Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090345

Novel Learning Algorithm based on BFE and ABC for Process Neural Network and its Application

Yaoming Zhou[1], *Xuzhi Chen*[1], *Wei He*[1,2] *and Zhijun Meng*[1*]

[1] School of Aeronautic Science and Engineering, Beihang University, 100191 Beijing, China
[2] Aviation Theory Department, Aviation University of Airforce, 130022 Changchun, China

Received: 15 Aug. 2014, Revised: 16 Nov. 2014, Accepted: 17 Nov. 2014
Published online: 1 May 2015

Abstract: In order to improve the generalization capability of process neural network (PNN), a novel learning algorithm is proposed based on basis function expansion (BFE) algorithm and artificial bee colony (ABC) algorithm, named BFE-ABC algorithm. First, the input functions and weight functions are simplified through BFE algorithm. The parameter space is transformed from function space to real number space in this way. Then, the PNN is designed to parametric representation through introducing two Boolean variables and one multidimensional parameter. At last, the multidimensional parameter composed of hidden neurons, expansion items and connection weights is optimized in real number space by ABC algorithm. BFE-ABC algorithm overcomes the premature problem and realizes the global optimization of the structure, connection weights and function expansion form at the same time. It is validated through the prediction experiment of Mackey-Glass chaotic time series. The test results in cylinder head temperature prediction prove the superiority of BFE-ABC algorithm over traditional learning algorithm and the applicability to time-dependent parameter prediction.

Keywords: process neural network, learning algorithm, basis function expansion, artificial bee colony, time-dependent parameter prediction

1 Introduction

Process neural network (PNN) is proposed by He for processing the spatio-temporal problem with multidimensional information in 2000 [1,2]. It is an extension of classic artificial neural network (ANN), in which the inputs, outputs and weights are time-dependent. Inputs such as time series are usually sensitive to the time parameter, so the weights should also be related to time in order to accumulate the effects of inputs more precisely. Theoretically, the approximation capabilities of PNN are better than classic ANN when solving time-dependent function problems. PNN breaks traditional input instantaneous synchronization restriction and simulates the physiology of the biological neurons better.

The learning algorithm of PNN is distinct from ANN as the inputs, outputs and weights of PNN are time-dependent. It not only includes space aggregation operation but also includes time aggregation operation. The main learning algorithms are numerical integration algorithm [3] and basis function expansion (BFE) algorithm combined with back-propagation (BP) algorithm [4] or other improved algorithms, such as resilient BP (RBP) algorithm [5]. The numerical integration algorithm utilizes numerical integration to process discrete inputs directly. The time aggregation operation in PNN is realized in this way [3]. The BFE algorithm introduces a group of orthogonal basis functions and expands the input functions and weight functions in limited series of this group, whose orthogonality can be used to simplify the complexity of process neuron in time aggregation operation. The application shows that BFE algorithm has not only simplified the operation of PNN but also increased the stability and convergence in network training as it retains the principal components of data [4]. However, the topology structure of PNN cannot be optimized through these learning algorithms and the connection weights obtained through these learning algorithms maybe local optimal solutions as the limitation of BP algorithm or other improved algorithms. The generalization capability of the model built in this way is restricted. In order to overcome these disadvantages, a parametric representation method of PNN should be introduced to

* Corresponding author e-mail: mengzhijun@buaa.edu.cn

realize optimization of topology structure. Besides, a global optimization algorithm should be introduced to realize the global optimization of PNN.

Many global optimization algorithms have been applied to train neural network such as genetic algorithm [6,7], particle swarm optimization algorithm [8,9,10,11], differential evolution algorithm [12,13], artificial bee colony (ABC) algorithm [14,15,16] and so on. As a new intelligent computing algorithm, ABC algorithm realizes global search and local search in every cycle and outperforms other intelligent computing algorithms. Karaboga applies ABC algorithm to optimize connection weights of ANN. He proves that ABC Algorithm outperforms BP algorithm and genetic algorithm in [14]. He also proves that ABC Algorithm has better performance than particle swarm optimization algorithm and differential evolution algorithm algorithm in [15]. Garro applies ABC algorithm to design an ANN automatically, including not only the connection weights but also the structure and the transfer functions [16]. The ANN obtained has fewer connections than ANN obtained in [10,12]. It is optimal in the sense that the number of connections is minimal without losing efficiency.

A novel learning algorithm named BFE-ABC algorithm is proposed in this paper. This algorithm utilizes BFE algorithm to simplify the complexity of process neuron in time aggregation operation. The parameter space is transformed from function space to real number space by this way. It also utilizes ABC algorithm to realize the global optimization of PNN as the superiority mentioned above. A parametric representation method of PNN is proposed in this algorithm by introducing two Boolean variables and one multidimensional parameter. The PNN can be designed automatically through this algorithm, including the connection weights, the topology structure and the function expansion form (the function expansion form is initialized through BFE algorithm). This algorithm will be validated in the prediction experiment of Mackey-Glass chaotic time series. The applicability to time-dependent parameter prediction will also be discussed.

2 Training PNN

2.1 PNN

PNN is a new type of artificial neuron system based on certain topological structure, which is made of some process neurons and traditional neurons (no time-dependent) [1,2].

The structure of process neuron is composed of three parts: weighting, aggregation and activation [1,2]. The process neuron is distinct from traditional neuron, whose inputs, outputs and weights can be time-varying, and aggregation operations are composed of multi-input aggregation in space domain and cumulative aggregation in time domain. The structure of a single process neuron is shown in Fig. 1. Here $x_i(t)$ $(i = 1,2,\ldots,n)$ represents the input function of the process neuron. $w_i(t)$ is the corresponding weigh function. $K(\cdot)$ is the time cumulative aggregation function. $f(\cdot)$ is the activation transfer function which maybe a linear function, a sigmoid function or a Gauss-type function and so on.

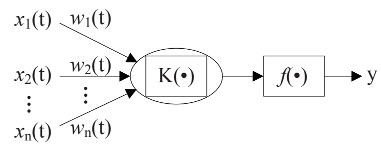


Fig. 1: Process neuron

The topology structure of cascade-forward PNN [5] researched in this paper is shown in Fig. 2. Here the input layer and the hidden layer are composed of process neurons. The output layer is composed of traditional neurons. The relationship between input and output is shown as below:

$$y = g\Big(\sum_{i=1}^{n} v_i f\Big(\int_0^T w_i(t)x(t)dt+\theta_i\Big)+\int_0^T u(t)x(t)dt+\theta\Big), \quad (1)$$

where $x(t)$ is system input function; $w_i(t)$ is the connection weight function between the neuron in input layer and the i_{th} neuron in hidden layer; θ_i is the output threshold of the i_{th} neuron in hidden layer; $[0,T]$ is sampling period; f is activation transfer function of hidden layer, and usually sigmoid function is selected; v_i is the connection weight between the i_{th} neuron in hidden layer and the neuron in output layer; $u(t)$ is the connection weight between the neuron in input layer and the neuron in output layer; θ is the output threshold of the neuron in output layer; g is the activation transfer function of output layer, which is usually described as the linear one; n is the number of hidden neurons; y is the system output.

2.2 Traditional learning algorithm

As mentioned above, the main learning algorithms of PNN are numerical integration algorithm and BFE algorithm combined with BP algorithm (named BFE-BP algorithm) or other improved algorithms. A brief description of BFE-BP algorithm is given here.

In BFE-BP algorithm, a group of proper orthogonal basis functions is brought into input space first. Input functions are expanded in limited series of this group by given precision. Meanwhile, weight functions are expressed as the expansion forms in the same group of

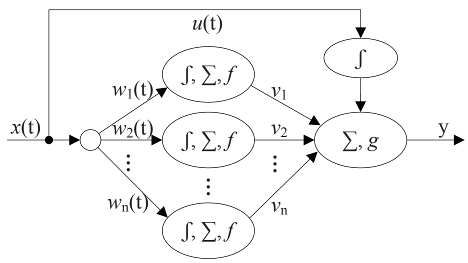


Fig. 2: Cascade-forward process neural network

basis functions whose orthogonality can be used to simplify the complexity of process neuron in time aggregation operation. The time-dependent parameters are eliminated in this way. After this operation, all the remaining parameters in PNN can be optimized by BP algorithm, i.e., there are the same complexities between training PNN and ANN according to this algorithm.

The steps of BFE-BP algorithm are shown as below:

Step 1: Initialize the network topology structure, expand input functions and weight functions based on selected orthogonal basis functions to obtain the corresponding expansion coefficients.

Step 2: Adjust weight expansion coefficients based on BP algorithm. (The initial weight expansion coefficients are randomly created.)

Step 3: Evaluate the performance of PNN.

Step 4: If the desired result is obtained, then stop; otherwise goto step 2.

The difficulties in this learning algorithm are how to choose the appropriate basis functions and how many expansion items should be reserved. In order to improve the training speed, RBP replaces BP in above learning algorithm to form BFE-RBP algorithm which has been discussed in [5]. However, the optimization of topology structure and function expansion form still remains unsettled.

3 Artificial bee colony algorithm

ABC algorithm was proposed by Karaboga in 2005 [17] for solving numerical optimization problems. This algorithm is based on the model proposed by Tereshko and Loengarov [18]. It simulates the intelligent foraging behaviour of honey bee swarm. The possible solutions x_i (the population) are represented by the position of the food sources. In order to find the best solution, three classes of bees are used: employed bees, onlooker bees and scout bees. They have different tasks in the colony.

Employed bees: Each bee searches for new neighboring food source near its hive using Eq. (2). It compares the food source against the old one, then saves in its memory the best food source. Finally, it returns to the dancing area in the hive, where the onlooker bees are.

$$v_i^j = x_i^j + \phi_i^j (x_i^j - x_k^j). \qquad (2)$$

In Eq. (2) $k \in 1,2,\ldots,SN$ and $j \in 1,2,\ldots,D$ are randomly chosen indexes and $k \neq i$. ϕ_i^j is a random number between $[-1,1]$. SN denotes the size of population. D is the dimensionality of solution space, representing the number of optimization parameters.

Onlooker bees: This kind of bees watches the dancing of the employed bee so as to know where the food source can be found. The onlooker bee chooses a food source depending on the probability value p_i associated with that food source, calculated by the following expression:

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}, \qquad (3)$$

where fit_i is the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i .

Scout bees: This kind of bees helps the colony to randomly create new solutions when a food source cannot be improved further through a predetermined number of cycles. The value of predetermined number of cycles is an important control parameter in the ABC algorithm, which is called "*limit*" for abandonment. Assume that the abandoned source is x_i, and then the operation can be defined as below:

$$x_i^j = x_{min}^j + rand(0,1)(x_{max}^j - x_{min}^j), \qquad (4)$$

where x_{max}^j and x_{min}^j are the maximum value and minimum value of the solution in dimension $j \in 1,2,\ldots,D$ respectively.

Detailed pseudocode of the ABC algorithm is given as below:

1) Randomly initialize the population of solutions x_i , $i = 1 \cdots SN$

2) Evaluate the population

3) Cycle=1

4) Repeat

5) Produce new solutions for the employed bees by Eq. (2) and evaluate them

6) Apply the greedy selection process

7) Calculate the probability values p_i for the solutions x_i by Eq. (3)

8) Produce new solutions v_i for the onlooker bees from the solutions x_i selected depending on p_i and evaluate them

9) Apply the greedy selection process

10) Determine the abandoned solution for the scout bees, if exists, and replace it with a new randomly produced solution x_i by Eq. (4)

11) Memorize the best solution achieved so far

12) Cycle= Cycle +1

13) Until Cycle=MCN (Maximum Cycle Number)

4 Novel learning algorithm

4.1 Algorithm description

A novel learning algorithm named BFE-ABC algorithm is proposed here, which combines the advantages of BFE algorithm and ABC algorithm to realize parameter space transformation from function space to real number space and parameter global optimization. Besides, a parametric representation method is introduced here to realize optimization of PNN. The BFE-ABC algorithm includes three key parts as below:

1) Parameter space transformation

The input functions and weight functions are simplified through BFE algorithm. The parameter space is transformed from function space to real number space in this way.

N groups of training samples are provided here, expressed as $\{p_l(t), q_l\}, (l = 1, 2, \dots, N)$, where $p_l(t)$ is the input and q_l is the expected output of the l_{th} training sample. The simplification of Eq. (1) based on BFE algorithm is shown as below:

$$y_l = \sum_{i=1}^{n} v_i f(\sum_{k=1}^{K} w_{ik} a_{kl} + \theta_i) + \sum_{k=1}^{K} u_k a_{kl} + \theta, \quad (5)$$

where y_l is the output of the PNN corresponding to the l_{th} training sample; a_{kl} is the expansion coefficient of input function $p_l(t)$; w_{ik} and u_k are expansion coefficients of corresponding connection weight functions; K is the maximum number of expansion items depending on the corresponding basis functions. The other parameters have the same meanings as in Eq. (1).

2) Parametric representation

Two Boolean variables and one multidimensional parameter are introduced to realize parametric representation of PNN.

A Boolean variable is defined here to represent the hidden layer:

$$B = (b_1, b_2, \dots, b_{n+1}), \quad (6)$$

where $b_i(i = 1, 2, \dots, n)$ is 1 or 0, which means the corresponding hidden neuron exists or not; b_{n+1} is 1 or 0, which means the connection between input and output layer exists or not. An operation is defined as in Eq. (7).

$$\rho = \sum_{i=1}^{n+1} b_i 2^{-i} \quad (7)$$

The conclusion $0 < \rho < 1$ is easily drawn. Therefore, a corresponding $(n+1)$-dimensional Boolean variable B must exist when $\rho \in (0, 1)$ is chosen randomly, i.e., the structure of PNN can be constructed by B drawn from ρ.

Another Boolean variable C is defined here to represent the expansion form of input function:

$$C = (c_1, c_2, \dots, c_K), \quad (8)$$

where $c_k(k = 1, 2, \cdots, K)$ is 1 or 0, which means the expansion item corresponding to one basis function exists or not. An operation is defined as in Eq. (9).

$$\eta = \sum_{k=1}^{K} c_k 2^{-k} \quad (9)$$

The conclusion $0 < \rho < 1$ is easily drawn. Therefore, a corresponding K-dimensional Boolean variable C must exist when $\rho \in (0, 1)$ is chosen randomly, i.e., the expansion form of input function can be determined through C drawn from η.

The relationship between input and output can be rewritten as below:

$$y_l = \sum_{i=1}^{n} b_i v_i f(\sum_{k=1}^{K} c_k w_{ik} a_{kl} + \theta_i) + b_{n+1} \sum_{k=1}^{K} c_k u_k a_{kl} + \theta. \quad (10)$$

A multidimensional parameter x is defined here to realize the parametric representation of PNN. Every solution of x represents a PNN:

$$x = \rho, \eta, w_{ik}, \theta_i, v_i, u_k, \theta, i = 1, 2, \cdots, n; k = 1, 2, \cdots, K. \quad (11)$$

Different dimensions in the multidimensional parameter x take values from different ranges. For the first two dimensions, ρ and η , the range is between $(0, 1)$. For the other dimensions, the range is $[-1, 1]$.

3) Global optimization

The multidimensional parameter x is optimized by ABC algorithm. It represents a food source position corresponding to the possible solution.

Each food source position in the solution space represents a PNN. ABC algorithm will search for a PNN that has appropriate structure and parameters to optimize the objective function. The objective function is defined as below:

$$J = lnE + \frac{\lambda_1}{N} \sum_{i=1}^{n+1} b_i + \frac{\lambda_2}{N} \sum_{k=1}^{K} c_k. \quad (12)$$

The first item in Eq. (12) represents the goodness of fit between the model output and the sample. E is mean squared error (MSE), given in Eq. (13). The second item is used to limit the complexity of network. The third item is used to limit the complexity of BFE, which limits the complexity of network indirectly. λ_1 and λ_2 are adjustable coefficients, which represent the proportion relative to the first item.

$$E = \frac{1}{N} \sum_{l=1}^{N} (q_l - y_l)^2 \quad (13)$$

According to the above description, the steps of BFE-ABC algorithm can be summarized as below:

Step 1: Expand input functions and weight functions through BFE algorithm.

Step 2: Represent PNN in a multidimensional parameter according to Eq. (11)

Step 3: Set up the objective function, i.e. Eq. (12), as the fitness function of ABC algorithm and set parameters (search space borderlines, λ_1, λ_2, SN, $limit$, MCN).

Step 4: Run ABC algorithm (the new solutions are decoded according to Eqs. (7), (9), (10) and (11) in every cycle to evaluate them).

Step 5: After stops, output the optimal solution and decode it to PNN.

4.2 Algorithm validation

In order to validate the algorithm proposed above, Mackey-Glass chaotic time series prediction is used here. The equation of Mackey-Glass [19] is shown in Eq. (14).

$$\frac{dx}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t). \tag{14}$$

Initialize the parameters in Eq. (14): $a = 2$, $b = 0.1$, $\tau = 17$, $x(0) = 1.2$. 406 data are obtained from Eq. (14), i.e. $x(t), t = 0,1,2,\cdots,405$. Six continuous data are fitted to form a time-dependent polynomial function as the input of PNN, and the seventh datum is set to output. 400 samples are obtained through this way. The first 200 samples are taken as training set. The last 200 samples are taken as test set.

The input function and connection weight functions are expanded based on shifted Legendre orthogonal basis functions. Therefore, K is equal to 6 here. The maximum number of hidden neurons is taken as 10, i.e., $n = 10$. Initialize the adjustable coefficients in Eq. (12), $\lambda_1 = 1$, $\lambda_2 = 1$. Initialize other parameters, $SN = 10$, $limit = 100$, $MCN = 2500$.

Besides MSE, mean relative error (MRE) is used to evaluate a PNN after optimization, defined as below:

$$MRE = \frac{1}{N}\sum_{l=1}^{N}\frac{|q_l - y_l|}{q_l}. \tag{15}$$

Run the algorithm proposed above to obtain an optimal PNN. The parameters in Eq. (12) can be obtained after training. The MSE is $1.7171 * 10^{-4}$. $B = (1,1,0,1,0,0,0,1,1,1,1)$ and $C = (1,1,1,1,0,1)$. Therefore, the topology structure of newly constructed PNN is 1-6-1. The connection between input and output layer exists. The basis functions used here are 0-3 order shifted Legendre polynomials and 5 order shifted Legendre polynomial.

The test set is employed to validate the generalization capability of newly constructed PNN model. The test results are shown in Fig. (3). The MSE is $1.2215 * 10^{-4}$. The MRE is 0.0103 . The results show that the algorithm proposed above could solve the prediction problem of Mackey-Glass chaotic time series effectively. The algorithm is validated in this way.

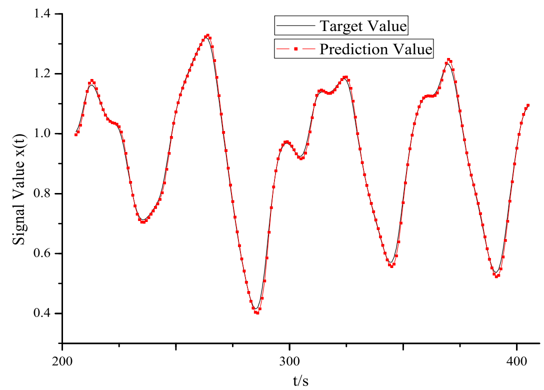


Fig. 3: Prediction of Mackey-Glass chaotic time series

5 The application in time-dependent parameter prediction

PNN breaks traditional input instantaneous synchronization restriction and can be applied to time-dependent parameter prediction as its time-dependent input characteristics [20,21]. The time-dependent characteristics of engine are so complicated that they can't be described by an existing analytical model. PNN is applied here to construct engine model to realize parameter prediction.

This section takes the cylinder head temperature prediction of engine as an example and compares the prediction performance of PNN models optimized by BFE-ABC and BFE-RBP receptively. The cylinder head temperature data is from several flight experiments of an unmanned helicopter, whose sampling frequency is 10 times per second. The relative error of the collected data is 0.1%. In order to get the change trend of engine parameter, the data is resampled every 10 seconds. The offline prediction and online prediction are carried out respectively.

5.1 Offline prediction

This part takes the data from one flight experiment. The data is divided into two sets: a training set and a testing set. 200 data are obtained after resampled. Part of the data are shown in Table 1. Six continuous data are fitted to form a time-dependent polynomial function as the input of PNN, and the seventh datum is set to output. 194 samples are obtained through this way. The first 100 samples are taken as training set to get the prediction model. The last 94 samples are taken as test set to validate the prediction precision.

PNN models are constructed based on BFE-ABC algorithm and BFE-RBP algorithm to realize offline

Table 1: The Resampled Data (part)

Number	Temperature (^{o}F)	Number	Temperature (^{o}F)
1	202.0	11	221.2
2	202.1	12	223.2
3	204.1	13	225.3
4	206.5	14	227.3
5	208.7	15	229.3
6	210.8	16	231.1
7	212.9	17	232.9
8	215.1	18	234.6
9	217.3	19	236.4
10	219.3	...	...

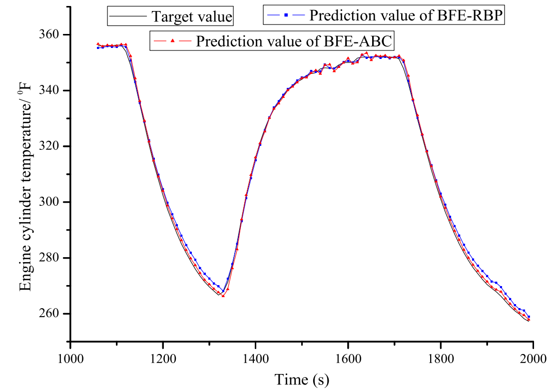


Fig. 4: Offline prediction

prediction. The parameters in BFE-RBP are set the same as in [5]. The parameters in BFE-ABC are set as below. The input function and connection weight functions are expanded based on shifted Legendre orthogonal basis functions. Therefore, K is equal to 6 here. The maximum number of hidden neurons is taken as 10, i.e., $n = 10$. Initialize the adjustable coefficients in Eq.(15), $\lambda_1 = 0.5$, $\lambda_2 = 0.2$. Initialize other parameters, $SN = 15$, $limit = 100$, $MCN = 3000$.

The training and prediction results are shown in Table 2. Figure 4 is the prediction curves. It is obvious that the PNN model optimized by BFE-ABC algorithm has simpler structure and basis function expansion form than BFE-RBP. The prediction accuracy of BFE-ABC algorithm is higher than BFE-RBP algorithm. Because all the solutions should be decoded in every cycle to evaluate the fitness value in BFE-ABC, the time consumed by BFE-ABC is longer than BFE-RBP. However, the time consumed is acceptable as offline prediction.The prediction results prove that BFE-ABC algorithm can give PNN more powerful learning capability and much better generalization capability.

5.2 Online prediction

In order to validate the practical effect of the PNN model built by BFE-ABC algorithm, online prediction experiments are done. The datum of next time is predicted based on the data collected now. The PNN model is constructed based on the data from one flight experiment, which is used to predict cylinder head temperature in other flight experiments.

All the cylinder head temperature data in offline prediction, i.e. 200 data, are used to train PNN. The initialization parameters in BFE-RBP algorithm and BFE-ABC algorithm are set the same as in offline prediction. Run BFE-RBP algorithm and BFE-ABC algorithm respectively to get PNN models. The topology structure of PNN optimized by BFE-ABC algorithm is 1-9-1, and the number of expansion terms is 5.

Three groups cylinder head temperature prediction experiments in three different flights are done to validate the prediction performance. The data is processed in real time to apply the PNN models. The prediction curves of three prediction experiments are shown in Figures 5,6,7 respectively. The statistics of prediction results are shown in Table 3.

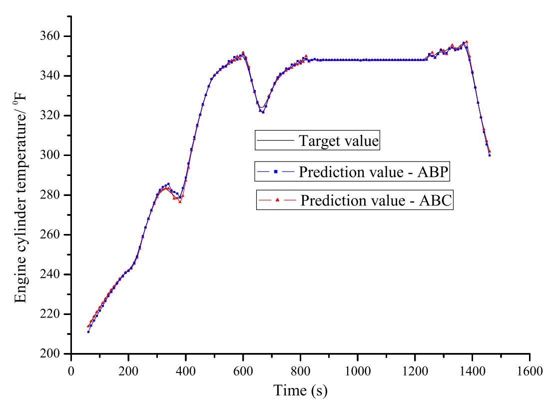


Fig. 5: Online prediction curves of flight 1

It is obvious that the MSE and MRE of the PNN model optimized by BFE-ABC algorithm are smaller than that of BFE-RBP algorithm, i.e., BFE-ABC algorithm outperforms BFE-RBP algorithm. Moreover, unlike BFE-RBP algorithm, the prediction results of the PNN model optimized by BFE-ABC algorithm are stable in these three prediction experiments. The MSE is stable at 0.8 or so, and the MRE is stable at 0.002 or so. It is

Table 2: Training and Prediction Results of Offline Prediction

Algorithm	Training			Prediction	
	Time Consumed (second)	Topology Structure	Number of Expansion Items	MSE	MRE
BFE-RBP	0.5625	1-10-1	6	3.3113	0.0050
BFE-ABC	148.2188	1-7-1	5	1.4241	0.0032

Table 3: Results of Online Prediction

Number	MSE		MRE		Mean Time Consumed (second)	
	BFE-RBP	BFE-ABC	BFE-RBP	BFE-ABC	BFE-RBP	BFE-ABC
Flight 1	0.8972	0.8332	0.0022	0.0017	0.0755	0.0633
Flight 2	1.0093	0.8734	0.0025	0.0019	0.0731	0.0601
Flight 3	7.0013	0.7254	0.0065	0.0019	0.0749	0.0633

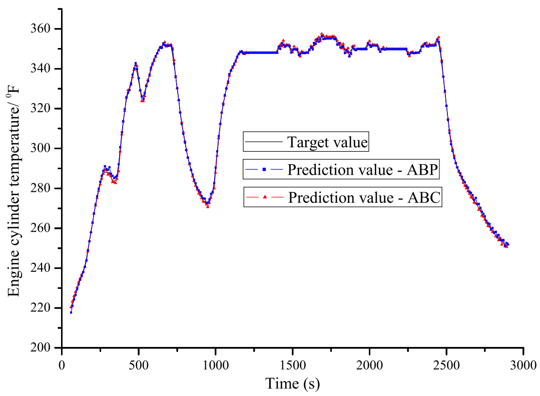


Fig. 6: Online prediction curves of flight 2

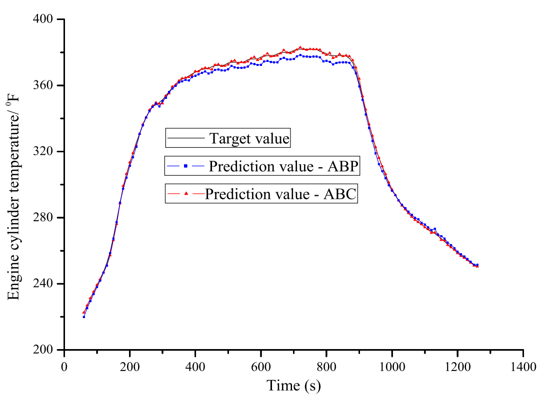


Fig. 7: Online prediction curves of flight 3

obvious that the PNN model constructed through BFE-ABC algorithm has good prediction results in all three flight experiments. However, the PNN model constructed through BFE-RBP algorithm sometimes (in the third flight experiment) predicts badly.

In addition, the PNN model optimized by BFE-ABC algorithm needs less time to predict the datum. It consumes about 0.06s after collecting new data, meanwhile, the PNN model optimized by BFE-RBP algorithm consumes about 0.07s. This is because the PNN model optimized by BFE-ABC algorithm is simpler (smaller topology structure and fewer expansion items) that the processor consumes less time.

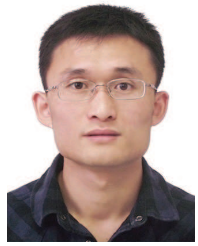
It can be concluded from the analysis of prediction results above that BFE-ABC algorithm has better performance than BFE-RBP algorithm. It can give PNN more powerful learning capability and much better generalization capability. The BFE-ABC algorithm can build a simpler and more precise PNN model. It is more suitable for the time-dependent parameter prediction.

6 Conclusions

A novel learning algorithm named BFE-ABC algorithm is proposed in this paper. BFE-ABC algorithm takes advantage of parameter space transformation in BFE algorithm and global optimization in ABC algorithm. BFE-ABC algorithm also realizes the automatic design of PNN through introducing parametric representation of PNN. The prediction experiment of Mackey-Glass chaotic time series validates the algorithm. The prediction experiment of cylinder head temperature proves that the PNN model optimized by BFE-ABC algorithm has higher prediction precision and consumes less prediction time than the PNN model optimized by BFE-RBP algorithm. The experiment results show that BFE-ABC algorithm can design an optimal PNN automatically in the sense that the number of hidden neurons and expansion items are minimal without losing efficiency. The PNN optimized through BFE-ABC algorithm has higher generalization capability and more effective in time-dependent parameter prediction.

References

[1] X.G. He and J.Z. Liang, Proceedings of the Conference on Intelligent Information Processing, 143-146 (2000).
[2] X.G. He and J.Z. Liang, Engineering Science **12**, 40-44 (2000).
[3] S.H. XU, Y. Wang, H. Wang and X.G. He, Computer Science **11**, 203-205 (2010).
[4] S.H. XU and X.G. He, Chinese Journal of Computerse **5**, 645-650 (2004).
[5] Y.M. Zhou, Z.J. Meng, X.Z. Chen and Z. Wu, Proceedings of IEEE International Conference on Prognostics and Health Management, 1-5 (2012).
[6] A.J. Jones, Neural Computing and Applications **1**, 32-45 (1993).
[7] S.H. Ling, H.K. Lam, F.H.F. Leung and Y.S. Lee, Proceedings of 29th Annual Conference of the IEEE Industrial Electronics Society, 436-441 (2003).
[8] D.K. Agrafiotis and W. Cedeno, Journal of Medicinal Chemistry **5**, 1098-1107 (2002).
[9] Z.W. Wang, G.L. Durst, R.C. Eberhart, D.B. Boyd and Z.B. Miled, Proceedings of 18th International Parallel and Distributed Processing Symposium, 1-8 (2004).
[10] B.A. Garro, H. Sossa and R.A. Vazquez, Proceedings of International Joint Conference on Neural Networks, 938-945 (2009).
[11] K. Liu, Y. Tan, X.G. He, Proceedings of 7th International Symposium on Neural Networks, 280-287 (2010).
[12] B.A. Garro, H. Sossa and R.A. Vazquez, Proceedings of 17th International Conference on Neural Information Processing: Models and Applications, (2010).
[13] J. Ilonen, J.I. Kamarainen and J. Lampinen, Neural Processing Letters **1**, 93-105 (2003).
[14] D. Karaboga, B. Akay and C. Ozturk, Proceedings of 4th International Conference on Modeling Decisions for Artificial Intelligence, 318-329 (2007).
[15] D. Karaboga and B. Akay, Proceedings of 15th Signal Processing and Communication Applications Conference, 318-329 (2007).
[16] B.A. Garro, H. Sossa and R.A. Vazquez, Proceedings of IEEE Congress on Evolutionary Computation, 1-4 (2011).
[17] D. Karaboga, Technical Report-TR06, Erciyes University, 2005.
[18] D. Tereshko, A. Loengarov **3**, 1-7 (2005).
[19] M.C. Mackey and L. Glass. Science **4300**, 287-289 (1977).
[20] G. Ding, M.Q. Xu and L.G. Hou, Journal of Aerospace Power **5**, 1035-1039 (2009).
[21] G. Ding, S.S. Zhong, Control and Decision **9**, 1037-1041 (2006).



Yaoming Zhou received the Ph.D. degree in Aircraft design from Beihang University. Presently he serves as lecturer at the School of Aeronautic Science and Engineering, Beihang University. His research interests include neural networks, fault diagnosis and state prediction, intelligent control for unmanned rotorcraft.

Xuzhi Chen is a Ph.D. candidate of the School of Aeronautic Science and Engineering at Beihang University. His research interests are in the areas of intelligent control for unmanned rotorcraft including path planning, obstacle avoidance, simulation and flight test technology.

Wei He is a Ph.D. candidate of the School of Aeronautic Science and Engineering at Beihang University. He also serves as lecturer at Aviation Theory Department, Aviation University of Airforce. His main research areas include intelligent optimization algorithms, multidisciplinary design optimization and Aerodynamics.

Zhijun Meng received the Ph.D. degree in Aircraft design from Beihang University. Presently he serves as lecturer at the School of Aeronautic Science and Engineering, Beihang University. His research interests include neural networks, intelligent control for unmanned rotorcraft.