

Time Series Forecasting Based on Cloud Process Neural Network*

Bing Wang[†]

*School of Computer & Information Technology, Northeast Petroleum University
Daqing, 163318, China[‡]
E-mail: wangbing0812@sina.com*

Shaohua Xu

*College of information Science and Engineer, Shandong University of Science and Technology
Qingdao, 266590, China
E-mail: xush62@163.com*

Xiaohong Yu

*Exploration and Development Research Institute, Daqing Oilfield Company
Daqing, 163712, China
E-mail: yuxiaohong@sina.com*

Panchi Li

*School of Computer & Information Technology, Northeast Petroleum University
Daqing, 163318, China
E-mail: lipanchi@vip.sina.com*

Received 2 March 2015

Accepted 20 August 2015

Abstract

Time series forecasting has been an important tool in many areas such as agriculture, finance, management, production or sales. In recent years, a large literature has evolved on the use of artificial neural networks (ANN) in time series forecasting. However conventional ANN is limited by its instantaneous synchronization inputs, it is difficult to express accumulative time effect and lacks certain processing ability for uncertainty factors (e.g., randomness, fuzziness) hidden in time series. Thus a cloud process neural network (CPNN) model is put forward in the paper for time series forecasting. It combines cloud model's expression ability for uncertainty concepts and process neural network's dynamic signal processing method, converts quantitative time series inputs into multiple qualitative sub-cloud concepts, and then finds out the association rule between input and output variables through mining inherent law among multiple sub-clouds. For CPNN learning, this paper proposes a learning strategy based on cat swarm optimization algorithm, which could optimize the network structure and learning parameters simultaneously to improve the network approximation and generalization ability. Finally, the model and algorithm is used in individual household electric power consumption time series forecasting and ASP flooding oil recovery index forecasting. In order to improve the quality of training samples, phase space reconstruction theory is employed to reconstruct one-dimensional time series into high-dimensional phase space as training sample set.

* For the title, try not to use more than 3 lines. Typeset the title in 12 pt Times Roman, uppercase and boldface.

[†]Typeset names in 10 pt Times Roman, uppercase. Use the footnote to indicate the present or permanent address of the author.

[‡]State completely without abbreviations, the affiliation and mailing address, including country typeset in 10 pt Times italic.

Simulation results show that compared to conventional process neural networks and adaptive neuro fuzzy inference system, the proposed method improves the prediction accuracy and provides a new solution for time series pattern classification and forecast analysis.

Keywords: Cloud process neural network; Time series forecasting; Cloud theory; Cat swarm optimization; Phase space reconstruction

1. Introduction

Time series forecasting is an important tool to support individual and organizational decisions making, and has become increasingly used in areas such as agriculture, finance, management, production or sales. Conventional time series forecasting methods mainly use regression analysis due to its mature theory, but its accuracy and fault tolerance are not very high. Since multi-layer feed-forward neural network was proved to be an uniform approximation of continuous functions^{1,2}, applications of artificial neural network for time series forecasting draws public attention. As a new time series forecasting method, for its good nonlinear property, parallel distributed storage structure and high fault tolerance, artificial neural network has achieved success in many practical applications^{3,4}. Time series forecasting is an extended forecasting method for historical data, and the historical data changes over time and also presents some randomness and fuzziness during its change. However conventional neural network is limited by its instantaneous synchronization inputs, it is difficult to express accumulative time effect and lacks certain processing ability for uncertainty factors (e.g., randomness, fuzziness) hidden in time series. Thus there exists some inadaptability for conventional artificial neural network to solve complicated nonlinear time series forecasting.

The goals of this paper are to design an ANN, which can catch and deal with randomness, fuzziness and accumulative time effect in time series simultaneously, and to develop an algorithm, which can validly identify the network structure and learning parameters at the same time. Thus a cloud process neural network (CPNN) model with discrete time series as its direct inputs is built in the paper by combining cloud model's expression ability for uncertainty concepts with process neural network's dynamic signal processing method. In order to improve the quality of training samples, a training sample construction method based on phase space reconstruction theory is put forward, and cat swarm optimization (CSO) algorithm is used to

optimize the network structure and learning parameters in parallel. Finally the model and algorithm is applied to individual household electric power consumption time series forecasting and ASP flooding oil recovery index forecasting, and satisfactory results are achieved.

The paper is organized as follows. Sec.2 reviews the related works about process neural networks and cloud theory. Sec.3 introduces basic knowledge about cloud model. Sec.4 gives the topology structure of CPNN model. Sec.5 explains how the structure and parameters of the CPNN is optimized by CSO. In Sec.6, phase space reconstruction method is used to construct experimental samples, and experimental setup and results are shown. And finally, conclusions are described in Sec.7.

2. Related Work

Process neural network (PNN) is a new artificial neural network model proposed by He⁵ at the beginning of this century. Its inputs, outputs and connection weights can all be functions related to time or process; its aggregation and activation operation can reflect both spatial weighted aggregation to time-varying input signals and the accumulative effect at time axis. It is an extension to conventional neural networks in time domain and helps to solve the problems associated with time or process. At the aspect of model construction, Refs. 6-10 put forward parallel, wavelet, Elman, multi-aggregation and quantum network models successively. Due to there is no unified expression on the network input function and connection weight function, it becomes the basis of PNN training and learning to execute basis expansion for the network input and weight function. Learning algorithms based on basis expansion with gradient descent¹¹⁻¹³ (e.g., Walsh basis function, Fourier basis function, Legendre basis function and Spline function) effectively simplify spatio-temporal aggregation operation. In recent years, optimization technique based on evolutionary computation is widely used in various engineering problems. In Refs. 9,14,15 genetic evolution, quantum optimization and particle swarm optimization algorithm

are also applied into PNN training respectively, and PNN approximation performance is better improved. For the network performance, Refs. 16,17 study its theoretical properties such as continuity, approximation and computing ability, and have proved that PNN could approximate any continuous functional. Therefore it is greatly adaptive to solve time series forecasting by PNN. At present some researchers have applied PNN in time series forecasting^{18,19} successfully.

Most of things and phenomena in the objective world have uncertainty, people usually research it from two aspects of fuzziness and randomness respectively, but randomness and fuzziness have high relation and are often inseparable, e.g., people usually seek membership function by statistical method in fuzzy set theory. Cloud model was proposed by Li^{20,21} on the interaction of probability theory and fuzzy mathematics. It is a transformation model between qualitative concept and quantitative expression through specific structure and algorithm. The cloud model reflects not only the concepts' uncertainty in natural language, but also the relevance between randomness and fuzziness, and has already been successfully applied in uncertainty reasoning, intelligent control, decision support, spatial clustering, and system evaluation²²⁻²⁴ etc. We will integrate cloud model into process neural networks which can only express and deal with precise and quantitative information in the next paper, it can thereby automatically extract random and fuzzy information in time series in order to improve the prediction accuracy.

3. Cloud Theory

Let U be a quantitative numerical domain, C is a qualitative concept on U . If the quantitative value $x \in U$, and x is a random realization of the qualitative concept C , then the certainty degree $\mu(x) \in [0, 1]$ of x to C is a random value with stable tendency. If $\mu: U \rightarrow [0, 1]$, $\forall x \in U, x \rightarrow \mu(x)$, then the distribution of x on the domain U is called a cloud. Each x is called a cloud droplet²⁰. The certainty degree of x to C is a probability distribution, rather than a constant value, thus the resulting cloud is not a clear curve. The certainty degree of a cloud droplet reflects how much it could represent the qualitative concept. The greater the certainty degree, the greater the contribution of a cloud droplet to the concept is.

A cloud is fully characterized by 3 digital features, that is, expectation (Ex), entropy (En) and hyper entropy (He).

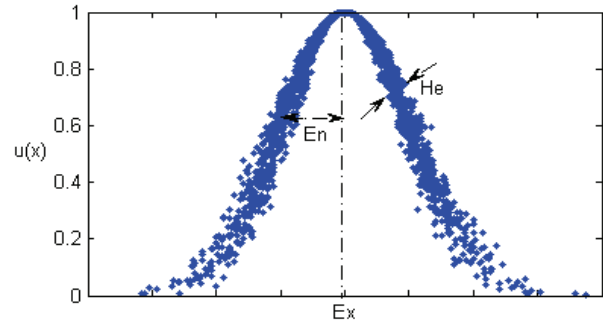


Fig. 1. Cloud model

Ex is an expectation of cloud droplets on the domain U . It is a central value of the concept on U , and a most representative point for qualitative concept. En is a fuzzy degree measurement for the qualitative concept, and it reflects the range of cloud droplets accepted by the concept in domain space. The greater the entropy, the bigger the value range accepted by the concept. He is an uncertainty measurement for En , namely the entropy of En . The greater the He , the stronger the discrete degree of cloud droplets and the randomness of the membership degree are.

There are different kinds of cloud models, such as symmetric cloud model, semi-cloud model, combination cloud model, 2-D cloud model and normal cloud model, etc. in which normal cloud model is the most basic cloud model, and it is one of powerful tools to characterize language atom and has universality. The mathematical expectation curve of the normal cloud could be determined by Ex and $En: \mu(x) = e^{-(x-Ex)^2/2En^2}$. In the view of fuzzy set theory, this curve is the membership curve of normal cloud model.

4. Cloud Process Neural Network Model

Cloud process neural network model is composed of time series input layer, cloud process layer, rule layer, standardization layer and output layer, its topology structure is shown in Fig.2. A general form of cloud reasoning rule the network model represents is as follows:

R_k : if X_1 is $C_{1k}(\mu_1^k)$, X_2 is $C_{2k}(\mu_2^k)$, ..., X_n is $C_{nk}(\mu_n^k)$, then $y = \bar{y}_k$

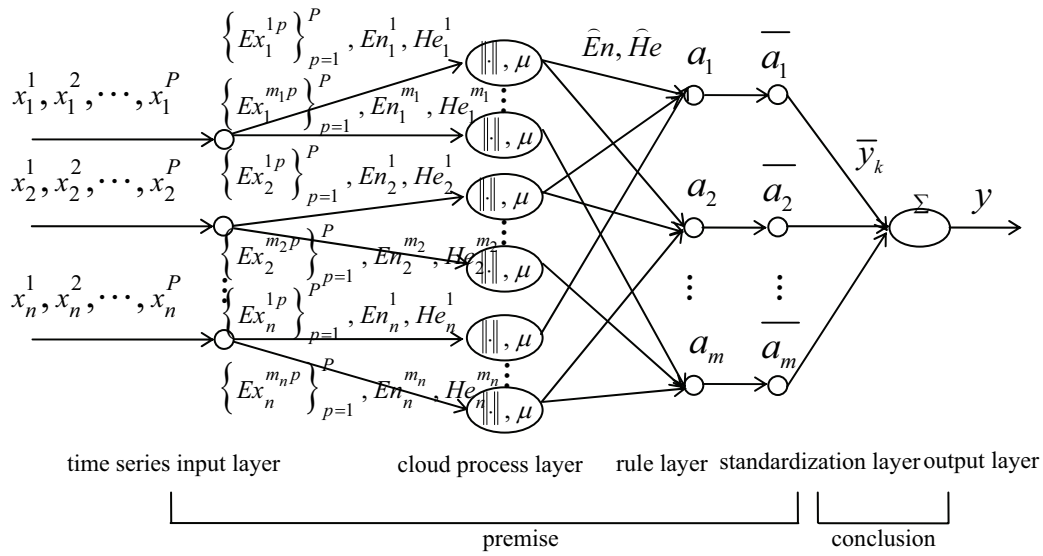


Fig.2. Cloud process neural network model

- (i) Time series input layer: Transfer one-dimensional or multi-dimensional time series into cloud process layer. Let the network input space be $(C[0, T])^n$, there exists P sampling points in the process interval $[0, T]$, then n discrete inputs with the length of P are denoted as $X_i = (x_i^1, x_i^2, \dots, x_i^P), i = 1, 2, \dots, n$.
- (ii) Cloud process layer: Each node represents a sub-cloud model $C_{ij}(\mu_i^j)$ which handles time series pattern from input layer and computes the certainty degree μ_i^j of each input component X_i for its sub-cloud.

$$\mu_i^j = e^{-\frac{\|X_i - Ex_i^j\|^2}{2(En_i^j)^2}} \quad (1)$$

where $Ex_i^j = (Ex_i^{j1}, Ex_i^{j2}, \dots, Ex_i^{jP})$, $i = 1, \dots, n, j = 1, \dots, m_i$, here m_i is the number of discrete sub-clouds the i^{th} time series input is softly divided into. Ex_i^j, En_i^j, He_i^j are 3 digital features of each sub-cloud model. En_i^j is a normal random value with the expectation En_i^j and the standard deviation He_i^j , namely

$En_i^j \sim N(En_i^j, (He_i^j)^2)$. $\|X_i - Ex_i^j\|$ is a measurement norm used to measure the matching degree between X_i and Ex_i^j in process sampling interval $[0, T]$, e.g. Euclidean distance, discrete Fréchet distance or mean Hausdorff distance.

- (iii) Rule layer: Each node represents “soft-and” operation and matches cloud rule to compute the activation degree a_k of this rule,

here $k = 1, 2, \dots, m$, $m = \prod_{i=1}^n m_i$. “Soft-and” operation could be expressed by multi-dimensional normal cloud model $C(1, \hat{En}_1^{s_1}, \hat{He}_1^{s_1}, 1, \hat{En}_2^{s_2}, \hat{He}_2^{s_2}, \dots, 1, \hat{En}_n^{s_n}, \hat{He}_n^{s_n})$ where $\hat{En}_i^{s_i}$ and $\hat{He}_i^{s_i}$ are the regulation parameters about “soft-and” degree.

$$a_k = e^{-\left(\frac{(\mu_1^{s_1}-1)^2}{2(\hat{En}_1^{s_1})^2} + \frac{(\mu_2^{s_2}-1)^2}{2(\hat{En}_2^{s_2})^2} + \dots + \frac{(\mu_n^{s_n}-1)^2}{2(\hat{En}_n^{s_n})^2}\right)} \quad (2)$$

where $s_i = 1, 2, \dots, m_i; i = 1, 2, \dots, n$. $\hat{En}_i^{s_i}$ is a normal random value with the expectation $\hat{En}_i^{s_i}$ and the standard deviation $\hat{He}_i^{s_i}$, namely $\hat{En}_i^{s_i} \sim N(\hat{En}_i^{s_i}, (\hat{He}_i^{s_i})^2)$.

- (iv) Standardization layer: Standardize the activation degree a_k from the rule layer.

$$\bar{a}_k = a_k / \sum_{i=1}^m a_i, k = 1, 2, \dots, m \quad (3)$$

(v) Output layer: Integrate standardized activation degree for output by weighted mean method.

$$y = \sum_{k=1}^m \bar{a}_k \bar{y}_k \quad (4)$$

The network is a multi-layer feed-forward neural network, and there are 3 groups of parameters to be adjusted, that is, the digital features $\{Ex_i^{jp}\}_{p=1}^P, En_i^j, He_i^j$ of each sub-cloud in cloud process layer, “soft-and” degree regulation parameters $\hat{E}n_i^{s_i}, \hat{H}e_i^{s_i}$ and \bar{y}_k in output layer. Premise part of the rules is implemented from the time series input layer to the standardization layer, and conclusion part is realized by the output layer.

5. Cloud Process Neural Network Learning Algorithm

For CPNN training, traditional learning algorithms based on basis function expansion with gradient descent¹² could be used, but there is not a uniform guideline on the choice of basis function and how to determine the expansion item number of the basis function. In addition, gradient descent method is easy to fall into local optimum, and may suffer from incalculable gradient. In fact, in addition to parameter learning, CPNN training should also learn the network structure, namely the learning process should involve two parts of structure and parameter adjustment. There exists more error through traditional experience or trial-and-error method to adjust the network structure. Therefore, how to determine the optimal network structure and optimize network learning parameters simultaneously to obtain the global optimal solution quickly and accurately is the following well-studied problem.

5.1. CSO fundamental

CSO algorithm²⁵ is presented in recent years. It is a new intelligent optimization algorithm based on swarm behavior, and gives much better performance than particle swarm algorithm and genetic algorithm, etc. In CSO, each cat has its own position, velocity, fitness value and working mode. A cat's position is composed of M -dimensional vector where M is the length of variables in optimization problem, and the M -dimensional component has its own velocity. Fitness value is obtained by evaluating fitness function, and ultimately the position with the optimal fitness value is

the solution of the optimization problem. During iteration, the cats' positions move into the direction of the optimal solution, and the global optimal solution is obtained finally.

5.2. Design of network objective function

Give S samples $\{(\hat{X}_s, d_s)\}_{s=1}^S$ where

$$\hat{X}_s = (X_1^s, X_2^s, \dots, X_n^s), X_i^s = (x_i^{1s}, x_i^{2s}, \dots, x_i^{Ps}).$$

Let the real output corresponding to the s^{th} sample be y_s , the network training objective function based on minimum MSE is defined as

$$E = \sum_{s=1}^S (y_s - d_s)^2 / S \quad (5)$$

where y_s could be calculated according to Eqs.(1)-(4).

In conclusion, the training objective function E of CPNN is a function on structure parameter m_i and learning parameters $Ex_i^{jp}, En_i^j, He_i^j, \hat{E}n_i^{s_i}, \hat{H}e_i^{s_i}, \bar{y}_k$. Here we use E as the fitness function of CSO, then the training problem of CPNN is converted into an optimization problem to minimize the function E .

5.3. Algorithm description

Step 1 Initialization: Set the size of the cat swarm N , the maximum iteration number T and present iteration number $t = 0$. Initialize the position and velocity of N cats randomly. The position of a cat could be denoted as

$$\mathbf{X} = (m_i, Ex_i^{jp}, En_i^j, He_i^j, \hat{E}n_i^{s_i}, \hat{H}e_i^{s_i}, \bar{y}_k) \\ i = 1, \dots, n; j = 1, \dots, m_i; p = 1, \dots, P;$$

$$s_i = 1, 2, \dots, m_i; k = 1, 2, \dots, \prod_{i=1}^n m_i \quad (6)$$

The dimension of a cat's position is

$$M = n + \sum_{i=1}^n m_i(P + 2) + \prod_{i=1}^n m_i(2n + 1) \quad (7)$$

Step 2 Allocation: Haphazardly pick number of cats and set them into tracing mode according to MR (mixture ratio), and the others set into seeking mode.

Step 3 Evaluation: Evaluate the fitness value of each cat by applying the positions of cats into the fitness function, which represents the criteria of our goal, and remember the position of the best cat (X_{best}) due to it represents the best solution so far.

Step 4 Movement: Move the cats according to their flags.

(i) Seeking mode

- (a) Duplication: Make SMP copies of the present position of cat c into seeking memory pool. If the value of SPC (self position consideration) is true, let $SMP = (SMP - 1)$, then retain the present position as one of the candidates.
- (b) Mutation: For each copy, according to CDC (counts of dimension to change), randomly plus or minus SRD (seeking range of the selected dimension) percents the present values and replace the old ones.
- (c) Evaluation: Calculate the fitness values of all candidate points.
- (d) Calculation: If all fitness values are not exactly equal, calculate the selecting probability of each candidate point by Eq.(8), otherwise set all the selecting probability of each candidate point be 1.

$$P_i = \frac{FS_{\max} - FS_i}{FS_{\max} - FS_{\min}}, i = 1, 2, \dots, SMP \quad (8)$$

- (e) Movement: Randomly pick the point to move to from the candidate points, and replace the position of cat c .

(ii) Tracing mode

- (a) Update the velocities for every dimension of cat c according to Eq. (9).

$$V_c(t+1) = V_c(t) + r_1 \cdot c_1 \cdot (X_{best} - X_c(t)) \quad (9)$$

where c_1 is a constant and r_1 is a random value in the range of [0,1].

- (b) Check if the velocities are in the range of maximum velocity V_{\max} . In case the new velocity is over-range, it is set equal to the limit.
- (c) Update the position of cat c according to Eq.(10).

$$X_c(t+1) = X_c(t) + V_c(t+1) \quad (10)$$

Step 5 Judgment: If $t > T$ or the objective function E is smaller than the setting value, then stop; otherwise $t = t + 1$ and return to Step 2.

5.4. Learning algorithm analysis

In above learning algorithm, reasonable limit to searching space of the cats helps to optimize the network structure and learning parameters. It can be seen from Eq. (7) that n is unchanged in an experiment, so the dimension size of cats' position depends on the structure parameter m_i . If m_i is too large, there are too many variables to be optimized, it will not only impact the algorithm performance, even may produce

dimension disaster. If m_i is too small, the network structure will be too simple to approximate the objective. The initial m_i could be set by some clustering methods

(e.g. k-means) in advance. For the expectation Ex_i^{jp} of each sub-cloud, two approaches could be adopted to set their initial values: one is to extract part of samples from the inputs of each sub-cloud randomly as their own initial expectations; the other is to use the cluster center of k-means method. The values of the entropy and hyper entropy $En_i^j, He_i^j, \widehat{En}_i^{s_i}, \widehat{He}_i^{s_i}$ are positive real values, and in order to prevent atomization, the hyper entropy should be smaller than 1/3 of its entropy.

The search strategy of CSO itself is not complicated; its computational complexity lies in evaluating the fitness, namely the network objective function E , for every candidate or update position in each iteration. So the computational complexity of the algorithm mainly depends on the cat swarm size N , the iteration number t and the size of seeking memory pool SMP . The number of computing the objective function is $\leq Nt(SMP + 1)$. In addition, the calculation of E is relevant to m_i . A good tradeoff between optimal performance and computation work will be found by setting reasonable value or range for N, t, SMP, m_i .

In a word, CSO algorithm makes the CPNN learning more holistic and global, and it avoids the premature phenomenon effectively. In addition, it solves incalculable gradient of the objective function using discrete Fréchet distance and mean Hausdorff distance, etc. The CPNN with CSO training has good approximation performance and generalization ability.

6. Application Simulation

6.1. Individual household electric power consumption forecasting

6.1.1. Experimental data selection

Individual household electric power consumption data set from UCI is used for simulation. This archive contains 2075259 measurements gathered between December 2006 and November 2010 (47 months) with a one-minute sampling rate over a period of almost 4 years and is always used to test forecasting and classification ability for time-varying samples.

Short-term forecasting about week data of electric power consumption will be done in the experiment. Because of partial lack of original data, for missing data we use intact latest data above to fill. Then 205 groups

of week data are gotten based on minute sampling data, as shown in Fig.3.

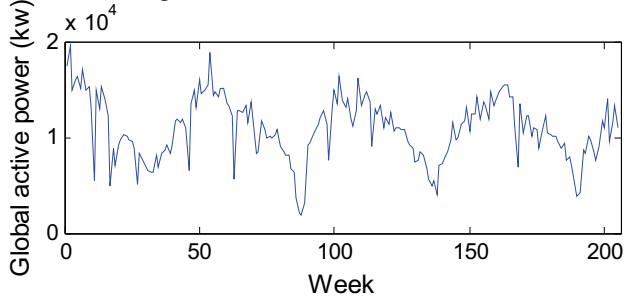


Fig.3. Week time series data

6.1.2. Phase space reconstruction-based data preprocessing

If the electric power consumption time series is seen as being produced by a deterministic nonlinear dynamical system, the task of CPNN is how to restore and depict original dynamical system by use of this time series. To fully reveal hidden information in given electric power consumption time series, here phase space reconstruction method²⁶ is used to restore or approximatively simulate the dynamics characteristics of the system, preprocess electric power data and construct experimental sample set in the paper.

Given a one-dimensional time series $\{l_i\}_{i=1}^N$, the reconstructed phase space is $L_k = (l_k, l_k + \tau, l_k + 2\tau, \dots, l_k + (m-1)\tau)$ where τ is time delay, m is embedded dimension, $k = 1, 2, \dots, M$, $M = N - (m-1)\tau$. Some researches show that as long as τ, m is reasonable, reconstructed phase space has the same topology nature as original system. Here we employ autocorrelation function method and G-P algorithm²⁷ to compute τ, m respectively. The first zero crossing value of autocorrelation coefficient is used as the best time delay of reconstructed phase space, as shown in Fig. 4. It is observed that the best time delay for week electric power data is $\tau = 13$.

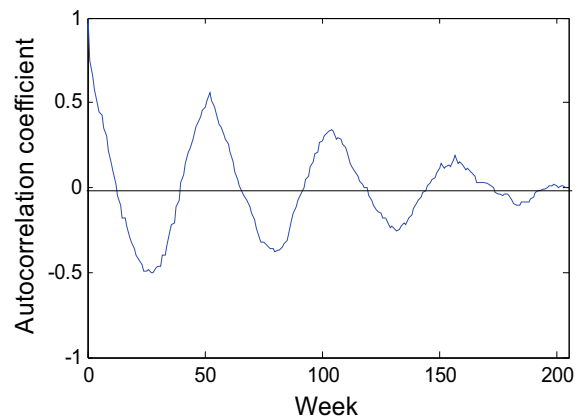


Fig.4. Week time series autocorrelation coefficient

Base on the best time delay, set $m = 1, 2, \dots, 12$.

For each m , compute and get a series of values about $\ln r, \ln C(r)$ according to G-P algorithm.

Draw relational diagram of $\ln C(r) \sim \ln r$ as shown in Fig.5. The scale-free area in Fig.5 is the correlation dimension. The change curve of correlation dimension with embedded dimension m is shown in Fig.6. It can be seen that with the increase of embedded dimension, when $m = 9$ the correlation dimension reaches saturated. It illustrates that the time series have chaotic nature, and $m = 9$.

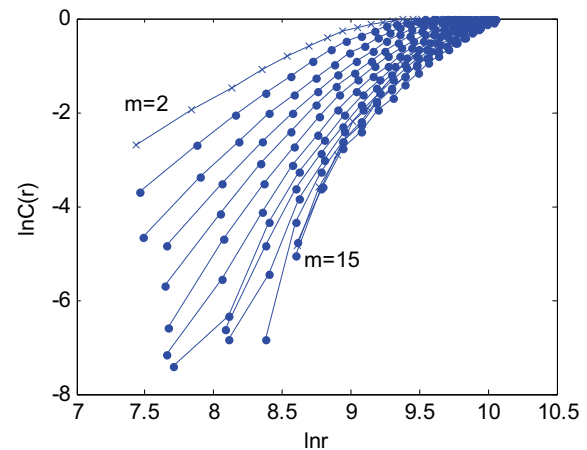


Fig.5. Relational diagram of $\ln C(r) \sim \ln r$

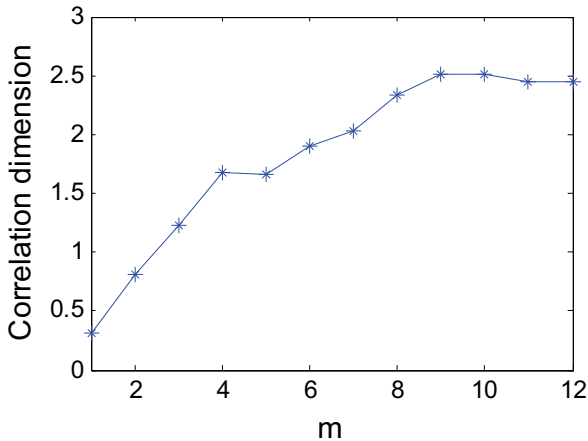


Fig.6. Relation between correlation dimension and embedded dimension

The electric power consumption data is reconstructed in terms of $\tau = 13$, $m = 9$. The reconstructed phase space vector is

$$\begin{cases} L_1 = (l_1, l_{14}, \dots, l_{105}) \\ L_2 = (l_2, l_{15}, \dots, l_{106}) \\ \dots \\ L_{101} = (l_{101}, l_{114}, \dots, l_{205}) \end{cases} \quad (11)$$

101 groups of experimental samples with the length of 9 are gotten. The first 8 dimensions of each group of samples are as the network inputs, the 9th dimension as the forecasting output, and according to the proportion of 5:2.5:2.5, the samples are randomly divided into training set, validation set and test set for CPNN training.

6.1.3. Network modeling and result analysis

To prove the superiority of CPNN model and its learning algorithm (CPNN-CSO) in time series forecasting, adaptive neuro fuzzy inference system (ANFIS), conventional process neural network (PNN) and cloud process neural network (CPNN) based on gradient descent combined with basis function expansion are used for comparison. Four models employ the sample set after phase space reconstruction. According to actual problem, ANFIS's input node number is 8 and its output node number is 1. The input/output node number of other three networks is all 1. For PNN and CPNN, cubic Spline function is used to fit discrete time series, Legendre basis function for expansion and gradient descent algorithm for learning. In CPNN-CSO, Euclidean distance is used to measure $\|X_i - Ex_i^j\|$. In CSO, N=100, SMP=5, MR=0.02, and the search range of m_i is [1,8]. In ANFIS, the

membership function of input variables uses bell-shaped function, and each input variable includes 3 fuzzy subsets. It adopts gradient descent algorithm for training. The maximum iteration number of four networks is all 1000. The MSE of normalized samples is 0.001. The network performance evaluation index Pr is defined as the product of final iteration number and convergent MSE. Obviously, the smaller the MSE and the less the iteration number, the better the results are. Through multiple experiments, the performance comparison of four methods on training set, validation set and test set is shown in Fig.7.

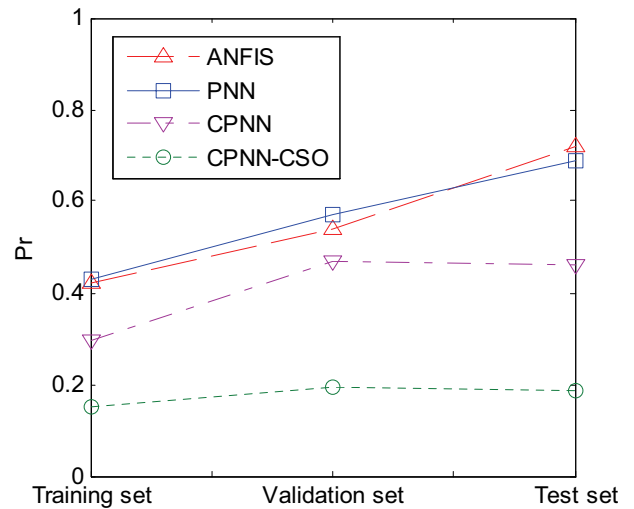


Fig.7. Performance comparison with four methods

It is observed that CPNN-CSO is superior to CPNN, PNN and ANFIS both in model approximation ability and generalization performance. The results could be illustrated as follows: firstly, compared to PNN, CPNN-CSO and CPNN convert electric power consumption quantitative data into multiple qualitative sub-cloud concepts, and find out the association rule between input and output variables through mining inherent law among multiple sub-clouds which well reflects the randomness and fuzziness of time series, therefore the prediction ability is better than PNN. Secondly, compared to ANFIS, CPNN-CSO and CPNN consider not only the fuzziness of time series but also the randomness, and CPNN-CSO and CPNN directly use time series as their inputs, which takes more accumulative time effect into account, so they have better forecasting result. Thirdly, for ANFIS and PNN, ANFIS considers the fuzziness of time series, but PNN includes more accumulative time effect, therefore the forecasting result of ANFIS is not superior to PNN. Lastly, compared to CPNN, it helps to decrease the error produced during basis expansion and function fitting by using discrete time series as direct CPNN-CSO inputs. In addition, CSO algorithm is used in

CPNN-CSO to optimize structure and learning parameters of the network simultaneously, which gives full play to local searching ability and global optimization ability of CSO and avoids unstable phenomenon due to premature problem of gradient descent, therefore the effect is better. In CPNN-CSO, the final sub-cloud number is $m_i = 3$. Result comparison of four methods is shown in Fig. 8.

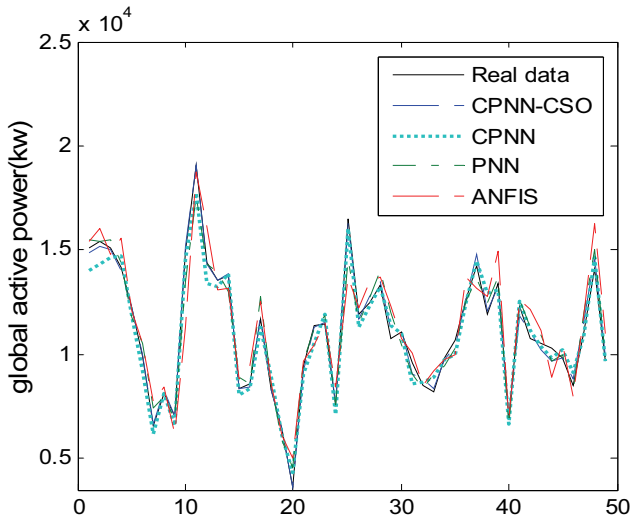


Fig.8. Prediction result comparison with four methods

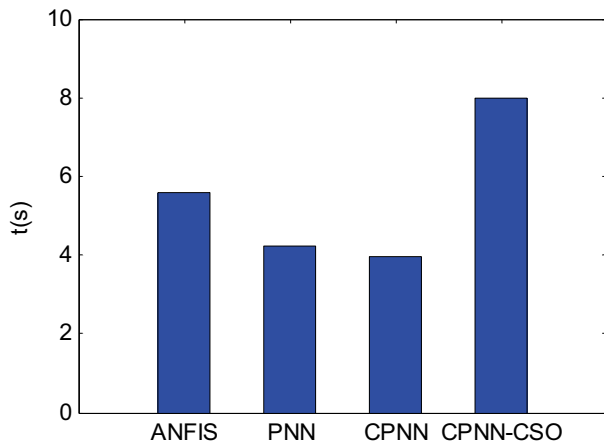


Fig.9. Average elapsed time comparison with four algorithms

Comparison is done with four methods on average elapsed time to carry out once training and predicting, as shown in Fig.9. The elapsed time of PNN and CPNN is similar, ANFIS consumes more time, and CPNN-CSO consumes the most time. This is because the network structure and learning algorithm of PNN and CPNN are almost the same just with different number of learning parameters. In ANFIS, it has several inputs and each input includes several fuzzy subsets, so the inference rules become more and the amount of ANFIS calculation is larger, ANFIS consumes more time than

CPNN and PNN. For CPNN-CSO, since all cats' fitness values need to be computed at every iteration, its elapsed time is the longest. However, the whole elapsed time is still acceptable for that the communication among cats helps to reduce the iteration number and find out the optimal value more effectively and faster.

In addition, to test the effect of training sample quality before and after phase space reconstruction for CPNN performance, electric power consumption time series data is now directly as the CPNN-CSO input. By trial-and-error method, the forecast scheme is to predict the 5th week data using the first 4 week data, and 101 groups of samples are randomly chosen for experiment from total 201 groups of samples. The other parameters are the same as that of the network with experimental data after phase space reconstruction. Experimental results are shown in Table 1.

Table 1. Performance comparison before and after phase space reconstruction

Data set	Iteration number	Elapsed time /s	MSE	Pr
Before reconstruction	495	7.86	0.001	0.495
After reconstruction	185	7.99	0.001	0.185

It is observed that in the case of the same network structure and parameter setting, the sample set after phase space reconstruction is more representative. It considers time series correlation in space and time, overcomes the influence of "false nearest neighbor" effectively, and improves the network generalization ability and forecasting accuracy.

6.2. ASP flooding oil recovery index forecasting in oilfield development

In oilfield development and production, many issues rely on change characteristics of oilfield development index, such as the design and adjustment of oilfield development planning and scheme, the evaluation and management of oilfield exploitation status, etc. Therefore, it is extremely important to dynamically predict oilfield development index in the work, and the study on all kinds of index forecasting methods has become a key topic in petroleum development area. ASP flooding is a tertiary oil recovery technique to enhance oil recovery rising in the mid-and-late 1980s in the world. It synthesizes alkaline flooding, surfactant flooding and polymer flooding, and can improve crude oil recovery ratio than water flooding by more than 20%. Water flooding development time is earlier and water flooding development has many reservoir

engineering index forecasting methods in oilfield, but these methods are not able to completely fit ASP flooding index forecasting. Here, we use a time series prediction method based on cloud process neural network for short-term prediction on ASP flooding oil recovery index.

6.2.1. Experimental data selection

This experiment will predict the whole block oil production index in an ASP flooding field test area. The experiment collects 126 monthly oil production data of the whole block from September 2004 to February 2015. In order to eliminate the influence to monthly oil production that there are different days in different month, calculate average daily oil production every month for index prediction. Part of monthly oil production and average daily oil production are shown in Table 2.

Table 2. Part of oil production data

No.	Monthly oil production (t)	Average daily oil production (t)
1	9846	317.61
2	8459	302.11
3	8923	287.84
4	8336	277.87
5	8831	284.87
6	8433	281.10
7	7961	256.81
8	8335	268.87
9	8019	267.30
10	8035	259.19

6.2.2. Sample set construction

The oil production time series does not have the chaos characteristics, therefore, we apply directly the first d average daily oil productions to predict the $d+1$ average daily oil production. In order to set the value of d reasonably, we compute the correlation coefficient between the $k—k+d-1$ oil productions and the $k+d$ oil production where $k=1,2,\dots,126-d$. The result is shown in Fig.10.

It is observed that when $d \geq 6$ the correlation coefficient is close to a maximum. Therefore, the forecasting scheme is to predict the 7th average daily oil production with the first 6 average daily oil productions. That is, 6 consecutive discrete data in oil production time series $\{o_i\}_{i=1}^{126}$ is chosen as the network process

input, the 7th adjacent data as the expected output, and then 120 groups of input-output samples are gotten.

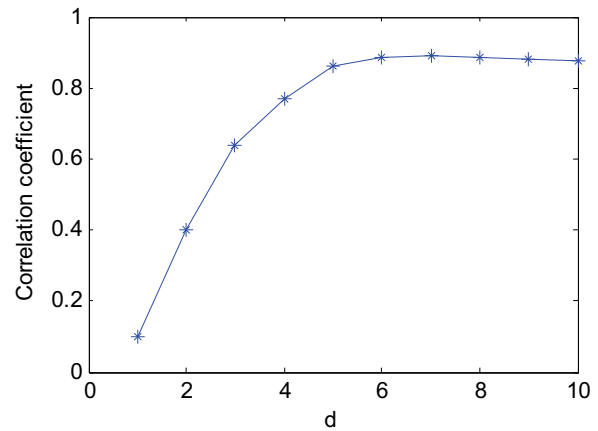


Fig.10. Correlation coefficient of oil production time series

6.2.3. Network modeling and result analysis

According to sample construction characteristics, both the time-varying input layer and the output layer of the CPNN have 1 node. Discrete Fréchet distance²⁸ is chosen to measure the matching degree between input vector and cloud expectation. The size of cat swarm is $N=80$, $SMP=5$, $MR=0.03$, and the search range of m_i is a integer in $[1,6]$. The maximum iteration number is 1000. The MSE of normalized samples is 0.002. To test the influence of different sample size to the CPNN performance, randomly pick out 40, 60, 80, 100 and 120 groups of samples to compose sample sets and then each sample set are randomly divided into training set, validation set and test set according to the proportion of 6:2:2. Several independent simulations are done for each sample set and the average simulation results are shown in Table 3.

Table 3. Simulation experiment results

No.	1	2	3	4	5
Sample size	40	60	80	100	120
Iteration number	116	145	166	197	204
Average time-consuming(s)	7.15	6.89	7.74	7.88	7.23
Convergence error	0.0009	0.0014	0.0008	0.0018	0.0016
Average relative error (%)	2.09	2.18	1.93	1.43	1.5
Maximum relative error (%)	4.15	3.87	3.34	2.92	2.79
m_i final value	3	4	3	3	3

It is observed that when the sample size is small, although the network has a faster convergence speed, the prediction error is bigger. With the increase of sample size, the network prediction effect becomes more stable. When the sample size reaches a certain number (100 in this experiment), the prediction accuracy achieves the maximum. The comparison between real oil production and prediction oil production of experiment No. 4 is shown in Fig.11.

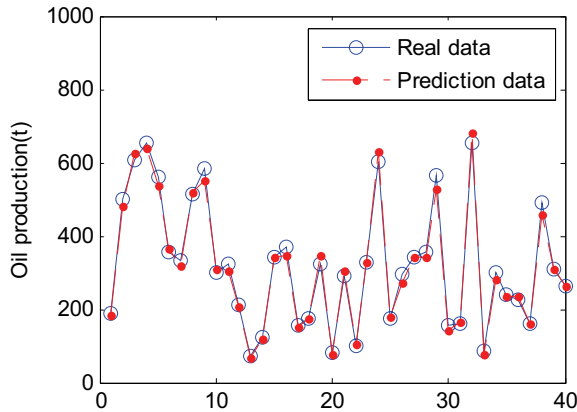


Fig.11. Comparison between real oil production and prediction oil production

To better master long-term change trend of oil production, CPNN could also be used for medium and long-term oil production rolling forecasts, which may combine organically with short-term forecasting method, so as to take corresponding control measures timely according to different period of production plan and to provide decision support for sustainable long-term development of oilfield.

7. Conclusion

A cloud process neural network model based on cloud theory is presented for time series forecasting. It combines cloud reasoning system with process neural network organically. On the one hand, self-learning and self-adaptive characteristics of cloud reasoning system is realized, at the same time the dynamic processing ability of process neural network for time series with continuous, random and fuzzy information is also enhanced. For the learning of the model, a learning algorithm based on CSO algorithm is put forward in this paper. The method can identify the network structure and learning parameters simultaneously, and improve approximation and generalization ability of CPNN effectively. Finally, individual household electric power consumption time series prediction and ASP flooding oil recovery index forecasting are used for simulation. The experimental results show that time series

forecasting method based on CPNN has faster convergence speed and better generalization ability. So it provides a new solution for classification and prediction problems on time series signal.

Acknowledgements

The work was supported by the National Natural Science Foundation of China (Grant No. 61170132) and the Natural Science Foundation of Heilongjiang Province, China (Grant No. F2015021). We would like to thank the reviewers. Thanks a lot for editor's helpful comment that improved the presentation of the paper.

References

1. Hornik K. Stinchcombe M. and White H. Multilayer feedforward networks are universal approximators, *Neural Networks*, **2**(5) (1989)359-366.
2. Funahashi K. On the approximate realization of continuous mappings by neural networks, *Neural Networks*, **2**(3) (1989)183-192.
3. Frank R. J., Davey N. and Hunt S. P. Time series prediction and neural networks, *J. Intell. Robot. Syst.* **31**(3) (2001) 91-103.
4. Zhang G. P. and Qi M. Neural network forecasting for seasonal and trend time series, *Eur. J. Oper. Res.* **160**(2) (2005)501-514.
5. X. G. He and J. Z. Liang, Process neural networks, in *Proc. 16th IFIP WCC200 Int. Conf. Intelligent Information Processing* (Publishing Houses of Electronics Industry, Beijing, 2000), pp. 143-146.
6. G. Ding and S. S. Zhong, Time series prediction by parallel feedforward process neural network with time-varied input and output functions, *Neural Netw. World*, **15**(2) (2005) 137-147.
7. S.S. Zhong, Y. Li and G. Ding, Continuous wavelet process neural network and its application. *Neural Netw. World*, **17**(5) (2007) 483-495.
8. G. Ding and L. Lin, Elman-style process neural network with application to aircraft engine health condition monitoring, *Lect. Notes Comput. Sci.* **6675** LNCS (1) (2011) 484-494.
9. Y. W. Pang and S. H. Xu, A train algorithm of multi-Aaggregation process neural networks based on chaos genetic optimization, in *Proc. Natl. Conf. Inf. Technol. Comput. Sci.* (Atlantis Press, Lanzhou, 2012), pp. 374-377.
10. P. C. Li, H. Y. Wang, Q. Dai and H. Xiao, Quantum process neural networks model algorithm and applications, *Acta Phys Sin-CH ED*, **61**(16) (2012)160303-1-9.
11. X. Li and C. T. Cheng, Discrete process neural networks and its application in the predication of sunspot number series, in *2009 Chinese Control and Decision Conference* (IEEE, Guilin, 2009), pp.4339-4342.
12. S. H. XU, Y. Q. CAI and S. M. XIE. Learning algorithm of procedure neural network based on orthogonal function base, *Computer Engineering and Design*, **27**(23) (2006)4413-4415,4418.
13. S. S. Zhong, .G. Ding and D. Z. Su, Parallel feedforward process neural network with time-varying input and output functions, in *Advances in Neural Networks- ISNN2005, Lecture Notes in Computer Science 3496*(Springer-Verlag,Chongqing, 2005),pp. 473-478.
14. M. J. Cao and F. H. Shang, Training of process neural networks based on quantum genetic algorithm with double chains, *Computer Measurement & Control*, **18**(2) (2010)407-410.
15. B. Wang, S. H. Xu, Y. H. Meng and X. L. Wang, A process neural network based on improved particle swarm optimization and its application in PID control, *International Journal On Advances in Information Sciences and Service Sciences*, **5**(7) (2013)701-709.
16. X. G. He and J. Z. Liang, Some theoretical issues on procedure neural networks, *Engineering Science*, **2**(12) (2000)40-44.
17. S. H. Xu and X. G. He, Some theoretical issues on continuous process neural networks, *Acta Electronica Sinica*, **34**(10) (2006) 1838-1842.
18. G. Ding and S. S. Zhong, Aircraft engine lubricating oil monitoring by process neural network, *Neural Netw. World*, **16**(1) (2006)15-24.
19. S. S. Zhong and Y. Li. Time series prediction by locally recurrent globally feedforward process neural network with time-varied input function, in *Proc. Int. Conf. Sensing, Computing and Automation*(Watam Press, Chongqing, 2006), pp. 914-918.
20. D. Y. Li, H. J. Meng and X. M. Shi, Membership clouds and membership cloud generators, *Journal of Computer Research and Development*, **32**(6) (1995)16-18.
21. D. Y. Li, C. Y. Liu and W. Y. Gan, A new cognitive model: cloud model, *Int. J. Intell. Syst.* **24**(3) (2009)357-375.
22. Y. Deng, S. H. Liu, W. T. Zhang, L. Wang and J. H. Wang, General multidimensional cloud model and its application on spatial clustering in Zhanjiang, Guangdong, *J. Geogr. Sci.* **20**(5) (2010)787-798.
23. F. Fang and A. M. Yu, The economic evaluation of the wind power projects based on the cloud model, *Advanced Technology in Teaching*, **163**(2013)443-448.
24. L. L. Tang, Y. Tian and J. S. Pan, Applications of cloud model in digital watermarking, *Intelligent Data Analysis and Applications*, **370**(2015)371-379.
25. S. C. Chu, P. W. Tsai and J. S. Pan, Cat swarm optimization, in *Proc. 9th Pacific Rim Int. Conf. Artificial Intelligence LNAI 4099*, eds. Q. Yang and G. Webb (Springer-Verlag, Berlin, 2006), pp. 854-858.
26. Ilhan Aydin, Mehmet Karakose and Erhan Akin, An approach for automated fault diagnosis based on a fuzzy decision tree and boundary analysis of a reconstructed phase space, *ISA T.* **53**(2) (2014) 220-229.
27. Grassberger P. and Procaccia I. Characterization of strange attractors, *Phy. Rev. Lett.* **50**(5) (1983)346-349.
28. Alt H. and Godau M. Computing the Fréchet distance between two polygonal curves, *International Journal of Computational Geometry & Applications*, **5**(1995) 75-91.