Proceedings of the 5th WSEAS Int. Conf. on SIMULATION, MODELING AND OPTIMIZATION, Corfu, Greece, August 17-19, 2005 (pp278-283)

Short-Term Load Forecasting Using PSO-Based Phase Space Neural Networks

Jiang Chuanwen*, Fang Xinyan*, Wang Chengmin*
* Department of Electrical Engineering, Shanghai Jiaotong University
Huashan Road 1954, Shanghai, P.R China, 200030
Lu Jianyu**, Wang Liang**
** East China Grid Company Limited. Shanghai, P.R China

Abstract: - The nonlinear theories of load forecasting, such as the applications of neural network and chaos, have recently made considerable progress. Generally, it is an effective method to combine phase space restructures theory with artificial neural networks (ANN) model for load forecasting. But, they are not so effective to forecast attractors with higher embedded dimension. The paper proposes a new idea based on incidence-degree to determine the nearest point in phase space. In the mean time, an artificial neural networks model based on particle swarm optimization (PSO) learning algorithm is presented for load forecasting. The proposed method has been examined and tested on a practical power system. The test result shows that the precision of load forecasting is improved by means of the new method when the embedded dimension is higher.

Key-Words: -Load forecasting, Particle Swarm Optimization, ANN

1. Introduction

Short-term load forecasting plays an important role in power system operation and planning. Generally, short-term load forecasting is always a difficult task in practice, because the load is affected by a variety of nonlinear factors such as weather conditions, daily, weekly and seasonal periodicity, etc. Nowadays, more and more scholars think that considering the nonlinear factors in the load forecasting modeling is the key to improving the load forecasting level. Recently, the theory of nonlinear chaos dynamics which links the determination and randomicity has become the foreland of load forecasting study, many scholars do lots of useful explore for load forecasting[1]-[4]. Specially, combining phase space restructures theory with neural networks is considered as one of the most effective methods[5][6]. However, the embedded dimension will be relatively high, because load forecasting is influenced by various intricate facts. The traditional chaos methods are proved of high precision to forecast time series with low-embedded dimension. But, they are not so effective to forecast attractors with high-embedded dimension [7][8]. On the other hand, it is attractive to find a fast network convergent arithmetic for attractors with high-embedded dimension.

The paper proposes a new idea based on incidence-degree to determine the nearest point in phase space. In the mean time, an artificial neural networks model based on particle swarm optimization (PSO) learning algorithm is presented for load forecasting. The proposed method has been examined and tested on a practical power system. The test result shows that the precision of load forecasting is improved by means of the new method when the embedded dimension is higher.

The paper was organized as follows: section 2 describes a phase space neural networks forecasting model suit for higher embedded dimension; an improved ANN model with particle swarm optimization learning algorithm is shown in section 3; followed by numerical examples in section 4, and conclusion in section 5.

2. PHASE SPACE NEURAL NETWORKS FORECASTING MODEL SUIT FOR HIGHER EMBEDDED DIMENSION

Now, there are several forecast ways based on chaos theory, such as local linear approximation method, linear interpolation method, artificial neural networks method and so on. All these models are based on the phase space restructures theory proposed by Takens in 1981.

For a time series $\{x_0, x_1, \cdots, x_{N-1}\}$, we set it in a d dimensional phase space according to the phase space restructures theory:

$$X_t = (x_{(t-1)\tau+1}, x_{(t-2)\tau+1}, \cdots, x_{(t-d)\tau+1}) \qquad (1)$$

To simplify the presentation, let $\tau = 1$, then:

$$X_t = (x_t, x_{t-1}, \cdots, x_{t-d+1}) \qquad (2)$$

where X_t is one state in the d-order state space. According to the famous Takens Theory [9], when $d \geq 2m+1$ (m is the order of the attractor), there exists a deterministic mapping with dimension being d: $F^{(d)}: R^d \to R^d$. It describes the evolving track of X_t in the state space and the mapping has the same geometric structure and topology with the original system. So:

$$X_{t+1} = F^{(d)}(X_t) \qquad (3)$$

(3) can also be presented as:

$$x_{t+1} = \tilde{F}^{(d)}(x_t, x_{t-1}, \cdots, x_{t-d+1}) \qquad (4)$$

The state space described by $\tilde{F}^{(d)}$ is called as the restructured state space. d is called as the embedded dimension, which is the dimension of the minimal state space that can completely contain the attractor sets comprised by state transition. Because $\tilde{F}^{(d)}$ is deterministic, we are only needed to made an estimate on $\tilde{F}^{(d)}$ in order to forecast x_{t+1}. The forecasting model combining phase space restructures theory and artificial neural networks as follows:

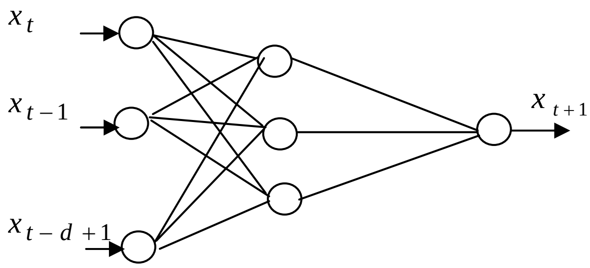

Fig. 1 **ANN Forecasting Model**

By selecting a suitable set of weights and transfer functions, it is known that the ANN can approximate any smooth, measurable function between the input and output vectors. In this paper, an output mean squared error (MSE) of ANN is considered and defined as:

$$MSE = \frac{1}{N}\sum_{s=1}^{N}\left[\sum_{k=1}^{M}(o_{sk} - l_{sk})^2\right] \qquad (5)$$

where o_{sk} is the expected output, l_{sk} is the predicted output, M is the number of output neurons, and N is the number of training set samples.

Now, the problem is how to find the K the nearest points to training neural networks, the current chaos forecasting methods are often based on the nearest points method. Those methods are based on Euclid distance:

$$|X_t - X_{t1}| = \sqrt{(x_t - x_{t1})^2 + (x_{t-1} - x_{t1-1})^2 + \cdots + (x_{t-d+1} - x_{t1-d+1})^2} \qquad (6)$$

and find K nearest points to X_t in the state space to approach the load evolving track. The above algorithm depends in a large degree on the nearest points found according to Euclid distance method. If the nearest points are related closely to the original state, the forecasting precision is high. Otherwise, it is low. When the embedded dimension d is small, the nearest points found according to Euclid distance method can approximately reflect the relationship with the original point. But when the dimension is increased, such close relationship will be decreased because the nearest distance doesn't mean the greatest relationship. This paper proposed to substitute the Euclid distance with the incidence degree. The incidence degree can evaluate the relationship according to the similarity of curves. The larger the incidence degree is, the better the fitting is. Relating degree is an efficient method to deal with high embedded dimension.

Let X_0, X_1, X_2 being three points in the state space with dimension of *d*.

Define:

$$\xi_i(k) = \frac{\min_i \min_k |x_0(k) - x_i(k)| + \rho \max_i \max_k |x_0(k) - x_i(k)|}{\left| x_0(k) - x_i(k) + \rho \max_i \max_k |x_0(k) - x_i(k)| \right|} \quad (7)$$

($k = 1,2,...,d$: generally ρ =0.5)

as the incidence degree coefficient between points X_0 and X_i at the *k*th element.

We call:

$$r_i = \frac{1}{d} \sum_{k=1}^{d} \xi_i(k) \quad (8)$$

as the incidence degree between point X_i and the reference point X_0. The larger the incidence degree is, the higher the similarity is.

3. Training ANN Using PSO

The above-mentioned method is very easy and feasible, but it still has disadvantage: the networks will converge very slowly when using common training arithmetic, especially, when the dimension *d* becomes enough large, the phenomenon is obvious. This paper proposes an improved particle swarm algorithm for training neural network.

Particle swarm optimization (PSO) was first introduced by Kennedy and Eberhart in 1995 [10]. Like evolutionary algorithms, PSO technique conducts search using a population of particles, corresponding to individuals. Each particle represents a candidate solution to the problem at hand. Now, particle swarm optimization has become the focus of research [11]-[16].During the calculation, the particle is affected by three factors when it is moving in space. One of the factors is the particle's current velocity $V(t)$. Another is the optimal point $pbest_i = (pbest_{i,1}, pbest_{i,2}, \cdots, pbest_{i,j})$ where the particle has reached before. The third factor is the optimal point $gbest = (gbest_1, gbest_2, \cdots, gbest_j)$ of the community or the sub-community. The particle's velocity is changed towards $pbest_i(t)$ and $gbest(t)$ in every iteration step. Meanwhile, V_i 、 $pbest_i(t)$ and $gbest(t)$ are assigned separately a weight at random. The velocity and position is updated according to the formula (9) and (10).

$$\begin{aligned} v_{i,j}(t) &= w \times v_{i,j}(t-1) \\ &+ c_1 \times r_1 \times (pbest_i(t-1) - x_{i,j}(t-1)) \\ &+ c_2 \times r_2 \times (gbest(t-1) - x_{i,j}(t-1)) \end{aligned} \quad (9)$$

$$x_{i,j}(t) = x_{i,j}(t-1) + v_{i,j}(t) \quad (10)$$

($i = 1,2,\cdots n \quad j = 1,2,\cdots m$)

Where,

c_1, c_2 are the learning factors, generally, $c_1 = c_2 = 2$.

w is the weight scale operator.

r_1, r_2 are the randoms within the interval of [0,1].

t is the number of iteration.

n is the number of particles.

m is the number of dimensions.

It is assumed that the three-layered perceptrons are chosen for all application cases in this study. W_1 is the connection weight matrix between the input layer and the hidden layer, W_2 is the connection weight matrix between the hidden layer and the output layer. The performance of each individual is measured according to a fitness function. The fitness function can be calculated by

$$f(W_1,W_2) = MSE(W_1,W_2) \qquad (11)$$

The procedure of the self-adaptive PSO for combined forecasting model weight optimization can be described as follows.

Step1 Initialization:

Set t=0. Let $X_i = \{W_1^j, W_2^j\}$ be a particle, generate randomly n particles $\{X_i(0), i=1,\cdots n\}$ (set n to 20 in this paper). All particles are set between the lower and upper limits. Similarly, generate randomly initial velocities of all particles, $\{\vec{V}_i(0), i=1,\cdots n\}$, where $\vec{V}_i(0)=\{\vec{v}_{i,1}(0),\cdots,\vec{v}_{i,m}(0)\}$. $\vec{v}_{i,k}(0)$ is generated by randomly selecting a value with uniform probability over the kth dimension $\left[-\vec{v}_k^{\max}, \vec{v}_k^{\max}\right]$. Each particle in the initial population is evaluated using the equation (11). For each particle, set $pbest(0) = X_i(0)$ and $f_i^* = f_i, i = 1,\cdots n$.

Let $f^{**} = \min\{f_1^*, \cdots f_n^*\}$. Set the particle associated with f^{**} as the global best $gbest(0)$.

Step2 Velocity and Position updating:

Let t=t+1. Using the global best and individual best of each particle, the ith particle velocity and position in the jth dimension is updated using the equation (12)-(13).

$$\begin{aligned} v_{i,j}(t) &= w \times v_{i,j}(t-1) \\ &+ c_1 \times r_1 \times (pbest_i(t-1) - x_{i,j}(t-1)) \\ &+ c_2 \times r_2 \times (gbest(t-1) - x_{i,j}(t-1)) \end{aligned} \qquad (12)$$

$$x_{i,j}(t) = x_{i,j}(t-1) + v_{i,j}(t) \qquad (13)$$

Step3 Individual and global best updating:

Each particle is evaluated according to its updated position.

If $f_i < f_i^*, i = 1,\cdots,n$, then

$$pbest_i(t) = X_i(t)$$
$$f_j^* = f_j$$

Else go to *Step3*

Search for the minimum value $f_{\min}$ among f_j^*.

If $f_{\min} < f^{**}$ then

$$gbest(t) = X_{\min}(t)$$
$$f^{**} = f_{\min}$$

Else go to *Step3*.

Step4 Stopping criteria:

If one of the stopping criteria is satisfied, then stop. Else go to *Step2*.

4. Numerical examples

Case studies for the proposed method were carried out for load forecasting using different historical data of shanghai grid of East China area in 2000. Dimensions of attractors are calculated by means of G-P algorithm which is put up by Grassberger and Procaccia, and the method which is put up by Wolf[17] to calculate Lyapunov exponential is also utilized. The corresponding parameters for the load series:

Maximum Lyapunov index: $\lambda_{\max} = 0.00392$;

Embedded dimension: D=6;

Time delay: $\tau = 1$.

The 3-dimension reconstructed state space of original load series are shown in Fig.2, Fig.3 shows the certain day's prediction results produced by the paper's method (the real line represents the real load value, and the dashed line the forecast value). Table.1 show the forecast result of 14 consecutive days.

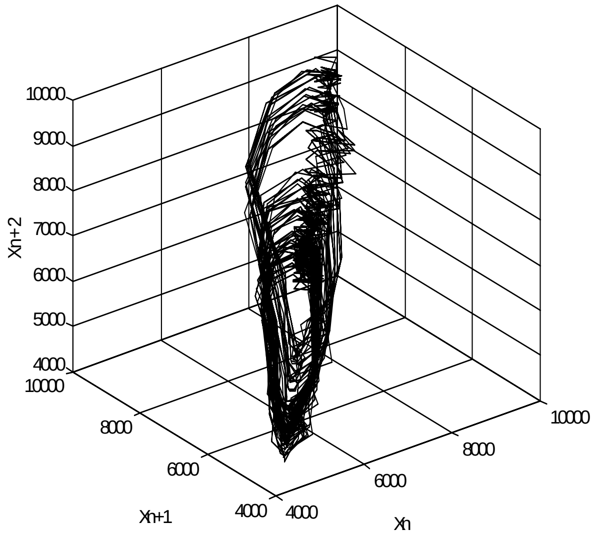

Fig. 2 Three-Dimension Reconstruction of Load

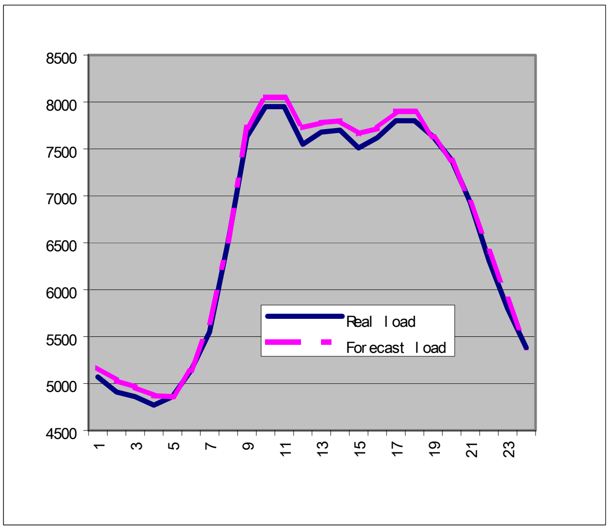

Fig 3. the certain day's prediction results

Table.1 the 14 consecutive days' forecast results

day	1	2	3	4	5	6	7
Mean Error(%)	1.45	2.16	2.04	1.87	2.24	1.67	1.52
Max Error(%)	4.91	3.21	4.42	3.06	-4.65	-4.17	-5.31
day	8	9	10	11	12	13	14
Mean Error(%)	1.32	2.04	1.18	1.60	1.39	2.04	2.16
Max Error(%)	3.68	4.40	-4.92	-4.32	-5.43	-5.11	3.89

Each algorithm is iterated 20 times in order to compare the PSO with BP in terms of the convergence character and the computation speed. Tab.2 gives the average values for comparison showing that the PSO is more efficient than BP.

Table 2. Performance of PSO and BP

	Mean iterative	Mean time (s)
BP	>130	15.5
PSO	<40	5.7

5. Conclusion

The chaotic load series with high-embedded dimension is very common in the nature. Therefore it is useful to study the forecasting methods of high-embedded dimension. The paper proposes a new idea based on incidence-degree to determine the nearest point in phase space. In the mean time, an artificial neural networks model based on particle swarm optimization learning algorithm is presented for load forecasting. The proposed method has been examined and tested on a practical power system. The results show that the method plays an important role to improve the precision of forecasting of load series with higher embedded dimensions.

References

[1] Iokibe,T., Fujimoto,Y., Kanke,M., and Suzuki,S.: Short-term prediction of chaotic time series by local fuzzy reconstruction method, *J.Intell.Fuzzy Systems* 5(1997),3-21

[2] James McNames,: Local averaging optimization for chaotic time series prediction, *Neurocomputing*, 48(2002), 279-297

[3] Mori H, Urano S. Short-term load forecasting with chaos time series analysis, *Proceedings of the International Conference on Intelligent Systems Applications to Power Systems, ISAP '96*, 28 Jan.-2 Feb. 1996, Page(s): 133 -137.

[4] Jiang Chuanwen, Wang Chengmin, Ma Yuchao. Short-term load nonlinear forecasting with high-embedded dimensions using wavelet decomposing and chaos theory. *Series on Energy and Power Systems*, *Proceedings of the Fourth IASTED International Conference on Power and Energy Systems*, 2004, p 41-45

[5] Spyros Tzafestas, Elpida Tzafestas. Computational intelligence techniques for short-term electric load forecasting, *Journal of Intelligent and Robotic Systems*,31(2001),7-68

[6] Moghram I, Rahman S. Analysis and evaluation of five short-term load forecasting techniques. *IEEE Trans Power Syst*, 4:1484–1491 (1989).

[7] Jiang Chuanwen, Li Tao. Forecasting method study on chaotic load series with high embedded dimension. *Energy Conversion and Management*, 46(5), 2005, 667-676.

[8] Hou yexian, He Pilian and Wang Lei.: Reconstructing high dimension phase space an improved approach to chaotic time series forecasting, *Journal of Tianjin University*, 32(5), pp. 594-599, 1999

[9] F. Takens.: Detecting strange attractors in turbulence, Dynamical Systems and Turbulence, Warwick 1980,Lecture Notes in mathematics, No.898, Edited by D.A. Rand and L-S Yong, Springer-Verlay, Berlin, pp. 366-381, 1981

[10] Kennedy J, Eberhart R. Particle Swarm Optimization. *In Proceedings of IEEE International Conference on Neural Networks*, 1995(4): 1942~1948

[11] Shi Y, Eberhart R. A modified particle swarm optimizer. *In: IEEE World Congress on Computational Intelligence*, 1998: 69-73

[12] Shigenori Naka, Takamu Genji, Toshiki Yura, Yoshikazu Fukuyama. A Hybrid Particle Swarm Optimization for Distribution State Estimation. *IEEE Transactions on Power Systems***,** 2003(18): 60~68

[13] Jiang Chuanwen, Etorre Bompard. A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimization. *Mathematics and Computers in Simulation*, 68(1), 2005, 57-65.

[14] M.A.Abido. Optimal power flow using particle swarm optimization. *Electrical Power and Energy Systems*, 2002,(24):563-571

[15] Jiang Chuanwen, Etorre Bompard. A self-adaptive chaotic particle swarm algorithm for short term hydroelectric system scheduling in deregulated environment. *Energy Conversion and Management*. 2005,46(17), *2689-2696*

[16] Ioan Cristian Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 2003,(85):317-325

[17] A. Wolf, Determining Lyapunov Exponents From A Time Series, *Physica D*.16, pp. 285-317, 1985