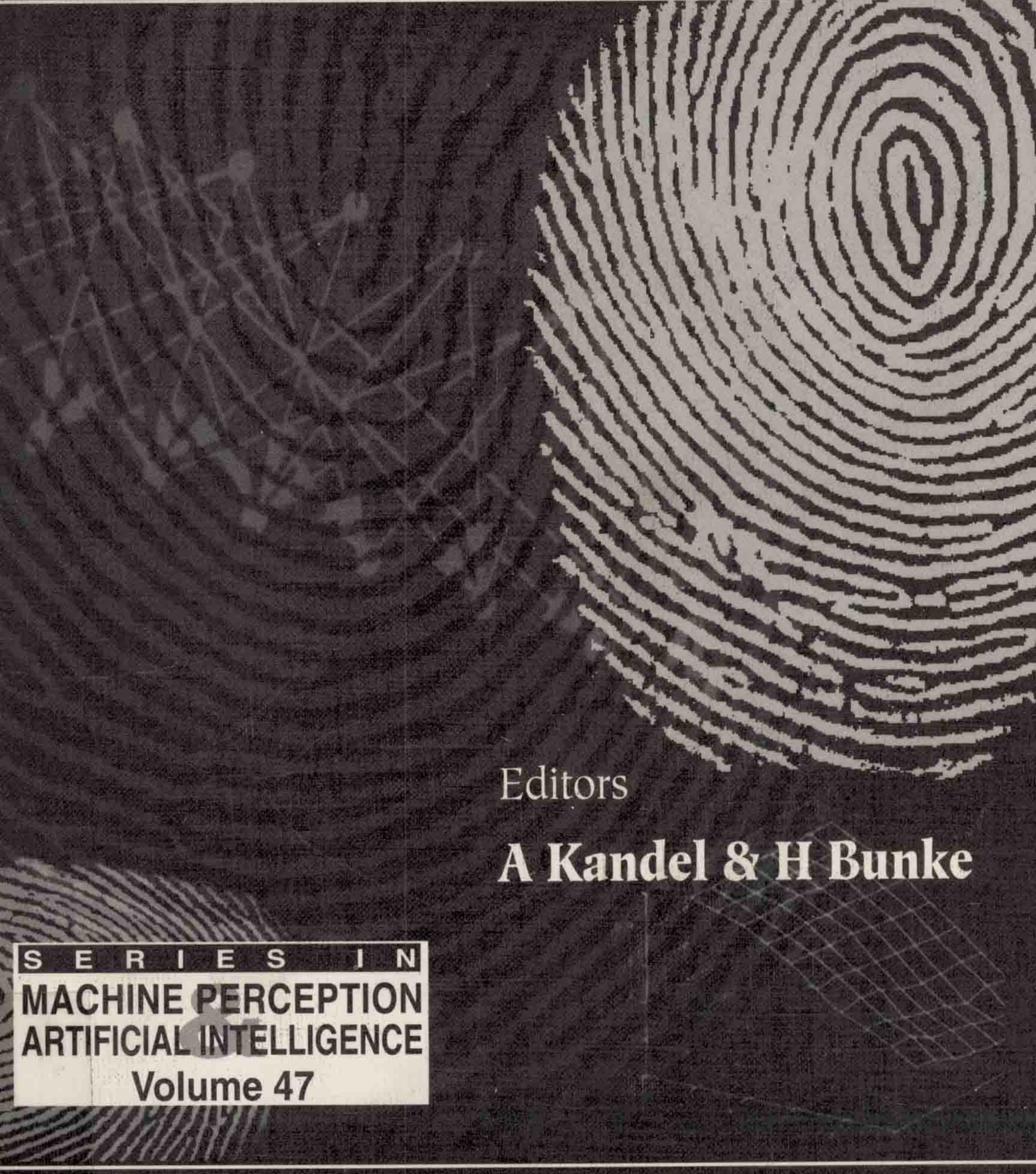
HYBRID METHODS IN PATTERN RECOGNITION



Editors

A Kandel & H Bunke

SERIES IN MACHINE PERCEPTION & ARTIFICIAL INTELLIGENCE
Volume 47

World Scientific

Series in Machine Perception and Artificial Intelligence – Vol. 47

HYBRID METHODS IN PATTERN RECOGNITION

Editors

H Bunke
University of Bern, Switzerland

A Kandel
University of South Florida, USA

World Scientific
New Jersey • London • Singapore • Hong Kong

SERIES IN MACHINE PERCEPTION AND ARTIFICIAL INTELLIGENCE*

Vol. 34: Advances in Handwriting Recognition
(Ed. *S.-W. Lee*)

Vol. 35: Vision Interface — Real World Applications of Computer Vision
(Eds. *M. Cheriet and Y.-H. Yang*)

Vol. 36: Wavelet Theory and Its Application to Pattern Recognition
(*Y. Y. Tang, L. H. Yang, J. Liu and H. Ma*)

Vol. 37: Image Processing for the Food Industry
(*E. R. Davies*)

Vol. 38: New Approaches to Fuzzy Modeling and Control — Design and Analysis
(*M. Margaliot and G. Langholz*)

Vol. 39: Artificial Intelligence Techniques in Breast Cancer Diagnosis and Prognosis
(Eds. *A. Jain, A. Jain, S. Jain and L. Jain*)

Vol. 40: Texture Analysis in Machine Vision
(Ed. *M. K. Pietikäinen*)

Vol. 41: Neuro-Fuzzy Pattern Recognition
(Eds. *H. Bunke and A. Kandel*)

Vol. 42: Invariants for Pattern Recognition and Classification
(Ed. *M. A. Rodrigues*)

Vol. 43: Agent Engineering
(Eds. *Jiming Liu, Ning Zhong, Yuan Y. Tang and Patrick S. P. Wang*)

Vol. 44: Multispectral Image Processing and Pattern Recognition
(Eds. *J. Shen, P. S. P. Wang and T. Zhang*)

Vol. 45: Hidden Markov Models: Applications in Computer Vision
(Eds. *H. Bunke and T. Caelli*)

Vol. 46: Syntactic Pattern Recognition for Seismic Oil Exploration
(*K. Y. Huang*)

Vol. 47: Hybrid Methods in Pattern Recognition
(Eds. *H. Bunke and A. Kandel*)

Vol. 48: Multimodal Interface for Human-Machine Communications
(Eds. *P. C. Yuen, Y. Y. Tang and P. S. P. Wang*)

Vol. 49: Neural Networks and Systolic Array Design
(Eds. *D. Zhang and S. K. Pal*)

*For the complete list of titles in this series, please write to the Publisher.

HYBRID METHODS IN PATTERN RECOGNITION

Published by

World Scientific Publishing Co. Pte. Ltd.

P O Box 128, Farrer Road, Singapore 912805

USA office: Suite 1B, 1060 Main Street, River Edge, NJ 07661

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

HYBRID METHODS IN PATTERN RECOGNITION
Series in Machine Perception and Artificial Intelligence — Vol. 47

ISBN 981-02-4832-6

Printed in Singapore.

Dedicated to

The Honorable Congressman C. W. Bill Young
House of Representatives

for his vision and continuous support in creating the National Institute for Systems Test and Productivity at the Computer Science and Engineering Department, University of South Florida

Preface

The discipline of pattern recognition has seen enormous progress since its beginnings more than four decades ago. Over the years various approaches have emerged, based on statistical decision theory, structural matching and parsing, neural networks, fuzzy logic, artificial intelligence, evolutionary computing, and others. Obviously, these approaches are characterized by a high degree of diversity. In order to combine their strengths and avoid their weaknesses, hybrid pattern recognition schemes have been proposed, combining several techniques into a single pattern recognition system. Hybrid methods have been known about for a long time, but they have gained new interest only recently. An example is the area of classifier combination, which has attracted enormous attention over the past few years.

The contributions included in this volume cover recent advances in hybrid pattern recognition. In the first chapter by H. Ishibuchi and M. Nii, a novel type of neural network architecture is introduced, which can process fuzzy input data. This type of neural net is quite powerful because it can simultaneously deal with different data formats, such as real or fuzzy numbers and intervals, as well as linguistic variables.

The following two chapters deal with hybrid systems that aim at the application of neural networks in the domain of structural pattern recognition. In the second chapter by G. Adorni *et al.*, an extension of the classical backpropagation algorithm that can be applied in the graph domain is proposed. This extension allows us to apply multilayer perceptron neural networks not only to feature vectors, but also to patterns represented by means of graphs. A generalization of self-organizing maps from n-dimensional real space to the domain of graphs is proposed in Chap. 3, by S. Günter and H. Bunke. In particular, the problem of finding the optimal number of clusters in a graph clustering task is addressed.

In Chap. 4, A. Bargiela and W. Pedrycz introduce a general framework for clustering through identification of information granules. It is argued that the clusters, or granules, produced by this method are particularly suitable for hybrid systems. The next two chapters describe combinations of neural networks and hidden Markov models. First, in Chap. 5, G. Rigoll reviews a number of possible combination schemes. Most of them originated in the context of speech and handwriting recognition; however, they are applicable to a much wider spectrum of applications. In Chap. 6, by T. Artieres *et al.*, a system for on-line recognition of handwritten words and sentences is investigated. The main building blocks of this system are a hidden Markov model and a neural net.

The following three chapters address the emerging field of multiple classifier systems. First, in Chap. 7, T. K. Ho provides a critical survey of the field. She identifies the lessons learned from previous work, points out the remaining problems, and suggests ways to advance the state-of-the-art. Then, in Chap. 8, F. Roli and G. Giacinto describe procedures for the systematic generation of multiple classifiers and their combination. Finally, in Chap. 9, A. Verikas *et al.* propose an approach to the integration of multiple neural networks into an ensemble. Both the generation of the individual nets and the combination of their outputs is described.

In the final three chapters of the book applications of hybrid methods are presented. In Chap. 10, A. Klose and R. Kruse describe a system for the interpretation of remotely sensed images. This system integrates methods from the fields of neural nets, fuzzy logic, and evolutionary computation. In Chap. 11, D.-W. Jung and R.-H. Park address the problem of fingerprint identification. The authors use a combination of various methods to achieve robust recognition at a high speed. Last but not least, M. Junker *et al.* describe a system for automatic text categorization. Their system integrates symbolic rule-based learning with subsymbolic learning using support vector machines.

Although it is not possible to cover all current activities in hybrid pattern recognition in one book, we believe that the papers included in this volume are a valuable and representative sample of up-to-date work in this emerging and important branch of pattern recognition. We hope that the contributions are valuable and will be useful to many of our colleagues working in the field.

The editors are grateful to all the authors for their cooperation and the timely submission of their manuscripts. Finally, we would like to thank Scott Dick and Adam Schenker of the Computer Science and Engineering Department at the University of South Florida for their assistance and support.

Horst Bunke, Bern, Switzerland
Abraham Kandel, Tampa, Florida
August 2001

Contents

Multiple Classifier Systems

CHAPTER 1

FUZZIFICATION OF NEURAL NETWORKS FOR CLASSIFICATION PROBLEMS

Hisao Ishibuchi
Department of Industrial Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 599-8531, Japan
E-mail: hisaoi@ie.osakafu-u.ac.jp

Manabu Nii
Department of Computer Engineering, Himeji Institute of Technology
2167 Shosha, Himeji, Hyogo 671-2201, Japan
E-mail: nii@comp.eng.himeji-tech.ac.jp

This chapter explains the handling of linguistic knowledge and fuzzy inputs in multi-layer feedforward neural networks for pattern classification problems. First we show how fuzzy input vectors can be classified by trained neural networks. The input-output relation of each unit is extended to the case of fuzzy inputs using fuzzy arithmetic. That is, fuzzy outputs from neural networks are defined by fuzzy arithmetic. The classification of each fuzzy input vector is performed by a decision rule using the corresponding fuzzy output vector. Next we show how neural networks can be trained from fuzzy training patterns. Our fuzzy training pattern is a pair of a fuzzy input vector and a non-fuzzy class label. We define a cost function to be minimized in the learning process as a distance between a fuzzy output vector and a non-fuzzy target vector. A learning algorithm is derived from the cost function in the same manner as the well-known back-propagation algorithm. Then we show how linguistic rules can be extracted from trained neural networks. Our linguistic rule has linguistic antecedent conditions, a non-fuzzy consequent class, and a certainty grade. We also show how linguistic rules can be utilized in the learning process. That is, linguistic rules are used as training data. Our learning scheme can simultaneously utilize linguistic rules and numerical data in the same framework. Finally we describe the architecture, learning, and application areas of interval-arithmetic-based

neural networks, which can be viewed as a basic form of our fuzzified neural networks.

1. Introduction

Multilayer feedforward neural networks can be fuzzified by extending their inputs, connection weights and/or targets to fuzzy numbers (Buckley and Hayashi 1994). Various learning algorithms have been proposed for adjusting connection weights of fuzzified neural networks (for example, Hayashi et al. 1993, Krishnamraju et al. 1994, Ishibuchi et al. 1995a, 1995b, Feuring 1996, Teodorescu and Arotaritei 1997, Dunyak and Wunsch 1997, 1999). Fuzzified neural networks have many promising application areas such as fuzzy regression analysis (Dunyak and Wunsch 2000, Ishibuchi and Nii 2001), decision making (Ishibuchi and Nii 2000, Kuo et al. 2001), forecasting (Kuo and Xue 1999), fuzzy rule extraction (Ishibuchi and Nii 1996, Ishibuchi et al. 1997), and learning from fuzzy rules (Ishibuchi et al. 1993, 1994). The approximation ability of fuzzified neural networks was studied by Buckley and Hayashi (1999) and Buckley and Feuring (2000). Perceptron neural networks were fuzzified in Chen and Chang (2000).

In this chapter, we illustrate how fuzzified neural networks can be applied to pattern classification problems. We use multilayer feedforward neural networks with fuzzy inputs, non-fuzzy connection weights, and non-fuzzy targets for handling uncertain patterns and linguistic rules such as "If x_1 is *small* and x_2 is *large* then Class 3". Fuzzy numbers and linguistic values are presented to neural networks as input values instead of real numbers. In this case, the input-output relation of each unit is defined by fuzzy arithmetic (Kaufmann and Gupta 1986). Numerical calculations of fuzzy outputs from neural networks are performed on level sets (i.e., α-cuts) of fuzzy inputs using interval arithmetic (Moore 1979). Classification of fuzzy input patterns and the derivation of learning algorithms are based on fuzzy outputs from neural networks.

We first discuss the classification of uncertain patterns by neural networks. Uncertain patterns are denoted by fuzzy input vectors such as $(\tilde{2}, \tilde{3})$ where $\tilde{2}$ and $\tilde{3}$ are fuzzy numbers meaning "about 2" and "about 3", respectively. In this chapter, we use "$\sim$" for explicitly denoting fuzzy numbers. Uncertain patterns are also represented by linguistic vectors such as (*small*, *large*) where "*small*" and "*large*" are linguistic values. Since the meaning of each linguistic value is specified by a membership function on the real axis

$\Re$, linguistic values can be handled in the same framework as fuzzy numbers. Next we discuss the learning of neural networks from fuzzy training patterns. Labeled fuzzy patterns are used as training data. That is, each training pattern is a pair of a fuzzy input vector and its class label. In the same manner as the well-known back-propagation algorithm (Rumelhart et al. 1996), a learning algorithm is derived from a cost function defined by a fuzzy output vector and a non-fuzzy target vector. Then we illustrate the linguistic rule extraction from neural networks. Linguistic rules of the following form are extracted from a neural network trained for an n-dimensional pattern classification problem.

$$\text{Rule } R_p : \text{If } x_1 \text{ is } \tilde{a}_{p1} \text{ and } \ldots \text{ and } x_n \text{ is } \tilde{a}_{pn} \text{ then Class } C_p \text{ with } CF_p\,, \quad (1)$$

where R_p is the label of the p-th rule, $\mathbf{x} = (x_1, \ldots, x_n)$ is an n-dimensional pattern vector, $\tilde{a}_{pi}$ is an antecedent linguistic value on the i-th feature, C_p is a consequent class, and CF_p is a certainty grade. The n antecedent linguistic values are presented as an n-dimensional fuzzy input vector to the trained neural network. The consequent class and the certainty grade are specified based on the corresponding fuzzy output vector. We also discuss the learning of neural networks from linguistic rules of the form in (1). In this case, the antecedent linguistic values are used as a fuzzy input vector as in the fuzzy rule extraction. The corresponding target vector is determined by the consequent class. The certainty grade can be used for adjusting the importance of each linguistic rule in the learning process. Finally, we describe interval-arithmetic-based neural networks. Since fuzzy arithmetic is numerically performed on the level set (i.e., α-cut) of the fuzzy input vector, interval-arithmetic-based neural networks can be viewed as a basic form of fuzzified neural networks. We illustrate some applications of interval-arithmetic-based neural networks to pattern classification problems. For example, they can be used for handing incomplete patterns with missing inputs where each missing input is represented by an interval including its possible values. They can also be used for decreasing the number of inputs required for the classification of new patterns. We show an interval-arithmetic-based approach where each unmeasured input is represented by an interval including its possible values. When human knowledge is represented by intervals such as "If x_1 is in [10, 30] and x_2 is in [4, 7] then Class 2", interval-arithmetic-based neural networks can be used for incorporating such knowledge into the learning of neural networks.

2. Classification of Fuzzy Patterns by Trained Neural Networks

In this section, we concentrate our attention on the classification of uncertain patterns by trained neural networks. The learning of neural networks from uncertain training patterns is discussed in the next section.

2.1. *Classification Task*

Let us assume that a standard three-layer feedforward neural network (Rumelhart et al. 1986) has already been trained for an n-dimensional pattern classification problem with c classes. The number of input units is the same as the dimensionality of the pattern classification problem (i.e., n). The number of hidden units, which is denoted by n_H in this chapter, can be arbitrarily specified. The number of output units is the same as the number of classes (i.e., c). Thus our three-layer feedforward neural network has the $n \times n_H \times c$ structure. When an n-dimensional real vector $\mathbf{x}_p = (x_{p1}, \ldots, x_{pn})$ is presented to our neural network, the input-output relation of each unit is written as follows (Rumelhart et al. 1986):

[Neural Network Architecture]

$$\text{Input units}: \; o_{pi} = x_{pi}, \quad i = 1, 2, \ldots, n, \tag{2}$$

$$\text{Hidden units}: \; o_{pj} = f\left(\sum_{i=1}^{n} w_{ji} o_{pi} + \theta_j\right), \quad j = 1, 2, \ldots, n_H, \tag{3}$$

$$\text{Output units}: \; o_{pk} = f\left(\sum_{j=1}^{n_H} w_{kj} o_{pj} + \theta_k\right), \quad k = 1, 2, \ldots, c. \tag{4}$$

In this formulation, w is a connection weight and θ is a bias. We use the following sigmoidal activation function for the hidden and output units:

$$f(x) = \frac{1}{1 + \exp(-x)}. \tag{5}$$

Normally the input vector $\mathbf{x}_p$ is classified by the output unit with the largest output value. This means that we use the following decision rule:

$$\text{If } o_{pk} < o_{pl} \text{ for } k = 1, 2, \ldots, c \; (k \neq l) \text{ then classify } \mathbf{x}_p \text{ as Class } l. \tag{6}$$

Fig. 1 is an example of the classification boundary generated by a trained neural network using this classification rule. Fig. 1 also shows training data used in the learning of the neural network.

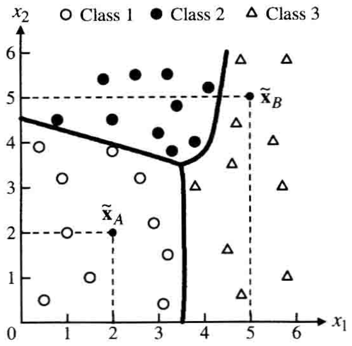


Fig. 1. Classification boundary and training patterns.

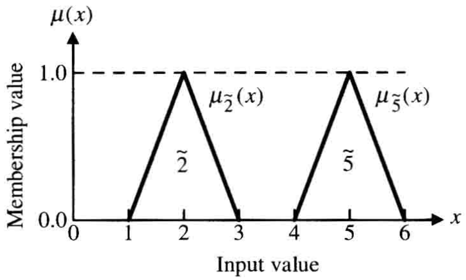


Fig. 2. Examples of membership functions of "about 2" and "about 5."

Our task in this section is to classify uncertain patterns represented by fuzzy vectors. For example, let us consider the classification of a fuzzy vector $\tilde{\mathbf{x}}_A = (\tilde{2}, \tilde{2})$ using the trained neural network in Fig. 1. The meaning of each fuzzy number is mathematically specified by a membership function on the real axis $\Re$. For example, the fuzzy number $\tilde{2}$ may be defined by a triangular membership function as shown in Fig. 2. Roughly speaking, the membership function $\mu_{\tilde{2}}(x)$ of $\tilde{2}$ specifies the possible range of $\tilde{2}$ on the real axis $\Re$. More specifically, the value of $\mu_{\tilde{2}}(x)$ for a specific input x denotes the extent (i.e., membership grade) to which x is compatible with the fuzzy concept "about 2". The membership function $\mu_{\tilde{2}}(x)$ of $\tilde{2}$ in Fig. 2 is written as

$$\mu_{\tilde{2}}(x) = \max\{0,\ 1 - |2 - x|\}\,. \tag{7}$$

While the fuzzy vector $\tilde{\mathbf{x}}_A = (\tilde{2},\ \tilde{2})$ involves a certain amount of uncertainty, the neural network may be able to classify $\tilde{\mathbf{x}}_A$ as Class 1 because $\tilde{\mathbf{x}}_A$ is located far from the classification boundary (see Fig. 1). On the other hand, it seems to be difficult for the neural network to classify another fuzzy vector $\tilde{\mathbf{x}}_B = (\tilde{5},\ \tilde{5})$ because $\tilde{\mathbf{x}}_B$ is located near the classification boundary (see Fig. 1 for the location of $\tilde{\mathbf{x}}_B$ and Fig. 2 for the membership function of $\tilde{5}$). In this section, we mathematically formulate these intuitive discussions as a decision rule for fuzzy input vectors.

2.2. *Calculation of Fuzzy Outputs*

Let $\tilde{\mathbf{x}}_p = (\tilde{x}_{p1}, \ldots, \tilde{x}_{pn})$ be an n-dimensional fuzzy input vector to our neural network. Note that $\tilde{x}_{pi}$ can be a real number and an interval because they are represented in the same framework as fuzzy numbers. For example, a real number a and an interval $A = [a_1,\ a_2]$ are represented by the following membership functions:

$$\mu_a(x) = \begin{cases} 1\,, & \text{if } x = a\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{8}$$

$$\mu_A(x) = \begin{cases} 1\,, & \text{if } a_1 \le x \le a_2\,, \\ 0\,, & \text{otherwise}\,. \end{cases} \tag{9}$$

Thus the fuzzy input vector $\tilde{\mathbf{x}}_p = (\tilde{x}_{p1}, \ldots, \tilde{x}_{pn})$ can be a mixture of fuzzy numbers, intervals and real numbers such as $\tilde{\mathbf{x}}_p = (\tilde{5}, [2,\ 3], 3.48)$.

When the fuzzy input vector $\tilde{\mathbf{x}}_p = (\tilde{x}_{p1}, \ldots, \tilde{x}_{pn})$ is presented to the neural network, the input-output relation of each unit in (2)-(5) is defined by fuzzy arithmetic (Kaufmann and Gupta 1985). For example, the fuzzy output $\tilde{o}_{pj}$ from the j-th hidden unit is calculated by extending the input vector $\mathbf{x}_p = (x_{p1}, \ldots, x_{pn})$ to the fuzzy vector $\tilde{\mathbf{x}}_p = (\tilde{x}_{p1}, \ldots, \tilde{x}_{pn})$ in (2)-(3) as

$$\tilde{o}_{pj} = f\left(\sum_{i=1}^{n} w_{ji}\tilde{x}_{pi} + \theta_j\right)\,. \tag{10}$$