Chin. J. Chem. Eng., **15**(2) 240—246 (2007)

Splicing System Based Genetic Algorithms for Developing RBF Networks Models*

TAO Jili(陶吉利) and WANG Ning(王宁)**
National Laboratory of Industrial Control Technology, Institute of Advanced Process Control, Zhejiang University, Hangzhou 310027, China

Abstract A splicing system based genetic algorithm is proposed to optimize dynamical radial basis function (RBF) neural network, which is used to extract valuable process information from input output data. The novel RBF network training technique includes the network structure into the set of function centers by compromising between the conflicting requirements of reducing prediction error and simultaneously decreasing model complexity. The effectiveness of the proposed method is illustrated through the development of dynamic models as a benchmark discrete example and a continuous stirred tank reactor by comparing with several different RBF network training methods.
Keywords RBF network, structure optimization, genetic algorithm, splicing system

1 INTRODUCTION

Radial basis function (RBF) networks attracted considerable interest in the past because of its several advantages compared with other types of artificial neural networks (ANNs), such as better approximation capabilities, simpler network structures, and faster learning algorithms[1]. However, the selection of appropriate number of basis functions is a critical issue for RBF networks[2]. The number of basis functions controls the complexity of the structure , *i.e.*, the generalization capability of RBF networks. A RBF network, containing very few basis functions, yields poor predictions on new data, *i.e.*, poor generalization, as the model has limited flexibility. The RBF network, containing several basis functions, also yields poor generalization, as it is too flexible and fits the noise in the training data. The best generalization performance is obtained *via* the compromise between the conflicting requirements of simultaneously reducing the prediction error and decreasing the complexity of the model. This trade-off highlights the importance of optimizing the complexity of RBF network to achieve the best generalization.

More specifically, most of the standard RBF training methods require the designer to fix the network structure. These training procedures usually proceed *via* two steps[3]: First, the centers of basis function are determined using clustering method. Second, the calculation of the final-layer weights is reduced to solve a simple linear system using least squares method. Therefore, the first stage is an unsupervised method, and separated from the actual objective to minimize the output prediction error. In this study, the RBF networks are constructed using the input data supervised by the output data.

The inclusion of the structure selection in the formulation of the network optimization problem is desirable, but it results in a rather difficult problem, which cannot be easily solved using the standard optimization methods. An interesting alternative for solving this complicated problem is offered by the use of the recently developed evolutionary computation methods. Perhaps the most popular and successful strategies are the so-called genetic algorithms (GAs), which are stochastic methods based on the principles of natural selection and evolution[4]. GAs have proved to be successful in the structure selection of several types of neural networks, such as BP neural networks[5,6] and recurrent neural networks[7,8]. As to the optimization of RBF networks, Vesin and Gruter used GA to solve the complete optimization problem, but the centers of the potential nodes were restricted among the set of training data[9]. Esposito *et al.* employed a GA based technique to determine the widths of Gaussian functions in RBF networks[10], whereas Sarimveis *et al.* used GA approach to optimize the parameters of RBF networks in terms of the error minimization criterion[11].

In this study, the structure selection is included, and the fitness of each chromosome is calculated on the basis of the prediction error and the structure complexity criterion. To simplify the optimization of RBF network, the radial basis function is chosen as thin-plate-spline function[12], where the determination of widths is not required. Therefore, the GA in this study is used to determine the centers of basis functions and the network structure. The final-layer weights are derived using recursive least squares (RLS) method with the same initial weight vector. The proposed algorithm starts with a random population of RBF networks, which are coded as chromosomes. As all the function centers generated by stochastic chromosomes are not feasible, two novel operators, *i.e.*, elongation and deletion, enlightened by DNA splicing system[13,14], are introduced in the GA approach.

2 SPLICING SYSTEM BASED GA FOR RBF NETWORKS

Generally, the determination of the RBF centers

Received 2006-04-11, accepted 2006-11-01.
* Supported by the National Natural Science Foundation of China (No.60421002), and the National High Technology Research and Development Program of China (863 Program, 2006AA040308).
** To whom correspondence should be addressed. E-mail: nwang@iipc.zju.edu.cn

is based on a self-organizing clustering approach, such as k-means clustering[15], the nearest neighbor clustering method[16]. The application of the above algorithms requires the transcendental knowledge of an appropriate clustering degree which is difficult to determine, and it considers only the input data. The proposed approach in this study does not require the transcendental knowledge of the plant; moreover, the structure and RBF centers can be synchronously optimized by utilizing the input output data.

GA is an optimization algorithm on the basis of Darwinism, which is very flexible. Depending on the features of the problem's solution space, there is a wide range of choices of fitness functions, the coding method, and the genetic operations, and all these factors affect the efficiency of genetic algorithm. This study is focused on the optimization of RBF network using the splicing system based GA.

2.1 Coding method

There are totally $n_r \times n$ real number parameters to be optimized in the RBF network, which means one chromosome should be able to give $n_r \times n$ real number values, where n_r is the number of hidden nodes, n is the number of input nodes. Hence, binary coding chromosome will become very complex, and decimal coding chromosome is used. The structure of the lth chromosome is shown below

$$\boldsymbol{C}_l = \begin{bmatrix} c^l_{1,1} & c^l_{1,2} & \cdots & c^l_{1,n} \\ c^l_{2,1} & c^l_{2,2} & \cdots & c^l_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c^l_{n_r,1} & c^l_{n_r,2} & \cdots & c^l_{n_r,n} \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{1}$$

where $l=1, 2, \cdots, L$, L is the size of the population, n_r is randomly produced between 1 and D, D is the maximum number of hidden nodes, the rows below n_r are set to zeros and do not correspond to the center. The elements of $\mathbf{C}_l$ are computed using the following equation:

$$c_{lj} = x_{j,\min} + r \cdot (x_{j,\max} - x_{j,\min})$$
$$1 \leqslant l \leqslant n_r, \quad 1 \leqslant j \leqslant n \tag{2}$$

where r is the random number between 0 and 1, $x_{j,\min}$ and $x_{j,\max}$ is the minimum and the maximum values of input variables given in the problem.

2.2 Fitness function

As mentioned in the above sections, the drawbacks of the general training methods of RBF network mainly lie in the absence of global optimization of both the approximation capability and the generalization performance. To overcome these drawbacks, the choice of appropriate fitness function is crucial.

In this study, the training procedures using splicing system based GA are also preceded in two steps: First, the network structure and the basis function parameters are determined using the chromosomes of one population. Second, the final-layer weights are calculated using least squares method. As the direct least squares method cannot obtain the solutions for a bad-conditioned matrix, the output weights of the lth RBF neural network are calculated using the following RLS method[17]:

$$\boldsymbol{w}(k) = \boldsymbol{w}(k-1) + \boldsymbol{K}(k)[y(k) - \boldsymbol{X}_r^{\mathrm{T}}(k)\boldsymbol{w}(k-1)]$$
$$\boldsymbol{K}(k) = \boldsymbol{P}(k-1)\boldsymbol{X}_r(k)[\boldsymbol{X}_r^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{X}_r(k)+1]^{-1}$$
$$\boldsymbol{P}(k) = \boldsymbol{P}(k-1) - \boldsymbol{K}(k)\boldsymbol{K}^{\mathrm{T}}(k)[\boldsymbol{X}_r^{\mathrm{T}}(k)\boldsymbol{P}(k-1)\boldsymbol{X}_r(k)+1] \tag{3}$$

where $1 \leqslant k \leqslant N$, N is the maximum iterative time, $\boldsymbol{X}_r(k)$ is the n_r^l dimension output vector of the hidden layer, $y(k)$ is the output of the actual system, $\boldsymbol{K}(k)$ is the n_r^l dimension assistant vector, $\boldsymbol{P}(k)$ is the n_r^l-by-n_r^l assistant matrix. From Eq.(3), the computational complexity of RLS solution for one iterative time is obtained as $O\left(\left(n_r^l\right)^2\right)$. Hence, the complexity of RLS solution for the lth network weight vector is $O\left(N\left(n_r^l\right)^2\right)$.

In every generation of GA, the calculation of the output weights completes the formulation of L RBF networks, which can be represented by the pairs ($\boldsymbol{C}_1$, $\boldsymbol{w}_1$), ($\boldsymbol{C}_2$, $\boldsymbol{w}_2$),$\cdots$ and ($\boldsymbol{C}_L$, $\boldsymbol{w}_L$). To obtain good generalization capability of RBF networks, the training data are divided into two groups, one group of data ($\boldsymbol{X}_1$, $\boldsymbol{Y}_1$) are used to calculate the final-layer weights, herein, $N=N_1$ (N_1 is the number of the first group data), and the other group of data ($\boldsymbol{X}_2$, $\boldsymbol{Y}_2$) are utilized to evaluate the produced RBF networks in each generation. This scheme incorporates a testing procedure into the training algorithm, and guarantees good generalization performance of the RBF networks. However, to obtain good approximation capability of RBF networks, the network structure still becomes much complex. The structure complexity of RBF network conflicts with the generalization performance of neural networks. Therefore, the objective function considering both approximation capability and generalization performance is shown as follows.

$$J(\boldsymbol{C}_l, \boldsymbol{w}_l) = N_2 \ln\left[\sum_{t=1}^{N_2} |Y_2(t) - \hat{Y}_2(t)|\right] + \eta n_r^l \ln(N_2) \tag{4}$$

Equation (4) expresses a compromise between the cost of modeling errors and the complexity of network structure[13], where $\hat{Y}_2(t)$ is the output of RBF network, N_2 is the number of the second group data, n_r^l is the number of hidden nodes in the lth chromosome, η is the weight coefficient, and $0 \leqslant \eta \leqslant 1$, the greater the η is, the stronger constraint of structure complexity would be considered. In this study, η is set as 1.