Available online at www.sciencedirect.com

ScienceDirect

Neurocomputing 71 (2007) 297–310

NEUROCOMPUTING

www.elsevier.com/locate/neucom

The design of neuro-fuzzy networks using particle swarm optimization and recursive singular value decomposition

Cheng-Jian Lin*, Shang-Jin Hong

Department of Computer Science and Information Engineering, Chaoyang University of Technology, 168 Gifong E. Rd., Wufong, Taichung County 413, Taiwan

Received 27 December 2005; received in revised form 30 August 2006; accepted 15 December 2006
Communicated by A. Abraham
Available online 22 February 2007

Abstract

In this paper, a neuro-fuzzy network with novel hybrid learning algorithm is proposed. The novel hybrid learning algorithm is based on the fuzzy entropy clustering (FEC), the modified particle swarm optimization (MPSO), and the recursive singular value decomposition (RSVD). The FEC is used to partition the input data for performing structure learning. Then, we adopt the MPSO to adjust the antecedent parameters of fuzzy rules. Two strategies in the MPSO, called the effective local approximation method (ELAM) and the multi-elites strategy (MES), are proposed to improve the performance of the traditional PSO. Moreover, we will apply RSVD to obtain the optimal consequent parameters of fuzzy rules. The proposed hybrid learning algorithm achieves superior performance in learning speed and learning accuracy than those of some traditional genetic methods.

Keywords: Neuro-fuzzy networks; Fuzzy entropy; Particle swarm optimization; Function approximation; Singular value decomposition (SVD)

1. Introduction

Neuro-fuzzy networks have been demonstrated to be successful [10,17–20,23–25,27]. It combines the semantic transparency of rule-based fuzzy systems and the learning capability of neural networks. Neuro-fuzzy networks have two typical types which are Mamdani-type and TSK-type models. For Mamdani-type neuro-fuzzy networks [19,27], the minimum fuzzy implication is used in fuzzy reasoning. Meanwhile, for TSK-type neuro-fuzzy networks (TNFNs) [10,17,20,23], the consequence of each rule is a function input variable. The general adopted function is a linear combination of input variables plus a constant term. Many researchers [10,17] have been shown that a TNFN achieves superior performance in network size and learning accuracy than that of a Mamdani-type neuro-fuzzy network.

The back-propagation (BP) learning algorithm [18] is widely used for training neuro-fuzzy networks by means of error propagation via variation calculus. However, the BP learning algorithm is a powerful training technique that can be applied in networks with feed-forward structure to transform them into adaptive systems. But the algorithm may reach the local minima and the global solution may never be found because the steepest descent optimization technique is used in BP training to minimize the error function. In addition, the performance of the BP learning algorithm depends on the initial values of the model parameters, and for different network topologies one has to derive new mathematical expressions for each network layer. About this, the advent of evolutionary computation has inspired new designs and models. In contrast to traditional computation systems, which may be good at accurate and exact computation but have brittle operations, evolutionary computation provides a more robust and efficient approach for solving complex real-world problems [2,6,28]. For this reason, many researchers use genetic algorithms (GAs) for learning of fuzzy models. In

*Corresponding author.
E-mail address: cjlin@mail.cyut.edu.tw (C.-J. Lin).

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.12.016

the literature [9,11,16], several GA-based approaches have appeared and have better candidates than BP algorithm.

Recently, a new algorithm, called particle swarm optimization (PSO), was proposed. It is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 [14]. The underlying motivation for the development of PSO algorithm was social behavior of animals such as bird flocking, fish schooling, and swarm theory. The PSO begins with a random population and searches for optima by updating the population. The advantages of PSO are that it has no evolution operators such as crossover and mutation and it need not adjust too many free parameters. Moreover, each potential solution is also assigned a randomized velocity. The potential solutions, called particles, are then "flown" through the problem space. Compared with the GA, the PSO has some attractive characteristics. First, the PSO has memory. The knowledge of good solutions in the PSO is retained by all particles; whereas in the GA, the previous knowledge of the problem is destroyed when the population is changed. Second, the PSO has constructive cooperation between particles. The particles in the swarm share information between them. Successful applications of the PSO for several optimization problems, like control problems [1,7,29] and feed-forward neural network design [12,21]. Therefore, we introduce a modified PSO (MPSO) to determine the antecedent parameters of fuzzy rule.

In this paper, we propose a novel hybrid learning algorithm for neuro-fuzzy networks. The construction of neuro-fuzzy networks involves two phases: structure learning and parameter learning. In structure learning, the input data set is partitioned into a set of clusters using fuzzy entropy clustering (FEC). Membership functions associated with each cluster are defined according to statistical means and variances of the data points that were included in the cluster. Then a fuzzy rule is extracted from each cluster to form a fuzzy rule base. In parameter learning, most neuro-fuzzy networks use BP to refine parameters of the system. However, BP suffers from the problems of local minima and lower convergence rate. To decrease the size of the search space and speed up the convergence, we propose a MPSO to adjust the antecedent parameters of fuzzy rules. In the MPSO, two strategies, called effective local approximation method (ELAM) and multi-elites strategy (MES), are proposed to improve the performance of the traditional PSO. Moreover, we adopt the recursive singular value decomposition (RSVD) to determine the optimal consequent parameters of fuzzy rules. The proposed hybrid learning algorithm has the following advantages: (1) Using the MPSO to find the global optimal solution is easier than the BP method. (2) Determining the initial particles of MPSO by FEC method can reduce blind search for parameters of fuzzy rules. (3) The proposed hybrid learning algorithm achieves superior performance in learning speed and learning accuracy.

This paper is organized as follows. Section 2 describes the TSK-type fuzzy model. Overview of the PSO is described in Section 3. In Section 4, we will describe the proposed hybrid learning algorithm for TNFNs. Section 5 presents the simulation results. Finally, conclusions are given in Section 6.

2. Structure of a TNFN

A fuzzy model is a knowledge-based system characterized by a set of rules, which models the relationship among control input and output. The reasoning process is defined by means of the employed aggregation operators, the fuzzy connectives and the inference method. The fuzzy knowledge base contains the definition of fuzzy sets stored in the fuzzy database and a collection of fuzzy rules, which constitute the fuzzy rule base. Fuzzy rules are defined by their antecedents and consequents, which relates an observed input state to a desired control action. Most fuzzy systems employ the inference method proposed by Mamdani in which the consequence parts are defined by fuzzy sets [18]. A Mamdani-type fuzzy rule has the form:

$$\begin{aligned}&\text{IF } x_1 \text{ is } A_{1j}(m_{1j},\sigma_{1j}) \text{ and } x_2 \text{ is } A_{2j}(m_{2j},\sigma_{2j})\ldots\text{and}\\&\quad x_n \text{ is } A_{nj}(m_{nj},\sigma_{nj}) \text{ THEN } y' \text{ is } B_j\,(m_j,\sigma_j),\end{aligned}\tag{1}$$

where m_{ij} and σ_{ij} represent a Gaussian membership function with mean and deviation with ith dimension and jth rule node. The consequences B_j of jth rule is aggregated into one fuzzy set for the output variable y'. The crisp action is obtained through defuzzification, which calculates the centroid of the output fuzzy set. Besides the more common fuzzy inference method proposed by Mamdani, Takagi, Sugeno and Kang introduced a modified inference scheme [23]. The first two parts of the fuzzy inference process, fuzzifier the inputs and applying the fuzzy operator are exactly the same. A Takagi–Sugeno–Kang (TSK)-type fuzzy model employs different implication and aggregation methods than the standard Mamdani controller. Instead of using fuzzy sets the conclusion part of a rule, is a linear combination of the crisp inputs:

$$\begin{aligned}&\text{IF } x_1 \text{ is } A_{1j}(m_{1j},\sigma_{1j}) \text{ and } x_2 \text{ is } A_{2j}(m_{2j},\sigma_{2j})\ldots\text{and}\\&\quad x_n \text{ is } A_{nj}(m_{nj},\sigma_{nj}) \text{ THEN } y' = w_{0j} + w_{1j}x_1 + \cdots + w_{nj}x_n.\end{aligned}\tag{2}$$

Since the consequence of a rule is crisp, the defuzzification step becomes obsolete in the TSK inference scheme. Instead, the control output is computed as the weighted average of the crisp rule outputs, which is computationally less expensive then calculating the center of gravity.

In this paper, the structure of the TNFN is shown in Fig. 1, where n and R are, respectively, the number of input dimensions and the number of rules. It is a five-layer network structure. The functions of the nodes in each layer are described as follows:

Layer 1 (*Input node*): No function is performed in this layer. The node only transmits input values to layer 2.

$$u_i^{(1)} = x_i, \quad i = 1\ldots n. \tag{3}$$