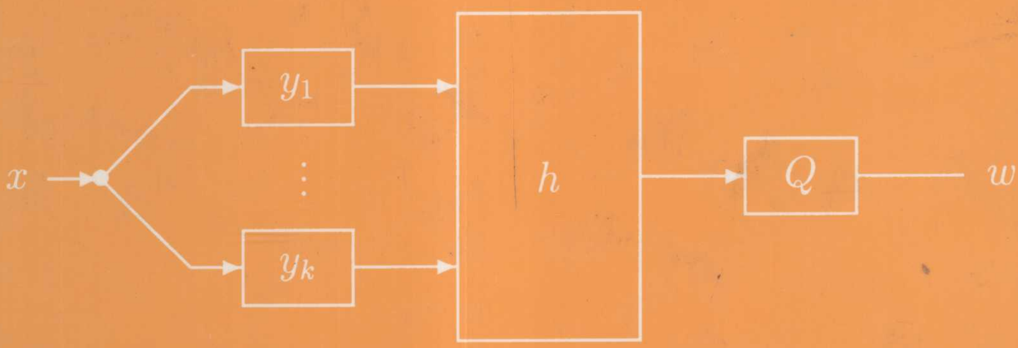
A Volume in the Wiley Series on Adaptive and Learning Systems for Signal Processing, Communications, and Control
Simon Haykin, Series Editor

Nonlinear Dynamical Systems

Feedforward Neural Network Perspectives



IRWIN W. SANDBERG, JAMES T. LO,
CRAIG L. FANCOURT, JOSE C. PRINCIPE,
SHIGERU KATAGIRI, and SIMON HAYKIN

NONLINEAR DYNAMICAL SYSTEMS

Feedforward Neural Network Perspectives

Irwin W. Sandberg
James T. Lo
Craig L. Fancourt
Jose C. Principe
Shigeru Katagiri
Simon Haykin

A WILEY-INTERSCIENCE PUBLICATION
JOHN WILEY & SONS, Inc.
New York / Chichester / Weinheim / Brisbane / Singapore / Toronto

This book is printed on acid-free paper. ♾

Published simultaneously in Canada.

Library of Congress Cataloging-in-Publication Data:

Nonlinear dynamical systems: feedforward neural network perspectives / Irwin Sandberg . . . [et al.].
p. cm.
Includes bibliographical references and index.
ISBN 0-471-34911-9 (cloth: alk. paper)
1. Neural networks (Computer science) 2. Dynamics. I. Sandberg, Irwin.
QA76.87.N56 2001
006.3′2—dc21

00–043384

Printed in the United Kingdom.
10 9 8 7 6 5 4 3 2 1

Adaptive and Learning Systems for Signal Processing, Communications, and Control

Editor: Simon Haykin

Beckerman / ADAPTIVE COOPERATIVE SYSTEMS

Chen and Gu / CONTROL-ORIENTED SYSTEM IDENTIFICATION: An $\mathcal{H}_\infty$ Approach

Cherkassky and Mulier / LEARNING FROM DATA: Concepts, Theory, and Methods

Diamantaras and Kung / PRINCIPAL COMPONENT NEURAL NETWORKS: Theory and Applications

Haykin / UNSUPERVISED ADAPTIVE FILTERING: Blind Source Separation

Haykin / UNSUPERVISED ADAPTIVE FILTERING: Blind Deconvolution

Haykin and Puthussarypady / CHAOTIC DYNAMICS OF SEA CLUTTER

Hrycej / NEUROCONTROL: Towards an Industrial Control Methodology

Kristić, Kanellakopoulos, and Kokotović / NONLINEAR AND ADAPTIVE CONTROL DESIGN

Nikias and Shao / SIGNAL PROCESSING WITH ALPHA-STABLE DISTRIBUTIONS AND APPLICATIONS

Passino and Burgess / STABILITY ANALYSIS OF DISCRETE EVENT SYSTEMS

Sánchez-Peña and Sznaler / ROBUST SYSTEMS THEORY AND APPLICATIONS

Sandberg, Lo, Fancourt, Principe, Katagiri, and Haykin / Nonlinear Dynamical Systems: Feedforward Neural Network Perspectives

Tao and Kokotović / ADAPTIVE CONTROL OF SYSTEMS WITH ACTUATOR AND SENSOR NONLINEARITIES

Tsoukalas and Uhrig / FUZZY AND NEURAL APPROACHES IN ENGINEERING

Van Hulle / FAITHFUL REPRESENTATIONS AND TOPOGRAPHIC MAPS: From Distortion- to Information-Based Self-Organization

Vapnik / STATISTICAL LEARNING THEORY

Werbos / THE ROOTS OF BACKPROPAGATION: From Ordered Derivatives to Neural Networks and Political Forecasting

NONLINEAR DYNAMICAL SYSTEMS

PREFACE

Feedforward neural networks have established themselves as an important part of the rapidly expanding field of artificial neural networks. This book *Nonlinear Dynamical Systems: Feedforward Neural Network Perspectives* addresses fundamental aspects and practical applications of the subject. To the best of the authors' knowledge, this is the first book to be published in this area.

Chapter 1 provides an introductory treatment of the different aspects of feedforward neural networks, thereby setting the stage for more detailed treatment of the subject matter in the succeeding four chapters.

Chapter 2 is concerned with classification problems and with the related problem of approximating dynamic nonlinear input–output maps. Attention is focused on the properties of nonlinear structures that have the form of a dynamic preprocessing stage followed by a memoryless nonlinear section. It is shown that an important type of classification problem can be solved using certain simple network structures involving linear functionals and memoryless nonlinear elements. The chapter addresses various aspects of the problem of approximating nonlinear input–output maps. One main result given is a theorem showing that if the maps to be approximated satisfy a certain "myopicity" condition, which is very often met, then they can be uniformly approximated arbitrarily well using a structure consisting of a linear preprocessing stage followed by a memoryless nonlinear network. Noncausal as well as causal maps are considered. (Approximations for noncausal maps are of interest in connection with image processing.) In the course of the study on which the material of this chapter is based, some interesting unexpected side issues arose. One such issue is discussed in an appendix where, in connection with the study of myopic maps, attention is focused on—and a correction is given of—a longstanding oversight concerning the cornerstone of digital signal processing. The chapter makes use of concepts drawn from the areas of real analysis and functional analysis.

Chapter 3 relates to one of the major research areas in the past 15 years, which pertains to the development of robust controllers and filters. Robust controllers and filters are intended to avert disastrous results or mitigate worst-case performances. The H-infinity norms, minimax errors, and risk-sensitive functionals are the main criteria used to induce robust performance and are proven to lead to the same robust controllers and filters for linear systems. Extending such results to nonlinear systems by the conventional analytic approach also has been a topic of extensive research. Dynamic programming equations characterizing the robust controllers and filters have been obtained. However, these equations are difficult, if not impossible, to use to derive practically useful results. In Chapter 3, the capabilities of neural networks to approximate functions and dynamic systems to any accuracy with respect to risk-sensitive error specifically are discussed. It is shown that under mild conditions, a function can be approximated, to any desired degree of accuracy with respect to a general risk-sensitive criterion, by a multilayer perceptron or a radial-basis function network. It is also shown that under relatively mild regularity conditions, dynamic systems can be approximated (or identified) by neural networks to any desired degree of accuracy with respect to general risk-sensitive criteria in both the series-parallel and parallel formulations. These capabilities of neural networks for universal risk-sensitive approximations of functions and dynamic systems qualify neural networks as powerful vehicles in a synthetic approach to robust processing (e.g., signal processing, communication, and control).

Chapter 4 discusses the practical issue of segmenting a time series. In this context, we note that many of the methods for detecting abrupt statistical changes in time series evolved in the field of statistical quality control in the 1950s. In fact, much of the nomenclature still reflects this origin. For example, in the statistics and control literature, time-series classification is often referred to as "isolation," which is short for "fault isolation." At that time, the emphasis was on the simpler problem of detecting changes in the moment(s) of an independent process, such as the dimension of a part coming off an assembly line. As digital signal processing advanced in the 1970s, the change detection methods were extended to include processes with memory. However, such methods generally still utilized linear models. In the 1990s, with the advent of powerful and general nonlinear modeling techniques, such as neural networks, new multiple-model algorithms appeared for modeling nonlinear but piecewise stationary time series. These algorithms were even able to model switching chaotic signals but seemingly had no connection with the prior work in quality control. Thus, the goal in Chapter 4 is twofold. From one side, we reexamine the classical methods for modeling piecewise stationary signals with an eye toward integration with new nonlinear models. We then push these algorithms into the realm of chaotic signals and examine whether they still function as before and why. From the other side, we put many of the new algorithms into a common framework and show their connections with the classical theory.

Finally, Chapter 5 deals with the application of feedforward neural networks to speech processing. A speech signal is the most fundamental communication medium, and it is also a typical example of dynamic (temporal and nonstationary) and nonlinear signals, which are usually difficult to handle in traditional system frameworks. To alleviate such difficulty, extensive research efforts have been expended on the application of feedforward networks to speech processing. Specifically, Chapter 5 starts by summarizing speech-related techniques and reviewing feedforward neural networks from the viewpoint of fundamental design issues such as the selection of network structures and the selection of training objective functions. We specially feature the recent design framework called the generalized probabilistic descent method in order to provide a comprehensive perspective about the issues involved in speech processing. We discuss the topic of speech recognition, to which feedforward neural networks have been most extensively applied. Other topics are summarized in an archived form. Through considerations of design fundamentals and application examples, the reader is enabled to understand the key points in the design of feedforward neural-network-based speech processing systems, the importance of special mechanism of shift tolerance and state transition.

The idea to write this volume came from Simon Haykin who selected the authors.

IRWIN W. SANDBERG
JAMES T. LO
CRAIG L. FANCOURT
JOSE C. PRINCIPE
SHIGERU KATAGIRI
SIMON HAYKIN

August 2000

CONTENTS

1

FEEDFORWARD NEURAL NETWORKS: AN INTRODUCTION

Simon Haykin

A *neural network* is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects (Haykin 1998):

1. Knowledge is acquired by the network through a learning process.
2. Interconnection strengths known as synaptic weights are used to store the knowledge.

Basically, learning is a process by which the free parameters (i.e., synaptic weights and bias levels) of a neural network are adapted through a continuing process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place. In a general sense, the learning process may be classified as follows:

- Learning with a teacher, also referred to as supervised learning
- Learning without a teacher, also referred to as unsupervised learning

1.1 SUPERVISED LEARNING

This form of learning assumes the availability of a labeled (i.e., ground-truthed) set of training data made up of N input—output examples:

$$T = \{(\mathbf{x}_i, d_i)\}_{i=1}^{N} \tag{1.1}$$

where $\mathbf{x}_i$ = input vector of ith example
d_i = desired (target) response of ith example, assumed to be scalar for convenience of presentation
N = sample size

Given the training sample T, the requirement is to compute the free parameters of the neural network so that the actual output y_i of the neural network due to $\mathbf{x}_i$ is close enough to d_i for all i in a statistical sense. For example, we may use the mean-square error

$$E(n) = \frac{1}{N} \sum_{i=1}^{N} (d_i - y_i)^2 \tag{1.2}$$

as the index of performance to be minimized.

1.1.1 Multilayer Perceptrons and Back-Propagation Learning

The back-propagation algorithm has emerged as the workhorse for the design of a special class of layered feedforward networks known as *multilayer perceptrons* (MLP). As shown in Fig. 1.1, a multilayer perceptron has an input layer of source nodes and an output layer of neurons (i.e., computation nodes); these two layers connect the network to the outside world. In addition to these two layers, the multilayer perceptron usually has one or more layers of hidden neurons, which are so called because these neurons are not directly accessible. The hidden neurons extract important features contained in the input data.

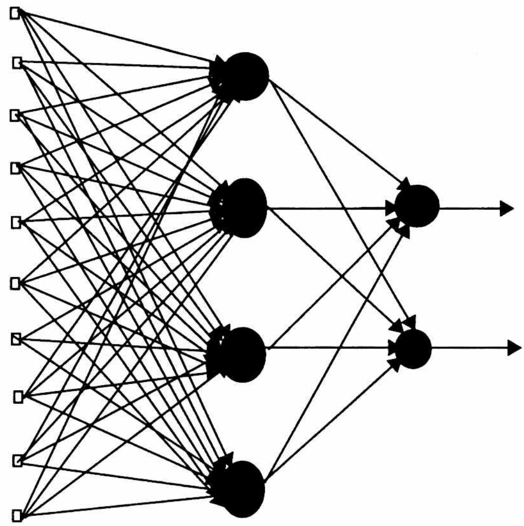


Figure 1.1 Fully connected feedforward with one hidden layer and one output layer.

The training of an MLP is usually accomplished by using a *back-propagation (BP) algorithm* that involves two phases (Werbos 1974; Rumelhart et al. 1986):

- *Forward Phase.* During this phase the free parameters of the network are fixed, and the input signal is propagated through the network of Fig. 1.1 layer by layer. The forward phase finishes with the computation of an error signal

$$e_i = d_i - y_i \tag{1.3}$$

  where d_i is the desired response and y_i is the actual output produced by the network in response to the input $\mathbf{x}_i$.
- *Backward Phase.* During this second phase, the error signal e_i is propagated through the network of Fig. 1.1 in the backward direction, hence the name of the algorithm. It is during this phase that adjustments are applied to the free parameters of the network so as to minimize the error e_i in a statistical sense.

Back-propagation learning may be implemented in one of two basic ways, as summarized here:

1. *Sequential mode* (also referred to as the on-line mode or stochastic mode): In this mode of BP learning, adjustments are made to the free parameters of the network on an example-by-example basis. The sequential mode is best suited for pattern classification.
2. *Batch mode*: In this second mode of BP learning, adjustments are made to the free parameters of the network on an epoch-by-epoch basis, where each epoch consists of the entire set of training examples. The batch mode is best suited for nonlinear regression.

The back-propagation learning algorithm is simple to implement and computationally efficient in that its complexity is linear in the synaptic weights of the network. However, a major limitation of the algorithm is that it does not always converge and can be excruciatingly slow, particularly when we have to deal with a difficult learning task that requires the use of a large network.

We may try to make back-propagation learning perform better by invoking the following list of heuristics:

- Use neurons with antisymmetric activation functions (e.g., hyperbolic tangent function) in preference to nonsymmetric activation functions (e.g., logistic function). Figure 1.2 shows examples of these two forms of activation functions.
- Shuffle the training examples after the presentation of each epoch; an epoch involves the presentation of the entire set of training examples to the network.
- Follow an easy-to-learn example with a difficult one.
- Preprocess the input data so as to remove the mean and decorrelate the data.
- Arrange for the neurons in the different layers to learn at essentially the same rate. This may be attained by assigning a learning rate parameter to neurons in the last layers that is smaller than those at the front end.
- Incorporate prior information into the network design whenever it is available.

One other heuristic that deserves to be mentioned relates to the size of the training set, N, for a pattern classification task. Given a multilayer perceptron with a total number of synaptic weights including bias levels, denoted by W, a rule of thumb for selecting N is

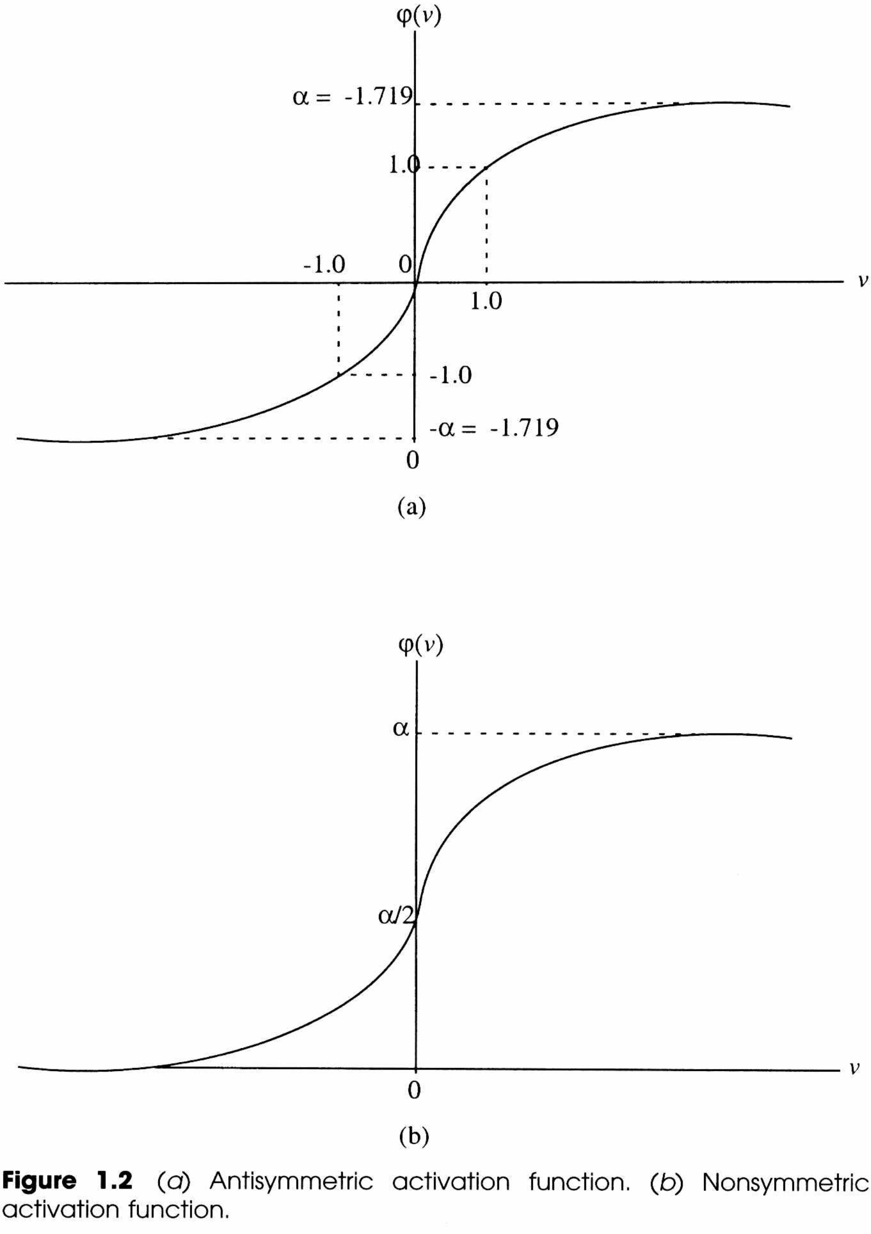


Figure 1.2 (*a*) Antisymmetric activation function. (*b*) Nonsymmetric activation function.

$$N = O\left(\frac{W}{\varepsilon}\right) \tag{1.4}$$

where O denotes "the order of," and ε denotes the fraction of classification errors permitted on test data. For example, with an error of 10% the number of training examples needed should be about 10 times the number of synaptic weights in the network.

Supposing that we have chosen a multilayer perceptron to be trained with the back-propagation algorithm, how do we determine when it is "best" to stop the training session? How do we select the size of individual hidden layers of the MLP? The answers to these important questions may be gotten though the use of a statistical technique known as *cross-validation*, which proceeds as follows (Haykin 1999):

- The set of training examples is split into two parts:
  - Estimation subset used for training of the model
  - Validation subset used for evaluating the model performance
- The network is finally tuned by using the entire set of training examples and then tested on test data not seen before.

1.1.2 Radial-Basis Function Networks

Another popular layered feedforward network is the radial-basis function (RBF) network which has important universal approximation properties (Park and Sandberg 1993), and whose structure is shown in Fig. 13. RBF networks use memory-based learning for their design. Specifically, learning is viewed as a curve-fitting problem in high-dimensional space (Broomhead and Lowe 1989; Poggio and Girosi 1990):

1. Learning is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data.
2. Generalization (i.e., response of the network to input data not seen before) is equivalent to the use of this multidimensional surface to interpolate the test data.

RBF networks differ from multilayer perceptrons in some fundamental respects: