Multimed Tools Appl (2017) 76:11961–11974
DOI 10.1007/s11042-016-3776-5

An improved particle swarm optimization algorithm used for BP neural network and multimedia course-ware evaluation

Tianhua Liu[1] · **Shoulin Yin**[1]

Received: 15 October 2015 / Revised: 22 February 2016 / Accepted: 10 July 2016 /
Published online: 29 July 2016

Abstract The original BP neural network has some disadvantages, such as slow convergence speed, low precision, which is easy to fall into local minimum value. So this paper proposes an improved particle swarm optimization (PSO) algorithm to optimize BP neural network. In this new algorithm, PSO uses improved adaptive acceleration factor and improved adaptive inertia weight to improve the initial weight value and threshold value of BP neural network. And we give the detailed improved process. At the end, simulation results show that the new algorithm can improve convergence rate and precision of prediction of BP neural network, which reduces the error of prediction. At the end, we use multimedia evaluation model to verify the new method's performance.

Keywords BP neural network · PSO · Adaptive acceleration factor · Adaptive inertia weight · Multimedia evaluation model

1 Introduction

Chaos phenomenon is a random phenomenon in a certain system. Modeling, prediction and control of chaotic system has become a hot topic. The state of chaos system builds the model of chaos. Its methods include global prediction, locality predication and adaptive prediction. Neural network is an effective tool for approximation and modeling of nonlinear system. It not only achieves global prediction, but realizes the local prediction. Many researchers have studied the model of chaos using neural network [5, 6, 12].

✉ Shoulin Yin
352720214@qq.com

Tianhua Liu
liutianhua@sina.com

[1] Software college, Shenyang Normal University, No.253, HuangHe Bei Street, HuangGu District, Shenyang P. C 110034, China

Wang et al. [10] employed linear matrix inequality and particle swarm optimization based on cellular neural networks. Among these templates obtained by using linear matrix inequality, it utilized the PSO to carry out the optimization parameters. (Jiang et al., 2014) represented that fault diagnosis based on fuzzy neural network and chaos theory could solve uncertain problem essentially, moreover, it was easy to understand and the system features was easy to maintain. Therefore, it was an effective method to diagnose complex system. Wang et al. [10] proposed a hybrid method comprising a chaos synchronization (CS)-based detection scheme and an extension neural network classification algorithm for power quality monitoring and analysis. The new method could detect minor changes in signals of the power systems. Chatzinakos et al. [1] proposed that a member of a time series could be optimally expressed as a deterministic function of the *d* past series values, where *d* is the dimension of the system. A new algorithm was for the estimation of the dimension of chaotic dynamical systems using neural networks and robust location estimate. Therefore, this article uses the BP neural network prediction model which has strong of ability of nonlinear fitting. But this model has the slow convergence speed and low prediction accuracy.

PSO is an effective global optimization algorithm which generates swarm intelligence optimization search by cooperation and competition between particles in the group. So we adopt PSO algorithm to optimize BP neural network. The algorithm divides particles into two parts according to the fitness value. Fitness value can measure the particles mass. By comparing the fitness value of individual particle and average fitness value of particle group, it can distinguishes the merits of particle mass. Also it reasonably adjusts the global searching ability and self-learning ability, social learning ability which is conducive to the rapid convergence of particle swarm. Using particle swarm algorithm optimizes initial weight and threshold of BP neural network. We can get better initial weight and threshold and pass it to the BP neural network. BP neural network will train and predict it. First, PSO uses improved adaptive acceleration factor and improved adaptive inertia weight to improve the initial weight value and threshold value of BP neural network. Second, This paper's new algorithm uses improved PSO to optimize BP neural network's initial weight value and threshold value. Finally, we conduct experiments to verify its feasibility. The follows are the structures of this paper. Section 2 is some basic knowledge about BP neural network and phase-space reconstruction. Section3 is multimedia course-ware evaluation model based on BP neural network. Section 4 introduces the PSO algorithm. Section 5 represents the improved scheme. Section 6 makes simulation and analysis. The conclusions are drawn in section 7.

2 BP neural network and phase-space reconstruction

Phase-space reconstruction [3, 9] is foundation of chaos theory. Setting *X* is the observed chaotic time series. *x(t), t = 1,2,3……n*. Based on Takens theory, it conducts phase-space reconstruction. In the state space, after reconstructing the state vector can be shown as:

$$X_i\left(x_i, x_{i+\omega}, \ldots\ldots, x_{i+(m-1)\omega}\right)^T, i = 1, 2, 3\ldots\ldots M$$

$M = n-(m-1)\omega$ is the number of phase point in phase space. ω is delay time. m is the number of embedded dimension. Takens found a suitable m, $m \geq 2d+1$, d is the dimensions of the power system, which can recover attractor. In the reconstruction space R^m, prime mover system maintain differential homeomorphism, thus it lays a solid theoretical basis for the prediction of chaotic time series.

In essence, prediction of chaotic time series is a reverse problem, it re-factors dynamic model of system through the state of dynamic system. The BP neural network [2, 13] has strong ability of nonlinear fitting, which is a good method to construct chaotic time series nonlinear prediction model. Typical three layer BP neural network is composed of input layer, hidden layer and output layer. Its main principle is that BP neural network transmits error and rectifies the error at the same time to adjust the weights and threshold of network and realize or approximate the input and output mapping relationship. Using three layers BP neural network to predict chaotic time series, the number of input layer neurons can select m which is equal to embedded dimension. The number of hidden layer neurons mostly relies on experience to choose, here it is p. Output layer neuron number is 1. and $f.R^m \rightarrow R^1$. The function of BP neural network is Sigmoid, its output is linear unit.

The input of hidden layer nodes is:

$$S_j = \sum_{i=1}^{m} w_{ij}x_i - \theta_j, j = 1, 2, \ldots\ldots, p$$

Where x_i state vector. w_{ij} is connecting weight between input layer and hidden layer. θ_j is threshold of hidden layer nodes.

The output of hidden layer nodes is:

$$b_j = \frac{1}{1 + e^{\theta_j - \sum_{i=1}^{m} w_{ij}x_i + \theta_j}}, j = 1, 2, \ldots\ldots, p$$

The input of output layer nodes is:

$$L = \sum_{j=1}^{p} v_j b_j - \gamma$$

Where v_j is connecting weight between output layer and hidden layer. γ is threshold of output layer nodes.

The output of output layer nodes is:

$$x_{i+1} = \frac{1}{1 + e^{\gamma - \sum_{j=1}^{p} v_j b_i}}$$

Inputting the chaos time series and training the BP neural network, it will get trained network, and then makes predictions. Before starting work, however, each layer connection weights and threshold values of the BP neural network are randomly initialized, which can reduce the convergence speed and accuracy of BP neural network and makes it easy to fall into local minimum value. Using PSO algorithm can optimize the initial weights and thresholds, the optimized initial weights and thresholds ensure that the BP neural network has better convergence speed and higher prediction precision. BP neural network structure is as Fig. 1.

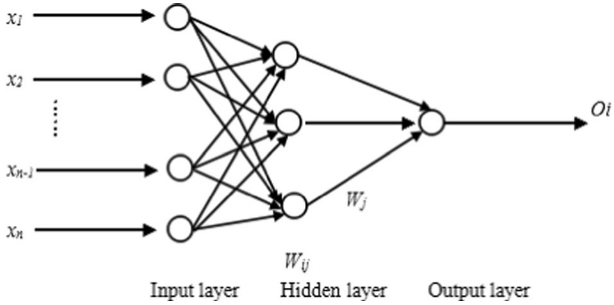


Fig. 1 The structure of BP neural network

3 PSO algorithm

First, PSO algorithm produces initial solution. It initializes N particles to form group randomly in the D-dimension feasible solution space. $z = \{z_1, z_2, \ldots, z_N\}$. Each particle has two position vectors and speed vectors: $z = \{z_{i1}, z_{i2}, \ldots, z_{iN}\}$and $v = \{v_{i1}, v_{i2}, \ldots v_{iN}\}$. Then it uses objective function to calculate fitness value of particles and iteratively searches in the S-dimension solution space. When iterating algorithm, particle will track two extreme value (one is the best solution p_{id} searched by particle itself , one is the best solution p_{gd} searched by group) to update itself. Every particle updates its speed and position by the following:

$$v_{id}(t+1) = \eta v_{id}(t) + a_1 r_1 [p_{id} - z_{id}(t)] + a_2 r_2 \left[p_{gd} - z_{id}(t)\right] \tag{1}$$

$$z_{id}(t+1) = z_{id}(t) + v_{id}(t+1) \tag{2}$$

Where η is weight. a_1 and a_2 are acceleration constant. $0 \le r_1, r_2 \le 1$ are random number. $v_{id}(t+1)$ is *i-th* particle's d-dimension speed in the $t+$ *1-th* iteration.

Linear descend inertia weight η is:

$$\eta = \eta_{\max} - \frac{\eta_{\max} - \eta_{\min}}{iter_{\max}} \times \text{iter} \tag{3}$$

Where *iter* is current iteration's number. $iter_{\max}$ is maximum iteration's number. $\eta_{\max}$is initial inertia weight value. $\eta_{\min}$ is final inertia weight value.

η changing completely dependents on the *iter* and $iter_{\max}$. In later iterations, its convergence rate is slow. It is difficult to jump out of local optimal solution. Since the particles are based on their own experience and the experience of all the particles in the optimal solution direction, under the larger study factor, particle is likely to miss the optimal solution, so that the convergence is slow. In addition, when the algorithm starts convergence, all the particles search solution in the direction of optimal solution. Particles in the group tend to accordance and lose the diversity of solution which reduces the convergence speed and accuracy of algorithm.

4 Multimedia course-ware evaluation model based on BP neural network

Multimedia course-ware is a huge nonlinear system, the relation of influence factors is very complex. Course-ware evaluation, however, is based on a large number of "experience" of original data, so BP network can better meet the needs of evaluation and represent the intrinsic essence of the course-ware. The following will introduce the process of evaluating.

When evaluating the quality of multimedia course-ware, the qualitative index information will be quantitatively conducted and transformed into input vector of neural network $T = \{t_1, t_2, \ldots, t_n\}$. Comprehensive evaluation value obtained by expert will be conducted Fuzzy processing and it will get $Y = \left\{\mu_{S_1}, \mu_{S_2}, \ldots, \mu_{S_n}\right\}$ as the output of neural network. Then it uses enough sample set to train network, through repeated iteration, gets stable training results. At this time, weight value W_{ij} and threshold θ is the correct expression through adaptive learning. And the training results will regarded as knowledge base to evaluate the multimedia course-ware.

- Step1. Determining learning parameters. Supposing expected minimum error value is err_{goal}, maximum iteration is $\max_{epoch}$. Learning rate of amending the weight is lr ranging from 0.01 to 0.7.
- Step2. Network initialization. Using smaller random number initializes the network weights W_{ij} and V_{jk}, threshold value θ_j and r_k.
- Step3. Inputting training set. Supposing input vector of k samples is $t_k = \left(T_1^k, T_2^k, \ldots, T_n^k\right)^T$ and output vector of expected value is $y_k = \left(Y_1^k, Y_2^k, \ldots, Y_n^k\right)^T$.
- Step4. Calculating network output of each layer. Using $t_k = \left(T_1^k, T_2^k, \ldots, T_n^k\right)^T$, W_{ij} and θ_j calculates each neuron of middle layer input : $U_j = \sum_{i=1}^{n} W_{ij} t_i + \theta_j$. Then it adopts activation function $f(x) = 1/(1 + e^{-x})$ to calculate the real output ($H_j = f(U_j)$) of each neuron in middle layer. And it will compute the input value $S_k = \sum_{j=1}^{p} V_{jk} H_j + r_k$ (p is node number of hidden layers) in the output layer and the corresponding output values $O_k = f(S_k)$.
- Step5. Calculating training error. It calculates the error of output layer weights $\delta_k = (O_k - y_k) \cdot (1 - O_k) \cdot O_k$ and the error of middle layer weights $\sigma_j = H_j \cdot \left(1 - H_j\right) \cdot \sum_{k=1}^{q} \delta_k V_{jk}$ (q is output layer node number).
- Step6. Correcting weights and thresholds of output layer and hidden layer. a is weight adjustment parameters, $0 < a < 1$; η is learning coefficient, $0 < \eta < 1$.

$$V_{jk} = V_{jk} + a\delta_k H_j,\ r_k = r_k + a\delta_k,\ W_{ji} = W_{ji} + \eta\sigma_j t_i,\ \theta_j = \theta_j + \eta\sigma_j$$

- Step7. Entering the next learning mode. Return step3 to iterate. Until the error is less than the set minimum error value err_{goal} or leaning number is more than $\max_{epoch}$, the training stops. And evaluation model is built.

5 Improved PSO algorithm

5.1 Improved PSO based on chaos

Chaotic time series has randomness, ergodicity and regularity. Tang et al. [8] combines particle swarm optimization algorithm with chaos and multi-agent system, it is applied to reactive power optimization of power system. It synthesizes advantages of the swarm search of PSO and the intelligent search of agents. She et al. [7] represents that dynamic chaos particle swarm optimization is the improvement of the standard PSO algorithm, is used to optimize the multi-objective optimized mathematical model of elevator configuration. So this paper proposes a new PSO algorithm based on chaos. In this scheme, it uses adaptive control particle's search capability ω and learning ability c_1, c_2 to jump out of local optimum. Algorithm adopts typical Logistic chaotic time series.

$$x_{i+1} = \mu x_i (1 - x_i), \mu = 4 \tag{4}$$

The algorithm divides particles into two parts according to the fitness value. Fitness value can measure the particles mass. By comparing the fitness value of individual particle and average fitness value of particle group, it can distinguishes the merits of particle mass. Also it reasonably adjusts the ω and c_1, c_2 which is conducive to the rapid convergence of particle swarm.

$$\begin{cases} \omega = \omega_0 + k_1 \cdot \mu \cdot x(1 - x) \\ c = c_{10} - k_2 \cdot \mu \cdot x(1 - x), f > f_{avg} \\ c_2 = c_{20} + k_3 \cdot \mu \cdot x(1 - x) \end{cases} \tag{5}$$

$$\begin{cases} \omega = \omega_0 + k_1 \cdot \mu \cdot x(1 - x) \\ c = c_{10} - k_2 \cdot \mu \cdot x(1 - x), f > f_{avg} \\ c_2 = c_{20} + k_3 \cdot \mu \cdot x(1 - x) \end{cases} \tag{6}$$

Where $0 < \omega_0 < 1$, $1 < c_{10} < 2$, $1 < c_{20} < 2$, $0 < k_1 , k_2 , k_3 < 1$ $\mu = 4$. $0 < x < 1$, f is *i-th* particle's fitness value. f_{avg} is all particles' average fitness value.

When $f > f_{avg}$, particle's quality is poor. Equation (5) reduces self-learning ability c_1 by enhancing global searching ability ω and social learning ability c_2. So the particle's quality will be improved. When $f < f_{avg}$, particle's quality is better. Equation (6) reduces social learning ability $c_2$by enhancing global searching ability$\omega$and self-learning ability c_1. So it will protect particles with better quality and solution diversity.

5.2 Improved PSO for BP neural network

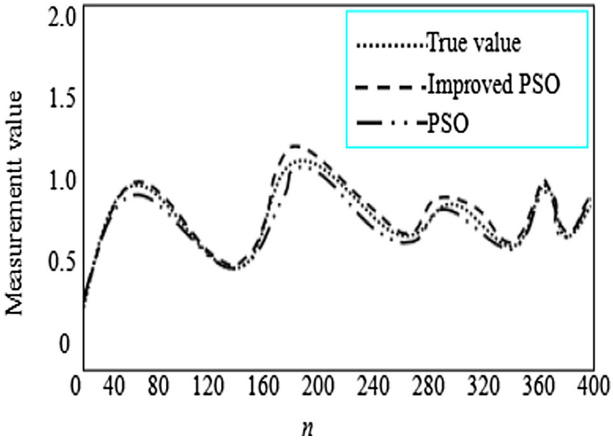
Using particle swarm algorithm optimizes initial weight and threshold of BP neural network. We can get better initial weight and threshold and pass it to the BP neural network. BP neural network will train and predict it. The detailed processes are as following:

- Step1. Initializing particle swarm parameters, population size, iteration number, learning factor, range of speed and position.
- Step2. Building the topological structure of BP neural network, generating particle swarm and initializing the particle's position and speed randomly.

$$z_i = \{z_{i1}, z_{i2}, \cdots, z_{\text{is}}\}, v_i = \{z_{i1}, z_{i2}, \cdots, z_{\text{is}}\}, i = 1, 2 \cdots s = R \times s_1 + s + s_2 \tag{7}$$

Where s is dimension of the particle, R is input layer node number. s_1 is hidden layer node number. s_2 is output layer node number.

Fig. 2 Measurement value $n \leq 400$



- Step3. Calculating fitness value of particle, the fitness value of *i-th* particle is:

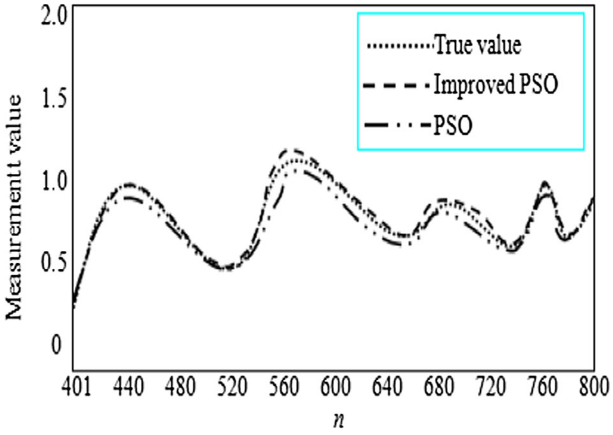
$$f(i) = \sum_{i=1}^{n} \left(y_k - y_k \right)^2, i = 1, 2, \ldots, N \tag{8}$$

Where *n* is training sample number, *N* is population size.

We calculate fitness value of each particle and set the current position as the optimal position p_{id}, set the optimal particle in the population as the global optimal position p_{gd}.

- Step4. Using [4, 11] to update ω, c_1, c_2 and [5, 12] to update position and speed of particle. After the above process, we calculate the fitness value of new particle. Then we update the individual extremum and group extremum according to the fitness value.
- Step5. When the iteration number is less than maximum iteration number, return to step4, otherwise exit the particle optimization algorithm. We get the optimal particle and pass it to the connection weight and threshold of BP neural network.

Fig. 3 Measurement value $400 \leq n \leq 800$



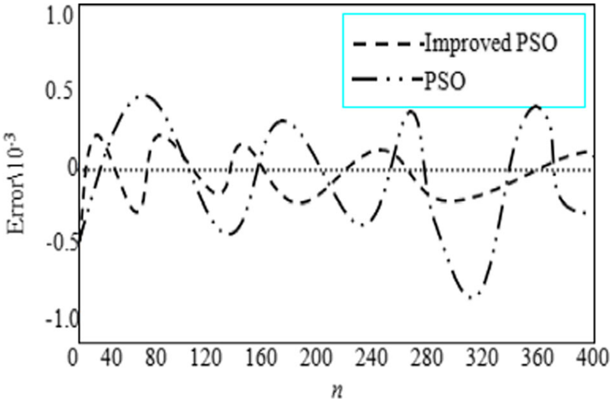


Fig. 4 Error $n \leq 400$

6 Simulation experiments and analysis

Example1 Under the MATLAB platform, we make experiments to compare improved PSO and PSO. We select 800 data to analysis (for the reason that at 400 s, it changes movement direction, we re-record its values at 401 s). In the simulation experiments, we suppose that prediction performance index is error *e*, root mean square error E_{rmse} and relative error *err* , *y* is target value, *y*is prediction value.

$$e = y_k - y_k \tag{9}$$

$$E_{\text{rmse}} = \left(\frac{1}{N}\sum_{k=1}^{N}\left(y_k - y_k\right)^2\right)^{1/2} \tag{10}$$

$$err = \left[\sum_{i=1}^{N}\left(y_i - y_i\right)^2\right] / \left[\sum_{i=1}^{N} y_1^2\right] \tag{11}$$

For the traditional PSO, we set the training number 800, learning rate 0.01, population size 30, evolutional generation 100, accelerated factor $c_1 = c_2 = 2$. The range of position and speed is [−5,5] and [−1,1] respectively.

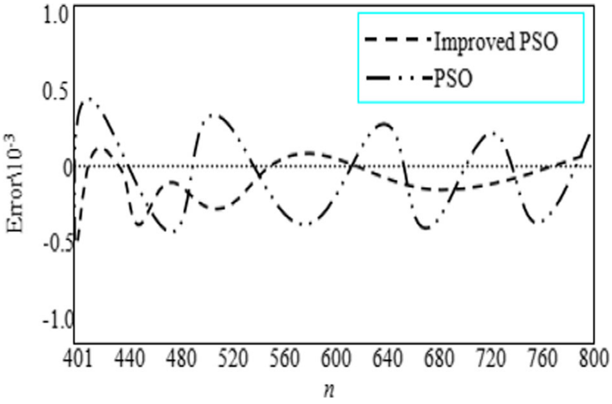


Fig. 5 Error $400 \leq n \leq 800$

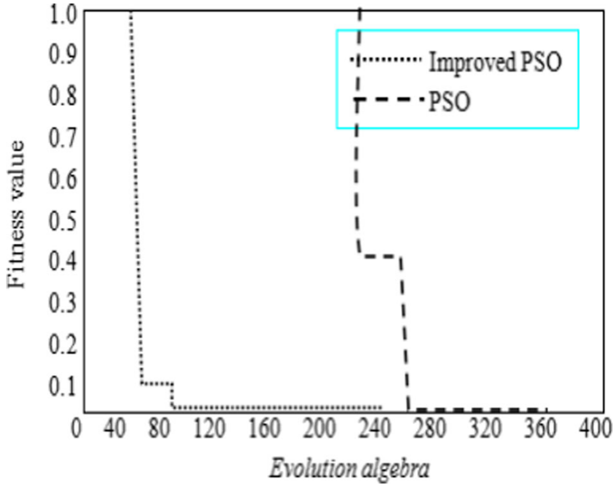


Fig. 6 Fitness value $n \leq 400$

For the improved PSO, we set training number 800, learning rate 0.01, population size 30, evolutional generation 100, $c_{10} = c_{20} = 1.6$, $k_1 = 0.3$, $k_2 = k_3 = 0.5$. Range of inertia weight and accelerating factor is [0.2,0.9] and [1.2,2.4] respectively. Range of particle position and speed is [−5,5] and [−1,1] respectively. We use these data to make experiments and get Figs. 2 and 3.

From Figs. 2 and 3, we can know that the improved PSO is much closer to the true value. We select the same experiments parameter. The convergence speed of improved PSO is faster than PSO algorithm.

Figs. 4 and 5 are the prediction error. The error of improved PSO is small than PSO algorithm. The smaller the fitness value is, the better the quality of the particles is (as Figs. 6 and 7). The initial weight and threshold are more conducive to the BP neural network training. BP neural network has fast convergence with good initial weight and threshold. It will improve the prediction accuracy and reduce the BP neural network to fall into a minimal value. From the above results, we can conclude that improved PSO is conducive to globally optimized solution.

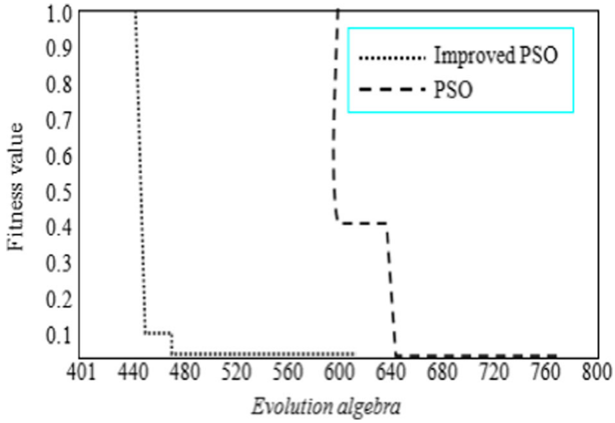


Fig. 7 Fitness value $400 \leq n \leq 800$

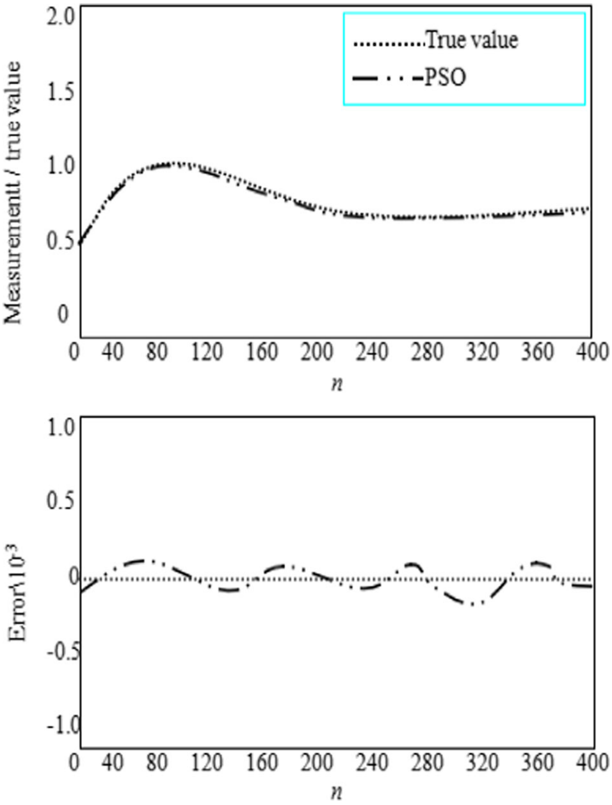


Fig. 8 PSO prediction and its error

Also we use PSO and improved PSO to predict chaotic system (Lorenz) at the same condition of physical. We get figs. 8 and 9. From this two figures, we can obtain that improved PSO are very precise prediction and has low error than PSO method.

Example 2 Selecting fifteen typical course-wares as the evaluation objects. Experts give evaluation interval for index effectiveness and course-ware integrated level for each course-ware. The evaluation basis is as follows: bad [0,0.55], pass [0.55,0.7], good [0.7,0.85], excellent [0.85,1]. We get a sample data as Table 1. And it calculates $Y = \left\{\mu_{S_1}, \mu_{S_2}, \dots, \mu_{S_n}\right\}$ as output data represented in Table 2. The former 10 data is as training set and the later 5 data is as testing set.

Table 2. shows that network training is successful, the error is small, and it can accurately evaluate the quality of multimedia course-ware for students.

7 Conclusions

This paper's aim is to perfect the convergence speed, accuracy of algorithm and to solve the problem with Local minimum shortage and slow convergence speed in BP neural network. Also we finally build multimedia course-ware evaluation model based on the improved PSO. First, we give a detailed explanation on BP neural network and phase-space reconstruction.

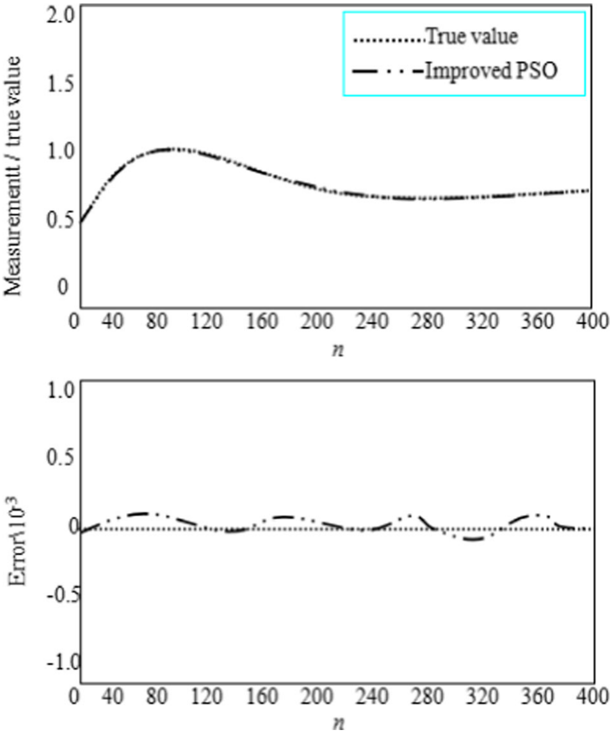


Fig. 9 Improved PSO prediction and its error

Table 1 Sample input data

Sample	Index set										
	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}
T1	0.95	0.88	0.91	0.95	0.84	0.91	0.94	0.98	0.87	0.84	0.96
T2	0.82	0.95	0.94	0.83	0.84	0.87	0.91	0.95	0.97	0.86	0.89
T3	0.84	0.87	0.77	0.78	0.94	0.85	0.90	0.91	0.97	0.86	0.88
T4	0.76	0.72	0.64	0.84	0.82	0.78	0.81	0.87	0.89	0.86	0.85
T5	0.89	0.92	0.81	0.75	0.56	0.64	0.89	0.84	0.82	0.80	0.73
T6	0.58	0.47	0.75	0.55	0.57	0.45	0.34	0.54	0.60	0.65	0.79
T7	0.25	0.34	0.40	0.28	0.17	0.64	0.35	0.44	0.48	0.68	0.65
T8	0.64	0.61	0.44	0.48	0.32	0.94	0.85	0.56	0.65	0.85	0.48
T9	0.74	0.67	0.65	0.77	0.79	0.80	0.79	0.69	0.76	0.78	0.82
T10	0.87	0.78	0.83	0.85	0.76	0.89	0.91	0.81	0.87	0.85	0.82
T11	0.84	0.89	0.86	0.82	0.91	0.84	0.87	0.95	0.88	0.76	0.79
T12	0.67	0.77	0.74	0.84	0.68	0.62	0.65	0.74	0.89	0.62	0.65
T13	0.73	0.84	0.71	0.77	0.92	0.85	0.70	0.91	0.90	0.91	0.88
T14	0.49	0.55	0.59	0.63	0.57	0.48	0.34	0.84	0.71	0.64	0.53
T15	0.65	0.74	0.67	0.84	0.85	0.83	0.89	0.92	0.55	0.86	0.61

Table 2 Sample output data

Sample output	Group evaluation value (v)	Fuzzy set				Evaluation degree
		μ_{S_1}	μ_{S_2}	μ_{S_3}	μ_{S_4}	
Y1	0.94	0	0	0	1.0	excellent
Y2	0.87	0	0	0.3	0.7	excellent
Y3	0.81	0	0	0.7	0.3	good
Y4	0.75	0	0.3	0.7	0	good
Y5	0.79	0	0	1.0	0	good
Y6	0.54	0.5	0.5	0	0	pass
Y7	0.38	1.0	0	0	0	bad
Y8	0.67	0	0.6	0.4	0	pass
Y9	0.82	0	0	0.6	0.4	good
Y10	0.84	0	0	0.5	0.5	good
Y11	0.81	0	0	0.7	0.3	good
Y12	0.73	0	0.2	0.8	0	good
Y13	0.81	0	0	0.7	0.3	good
Y14	0.52	0.8	0.2	0	0	bad
Y15	0.82	0	0	0.8	0.2	good

Multimedia course-ware evaluation model based on BP neural network is further illustrated. Then we propose improved PSO to optimize the initial weight and threshold value. Compared with the traditional PSO, the convergence speed of improved PSO becomes faster. It will add the probability of jumping out of local optimal solution and it is conducive to find the global optimal solution. This paper makes experiments with chaos system to verify the superiority of improved PSO. The new scheme improves convergence speed and prediction accuracy and reduces the probability of falling into local value with BP neural network. Multimedia course-ware evaluation model is well improved, which is conducive to valuating multimedia course-ware in some universities. In the future, we will study the advanced PSO algorithm based on this paper's method to solve the accuracy and convergence speed problem.

Acknowledgments Foundation item: The National Nature Science Foundation of China (No.: 6097011). Authors are grateful to the Software College Shenyang Normal University, Government of China for financial support to carry out this work.

References

1. Chatzinakos C, Tsouros C (2015) Estimation of the dimension of chaotic dynamical systems using neural networks and robust location estimate [J]. Simul Model Pract Theory 51:149–156
2. Du, W. (2015) Research on Evaluation of Enterprise Performance Based on BP Neural Network Improved by Levenberg-marquardt Algorithm [C]// International Conference on Automation, Mechanical Control and Computational Engineering. Atlantis Press, 2015
3. Han L, Romero CE, Yao Z (2015) Wind power forecasting based on principle component phase space reconstruction [J]. Renew Energy 81:737–744
4. Jiang YL, Shao YX Fault diagnosis of rolling bearing based on fuzzy neural network and chaos theory [J]. Vibroeng PROCEDIA 4:211–216
5. Jinkun L, Yi G. (2014) Neural network based adaptive dynamic surface control for flexible-joint robots[C]// Control Conference (CCC), 33rd Chinese IEEE, 2014: 8764–8768
6. Liu B, Che Y, Cui X, et al. (2015) Synchronization of Light-Emitting Diodes Networks via Adaptive Neural Network Control [C]//2015 International Symposium on Computers & Informatics. Atlantis Press
7. She K, Ma LZ, Wan JR, et al. (2015) Application of Dynamic Chaos PSO Algorithm in Elevator Configuration [C]. Appl Mech Mater 734:548–553
8. Tang X, Zhang H, Cui Y, et al. (2014) A novel reactive power optimization solution using improved chaos PSO based on multi-agent architecture [J]. Int Trans Electr Energy Syst 24(5):609–622
9. Wan L, Zhu YQ, Deng XC (2015) Analysis on phase space reconstruction and chaotic characteristic of element concentration series of gold deposit [J]. Analysis 5:6
10. Wang Z, Li L (2014) Image Edge Detection Based on Cellular Neural Network and Particle Swarm Optimization [J]. Trans. Computer Sci Technol 3(1)
11. Wang MH, Yau HT (2014) New Power Quality Analysis Method Based on Chaos Synchronization and Extension Neural Network [J]. Energy 7(10):6340–6357
12. Yoshida H, Murakami T, Liuc Z. (2015) High-Speed and Highly Secure Pseudo-Random Number Generator based on Chaos Neural Network [J]. New Trends on System Science and Engineering: Proceedings of ICSSE 2015, 276: 224
13. Zhang Y, Gao X, Katayama S (2015) Weld appearance prediction with BP neural network improved by genetic algorithm during disk laser welding [J]. J Manuf Syst 34:53–59

Tianhua Liu received the B. Eng. and M. Eng. degrees from the Northeastern University, Shenyang, China, in 1983 and 1987, respectively, the Ph.D. degree from Northeastern University, Shenyang, China, in 1991, all in computer science and engineering. He is currently the Shenyang Normal University Professor, head of the School of of Academic Affairs with Shenyang Normal University. He is the manuscript reviewer in some famous international journals. His research interests include multimedia application, distributed systems, network security, Searchable Encryption Schemes, and e-Learning

Shoulin Yin received the B. Eng. and M. Eng. degree from Shenyang Normal University, Shenyang , Liaoning province, China in 2013 and 2015 respectively. His research interests include Multimedia Security, Network Security, Filter Algorithm and Data Mining. He received School Class Scholarship in 5.2015