Neural Networks 21 (2008) 1556–1566

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Automatic generation of fuzzy inference systems via unsupervised learning

Meng Joo Er [a], Yi Zhou [b,*]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, S1, 50 Nanyang Ave, Singapore 639798, Singapore

[b] School of Electrical and Electronic Engineering, Singapore Polytechnic, 500 Dover Road, Singapore 139651, Singapore

ARTICLE INFO

Article history:
Received 16 November 2007
Received in revised form
5 May 2008
Accepted 14 June 2008

Keywords:
Fuzzy inference systems
Unsupervised learning
Reinforcement learning
Fuzzy neural networks
Structure identification

ABSTRACT

In this paper, a novel approach termed Enhanced Dynamic Self-Generated Fuzzy Q-Learning (EDSGFQL) for automatically generating Fuzzy Inference Systems (FISs) is presented. In the EDSGFQL approach, structure identification and parameter estimations of FISs are achieved via Unsupervised Learning (UL) (including Reinforcement Learning (RL)). Instead of using Supervised Learning (SL), UL clustering methods are adopted for input space clustering when generating FISs. At the same time, structure and preconditioning parts of a FIS are generated in a RL manner in that fuzzy rules are adjusted and deleted according to reinforcement signals. The proposed EDSGFQL methodologies can automatically create, delete and adjust fuzzy rules dynamically. Simulation studies on wall-following and obstacle avoidance tasks by a mobile robot show that the proposed approach is superior in generating efficient FISs.

1. Introduction

Fuzzy Logic (FL) was first proposed by Zadeh (1965) in 1965 and it has been deployed in a wide range of applications (Dubois, Prade, & Yager, 1997). However, some shortcomings of FL are the lack of systematic design and learning capability. To circumvent these problems, FL is usually combined with Neural Networks (NNs) and the hybrid technology of Fuzzy Neural Networks (FNNs) combines the profound learning capability of NNs (Lee & Kil, 1991; Zurada & Malinowski, 1994) with the mechanism of explicit and easily interpretable knowledge presentation provided by FL.

The main issues for generating a FIS are structure identification and parameter estimation. Structure identification is concerned with how to determine the number of fuzzy rules according to the task requirement while parameter estimation involves the determination of parameters for both premises and consequents of fuzzy rules (Wu & Er, 2000). According to the information sources of learning, structure identification and parameter estimation can be accomplished by Supervised Learning (SL), Unsupervised Learning (UL) and Reinforcement Learning (RL). Note that RL is considered as a special type of UL in some works.

The key characteristic of SL is the existence of a "teacher", the instructive training input-output data. A great number of research results for generating FISs via SL paradigms have been reported (Jang, 1993; Wu & Er, 2000; Wu, Er, & Gao, 2001). However, the instructive training data required in SL are not always available especially when a human being has little knowledge about the system or the system is uncertain. In those situations, UL and RL are preferred over SL as UL and RL are learning processes that do not need any instructive information.

Recently, a number of researchers have applied RL to train the consequent parts of a FIS (Er & Deng, 2004; Jouffe, 1998; Juang, 2005). In the Fuzzy Q-Learning (FQL) of Jouffe (1998), the consequent parts of a FIS are selected by Q-learning (Watkins & Dayan, 1992). However, structure and premise parameters are still determined by a *priori* knowledge. To circumvent this problem, a Dynamic Fuzzy Q-Learning (DFQL) was proposed in Er and Deng (2004) and an online Clustering and Q-value based Genetic Algorithm learning scheme for Fuzzy system design (CQGAF) was proposed in Juang (2005).

Both DFQL and CQGAF methods achieve online structure identification by creating fuzzy rules when the input space is not well partitioned. However, both methods cannot adjust the premise parameters except during rule creation. Structure and preconditioning parts of FISs are generated without considering the system performance. Moreover, both methods cannot delete fuzzy rules once they are generated even when the rules become redundant. Although the author of Juang (2005) declared that the number of rules could be reduced by a similarity test of the Membership Functions (MFs), the generated rules cannot be deleted as there is no criterion to delete them. As an efficient FIS, the system should be adjusted according to system performance and dormant or unnecessary rules should be deleted.

A weight decay method was discussed in the work of Reed (1993), a pruning approach based on the output of each neuron

* Corresponding author. Tel.: +65 68707873; fax: +65 67721974.
E-mail addresses: emjer@ntu.edu.sg (M.J. Er), zhouyi@sp.edu.sg (Y. Zhou).

0893-6080/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2008.06.007

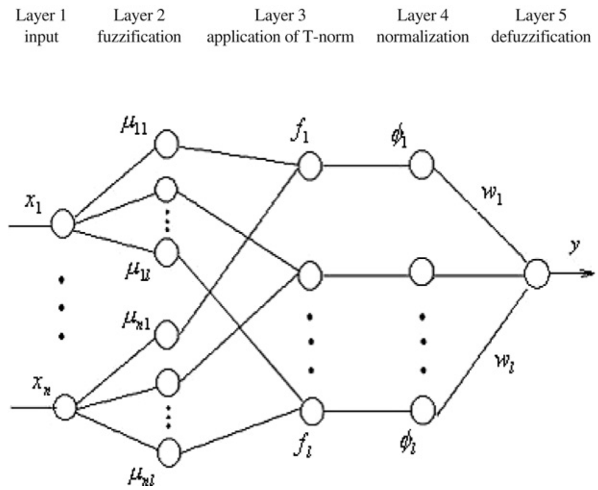

Fig. 1. Structure of the fuzzy inference system in neural networks.

unit was presented in Lu, Sundararajan, and Saratchandran (1997) and the Error Reduction Ratio method presented in Chen, Cowan, and Grant (1991) was adopted as a pruning strategy in Wu and Er (2000) and Wu et al. (2001). However, they are suitable for SL approaches only and cannot work well without instructive training data. All those pruning methods delete redundant rules which are with small sensitivity or significance. The idea of considering sensitivity and significance of fuzzy rules has been adopted in this paper. Contributions and participation of fuzzy rules are regarded analogously to the significance and sensitivity which are evaluated via UL (including RL) approaches and novel pruning methods without SL are proposed in this paper.

A novel UL algorithm termed Enhanced Dynamic Self-Generated Fuzzy Q-learning (EDSGFQL) is proposed for determining structure and preconditioning parameters of FISs in this paper. Consequent parameters of FISs are estimated following the FQL in Jouffe (1998). An ε-completeness criterion is adopted for clustering the input space and generating fuzzy rules. At the same time, an extended Self Organizing Map (SOM) algorithm is proposed to allocate the centers of fuzzy MFs as an UL approach in the EDSGFQL. Contributions of each fuzzy rules are evaluated through a reinforcement sharing mechanism. The EDSGFQL is capable of automatically generating and pruning fuzzy rules. Furthermore, it can tune both premise and consequent parameters of FISs simultaneously according to the reinforcement signals. Comparative studies on wall-following and obstacle avoidance tasks by a mobile robot demonstrate that the proposed EDSGFQL approach is superior in generating FISs.

The organization of this paper is as follows: The architecture of EDSGFQL system is introduced in Section 2, while the EDSGFQL algorithm for self-generation of FISs is proposed in Section 3. Simulation results and comparison studies with related works are presented in Section 4. Finally, concluding remarks are given in Section 5.

2. Architecture of the EDSGFQL system

In this paper, training of a FIS is based on extended Ellipsoidal Basis Function (EBF) neural networks, which are functionally equivalent to a simple Takagi-Sugeno-Kang (TSK) fuzzy system (Jang, 1993). The structure of the neural-networks-based FIS is depicted in Fig. 1.

Layer one is an input layer and layer two is a fuzzification layer which evaluates the MFs of the input variables. The MF is chosen as a Gaussian function and each input variable x_i $(i = 1, 2, \ldots, N)$ has L MFs given by

$$\mu_{ij}(x_i) = \exp\left[-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right] \quad i = 1, 2 \ldots N,\ j = 1, 2 \ldots, L \qquad (1)$$

where μ_{ij} is the jth MF of x_i, while c_{ij} and σ_{ij} are the center and width of the jth Gaussian MF of x_i respectively. Layer three is a rule layer. The output of the jth rule $R_j (j = 1, 2, \ldots L)$ in layer three is given by

$$f_j(x_1, x_2, \ldots, x_N) = \exp\left[-\sum_{i=1}^{N}\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right] \quad j = 1, 2, \ldots, L \qquad (2)$$

if multiplication is adopted for the T-norm operator.

Normalization takes place in layer four and we have

$$\phi_j = \frac{f_j}{\sum_{i=1}^{L} f_i} \quad j = 1, 2, \ldots, L. \qquad (3)$$

Lastly, nodes of layer five define output variables. If the Center-of-Gravity method is performed for defuzzification, the output variable, as a weighted summation of the incoming signals, is given by

$$y = \sum_{j=1}^{L} \phi_j \omega_j \qquad (4)$$

where ω_j is the consequent weighting parameter and ϕ_j is the normalized firing strength of the jth rule.

A fuzzy rule in this structure is expressed as follows:

If x_1 *is* (c_{1j}, σ_{1j}) *and* x_2 *is* (c_{2j}, σ_{2j}) $\cdots$ *and* x_N *is* (c_{Nj}, σ_{Nj})
Then, y *is* ω_j

where c_{ij} and $\sigma_{ij}, i = 1, 2, \ldots, N$, are the center positions and widths of memberships of the jth fuzzy rule while ω_j is the weight of jth fuzzy rule in the system. The terms c_{ij} and σ_{ij} are commonly known as premise parameters while ω_j denotes consequent parameters.

In generating a FIS, the three issues of major concern are:

- How to determine the number of rules?
- How to allocate and adjust the premise parameters, e.g. center position and width of fuzzy membership functions?
- How to determine the consequent parameters?

3. Self-generated fuzzy inference systems by EDSGFQL

3.1. ε-completeness criterion for rule generation

The ε-completeness criterion is used to determine clustering of the input space. As pointed out by the author of (Juang, 2005), a rule in a FIS corresponds to a cluster in the input space geometrically. An input data with higher firing strength of a fuzzy rule means that its spatial location is closer to the cluster center compared to those with smaller strengths. For any input in the operating range, if there does not exist any fuzzy rules so that the match degree (or firing strength) is no less than ε, more fuzzy rules should be recruited to accomplish the input space. In fuzzy applications, the minimum value of ε is usually selected as $\varepsilon = 0.5$ (Er & Deng, 2004).

Once a new rule is considered, the next step is to assign centers and widths of the corresponding MFs. The incoming multidimensional input vector X is projected to the corresponding one-dimensional MF for each input variable i $(i = 1, 2, \ldots, N)$. Assume that L MFs have been generated in the ith input variable