitted lightwave is divided into signal and reference beams. The former is modulated by SLM#1, and becomes an input signal vector $\boldsymbol{x}$. Then the signal beams $\boldsymbol{x}$ are incident on SLM#2 and multiplied by the weights $\mathbf{W}$ whose values are determined by the backlight (LD#2) with an optical mask placed behind SLM#2. Figure 8.3 illustrates the SLM-surface assignment when the neuron number is 3 and the synapse number is 9, as well as the CCD surface to yield the summation.

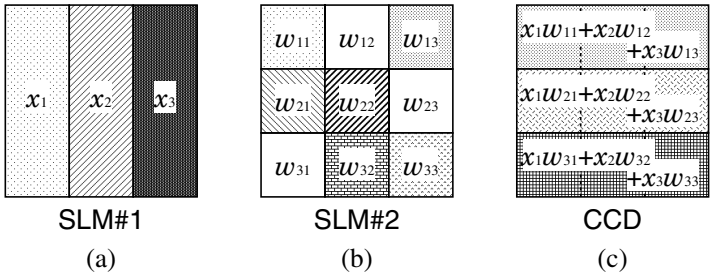


Fig. 8.3 Modulation-surface assignment of (a)input-signal generating SLM and (b)connection-weight multiplexing SLM, and (c)CCD detection-surface assignment in a 3-neuron and 9-synapse case for example. (Reprinted from Fig.3 in [204] in figure caption of Fig.8.1 with permission.)

The yielded neural output signals are mixed with the reference light beam at the half mirror to be homodyne-detected. To obtain two components orthogonal to each other, we modulate the input signal phase additionally by 0, 90, 180, and 270 degrees so that they yield four different interferences, resulting in extraction of the amplitude and phase information. The captured image is fed to the personal computer (PC) to generate the sum. (If we use a striped-surfaced detector, we can optically obtain the sum.) The PC generates the next signal vector to be sent to SLM#1 recurrently. The setup photograph is shown in Fig.8.4.

Figure 8.5 shows schematically the frequency-dependent behavior expressed as the modulation signals on the SLM surfaces. Even though the delay-time values τ_{ji} on SLM#2 are fixed, the resulting phase values of the weights $2\pi f\tau_{ji}$ are variable depending on the carrier frequency f. Hence, for an identical input vector, the system yields different output vectors dependent on f. In other words, the association behavior is dependent on the carrier frequency. This characteristic is effectively used, for both the learning and recall are consistently dependent on the frequency.

8.4 Frequency-Dependent Learning

The dependence of the homodyne output signals on the carrier frequency f is determined by the optical-path difference between the signal and reference light beams shown in Fig.8.2. When f varies, the interference fringe changes periodically with an interval of $c/\Delta L$.

The initial delay time $\tau_{ji0}^{\mathrm{Hebb}}$ is related to an arbitrary basis frequency f_0 as

$$\tau_{ji0}^{\mathrm{Hebb}} = \frac{\theta_0^{\mathrm{SLM}}}{2\pi f_0} + \frac{\Delta L}{c} \tag{8.6}$$

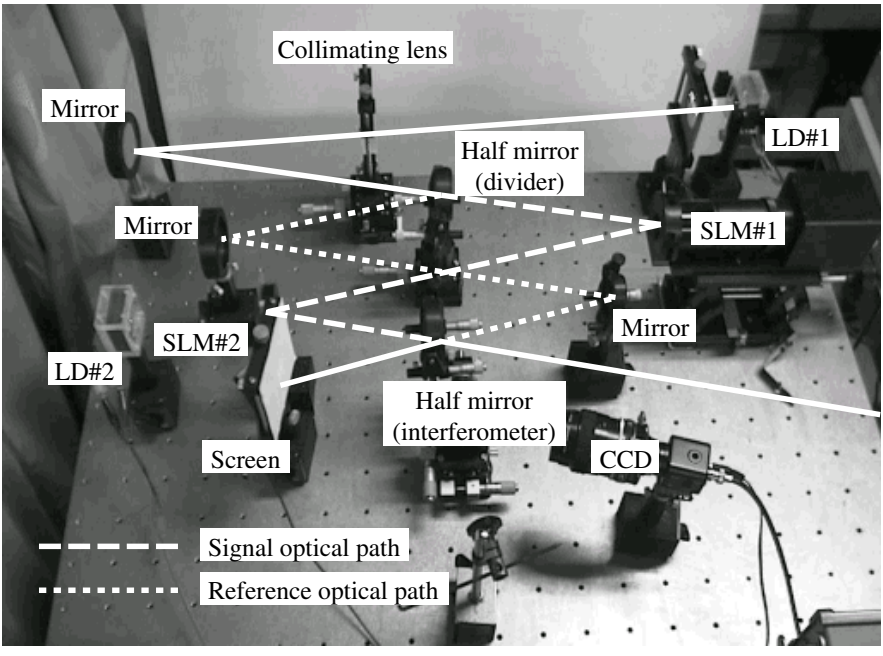


Fig. 8.4 Experimental setup.

where θ_0^{SLM} is an equivalent initial phase value of the SLM modulation, and is chosen at random in $[0, 2\pi]$. Generally, a large delay yields a large phase change even against a small frequency deviation. In the present system, the network learns an appropriate delay time $(\theta_0^{\mathrm{SLM}}/2\pi f_0) + (\Delta L/c)$ to have a suitable sensitivity to the frequency change.

In the experiment shown below, first we conduct learning numerically in the PC on the basis of the above-mentioned complex-valued Hebbian rule. Then, after learning is finished, we move to optical recall experiment. We named the μ th vector to be memorized at ν th frequency as $\boldsymbol{s}_{\mu,\nu}$. We show the vectors $\boldsymbol{s}_{\mu,\nu}$ by adjusting the carrier frequency at f_ν, one by one, to the neural network, and embed the vectors $\boldsymbol{s}_{\mu,\nu}$ in the memory in a frequency-dependent way. The relationship among the modulation phase $\theta_{ji}^{\mathrm{SLM\#2}}(f_0)$ at SLM#2, the corresponding delay time $\tau_{ji}^{\mathrm{SLM\#2}}$, and the optical-path difference ΔL are expressed as

$$\theta_{ji}^{\mathrm{SLM\#2}}(f_0) \equiv 2\pi f_0 \tau_{ji}^{\mathrm{SLM\#2}} = 2\pi f_0 (\tau_{ji}^{\mathrm{Hebb}} - \frac{\Delta L}{c}) \tag{8.7}$$

where $\tau_{ji}^{\mathrm{SLM\#2}}$ and $\tau_{ji}^{\mathrm{Hebb}}$ are the SLM delay time corresponding to $\theta_{ji}^{\mathrm{SLM\#2}}(f_0)$ and the total connection delay in (8.5), respectively. When the carrier

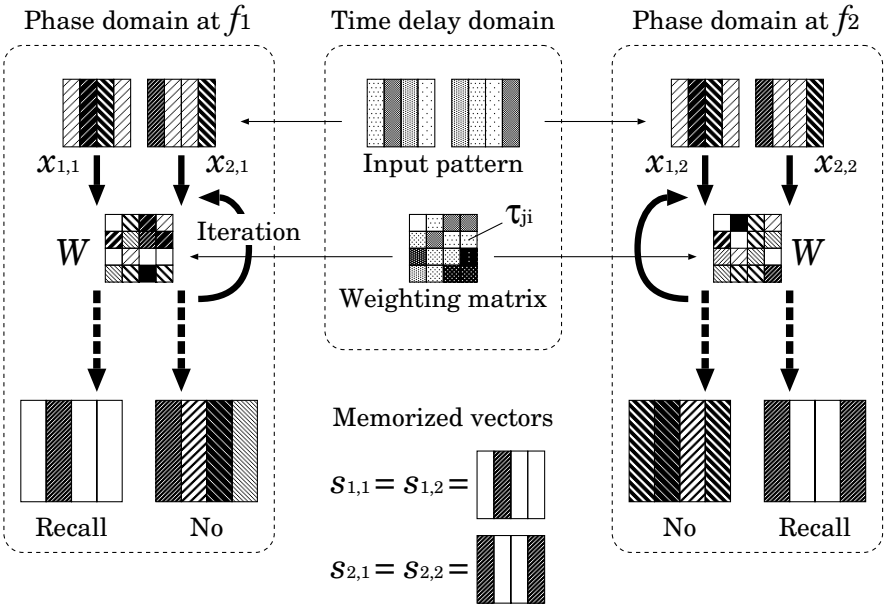


Fig. 8.5 Schematic illustration showing the frequency-dependent behavior expressed as the modulation signals on the SLM surfaces and results on the CCD. (Reprinted from Fig.4 in [204] in figure caption of Fig.8.1 with permission.)

frequency is f, the equivalent phase of the weight $\theta_{ji}(f)$ is determined frequency dependently as

$$\begin{aligned}\theta_{ji}(f) &= 2\pi f(\tau_{ji}^{\mathrm{SLM\#2}} + \frac{\Delta L}{c}) \\ &= \frac{f}{f_0}\theta_{ji}^{\mathrm{SLM\#2}}(f_0) + 2\pi f\frac{\Delta L}{c} \ . \end{aligned} \tag{8.8}$$

The behavior shown in Fig.8.5 is just an example of the intended operation of the system. The system has a certain function at a chosen frequency. From another viewpoint, we can regard the system as a fully parallel processor in the frequency domain when we use multiple optical sources simultaneously, just like the frequency-domain multiplexing (FDM) in the communications. On the other hand, if the system finds out an optimal frequency on its own with a feedback mechanism, we can recognize a type of *volition* in the system, resulting in a self-organizing context-dependent information processor [208]. The volition is described in Chapter 10 together with the concept of developmental learning.

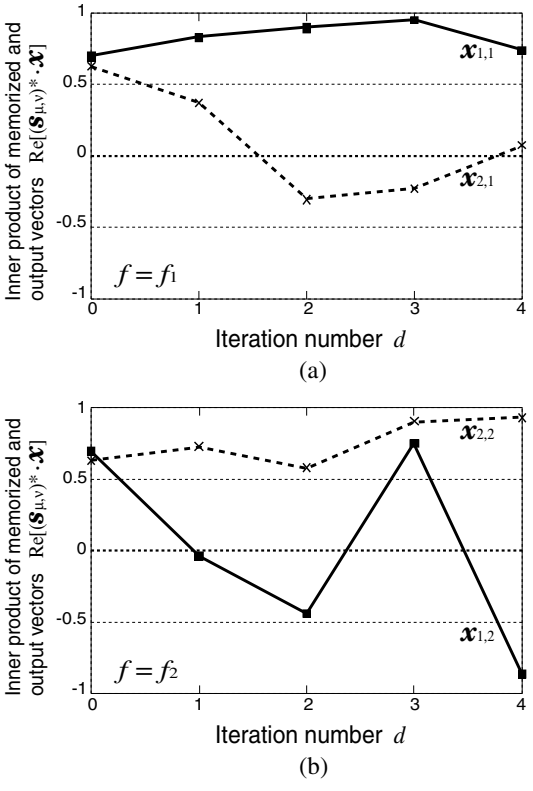


Fig. 8.6 Real part of inner products $\mathrm{Re}[(\boldsymbol{s}_{\mu,\nu})^* \cdot \boldsymbol{x}]$ versus recall iteration number. A unity inner product means a successful recall. Optical carrier frequency is (a)$f = f_1$, while (b)$f = f_2$. (Reprinted from Fig.5 in [204] in figure caption of Fig.8.1 with permission.)

8.5 Frequency-Dependent Recall Experiment

In Fig.8.2, the frequency of the light source LD#1 (wavelength $\cong$ 635[nm] ; $f_0 \cong 472$[THz]) is changed by injection current control. The frequency sensitivity is 10.5[GHz/mA]. The temperature is stabilized electrically. The resulting frequency controllability (resolution) is better than 0.016[GHz]. SLM#1 (Hamamatsu Photonics X6345) is modulated by a video input, while SLM#2 (X7665) is done by a spatially parallel optical input. However, basically we can use any modulation types of PAL-SLMs.

The optical path-length difference is about ΔL=6.8[mm], resulting in the frequency period of $c/\Delta L = 44.1$[GHz]. Numbers of neurons and synaptic connections are 9 and 81, respectively. We choose two signal vectors $\boldsymbol{s}_{\mu,\nu}(\equiv \boldsymbol{s}_\mu(f_\nu))$ and two carrier frequencies f_ν ($\mu = 1$ only, while $\nu = 1, 2$) where we intend that $\boldsymbol{s}_{1,1}$ be recalled at f_1, and $\boldsymbol{s}_{1,2}$ at f_2, respectively. The frequencies

are chosen as $f_1 = f_0$ and $f_2 = f_0 + c/(4\ \Delta L)$ so that the network behavior becomes independent even in a simple connection case.

We select two almost-orthogonal vectors to be memorized, $\boldsymbol{s}_{1,1}$ and $\boldsymbol{s}_{1,2}$. The amplitudes of all the vector elements are fixed to unity. We generate input vectors $\boldsymbol{x}_{1,1}$ and $\boldsymbol{x}_{1,2}$ by adding phase noise whose distribution is homogeneous within 30 % range of $\pm\pi$. The elements' amplitudes of the both input vectors are unity again. The learning process generates the weights w_{ji} on a PC. First, we initialize the delays τ_{ji} by choosing the initial phase θ_0 in (8.6) at random. Then the learning process based on (8.5) is iterated 1000 times, with which the weights settle at a sufficiently steady state. The learning gain K is 0.5.

Figure 8.6 shows the recall result for noisy input vectors $\boldsymbol{x}_{1,1}$ (near to $\boldsymbol{s}_{1,1}$ at f_1) and $\boldsymbol{x}_{1,2}$ (near to $\boldsymbol{s}_{1,2}$ at f_2) when one of the carrier frequencies f_1 or f_2 is chosen. In the case of f_1 shown in Fig.8.6(a), the real part of the inner product $\mathrm{Re}[(\boldsymbol{s}_{1,1})^* \cdot \boldsymbol{x}_{1,1}]$ converges almost at unity, which means that the system recalls the corresponding vector $\boldsymbol{s}_{1,1}$. On the other hand, the inner product $\mathrm{Re}[(\boldsymbol{s}_{1,2})^* \cdot \boldsymbol{x}_{1,2}]$ presents oscillatory behavior, which means failure in the recall of $\boldsymbol{s}_{1,2}$. In Fig.8.6(b), contrarily, the system recalls $\boldsymbol{s}_{1,2}$, while it does not recall $\boldsymbol{s}_{1,1}$. In this way, the frequency-dependent associative recall is realized.

8.6 Summary

We have presented a coherent optical associative memory whose behavior can be changed by the modulation of the carrier frequency. The system has a homodyne-detector structure where the signal and reference optical-path lengths are slightly different from each other. The path-length difference resulted in the frequency-dependent behavior. The carrier-frequency dependent complex-valued Hebbian rule functioned well to realize the frequency-dependent recall consistently.

This basic idea leads to future frequency-domain parallelism utilizing the vast optical-frequency bandwidth in optical neural networks. It is also applicable to future FDM systems and wavelength / wave number division multiplexed systems related to various wave phenomena. In addition, there have been several exploring ideas and analyses reporting, for example, optical frequency-multiplexed learning logic circuit [51] [52].

9

Adaptive Optical-Phase Equalizer

In the framework of the complex-valued neural networks dealing with phase values adaptively, we can realize various adaptive subsystems required in optical communications such as a learning phase equalizer. Modern optical communications attains a high degree of development mainly in trunk lines. Moreover, near-future networks provide subscribers with high-speed and multichannel information transmission over all-optical routers and switches. Thereby, we have to compensate the fiber dispersion varying with successively switched optical routes. The dispersion variation is very large since the high-speed multichannel optical communications occupies a wide frequency bandwidth. The optical-phase equalizer to be presented in this chapter can be one of the principles useful in such applications. As an example, we consider a system with supervised learning here.

9.1 System Construction

Figure 9.1 shows the neuron, the basic element, constructing the system. Its main characteristic is the multiple connections between m-th single input and n-th single neurons. Each connection possesses its own time delay and transparency. The connections altogether generate interference dependent on the optical carrier frequency.

We can directly relate the amplitude and phase of lightwave to those of signals in complex-valued neural networks. Then, to modulate phase in parallel two-dimensionally, we use a parallel-aligned liquid-crystal spatial light modulator (PAL-SLM), as we did in Chapter 8. To modulate amplitude, on the other hand, we use a PAL-SLM or ordinary (polarization-type) spatial light modulator (SLM) in combination with a polarizer. We express the input signal as $x_m \equiv |x_m| \exp(i\alpha_m)$, output signal as $y_n \equiv |y_n| \exp(i\beta_n)$, and connection weight as $w_{nm,h} \equiv |w_{nm,h}| \exp(i2\pi f \tau_{nm,h})$, respectively, where m and n are indices for input and output vectors, while h is that for multiple connections mentioned above.

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 143–149.
springerlink.com

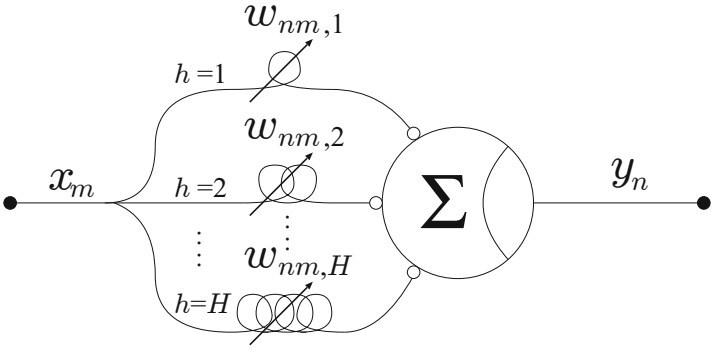


Fig. 9.1 Element constructing the adaptive phase equalizer. (Reprinted from Fig.1 in [209]: Sotaro Kawata and Akira Hirose: A coherent optical neural network that learns desirable phase values in frequency domain by using multiple optical-path differences, *Opt. Lett.,* 28(24):2524–2526, 2003, with permission.)

The purpose of the network is to generate a desirable phase value against input phase by adaptive processing. The task is classified into the function approximation described in Section 4.4. We realize function approximation in a frequency-dependent manner by the method explained in Section 4.4.5. The processing conducted by the neuron is expressed as

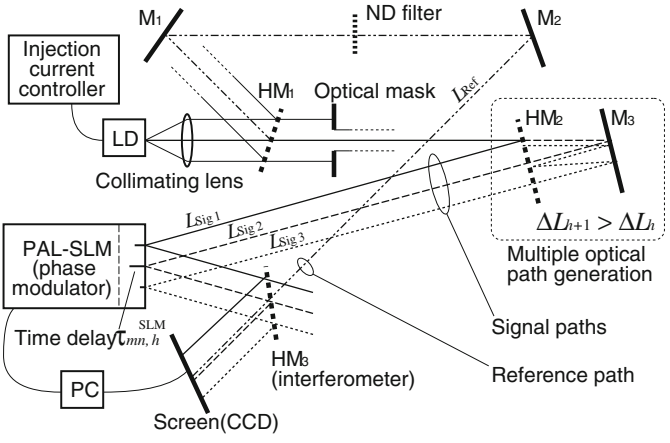
$$y_n = g\left(\sum_m \sum_h \left(|w_{nm,h}| \exp(i2\pi f \tau_{nm,h})\ x_m\right)\right) \tag{9.1}$$

$$g(u) \equiv A\ \tanh(B\ |u|)\ \exp(i \arg(u)) \tag{9.2}$$

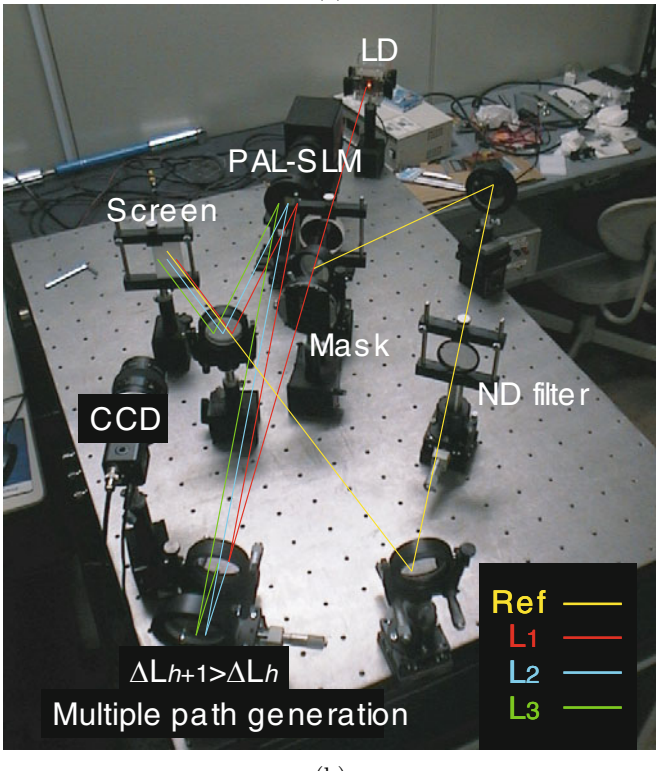
All the weighted inputs are summed in the complex domain to yield the internal state u. The activation function $g(u)$ is an amplitude-phase-type function defined in (9.2), A is saturation amplitude in output signals, and B is amplitude gain.

9.2 Optical Setup

We consider one of the simplest examples, i.e., an optical circuit consisting of a single neuron [209], [192]. Though the construction is simple, the function is fulfilling. We implement the neuron shown in Fig.9.1 as an optical circuit illustrated in Fig.9.2(a). Figure 9.2(b) is a photograph of the optical setup, where a three-armed optical interferometer is constructed as a set of self-homodyne circuits. We examine the phase of the circuit output by self-homodyning. We realize a frequency modulation by employing a semiconductor laser diode. The basis of the optical circuit is the same as that in the associative memory presented in Chapter 8.



(a)



(b)

Fig. 9.2 (a)Construction [209] and (b)photograph of the optical setup. (Reprinted from Fig.2 in [209] in figure caption of Fig.9.1 with permission.)

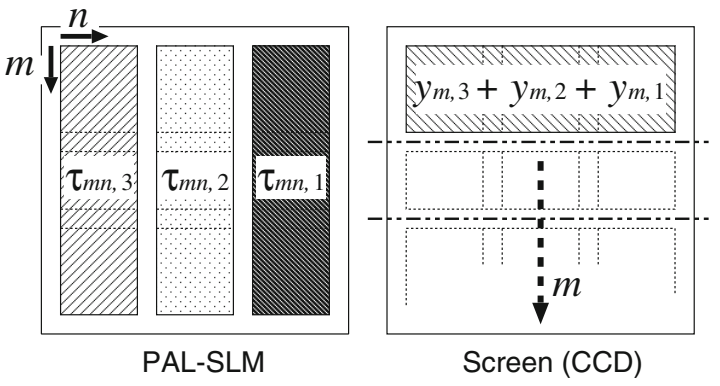


Fig. 9.3 Assignments of signals on the PAL-SLM and the CCD. (Reprinted from Fig.3 in [209] in figure caption of Fig.9.1 with permission.)

The delay length for input signals of the neuron is physically represented by the difference between the signal optical-path length L_{Sigh} and the reference length L_{Ref}, i.e., $\Delta L_h \equiv L_{\mathrm{Sigh}} - L_{\mathrm{Ref}}$, where we assume $\Delta L_1 < \Delta L_2 < \Delta L_3$. The delay time $\tau_{nm,h}$ of the connection weight $w_{nm,h}$ in total is the sum of the delay time of the SLM modulation, shown in Fig.9.3, and the delay in the optical-path difference, i.e., $\tau^{\mathrm{SLM}}_{nm,h} + \Delta L_h/c$. Here we assume $|w_{nm,h}|$=1 for simplicity. The optical sum and the nonlinearity in $g(\cdot)$ is realized by the optical detector (CCD: Charge Coupled Device) that detects interference result as shown in Fig.9.3.

9.3 Dynamics of Output Phase-Value Learning

We synthesize a desired output signal (phase) by adjusting the connection weights $w_{nm,h}$ in learning. As we discussed in Chapter 4, we have two frameworks in supervised learning in single-layered complex-valued neural networks, i.e., the complex-valued steepest-descent method and the complex-valued correlation learning superficially identical to the complex-valued Hebbian rule. Here we employ the delay-time learning, (4.97), which is the frequency-dependent version of the complex-valued correlation learning mentioned in Section 4.3.8. That is, we have

$$\tau \frac{d\tau_{nm,h}}{dt} = \frac{1}{2\pi f} \frac{|y_n||x_m|}{|w_{nm,h}|} \sin(\beta_n - \alpha_m - 2\pi f \tau_{nm,h}) \tag{9.3}$$

where τ without index is the time constant determining the learning speed.

The learning procedure is described as follows. First, we determine an arbitrary frequency $\hat{f}$. Then we present signal sets $(\hat{f}, \hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ to the system, where $\hat{\boldsymbol{y}}$ is the output to be learned for an input $\hat{\boldsymbol{x}}$. We update the connection

weights $w_{nm,h}$ according to (9.3) for all the connections. We repeat the above process for all the frequency–signal sets $\hat{f}$, $\hat{\boldsymbol{x}}$, and $\hat{\boldsymbol{y}}$ to be learned.

We choose initial weight values $w_{nm,h}$ such that the transparency is $|w_{nm,h}|=1$ and that the delay time is $\tau_{nm,h}=\theta_0/(2\pi f_0) + \Delta L_h/c$ where θ_0 is random value in the range of $0 \sim 2\pi$, where f_0 is the optical center frequency (the frequency without modulation). As the learning proceeds, the system learns to generate the desired output $\hat{\boldsymbol{y}}$ for input $\hat{\boldsymbol{x}}$ when the carrier frequency is $f = \hat{f}$. In the case that $f \neq \hat{f}$, the output can be different.

9.4 Performance of Phase Equalization

The setup for the optical experiment is shown in Fig.9.2. First, we confirm the learning dynamics in simulation using parameters in the optical experiments to be conducted. Then we proceed to an actual experiment. We prepare various desirable output phase values at four carrier-frequency points from 472.002 THz to 472.008 THz with an interval of 2GHz, and make the system learn the values. The upper limit of the learning iteration has been chosen at 200. We investigate the generalization characteristics in the frequency domain. We also define an error function as (9.4), and examine its evolution, i.e.,

$$E \equiv \frac{1}{2} \sum_{\mu} |\boldsymbol{y}(\hat{\boldsymbol{x}}_\mu) - \hat{\boldsymbol{y}}_\mu|^2 \tag{9.4}$$

where μ is the index for memorized vectors. In the following simulation and experiment, the learning is effective not only for phase but also for amplitude. However, with the present purpose of phase equalization, we investigate the output phase values in particular.

Figure 9.4 shows typical results of simulation (broken curves) and optical experiment (crosses). Closed circles in Fig.9.4 show the teacher signals that the system should learn. Figure 9.4(a) shows random output phase values before learning. However, for the output values at 80 iterations in Fig.9.4(b) and those at 200 iterations in Fig.9.4(c), we find that the system learns gradually the desirable outputs. Figure 9.4(d) presents the reduction of the error function. The almost monotonic decrease reveals the practical effectiveness of the learning dynamics.

By combining the processing elements shown here, we will obtain more complex phase curves in the frequency domain. In combination with the scaling in the frequency sensitivity, we can realize various types of dispersion compensation, besides wavelength-selecting routers. The scaling is actualized by preparing appropriate rough delays ΔL. The frequency-dependent learning will ultimately lead to novel parallelism based on frequency-domain multiplexing in neural networks utilizing the vast frequency bandwidth in lightwave.

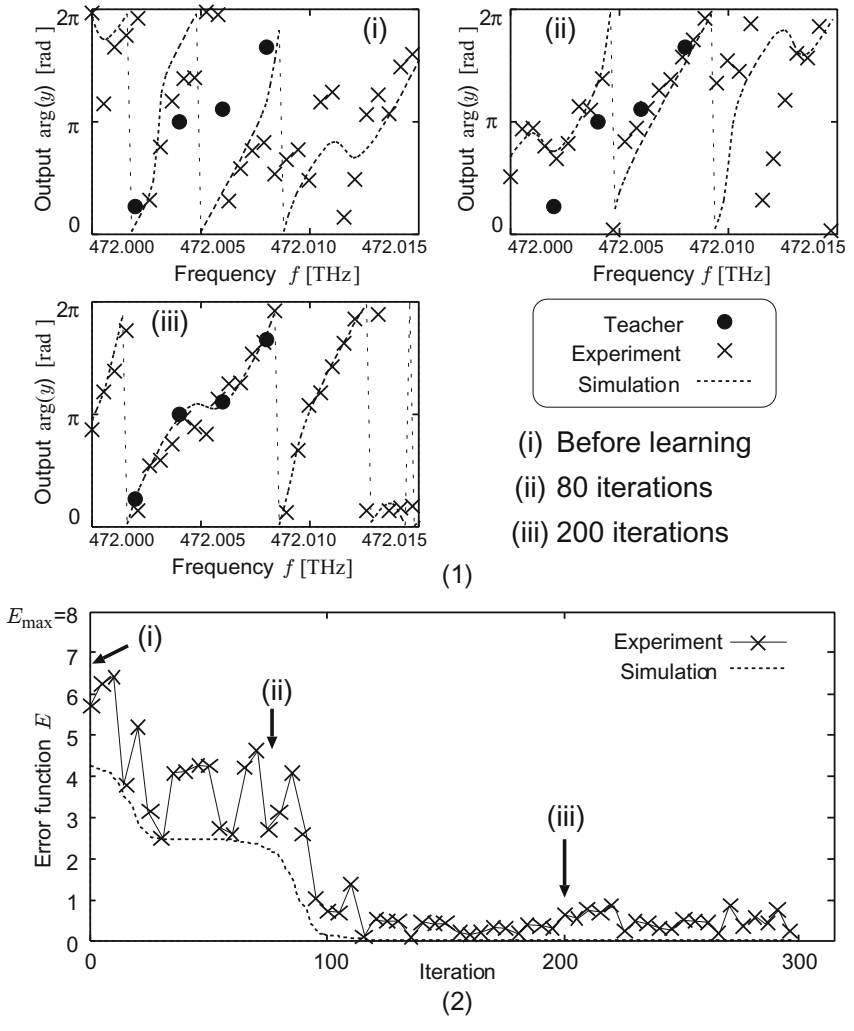


Fig. 9.4 Results of the optical experiment. Output phase values (a)before learning, (b)at 80 learning iterations, and (c)at 200 learning iterations. (d)Change in the error function. (Reprinted from Fig.5 in [209] in figure caption of Fig.9.1 with permission.)

Note that, in the present experiment mentioned here, we used a two-dimensional spatial light modulator (SLM) and other bulky optics. Therefore, the system is not so small. However, the principle of the learning and self-organization is widely applicable so that this type of system can be constructed based on, for instance, optical waveguides and photonic crystals. With such devices, we can design micro-optical circuits with high tolerance to mechanical turbulence.

9.5 Summary

In this chapter, we described a lightwave phase equalizer as a communications-network application of a coherent neural network. The learning dynamics and characteristics were presented. There are many other application areas in optical communications. One example is an adaptive recognition and classification of optical binary phase shift keying (BPSK) labels in photonic label routing for high-speed optical networks [210]. Such learning networks that treat the phase of waves adaptively are directly applicable also to sonic, ultrasonic, and other wave-related systems. They are also expected to pioneer novel future quantum devices that are highly adaptive.

10

Developmental Learning with Behavioral-Mode Tuning by Carrier-Frequency Modulation

We realize a so-called developmental learning with which a motion-control system learns multiple tasks similar to each other, or advanced ones, incrementally and efficiently by tuning its behavioral mode. The system is based on a coherent neural network whose carrier frequency works as a mode-tuning parameter. The coherent neural network is a class of the complex-valued neural networks. As presented in the previous chapters, we can modulate the behavior of the coherent neural network, such as learning and processing, by changing the carrier frequency. We make the carrier frequency represent the internal mode of the system, and utilize the carrier frequency as the key to realize the developmental learning. In this chapter, we consider two tasks related to bicycle riding. The first is to ride as temporally long as the system can before it falls down (Task 1). The second is an advanced one, i.e., to ride as far as possible in a certain direction (Task 2). We compare developmental learning to learn Task 2 after Task 1 with the direct learning of Task 2. Experiments demonstrate that the developmental learning enhances the efficiency in learning in total. We confirm the effectiveness of the developmental learning utilizing the carrier frequency as the mode-tuning key in the coherent neural network.

10.1 Development, Context Dependence, Volition, and Developmental Learning

Development is an important concept in the science to understand human beings. It has been widely studied in various fields such as cognitive science, psychology and neuroscience. In robotics, development is also expected to play an important role in various applications. For example, Asada et al. [211] proposed cognitive developmental robotics. If we regard a learning process as a search for an appropriate system state, we can interpret the developmental learning as follows. First we begin with a simple and dimensionally small

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 151–162.
springerlink.com

state, and then we increase the search dimension, if needed, to realize efficient learning.

The search dimension is often so large in solving real-world problems that developmental learning will be crucial to successful learning or self-organization workable in a realistically short time. The dimension-increasing procedure is also compared to a situation that, for example, a mother will first present a most basic goal to a child and, afterward, give him or her gradually advanced tasks one after another.

What condition is required for the system to realize such developmental learning? Among others, the most important is an appropriate structure that increases the effective number of internal states[1] or the dimension of them efficiently. To be simple, the system should generally prepare a small set of sufficient and effective parameters or variables incrementally. Thereby, in the parameter enlargement, it is significantly important that the parameter increment leads firmly to an expansion of behavior. At this point, the use of a coherent neural network (CNN) is promising, which we describe later.

Regarding the preparation and control of internal states in artificial neural networks, we have several proposals to utilize internal-mode modulations to extend learning and self-organization for the emergence of context-dependent behavior [212]. The context dependence is a behavioral feature explained as follows. For example, assume that people ask you, "What do you like?" If they were talking about sports, then you may answer, "I like tennis." However, if they were discussing swimming, you may answer, "I like backstroke." In such a way, we catch the course of the talk, and respond accordingly. This is one of the context-dependent behavior. Such behavior is regarded as emergence of volition. In other words, we possess a situation-dependent direction or intention, i.e., internal state, inside us, and decide what to do based on it.

In PATON proposed by Omori et al. [213],[214], the behavior of recognition and association in an associative memory system is controlled by a context-dependent switch. In motion control, Wolpert and Kawato [215],[216] prepared multiple neural-network modules. In their system, each output is weighted by a "responsibility coefficient" determined by the closeness of the tentative output value of each module to a desirable one. The coefficient is also used effectively as a weight in the learning process. Then the outputs are weighted and summed to yield a total output signal of the neural system. In such a manner, the system consistently learns an appropriate motion control, and then processes input sensory signals properly. Hartono and Hashimoto [217] also reported the successful introduction of annealing in the module-output integration.

Such switching and weighting-and-integrating methods increase the variety of neural states in learning and processing. In the extension, a crucial

[1] The internal state mentioned here is NOT the neuron's internal state described in Section 2.3 or 4.1.1, but a parameter existing in the network and determining the neural-network's behavior. By changing it, we realize the modulation of mood or intent of the network.

characteristic is again how flexible and effective the system can change its behavior. Simultaneously, a smooth behavioral variation, i.e., the generalization, is also an indispensable characteristic for natural neural processing.

We can expect that developmental learning will also be realized on the basis of CNNs using carrier frequency modulation for the behavioral mode tuning. In general, the CNN has a large freedom of tuning and a flexible generalization characteristic in its behavior by utilizing the complete-orthogonal property of the trigonometric basis functions used ($\cos\theta$ and $\sin\theta$, or $\exp[i\theta]$), for example, $e^{i2\pi f\tau_{ji}} = \cos 2\pi f\tau_{ji} + i\sin 2\pi f\tau_{ji}$ in the complex-valued Hebbian rule in (4.47) and (4.48) [208], [218]. In other words, the summation of a set of weighted sinusoidal curves is potentially capable of yielding a large variety of functions [204],[56].

In this chapter, we present a developmental learning architecture based on the CNN with carrier-frequency modulation for behavioral mode tuning [219]. The developmental learning is also regarded as a short-time growth. First, the network learns a certain task. Then, it learns a similar or advanced task quickly by utilizing the skill obtained previously. We consider bicycle riding. The first task is to ride a bicycle as temporally long as possible (Task 1), while the second and advanced one is to ride as far as possible in a certain direction (Task 2). This procedure is a class of developmental learning, though the situation is very simple. We compare the performance of developmental learning with that of the direct learning of Task two.

Note that the basic idea is already presented in Chapters 8 and 9 where we describe lightwave neural networks whose behavior is dependent on the optical carrier frequency. The same framework realizes a developmental learning. Additionally, in the process, the system finds the best frequency by itself. In this sense, the developmental learning is realized by self-organization.

10.2 Neural Construction and Human-Bicycle Model

Figure 10.1 shows a neuron in the coherent neural network. The input signal x_m, output signal y_n and weight w_{nm} are all complex numbers and composed of amplitude and phase. We adopt an amplitude-phase-type neuron activation function, which is introduced in Section 3.3.5, expressed in terms of the complex-valued input summation $s_n \exp[i\beta_n]$ with amplitude s_n, phase β_n and $i \equiv \sqrt{-1}$ as

$$s_n \exp[i\beta_n] \equiv \sum_m w_{nm} x_m \tag{10.1}$$

$$y_n = A \tanh(g s_n) \exp[i\beta_n] \tag{10.2}$$

where A and g (real numbers) denote saturation amplitude and small-signal gain that determines unsaturated gain, respectively. The function transforms

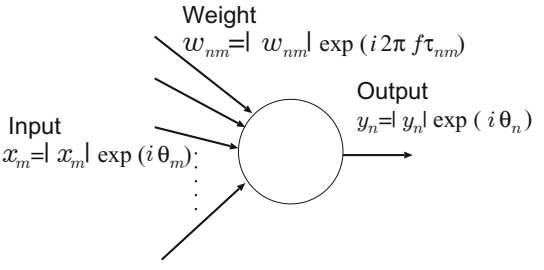


Fig. 10.1 Amplitude-phase-type complex-valued neuron in coherent network.

the signal amplitude in a saturation manner, just like the real-valued sigmoid function does, while it leaves the signal phase unchanged. The operation is a natural extension of the real-valued sigmoidal activation function.

A set of the neurons form a CNN with a carrier frequency, f, that works as the mode parameter. The neural connection weight w_{nm} is expressed by the connection amplitude (transparency) $|w_{nm}|$, delay time τ_{nm} and the carrier frequency f common to all the weights as

$$w_{nm}(f) = |w_{nm}| \exp[i2\pi f \tau_{nm}] \tag{10.3}$$

Therefore, the behavior of the coherent neural network depends on f according to (10.1)-(10.3). As mentioned above, we use this carrier frequency as the modulation parameter of the behavioral mode. If we fix the parameter value f, the behavioral mode is also fixed, whereas if we release it free to move to an optimal point self-organizingly, then the network learns and processes properly with the optimal parameter different from the previous one. A context-dependent behavior is also expected to emerge with this dynamics.

Figure 10.2 shows the construction of the coherent neural network interacting with a bicycle. It is a single-layered feedforward network. Variables are explained below in relation to human-bicycle model. The human-bicycle physical model is shown in Fig.10.3. We have variables such as handlebar azimuth ϕ, bicycle velocity v, wheel torque T, human rolling angle relative to bicycle σ, rolling angle of the total center of gravity of human and bicycle α. We have developed a mechanics simulator which is similar to that used in the study of walking. Figure 10.4 presents a window capture. The $x-y$ section shows a bicycle (larger box) and a human (smaller one) projected on the ground, while the $y-z$ and $x-z$ sections present their elevations. The angle-of-roll section illustrates their rear view of the rolling angle. The curves on the right-hand side show time evolutions of the rolling angle of the total human-bicycle gravity center α, handlebar azimuth ϕ, bicycle velocity v and wheel torque T.

The above variables are shown also in Fig.10.2, together with another variable γ which stands for the azimuth of the bicycle running direction. The

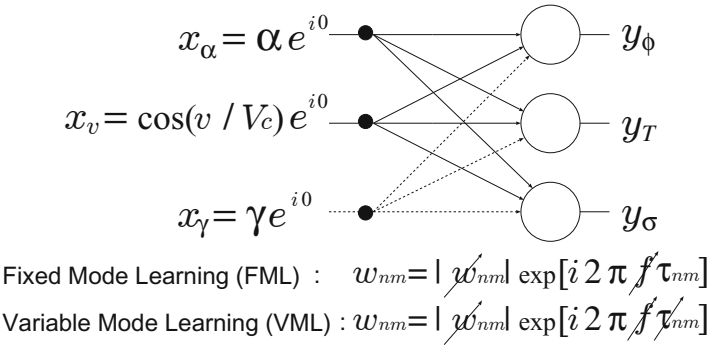


Fig. 10.2 Construction of the neural network. (Reprinted from Fig.2 in [219]: Akira Hirose, Yasufumi Asano, and Toshihiko Hamano: Mode-utilizing developmental learning based on coherent neural networks. In *International Conference on Neural Information Processing (ICONIP) 2004 Calcutta (Lecture Notes in Computer Sciences 3316)*, pages 116–121, Berlin, November 2004, Springer, (C) Springer-Verlag Berlin Heidelberg 2004, with permission.)

information γ works as a sight in the advanced task mentioned below. The sensory signals are real numbers and are fed to the CNN as

$$x_\alpha = \alpha \quad (= \alpha e^{i0}) \tag{10.4}$$
$$x_v = \cos(v/V_c) \quad (= \cos(v/V_c)\ e^{i0}) \tag{10.5}$$
$$x_\gamma = \gamma \quad (= \gamma e^{i0}) \tag{10.6}$$

where angles are represented in radians. Velocity v is normalized by a constant V_c and converted simply into an even function.

On the other hand, the motor signals are obtained at the neural outputs with the constants ϕ_c, T_c and σ_c as

$$\phi = \phi_c \, \mathbf{Im}[y_\phi], \tag{10.7}$$
$$\sigma = \sigma_c \, \mathbf{Im}[y_\sigma] \tag{10.8}$$
$$T = T_c \, \mathbf{Re}[y_T], \tag{10.9}$$

Then, provided that the neural input values and the initial neural connection phase values are chosen at around zero, which means a neutral condition, the bicycle will be controlled almost in neutral by the neural output, i.e., the handlebar is directed straight, the relative human angle of roll is zero, and the wheel torque is moderate. Such a situation may be helpful for the control if the initial weight delays are very small, though they are more at random actually in the experiment. However, note that, this condition is natural and does not violate generality. The values of the constants and the parameters are presented together with other parameters in the next section.

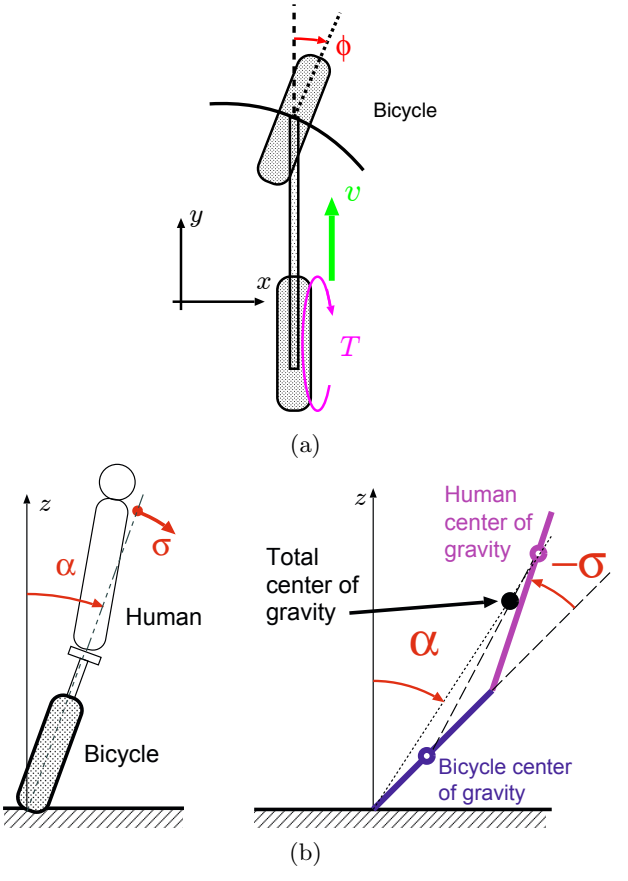


Fig. 10.3 Physical model of the human-bicycle system with variables and parameters: (a)Plan view and rear elevation. (Reprinted from Fig.3 in [219] in figure caption of Fig.10.2 with permission.)

10.3 Developmental Learning in Bicycle Riding

In the present developmental learning experiment, we employ the reinforcement learning having two learning stages. The first one is the random trial where the network changes neural connection weights $|w_{nm}|$ and τ_{nm} at random. The initial values are chosen also at random within certain ranges of the variables, e.g., $|w_{nm}| = 0.01 \sim 0.99$ and $\tau_{nm} = 0.1 \sim 99$ [ms]. We repeat the random trial for certain times, and we find the best trial. This stage is analogous to our rough trials in various ways to ride a bicycle in the real life.

The second stage employs the hill-climbing method by starting at the best condition obtained in the first random trial stage. The hill-climbing process changes the weight components with small fractions $\Delta|w|$ and $\Delta\tau$

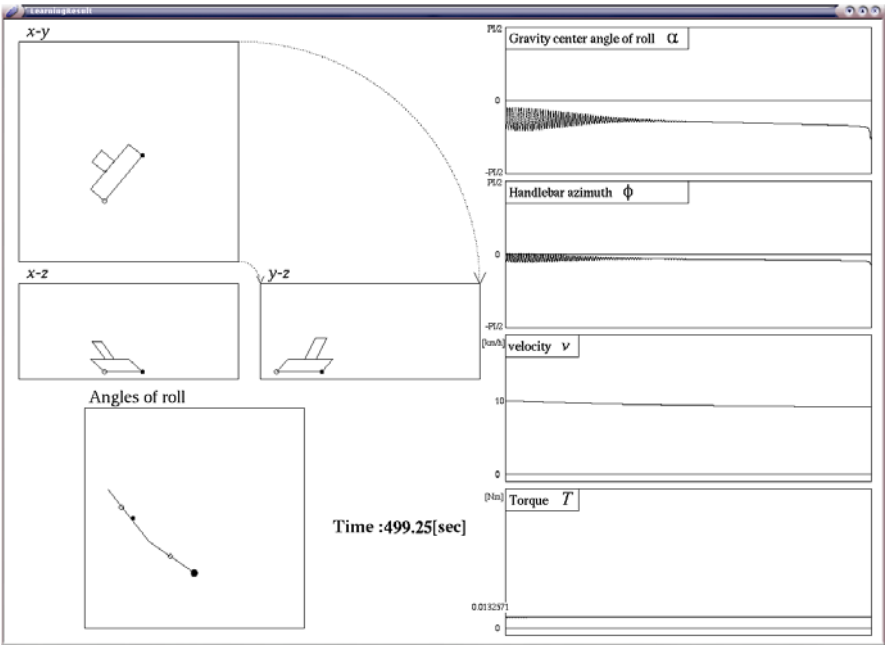


Fig. 10.4 Captured simulator display where the larger rectangle shows a bicycle, while the smaller one shows a human body. (Reprinted from Fig.4 in [219] in figure caption of Fig.10.2 with permission.)

as $|w_{nm}| \longleftarrow \left| |w_{nm}| + \Delta w \right|$ (i.e., $|w_{nm}| \geq 0$) and $\tau_{nm} \longleftarrow \tau_{nm} + \Delta\tau$, respectively. If the resulting effect is desirable, the network accepts the small changes. Otherwise, it rejects them. By repeating the process, the network searches a better set of connections. This stage may correspond to learning by iteration of fine adjustment for the human beings.

10.3.1 Task 1: Ride as Long as Possible

First, we try to learn Task 1, i.e., to ride as temporally long as possible. The carrier frequency in (10.3) is fixed at $f_0 = 100$[Hz] so that the behavioral mode is also fixed. The frequency f is kept unchanged at f_0. We call this learning style the fixed-mode learning (FML). In Task 1, the system does not use the direction information γ, which means a *blind* condition.

Figures 10.5 and 10.6 present typical results in Task 1. Figure 10.5(a) shows the riding time before falling down, t_{R}, for every random trial, Fig.10.5(b) shows the riding time t_{R} for the following hill-climbing learning by starting at the best trial condition in the random trial, and Fig.10.5(c) presents the riding locus for the longest-time trial after the hill-climbing

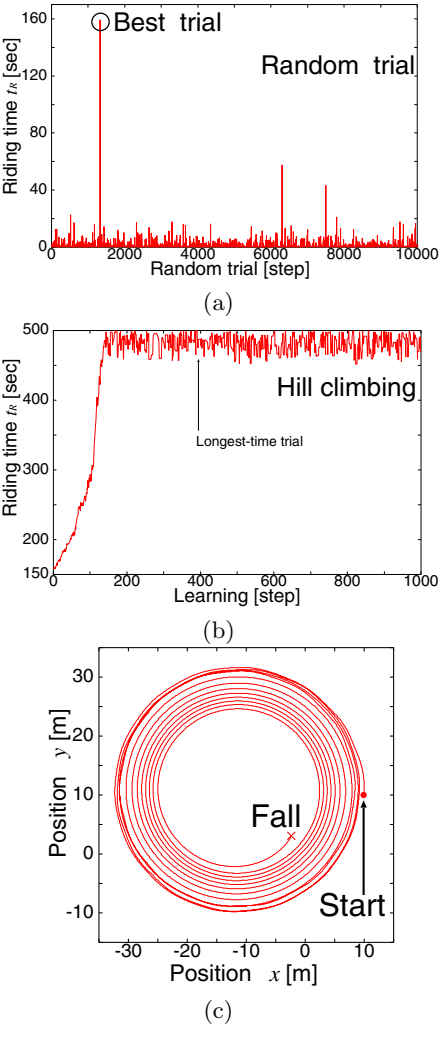


Fig. 10.5 Typical result in Task 1: (a)Riding time t_{R} versus random trial, (b)that versus hill-climbing learning with starting under the best weight-set condition in (a), and (c)riding locus for the longest-time trial after the hill-climbing learning converged. (Reprinted from Fig.5 in [219] in figure caption of Fig.10.2 with permission.)

process converged. The hill-climbing learning is found to extend the t_{R} increasingly, and to accomplish the goal of the long-time riding. However, the locus in Fig.10.5(c) reveals a round course.

However, the obtained behavior is found human-like and very attractive as follows. Figure 10.6 shows the (a)angle-of-roll of the center of gravity α,

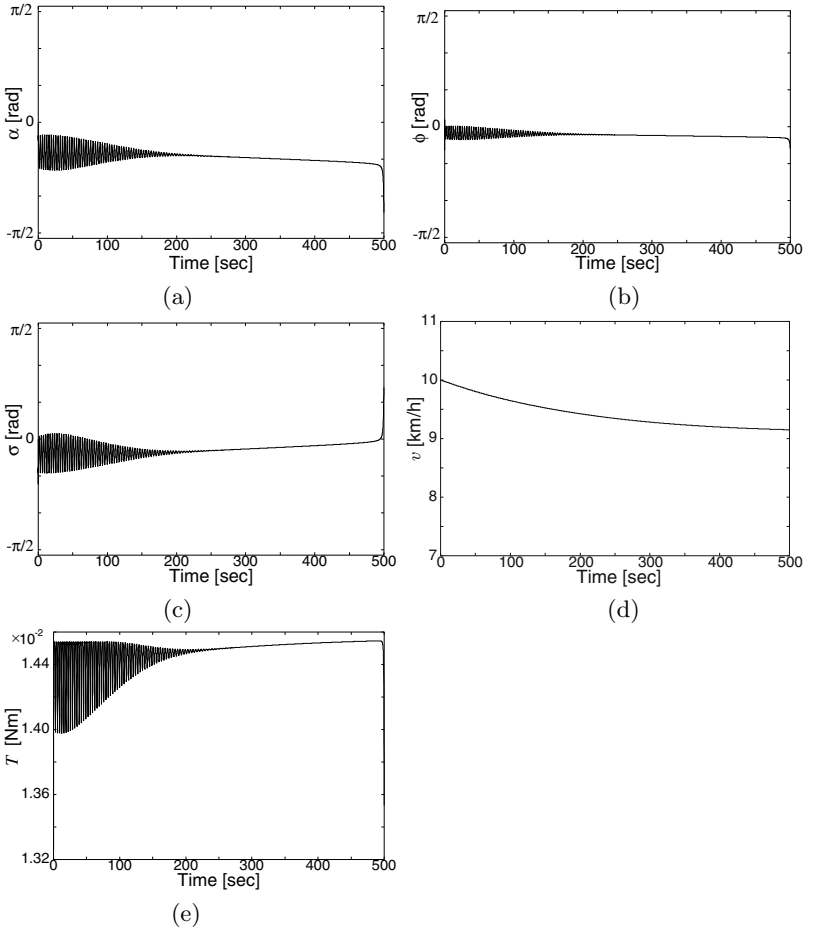


Fig. 10.6 Evolutions of the variables in typical Task 1 result corresponding to the ride in Fig.10.5(c): (a)Angle-of-roll of the center of gravity α, (b) handlebar azimuth ϕ, (c)human-bicycle rolling angle σ, (d)velocity v, and (e)torque T.

(b)handlebar azimuth ϕ, (c)human-bicycle rolling angle σ, (d)velocity v and (e)torque T, all against time, corresponding to the ride in Fig.10.5(c). At the beginning of the ride, the fluctuation of the roll α is large. But gradually the instability disappears. Other variables also present similar evolutions. That is to say, the neural learning has been performed so that a good riding becomes a stable point in the dynamics. This fact is evidence of the appropriateness of the learning.

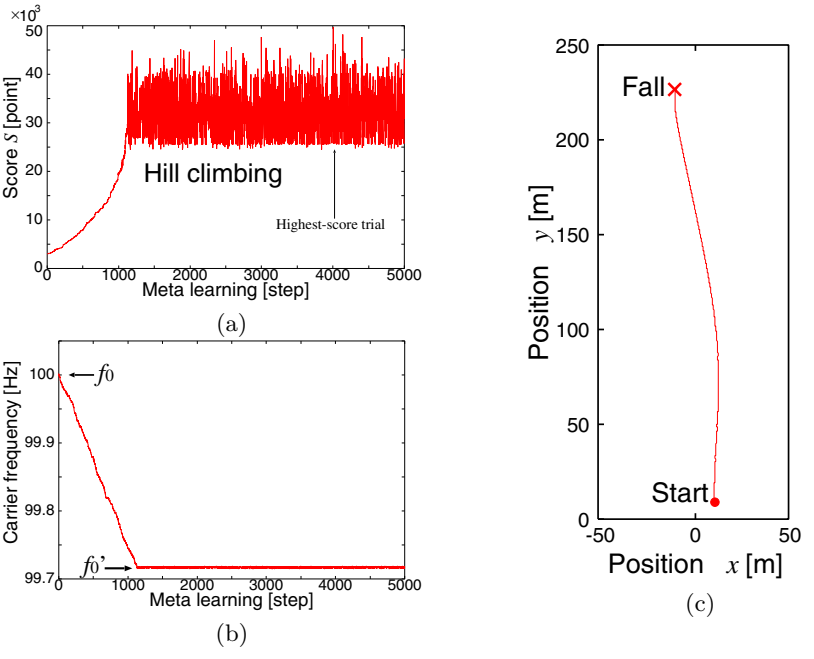


Fig. 10.7 Typical result in developmental VML in Task 2 in hill-climbing process starting with the best result in Task 1: (a)Score S for each riding, (b)self-organization of carrier frequency f, and (c)riding locus for the highest-score riding (Reprinted from Fig.6 in [219] in figure caption of Fig.10.2 with permission.)

10.3.2 Task 2: Ride as Far as Possible

Next, we assign an advanced task (Task 2), i.e., to ride as far as possible. We also prepare an eye to see in which direction the bicycle runs. The direction information $x_\gamma = \gamma$ is fed to the network as mentioned in Section 10.2. This is a *sighted* condition.

We set free the carrier frequency f in (10.3) to enable the network to change the behavioral mode. We call this learning style the variable-mode learning (VML). We expect that the system utilizes the variable frequency. The carrier frequency f is also changed by the hill-climbing method with a frequency fraction Δf as $f \longleftarrow f + \Delta f$ in addition to the hill-climbing learning of $|w_{nm}|$ and τ_{nm}. The frequency shift is equivalent to the variation in behavioral mode. The network searches a mode suitable for a far riding in a self-organizing manner.

In Task 2, we begin with the hill-climbing process by starting at the best result condition in Task 1. In addition, we define an evaluation function (score) S of the far riding so that, the further the bicycle runs, the higher the score becomes. (See, e.g., Ref.[219], [220] for details.)

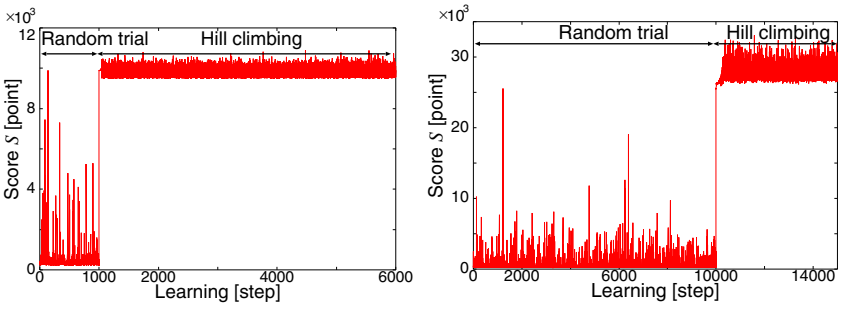


Fig. 10.8 Typical examples of scores in direct FML in Task 2 versus learning steps with (a)1,000- or (b)10,000-times random trials and following hill-climbing learning. (Reprinted from Fig.7 in [219] in figure caption of Fig.10.2 with permission.)

Figure 10.7 shows a typical result of the developmental VML in Task 2. The starting condition is the best result one in Task 1. In Fig.10.7(a), we find a quick increase in the score S. Figure 10.7(b) presents the variation of the carrier frequency f, which works as the behavioral mode parameter. It moves self-organizingly from $f_0 = 100$[Hz] to an optimal value f_0'. That is, the network finds out the mode most suitable for the environment by itself.

Figure 10.7(c) shows the riding locus of the highest-score result after the hill-climbing learning converged. In comparison with 10.5(c), the course has been clearly straightened. The result is obtained quickly by the mode modulation of the longest-time-ride condition to adapt to the new environment, i.e., the advanced task of long-distance riding.

10.3.3 Comparative Experiment: Direct FML in Task 2

We also conduct experiments on developmental FML, direct VML, and direct FML. (See details in Ref. [220].) Figure 10.8 shows typical results of the direct FML experiments without the learning in Task 1. We repeat random trials for 1,000 or 10,000 times and, then, move to hill-climbing learning afterward. The direction information γ is fed to the neural network. The initial state is statistically the same as that of Task 1.

In Fig.10.8(a) and (b), we find that, in the random trial, a high-score probability is very low. Moreover, even in the hill-climbing learning, the score increases only slightly, which suggests that the random trial does not bring the network to the vicinity of a truly ideal state.

10.3.4 Comparison between the Results

When we compare the developmental VML (Section 10.3.2), developmental FML, direct VML, and direct FML (Section 10.3.3), we find that the

developmental VML is the most effective method in total [220]. In Fig. 10.7(b), we also find that the carrier frequency self organizes to realize such effective learning. We can see that the network learns similar or advanced tasks quickly by changing the internal mode parameter.

10.4 Summary

We have presented the idea of the mode-utilizing developmental learning architecture based on the coherent neural network. The network learns similar or advanced tasks incrementally by using its cumulative skill by changing the behavioral mode-tuning parameter, i.e., the carrier frequency of the coherent network. The mode parameter has been found adjusted self organizingly and smoothly in the developmental learning. The developmental learning and required architecture will be of growing importance in building so-called brain-like systems.

11

Pitch-Asynchronous Overlap-Add Waveform-Concatenation Speech Synthesis by Optimizing Phase Spectrum in Frequency Domain

To obtain high-quality results in speech synthesis, we should record utterance elements, which is to be concatenated, as temporally long as possible to avoid a sense of discomfort in listeners. In the case of announcements in trains, for example, we prepare word- or segment-long utterances, and concatenate them to generate simple sentences. However, when we try to synthesize free sentences required in daily life with this method, we need such a huge database that we cannot construct it by recording real utterances. Instead, we may be able to synthesize speech by sampling short elements of utterance, and memorize their short-time spectra to be concatenated. In practice, however, it is very difficult to tune the way of concatenation since, to yield reasonable speech, it is crucial to reproduce the features in waveforms such as pulse sharpness. In this chapter, we present a complex-valued neural network that adjusts phase values in frequency spectra adaptively to realize an ideal concatenation. The network functions in the frequency domain to obtain desired waveforms in the time domain. Phase shift in the frequency domain corresponds to temporal shift in the time domain. Such frequency-domain processing using complex-valued neural networks is useful in various fields such as image processing where we deal with spatial frequency.

11.1 Pitch-Synchronous and -Asynchronous Methods in Waveform Concatenation

11.1.1 Pitch Mark and Pitch-Synchronous Method

In short-waveform-based phoneme synthesis, the so-called Pitch-Synchronous OverLap-Add (PSOLA) method is often employed. The PSOLA method utilizes the periodicity included in vowels [221]. That is to say, first, we extract the pitch mark, indicated by closed circles in the left-top corner in Fig.11.1, which is the position in time where the amplitude takes a maximum positive or negative peak value within a period. Then, synchronously to the pitch

A. Hirose: Complex-Valued Neural Networks, SCI 400, pp. 163–175.
springerlink.com

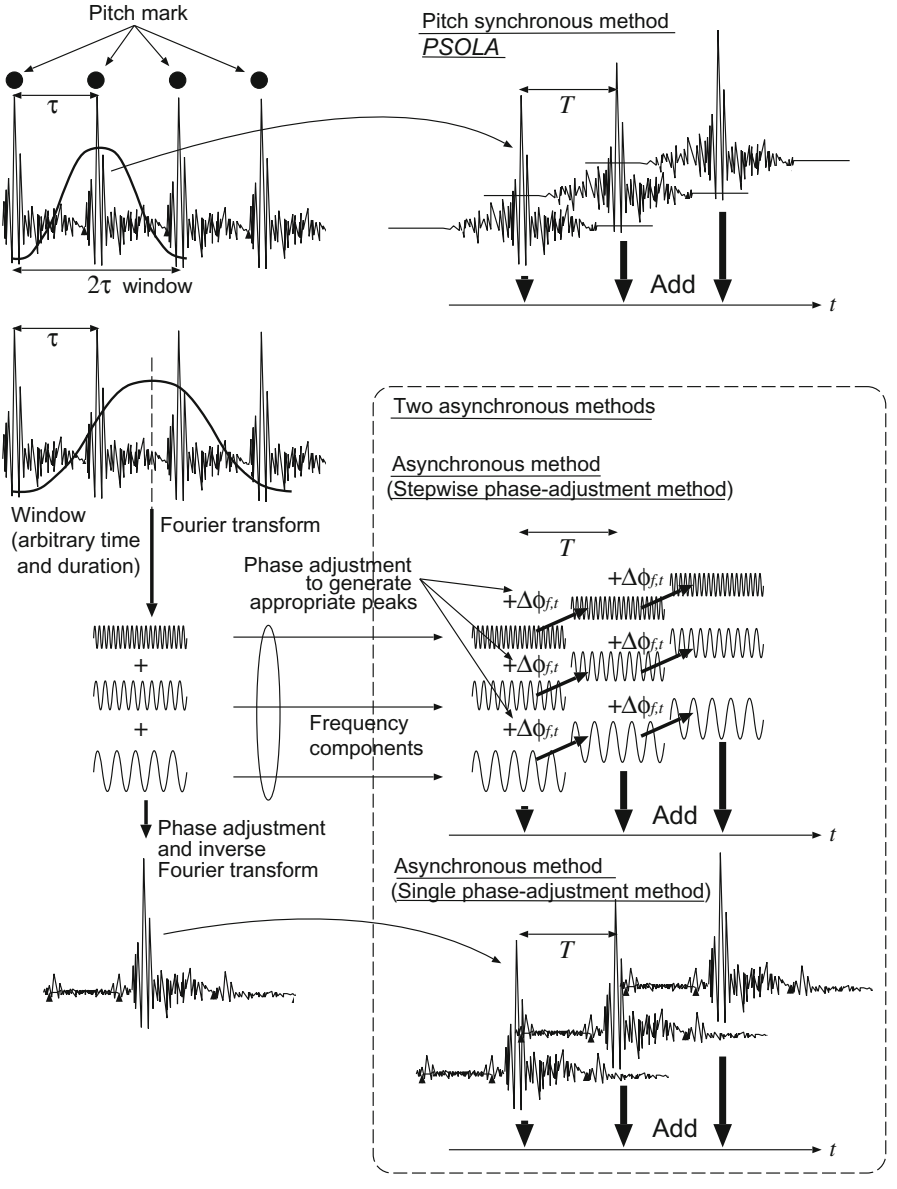


Fig. 11.1 Schematic diagram comparing conventional PSOLA method and asynchronous methods proposed.

mark, we put a sampling window, e.g., a Hanning window, and sample the waveform. The length of the window is chosen double the pitch-mark period. Then we concatenate the sampled short waveform to reconstruct a vocal waveform having a required duration. When the fundamental period of the

recorded waveform is τ, we sample the waveform with a 2τ-length Hanning window, and put the copies of the sampled short waveform sequentially with the period of τ, or another desired period of T, in an overlapping manner. We iterate the copy to obtain a desired duration of the vocal as shown in the top in Fig.11.1. This method is classified into so-called synchronous methods, for the process synchronizes with the pitch.

The PSOLA method makes good use of the periodicity in voiced sound such as vowels and sonant, and achieves a high performance in synthesis widely. However, when it fails in extracting pitch marks, the performance is greatly deteriorated. Additionally, it cannot be employed to deal with unvoiced sound such as consonant because it has no pitch. In this sense, this method has fundamental inconsistency. In this context, we expect a novel asynchronous method to synthesize phonemes, voice, and speech. The basic idea is presented below.

11.1.2 Human Senses Sound Spectrum

We hear others speaking, and understand what they say or who is speaking. What mechanism do we have in ourselves? In the inner ears, we have cochlear canals filled with lymph fluid. The inside walls of the cochlear canals are ciliated. The cilia sense sonic wave having propagated through outer ears and drum membranes, and transmit neural signals to auditory fields. In the cochlear, when the sound frequency is high, the cilia placed near the entrance are shaken with the sonic wave. If the frequency is low, the cilia at the bottom are shaken well. In other words, the cochlear translates the frequency spectrum of the sound into the spatial distribution of the cilia's movement. The information about the location and intensity of the cilia's movement is transmitted to the brain. Consequently, we can say that human beings listen to the frequency spectrum of sounds. Actually, the vowel /a/ has a spectrum profile completely different from that of /i/.

Therefore, we introduce the following synthesis procedure. We obtain the short-time frequency spectrum of a sampled short waveform by using the short-time Fourier transform or wavelet transform. The spectrum is expected to contain the voice and speaker's information. Then, by concatenating the short-time spectrum and inversely transforming them into a time-domain waveform, we will obtain the expected voice with desirable duration. This process can be asynchronously conducted if human beings listen to the frequency spectrum.

11.1.3 Problem in Simple Asynchronous Speech Synthesis

We can actually try the asynchronous synthesis to hear the result. Then we find that, although an almost desirable voice is obtained, the voice seems to be made by many people, just like a chorus. In a precise investigation of

the synthesized waveform, we find that the pitch-mark pulse, to which we pay attention in the PSOLA method, has disappeared. (We observe it later in Fig.11.4.) This is the problem. We hear the chorus as if many identical people were uttering the sound at an identical tone. Accordingly, though human beings listen fundamentally the spectrum, we catch also a little more precise waveform at the same time.

Why does the pulse disappear? In the above method, we obtain the time-domain waveform by inversely transforming the spectrum. In the process, we weight and sum sinusoidal waveforms having various frequencies. If the phases of the sinusoidal waves are *in-phase* at a certain moment, they yield a sharp pulse like a delta function. Contrarily, if they have random phase values, they result in an averaged noisy waveform.

Therefore, we would expect a better voice if we can align the phases at a certain interval in some fashion, and generate a waveform with appropriate pulse sharpness. Adaptive processing of phase is well realized by use of the complex-valued neural networks. In this chapter, we describe an asynchronous method to synthesize time-domain waveform with appropriate pulse sharpness by using a complex-valued neural network [222].

11.1.4 Pitch-Asynchronous Methods: Single Phase-Adjustment Method and Stepwise Phase-Adjustment Method

We synthesize a waveform based on the above idea. We process frequency-domain signal adaptively to generate a waveform having appropriate pulse sharpness in the time domain. That is, we first employ the Fourier transform or wavelet transform to convert a time-domain signal into a frequency-domain one. Then we manipulate the phase and amplitude with a complex-valued neural network. Afterwards, we go back to the time domain to obtain a desirable waveform. In this process, a phase shift of each frequency component is directly related to a temporal shift in the waveform. Therefore, the time-domain adjustment is realized by the frequency-domain phase shift.

In Fig.11.1, we illustrate two types of the methods.

1. **Single phase-adjustment method** We synthesize a short-time waveform having the same duration as that of the window from the short-time spectrum. (See bottom in Fig.11.1.) In this process, we adjust the phases in the frequency components to obtain appropriate pulse sharpness in the time domain. Then we concatenate the short waveform sequentially. In this method, we have no need to extract the pitch. Once we obtain a short-time waveform, we can generate voice with arbitrary duration with a small calculation cost. However, we cannot vary the pulse sharpness time-dependently. In this sense, this method lies between the PSOLA method and the stepwise phase-adjustment method mentioned below.

2. **Stepwise phase-adjustment method** We generate short-time spectrum time-sequentially and synthesize them. (See middle in Fig.11.1.) We adjust the phase values in short-time spectrum components to generate a pulse where we want each time. Then we concatenate the short waveforms time-sequentially. This method requires phase adjustment every time, and the calculation cost becomes slightly larger. However, it can realize arbitrary pulse sharpness at any time.

The single phase-adjustment method may be more desirable because of the smaller cost in many cases. However, in this chapter, we focus on the stepwise phase-adjustment method, which has larger freedom in synthesis, to examine the phase-adjustment ability in the complex-valued neural network. Then we can examine the effectiveness of the network more keenly. Incidentally, the processing elements in the single phase-adjustment method are identical with those in the stepwise method, and can be implemented in the same way.

11.1.5 Convolutions and Neural Networks

Complex-valued neural networks functioning in the frequency domain are applicable in various wave-information processing. For example, blurring in images, which is caused by spatial misalignment, can be compensated by phase shifts in the spatial-frequency domain [223]. In this way, we can utilize the frequency-domain complex-valued neural networks in wide areas.

Moreover, the Fourier transform itself has a neural aspect. That is to say, in the Fourier transform, we multiply a time-domain waveform by $e^{-j2\pi ft}$, and integrate the result in terms of t. This process is weighting and summing, which is the most fundamental process in neural networks. In general, we can say that the so-called convolution operation is a type of neural operations.

Therefore, the above-mentioned phase-adjustment operation can also be constructed as a double-stage neural network where we have a neural layer for (adaptive) Fourier-transform-like processing and another one for adaptive phase shift. However, in this chapter, we have the Fourier transform as it is and, after the transform, we adjust the phase values independently. We can understand what happens more easily that way.

11.2 Construction of Stepwise Phase-Adjustment System

Figure 11.2 shows the block diagram of the whole system. We explain the dynamics of the asynchronous synthesis system using a complex-valued neural network as follows. We record voice, and obtain the complex spectrum $X^{\mathrm{org}}_{f,t}$. In a digital system, we deal with discrete time t and frequency f. Therefore, we express them as suffixes here. They are sequentially numbered with intervals of time-shift length and frequency resolution, respectively.

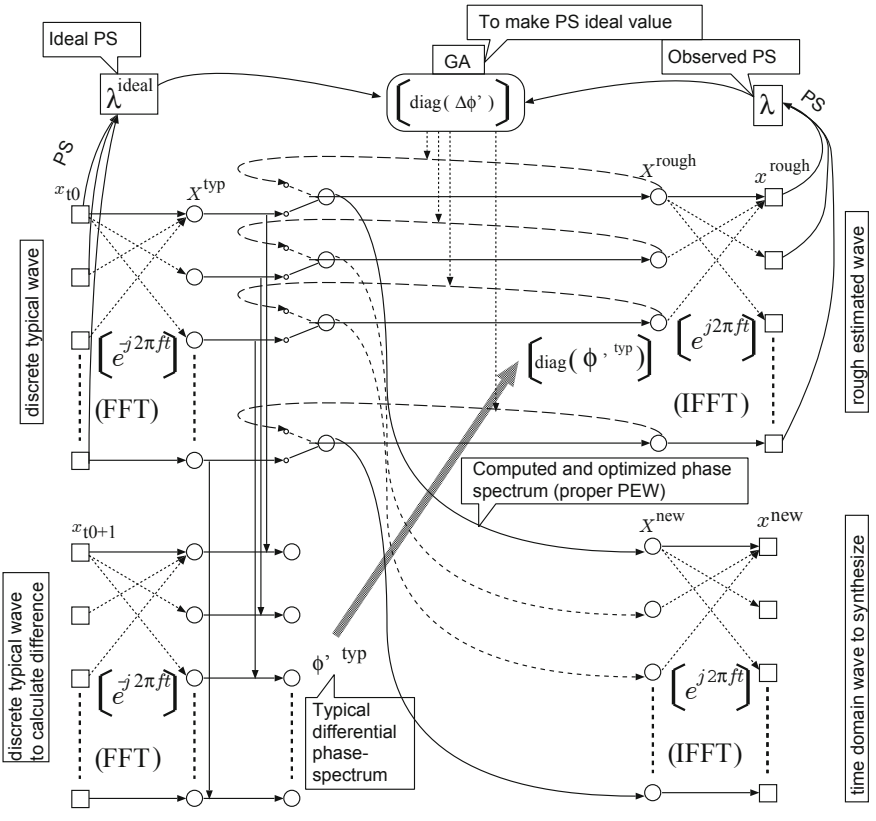


Fig. 11.2 Block diagram of the asynchronous speech-synthesis system with adaptive adjustment of phase values in frequency components using the complex-valued neural network. (Reprinted from Fig.2 in [222]: Keiichi Tsuda and Akira Hirose: Pitch-asynchronous overlap-add waveform-concatenation speech synthesis by using a phase-optimizing neural network, In *International Conference on Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies (KES) 2003 Oxford (Lecture Notes in Computer Sciences / Artificial Intelligence 2774)*, volume Proc. 2, pages 332–339, Berlin, September 3–5, 2003, Springer, (C) Springer-Verlag Berlin Heidelberg 2003, with permission.)

In the recorded voice, we determine a moment t_0 when the voice is in a sufficiently steady state. We call the complex spectrum at this moment the typical complex spectrum X_f^{typ}.

$$X_f^{\mathrm{typ}} = X_{f,t_0}^{\mathrm{org}} \tag{11.1}$$

Next, we define the typical difference of phase spectrum ϕ_f^{typ} by considering how much the phase spectrum (a set of phase values in Fourier coefficients) of

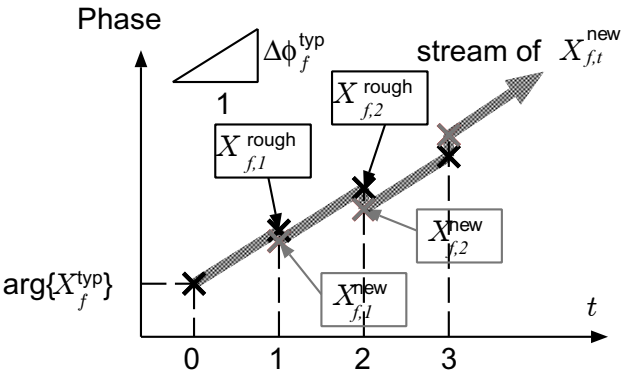


Fig. 11.3 Conceptual illustration of the determination of running spectrum with phase optimization. (Reprinted from Fig.3 in [222] in figure caption of Fig.11.2 with permission.)

the recorded voice proceeds in a unit time shift. This value is a typical phase difference determined in every Fourier component. To determine the value with accuracy, we may take an averaged phase shift in a certain period as

$$\phi_f^{\text{typ}} = \frac{1}{m_\phi} \arg\left(\frac{X_{f,t_0+m_\phi}^{\text{org}}}{X_{t_0}^{\text{org}}} \right) \tag{11.2}$$

That is, we calculate the average shift value of the phase in m_ϕ-times temporal shifts, and name this value, ϕ_f^{typ}, the typical difference phase spectrum. We expect that, the larger the time shift m_ϕ is, the less the possible transient noise and fluctuation will become. A longer observation time is desired. However, it may be variable according to long-time changes in the vocal.

Then we synthesize complex spectrum $X_{f,t}^{\text{new}}$ time-sequentially as shown in Fig.11.3. We call the time-sequential spectrum the running complex spectrum. First, we use the typical complex spectrum X_f^{typ} as the running spectrum at $t = 0$.

$$X_{f,0}^{\text{new}} = X_f^{\text{typ}} \tag{11.3}$$

We calculate roughly the complex-spectrum candidate at the next time step, $X_{f,1}^{\text{rough}}$, by shifting the phase of the preceding-time spectrum (in the present case, the typical complex spectrum itself X_f^{typ}) by the value of the typical difference phase spectrum ϕ_f^{typ}.

$$X_{f,1}^{\text{rough}} = X_{f,0}^{\text{new}} \exp\left(j\phi_f^{\text{typ}} \right) \tag{11.4}$$

where $j \equiv \sqrt{-1}$. However, we need to optimize the phase value with an appropriate adjustment. We express this phase-optimization operation as $phopt[\cdot]$. The detail will be given later.

$$X_{f,1}^{\text{new}} = phopt\left[X_{f,1}^{\text{rough}}\right] \tag{11.5}$$

We synthesize the running spectrum $X_{f,t}^{\text{new}}$ by repeating the above operation time-sequentially.

$$\begin{cases} X_{f,t+1}^{\text{rough}} = X_{f,t}^{\text{new}} \cdot \exp\left(j\phi_f^{\text{typ}}\right) \\ X_{f,t+1}^{\text{new}} = phopt\left[X_{f,t+1}^{\text{rough}}\right] \end{cases} \tag{11.6}$$

We apply the short-time inverse Fourier transform to the obtained running spectrum to generate a time-domain waveform. Then we concatenate a series of the waveforms into voice with a desirable duration.

In Fig.11.2, we found the block diagram showing the whole process. At the left-top, the recorded signal is input, and Fourier transformed, to yield the typical complex spectrum. Then the left-lower part calculates the typical difference phase spectrum which is used in the recurrent calculation of the running complex spectrum (upper center). The upper-right part inversely Fourier transforms the running spectrum to generate the synthesized waveform. The lower-right part does the same in parallel for pulse-sharpness evaluation.

11.3 Optimization of Pulse Sharpness

We conduct the operation of the phase optimization $phopt[\cdot]$ in the frequency domain (upper center in Fig.11.2) as follows.

1. We keep the amplitude unchanged.

$$\left|X_{f,t}^{\text{new}}\right| = \left|X_{f,t}^{\text{rough}}\right| \tag{11.7}$$

2. We adjust the output phase value so that the value satisfies the following condition:
   (a) It is near to the input rough estimate.
   (b) Pulse sharpness (PS) λ, which is explained below, is near to an ideal PS value λ^{ideal}

The above ideal PS, λ^{ideal}, can be calculated with an actual voice waveform, which we do not need to know a priori, or we can assign a certain value. The unnecessity of the knowledge of λ^{ideal} is one of the advantages of the present method. For example, if we want to synthesize unvoiced fricative, of which PS is low, it can become automatically low. This feature solves the problem that a conventional vocoder often suffers from errors caused by mal-discrimination of voiced sound from unvoiced one to switch its signal sources (pulse source and white noise). Moreover, the proposed method is capable of generating intermediate waves between pulse and white noise. In addition, particular sounds such as plosive sound (unvoiced but having a high

PS) may be synthesized naturally if the PS value represents the particular characteristics.

We have a variety of values that may represent pulse sharpness. In this chapter, we introduce a discrete version of the index defined by Hamagami [224]. (In Ref.[224], the value is called PEW, though we call it PS here.) The PS, λ, is defined for a short-time waveform x_t in discrete time t as

$$\lambda \equiv \sum_{t=-N/2}^{N/2-1} \frac{x_t^2}{N} \Bigg/ \left(\sum_{t=N/2}^{N/2-1} \frac{|x_t|}{N} \right)^2 \tag{11.8}$$

where N is the number of sampling points in the window. When the power in the window is constant, we have $\lambda = 1$. In contrast, when there exists a single impulse in the window, then $\lambda = N$. In this way, λ represents the pulse sharpness of the waveform in the window.

To construct an effective neural dynamics, we define an energy function to be minimized in relation to the PS. We consider a linear sum of two energies. That is, (a)E^{bind} that is minimized when the output-waveform PS is the same as that of the input rough estimate, and (b)E^{pulse} that is minimized when the PS is the same as the ideal PS. We possess only the phase spectrum $\phi_f \equiv \arg(X_f)$ as the adjustment parameters without amplitude spectrum $A_f \equiv |X_f|$ which is unchanged.

$$E(\phi_f) = \alpha E^{\text{bind}}(\phi_f) + E^{\text{pulse}}(\phi_f) \tag{11.9}$$

where α is a constant. The terms on the right-hand side are defined as follows.

(a) $\underline{E^{\text{bind}}}$: This energy is low when the phase spectrum is near to the rough estimate $\phi_f^{\text{rough}} \equiv \arg(X_f^{\text{rough}})$. Note that, since the phase has a modulo-2π arbitrary property, the energy is defined with a 2π periodicity, and also with a minimum point at $\phi_f = \phi_f^{\text{rough}}$, as

$$E^{\text{bind}}(\phi_f) = \frac{1}{P} \sum_f A_f^{\text{typ}} \left\{ 1 - \cos\left(\phi_f - \phi_f^{\text{rough}} \right) \right\} \tag{11.10}$$

where $A_f^{\text{typ}} \equiv |X_f|$ is weights to give high-power components more effectiveness, and $P \equiv \sum_f A_f^{\text{typ}}$ is a value similar to total power.

(b) $\underline{E^{\text{pulse}}}$: This energy is low when the obtained PS is near to the ideal PS, and expressed as

$$E^{\text{pulse}}(\phi_f) = (\lambda - \lambda^{\text{ideal}})^2 \tag{11.11}$$

We determine the neurodynamics in such a manner that the energy is to be minimized. A typical way is the steepest descent method. In the present case, we calculate partial derivatives of the energy $E(\phi_f)$ in terms of ϕ_f. This treatment is simple if we assume that the total power P is unchanged, though it actually changes.

Table 11.1 Conditions of recording, analysis, and synthesis of voice. (Reprinted from Table 1 in [222] in figure caption of Fig.11.2 with permission.)

Uttered phoneme	/a/
Environment	Sound-insulated room
Sampling frequency	10kHz
A/D conversion	16bit linear quantization
Time-frequency analysis	Short-time Fourier transform
Window	Hanning window
Window length	512 sampling points
Window shift length	128 sampling points

Here we introduce a process easier to conduct, i.e., the hill-climbing (or hill-descending) method. We make the phase values ϕ_f fluctuate and, if the energy decreases, we accept the change. If the energy does not decrease, we reject it. We reduce the energy as if we walk around on a mountain to descend. (Note that this method has a weakness in nonlocality since neurons have to be informed of the output state.) In this process, we impose a condition of

$$\phi_f = -\phi_{-f} \tag{11.12}$$

so that the waveform to be generated in the time domain becomes real-valued.

11.4 Experimental Results

Table 11.1 shows the conditions of recording, analysis, and synthesis. Parameters and variables are obtained from the recorded voice as follows. Figure 11.4(a) shows the recorded waveform. In this case, we find the pitch in the negative direction, though we do not use it. We intend to keep this PS in the synthesis.

We put a window at the vertical broken line in Fig.11.4(a) to extract the typical complex spectrum X_f^{typ}. We also obtain the typical difference phase spectrum ϕ_f^{typ} by observing the phase changes in a single time shift, or an average in 10 time shifts. We calculate the ideal PS, λ^{ideal}, from the inversely Fourier transformed X_f^{typ}.

Figures 11.4(b)–(e) present synthesis results. Waveforms obtained without any phase optimization are shown in Figs.11.4(b) and (c). In (b), the typical difference phase spectrum was calculated for a single time shift ($m_\phi = 1$), while in (c), it was done for 10 shifts ($m_\phi = 10$). In these results, we find that, as time advances, the negative pulse height reduces. However, note that the synthesis procedure suggests no changes in the amplitude spectrum. That is to say, we observe the reduction of the PS. When we use fixed phase differences ϕ_f^{typ}, slight errors possibly included accumulate gradually but steadily, and deteriorate the PS very quickly. The degradation in (c) is found slower

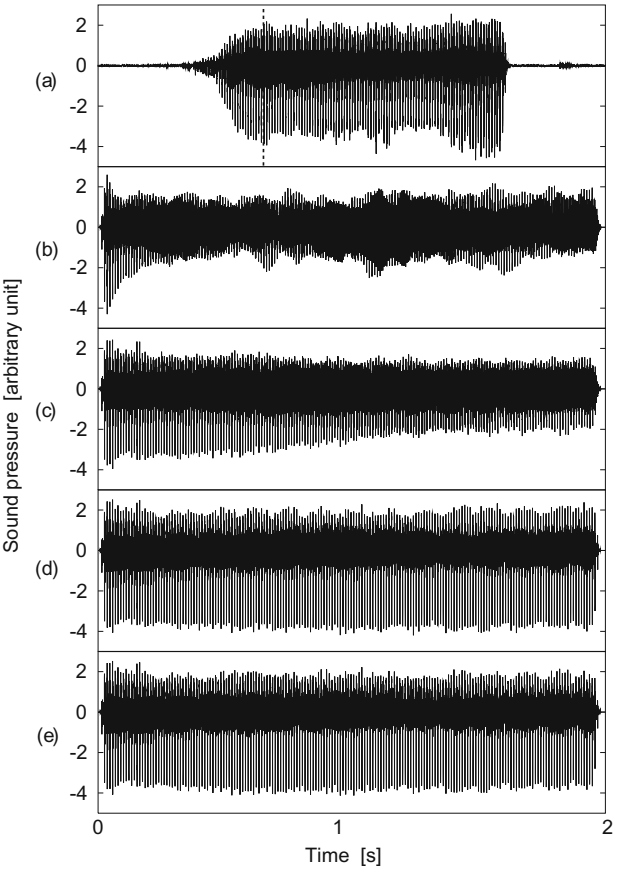


Fig. 11.4 Voice waveforms: (a)recorded voice, (b)synthesized voice without phase optimization ($m_\phi = 1$), (c)synthesized voice without phase optimization ($m_\phi = 10$), (d)synthesized voice with random-search phase optimization, (e)synthesized voice with group-search phase optimization, (Reprinted from Fig.4 in [222] in figure caption of Fig.11.2 with permission.)

than that in (b), since the long observation resulted in a more precise estimation of ϕ_f^{typ}.

When we listen to the low PS voice, it sounds like a chorus uttered by many identical people. We can analyze the effect qualitatively in such a way that, when many people utter at an identical tone, their asynchronous pitch pulses cancel out each other.

On the other hand, Figs.11.4(d) and (e) show the results obtained with the help of phase optimization process based on the complex-valued neural network. The waveform in (d) is the result optimized by the so-called Matyas' random search (simple random search). The initial difference phase spectrum

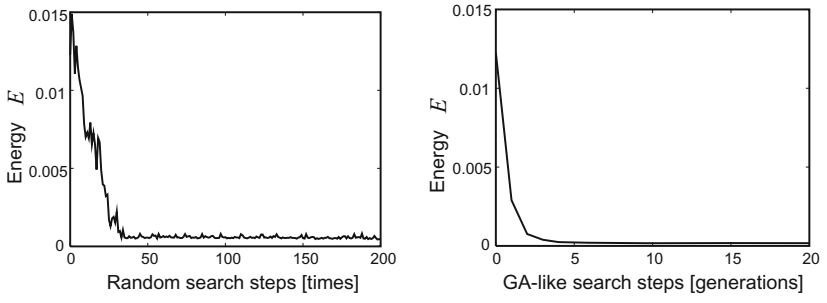


Fig. 11.5 Energy versus optimization steps in (a)Matyas' random search and (b)group search. (Reprinted from Fig.5 in [222] in figure caption of Fig.11.2 with permission.)

in the random search was chosen as the rough estimate ϕ_f^{rough}. The waveform in (e) is the result obtained with a group search similar to genetic algorithm. (See Ref. [222] for details.)

In Figs.11.4(d) and (e), there is no degradation of the PS observed in Figs.11.4(b) and (c). We confirm the effectiveness of the phase optimization. In a listening test, we also find a good result, instead of chorus-like voice.

Figure 11.5(a) and (b) shows the energy changes versus optimization steps in the case of Figs.11.4(d) and (e), respectively. In both cases, the energies decrease almost monotonically. The results suggests an ideal adjustment in the complex-valued neural-network processing.

11.5 Summary

In this chapter, we presented the pitch-asynchronous voice synthesis method in which the complex-valued neural network adjusts the phase values in the Fourier components. In this method, first we transform time-domain signals into frequency-domain spectra. Then we process the information in the frequency domain, and inversely transform it into time-domain data. In general, frequency-domain information is often incompatible with complex-valued processing. For example, as we had a glance in Section 4.8, when we apply the independent component analysis (ICA) to blind separation related to cocktail-party effect in the frequency domain, we inevitably process complex spectrum by a complex-valued neural network. In such wave-related applications, the amplitude-phase-type activation function works better than the real-imaginary-type one [122]. Treatment suitable for information in respective applications enhances the adaptability in neural networks.

Besides sound applications, we can apply similar processing in image applications where we deal with pixel values and their spatial-frequency data.

In this case, the phase in the spatial-frequency domain corresponds to the location shift. Another analogical application is frequency-modulation continuous-wave (FMCW) radars. We convert obtained frequency-domain reflection data into time-domain one, which is almost identical to what we obtain with a pulse-type radar system. In electron-wave devices, we will be able to apply the Fourier-domain neural-network approach in dynamics consideration in the wave-number space. In this way, complex-valued neural networks have wide application fields in the Fourier domain.

Closing Remarks

In Part I, we described the specific features and the fundamentals of complex-valued neural networks. Then, in Part III, we presented various applications, and explained their dynamics in learning and processing. Neural networks are more and more expected to deal with complex data. This fact may lead to unnecessity of calling the networks "complex-valued" neural networks in the future. On the other hand, their basics become increasingly essential and indispensable in neural networks. In particular, it will be of greater importance to be conscious of what feature we expect to use them, and then what type of complex neurons and network structure we should choose. Originally, the artificial neural networks have been inspired by the human brain, and the brain has a variety of structures dependent on expected functions. It will become further significant to make clear the purpose of using complex-valued networks. The author hopes that this book sparks up discussions on complex-valued neural networks so that they become more and more useful in various applications.

References

[1] Hirose, A.: Special session on complex-valued neural networks. In: Baba, N., Jain, L.C., Howlett, R.J. (eds.) International Conference on Knowledge-Based Intelligent Information Engineering System & Applied Technologies (KES 2001), Osaka, Amsterdam, pp. 550–580. IOS Press, Ohmsha (2001)

[2] Hirose, A.: Special session on complex-valued neural networks. In: Damiani, E., Howlett, R.J., Jain, L.C., Ichalkaranje, N. (eds.) International Conference on Knowledge-Based Intelligent Information Engineering System & Applied Technologies (KES 2002), Crema, pp. 623–647. IOS Press, Ohmsha (2002)

[3] Nitta, T.: Special session on complex-valued neural networks. In: Wang, L., Rajapaske, J.C., Tan, K.C., Halgamuge, S., Fukushima, K., Lee, S.Y., Furuhashi, T., Kim, J.H., Yao, X. (eds.) International Conference on Neural Information Processing (ICONIP 2002), Singapore. Nanyang Technological University (November 2002)

[4] Hirose, A.: Special session on complex-valued neural networks: Theories and applications. In: Kaynak, O., Alpaydin, E., Oja, E., Xu, L. (eds.) International Conference on Artificial Neural Networks (ICANN)/International Conference on Neural Information Processing (ICONIP 2003), Istanbul, pp. 941–1000. Springer, Heidelberg (2003)

[5] Hirose, A.: Special session on complex-valued neural networks. In: Baba, N., Jain, L.C., Howlett, R.J. (eds.) International Conference on Knowledge-Based Intelligent Information Engineering System & Applied Technologies (KES 2003), Oxford, pp. 304–357. Springer, Heidelberg (2003)

[6] Hirose, A.: Special session on complex-valued neural networks. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) International Conference on Neural Information Processing (ICONIP 2004), Calcutta, pp. 104–135. Springer, Heidelberg (2004)

[7] Hirose, A.: Special session on complex-valued neural networks. In: World Congress on Computational Intelligence (WCCI)/International Joint Conference on Neural Networks (IJCNN 2006), Vancouver. IEEE (July 2006)

[8] Aizenberg, I., Hirose, A., Zurada, J.M.: Special session on complex-valued neural networks. In: de Sa, J.M., Alexandre, L.A., Duch, W., Mandic, D. (eds.) International Conference on Artificial Neural Networks (ICANN 2007), Porto, pp. 838–893. Springer, Heidelberg (2007)

[9] Aizenberg, I., Hirose, A., Zurada, J.M.: Special session on complex-valued neural networks. In: Liu, D. (ed.) World Congress on Computational Intelligence (WCCI)/International Joint Conference on Neural Networks (IJCNN 2008), Hong Kong. Research Publishing Services (2008)
[10] Zurada, J.M., Hirose, A., Aizenberg, I.: Special session on complex-valued neural networks. In: IEEE/INNS International Joint Conference on Neural Networks (IJCNN 2009), Atlanta. IEEE (2009)
[11] Hirose, A., Sato, S.: Practical Applications of Complex-Valued Neural Networks, Project. In: Tohoku Univ., RIEC, Nation-wide Cooperative Research Projects. RIEC, Tohoku University (2009)
[12] Mandic, D.P., Aizenberg, I., Hirose, A.: Special session on complex-valued neural networks: Theory and applications. In: World Congress on Computational Intelligence (WCCI)/International Joint Conference on Neural Networks (IJCNN 2010), Barcelona. IEEE (2010)
[13] Hirose, A., Sato, S.: Practical Applications of Complex-Valued Neural Networks, Project. In: Tohoku Univ., RIEC, Nation-wide Cooperative Research Projects. RIEC, Tohoku University (2010)
[14] Kuroe, Y., Nitta, T.: Special session on theories and applications of neurocomputing: From complex number to clifford algebra. In: FAN Symposium. SICE (2010)
[15] Aizenberg, I., Hirose, A., Mandic, D.P., Zurada, J.M.: Special session on complex-valued neural networks. In: IEEE/INNS International Joint Conference on Neural Networks (IJCNN 2009), Atlanta. IEEE (2011)
[16] Nitta, T., Kuroe, Y.: Special session on clifford algebraic neural networks. In: International Conference on Neural Information Processing (ICONIP 2011), Shanghai (2011)
[17] Hirose, A.: Complex-valued neural networks. In: Tutorial, International Joint Conference on Neural Networks (IJCNN), Atlanta. IEEE Computational Intelligence Society (CIS) video archive (2009)
[18] Mandic, D.P., Aizenberg, I., Hirose, A.: Complex-Valued Neural Networks. In: Tutorial, World Congress on Computational Intelligence (WCCI)/International Joint Conference on Neural Networks (IJCNN 2010), Barcelona (2010)
[19] Hirose, A.: Complex-Valued Neural Networks in Remote Sensing and Imaging. In: Tutorial, IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2011), Vancouver. IEEE Gioscience and Remote Sensing Society, IEEE GRSS (2011)
[20] Hirose, A.: Task Force on Complex-valued Neural Networks (TF-CVNN), IEEE Computational Intelligence Society (CIS) Neural Network Technical Committee (NNTC)
[21] Hirose, A.: Complex-valued neural networks for more fertile electronics. Journal of the the IEICE 87(6), 447–449 (2004) (in Japanese)
[22] Hirose, A.: The "super brain" and the complex-valued neural networks. Suuri-Kagaku, Saiensu-sha, Japan (492), 78–83 (2004) (in Japanese)
[23] Hirose, A.: Complex-valued neural networks (review). Journal of the Institute of Electrical Engineers of Japan 131(1), 2–8 (2011) (in Japanese)

[24] Hirose, A.: Highly functional pattern information processing – the "super brain". In: Nitagai, K. (ed.) The Third-Generation University – The University of Tokyo Challenges the Frontiers of Knowledge, pp. 64–65. University of Tokyo Press, Tokyo (2002)
[25] Goto, E.: The parametron – A new circuit element which utilizes non-linear reactors. Paper of Technical Group of Electronic Computers and Nonlinear Theory, IECE (July 1954) (in Japanese)
[26] Goto, E.: On the application of parametrically excited non-linear resonators. The Journal of the Institute of Electrical Communication Engineers of Japan (IECE) 38(10), 2761 (1955) (in Japanese)
[27] Aizenberg, N.N., Ivaskiv, Y.L., Pospelov, D.A.: A certain generalization of threshould functions. Dokrady Akademii Nauk SSSR 196, 1287–1290 (1971)
[28] Hirose, A., Suksmono, A.B., Kawata, S.: A super brain watches phase and amplitude – applications to interferometric radar imaging systems and adaptive lightwave equalizers. Journal of the IEICE 87(6), 482–487 (2004) (in Japanese)
[29] Hebb, D.O.: Textbook of Psychology, 3rd edn. W.B.Saunders Company (1972)
[30] Aspray, W.: John von Neumann and the origins of modern computing. MIT Press, Cambridge (1990)
[31] Future challenges in soft computing: Theory? computational paradigms? applications? In: International Conference on Neural Information Processing (ICONIP 2002), Singapore, Panel Discussion #1, Jacek M. Zurada, organizer, November 18-22 (2002)
[32] Ishida, F., Sawada, Y.E.: Human hand moves proactively to the external stimulus: An evolutional strategy for minimizing transient error. Physical Review Letters 93(16), 168105 (2004)
[33] Kohonen, T.: Self-Organization and Associative Memory, 3rd edn. Springer, Heidelberg (1989)
[34] Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (1995)
[35] Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan Coll. Div. (December 1994)
[36] Marr, D.: Vision – A Computational Investigation into the Human Representation and Processing of Visual Information. W.H.Freeman and Company, New York (1982)
[37] Mead, C.: Analog VLSI and Neural Systems. Addison-Wesley, Reading (1989)
[38] Ebbinghaus, H.-D., Hermes, H., Hirzebruch, F., Koecher, M., Mainzer, K., Neukirch, J., Prestel, A., Remmert, R.: Numbers (Chapter 3, Section 2). Springer, Heidelberg (1983)
[39] Copson, E.T.: An Introduction to the Theory of Functions of a Complex Variable. Clarendon Press, Oxford (1935)
[40] Hirose, A.: Nature of complex number and complex-valued neural networks. Frontiers of Electrical and Electronic Engineering in China 6(1), 171–180 (2011)
[41] Hirose, A., Yoshida, S.: Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. IEEE Transactions on Neural Networks and Learning Systems 23(4), 541–551 (2012)
[42] Picinbono, B., Chevalier, P.: Widely linear estimation with complex data. IEEE Transactions on Signal Processing 43(8), 2030–2033 (1995)

[43] Mandic, D.P., Goh, V.S.L.: Complex Valued Nonlinear Adaptive Filters – Noncircularity, Widely Linear and Neural Models. Wiley (April 2009)
[44] Xia, Y., Jelfs, B., Van Hulle, M.M., Principe, J.C., Mandic, D.P.: An augumented echo state network for nonlinear adaptive filetering of complex noncircular signals. IEEE Transactions on Neural Networks 22(1), 74–83 (2011)
[45] Mandic, D.P., Still, S., Douglas, S.C.: Duality between widely linear and dual channel adaptive filtering. In: IEEE Internatinal Conference on Acoustics, Speech, and Signal Processing 2009, Taipei, pp. 1729–1732 (2009)
[46] Widrow, B., McCool, J., Ball, M.: The complex lms algorithm. Proceedings of the IEEE 63, 719–720 (1975)
[47] Kuroe, Y., Taniguchi, Y.: Models of Orthogonal Type Complex-Valued Dynamic Associative Memories and their Performance Comparison. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 838–847. Springer, Heidelberg (2007)
[48] Leung, H., Haykin, S.: The complex backpropagation algorithm. IEEE Transactions on Signal Processing 39, 2101–2104 (1991)
[49] Benvenuto, N., Piazza, F.: On the complex backpropagation algorithm. IEEE Transactions on Signal Processing 40, 967–969 (1992)
[50] Hirose, A.: Applications of complex-valued neural networks to coherent optical computing using phase-sensitive detection scheme. Information Sciences –Applications 2, 103–117 (1994)
[51] Kawata, S., Hirose, A.: Frequency-multiplexed logic circuit based on a coherent optical neural network. Applied Optics 44(19), 4053–4059 (2005)
[52] Kawata, S., Hirose, A.: Frequency-multiplexing ability of complex-valued Hebbian learning in logic gates. International Journal of Neural Systems 12(1), 43–51 (2008)
[53] Hirose, A., Higo, T., Tanizawa, K.: Efficient generation of holographic movies with frame interpolation using a coherent neural network. IEICE Electronics Express 3(19), 417–423 (2006)
[54] Tay, C.S., Tanizawa, K., Hirose, A.: Error reduction in holographic movies using a hybrid learning method in coherent neural networks. Applied Optics 47(28), 5221–5228 (2008)
[55] Aoyagi, T., Radenamad, D., Nakano, Y., Hirose, A.: Complex-valued self-organizing map clustering using complex inner product in active mmillimeter-wave imaging. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2010), pp. 1346–1351. IEEE/INNS, Barcelona (2010)
[56] Hirose, A. (ed.): Complex-Valued Neural Networks: Theories and Applications. World Scientific Publishing Co. Pte. Ltd. (2003)
[57] Hirose, A.: Electrical and Electronic Measurement. Suurikogaku-sha, Tokyo (October 2003) (in Japanese)
[58] Okoshi, T., Kikuchi, K.: Coherent Optical Fiber Communications. KTK Scientific Publishers, Tokyo (1988)
[59] Hirose, A., Nagashima, T.: Predictive self-organizing map for vector quantization of migratory signals and its application to mobile communications. IEEE Transactions on Neural Networks 14(6), 1532–1540 (2003)
[60] Suksmono, A.B., Hirose, A.: Beamforming of ultra-wideband pulses by a complex-valued spatio-temporal multilayer neural network. International Journal of Neural Systems 15(1-2), 85–91 (2005)

[61] Hirose, A., Onishi, H.: Proposal of relative-minimization learning for behavior stabilization of complex-valued recurrent neural networks. Neurocomputing 24, 163–171 (1999)
[62] Kinouchi, M., Hagiwara, M.: Memorization of melodies by complex-valued recurrent network. In: IEEE International Conference on Neural Networks (ICNN 1996), Washington D.C., vol. 2, pp. 1324–1328. IEEE, New York (1996)
[63] Nitta, T.: On the inherent property of the decision boundary in complex-valued neural networks. Neurocomputing 50, 291–303 (2003)
[64] Nemoto, I., Saito, K.: A complex-valued version of nagumo-sato model of a single neuron and its behavior. Neural Networks 15, 833–853 (2002)
[65] Suksmono, A.B., Hirose, A.: Fractal estimation method to reduce the distortion in phase unwrapping process. IEICE Trans. on Commun. E88-B(1) (January 2005) (to be published)
[66] Takahasi, H., Goto, E.: Parametron counters. Paper of Technical Group of Electronic Computers, IECE (September 1955) (in Japanese)
[67] Kiyasu, Z.: The analogy between parametric oscillation and frequency demultiplication by feedback. Paper of Technical Group of Electronic Computers, IECE (September 1955) (in Japanese)
[68] Kiyasu, Z., Yamada, S.: Ultrasonic delay line memory for parametron computers. Paper of Technical Group of Electronic Computers, IECE (February 1956) (in Japanese)
[69] Takahasi, H.: An exerimental decimal calculator. Paper of Technical Group of Electronic Computers, IECE (March 1956) (in Japanese)
[70] Takahasi, H.: Parametron. Kagaku 26(3), 113–118 (1956) (in Japanese)
[71] Takahasi, H.: Dual-frequency parametron. Paper of Technical Group of Electronic Computers, IECE (May 1956) (in Japanese)
[72] Aizenberg, I.N., Aizenberg, N.N., Joos, V.: Multi-Valued and Universal Binary Neurons – Theory, Learning and Applications. Kluwer Academic Publishers, Boston (2000)
[73] Noest, A.J.: Discrete-state phasor neural networks. Physical Review A 38(4), 2196–2199 (1988)
[74] Noest, A.J.: Associative memory in sparse phasor neural networks. Europhysics Letters 6(6), 469–474 (1988)
[75] Nitta, T.: An extension of the back-propagation algorithm to complex numbers. Neural Networks 10, 1391–1415 (1997)
[76] Nitta, T.: Complex-valued backpropagation learning. Journal of Information Processing Society of Japan 32(10), 1319–1329 (1991) (in Japanese)
[77] Kim, M.S., Guest, C.C.: Modification of backpropagation networks for complex-valued signal processing in frequency domain. In: International Joint Conference on Neural Networks (IJCNN 1990), San Diego, pp. 27–31. IEEE, New York (1990)
[78] Birx, D.L., Pipenberg, S.J.: A complex mapping network for phase sensitive classification. IEEE Transactions on Neural Networks 4(1), 127–135 (1993)
[79] Georgiou, G.M., Koutsougeras, C.: Complex domain back propagation. IEEE Transactions on Circuits and Systems II 39(5), 330–334 (1992)
[80] Hirose, A.: Dynamics of fully complex-valued neural networks. Electronics Letters 28(16), 1492–1494 (1992)

[81] Hirose, A.: Continuous complex-valued back-propagation learning. Electronics Letters 28(20), 1854–1855 (1992)
[82] Takeda, M., Kishigami, T.: Complex neural fields with a hopfield-like energy function and an analogy to optical fields generated in phase-conjugate resonators. Journal of Optical Society of America A 9(12), 2182–2191 (1992)
[83] Hirose, A., Eckmiller, R.: Coherent optical neural networks that have optical-frequency-controlled behavior and generalization ability in the frequency domain. Applied Optics 35(5), 836–843 (1996)
[84] Hirose, A., Eckmiller, R.: Proposal of frequency-domain multiplexing in optical neural networks. Neurocomputing 10(2), 197–204 (1996)
[85] Hirose, A., Eckmiller, R.: Coherent optical neural networks and the generalization characteristics. Optical Review 3(6A), 418–422 (1996)
[86] Hirose, A., Eckmiller, R.: Behavior control of coherent-type neural networks by carrier-frequency modulation. IEEE Transactions on Neural Networks 7(4), 1032–1034 (1996)
[87] Hanna, A.I., Mandic, D.P.: A fully adaptive normalized nonlinear gradient descent algorithm for complex-valued nonlinear adaptive filter's. IEEE Transactions on Signal Processing 51, 2540–2549 (2003)
[88] Hanna, A.I., Mandic, D.P.: A complex-valued nonlinear neural adaptive filter with a gradient adaptive amplitude of the activation function. Neural Networks 16, 155–159 (2003)
[89] Hanna, A.I., Mandic, D.P.: A data-reusing nonlinear gradient descent algorithm for a class of complex-valued neural adaptive filters. Neural Processing Letters 17, 85–91 (2003)
[90] Mandic, D.P., Chambers, J.A.: Recurrent Neural Networks For Prediction – Learning Algorithms, Architectures and Stability. John Wiley & Sons (August 2001)
[91] Casasent, D., Natarajan, S.: A classifier neural net with complex-valued weights and square-law nonlinearities. Neu. 8(6), 989–998 (1995)
[92] Nitta, T.: An extension of the back-propagation algorithm to complex numbers. Neural Networks 10(8), 1391–1415 (1997)
[93] Nitta, T.: Solving the XOR problem and the detection of symmetry using a single complex-valued neuron. Neural Networks 16, 1101–1105 (2003)
[94] Goh, S.L., Mandic, D.P.: An augmented CRTRL for complex-valued recurrent neural networks. Neural Networks 20, 1061–1066 (2007)
[95] Goh, S.L., Mandic, D.P.: An augmented extended Kalman filter algorithm for complex-valued recurrent neural networks. Neural Computation 19, 1039–1055 (2007)
[96] Goh, S.L., Mandic, D.P.: A complex-valued rtrl algorithm for recurrent neural networks. Neural Computation 16, 2699–2713 (2004)
[97] Fiori, S.: Nonlinear complex-valued extensions of Hebbian learning: An essay. Neural Computation 17, 779–838 (2005)
[98] Xu, D., Zhang, H., Liu, L.: Convergence analysis of three classes of split-complex gradient algorithms for complex-valued recurrent neural networks. Neural Computation 22, 2655–2677 (2010)
[99] Yang, S.-S., Ho, C.-L., Siu, S.: Sensitivity analysis of the split-complex-valued multilayer perceptron due to the errors of the i.i.d. inputs and weights. IEEE Transactions on Neural Networks 18(5), 1280–1293 (2007)

[100] Kobayashi, M.: Exceptional reducibility of complex-valued neural networks. IEEE Transactions on Neural Networks 21(7), 1060–1072 (2010)
[101] Cao, Q., Gao, S., Zhang, J., Tang, Z., Kimura, H.: A stochastic dynamic local search method for learning multiple-valued logic networks. IEICE Transactions on Fundamentals E90-A(5), 1085–1092 (2007)
[102] Gao, S., Cao, Q., Ishii, M., Tang, Z.: Local search with probabilistic modeling for learning multiple-valued logic networks. IEICE Transactions on F E94-A(2), 795–805 (2011)
[103] Kim, T., Adali, T.: Approximation by fully complex multilayer perceptrons. Neural Computation 15, 1641–1666 (2003)
[104] Kim, T., Adali, T.: Fully complex multi-layer perceptron network for nonlinear signal processing. Journal of VLSI Signal Processing Systems for Signal Image and Video Technology 32, 29–43 (2002)
[105] Jankowski, S., Lozowski, A., Zurada, J.M.: Complex-valued multistate neural associative memory. IEEE Transactions on Neural Networks 7(6), 1491–1496 (1996)
[106] Kuroe, Y., Hashimoto, N., Mori, T.: Qualitative analysis of a self-correlation type complex-valued associative memories. Nonlinear Analysis-Theory Methods & Applications 47, 5795–5806 (2001)
[107] Hirose, A.: Complex-valued neural networks utilizing asymptotic transitions. Journal of Artificial Neural Networks 2(1&2), 97–105 (1995)
[108] Nemoto, I., Kubono, M.: Complex associative memory. Neural Networks 9, 253–261 (1996)
[109] Lee, D.-L., Wang, W.-J.: A mutivalued bidirectional associative memory operating on a complex domain. Neural Networks 11, 1623–1635 (1998)
[110] Aoki, H., Kosugi, Y.: Characteristics of the complex-valued associative memory model having penalty term. Electronics and Communications in Japan Part III – Fundamental Electronic Science 83, 62–70 (2000)
[111] Lee, D.L.: Relaxation of the stability condition of the complex-valued neural networks. IEEE Transactions on Neural Networks 12(5), 1260–1262 (2001)
[112] Lee, L.D.: Improving the capacity of complex-valued neural networks with a modified gradient descent learning rule. IEEE Transactions on Neural Networks 12, 439–443 (2001)
[113] Lee, D.-L.: Improvements of complex-valued Hopfield associative memory by using generalized projection rules. IEEE Transactions on Neural Networks 17(5), 1341–1347 (2006)
[114] Prashanth, A.: Investigation on complex-variable based backpropagation algorithm and applications. PhD thesis, Department of Electrical Engineering, Indian Institute of Technology Kanpur (2003)
[115] Takahashi, H.: Covariance phasor neural network as a mean field model. In: International Conference on Neural Information Processing (ICONIP 2002), Singapore, vol. 3, pp. 1089–1093. IEEE, New York (2002)
[116] Müezzinoğlu, M.K., Güzeliş, C., Zurada, J.M.: A new design method for the complex-valued multistate hopfield associative memory. IEEE Transactions on Neural Networks 14(4), 891–899 (2003)
[117] Agu, M., Yamanaka, K., Takahashi, H.: A local property of the phasor model of neural networks. IEICE Transactions on Information and Systems E79D, 1209–1211 (1996)

[118] Hirose, A.: Abrupt variations of attractors caused by argumental discreteness in non-Hermitian associative memories. IEICE Transactions on Fundamentals E76-A(5), 777–779 (1993)
[119] Nitta, T.: An analysis of the fundamental structure of complex-valued neurons. Neural Processing Letters 12, 239–246 (2000)
[120] Nitta, T.: Orthogonality of decision boundaries in complex-valued neural networks. Neural Computation 16, 73–97 (2004)
[121] Zhang, Y., Ma, Y.: CGHA for principal component extraction in the complex domain. IEEE Transactions on Neural Networks 8, 1031–1036 (1997)
[122] Sawada, H., Mukai, R., Araki, S., Makino, S.: Polar coordinate based nonlinear function for frequency-domain blind source separation. IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences E86A, 590–596 (2003)
[123] Makino, S., Sawada, H., Mukai, R., Araki, S.: Blind source separation of convolutive mixtures of speech in frequency domain. IEICE E88-A(7), 1640–1654 (2005)
[124] Rattan, S.S.P., Hsieh, W.W.: Complex-valued neural networks for nonlinear complex principal component analysis. Neural Networks 18, 61–69 (2005)
[125] Uncini, A., Piazza, F.: Blind signal processing by complex domain adaptive spline neural networks. IEEE Transactions on Neural Networks 14(2), 399–412 (2003)
[126] Anemuller, J., Sejnowski, T.J., Makeig, S.: Complex independent component analysis of frequency-domain electroencephalographic data. Neural Networks 16, 1311–1323 (2003)
[127] Li, H., Adali, T.: A class of complex ICA algorithms based on the kurtosis cost function. IEEE Transactions on Neural Networks 19(3), 408–420 (2008)
[128] Li, X.-L., Adali, T.: Complex independent component analysis by entropy bound minimization. IEEE Transactions on Circuits and Systems – 1: Regular Papers 57(7), 1417–1430 (2010)
[129] Novey, M., Adali, T.: Complex ICA by negentropy maximization. IEEE Transactions on Neural Networks 19(4), 596–609 (2008)
[130] Burwick, T.: Oscillatory networks: Pattern recognition without a superposition catastrophe. Neural Computation 18, 356–380 (2006)
[131] Burwick, T.: Oscillatory neural networks with self-organizing segmentation of overlapping patterns. Neural Computation 19, 2093–2123 (2007)
[132] Pearson, J.K., Bisset, D.L.: Neural networks in the clifford domain. In: IEEE World Congress on Computatinal Intelligence (WCCI)/IEEE International Conference on Neural Networks (ICNN 1994), Orlando, vol. 3, pp. 1465–1469. IEEE, New York (1994)
[133] Arena, P., Fortuna, L., Muscato, G., Xibilia, M.G.: Multipayer perceptrons to approximate quaternion valued functions. Neural Networks 10, 335–342 (1997)
[134] Isokawa, T., Kusakabe, T., Matsui, N., Peper, F.: Quaternion Neural Network and its Application. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS, vol. 2774, pp. 318–324. Springer, Heidelberg (2003)
[135] Kusamichi, H., Isokawa, T., Matsui, N., Ogawa, Y., Maeda, K.: A new scheme for color night vision by quaternion neural network. In: Proceeding of the 2nd International Conference on Autonomous Robots and Agents (ICARA 2004), Palmerston North, NewZealand, December 13-15, pp. 101–106 (2004)

[136] Buchholz, S., Sommer, G.: Quaternionic spinor MLP. In: European Symposium on Artificial Neural Networks (ESANN 2000), Bruges (2000)
[137] Buchholz, S.: A Theory of Neural Computation with Clifford Algebras. PhD thesis, Christian-Albrechts-Universität Kiel (May 2005)
[138] Bayro-Corrochano, E.J., Arana-Daniel, N.: Clifford support vector machines for classification, regression, and recurrence. IEEE Transactions on Neural Networks 21(11), 1731–1746 (2010)
[139] Mishra, R.K., Patnaik, A.: Neurospectral computation for input impedance of rectangular microstrip antenna. Electronics Letters 35, 1691–1693 (1999)
[140] Du, K.L., Lai, A.K.Y., Cheng, K.K.M., Swamy, M.N.S.: Neural methods for antenna array signal processing: A review. Signal Processing 82, 547–561 (2002)
[141] Mishra, R.K., Patnaik, A.: Designing rectangular patch antenna using the neurospectral method. IEEE Transactions on Antennas and Propagation 51, 1914–1921 (2003)
[142] Yang, W.H., Chan, K.K., Chang, P.R.: Complex-valued neural-network for direction-of-arrival estimation. Electronics Letters 30, 574–575 (1994)
[143] Yang, C.-C., Bose, N.K.: Landmine detection and classification with complex-valued hybrid neural network using scattering parameters dataset. IEEE Transactions on Neural Networks 16(3), 743–753 (2005)
[144] Suksmono, A.B., Hirose, A.: Progressive transform-based phase unwrapping utilizing a recursive structure. IEICE Transactions on Communications E89-B(3), 929–936 (2006)
[145] Yamaki, R., Hirose, A.: Singularity-spreading phase unwrapping. IEEE Transactions on Geoscience and Remote Sensing 45(10), 3240–3251 (2007)
[146] Yamaki, R., Hirose, A.: Singular unit restoration in interferograms based on complex-valued Markov random field model for phase unwrapping. IEEE Geoscience and Remote Sensing Letters 6(1), 18–22 (2009)
[147] Chang, A.-C., Jen, C.-W., Su, I.-J.: Robust adaptive array beamforming based on independent component analysis with regularized constraints. IEICE E90-B(7), 1791–1800 (2007)
[148] Hirose, A., Minami, M.: Complex-valued region-based-coupling image clustering neural networks for interferometric radar image processing. IEICE Trans. Electron E84-C(12), 1932–1938 (2001)
[149] Hara, T., Hirose, A.: Plastic mine detecting radar system using complex-valued self-organizing map that deals with multiple-frequency interferometric images. Neural Networks 17(8-9), 1201–1210 (2004)
[150] Hara, T., Hirose, A.: Adaptive plastic-landmine visualizing radar system: effects of aperture synthesis and feature-vector dimension reduction. IEICE Transactions on Electronics E88-C(12), 2282–2288 (2005)
[151] Masuyama, S., Hirose, A.: Walled LTSA array for rapid, high spatial resolution, and phase sensitive imaging to visualize plastic landmines. IEEE Transactions on Geoscience and Remote Sensing 45(8), 2536–2543 (2007)
[152] Masuyama, S., Yasuda, K., Hirose, A.: Multiple mode selection of walled-ltsa array elements for high resolution imaging to visualize antipersonnel plastic landmines. IEEE Geoscience and Remote Sensing Letters 5(4), 745–749 (2008)

[153] Nakano, Y., Hirose, A.: Improvement of plastic landmine visualization performance by use of ring-CSOM and frequency-domain local correlation. IEICE Transactions on Electronics E92-C(1), 102–108 (2009)
[154] Nakano, Y., Hirose, A.: Adaptive identification of landmine class by evaluating the total degree of conformity of ring-SOM. Australian Journal of Intelligent Information Processing Systems 12, 23–28 (2010)
[155] Nishino, T., Yamaki, R., Hirose, A.: Ultrasonic imaging for boundary shape generation by phase unwrapping with singular-point elimination based on complex-valued Markov random field model. IEICE Tarnsactions on Fundamentals E93-A(1), 219–226 (2010)
[156] Aizenberg, I., Myasnikova, E., Samsonova, M., Reinitz, J.: Temporal classification of drosophila segmentation gene expression patterns by the multi-valued neural recognition method. Journal of Mathematical Biosciences 176(1), 145–159 (2002)
[157] Handayani, A., Suksmono, A.B., Mengko, T.L.R., Hirose, A.: Blood vessel segmentation in complex-valued magnetic resonance images with snake active contour model. International Journal of E-Health and Medical Communications 1, 41–52 (2010)
[158] Aoki, H., Azimi-Sadjadi, M.R., Kosugi, Y.: Image association using a complex-valued associative memory model. IEICE Transactions on Fundamentals of Electronics, Communications, and Computer Sciences E83A, 1824–1832 (2000)
[159] Aizenberg, I., Butakoff, C.: Image processing using cellular neural networks based on multi-valued and universal binary neurons. Journal of VLSI Signal Processing Systems for Signal Image and Video Technology 32(1-2), 169–188 (2003); erratum: 34(3), 301 (2003)
[160] Aizenberg, I., Paliy, D., Zurada, J.M., Astola, J.T.: Blur identification by multilayer neural network based on mutivalued neurons. IEEE Transactions on Neural Networks 19(5), 883–897 (2008)
[161] Tanaka, G., Aihara, K.: Complex-valued multistate associative memory with nonlinear multilevel functions for gray-level image reconstruction. IEEE Transactions on Neural Networks 20(9), 1463–1473 (2009)
[162] Miyajima, T., Baisho, F., Yamanaka, K., Nakamura, K., Agu, M.: A phasor model with resting states. IEICE Transactions on Information and Systems E83D, 299–301 (2000)
[163] Martin, K.W.: Complex signal processing is not complex. IEEE Transactions on Circuits and Systems – 1: Regular Papers 51(9), 1823 (2004)
[164] You, C., Hong, D.: Nonlinear blind equalization schemes using complex-valued muptilayer feedforward neural networks. IEEE Transactions on Neural Networks 9, 1442–1455 (1998)
[165] Gan, Q., Saratchandran, P., Sundararajan, N., Subramanian, K.R.: A complex valued radial basis function network for equalization of fast time varying channels. IEEE Transactions on Neural Networks 10(4), 958–960 (1999)
[166] Wang, X., Lin, H., Lu, J., Yahagi, T.: Combining recurrent neural networks with self-organizing map for channel equalization. IEICE Transactions on Communications E85-B(10), 2227–2235 (2002)
[167] Jianping, D., Sundararajan, N., Saratchandra, P.: Communication channel equalization using complex-valued minimal radial basis fuction neural networks. IEEE Transactions on Neural Networks 13(3), 687–696 (2002)

[168] Park, D.-C., Jeong, T.-K.J.: Complex-bilinear recurrent neural networks for equilization of a digital satellite channel. IEEE Transactions on Neural Networks 13(3), 711–725 (2002)
[169] Kawamoto, M., Inouye, Y.: Blind deconvolution of MIMO-FIR systems with colored inputs using second-order statistics. IEICE Transactions on Fundamentals E86-A(3), 597–604 (2003)
[170] Koike, S., Noda, S.: Pre-compensation of transmitter nonlinearity with memory effects in digital QAM systems. IEICE Transactions on Fundamentals E87-A(10), 2744–2754 (2004)
[171] Deng, Y., Yang, Z.: Comments on "complex bilinear recurrent neural network for equalization of a digital satellite channel". IEEE Transactions on Neural Networks 17(1), 268 (2006)
[172] Chen, S., Hanzo, L., Tan, S.: Symmetric complex-valued RBF receiver for muptiple-antenna-aided wireless systems. IEEE Transactions on Neural Networks 19(9), 1657–1663 (2008)
[173] Chakravarthy, S.V., Ghosh, J.: A complex-valued associative memory for storing patterns as oscillatory states. Biol. Cybern. 75, 229–238 (1996)
[174] Hirose, A.: Fractal variation of attractors in complex-valued neural networks. Neural Processing Letters 1(1), 6–8 (1994)
[175] Hirose, A., Onishi, H.: Observation of positive lyapunov exponent induced by gain increase of neuron nonlinearity in complex-valued associative memories. Electronics Letters 32(8), 745–746 (1996)
[176] Rajagopal, P., Kak, S.: Instantaneously trained neural networks with complex inputs. In: Hirose, A. (ed.) Complex-Valued Neural Networks –Theories and Applications, pp. 157–179. World Scientific Publishing, Singapore (2003)
[177] Nishikawa, I., Kuroe, Y.: Dynamics of Complex-Valued Neural Networks and its Relation to a Phase Oscillator System. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 122–129. Springer, Heidelberg (2004)
[178] Song, J.Y., Yam, Y.: Complex recurrent neural network for computing the inverse and pseudo-inverse of the complex matrix. App. Math. Comput. 93, 195–205 (1998)
[179] Hirose, A., Nakazawa, K.: Analog recurrent decision circuit with high signal-voltage symmetry and delay-time equality to improve continuous-time convergence performance. IEEE Transactions on Neural Networks 14(5), 1201–1206 (2003)
[180] Kinjo, M., Sato, S., Nakajima, K.: Quantum Adiabatic Evolution Algorithm for a Quantum Neural Network. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 951–958. Springer, Heidelberg (2003)
[181] Sato, S., Kinjo, M., Nakajima, K.: An approach for quantum computing using adiabatic evolution algorithm. Jpn. J. Appl. Phys. 1, Regul. Pap. Short Notes 42(11), 7169–7173 (2003)
[182] Kouda, N., Matsui, N., Nishimura, H., Peper, F.: Qubit Neural Network and its Efficiency. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS, vol. 2774, pp. 304–310. Springer, Heidelberg (2003)
[183] Kinjo, M., Sato, S., Nakamiya, Y., Nakajima, K.: Neuromorphic quantum computation with energy dissipation. Physical Review A 72, 052328 (2005)

[184] Nakamiya, Y., Kinjo, M., Takahashi, O., Sato, S., Nakajima, K.: Quantum neural network composed of Kane's qubits. Japanese Journal of Applied Physics 45(10A), 8030–8034 (2006)
[185] Georgiou, G.M.: Book Review on A.Hirose: Complex-valued neural networks. IEEE Transactions on Neural Networks 19(3), 3 (2008)
[186] Hirose, A.: Complex-Valued Neural Networks, 1st edn. Springer, Heidelberg (2006)
[187] Aizenberg, I.: Complex-Valued Neural Networks with Multi-Valued Neurons. Springer, Heidelberg (2011)
[188] Aizenberg, I.: Book Review on A.Hirose: Complex-valued neural networks: Theories and applications. IEEE Transactions on Neural Networks 17(2), 534 (2006)
[189] Nitta, T. (ed.): Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters. IGI-Global (2009)
[190] Hirose, A. (ed.): Complex-Valued Neural Networks: Advances and Applications. The IEEE Press Series on Computational Intelligence. IEEE / Wiley (2012)
[191] Limmanee, A., Kawata, S., Hirose, A.: Phase signal embedment in densely frequency-multiplexed coherent neural networks. In: OSA Topical Meeting on Information Photonics (OSA–IP 2005), Charlotte, number ITuA2 (June 2005)
[192] Kawata, S.: Learning Lightwave Information Processing Systems with Adaptability in Optical Frequency Domain. Ph.D. Thesis, The University of Tokyo (2005) (in Japanese)
[193] Hirose, A.: Coherent neural networks and their applications to control and signal processing. In: Tzafestas, S.G. (ed.) Soft Computing in Systems and Control Technology. World Scientific Series in Robotics and Intelligent Systems, vol. 18, pp. 397–422 (May 1999)
[194] Suksmono, A.B., Hirose, A.: Adaptive noise reduction of insar image based on complex-valued mrf model and its application to phase unwrapping problem. IEEE Trans. on Geoscience and Remote Sensing 40(3), 699–709 (2002) (followed by publisher's errata on Fig.12)
[195] Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley-Interscience (May 2001)
[196] Lee, T.-W.: Independent Component Analysis – Theory and Applications. Kluwer Academic Publishers (November 1998)
[197] Suksmono, A.B., Hirose, A.: Adaptive complex-amplitude texture classifier that deals with both height and reflectance for interferometric SAR images. IEICE Trans. on Electron. E83-C(12), 1912–1916 (2000)
[198] Suksmono, A.B.: Adaptive Processing of Interferometric Radar Images. PhD thesis, The University of Tokyo (2002)
[199] Hirose, A., Hara, T.: Complex-valued self-organizing map dealing with multi-frequency interferometeric data for radar imaging systems. In: International Workshop on Self-Organizing Map (WSOM 2003), Hibikino, pp. 255–260 (September 2003)
[200] Hirose, A.: Ground penetrating radar technology to visualize landmines (review). Journal of SICE 48(10), 754–760 (2009) (in Japanese)

[201] Hirose, A., Hara, T.: Complex-valued self-organizing map: A framework of adaptive processing for multiple-frequency millimeter-wave interferometric imaging systems. In: Arai, T., Yamamoto, S., Makino, K. (eds.) System and Human Science – For Safety, Security and Dependability, pp. 299–308. Elsevier B.V., Amsterdam (2004)
[202] Hirose, A., Sugiyama, K.: A radar system with phase-sensitive millimeter-wave circuitry and complex-amplitude neural processing. In: Proc. of Int'l Conf. on Artificial Neural Netw. (ICANN 1998), vol. 2, pp. 707–712. ENNS (September 1998)
[203] Suksmono, A.B., Hirose, A.: Interferometric sar image restoration using monte-carlo metropolis method. IEEE Trans. on Signal Processing 50(2), 290–298 (2002)
[204] Kawata, S., Hirose, A.: Coherent lightwave associative memory system that possesses a carrier-frequency-controlled behavior. Optical Engineering 42(9), 2670–2675 (2003)
[205] Hirose, A., Kiuchi, M.: Coherent optical associative memory system that processes complex-amplitude information. IEEE Photon. Tech. Lett. 12(5), 564–566 (2000)
[206] Weverka, R.T., Wagner, K., Saffman, M.: Fully interconnected, two-dimensional neural arrays using wavelength-multiplexed volume holograms. Opt. Lett. 16(11), 826–828 (1991)
[207] Mos, E.C., Schkeipen, J.J.H.B., de Waardt, H.: Optical-mode neural network by use of the nonlinear response of a laser diode to external optical feedback. Appl. Opt. 36(26), 6654–6663 (1997)
[208] Hirose, A., Tabata, C., Ishimaru, D.: Coherent neural network architecture realizing a self-organizing activeness mechanism. In: Baba, N. (ed.) International Conference on Knowledge-Based Intelligent Information Engineering System & Applied Technologies (KES 2001), Osaka, pp. 576–580. Springer, Berlin (2001)
[209] Kawata, S., Hirose, A.: A coherent optical neural network that learns desirable phase values in frequency domain by using multiple optical-path differences. Optics Letters 28(24), 2524–2526 (2003)
[210] Fujimoto, T., Terai, M., Gogo, N., Yanagiya, S.-I.: All-optical label recognition and classification using complex-valued neural network. In: Opto-Electronics and Communications Conference (OECC)/Australian Conference on Optical Fibre Technology (ACOFT 2008), Sydney, number ThK-4 (July 2008)
[211] Asada, M., MacDorman, K.F., Ishiguro, H., Kuniyoshi, Y.: Cognitive developmental robotics as a new paradigm for the design of humanoid robots. Robotics and Autonomous Systems 37, 185–193 (2001)
[212] Narendra, K.S., Balakrishnan, J., Ciliz, M.K.: Adaptation and learning using multiple models, switching and tuning. IEEE Control Systems Magazine, 37–51 (1995)
[213] Omori, T., Mochizuki, A., Mizutani, K.: Emergence of symbolic behavior from brain like memory with dynamic attention. Neural Networks 12, 1157–1172 (1999)
[214] Omori, T., Mochizuki, A.: Rethinking innateness: A connectionist perspective on development. In: Brain Processes, Theories and Models, pp. 134–143. MIT Press, Cambridge (1995)

[215] Wolpert, D.M., Kawato, M.: Multiple paired fotward and inverse models for motor control. Neural Networks 11, 1317–1329 (1998)
[216] Wolpert, D.M., Miall, C.M., Kawato, M.: Internal models in the cerebellum. Trends in Cognitive Sciences 2(9), 79–87 (1998)
[217] Hartono, P., Hashimoto, S.: Temperature switching in neural network ensemble. Journal of Signal Processing 4, 395–402 (2000)
[218] Hirose, A., Ishimaru, D.: Context-Dependent Behavior of Coherent Neural Systems Based on Self-Organizing Mapping of Carrier Frequency Values. In: International Conference on Knowledge-Based Intelligent Information Engineering System & Applied Technologies (KES 2002), Crema, pp. 638–642. Springer, Berlin (2002)
[219] Hirose, A., Asano, Y., Hamano, T.: Mode-Utilizing Developmental Learning Based on Coherent Neural Networks. In: Pal, N.R., Kasabov, N., Mudi, R.K., Pal, S., Parui, S.K. (eds.) ICONIP 2004. LNCS, vol. 3316, pp. 116–121. Springer, Heidelberg (2004)
[220] Hirose, A., Asano, Y., Hamano, T.: Developmental learning with behavioral mode tuning by carrier-frequency modulation in coherent neural networks. IEEE Transactions on Neural Networks 17(6), 1532–1543 (2006)
[221] Charpentier, F.J., Stella, M.G.: Diphone synthesis using an overlap-add technique for speech waveforms concatenation. In: ICASSP 1986, pp. 2015–2018 (1986)
[222] Tsuda, K., Hirose, A.: Pitch-Asynchronous Overlap-Add Wveform-Concatenation Speech Synthesis by Using a Phase-Optimizing Neural Network. In: Palade, V., Howlett, R.J., Jain, L. (eds.) KES 2003. LNCS, vol. 2774, pp. 332–339. Springer, Heidelberg (2003)
[223] Aizenberg, I., Bregin, T., Butakoff, C., Karnaukhov, V., Merzlyakov, N., Milukova, O.: Type of Blur and Blur Parameters Identification Using Neural Network and its Application to Image Restoration. In: Dorronsoro, J.R. (ed.) ICANN 2002. LNCS, vol. 2415, pp. 1231–1236. Springer, Heidelberg (2002)
[224] Hamagami, T.: Speech synthesis using source wave shape modification technique by harmonic phase control. Journal of Acoustical Society of Japan 54(9), 623–631 (1998) (in Japanese)

Index