Proceedings of the 35th
Conference on Decision and Control
Kobe, Japan • December 1996

Designing a Soft Sensor for a Distillation Column with the Fuzzy Distributed Radial Basis Function Neural Network

Xudong Wang Rongfu Luo Huihe Shao
Automation Department
Shanghai Jiao Tong University
Shanghai, 200030
People's Republic of China

Abstract: A soft Sensor is a model which is used to estimate the unmeasurable output of an industrial process. It is very useful in process control because it can be used to control and monitor many industrial processes. But designing a soft sensor is usually difficult because its modeling is often based on case data. These data have the features of discreteness, nonlinearity, contradiction, and complexity. In this paper, modeling based on case data is defined as a case based modeling problem. In order to solve the case based modeling problem and successfully design a soft sensor, this paper constructs a kind of fuzzy distributed radial basis function neural network. The fuzzy distributed RBF neural network is easy to solve the case based modeling problem. In this paper, it is applied in designing a soft sensor for a high purity distillation column. The simulation is based on the actual operation data and analysis data of the distillation column. The results show that the fuzzy distributed RBF neural network based soft sensor has good performance. Thus, the fuzzy distributed RBF neural network has successfully solved the case based modeling problem. It is very promising in process control.

I. INTRODUCTION

In many industrial processes, some process outputs can not be measured by a sensor. But these outputs are essential to control and monitor the industrial processes. For example, the product quality of a CSTR(continuous stirred tank reactor) can not be measured on line. It is often analyzed through the chemical experiment. The product composition of a distillation column also can not be measured by a sensor. It is often analyzed by a gas chromatographic (GC) instrument. Both chemical analysis and GC analysis take a lot of time, so controlling and monitoring a CSTR or a distillation column are very difficult. In order to solve this problem, a process model should be designed to estimate the unmeasurable process output. Such a model is a soft sensor which can indirectly measure the unmeasurable process output. Because the data for designing a soft sensor are the operation data and analysis data of an industrial process, they have the features of discreteness, nonlinearity, contradiction, and complexity. These data are like the cases. In this paper, the set of these data is defined as a case data set and modeling based on these data is called a case based modeling problem. A model based on case data is not an dynamic model. So designing a soft sensor is different from modeling a dynamic system. The methods for modeling a dynamic system are not fit for solving the case based modeling problem. They can not design a soft sensor. In this paper, a new kind of feedforward neural network is constructed to solve the case based modeling problem. Such a neural network is based on the general RBF neural network.

A RBF neural network has the characteristics of universal approximation [1] and best approximation [2][3]. It can be used to perform a curve fitting operation in a multidimensional space. Although it is highly nonlinear in parameters, its learning procedure has no local minimum problem [4]. This is the learning advantage of a RBF neural network. The output of a RBF neural network is linear to the weights in the network. This is the structure advantage of a RBF neural network. Because of the two advantages, a RBF neural network is promising in application. But when solving the case based modeling problem, a RBF neural network is not ideal. The case based modeling problem is characterized by that the training data is a set of independent cases. The case data more or less contradict each other. The relation between the input data and the output data in the case data set is strongly nonlinear and the number of cases is usually very large. Because of such difficulties, the general RBF neural network can not accurately learn the case data. Even if the case data could be learned , the number of neurons in the neural network would be so large that the learning process is computationally complicated and the performance of the network is bad.

In order that the case data can be learned accurately, it can be hierarchically divided into subsets so that they can be easily learned by the general RBF neural networks. This is the idea of hierarchy clustering. The clustering algorithm is based on the competitive learning. Because rival penalized competitive learning algorithm [5] has

0-7803-3590-2/96 $5.00 © 1996 IEEE

many advantages, it is adopted in this paper. Because there is a problem in RPCL, it is modified in this paper. Based on the modified RPCL, hierarchy clustering can proceed easily. The criterion for judging whether or not a subset should be further divided is that if this subset can be easily learned by a general RBF neural network. After the whole case data set is clustered by hierarchy clustering and learned by RBF neural networks, the distributed RBF neural network is formed. In order to synthesize the distributed neural network, a fuzzy classifier is incorporated in the network. Although the fuzzy distributed neural network is composed of several general RBF neural networks, its integrity is assured by the fuzzy classifier.

The fuzzy distributed RBF neural network is very useful in modeling the industrial process in which the input-output data that can be acquired for modeling are case data. So such a neural network is promising in process control. It can be used to design a soft sensor for an industrial process. In this paper the fuzzy distributed neural network is used to design a soft sensor for a high purity distillation column. The data in simulation is actual operation data and analysis data of a distillation column in Shi Jia Zhuang refinery. The results of simulation show that the fuzzy distributed neural network is efficient

II. THE GENERAL RBF NEURAL NETWORK

A schematic of the general RBF neural network is depicted in Fig. 1.

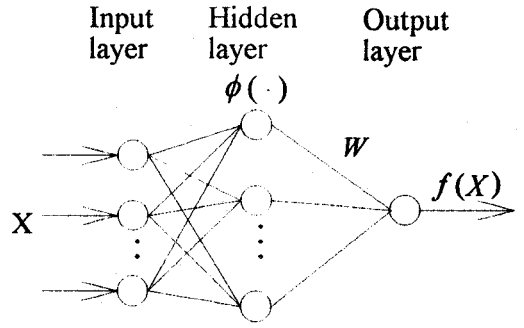

Fig. 1. The schematic of a general RBF neural network

Such a network implements a mapping $f: R^n \to R$ according to the following general form:

$$f(X) = \sum_{i=1}^{M} W_i \cdot \phi([X - C_i]^T \Sigma^{-1} [X - C_i]) \quad (1)$$

where $X \in R^n$ is the input vector, $\phi(\cdot)$ is a kind of radial basis function, Σ is a positive matrix that controls the receptive field of the function $\phi(\cdot)$, W_i, $1 \le i \le M$ are the weights, $C_i \in R^n$, $1 \le i \le M$ are the RBF centers, M is the number of centers or the number of neurons in the hidden layer.

For the RBF neural network whose basis function $\phi(\cdot)$ is fixed, there are three sets of parameters should be determined: (1) the W_i, $1 \le i \le M$, which are the weight vectors of the RBF net; (2) the centers C_i, $1 \le i \le M$; (3) the matrix Σ. For convenience , Θ is used to denote the vector consisting of all these parameters. Given a sample set $D_N = \{X_i, Y_i\}_1^N$, a specific value of Θ can be usually decided by the following minimization [6]:

$$\varepsilon_{RBF}^2(D_N, \hat{f}) = \min_{f \in F}(\varepsilon_{RBF}^2(D_N, f)$$

$$= \min_{\Theta} \varepsilon_{RBF}^2(D_N, f(X, \Theta)) \quad (2)$$

$$\varepsilon_{RBF}^2(D_N, f(X, \Theta)) = \frac{1}{N}\sum_{i=1}^{N} |Y_i - f(X_i, \Theta)|^2$$

where F is the set of function f. Each specific value of Θ specifies a function f in the set F.

The RBF neural network has three types according to its training method [6]. The types are defined as follows:

Type 0: All the parameters of such a RBF neural network is determined by the minimization of equation (2). This type is also called ideal-type.

Type I: The parameter Σ is prespecified and the centers C_i are randomly selected from D_C^X. This type is also called basic-type.

Type II: The parameter Σ is prespecified and the centers C_i are determined by some clustering algorithm. This type is also called clustering-aided-type.

As a type 0 RBF neural network is not efficient in training, the widely used RBF neural network is almost type I and type II. For some examples, Poggio and Girosi's algorithm [7] is based on type I. Local training algorithm [8] and orthogonal least squares (OLS) algorithm [9] are based on type II.

The OLS learning algorithm is a constructing algorithm. The structure of a RBF neural network is gradually formed during learning. The number of neurons in hidden layer is determined after training. Because of this advantage, the OLS algorithm is adopted to train the RBF neural network in this paper. The computational procedure of OLS algorithm is described in detail in S. Chen's paper [9]. Although the OLS learning algorithm has the advantage of determining the hidden neuron number, it exists two problems. First is how to efficiently determined the RBF centers. Second is the performance of the RBF neural network could be deteriorated by over-fitting in training. These two problems have been analyzed in paper [10]. In order to solve the first problem, the rival penalized competitive learning is used to calculated the RBF centers. The second problem can be solved by constraining the module of orthogonal vector. The module should be larger than a little positive real number.

III. THE CASE BASED MODELING PROBLEM

A case based modeling problem often exists in industrial processes. Its definition is given as follows:

Definition: When modeling a system, the input and output data of the system at different sampling time are independent. Some data contradict each other. These

independent data are called cases. The set of these data is called a case data set. Modeling based on a case data set is defined as the case based modeling problem.

Here, a case is different from a discrete event. Although the case data are discrete, the case based modeling problem exists not only in a discrete event system but also in a continuous system. The following is a example of the case based modeling problem.

Example: The product composition of a distillation column in a refinery has to be estimated through a process model because it can not be measured through a sensor. The data for designing the process model are operation data (process input) and product composition (process output). Usually the product composition is analyzed through a GC instrument. GC analysis takes so much time that the data that can be gotten at different sampling time are independent and discrete. They have no cause and effect relation. Thus the dynamic model such as ARMA model can not be designed based on these data.

The example shows that the characteristics of the case based modeling problem is that not the modeled system but the data for modeling is discrete. The case based modeling problem has four features:

1. **Discreteness:** The data for modeling a system is discrete;
2. **Nonlinearity:** The modeled system is nonlinear. So the relation between the input data and output data are nonlinear;
3. **Contradiction:** Some of the input-output data contradict each other;
4. **Complexity:** The number of the data is very large, so modeling based on these data is very complex.

In a case based modeling problem, the data set for modeling is so complex that it can hardly be learned by a neural network. So it had better be divided into subsets by some clustering algorithm.

IV. MODIFIED RIVAL PENALIZED COMPETITIVE LEARNING

Competitive learning is a kind of adaptive K-means clustering algorithm. It has four important kinds: unsupervised competitive learning (UCL), frequency sensitive competitive learning (FSCL) [11], rival penalized competitive learning (RPCL) [5], and fuzzy competitive learning (FCL) [12].

UCL has the problem that some units are under-utilized. These units are called dead units. In order to solve the dead unit problem, several techniques can be used to improve UCL. A notable improvement is the strategy of reducing the winner rate of the frequent winners, sometimes called "conscience". FSCL is a good example of this strategy. Although FSCL does solve the dead unit problem, it encounters another critical problem: the selection of an appropriate number of units needed. For tackling this problem, the rival penalized competitive learning is proposed [5]. It is developed by adding a new mechanism into FSCL. The basic idea is that for each input, not only the weights of the winner unit are modified to adapt to the input, but also the weights of its rival (the second winner) are delearned by a smaller learning rate. After learning, RPCL can delete the redundant units. Thus it can appropriately select the needed number of competitive units. But there is a serious problem hidden in RPCL. RPCL combines rival penalizing with FSCL, but in fact rival penalized mechanism and FSCL sometimes contradict each other. The rival penalized mechanism tries to push the rival far away from the cluster towards which the winner is moving. And FSCL ensures that the failed units including the rival have more chances to win than the winner at the next competition. Sometimes it pushes the rival back to the cluster. So FSCL is somewhat inconsistent with the rival penalized mechanism. If the contradiction is persistent, the redundant units can not be deleted and the learning process vibrates.

In order to end the vibration and delete the redundant units efficiently, a modified RPCL algorithm is put forward. The basic idea is that the rival penalized mechanism is not parallel with FSCL in competitive learning. The procedure of the modified RPCL is described in [13].

V. HIERARCHY CLUSTERING

A case data set can be correctly clustered into several subsets with the modified RPCL, but these subsets still have the characteristics of the case based modeling problem. They can not be learned accurately by general RBF neural networks as well as other kind of neural networks. In order to solve this problem, these subsets should be further clustered through the modified RPCL algorithm. This is the idea hierarchy clustering.

Given a case data set $S = \{X_i, i = 1, \cdots, N\}$, X_i is a sample or a case, N is the number of cases. Using the modified RPCL, S can be divided into M subsets $S_i = \{X_j, j = 1 \cdots N_i\}$, $i = 1 \cdots M$ where N_i is the number of cases in the ith subset. This is the first step of hierarchy clustering. After this step, the subsets that can not be learned by the general RBF neural networks should be further clustered. This is the second step of hierarchy clustering. If the clustering process continues step by step, the whole set S can be divided into the subsets which can be learned easily by the general RBF neural networks. Fig. 2 is the process of hierarchy clustering.

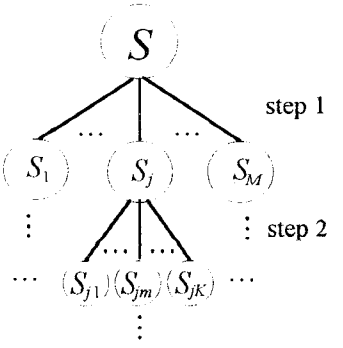

Fig. 2. The process of hierarchy clustering

Combining the modified RPCL algorithm with hierarchy clustering method, a case data set can be decomposed into several subsets which are without the difficulties of the case based modeling problem. After hierarchy clustering, the case data set S can be easily learned by several RBF neural networks.

VI. THE FUZZY DISTRIBUTED RBF NEURAL NETWORK

The OLS algorithm is used to learn the subsets. After the general RBF neural networks have learned the subsets of a case data set, a distributed RBF neural network for the case data set is formed. Such a neural network can not be applied because each RBF neural network in it is independent. If given a new input, the distributed neural network can not determined its output. So it should be synthesized by some techniques. Fig. 3 is one of the methods which synthesize the distributed RBF neural network. The fuzzy classifier is a synthesizing tool.

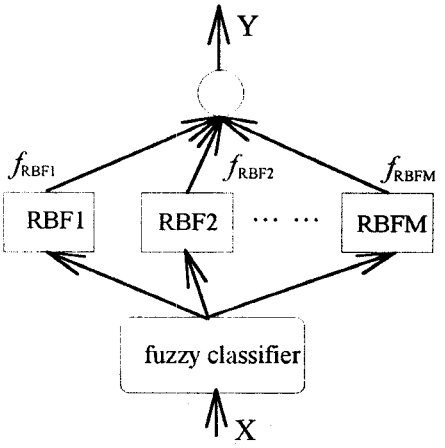

Fig. 3. The fuzzy distributed RBF neural network

The membership μ_i can be calculated as follows: First, the membership of X to each sample of the data set $S=\{X_i, i=1,\cdots,N\}$ is calculated. The membership to sample X_i is denoted by η_i. Second, μ_i is calculated on the basis of $\eta_i, i=1,\cdots,N$. After η_i is determined, μ_i can be calculated through equation(3).

$$\mu_i = \sum_{j=1}^{N_i} \eta_j \quad (3)$$

where N_i is the number of samples for the ith RBF neural network $RBFi$.

After the distributed RBF neural network is synthesized by a fuzzy classifier, it has successfully solved the case based modeling problem and the model of a system is designed. The fuzzy distributed neural-net based model not only can remember the learning data but also can estimate the output of the modeled system when given a new input.

Suppose that the distributed RBF neural network has M general RBF $\{RBFi, i=1,\cdots,M\}$, and that the membership of a new input X to the ith general RBF neural network $RBFi$ is μ_i, the output Y of the distributed RBF neural network is as equation (4) shows.

$$Y = \sum_{i=1}^{M} \mu_i \cdot f_{RBFi}(X) \quad (4)$$

VII. DESIGNING A SOFT SENSOR FOR A DISTILLATION COLUMN

The fuzzy distributed RBF neural network can solve the case based modeling problem. It is useful in process control because modeling of many industrial processes is based on case data. For example, designing a soft sensor for a high purity distillation column is based on the operation data (process input data) and GC analysis data (process output data). These data are independent and can be viewed as cases. Modeling based on these data is difficult. So in this paper, the fuzzy distributed RBF neural network is adopted

Fig. 4 is a distillation column of Shi Jia Zhuang refinery in Hubei province.

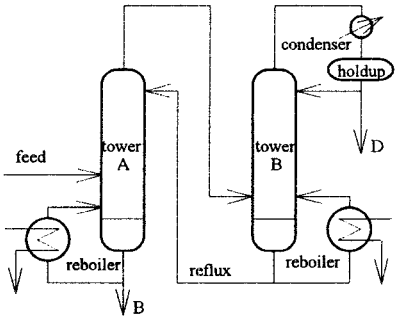

Fig. 4. The high purity distillation column

The product B at the bottom of tower A is mainly of propane. The product D at the top of tower B is mainly of propylene. In the refinery, the composition of propane in product B (denoted by C_B) and composition of propylene in product D (denoted by C_D) are analyzed by a GC instrument. The analysis time is so long that C_B and C_D can not be measured on line. So controlling and monitoring of such a distillation column are very difficult. In order to solve this problem, a industrial process model for estimating C_B and C_D should be designed. This model is called a soft sensor.

The procedure of designing a soft sensor for an industrial process has three steps [14]. First is selecting the secondary measurements of the process, second is collecting data and processing data, and third is modeling the process based on the selected secondary measurements and processed data.

Three secondary measurements are selected to design a soft sensor for the distillation column in Fig. 4. They are ΔT_A which is the differential temperature of two measuring points in tower A, ΔT_B which is the differential temperature of two measuring points in tower

B, and p which is the pressure at the top of tower B. So the soft sensor can be described as equation (5).

$$\begin{cases} C_B = f_1(\Delta T_A, \Delta T_B, p) \\ C_D = f_2(\Delta T_A, \Delta T_B, p) \end{cases} \quad (5)$$

Collecting the operation data and analysis data of the distillation column, a data set for modeling can be acquired. In Shi Jia Zhuang refinery, the product of the distillation column is analyzed every other two hours and only a few data can be gotten on each day. So the data in two months are collected.

Before modeling, the data should be normalized. In this paper , the normalization is as follows:

$$X_i' = (X_i - \min(X)) / (\max(X) - \min(X)) \quad (6)$$

where X_i' is the normalized X_i, X is the data sequence.

The normalized data set should be clustered and divided into several subsets through the hierarchy clustering algorithm according to the output data. After the first step of hierarchy clustering, the data set is divided into 5 subsets. Fig. 5 shows the result of the first step of hierarchy clustering. After the data set is hierarchically clustered, it is divided into 25 subsets. These subsets are learned by 25 general RBF neural network separately.

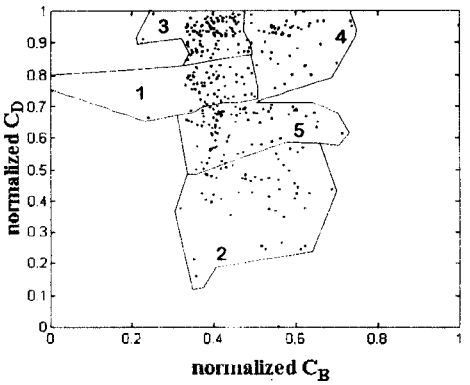

Fig. 5. The result of hierarchy clustering in the first step

After the data set is hierarchically clustered and all the subsets are learned, a whole distributed RBF neural network is constructed. The soft sensor can be successfully designed by incorporating a fuzzy classifier in such a distributed neural network.

Table 1 is the comparison of the product composition C_B and C_D between the process output and the estimation of the soft sensor. It shows that the fuzzy distributed RBF neural network based soft sensor is ideal in estimating the product composition C_B and C_D.

Table 2 is the estimation standard deviation of the fuzzy distributed RBF neural network based soft sensor and Table 3 is the estimation standard deviation of the Kalman filter based estimator. Their comparison shows that the fuzzy distributed RBF neural network is rather good at designing a soft sensor than the Kalman filter based algorithm.

Table 1 The comparison between the output of soft sensor and that of the process. (The unit of C_B and C_D is mole fraction.)

C_D of soft sensor	C_B of soft sensor	C_D of process	C_B of process
99.432%	85.100%	99.444%	85.638%
99.426%	79.150%	99.413%	78.967%
99.435%	96.870%	99.428%	96.074%
99.105%	79.600%	99.210%	80.592%
99.140%	93.050%	99.254%	92.024%

Table 2 The estimation standard deviation of fuzzy distributed RBF neural network based soft sensor

process output	standard deviation
propylene composition C_D	0.0133% (mole fraction)
propane composition C_B	1.0755% (mole fraction)

Table 3 The estimation standard deviation of Kalman filter based estimator

process output	standard deviation
propylene composition C_D	0.2028% (mole fraction)
propane composition C_B	1.9310% (mole fraction)

VIII. CONCLUSION

This paper puts forward a fuzzy distributed RBF neural network to solve the case based modeling problem. Applications in designing a soft sensor for a distillation column shows that such a fuzzy distributed neural network is promising.

REFERENCES

[1] Moshe Leshno, Vladinier Ya Lin, Allan Pinkus and Shimon Schocken, " Multilayer feedforward networks with a nonpolynomial activation function can approximate any function, " Neural Networks, vol. 6 pp.861-867, 1993

[2] F. Girosi and T. Poggio, " Networks and the best approximation property, " Biological Cybernetics, vol. 63, pp.169-176, 1990.

[3] Jonathan Wray and Gray G.R. Green, " Neural networks, approximation theory and finite precision

computation, " Neural Networks, vol. 8, no. 1, pp.31-37, 1995.

[4] Monica Bianchini, Pnolo Frasconi, Marco Gori, " Learning without local minima in radial basis function networks, " IEEE Trans. on Neural networks, vol. 6, no. 3, pp.749-755, 1995.

[5] Lei Xu, Adam Krzyzak, Erkhi Oja., " Rival penalized competitive learning for clustering analysis, RBF net, and curve detection, " IEEE Trans. on Neural Networks, vol. 4, no. 4, pp.636-649, 1993.

[6] Lei Xu, Adam Krzyzak, and Alan Yuille, " On radial basis function nets and kernel regression: statistical consistency, convergence rates and receptive field size, " Neural Networks, vol. 7, no. 4, pp.609-628, 1994.

[7] T. Piggio, F. Girosi, " A theory of networks for approximation and learning, " A.I. Memo No 1140. Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge Mass. 1989.

[8] J. Moody and C. Darken. " Fast learning in networks of locally-tuned processing units, " Neural Computation, vol. 1, pp.281-294, 1989.

[9] S. Chen, C.F.N. Cowan , and P.M. Grant, " Orthogonal least squares learning algorithm for radial basis function networks, " IEEE Trans. on Neural Networks, vol. 2, no. 2, pp.302-309, 1991.

[10] Xudong Wang, Huihe Shao, and Zhongjun Zhang, " Modeling of nonlinear systems based on the radial basis function neural network, " Submitted to IEEE Trans. on Neural Networks (revised).

[11] S. C. Ahalt, A. K. Krishnamurty, P. Chen, and D. E. Melton, " Competitive learning algorithms for vector quantization, " Neural Networks, vol. 3, pp.277-291, 1990.

[12] Fulai Chung and Tong Lee, " Fuzzy competitive learning, " Neural Networks, vol. 7, no. 3, pp.539-551, 1994.

[13] Xudong Wang, Huihe Shao, " An Improved Rival Penalized Competitive Learning Algorithm " , Submitted to Pattern Recognition and Artificial Intelligent (in Chinese)

[14] Rongfu Luo, Huihe Shao, and Zhongjun Zhang, " Fuzzy-neural net-based inferential control for a high-purity distillation column, " Control Engineering Practice, vol. 3, no. 1, pp.31-40 1995