NEURO-FUZZY PATTERN RECOGNITION

Editors

H. Bunke

A. Kandel

SERIES IN MACHINE PERCEPTION & ARTIFICIAL INTELLIGENCE

Volume 41

World Scientific

Series in Machine Perception and Artificial Intelligence – Vol. 41

NEURO-FUZZY PATTERN RECOGNITION

Editors

H. Bunke
University of Bern, Switzerland

A. Kandel
University of South Florida, USA

World Scientific
Singapore • New Jersey • London • Hong Kong

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.

NEURO-FUZZY PATTERN RECOGNITION
Series in Machine Perception and Artificial Intelligence — Vol. 41

ISBN 981-02-4418-5

Printed in Singapore by FuIsland Offset Printing

Preface

The discipline of pattern recognition has a history of more than four decades now. Over the years various approaches emerged. Initially, the statistical approach was prevailing. In this approach, patterns are represented as vectors, or points, in an n-dimensional feature space, and recognition is based on the idea of partitioning this space into disjoint regions using methods from statistical decision theory. Later, during the 1970s structural and syntactic techniques became a focus of research. In structural and syntactic pattern recognition, symbolic data structures, such as strings, trees and graphs, are used for pattern representation, and recognition is cast as a matching or a parsing problem. In the beginning of the 1980s knowledge based approaches to pattern recognition emerged. Here, the patterns of a class are described by rules and pattern recognition is accomplished through automated reasoning or inference procedures. Later in the 1980s the potential of neural networks for pattern recognition was discovered. Neural networks are based on a pattern representation that is similar to the one used in the statistical approach. They employ a number of simple processing elements to map the feature vector of an input pattern to one or several output classes. During the evolution of the various approaches mentioned above, concepts from the area of fuzzy logic and soft computing have proven useful for various pattern recognition tasks. These concepts are characterized by their ability to cope with uncertainty, including the imprecision and ambiguity present in human languages.

In recent years one could observe a consolidation and further development of all these approaches to pattern recognition. In particular, many hybrid schemes combining different pattern recognition methods with each other were proposed. In general, a hybrid system aims at combining the advantages of different paradigms with each other. The focus of this book is on neuro-fuzzy pattern recognition. It is characterized by the combination of neural networks with techniques from fuzzy sets and systems. Neuro-fuzzy systems exhibit the noise robustness and learning capabilities of neural networks together with the ability of fuzzy systems to explicitly model uncertainty, linguistic concepts, and the knowledge of human experts.

The contributions included in this book cover a broad spectrum of novel methods and applications from neuro-fuzzy pattern recognition. In the first chapter, by N.R. Pal and D. Chakraborty, a neuro-fuzzy system capable of performing feature selection and pattern recognition in an integrated manner is described. Feature selection is also the topic of the next chapter. R.K. De et al. propose an unsupervised approach, combining a neural network with concepts from the theory of fuzzy systems. Then, M.B. Gorzalczany describes a new classification method that combines neural networks, fuzzy logic, and genetic algorithms. In Chapter 4, the problem of clustering is addressed. J.-S. Lin discusses various clustering strategies, including the combination of the well-known fuzzy c-means algorithm with a neural network. Next, A. Rizzi introduces min-max classifiers, and proposes a new learning strategy for this type of neuro-fuzzy system. The organization of a set of data into basic conceptual entities, so-called granules, is the topic of Chapter 6, by W. Pedrycz and G. Vukovitch. The authors discuss various properties of information granules, including their application in a neural classifier. In the next chapter, Sainz Palmero et al. describe a family of classifiers based on Adaptive Resonance Theory (ART) and demonstrate applications in the area of printed document analysis. While the focus of the first seven contributions is on methodological issues, the remaining chapters emphasize various applications. N. Kasabov and G. Iliev describe methods for robust speech recognition using neuro-fuzzy techniques. In Chapter 9, P.D. Gader et al. present a neuro-fuyyz system for automatic land mine detection. The segmentation of MR images of the human brain by means of clustering and a neuro-fuzzy network is the subject of Chapter 10, by S.Y. Lee et al. In the final chapter, by Y.J. Ryoo, a neuro- fuzzy controller for steering an autonomous vehicle in proposed.

It is not intended to cover the whole area of neuro-fuzzy pattern recognition in this book. Nevertheless, the editors believe that the papers included here are a valuable and representative sample of up-to-date work in this emerging and important branch of pattern recognition, and will assist many researchers in the field. We want to thank all authors for their cooperation and the timely submission of their manuscripts. Further thanks are due to S. Dick of the University of South Florida, and Ch. Irniger of the University of Bern for editorial assistance.

Horst Bunke, Bern, Switzerland

Abraham Kandel, Tampa, Florida

June 2000

CONTENTS

METHODOLOGY

Simultaneous Feature Analysis and System Identification in a Neuro-Fuzzy Framework

N.R. Pal and Debrup Chakraborty
Indian Statistical Institute, Calcutta
India

Abstract

Most methods of fuzzy rule based system identification either ignore feature analysis or do it in a separate phase. In this chapter we propose a novel neuro-fuzzy system that can simultaneously do feature analysis and system identification in an integrated manner. It is a five-layered feed-forward network for realizing a fuzzy rule based system. The second layer of the net is the most important one, which along with fuzzification of the input also learns a modulator function for each input feature. This enables online selection of important features by the network. The system is so designed that learning maintains the non-negative characteristic of certainty factors of rules. The proposed method is tested on both synthetic and real data sets and the performance is found to be quite satisfactory.

Keywords : System Identification, Fuzzy Systems, Feature Analysis, Rule Extraction

1 Introduction

Let $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\} \subset R^s$ and $Y = \{\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_N\} \subset R^t$ and let there be an unknown function $\mathbf{S} : R^s \Rightarrow R^t$ such that $\mathbf{y}_k = \mathbf{S}(\mathbf{x}_k)\ \forall k = 1, ..., N$.

In other words, there is an unknown function $\mathbf{S}$ which transforms $\mathbf{x}$ to $\mathbf{y}$. The problem of System Identification (SI) is to find S explicitly or implicitly. SI appears in various forms in science and engineering. There are many approaches to SI. Some models, like regression, are explicit in nature while others such as neural networks and fuzzy systems are computational transforms that do SI implicitly.

It is known that neural networks can act as universal approximators for a large class of non-linear functions, hence the choice of neural networks for SI is quite justified and has been proved to be successful [Haykin, 1994]. Neural Networks are usually robust, possess parallelism and good generalizing capabilities but they do not have readability and work as a black box. Hence, the underlying relation in a system which has been approximated by a Neural Network cannot be understood from the network by any easy means. On the other hand, fuzzy rule-based systems which have also been used for SI are highly interpretable in terms of linguistic rules. As fuzzy if-then rules can be easily understood by human beings and often an initial rule-base can be provided by an expert, there is no problem of readability. However, fuzzy rule based systems, as such are not capable of learning. So to extract the rules from a given data one has to depend on techniques like clustering or other tools of exploratory data analysis [Pal *et al., 1997*] or an initial rule base is supplied by an expert, which is then tuned using data. Thus, judicious integrations of neural networks and fuzzy logic are expected to result in systems with merits of both paradigms. Several attempts have been made to integrate fuzzy systems and neural networks, with a view to achieving systems which are interpretable, robust and have learning abilities [Lee and Lee, 1975; Lin and Lee, 1993; Lin and Lee, 1996; Pal and Pal, 1996].

The various neuro fuzzy unification schemes developed till date can possibly be classified into three major groups :

- Neural Fuzzy Systems
- Fuzzy Neural Systems
- Co-operative Systems

Neural fuzzy systems are fuzzy systems implemented by neural networks [Keller *et al.*, 1992; Keller and Tahani 1992; Pal and Pal, 1996; Pal *et al., 1998*; Lin and Lee, 1993; Lin and Lee, 1996]. Fuzzy neural systems are neural networks, capable of handling fuzzy information [Hayashi *at al.*, 1993; Ishibuchi *et al.*, 1993; Pal and Pal, 1996]. The inputs, outputs and weights of fuzzy neural networks could be fuzzy sets, often fuzzy numbers or membership values. The Co-operative systems are those which use different paradigms (neuro or fuzzy) to solve various facets of the same problem [Pal and Pal, 1996]. All these three paradigms taken together is known as neuro-fuzzy computing. The scheme that we are going to present here is a neural fuzzy system. Hence to begin with we discuss some previous attempts in this direction.

Lee *et al.*(1994) proposed a neural network model for fuzzy inferencing. They developed an algorithm for adjusting (tuning) the membership functions of antecedent linguistic values of the rule set by error backpropagation (EBP), where the consequent parts were considered fixed. Thus the extracted fuzzy rules after tuning retain the same linguistic description as the initial rules. Li and Wu (1994) proposed a neuro fuzzy hierarchical system with if-then rules for pattern classification problem. A five layer network is also presented by Yao *et al.* (1996). The parameters of the net are identified using evolutionary programming and the tuned network is then pruned to extract a small set of rules. Lin and Lee (1993) presented a multilayered feedforward connectionist model designed for fuzzy logic control and decision making. A hybrid two step learning scheme that combined self-organized (unsupervised) and supervised learning algorithms for selection of fuzzy rules and tuning of membership functions were developed. Lin and Lee used Kohonen's self-organizing feature map [Kohonen, 1998] for finding the centers of the membership functions. After selection of the rule set, i.e., when the network architecture is established, the second step of supervised learning begins. Some heuristic guidelines for rule reduction and combination were also provided. Shann and Fu (1995) presented a layered network for selection of rules. Initially, the network was constructed to contain all possible fuzzy rules. After EBP training, the redundant rules were deleted by a rule pruning process for obtaining a concise rule base. The architecture of Shann and Fu is similar to that of Lin and Lee in several respects. Pal and Pal (1996) discussed some limitations of the scheme by Shann and Fu and provided a better rule tuning and pruning strategy. Lin and Cunningham (1995) also developed a layered network for system identification. They used fuzzy curves for feature selection, but the feature selection phase was a part of preprocessing on the data before the data get into the network.

None of the methods discussed here explicitly perform any feature analysis. However, it is well known that feature analysis plays an important role in SI [Pal, 1999; Sugeno and Yasukawa, 1993]. For example, consider a system with input $\mathbf{x} \in R^s$ and output $\mathbf{y} \in R^t$. It may be possible that not all the s input features are required to understand the relation between the input and output or, may be some of the features are redundant or indifferent to the output of the system. Moreover, more features are not necessarily good, some features may even have some derogatory effect on the output. Thus, selection of an appropriate subset of features, for the given task at hand, not only can reduce the cost of the system but also can, and usually will, improve the performance of the system.

There are many methods of feature analysis or feature ranking. Details of some of the feature analysis methods using soft computing tools like fuzzy logic, neural networks and genetic algorithms can be found in [Pal, 1999; De *et al*,1997]. Following the concept of Pal and Chintalapudi (1997) the feature selection scheme proposed here uses a modulator function. Pal and Chintalapudi used a multilayered feed-forward architecture. Every input feature was multiplied by an attenuation function prior to its entry in the network. The attenuation functions were so designed that they took values between 0 and 1. The parameters of the attenuation functions were learned by the EBP learning scheme. After training, for a bad or indifferent feature, the attenuation function acquires a value close to 0 and for a good feature a value close to 1. The present work is inspired by the feature selection scheme of Pal and Chintalpudi but our formulation is quite different.

Here we present a neural fuzzy system for simultaneous feature selection and SI. To our knowledge no connectionist system exists which does feature selection and SI simultaneously. In subsequent sections we discuss the network structure of the proposed system followed by the learning rules and some simulation results. And finally, the paper is concluded in Section 5, which also gives some directions of future works on the proposed system.

2 The Network Structure

We consider a system with s input features $(x_1, x_2, ..., x_s)$ and t output features $(y_1, y_2, ..., y_t)$. The proposed network (neural-fuzzy) system will deal with fuzzy rules of the form, R_i : If x_1 is A_{1i} and x_2 is A_{2i} ..... and x_s is A_{si} then y_j is B_{ji}. Here A_{ji} is the i-th fuzzy set defined on the domain of x_j and B_{ji} is the i-th fuzzy set defined on the domain of y_j.

From our notation one might think that for each rule we are using

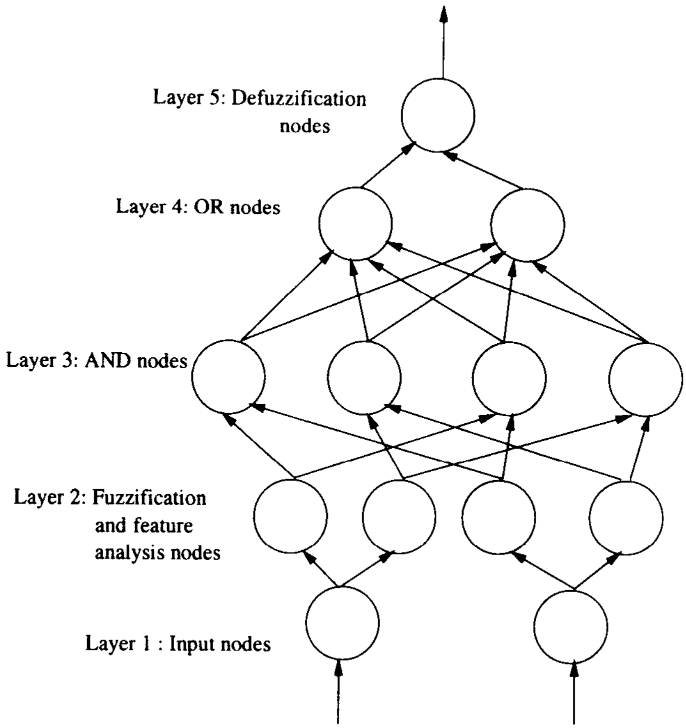

Fig. 1 The network structure.

a different set of antecedent linguistic values (fuzzy sets) but that is not necessarily true; in fact, for every feature only a few fuzzy sets are defined and hence some of the $A_{ij} = A_{ik}$ for some j and k. Similar is the case for the linguistic values defined on the output variables.

The neural-fuzzy system is realized using a five layered network as shown in Figure 1. The node functions with its inputs and outputs, are discussed layer by layer. We use suffixes p, n, m, k, l to denote respectively the suffixes of the nodes in layers 1 through 5 in order. The output of each node is denoted by z.

Layer 1: Each node in layer 1 represents an input linguistic variable of the network and is used as a buffer to transmit the input to the next layer, that is to the membership function nodes of its linguistic values. Thus the number of nodes in this layer is equal to the number of input features in the data. If x_p denotes the input to any input node then the output of any node in layer 1 will be

$$z_p = x_p. \tag{1}$$

Layer 2: Each node in layer 2 represents the membership functions of a linguistic value associated with an input linguistic variable. Moreover, this layer also does the feature analysis. The output of these nodes lies in the interval [0,1] and represents the membership grades of the input with respect to different linguistic values. Therefore, the nodes in this layer act as fuzzifiers. The most commonly used membership functions are triangular, trapezoidal and bell shaped. Although any one of these choices may be used, we consider bell shaped membership functions. The weights to the input link in this layer are unity. If there be N_i fuzzy sets associated with the i^{th} feature and if there are a total of s input features then the number of nodes in this layer would be $\sum_{i=1}^{s} N_i$. The output of a node in layer 2 is denoted by

$$\bar{z}_n = exp\{-\frac{(z_p - \mu_n)^2}{\sigma_n^2}\}. \tag{2}$$

In Equation (2) the subscript n denotes the n-th term (fuzzy set) of the linguistic variable x_p. μ_n and σ_n represent the mean and spread respectively of the bell shaped function representing a term of the linguistic variable x_p associated to node n.

For the purpose of feature selection, the output of this layer is modified so that every indifferent feature x_p gets eliminated. If a linguistic variable x_p is not important (or is indifferent) for describing the system behavior, i.e., for defining the input-output relation, then the values of x_p should not have any effect on the firing strength of the rules involving that input variable. This is our main guiding principle for feature analysis and it makes our approach completely different from the work of Pal and Chintalapudi. If we restrict ourselves to minimum or product to compute the firing strength of a rule, then this can be realized if an indifferent feature always generates a membership of unity. This may appear impossible at the first sight. Note that for an indifferent feature, all of its terms (i.e., all of its linguistic values) should have no effect on the firing strength. Next we explain how this can be realized.

Let us associate a function f_n with each node n in layer 2. We call f_n as a modulator function. For an indifferent (or bad) feature we want all linguistic values defined on that feature to give a membership of 1. To achieve this we model f_n as :

$$f_n = exp\left[\lambda_p \, ln\left(\frac{1}{\bar{z_n}}\right)\right]. \tag{3}$$

Here $\lambda_p \in [0,1]$ is a parameter associated with a particular linguistic variable x_p of which node n is a term. From Equation (3) we see that when λ_p is nearly 1 then f_n is nearly $\frac{1}{\bar{z_n}}$, and when λ_p is nearly 0 then f_n is nearly 1. So for bad features λ_p should get large values (close to 1) and small values (close to 0) for good features. Thus, for a bad feature, the modulated membership value would be $f_n \bar{z_n} \approx \bar{z_n}\frac{1}{\bar{z_n}} \approx 1$ irrespective of the value of x_p. Similarly, for a good feature, the modulated membership value would be $f_n \bar{z_n} \approx 1\bar{z_n} \approx \bar{z_n} \approx$ the actual membership value. Since λ_p must take values between zero and one, we model λ_p by $e^{-\beta_p^2}$. Thus, the activation function of any node n in layer 2 would be as :

$$z_n = \bar{z_n} \; exp\left[e^{-\beta_p^2} ln\left(\frac{1}{\bar{z_n}}\right)\right], \tag{4}$$

which can be simplified to

$$z_n = \bar{z}_n^{(1-e^{-\beta_p^2})}, \tag{5}$$

where $\bar{z}_n$ is computed using Equation (2). The parameter β_p can be learnt by back-propagation or by some other technique. We see that when β_p^2 takes a large value then z_n tends to $\bar{z}_n$ and for small values of β_p^2, z_n tends to 1, thereby making the feature indifferent. Therefore, our objective would be to make β_p^2 take large values for good features and small values for bad ones through the process of learning. Layer 2 can be better realized using two layers of neurons, the first one for computation of the membership value,$\bar{z}_n$ and second layer for the modulated output using Equation (5).

Layer 3: This layer is called the AND layer. Each node in this layer represents a possible IF part of the fuzzy rules. There are many operators (T-norms) for fuzzy intersection [Klir and Yuan, 1995]. Here we choose product as the operator for intersection. The number of nodes in this layer would be $\prod_{i=1}^{s} N_i$. The output of the m-th node in the layer would be

$$z_m = \prod_{n \in P_m} z_n \tag{6}$$