Neurocomputing 155 (2015) 186–193

Contents lists available at ScienceDirect

Neurocomputing

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

Adaptive structure radial basis function network model for processes with operating region migration

D.K. Siong Tok [a], Ding-Li Yu [a,*], Christian Mathews [a], Dong-Ya Zhao [b], Quan-Min Zhu [c]

[a] Process Control Group, Liverpool John Moores University, Liverpool, UK
[b] Department of Chemical Industrial Equipment and Control Engineering, College of Chemical Engineering, China University of Petroleum, Qingdao, China
[c] Department of Engineering, University of West of England, Bristol, UK

ARTICLE INFO

Article history:
Received 29 April 2014
Received in revised form
2 December 2014
Accepted 12 December 2014
Communicated by: J. Zhang
Available online 22 December 2014

Keywords:
RBF networks
Neural network model
Adaptive structure network
ROLS algorithm
RBF structure adaptation

ABSTRACT

An adaptive structure radial basis function (RBF) network model is proposed in this paper to model nonlinear processes with operating point migration. The recursive orthogonal least squares algorithm (ROLS) is adopted to select new centers on-line, as well as to train the network weights. Based on the *R* matrix in the orthogonal decomposition, an initial center bank is formed and updated in each sample period. A new learning strategy is proposed to gain information from the new data for network structure adaptation. A center grouping algorithm is also developed to divide the centers into active and non-active groups, so that a structure with a smaller size is maintained in the final network model used for output prediction. The proposed RBF model is evaluated and compared with the three existing adaptive structure RBF networks by modeling a nonlinear time-varying numerical example. Simulation results demonstrate that the proposed algorithm has several advantages in term of the adaptive tracking ability and a better recovery speed over the existing methods during the migration of system's operating point.

1. Introduction

The radial basis function network (RBFN) has been successfully applied as a nonlinear function estimator for dynamical system modeling due to its simple architecture and online training ability [1,2]. The RBFN's structures can be classified into two categories: fixed-structure and adaptive structure. For a fixed-structure RBFN, the number and location of centers are fixed during the modeling and operation process and the model parameters (weights) may be adapted. While, an adaptive structure RBFN has the number and location of its hidden layer neurons adapted to better fit the dynamics of the process to be modeled, in addition to the adaptation of the network parameters. In general, it produces a comparatively satisfactory performance. Thus, the performance of an RBFN is heavily dependent on its structure and it is imperative to optimize the RBFN's structure to achieve a satisfactory performance, especially in modeling a highly time-varying process. In order to achieve a satisfactory network performance, a sufficient number of centers is required and there is no prior knowledge to find the exact number of centers that needed [3]. Thus, an unnecessary large RBFN is usually used, which causes numerical ill-conditioning in the training of the network and the worsen generalization of the trained model [4].

In the past decades, the adaptation of RBFN's structures has been intensively investigated. First of all, Platt [5] made a great contribution to the dynamic RBFN's structure by introducing an algorithm called resource allocating network (RAN). For an RAN, the hidden units are gradually inserted into the hidden layer based on the novelty of new data. In a latter attempt, Karayiannis and Min [6] developed a framework for growing RBFNs which merged supervised and unsupervised learning with network growth techniques. They proposed that the structure of network could be gradually constructed by splitting and increasing the prototypes which represented the network centers. However, the insignificant hidden neurons in [5,6] were not pruned which led to a final network with a huge structure. To solve the oversized problem, Lu et al. [7,8] proposed a sequential learning scheme for function approximation using a minimal RBFN which was referred to as minimal RAN (M-RAN). Their pruning strategy was to prune the hidden units that had insignificant contributions to the network performance. However, the optimal network structure achieved in [7,8] is only for a certain data sets, while the performance would be degraded if it is used to predict future behavior in other regions. In recent years, a few methods have been proposed for self-organizing RBFNs [9,10]. Although it was claimed that these methods [9,10] outperformed M-RAN [7] and GGAP-RBF [11], the convergence of their algorithms needed to be investigated carefully for successful applications, which complicates the entire training algorithms. Moreover, there are many unknown parameters in [9,10] which needs preliminary runs to find optimal

* Corresponding author.
E-mail address: D.Yu@ljmu.ac.uk (D.-L. Yu).

http://dx.doi.org/10.1016/j.neucom.2014.12.030
0925-2312/

values for the parameters before the adaptation of network take places.

Orthogonal decomposition is a numerically stable method for solving the least squares problems. Chen and Billings [12,13] proposed a forward regression learning approach based on the batch orthogonal least squares (OLS) algorithm to determine an RBFN's structure. In their approach, the OLS algorithm was employed to determine an appropriate set of centers from a large set of candidate centers. The center was chosen, one by one, until an adequate RBFN's structure was achieved. Chen and Grant [4] further extended this method [12,13] to train a multi-input multi-output (MIMO) RBFN. In addition, Chng et al. [14] extended the work of Chen and Billings [12,13] by introducing a local adaptation process for an RBFN's structure. In the work of Chng et al. [14], the subset models with higher accuracy were achieved compared to [12,13]. The advantage in [12–14] is that the structure and parameters of the RBFN are decided simultaneously by evaluating the contributions of centers to network performance. However, one major drawback is that the optimization of network's weights is of off-line training mode as their methods [12–14] are based on batch OLS algorithm, which means that no new data can be considered during the training process. For online application in training the weights, Yu *et al.* [15] showed that ROLS training algorithm was capable of maintaining the same accuracy of the RBFN model as the off-line training while requiring less computation. Gomm and Yu [3] developed a forward and a backward center selection algorithms using ROLS training algorithm. For the backward selection algorithm, the structure of network is simplified by removing the centers which had smallest contribution to the network performance. On the other hand, for the forward selection algorithm the technique is to build a network by adding centers which will maximally enhance the network performance. Their method [3] resulted in an acceptable level of efficiency and accuracy with a smaller network's size. However, the developed RBFN models in [3] was not 'fully' adaptive as the centers can only be selected from a pre-specified candidate center set. The use of the backward center selection method was extended in [16] to develop an adaptive RBFN model but the performance was not satisfactory due to the lack of efficiency in the selection of centers. In further work, Yu and Yu [17] proposed an adaptive algorithm that incorporated the pruning strategy in [3] to adapt an RBFN model using the ROLS training algorithm. The adding and pruning of centers was based on the error index between the desired and measured modeling performances. New data was added as new center if the desired modeling performance was not achieved. Results showed that a compact RBFN was achieved while the desired modeling performance was maintained. However, in this method the added new centers did not play a role immediately as the performance was degraded for a few sample periods before the positive role is observed during the migration of the process's operating point.

To address these problems, this paper proposes a new algorithm for the adaptation of an RBFN structure for modeling process with operating point migration using ROLS training algorithm. The advantage of this proposed algorithm is that the RBFN is able to be adapted effectively and immediately to fit the new dynamics in the new operating region of the process and achieving a satisfactory overall prediction performance. In this developed algorithm, the RBFN's structure, the number and location of centers, and parameter (weight) are adapted based on the novelty of new data. An initial center bank with a pre-specified number of centers is formed which involves the actions of adding, pruning and grouping of centers. In adding new centers, a new strategy is designed to spread more significant centers in the current operating regions to maximize the network performance. The pruning method in [17] is extended to prune insignificant centers from the center bank. Then, the centers in the center bank are divided into two groups – active center and redundant center groups. The center grouping algorithm is developed using a different criterion that improves the selection of more efficient centers. Active centers are used to predict the process output, while redundant centers are preserved for next sample time. When the process operating point migrates largely, the original centers will not be effective to act for output prediction and the new centers in the region where the operating point moves to will be added. The developed algorithm is evaluated using a nonlinear operating point-migrating numerical example. The effectiveness of the developed algorithm is verified by comparing it with three adaptive structure models. The paper is organized as follows. Section 2 explains the ROLS training algorithm. The adaptation algorithm is presented in Section 3 which includes the adding, pruning and grouping of centers. The evaluation of the developed adaptive RBFN and comparison studies is demonstrated in Section 4.

2. ROLS training algorithm of an RBFN

A standard RBFN, as shown in Fig. 1, has three layers: the input layer, hidden layer and the output layer. The hidden layer consists of hidden neurons and each hidden neuron has a vector called its center. In Fig. 1, $[x_1, \ldots, x_m]$ and $\left[\hat{Y}_1, \ldots, \hat{Y}_p\right]$ are the input and output vectors with their entries being network m inputs and p outputs, respectively.

A non-linear dynamic system is presented by an NARX model in (1).

$$y(k) = f[(y(k-1), \ldots, y(k-n_y), u(k), \ldots, u(k-n_u)] + e(k) \tag{1}$$

where $u \in \Re^m$ and $y \in \Re^p$ are system input and output, and n_u and n_y are input and output orders, respectively. $e \in \Re^p$ is measurement noise. An RBFN is used as an approximate for the nonlinear function in (1), where the RBFN performs a nonlinear static mapping via the linear output transformation [3]. The input vector x of the RBFN includes all variables in function $f(*)$ in (1), while the network output is $\hat{y}$. Here, the Gaussian function is used in the RBFN as the nonlinear basis function in (1).

$$\phi_i(k) = \exp\left(-\frac{\|x(k) - c_i^2\|}{\sigma_i^2}\right), \; i = 1, \ldots, n_h \tag{2}$$

where $\phi(k)$ is the hidden layer output, n_h is the number of hidden layer nodes (center); $x(k)$ is the network input vector and c_i is the ith center with $i = 1, \ldots, n_h$. The network output is the weighted sum of the hidden layer output and is given by,

$$y(k) = W\phi$$

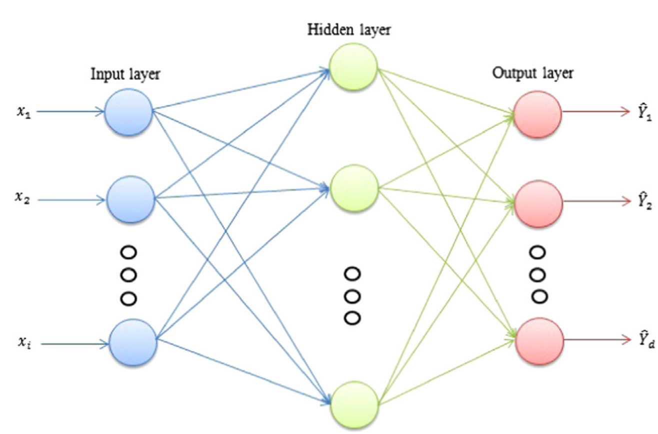

Fig. 1. The structure of an RBFN.