Particle Swarm Optimization Neural Network and Its Application in Soft-Sensing Modeling

Guochu Chen and Jinshou Yu

Research Institute of Automation, East China University of Science and Technology,
Shanghai 200237, China
chgcsh@sohu.com

Abstract. Particle swarm optimization algorithm (PSO) is applied to train artificial neural network (NN) to construct a neural network based on particle swarm optimization algorithm (PSONN). Then, PSONN is employed to construct a practical soft-sensor of gasoline endpoint of main fractionator of fluid catalytic cracking unit (FCCU). The obtained results indicate that soft-sensing model based on PSONN has better performance than soft-sensing model based on BPNN and the new method proposed by this paper is feasible and effective in soft-sensing modeling of gasoline endpoint.

1 Introduction

Soft sensing techniques have been used more frequently as attractive and effective methods of process modeling and a replacement of expensive and ineffective online analytical instrument to some extent [1]. Now, there are two types of models usually used in the soft sensing modeling of the chemical industrial process [1]: mechanistic models (or first principle model, FPM) developed from the underlying physical and chemical knowledge about a process, and empirical models (EM) developed from the operational data of a process. FPM is based on the analysis of the mass, momentum, and energy balance as well as empirical correlation. However, only major characteristics and trends of the process are described by the FPM. Additionally, FPM includes many assumptions, and lacks in considering random disturbances that are present in many real systems. However, the development of FPM for some processes, especially some complex processes, can be too difficult or even not possible. For such processes, empirical models (EM) based on process operational data should be preferred. Many industrial processes exhibit nonlinear dynamic behavior, and nonlinear model should be developed. Artificial neural network (NN) has been shown to be able to approximate any continuous nonlinear functions [2] and is an attractive technique that can be applied to nonlinear process modeling.

Artificial neural network is a representation that attempts to mimic the functionality of the brain. For several decades scientists have being trying to emulate the real neural structure of the brain, believing that the human process of learning might be reproduced by an algorithmic equivalent. Initially the principal motivation behind this research was the desire to achieve the sophisticated level of information processing that could be achieved by the brain. However, it is apparent that present research aims

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3611, pp. 610 – 617, 2005.

are not directed at emulating the sheer complexity of the brain. Generally, the methodology is used on a more modest scale to develop nonlinear models.

An important issue in NN is the train algorithm. Now back-propagation algorithm (BP) is most commonly used to train NN [2]. BP is a gradient-based method, so some inherent problems are frequently encountered in the use of this algorithm, e.g., very slow convergence speed in training, easily to get stuck in a local minimum, etc. Some techniques are therefore introduced in an attempt to resolve these drawbacks, but all of them are still far from satisfaction [2], so new train algorithm needs developing.

Particle swarm optimization algorithm (PSO) is an evolutionary computation algorithm proposed by Eberhart and Kennedy in 1995 [3-4]. The idea of PSO is based on the simulation of simplified social models, such as bird flocking, fish schooling, and the swarming theory. PSO is a simple algorithm and can be developed over a very simple theoretical framework and can be implemented with a few lines of computer code, requiring only primitive mathematical operators. PSO is computationally inexpensive in terms of both memory requirements and speed. Besides, PSO is indeed a population-based stochastic algorithm. It does not need gradient information, as the gradient-based algorithm does. This allows functions whose gradients are either unavailable or computationally expensive to be solved using the PSO algorithm. It was originally developed for optimization in a continuous space and it has been recently adapted to optimization in binary spaces, presenting good performance also when applied to discontinuous objective functions and is an attractive algorithm in artificial neural network training. Here, PSO is employed to train NN to construct an artificial neural network based on particle swarm optimization algorithm (PSONN). Then PSONN is applied to construct a practical soft-sensor of gasoline endpoint of main fractionator of fluid catalytic cracking unit (FCCU).

2 Particle Swarm Optimization Neural Network (PSONN)

2.1 PSO Algorithm

PSO algorithm uses a population of individual called "particles". Each particle has its own position and velocity to move around the search space. Using the term "particle" may convey finite mass-volume objects, which is not true of the PSO algorithm. These particles are, in fact, points in space. However, since these points have velocity and position, the term "particle" is more suitable than "point". Particles move to trying to find the solution for the problem being solved.

Suppose that the search space is D-dimensional and a particle swarm consists of m particles, then the i-th particle of the swarm can be represented by a D-dimensional vector, $X_i = (x_{i1}, x_{i2}, x_{i3}, \cdots, x_{iD})$, $i = 1,2,\cdots,m$. The velocity of this particle can be represented by another D-dimensional vector, $V_i = (v_{i1}, v_{i2}, v_{i3}, \cdots, v_{iD})$. The fitness of every particle can be evaluated according to the objective function of optimization problem. The best previously visited position of the i-th particle is denoted as its individual best position, $P_i = (p_{i1}, p_{i2}, p_{i3}, \cdots, p_{iD})$. Define g as the index of the best particle of the whole swarm, the position of the best individual of the whole swarm is denoted as the global best position P_g, and the fitness of the global best position is denoted as the global best fitness F_g. Then the velocity of particle and its new position will be assigned according to the following two equations [3-5]:

$$v_{id} = \chi \cdot (\omega v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id})) \tag{1}$$

$$x_{id} = x_{id} + v_{id} \tag{2}$$

where χ is a constriction factor; ω is called inertia weight; c_1 and c_2 are two positive constants called acceleration coefficients; r_1 and r_2 are two random numbers uniformly from the interval [0, 1].

2.2 The Structure of PSONN

An artificial neural network consists of a system of simple interconnected neurons, or nodes, as illustrated in Fig. 1. It is a model representing a non-linear mapping between input and output vectors. The nodes are connected by weights and output signals, which are a function of the sum of the inputs to the node modified by a simple non-linear transfer function, or activation function. It is the superposition of many simple non-linear transfer functions that enables the neural network to approximate extremely non-linear functions. The output of a node is scaled by the connecting weight and feed forward to be an input to the nodes in the next layer of network. The architecture of a neural network is variable, but, in general, consists of several layers of neurons. The input layer plays no computational role but merely serves to pass the input vector to the network. A neural network may have one or more hidden layers and have only an output layer. The neural network is described as being fully connected to every node in the next and the previous layer.

In Fig. 1, x is the input of NN. *net* is the sum of the inputs to the node modified by activation function. O is the output of neural node. y is the output of NN. PSONN assumes the weight and threshold of NN as the position of particle of PSO, evaluates the fitness of particle according to objective function of system, then searches for the global best weight and the global best threshold by PSO. If the search is accomplished, the position of the global best particle is the combination of the best weight and the best threshold of PSONN.

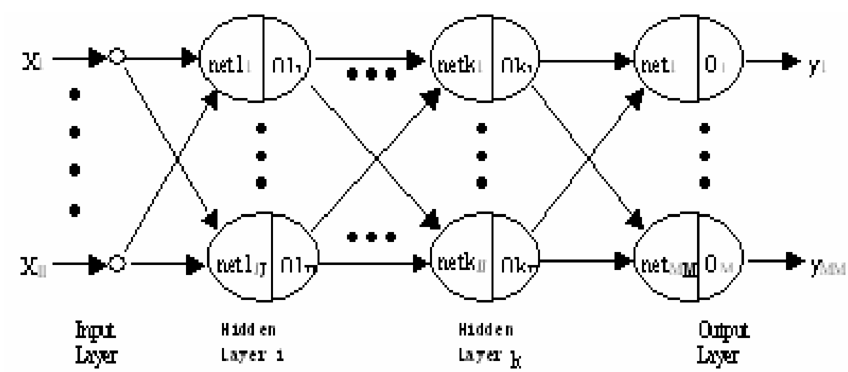

Fig. 1. The structure of PSONN

By selecting a suitable set of weights and transfer functions, it is known that a neural network can approximate any smooth, measurable function between the input and output vectors. The neural network has the ability to learn through training. The training requires a set of training data, i.e., a series of input and associated output vectors. During the training, the neural network is repeatedly presented with the training data and the weights in the network are adjusted from time to time till the desired input–output mapping occurs. If, after the training, the neural network is presented with an input vector, not belonging to the training pairs, it will simulate the system and produce the corresponding output vector. The error between the actual and the predicted function values is an indication of how successful the training is.

2.3 Train Algorithm of PSONN

PSONN train algorithm can be summarized in the following steps:

1. Initialize the structure, activation function and objective function of PSONN.
2. Initialize the algorithm parameters of PSO.
3. Store initial position of each particle. Evaluate and store initial fitness of each particle. Evaluate and store the global best position and global best fitness of the swarm.
4. Update particles' velocities and positions by equation (1) & equation (2), and set a limit to particles' positions and particles' velocities.
5. Update the individual best fitness and the individual best position of each particle; Update the global best fitness and the global best position of the swarm.
6. If the stopping condition is not satisfied, go to step 4. Otherwise, stop iterating and obtain the best weight and the best threshold from the global best position.

3 Practical Application in Soft-Sensing Modeling

3.1 Introduction of Engineering Background

Main fractionator is one of the most important equipment of FCCU, which is a most important unit in refineries [6]. The feed-in material of the tower is get from catalytic cracking reactor. The effluent product of reaction goes to the main fractionator, where the heat is removed in its various pump-around and loops and initial product separation is accomplished, then usable products including gas, gasoline, light diesel oil and heavy diesel oil are produced.

Gasoline endpoint is a most important product quality indicator of main fractionator of FCCU [6]. But gasoline endpoint can't be measured on-line directly. At present, gasoline endpoint is usually acquired mainly by artificial analyzing once every 4 hours. It can cause a long delay in control and the product will be unqualified if the component of the reactor product changes a lot. So it is very important for refineries to acquire gasoline endpoint on-line.

3.2 Soft-Sensing Model of Gasoline Endpoint Based on PSONN

Gasoline endpoint can't be measured directly like temperatures, pressures and flow rates. But it can be estimated by soft-sensor. According to the analysis of system's technological mechanism and the principal component analysis of the practical industrial data, gasoline endpoint is related with these nine variables that can be measured and recorded on-line: the pressure of the top of the tower, the temperature of oil-gas at the top of the tower, the temperature of reflux at the top of the tower, the temperature of the 18-th floor tray, the temperature of the 9-th floor tray, the temperature of the first middle reflux of the tower, the temperature of gasoline of the tower, the flow rate of reflux flow of the first middle reflux and the temperature of the feed oil-gas of main fractionator. In this section, gasoline endpoint is studied. The relationship between gasoline endpoint and the above-mentioned nine variables is complex nonlinear relationship. To estimate gasoline endpoint, we must find the relationship between gasoline endpoint and the nine variables. In this section, a PSONN that has nine input signals that correspond with the above nine variables, a middle layer whose number of node is twenty and an output signal that is gasoline endpoint is employed to find the relationship between gasoline endpoint and the nine variables. The structure of PSONN is 9-20-1. The transfer function of neurons of PSONN takes hyperbolic tangential function. The objective function of the soft-sensing model can be expressed as follow:

$$\min E = \frac{1}{2} \cdot \sum_{kk=1}^{n_p} (t^{kk} - y^{kk})^2 \tag{3}$$

Where t is the real value of gasoline endpoint, y is the estimated value of gasoline endpoint, kk is the serial number of samples, n_p is the total number of samples.

To evaluate the performance of soft-sensing model conveniently, the mean square error and the mean absolute error are defined as follow:

$$RMSE = \frac{1}{n_p} \sum_{1}^{n_p} (t^{kk} - y^{kk})^2 \tag{4}$$

$$Meanae = \frac{1}{n_p} \sum_{1}^{n_p} \left| t^{kk} - y^{kk} \right| \tag{5}$$

In searching for the global best weight and threshold of NN by PSO, population of swarm is set to100; The maximal number of iteration step is set to 20000; Error limit of objective function is 0.2; c_1 and c_2 are set to 2.0; ω is gradually decreased from 1.8 to 0.06; χ is set to 0.8.

In order to compare the result of soft-sensor based on PSONN with the result of soft-sensor based on BPNN that a NN based on BP algorithm, this paper constructs another soft-sensing model of gasoline endpoint based on BPNN. In BPNN, the structure of NN, the transfer function of neurons and the sample data all are the same as that of PSONN. The differences are: the train algorithm is BP algorithm, the learning velocity is 0.016 and the momentum factor is 0.012.

3.3 Discussion of Application Results

There are 127 sets of sample data that consist of nine operating variables in different operating states and one output variable, the real value of gasoline endpoint. 77 pairs of them are used as off-line training data sets and another 50 pairs are used as on-line examining data sets. All the sample data are processed by error-detected, smoothed, filtered and standardized in the intervals [-1, +1] before they are used as input or output of the two soft-sensors.

After the learning and statistical accounting, the errors of 66.2 percent of learning samples are less than ±1℃; The errors of 98.7 percent of learning samples are less than ±2℃; The mean square error of the learning samples is 0.9248℃; The mean absolute error of the learning samples is 0.7671℃ in soft-sensor based on PSONN. However, in soft-sensor based on BPNN, the errors of 64.9 percent of learning samples are less than ±1℃; The errors of 96.1 percent of learning samples are less than ±2℃; The mean square error of the learning samples is 1.0626℃; The mean absolute error of the learning samples is 0.8483℃. Table 1 and Fig. 2 show the comparison between learning result of soft-sensing model based on PSONN and learning result of soft-sensing model based on BPNN. These experiment data, Table 1 and Fig. 2 show that the learning result of soft-sensing model based on PSONN is better than learning result of soft-sensing model based on BPNN. The soft-sensing model based on PSONN has higher learning precision and better learning ability than the soft-sensing model based on BPNN.

Table 1. Comparison of the learning results of the two soft-sensing models

<table>
<tr><td colspan="2"></td><td>Avera-value</td><td>Mini-value</td><td>Maxi-value</td></tr>
<tr><td colspan="2">Real values (℃)</td><td>198.0</td><td>192</td><td>207</td></tr>
<tr><td rowspan="2">Predictive values</td><td>PSONN</td><td>197.9</td><td>191.6</td><td>206.3</td></tr>
<tr><td>BPNN</td><td>198.7</td><td>192.2</td><td>207.0</td></tr>
<tr><td rowspan="2">Meanae</td><td colspan="3">PSONN (℃)</td><td>0.7671</td></tr>
<tr><td colspan="3">BPNN (℃)</td><td>0.8483</td></tr>
<tr><td rowspan="2">the maximum relative error</td><td>PSONN</td><td colspan="3">0.01061</td></tr>
<tr><td>BPNN</td><td colspan="3">0.01783</td></tr>
</table>

<table>
<tr><td>|e|</td><td>|e|<1</td><td>1<|e|<2</td><td>|e|>2</td></tr>
<tr><td>Total</td><td colspan="3">77</td></tr>
<tr><td>Numbers</td><td>51</td><td>25</td><td>1</td></tr>
<tr><td>Numbers</td><td>50</td><td>24</td><td>3</td></tr>
<tr><td rowspan="2">RMSE</td><td>PSONN</td><td colspan="2">0.9248 (℃)</td></tr>
<tr><td>BPNN</td><td colspan="2">1.0626 (℃)</td></tr>
<tr><td rowspan="2">the minimum relative error</td><td>PSONN</td><td colspan="2">0</td></tr>
<tr><td>BPNN</td><td colspan="2">0.00005</td></tr>
</table>

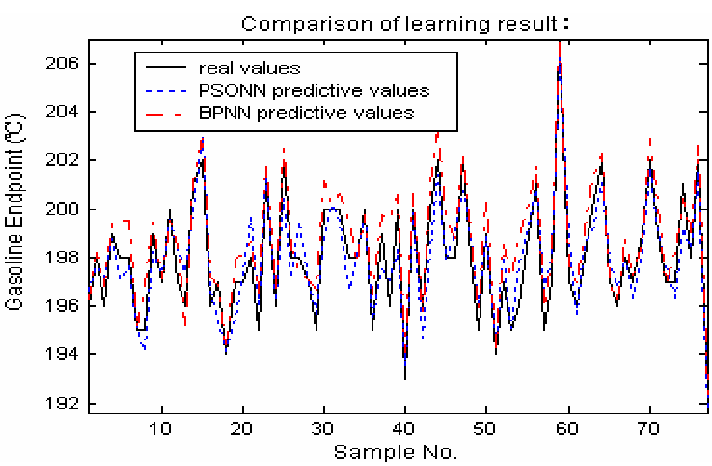

Fig. 2. Comparison of learning result

After the examining and statistical accounting, the errors of 52 percent of examining samples are less than ±1℃; The errors of 88 percent of examining samples are less than ±2℃; The mean square error of the examining samples is 1.3787℃; The mean absolute error of the examining samples is 1.1034℃ in soft-sensor based on PSONN. However, in soft-sensor based on BPNN, the errors of 54 percent of examining samples are less than ±1℃; The errors of 86 percent of examining samples are less than ±2℃; The mean square error of the examining samples is 1.4425℃; The mean absolute error of the examining samples is 1.1284℃. Table 2 and Fig. 3 show the comparison between examining result of soft-sensing model based on PSONN and examining result of soft-sensing model based on BPNN. These experiment data, Fig. 3 and Table 2 show that the examining result of soft-sensing model based on PSONN is better than examining result of soft-sensing model based on BPNN. The soft-sensing model based on PSONN has higher examining precision and better generalization ability than the soft-sensing model based on BPNN.

Table 2. Comparison of the examining results of the two soft-sensing models

<table>
<tr><td colspan="2"></td><td>Aver-values</td><td>Mini-values</td><td>Maxi-values</td><td>|e|</td><td>|e|<1</td><td>1<|e|<2</td><td>|e|>2</td></tr>
<tr><td colspan="2">Real values (℃)</td><td>197.6</td><td>192</td><td>205</td><td>Total</td><td colspan="3">50</td></tr>
<tr><td rowspan="2">Predictive values</td><td>PSONN</td><td>197.9</td><td>192.3</td><td>204.9</td><td>Numbers</td><td>26</td><td>18</td><td>6</td></tr>
<tr><td>BPNN</td><td>198.0</td><td>191.2</td><td>204.8</td><td>Numbers</td><td>27</td><td>15</td><td>8</td></tr>
<tr><td rowspan="2">Meanae</td><td colspan="3">PSONN (℃)</td><td>1.1034</td><td rowspan="2">RMSE</td><td>PSONN</td><td colspan="2">1.3787 (℃)</td></tr>
<tr><td colspan="3">BPNN (℃)</td><td>1.1284</td><td>BPNN</td><td colspan="2">1.4425 (℃)</td></tr>
<tr><td colspan="2" rowspan="2">the maximum relative error</td><td>PSONN</td><td colspan="2">0.01904</td><td rowspan="2">the minimum relative error</td><td>PSONN</td><td colspan="2">0.00015</td></tr>
<tr><td>BPNN</td><td colspan="2">0.01986</td><td>BPNN</td><td colspan="2">0.00018</td></tr>
</table>

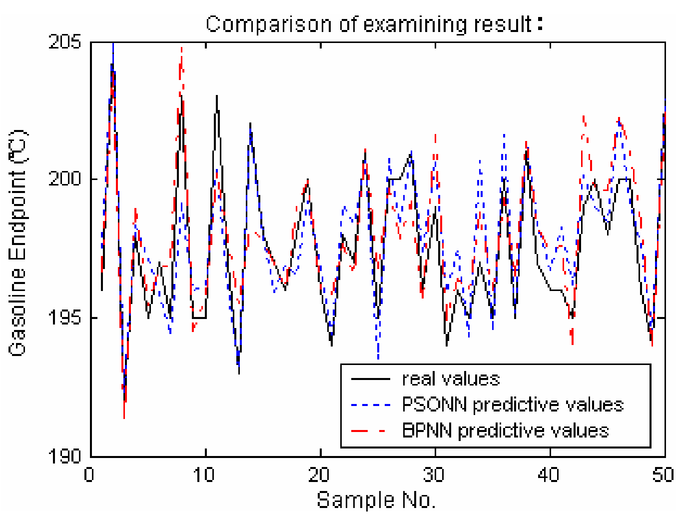

Fig. 3. Comparison of examining result

4 Conclusion

A neural network based on PSO is proposed for soft-sensing modeling of gasoline endpoint of main fractionator of FCCU. The approach takes a novel kind of optimization algorithm, i.e., particle swarm optimization algorithm, to train the neural network. A performance comparison is emphasized on the PSO-based soft-sensing model with the most commonly used BP-based soft-sensing model. The results show that the soft-sensor based on PSONN has better training performance, higher precision and better predicting ability than soft-sensor based on BPNN. It is convenient for refineries to estimate, display, record and analyze gasoline endpoint on-line. It is worth to mention that the current study is very preliminary for the PSO-based neural network approach applied in soft-sensing modeling of gasoline endpoint of main fractionator of FCCU. More researches need to be done, for example, to improve current approach and apply it to other product quality estimate of industrial process or more complex industrial cases.

References

1. Jinshou Yu, Ailun Liu, Kejin Zhang: Soft-senor Technique and Its Applied in Petrochemical Industry Process. Chemical Industry Press, Beijing, China (2000)
2. Licheng Jiao: Theory of Neural Networks System. Xidian University Press, Xi'an, Shanxi, China (1996)
3. Kennedy J, Eberhart R C: Particle Swarm Optimization. Proc. IEEE Int. Conf. on Neural Networks. Perth, WA, Australia (1995) 1942-1948
4. Eberhart R C, Kennedy J: A New Optimizer Using Particle Swarm Theory. Proc. the Sixth Int. Symposium on Micro Machine and Human Science. Nagoya, Japan (1995) 39-43
5. K.E. Parsopoulos, M.N. Vrahatis: Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization. Natural Computing 1 (2002) 235–306
6. Shixiong Lin: Petroleum Processing Engineering. Petroleum Industry Press, Beijing, China (2000)