International Journal of Hybrid Information Technology
Vol. 9, No.10 (2016), pp. 313-324
http://dx.doi.org/10.14257/ijhit.2016.9.10.31

The Research of Constrained Optimization Method Based on BP Neural Network and Its Application

Shuang-jing Li and Fu-lin Wang

(Northeast Agricultural University, Harbin 150030, China)
fulinwang@yahoo.com.cn;lishuangjing1030@163.com

Abstract

This essay proposes a method of BP neural network constrained optimization based on previous studies. The optimization method based on the BP neural network, takes the minimum output of a neural network as an example, gives the general mathematical models, derives and gives the partial derivatives of BP network's output to input, and uses the Sigmoid Function as the transmission function in the article. On the previous studies basis, the basic ideas, algorithms and related models are given, based on the constrained optimization issues of BP neural network. We can adjust the input values of BP neural network to obtain the minimum or maximum output value by using this method. This optimization method links the optimization and fitting of BP networks together and expands the application of the BP neural network. At last, in this essay, the optimization method is applied in an example.

***Keywords**: Back-propagation; Neural network; Constrained; Optimization method*

1. Preface

A neural network model is an intelligent data processing and mapping model that is conducted by imitating the human neural system. In the field of intelligent control, the neural network model is a hot research spot. [1, 2]Although individual neuron or simple structure networks have limited functionality, the network system with many neurons can realize powerful functions.

BP neural network is called back-propagation neural network. It is the most widely applied in the neural network field. The back-propagation neural network algorithm idea is that the network training course consists of the forward-propagation course and deviation back-propagation course. [3] Through signal forward-propagation and deviation back-propagation, the weight of each network layer is adjusted. Many scholars have conducted the research network structure problems. It has been shown that a 3-layer BP neural network has the ability of simulating arbitrarily complex nonlinear mapping, provided the numbers of implied layer nodes are enough. [4-7] Strong fitting ability is the advantage of BP neural network.

In fact, sometimes many people not only care about the matching effect of the neural network, but are also concerned about how to adjust the input value to obtain the maximum or minimum value. As a matter of fact, this issue is an optimized issue of the BP neural network, and the studies of optimization are rare. [8-10]

Although there are some researches on optimization of BP neural network, the studies are focused on the parameters of BP neural network optimization, such as rate of learning, initial weights, the structure of networks, or choice of a better result according to the relations of BP neural network's input and output.[11-14] These researches are not the real optimization studies but a kind of simulation that selects a more optimal solution from the results. Therefore, the study of the real optimization method has important theoretical significance and application value.

In the essay, we do further research on the basis of the previous research. A method of constrained optimization is proposed. That is based on the neural network fitting; we can adjust the input values to obtain the maximum or minimum output. Therefore, the fitting and optimization of BP neural network are linked together in this way. The optimization method expands the application of BP neural network. At last, the method is applied in an example in this essay because of its effectiveness.

2. The BP Neural Network's Structure and Algorithm

2.1. The BP Neural Network's Structure

In practical application, we always use 3- layer BP neural network, as shown in Figure 1.

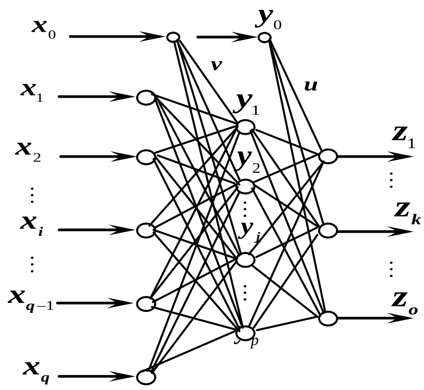

Figure 1. The 3-Layer BP Network

2.2. The BP Neural Network's Algorithm

The basic algorithm thought of BP Neural Network is that the training course consists of signal forward-propagation course and deviation back-propagation course. In forward-propagation phase, the input signals enter from the input layer, go through implied layers and be treated, then enter the output layer.[15] Signals will go back to the back-propagation phase if the real output of output layer are not accordant with expected output. Back-propagation phase is the stage that export errors are transferred back from output layer to implied layer, step by step, until to the input layer. The deviations will be shared by every layer and unit. Unit weights are adjusted according to the deviation signals. The two phases are carried out repeatedly and the weights adjusted repeatedly. The phase of weights' adjustment is the training phase of the neural network. The two phases are executed repeatedly until the deviation of output is decreased to an appropriate level or reaching the certain learning times. [16-17]

(1) The signal forward-propagation process: The signals input from the input layer, then go through implied layers, and output signals are produced on the output side. When the output signals can accord with the requirement of a given output, the calculation is over; If the output value can not meet the demands of a given value, the export signals will be passed to the counterpropagation phase.

The process of the forward-propagation is as follow:

Assuming that, there are $q+1$ inputs of import layer, and any input signal express as i . There are $p+1$ neurons in implied layer, and any neuron express as j; In export layer, there are o neurons and any neuron express as k . The weights of import layer to implied layer are expressed as $v_{ij}\left(i = 0,..., q; j = 1, 2,..., p\right)$, v_{0j} is the threshold of implied layer; The implied layer to export layer weights are expressed as $u_{jk}\left(j = 0,..., p, k = 1, 2,..., o\right)$, u_{ok} is the threshold value of export layer. Assuming that, net_j, $(j = 1, 2, \ldots p)$ is the import signals of implied layer; The implied layer to export layer weights are expressed as y_j, $(j = 1, 2,..., p)$, net_k, $(k = 1, 2,..., o)$ is the import of the output layer, z_k, $(k = 1, 2,..., o)$ is the output of the export layer. Assuming that, the training sample set is $X = \left[X_1, X_2,..., X_r,..., X_n\right]$, corresponding to any training sample is $X_r = [x_{r0}, x_{r1}, x_{r2},..., x_{rq}]$, $r = 1, 2,..., n$, and $x_{r0} = -1$. Real outputs are $z_r = \left[z_{r1}, z_{r2},..., z_{ro}\right]^T$, and expected outputs are $d_r = \left[d_{r1}, d_{r2},..., d_{ro}\right]^T$. If m is the number of iterations, the weight is the function of m. and the real output is also.

The training sample network input is X_r, the signal forward-propagation process of work signal is:

$$net_j = \sum_{i=0}^{q} v_{ij} x_{ri} \quad j = 1,2,\cdots,p \tag{1}$$

$$y_j = f\left(net_j\right) = f\left(\sum_{i=0}^{q} v_{ij} x_{ri}\right) \quad j = 1,2,\cdots,p \tag{2}$$

$$net_k = \sum_{j=0}^{p} u_{jk} y_{rj} \quad k = 1,2,\cdots,o \tag{3}$$

$$z_k = f\left(net_k\right) = f\left(\sum_{j=0}^{p} u_{jk} y_{rj}\right) \quad k = 1,2,\cdots,o \tag{4}$$

The first k neurons error signal of the output layer is,

$$e_{rk}\left(m\right) = d_{rk}\left(m\right) - z_{rk}\left(m\right)$$

The error energy of neuron k is $e_{rk}^2\left(m\right)/2$, The total error of all the output layer neurons is $E(m)$:

$$E\left(m\right) = \frac{1}{2}\sum_{k=1}^{o} e_{rk}^2\left(m\right) \tag{5}$$

When $E(m) \le e$ (e is calculating expected accuracy), Calculation is over. Otherwise, the back-propagation calculation begins.

(2) The error back-propagation process: The network error is the difference between the real output and expected output value. Then the error signals spread forward step by step. This is the error back-propagation process. The network's weight values are corrected by the error feedback in the back-propagation phase. The real output gradually close to the expected output by adjusting the weights.

The related types of error back-propagation are:

$$\Delta u_{jk}(m) = -\eta \frac{\partial E(m)}{\partial u_{jk}(m)} \tag{6}$$

$$u_{jk}(m+1) = u_{jk}(m) + \Delta u_{jk}(m) \tag{7}$$

$$\Delta v_{ij}(m) = -\eta \frac{\partial E(m)}{\partial v_{ij}(m)} \tag{8}$$

$$v_{ij}(m+1) = v_{ij}(m) + \Delta v_{ij}(m) \tag{9}$$

In these formulas, η is called the rate of learning, it is usually a constant we have give.

While the new weight values are calculated, the forward-propagation process will begin.

3. The Method of Constrained Optimization

3.1. Mathematical Types

Because $minF(X) = max[-F(X)]$, the problem of maximum can be changed to the problem of minimum. In order to illustrate this problem conveniently, for an example, we want to seek the maximum of network output. Assuming that, the function of input and output relationship is $F(X)$, the constraints are $g_j(X) \geq 0, h_i(X) = 0$, then the mathematical types of BP neural network constrained optimization problem can be expressed as follows:

$$maxZ = maxF(X) \tag{10}$$

$$\begin{cases} h_i(X) = 0 & i = 1,2,...,a \\ g_j(X) \geq 0 & j = 1,2,\mathrm{L}\ ,b \end{cases} \tag{11}$$

In the formulas, $X = (x_1,\ x_2,...,\ x_q)^T$ is the input vector, the BP neural network output value is Z. a,b are the equation and inequation numbers in constraint conditions.

Because the equality constraints can be eliminated through elimination method, the mathematical types of the constrained optimization problem can also be expressed as follows:

$$maxZ = maxF(X) \tag{13}$$

$$s.t.\ \ g_j(X) \geq 0 \qquad j = 1,2,\cdots,b \tag{14}$$

In the formulas, X is the input. $X = (x_1,\ x_2,...,\ x_{q-a})^T$.

3.2. The Basic Ideas

We select or generate a initial point that is feasible randomly, express as $X(0)$, then calculate the gradient of $X(0)$. The point $X(0)$ is the optimal solution if the gradient of $X(0)$ is 0. If the gradient of this point is not 0, we will add a step length, called rate of learning, and create a new point along the gradient direction of $X(0)$ point, express as $X(1)$. If $X(1)$ does not meet the constraint condition, the step length will be halved, and create a new point again, if the new point still do not meet the constraint condition, halved the step length, until meet the constraint conditions. Then check whether the step length is 0, if it is zero, $X(0)$ is the optimal solution, calculation is over; If the step length is not 0, and the new point is superior to $X(0)$, then create the new point, express as $X(0)$, iteration is completed, and then, begin the next iteration calculation from the new point. If the step length is not 0, and the new point is not superior to $X(0)$, continue to reduce the step length, until the new point is superior to $X(0)$, or the terminal conditions is met. If the new point was better than that of $X(0)$, then created the new point $X(0)$, iteration is completed, and then starting from the new point, the next iteration calculation began.

If $X(1)$ satisfy the constraint conditions, and $X(1)$ is better than that of $X(0)$, then increase the step length, to create a new point, if the new point still meet the constraint conditions, and is better than the last point, continue to increase the step length, until the new point is not better than the last point or do not meet the constraint condition, then create the last calculated point as $X(0)$, completed an iterative calculation, and then starting from this point, the next iteration calculation begin.

If $X(1)$ satisfy the constraint conditions, but $X(1)$ was not better than $X(0)$, then halved the step length, created a new point, if the new point do not meet the constraint condition, then continue to halve the step length, until meet the constraint conditions. If the new point satisfy the constraint conditions, but still not better than $X(0)$, then continue to halve the step length, until the new point is superior to $X(0)$ or meet the terminal condition. If the new point is superior to $X(0)$, then create the new point as $X(0)$. Iteration is completed, and then from the point, the next iteration calculation will begin.

Continue to calculate, until meet the terminal conditions (that is, the step length or the gradient is zero).

3.3. Calculation of Partial Derivative

Because of the gradient function of $F(X)$ is,

$$gradF(X) = (\frac{\partial F}{\partial x_1}, \frac{\partial F}{\partial x_2}, \cdots, \frac{\partial F}{\partial x_n}) \qquad (15)$$

So, if we obtained the partial derivative of function $F(X)$, we could obtain the gradient of $F(X)$. The following is the derivation process of output to input partial derivative in the BP neural network.

Generally, the transmission function is sigmoid function, that is,

$$f(x) = \frac{1}{1+e^{-x}} \qquad (16)$$

Now take the transmission function as a single polarity sigmoid function, the partial derivative of output to input is derived.

Due to the derivative of $f\left(x\right)$ is,

$$f^{'}(x)=f(x)[1-f(x)] \tag{17}$$

$$\frac{\partial Z_k}{\partial x_i}=\frac{\partial Z_k}{\partial net_k}\cdot\left(\sum_{j=0}^{p}\frac{\partial net_k}{\partial y_j}\cdot\frac{\partial y_j}{\partial net_j}\cdot\frac{\partial net_j}{\partial x_i}\right) \tag{18}$$

$(k=1,2,\ldots,o;\ i=1,2,\ldots,q;\ j=1,2,\ldots,p)$

$$\frac{\partial Z_k}{\partial net_k}=Z_k(1-Z_k) \tag{19}$$

$$\frac{\partial net_k}{\partial y_j}=u_{jk} \tag{20}$$

$$\frac{\partial y_j}{\partial net_j}=y_j(1-y_j) \tag{21}$$

$$\frac{\partial net_j}{\partial x_i}=v_{ij} \tag{22}$$

So,

$$\frac{\partial Z_k}{\partial x_i}=Z_k\left(1-Z_k\right)\cdot\sum_{j=0}^{p}\left[u_{jk}\cdot v_{ij}\cdot y_j\left(1-y_j\right)\right] \tag{23}$$

Order, $\omega_k=\frac{\partial Z_k}{\partial net_k}=Z_k(1-Z_k)$ (24)

$$\begin{aligned}\sigma_k&=\sum_{j=0}^{p}\left(\frac{\partial net_k}{\partial y_j}\cdot\frac{\partial y_j}{\partial net_j}\cdot\frac{\partial net_j}{\partial x_i}\right)\\&=\sum_{j=0}^{p}[\mathrm{u}_{jk}\cdot v_{ij}\cdot y_j(1-y_j)]\end{aligned} \tag{25}$$

So, $\frac{\partial Z_k}{\partial x_i}=\omega_k\sigma_k$ (26)

3.4. The Method of Constrained Optimization

Supposed, we give or generate an initial point $X(0)$ randomly, and it is feasible. Then create $X(\mathrm{m})$ as the feasible point of the first m iteration, then gradient of $X(\mathrm{m})$ can be get. That means:

$$\frac{\partial Z}{\partial X}\Big|_{X=X(m)} \qquad m\in[0,1,2,\mathrm{L}\ .] \tag{27}$$

If $X(\mathrm{m})$ satisfy the terminating condition, then:

$$\left\| \frac{\partial Z}{\partial X} \Big|_{X=X(m)} \right\| = 0 \quad m \in [0,1,2,\ldots] \tag{28}$$

In the formula,

$\left\| \frac{\partial Z}{\partial X} \Big|_{X=X(m)} \right\|$ is $\frac{\partial Z}{\partial X} \Big|_{X=X(m)}$'s module.

So, the optimal solution is

$$X^{*}=X(m) \tag{29}$$

The corresponding Z^{*} is the optimal solution.

If it do not meet the formula (28), then order:

$$\Delta X(m) = \lambda \frac{\partial Z(m)}{\partial X(m)} \tag{30}$$

$$X(m+1) = X(m) + \Delta X(m) \tag{31}$$

In the formula， λ is the step length factor, and $\lambda > 0$.

If $X(m+1)$ do not meet the constraint conditions, then order

$$X(m+1) \Leftarrow X(m) + \alpha X(m+1) \tag{32}$$

In the formula, α is contraction coefficient, and $0 < \alpha < 1$， in general, $\alpha = 0.5$.

If the new point $X(m+1)$ does not meet the constraint condition, then order:

$$\alpha \Leftarrow 0.5\alpha \tag{33}$$

According to formula(32), calculate until the $X(m+1)$ constraint is satisfied. And then determine whether meet the formula

$$\alpha = 0 \tag{34}$$

If it meets the formula(34), then order:

$$X^{*}= X(m+1) \tag{35}$$

In the formula, $X*$ is the best solution, the corresponding $Z*$ is the required best solution.

If it does not meet the formula (34), then decided whether $X(m+1)$ is better than that of $X(m)$, that is, whether it meets the formula:

$$F(X(m+1)) > F(X(m)) \tag{36}$$

If it meets formula (36), starting from point $X(m+1)$, the next iteration calculation will begin; If $X(m+1)$ does not meet the formula (36), then according to formula (33) , continue to reduce the value of α, until it meets formula (34) or formula (36).

If $X(m+1)$ meets the constraint conditions and is superior to $X(m)$, then order

$$\lambda \Leftarrow 2\lambda \tag{37}$$

$$X(m) \Leftarrow X\left(m+1\right) \tag{38}$$

Then we calculate again according to the formula (30) with formula (31), and get a new point $X\left(m+1\right)$. If the new values of $X\left(m+1\right)$ does not meet the constraint conditions or does not meet the formula (36), the value of $X\left(m+1\right)$ takes the value of $X\left(m\right)$. Then starting from point $X\left(m+1\right)$, the next iteration calculation will begin. If the new values of $X\left(m+1\right)$ meets the constraint conditions and meets the formula(36), then put the value of $X\left(m+1\right)$ to $X\left(m\right)$ according to formula(38), and continue to increase the step length according to formula(37), then calculate again according to formula(30) and formula(31), until the values of $X\left(m+1\right)$ does not meet the constraint conditions or not better than that of the $X\left(m\right)$, then the value of $X\left(m+1\right)$ take the value of $X\left(m\right)$. Then starting from point $X\left(m+1\right)$ the next iteration calculation began.

If $X\left(m+1\right)$ satisfy the constraint conditions but is not superior to $X\left(m\right)$, halved the step length according to the formula(33), and then calculate again according to the formula(32), create a new point $X\left(m+1\right)$.If the new point $X\left(m+1\right)$ does not meet the constraint condition, halved the step length, until meet the constraint conditions. If the new point $X\left(m+1\right)$ meets the constraint conditions, but still not better than $X\left(m\right)$, then continue to reduce the value according to formula(33), and then calculate again according to the formula (32), and obtain a new $X\left(m+1\right)$, so go ahead, until you got the new point $X\left(m+1\right)$ that is better than that of $X\left(m\right)$ or meet the terminal condition. If the new point $X\left(m+1\right)$ is better than that of $X\left(m\right)$, starting from point $X\left(m+1\right)$, the next iteration calculation will begin.

Continue to calculate, until the point meets the terminal conditions (that is, the gradient or step length 0).

4. The Example Calculation

The following constrained optimization problem:

$$\begin{cases} \min F(X) = x_1^2 - 2x_1x_2 + 2x_2^2 - 3x_1 - 1.5x_2 \\ x_1 + x_2 \le 30 \\ 4x_1 + x_2 \le 90 \\ x_1, x_2 \ge 0 \end{cases} \tag{45}$$

In the problem, the theoretical results is: $minF\left(X\right) = minF\ (1.875,1.125) = -4.8125$. Function fitting is used base on BP neural network, and the network output minimum calculated after fitting.

First, change the objective function of the formula (45) into a maximum, that is:

$$\max F(X) = \min[-F(X)] \tag{46}$$

x_1 selects 1 point every 1.5 unit in the interval $[0, 22.5]$, a total of 16 points, x_2 selects 1 point every 1.2 unit in the interval $[0, 30]$, a total of 16 points, removed the points that did not meet the constraint condition, a total of 157 points, and then calculated the corresponding value of $F(X)$ according to objective function of the type(45) ,and using BP neural network for function fitting, the network structure is 2-10-1, when meet the given fitting precision $e = 1 \times 10^{-8}$, from import to hidden layer, v matrix expresses as the weights, from hidden to export layer , u matrix expresses as the weights. In hidden layer, x_0 expresses as the thresholds matrix. In output layer, y_0 is the threshold matrix.

$$v = \begin{bmatrix} v_{1,1} & v_{2,1} \\ v_{1,2} & v_{2,2} \\ v_{1,3} & v_{2,3} \\ v_{1,4} & v_{2,4} \\ v_{1,5} & v_{2,5} \\ v_{1,6} & v_{2,6} \\ v_{1,7} & v_{2,7} \\ v_{1,8} & v_{2,8} \\ v_{1,9} & v_{2,9} \\ v_{1,10} & v_{2,10} \end{bmatrix} = \begin{bmatrix} -1.6321 & 4.3337 \\ 1.3042 & -2.3441 \\ 0.6377 & -1.1747 \\ 0.0666 & 0.4171 \\ 0.3307 & 0.7328 \\ -0.6434 & -0.1003 \\ 0.1741 & -1.6626 \\ 0.3430 & 1.0162 \\ -0.0703 & -0.4206 \\ -0.0722 & -0.4423 \end{bmatrix} \tag{47}$$

$$u = \begin{bmatrix} u_{1,1} & u_{2,1} & u_{3,1} & u_{4,1} & u_{5,1} & u_{6,1} & u_{7,1} & u_{8,1} & u_{9,1} & u_{10,1} \end{bmatrix} = \begin{bmatrix} 2.5565 \ -0.1062 \ -1.5305 \ 1.2468 \ -2.3204 \\ -1.6470 \ 1.2567 \ 1.2641 \ -1.2612 \ -1.9443 \end{bmatrix} \tag{48}$$

$$x_0 = \begin{bmatrix} x_{0,1} & x_{0,2} & x_{0,3} & x_{0,4} & x_{0,5} & x_{0,6} & x_{0,7} & x_{0,8} & x_{0,9} & x_{0,10} \end{bmatrix} = \begin{bmatrix} -4.1706 \ 0.0862 \ 0.8575 \ 0.6721 \ 0.1738 \ 0.9385 \\ -1.2111 \ 0.2072 \ -0.6950 \ -1.0896 \end{bmatrix} \tag{49}$$

$$y_0 = 3.1455 \tag{50}$$

The average relative error is 0.055114%. Then we calculate $minF(X)$ according to the optimization method from the essay. When $\varepsilon = 0$, choosing ten different initial points, the results are demonstrated in the Table. And $\bar{F}$ is the optimal results average in Table 2.

Table 2. Results of $F(x)$ Values

Num	1	2	3	4	5	6	7	8	9	10
X(0)	0	3	6	9	12	15	18	21	24	27
	0	2	4	6	8	10	12	14	16	18
x_1	3.7512	3.7512	3.7512	3.7512	3.7512	3.7512	3.7512	3.7512	3.7512	3.7512
x_2	2.2537	2.2537	2.2537	2.2537	2.2537	2.2537	2.2537	2.2537	2.2537	2.2537
minF	-4.810	-4.810	-4.810	-4.810	-4.810	-4.810	-4.810	-4.810	-4.810	-4.810
$\bar{F}$	-4.810									
β_i	1	1	1	1	1	1	1	1	1	1

The Table 2 shows that the optimized results are stable, the optimized results come near to the theoretical optimized results, the relative average error of x_1 is 0.022%, the average relative error of x_2 is 0.1644%. The BP neural network optimization method come near to the optimized result of theoretically, the fractional error (including fitting error) is 0.0517%, thus, the precision of the optimization method is high-precision.

5. Conclusion

(1) The essay puts forward a constrained optimization method of BP neural network based on the previous studies.

(2) We can adjust the network input values to obtain the minimum or maximum output value by using this method. Therefore, the optimization method expands the application of the BP network because the method links the optimization and fitting of BP network together.

(3) This essay proposes a method of constrained optimization BP neural network based on previous studies. The method is based on the BP neural network, takes the minimum output of neural network as an example, gives the general mathematical models, and derives and gives the BP network's partial derivatives of output to input. The constrained optimization issue is discussed. The Sigmoid Function is used as the transmission function in the article. On this basis, the basic ideas, algorithms and related models are given, that are based on the constrained optimization issues of BP neural network.

(4)The method is an effective optimization method through the example of calculation.

Acknowledgment

This work is supported the National Natural Foundation of China (31071331), the national "863" high-tech Fund thematic (2006AA10A310-01).

References

[1] W.-P. Yu and Y.-G. Peng, “Intelligent control technology research”, Proceedings of The 8th Industrial Instrumentation and Automation Academic Conference, Shanghai, China, **(2007)**, pp. 415-418.
[2] W. Tang, “The research on BP neural network structure optimization issue”, Systems Engineering-Theory & Practice, vol. 10, **(2005)**.
[3] X. Shen; H. Zhang and J. Zhang, “BP neural network optimization by improved Particle Swarm Optimization algorithm”, Computer Systems & Applications, vol. 2, **(2010)**.
[4] M. Chen, J. Ni and K. Zha, “Genetic Neural Network Application in The Two-Dimensional Tidal Current Characteristics Simulation”, Journal of Water Conservancy, vol. 10, **(2003)**.
[5] L. Zhou, J. Sun and Y. Yuan, “Mixed activation function effect on the convergence rate of BP algorithm”, Journal of Hohai University, vol. 5, no. 27, **(1999)**.
[6] W. Tang, “The research of the BP neural network structure optimization”, Journal of Systems Engineering Theory and Practice, vol. 10, **(2005)**.
[7] Y. Cai and J. Zhang, “Industrial enterprise economic benefit evaluation based on the BP neural network optimization algorithm”, Statistics and Decision, vol. 5, **(2012)**.
[8] Y. Liang and S. Liu, “The research on BP neural network combination forecast model based on GA optimization”, Scientia Agricultural Sinica, vol. 12, **(2012)**.
[9] S. Zhang and J. He, “Optimized algorithm of BP neural network study”, Computer and Modernization, vol. 1, **(2009)**.
[10] X. Fu, F. Wang and J. Shang, “The research on BP neural network optimization based on genetic algorithm of offsprings”, Computer Simulation, vol. 03, **(2016)**.
[11] F. Wang, C. Wu and H. Yang, “The study on initial population generation method while constrained optimized problems was solved by genetic algorithm”, Journal of Northeast Agricultural University, vol. 10, **(2004)**.
[12] F. Wang, H. Zhu and J. Wang, “An improved evolution strategy of genetic algorithm”, Journal of Biomathematics, vol. 03, **(2015)**.
[13] J. Wang and F. Wang, “Total power of agricultural machinery based on BP neural network prediction”, Journal of Agricultural Machinery, vol. 12, **(2011)**.
[14] P. Zhou, “MATLAB neural network design and application”, Tsinghua University Publishers, Beijing, **(2013)**.

[15] W. Sun, "The Modified BFGS Method for The nonlinear unconstrained optimization problem", Journal of Chifeng Institute (Natural Science Edition), vol. 10, no. 30, **(2014)**.
[16] S. Jia, Y. Ou and G. Wang, "An ODE type hybrid method solving unconstrained optimization", Applied Mathematics, vol. 4, no 28, **(2015)**.
[17] F. Wang, G. Liu and J. Wang, "An unconstrainted optimization method based on BP neural network", 2010 International Conference on E-Product E-Service and E-Entertainment, **(2010)**.

Authors

Shuang-Jing Li (1978-), she is a Doctoral student, research direction: Systems Engineering and Management Engineering,

Fu-Lin Wang (1960-), he is a Doctor, Professor, Research Direction: Systems Engineering and Management Engineering,