Dimensionality Reduction with Input Training Neural Network and Its Application in Chemical Process Modelling*

ZHU Qunxiong(朱群雄) and LI Chengfei(李澄非)**
School of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China

Abstract Many applications of principal component analysis (PCA) can be found in dimensionality reduction. But linear PCA method is not well suitable for nonlinear chemical processes. A new PCA method based on improved input training neural network (IT-NN) is proposed for the nonlinear system modelling in this paper. Momentum factor and adaptive learning rate are introduced into learning algorithm to improve the training speed of IT-NN. Contrasting to the auto-associative neural network (ANN), IT-NN has less hidden layers and higher training speed. The effectiveness is illustrated through a comparison of IT-NN with linear PCA and ANN with experiments. Moreover, the IT-NN is combined with RBF neural network (RBF-NN) to model the yields of ethylene and propylene in the naphtha pyrolysis system. From the illustrative example and practical application, IT-NN combined with RBF-NN is an effective method of nonlinear chemical process modelling.

Keywords chemical process modelling, input training neural network, nonlinear principal component analysis, naphtha pyrolysis

1 INTRODUCTION

In the chemical process field, many analytical or measuring instruments can easily acquire values of many process variables in a very short period of time. In this way, one has to face multidimensionality problems. The multidimensionality complicates the data interpretation, increases the complexity of the constructed models and even worse, makes data visualization difficult. Therefore, the first step in chemical process modelling is to reduce data dimensionality. There are many methods which help to achieve this goal. The best known is principal component analysis (PCA)[1]. However, linear PCA by definition cannot deal efficiently with nonlinearly correlated variables. Nonlinear PCA is the best remedy to overcome such deficiency, which deals with both types of relations between variables: linear and nonlinear ones. Most nonlinear PCA methods, which are developed in recent years, are the extensions of linear PCA. The difference between linear PCA and nonlinear PCA is that projection function of linear PCA is linear while that of nonlinear PCA is nonlinear. To trace the past, the most common nonlinear PCA methods are principal curves or principal surface[2], auto-associative neural network (ANN)[3], and input training neural network (IT-NN)[4—7]. However, principal curves or principal surface can only represent several input linear models and cannot be commonly applied to all nonlinear systems. The ANN compresses the original variables to few features, and therefore it is possible to visualize the data set structure in form of the continuous projections. Additionally, it is possible to predict the properties of new samples[8] once the network is trained. But ANN has a complicate net structure, one input layer, one output layer and 3 hidden layers (mapping layer, bottleneck layer, de-mapping layer), which makes it much difficult to be trained. ANN is a kind of error back propagation forward neural network. Generally, the performance of back propagation deteriorates while the number of hidden layers get larger. To overcome this limitation of ANN, Tan and Mavrovouniotis presented a 3-layer IT-NN composed of input layer, de-mapping layer, and output layer. Three layers of IT-NN are analogical to layers of ANN: bottleneck layer, de-mapping layer, and output layer. Since the simplified neural network has only one hidden layer, IT-NN is easier to train than ANN. For these reasons, IT-NN seems to be a very attractive alternative to the well-known nonlinear approaches. Here, we explore the basic theory of IT-NN being used as a nonlinear PCA method, present its applications in the data reduction of the nonlinear system, and make the comparison among linear PCA, ANN, and IT-NN. The main advantages of IT-NN are illustrated on experimental data set.

This article also studies the improvement of the learning algorithm. We introduced momentum factor and adaptive learning rate in the learning algorithm of IT-NN, which restrain the vibrancy and accelerate the convergence of training IT-NN. The effectiveness of the improved learning algorithm is illustrated by its comparison with original learning algorithm when training an IT-NN to gain nonlinear principal components with same dimensions. This paper represents a

Received 2005-05-31, accepted 2006-05-15.

* Supported by Beijing Municipal Education Commission (No.xk100100435) and the Key Research Project of Science and Technology from Sinopec (No.E03007).

** To whom correspondence should be addressed. E-mail: zhuqx@mail.buct.edu.cn new nonlinear system modelling method by combining IT-NN with RBF neural network (RBF-NN), which takes both advantages of the two neural networks: the effectiveness of IT-NN in dimensionality reduction and the goodness of RBF neural network in input and output fast mapping. This method is applied to simulate cracking furnace and predict the yields of ethylene and propylene. The simulation in section 5 shows that this nonlinear system model can predict the yields of ethylene and propylene.

2 METHODS OF DIMENSIONALITY REDUCTION

2.1 Linear PCA

For m-D matrix, $\boldsymbol{X}=[\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m]$, linear PCA can find a reversible linear transformation, which maps data from higher space $\boldsymbol{X}$ to a lower space $\boldsymbol{T}$. The whole transformation can be described as follows:

$$\boldsymbol{X} = \boldsymbol{t}_1\boldsymbol{p}_1^{\mathrm{T}} + \boldsymbol{t}_2\boldsymbol{p}_2^{\mathrm{T}} + \cdots + \boldsymbol{t}_r\boldsymbol{p}_r^{\mathrm{T}} + \boldsymbol{E} \tag{1}$$

$\boldsymbol{p}_i$ is an eigenvector of matrix $\boldsymbol{X}$, $\boldsymbol{t}_i$ is the ith principal component, and $\boldsymbol{E}$ is the residual. More generally, we can write:

$$\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\mathrm{T}} + \boldsymbol{E} \tag{2}$$

where $\boldsymbol{T}$ is defined as principal component scores, $\boldsymbol{P}$ is defined as principal component loadings, and $\boldsymbol{E}$ consists of minor components which involve noise or unimportant variance.

Usually, PCA compression is very efficient if the variables have linear relationships. If the nonlinear relations among variables exist, PCA as a linear method becomes inefficient. For example, when PCA is used in a nonlinear system, the minor components do not always contain noise or unimportant components, but contain important information. If they are discarded, important information will be omitted. But if they are kept, PCA may contain too many components. To handle this problem, it is necessary to turn to nonlinear PCA methods.

2.2 ANN & nonlinear PCA

The difference between linear PCA and nonlinear PCA is the mapping function. Linear PCA projects original data set X to new linear principal components space T by using a linear mapping function (illustrated in section 2.1). Nonlinear PCA gains nonlinear principal components by using a nonlinear mapping function. There are different ways to represent this nonlinear mapping function, thus to form different nonlinear PCA methods. Hastie and Stuetzle[9] used a linear smooth curve or surface called principal curve or principal surface. Neural network is commonly used as a nonlinear PCA method, such as ANN and IT-NN.

ANN maps nonlinearly the data sets into few latent variables. The ANN has a structure consisting of three hidden layers, namely, mapping layer, bottleneck layer, and de-mapping layer. The number of nodes in the bottleneck layer is much smaller than in the mapping and de-mapping layers. Actually, the ANN can be seen as a hybrid of the two single hidden layer networks, as presented in Fig.1. The network in the left side is functioned as data compressing while that in the right side is functioned as data de-compressing.

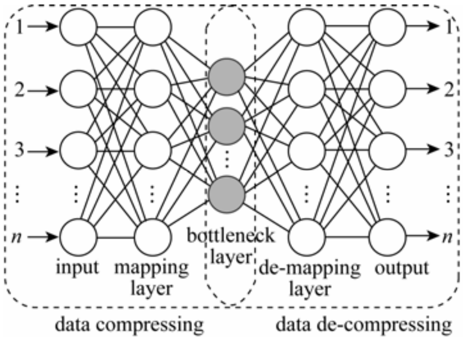

Figure 1 The structure of ANN

During training of ANN, the data set is compressed to few latent variables, the number of which corresponds to the number of nodes in the bottleneck layer. Then the output of the bottleneck layer is decompressed in the de-mapping layer. If the reconstruction error is small, this ensures that the output from the bottleneck layer contains a compact representation of the data set. The abilities of ANN to deal with linear and nonlinear correlation between variables are related to the type of the transfer function used in each layer. In order to guarantee proper functioning of the network, *i.e.*, to uncover nonlinearities between data variables, the neurons in the mapping and de-mapping layers should be nonlinear. Usually, as a transfer function in the mapping and de-mapping layers, like the sigmoid is used. The form of function is given by Eq.(3).

$$\sigma(x) = \frac{1}{1+\mathrm{e}^{-x}} \tag{3}$$

ANN can deal with linear and nonlinear correlations between variables and leads to a compact data representation. It is referred in the literature as nonlinear PCA. Thus, the outputs of nodes in the bottleneck layer can be viewed as nonlinear principal components.

2.3 IT-NN & its learning algorithm

Contrasting to ANN, the network structure of IT-NN is similar to the de-mapping of ANN, simplified with only one hidden layer, as presented in Fig.2. IT-NN has two basic characteristics: fewer input layer nodes than any other layer and fixed inputs which are not given but have to be gained by adjusting.

Instead of training a whole 3-hidden-layer ANN, we only train its de-mapping subnet. Training such a subnet is a little different with training other feed forward network, because the input of the subnet is not given. Therefore, we adjust not only the weights of the network but also the input values reproduce the given

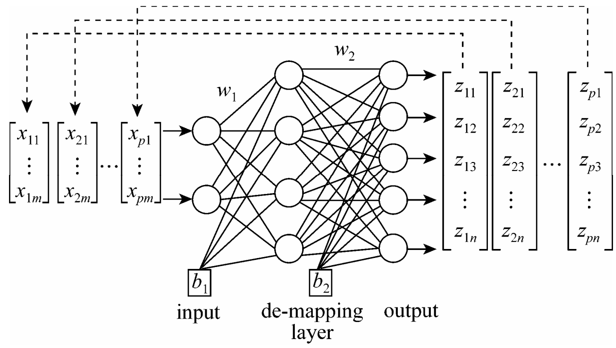
w_1
w_2
x_{11} … x_{1m}
x_{21} … x_{2m}
x_{p1} … x_{pm}
z_{11} z_{12} z_{13} … z_{1n}
z_{21} z_{22} z_{23} … z_{2n}
z_{p1} z_{p2} z_{p3} … z_{pn}
b_1
b_2
input
de-mapping layer
output

Figure 2 The structure of IT-NN

data as accurately as possible. For each input vector (x_{p1}, x_{p2},⋯, x_{pm}), it is adjusted to minimize the error between its corresponding output of IT-NN (z_{p1}, z_{p2},⋯, z_{pn}) and the original sample (t_{p1}, t_{p2},⋯, t_{pn}). After the subnet and its inputs are properly adjusted, we can gain a $p\times m$ matrix $\boldsymbol{X}$ and a de-mapping neural network model. The matrix $\boldsymbol{X}$ can be viewed as nonlinear principal components. Thus all the requirements for data dimensionality reduction is fulfilled through training a de-mapping network and its input layer simultaneously. Therefore, the IT-NN can be considered as an alternative nonlinear PCA method of ANN. The following paragraphs will illustrate the adjusting algorithm in detail.

Let t_{pk} be the value of the kth observed variable in the pth sample and z_{pk} the corresponding output of IT-NN. The aim of training IT-NN is to minimize the following objective function:

$$\min \boldsymbol{E}(\boldsymbol{X},\boldsymbol{W}) = \min \sum_{p}\sum_{k}(z_{pk}-t_{pk})^2 \quad (4)$$

The steepest descent direction for optimizing network inputs Δx_{pi} is defined as follows:

$$\Delta x_{pi} = -\frac{\partial \boldsymbol{E}}{\partial x_{pi}} = \sum_{k}(t_{pk}-z_{pk})\frac{\partial z_{pk}}{\partial x_{pi}} \quad (5)$$

The steepest descent direction for optimizing network weights Δw_{ji} is defined as follows:

$$\Delta w_{ji} = -\frac{\partial \boldsymbol{E}}{\partial w} = \sum_{k}(t_{pk}-z_{pk})\frac{\partial z_{pk}}{\partial w_{ji}} \quad (6)$$

The network output is given by:

$$z_{pk} = \sigma\left[f_k + \sum_{j} w_{kj}^{(2)}\sigma\left(b_j + \sum_{i} w_{ji}^{(1)}x_{pi}\right)\right] \quad (7)$$

where $\sigma(\cdot)$ is sigmoid function, b_{j} and f_k are the bias of the jth hidden node and kth output node, $w_{ij}^{(1)}$, $w_{kj}^{(2)}$ are the weights of IT-NN. The steepest descent direction for training network is

$$\Delta x_{pi} = \sum_{j} w_{ji}^{(1)}\delta_{pj} \quad (8)$$

where δ_{pj} is the propagated error at the hidden layer and has been given by

$$\delta_{pj} = \sigma'\left(b_j + \sum_{i} w_{ji}^{(1)}x_{pi}\right)$$
$$\sum_{k} w_{kj}^{(2)}(t_{pk}-z_{pk})\sigma'\left[f_k + \sum_{j} w_{kj}^{(2)}\sigma\left(\sum_{i} w_{ji}^{(1)}x_{pi}\right)\right] \quad (9)$$

Weight adjusting is given by

$$\Delta w_{ji} = \sum_{p} x_{pi}\delta_{pj} \quad (10)$$

2.4 Improvement of IT-NN learning algorithm

A disadvantage of this approach is that the complexity of training the IT-NN increases exponentially with the dimensionality of the training data set increasing[6]. To improve the algorithm and accelerate the convergence of training IT-NN, here we introduce a momentum factor η and adaptive learning rate, r_x and r_w. Based on original learning algorithm, the input is adjusted as follows,

$$x_{pi}(m+1) = x_{pi}(m) +$$
$$r_x(m)\left[(1-\eta)\frac{\partial \boldsymbol{E}}{\partial x_{pi}(m)} + \eta\frac{\partial \boldsymbol{E}}{\partial x_{pi}(m-1)}\right] \quad (11)$$

$$r_x(m) = 2^{\lambda_x} r_x(m-1) \quad (12)$$

$$\lambda_x(m) = \mathrm{sign}\left[\frac{\partial \boldsymbol{E}}{\partial x_{pi}(m)} \times \frac{\partial \boldsymbol{E}}{\partial x_{pi}(m-1)}\right] \quad (13)$$

Weight adjusting:

$$w(m+1) = w(m) + r_w(m)\left[(1-\eta)\frac{\partial \boldsymbol{E}}{\partial w(m)} + \eta\frac{\partial \boldsymbol{E}}{\partial w(m-1)}\right] \quad (14)$$

$$r_w(m) = 2^{\lambda_w} r_w(m-1) \quad (15)$$

$$\lambda_w = \mathrm{sign}\left[\frac{\partial \boldsymbol{E}}{\partial w(m)} \times \frac{\partial \boldsymbol{E}}{\partial w(m-1)}\right] \quad (16)$$

In this algorithm, momentum factor η helps to meliorate the convergence of training the network. From the equations of the learning algorithm, we can easily find that if the steepest descent directions are same when continuously iterating twice, the learning rate will be doubled. On the contrary, the learning rate will be halved. Therefore, the adaptive learning rate can accelerate the convergence of training IT-NN.

As a whole, the iteration can follow the following steps after the sample $\boldsymbol{T}$ (t_{pk} , the kth observed variable in the pth sample) has been properly prepared:

Step 1 To scale the variables of sample $\boldsymbol{T}$ into the range [0,1]. In our application we use the flowing equation,

$$t'_{pk} = \frac{t_{pk} - \min_k}{\max_k - \min_k} \quad (17)$$

where t'_{pk} is the scaled value of t_{pk}, $\max_k$ and $\min_k$ are the maximum and minimum values of kth variable in sample ***T***.

Step 2 To initialize input matrix ***X*** and weights ($\boldsymbol{W}_1$ and $\boldsymbol{W}_2$) with random values in the range [0,1].

Step 3 For each input $x_{pi}(i=1,\cdots,m)$, calculate the corresponding output $z_{pk}(k=1,\cdots,n)$ of the IT-NN by using Eq.(7).

Step 4 To calculate the errors between the output $z_{pk}(k=1,\cdots,n)$ and the scaled $t'_{pk}\,(k=1,\cdots,n)$.

Step 5 To adjust the weights ($\boldsymbol{W}_1$ and $\boldsymbol{W}_2$), weights learning rate $r_w(p)$, input vectors ***X*** and input learning rate $r_x(p)$ by using Eqs.(11)—(16).

Step 6 For next input $x_{(p+1)i}$ $(i=1,\cdots,m)$, repeat Step 3 to Step 5 until all inputs has been adjusted and calculate the total error between the outputs ***Z*** and the scaled sample ***T′***. If the error is smaller than aim error, and at the same time the error doesn't change any more, exit the iteration. Otherwise, next step.

Step 7 To repeat the iteration from Step 3 to Step 6. After the iteration is finished, we acquire the input matrix ***X*** and a neural network model with fixed weights. For the testing sample, to get the new input matrix, we don't need train the weights any more but only train inputs. Iteration can be repeated from Step 1 to Step 7 but with fixed values of weights. The new input can be viewed as the nonlinear principal components of the testing sample.

3 EXAMPLES

3.1 Comparison between linear PCA, ANN and IT-NN

Example 1: Considering the following nonlinear system,

$$x_1 = 0.5t^2 - 2t + 0.5 + e_1 \tag{18}$$

$$x_2 = t^2 + t + \sin\pi t + e_2 \tag{19}$$

$$x_3 = 2t^2 - t - 2\cos\pi c + e_3 \tag{20}$$

where t is random among [－1,1], e_1, e_2, e_3 are Gaussian noise $\mu[0,\ 0.01]$. Actually, the system is affected only by one variable t, which means the 3-D data sets are relative and have only one independent variable. Therefore the 3-D nonlinear data sets can be compressed to 1-D data set. To illustrate the effectiveness of IT-NN, we compare the capabilities of reducing the dimensionality of the given nonlinear system with linear PCA, ANN (with net structure 3-7-1-7-3), and IT-NN (with the structure 1-7-3). The training error, testing error, and reconstruction error is computed by the Eq.(21), defined as root-mean-square error:

$$e_{\text{RMS}} = \sqrt{\frac{\sum_{p=1}^{m}\sum_{k=1}^{n}(z_{pk} - t_{pk})^2}{mn}} \tag{21}$$

where, m and n is the dimension of sample ***T*** (z_{pk}) and (t_{pk}), respectively.

The results are listed in Table 1. Data reconstructed by linear PCA, ANN, and IT-NN are presented in Fig.3. From Table 1 and Fig.3 we can see that the data sets are represented through one linear principal component have much error with original data sets. One linear principal component can only explain 63.5% information of the original data sets. While new data sets represented by one nonlinear principal component have the same distribution and nearly no difference with the original data sets. One nonlinear principal component can explain nearly all the information of original datasets (99.1% with ANN and 99.6% with IT-NN). This comparison illustrates that when reducing the dimensionality of a nonlinear system to its real value, linear PCA may cause information lost. To solve that problem, we need more principal components to explain the representation original data sets, which will increase the dimensionality of the new principal components space. But for nonlinear PCA, that is not a problem. Dimensionality reduction of nonlinear systems can be done effectively with much less principal components.

Table 1 Results for Example 1

Model	Reconstruction error	Retained variation
linear PCA	8.44	63.5%
ANN	0.017	99.1%
IT-NN	0.010	99.6%

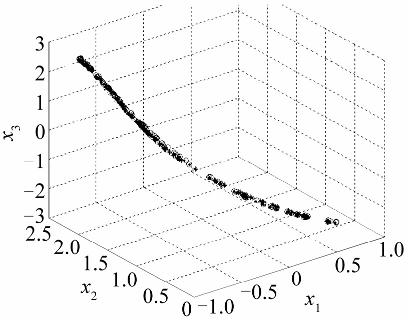

Figure 3 Data reconstructed by LPCA, ANN and IT-NN
● original data; * by ANN; ◇ by LPCA; ○ by IT-NN

3.2 Effectiveness of improved IT-NN

In section 3.1, we mentioned that the improved training algorithm is much faster than the original training algorithm when training IT-NN. This section will illustrate such conclusions with an example. Reconsider the nonlinear system of Example 1 in section 3.1, we use ANN (with 3-7-1-7-3 net structure) and IT-NN (with 1-7-3 net structure) trained with both learning algorithms to obtain one nonlinear principal component. The training results are listed in Table 2 and illustrated in Fig.4. All the errors are computed by Eq.(21). For the new training algorithm of IT-NN, the error decreases sharply when the number of iteration increases to 100. The training error drops down with increasing of the iteration and reaches the level 0.01 at 400. While the original algorithm, training the same IT-NN, needs more iteration to drop down the training error. About 200 iterations are needed to decrease the error sharply, which is nearly twice the number of the iteration for the improved algorithm. When iteration increases to 650, training error reaches the goal. Comparison was also made between ANN and IT-NN. For ANN, it needs more than 1200 iteration steps to decrease the training error to the same goal. The reason why ANN needs more iterations than IT-NN is that ANN has a complicated net structure, and the performance of back propagation deteriorates while the number of hidden layer gets larger [10].

Table 2 Comparison between ANN and IT-NN

Model	Training error	Testing error	Iterations
ANN	0.017	0.023	1250
IT-NN	0.010	0.016	650
improved IT-NN	0.009	0.013	400

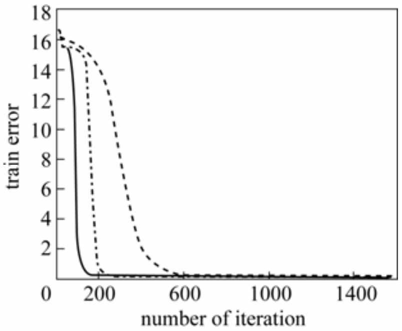

Figure 4 Train ANN, original IT-NN and improved IT-NN
—— improved IT-NN; –·–· original IT-NN; – – – ANN

From the experiment, we can see that the adaptive learning rate and momentum factor improves the performance of IT-NN, meliorates the divergence of training process. Actually, as an extension, the adaptive learning rate and momentum factor can be applied to the learning algorithms of any feed-forward neural network, which can save a lot of time during the training process.

4 NONLINEAR SYSTEM MODELLING WITH IT-NN & RBF-NN

The purpose of combining IT-NN with RBF is to take advantage of these two neural networks. IT-NN is a good way to reduce the dimensionality of sample data sets and remain the nonlinear characters between variables. RBF neural network is good at mapping the input and output. But high dimensionality of input layer will lead to more input nodes in RBF, which will make the net structure more complicated and harder to be trained. So modified IT-NN is taken as the input vector of the RBF-NN and reduces the dimension of the input sample in RBF-NN. Generalized RBF structure is presented in Fig.5. The Gaussian function is taken as the radial basis function. For training, learning algorithm based on gradient descent is adopted in this paper. The whole model structure is presented in Fig.5.

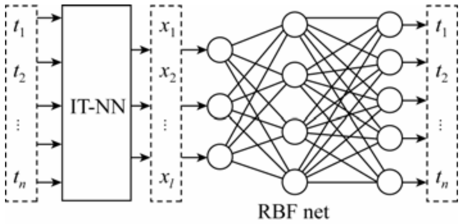

Figure 5 Combine IT-NN with RBF-NN for nonlinear system modeling

Let Y=F($\boldsymbol{T}$) be a nonlinear function, where inputs $\boldsymbol{T}$ have high dimension. To model this nonlinear function using RBF neural network, firstly, we use IT-NN to reduce the dimensionality of original data sets t_i(i=1,⋯,n) from m dimension to l dimension. For each input $\boldsymbol{T}$(t_1, t_2,⋯, t_n), we can calculate the nonlinear principal components $\boldsymbol{X}$(x_1, x_2,⋯, x_l), by training the inputs of IT-NN. Secondly, we use RBF to map the relations between output Y and nonlinear principal components $\boldsymbol{X}$. The simulation in section 5 will illustrate how to apply this model to nonlinear system.

5 APPLICATION IN ETHYLENE CRACKING FURNACE SYSTEM MODELLING

The key of an ethylene plant with huge economic impact is a cracking furnace. And the key factor of optimal design and operation in furnace is the prediction. There are 20 factors (4 group naphtha feed flows, 4 group high-pressure dilute steam flows, 4 group pipe inlet temperatures in radiant box, 4 groups coil outlet temperature (COT), 1 fuel flow in the bottom of cracking furnace, 1 fuel flow in the wall of cracking furnace, 1 flue gas cross temperature, and 1 flue temperature in the hearth) that directly or indirectly affect the yields of ethylene and propylene[11,12]. The 20 variables are marked in Fig.6.

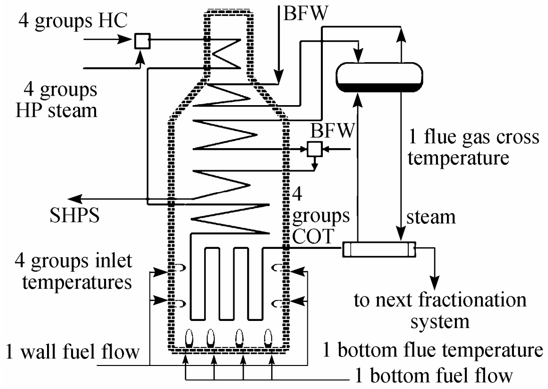

Figure 6 Cracking furnace

Network training will be carried out once again while the change of cracking raw material and operation condition is taken place. Retraining network will ensure to the soft sensing model reliability and accuracy. The sample data must be preprocessed, such as data rectification and data transformation before modelling in order to ensure model accuracy, and the description of data preprocessing is no longer mentioned.

Limited by the sample data, the outputs are only the yields of C_2H_4 and C_3H_6. So the predictive model of the yields of ethylene and propylene has 20 inputs and 2 outputs. If we only use one RBF-NN to map the inputs and outputs, the net structure would be quite complicated and training such complicated network would consume time and even diverge if the net structure were not selected properly.

We use two neural networks, IT-NN and RBF-NN, to make the model. Through training the inputs and weights of IT-NN we reduce the dimensionality of the sample data sets consisted of 20 variables. In order to determine the input nodes of IT-NN, we implement the linear PCA method, and get 10 linear principal components if selecting the control level 90%. Since a nonlinear method will describe the data with greater accuracy and fewer latent variables than a linear method, the number of nonlinear principal components gained through IT-NN would be less than 10. Therefore, we select an IT-NN with 10 to 5 input nodes and 15 hidden layer nodes to gain different nonlinear PCs. The errors of data reconstructed by these nonlinear principal components is listed in Table 3. Firstly, we train IT-NN and RBF with different PCs, and then use these models to predict the yields of ethylene and list the errors in Table 4. From Tables 3 and 4, we can see that the IT-NN with 7 nonlinear PCs will be the best choice. Fewer PCs (6 and 5) cause information lost, and more PCs (10, 9, 8, 7) increase the input nodes of RBF. Fig.7 shows the predictive curves produced by the model (IT-NN with 7 nonlinear PCs + RBF). Further, 5 different kinds of naphtha with two operation conditions are selected to verify the model precision, and the testing results of comparing the practical yields with the model calculation are showed in Table 5 and Table 6. In the tables, nP is normal paraffin, iP is isomeric paraffin, N is naphthene, and A is aromatics. These values are percentages of mass. IP, 10%,···, 90%, EP are fractionating points. Thereinto, IP is initial fractionating point, EP is end fractionating point, and ASTM is American Society of Testing Materials. The results have shown that this model is accurate in predicting the yields of ethylene and propylene and can be applied in cracking furnace system as a soft predict model.

Table 3 Reconstruction errors with different input layer nodes of IT-NN

Net structure	Construction error
10-15-20	0.0092
9-15-20	0.0106
8-15-20	0.0126
7-15-20	0.0130
6-15-20	0.05329
5-15-20	0.13785

Table 4 Training and predicting errors with different PCs in IT-NN+RBF-NN model

IT-NN net structure	Training error		Predicting error	
	C_2H_4	C_3H_6	C_2H_4	C_3H_6
10-15-20	0.0106	0.0135	0.0593	0.06053
9-15-20	0.0120	0.0132	0.0631	0.06539
8-15-20	0.0124	0.0141	0.0677	0.06900
7-15-20	0.0130	0.0150	0.0731	0.0901
6-15-20	0.2001	0.1689	0.1896	0.3090
5-15-20	0.3901	0.2155	0.3408	0.3754

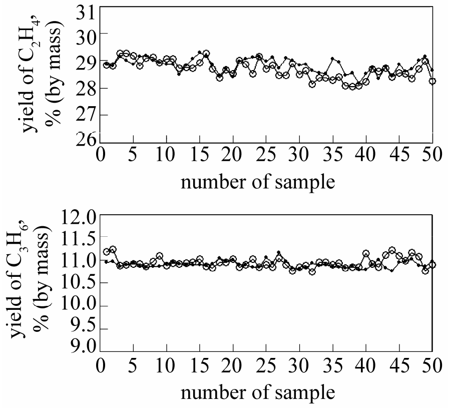

Figure 7 The yields of ethylene and propylene
○ data measured by chromatogram instrument;
● data predicted by IT-NN+RBF with 7 PCs

6 CONCLUSIONS

This paper illustrates the limitation of linear PCA and the effectiveness of nonlinear PCA when analyzing nonlinear system. And it also studies the theory of ANN and IT-NN as nonlinear PCA methods. A momentum factor and adaptive learning rate are presented to improve the performance of IT-NN and meliorate the convergence of training process. The improved IT-NN is an effective way to reduce the dimensionality of data sets where relations among variables are nonlinear. Comparison is also made between linear PCA, ANN, and IT-NN by examples. Experiments have shown that the improved IT-NN has a better performance than ANN and original IT-NN. In order to predict the yields of ethylene and propylene, which are the key factors of optimal design and operation in furnace, this paper studies a soft sensing model based on IT-NN and RBF-NN. Compared with traditional chromatogram instrument, the predictive model works effectively and has many advantages: timely prediction, less investment, easy maintenance, and convenient operation. Based on this model, advanced process control, operation optimization, performance monitoring, and production evaluation can be implemented effectively.

Table 5 Predicted results for 5 different naphtha (COT:840℃, dilute ratio: 0.5)

Density	nP	iP	N	A	ASTM fractionating temperature, ℃							Practical	Calculation	Relative	Practical	Calculation	Relative
					IP	10%	30%	50%	70%	90%	EP	C_2H_4	C_2H_4	error	C_3H_6	C_3H_6	error
0.663	33.92	46.16	16.95	3.29	29.6	41	51	64	79	120	164	29.18	29.4599	0.9592	11.03	11.0038	0.2375
0.680	33.30	39.20	20.98	5.89	30	43	54	68.8	83	118	163	28.92	28.6563	0.9118	10.77	10.8117	0.3871
0.670	35.22	41.33	18.05	4.55	31	43.9	52.2	62.4	78.9	111.2	162	28.81	29.1193	1.0730	10.93	10.9275	0.0229
0.682	41.26	33.55	22.93	2.26	26.8	47.5	65.0	81.0	100.0	134.8	168.0	31.01	30.9690	0.1322	13.96	13.8901	0.5007
0.697	31.13	34.48	26.39	8.00	37.1	55.9	73.2	91.6	111.8	140.5	158.0	28.06	28.1350	0.2672	12.62	12.5603	0.4730

Table 6 Predicted results for 5 different naphtha (COT:840℃, dilute ratio: 0.6)

Density	nP	iP	N	A	ASTM fractionating temperature, ℃							Practical	Calculation	Relative	Practical	Calculation	Relative
					IP	10%	30%	50%	70%	90%	EP	C_2H_4	C_2H_4	error	C_3H_6	C_3H_6	error
0.663	33.92	46.16	16.95	3.29	29.6	41	51	64	79	120	164	28.32	28.2755	0.1571	11.57	11.476	0.8124
0.680	33.30	39.20	20.98	5.89	30	43	54	68.8	83	118	163	28.89	28.7402	0.5185	11.91	11.850	0.5037
0.670	35.22	41.33	18.05	4.55	31	43.9	52.2	62.4	78.9	111.2	162	29.41	29.2768	0.4529	11.62	11.562	0.4991
0.682	41.26	33.55	22.93	2.26	26.8	47.5	65.0	81.0	100.0	134.8	168.0	31.06	30.9821	0.2508	13.98	14.023	0.3075
0.697	31.13	34.48	26.39	8.00	37.1	55.9	73.2	91.6	111.8	140.5	158.0	28.17	28.1612	0.0312	12.72	12.827	0.8411

ACKNOWLEDGEMENTS

The authors are grateful to Dr. Geng Zhiqiang for his assistance with checking computations.

NOMENCLATURE

b_j	bias of the *j*th hidden node
$\boldsymbol{E}$	residual matrix
e_i	*i*th Gaussian noise
e_{RMS}	root-mean-square error
f_k	*k*th output node
m	dimension of sample $\boldsymbol{T}$ (t_{pk})
max *k*	the maximum values of *k*th variable in the sample $\boldsymbol{T}$
min *k*	the minimum values of *k*th variable in the sample $\boldsymbol{T}$
n	dimension of sample $\boldsymbol{T}$ (t_{pk})
P	principal component loadings
$\boldsymbol{p}_i$	an eigenvector of matrix $\boldsymbol{X}$
r_{w}, r_x	adaptive learning rate
$\boldsymbol{T}$	principal component scores
t_i	the *i*th principal component
t_{pk}	the *k*th observed variable in the *p*th sample
t'_{pk}	the scaled value of t_{pk}
Δw_{ji}	weight increment between two nodes
$\boldsymbol{X}$	matrix
Δx_{pi}	increment of the *i*th input variable for the *p*th sample
Y	nonlinear function
z_{pk}	IT-NN output of the *k*th variable for the *p*th sample
δ_{pj}	propagated error at the hidden layer
η	momentum factor
μ[0, 0.01]	Gaussian noise
$\sigma(\cdot)$	sigmoidal function

REFERENCES

1 Jackson, J.E., "Principal components and factor analysis (Ⅰ) Principal components", *J.Quality Technol.*, **12**, 201—203(1980).

2 Dong, D., McAvoy, T.J., "Nonlinear principal component analysis based on principal curve and neural networks", *Comput. Chem. Eng.*, **20**(1), 65—78(1996).

3 Kramer,M.A. "Nonlinear principal component analysis using autoassociative neural networks", *AIChE J.*, **37**(2), 233—243(1991).

4 Tan, S., Mavrovouniotis, M.L., "Reducing data dimensionality through optimizing neural network inputs", *AIChE J.*, **41**(6), 1471—1480(1995).

5 Jia, F., Martin, E.B., Morris, A.J., "Non-linear principal components analysis for process fault detection", *Comput. Chem. Eng.*, **22**, (S1), S851—S854(1998).

6 Shao, R., Jia, F., Martin, E.B., Morris, A.J., "Wavelets and non-linear principal components analysis for process monitoring", *Control Eng. Prac.*, **7**, 865—879(1999).

7 Geng, Z.Q., Zhu, Q.X., "Multiscale nonlinear principle component analysis(NLPCA) and its application for chemical process monitoring", *Ind. Eng. Chem. Res.*, **44**, 3585—3593(2005).

8 Hu, T., Du, H.B., Yao, P.J., "Prediction of process trends based on neural networks", *Chin. J. Chem. Eng.*, **10**(3), 286—289(2002).

9 Hastie, T.J., Stuetzle, W., "Principal curves", *J. Amer. Stat. Assoc.*, **84**, 502—516(1989).

10 Hertz, J., Krogh, A., Palmer, R.G., Introduction to the Theory of Neural Computation, Wesley Publisher Comp., Redwood City, CA, 198(1991).

11 Kumar, P., Kunzru, D., "Modeling of naphtha pyrolysis", *Ind. Eng. Chem. Process Des. Dev*, **24**, 774—782(1985).

12 Yang, E.F., Zhou, Q., Hu, Y.F., Xu, Y.M., "A soft-sensing approach to on-line predict the yields of industrial pyrolysis furnace based on PCA-RBF neural networks", *J. Syst. Simulation*, **13**(Suppl), 194—197(2001). (in Chinese)