The Enhancement Arithmetic of BP Neural Network Based on Target Optimizing

Shuang Shi[1,*], Daqi Zhang[2], Pengfei Feng[1] and Lingli Han[1]
[1]Anhui Sanlian University, Hefei, China
[2]Qinghai University, Xining, China
[*]Corresponding author

***Abstract*—BP neural network recognition algorithm was used to carry through target recognition, then we adopted quadratic programming and generalized least squares to optimize the network weights and thresholds, at last the algorithm formula was deduced. In the neural network training process, a tendentious training was achieved. Experiments show that the algorithm had a better effect.**

***Keywords*—BP neural network; quadratic programming; target optimizing**

I. INTRODUCTION

Target recognition is a hot problem in area of mode recognition. It is very important to recognize target in image for target recognition. There are a few representative theories, such as syntax recognition, artificial intelligence recognition, fitted recognition. Dellepiane[1] solved problem of invariability recognition with illegibility arithmetic, and Aguado[2] recognized multiform targets with Hough transform. They spent overabundance time as aims are excessive. Better effect can be obtained if using neural network for recognition. New BP ANN is a kind of mature and effective method. It adjust right and limen through Error Back Propagation.

II. BP NEURAL NETWORK ALGORITHM

BP algorithm is defined error back propagation algorithm of multilayer feedforward network. In the network learning process, the signal is constituted by forward and back propagation. As forward propagation, the input samples transmit from input layer passing through the hidden layer and disposing one by one, then turn into output layer. If the actual output of the output layer is different from the expectation, it will tend to error back propagation. Error back propagation transmit output error from hidden layer to output layer one by one as a certain form, and apportion the error to all the units of each layer in order to obtain unit error of each layer, which takes as the gist of revising every unit weight. This is the basic ideology of error back propagation algorithm, and the instance is shown in figure 1.

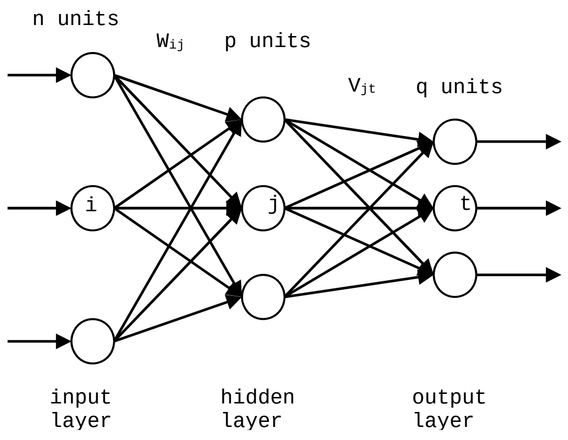

FIGURE I. THREE LAYERS OF BP NETWORK

III. QUADRATIC PROGRAMMING

In the constraint conditional extreme problems, the commonly used method is first transforming the original problem into easier subsystem problem, then solving it and used as the basis of an iterative process. The form of constrained optimization problem is in the following formulas.

$$\begin{cases} \min\ f(x); \\ s.t.\quad g_i(x) \geq 0, i = 1,2,\cdots,m, \\ \qquad h_j(x) = 0,\ j = 1,2,\cdots,l, \end{cases} \tag{1}$$

In the above formulas, $f : R^n \to R, g_i : R^n \to R(i = 1,2,\cdots,m), h_j : R^n \to R(j = 1,2,\cdots,l)$

Using Kuhn-Tucker (KT) equation can solve this kind of problems. KT equation is a necessary condition of conditional extreme problem. If the problem to be solved is convex programming problem (that is to say $f(x)$ and $G_j(x)$ both convex function), then KT equation will be the necessary and sufficient condition of conditional extreme problem.

IV. THE OPTIMIZATION OF GENERALIZED LEAST SQUARES METHOD

The standard form of the constrained linear generalized least squares is in the following formulas.

$$\min_x \quad \frac{1}{2}\| Cx - d \|_2^2$$

$$\text{sub.to} \quad A \cdot x \le b$$

$$Aeq \cdot x = beq$$

$$lb \le x \le ub$$

C, A and Aeq are matrixes, while d, b, beq, lb, ub and x are vectors.

Along with x closing to the optimum solution, the value of $\frac{1}{2}\| Cx - d \|_2^2$ will tend to be zero.

When $\frac{1}{2}\| C x - d \|_2^2$ quite small at the optimum solution, the search direction of Gauss-Newton can be as the optimum search direction, which is a more effective method.

V. The Enhancement Arithmetic Of BP Neural Network Based On Target Optimizing

A. *The Principle of Enhancement Arithmetic*

The enhancement arithmetic is generally considered collectivity in the network training process, but rarely about each step adjustment of the weights. The weights and thresholds can be interfered in the adjustment process. For example, when the weights and thresholds adjustment joined the constraint conditions, the adjusted value had a tendency to reflect the different impact factors that had different status in the network training.

1) The optimum enhancement arithmetic based on network weights of quadratic programming

a) The weight matrix V from hidden layer to output layer

To the number k learning mode, the network expectable output is d_t^k, while the actual output is y_t^k, so the deviation is $\delta_t^k = (d_t^k - y_t^k), t = 1,2,\cdots,q$.

S function is used as activation function. The relationship about actual output y_t^k, the connection weight V_{jt} from hidden layer to output layer, and the threshold γ_t of each output layer unit is shown in the following formula.

$$y_t = f(\sum_{j=1}^{p} v_{jt} b_j - r_t)$$

Quadratic programming optimization method is adopted, which makes y_t close to d_t, while δ_t close to zero.

Suppose $X_t = [v_{1t}, v_{2t}, \cdots, v_{pt}]^T, t = 1,2,\cdots,q$

$$B_k = [b_1^k, b_2^k, \cdots, b_p^k];$$

The above problem is transformed into a standard form of nonlinear constrained programming, which is shown in the following formulas.

$$\min_{X_t} \quad g(x_t) = [1/(1 + e^{-B_k \cdot X_t + r_t}) - d_t]^2$$

$$\text{sub.to}$$

$$A \cdot X_t \le b$$

$$Aeq \cdot X_t = beq$$

$$lb \le X_t \le ub$$

In the above formulas, r_t, b, beq, lb and ub are vectors, while A and Aeq are matrixes.

When t from 1 to q, solve the above problems that can totally obtain q vectors, then these q vectors constitute a weight matrix *V*.

b) The weight matrix W from input layer to hidden layer

To the number k learning mode according to, the correction error formula of each hidden layer unit is in the following.

$$e_j^k = -\left[\sum_{t=1}^{q} c_t^k \cdot v_{jt}\right] \cdot b_j^k \cdot (1 - b_j^k) \qquad (2)$$

A new weight matrix W is obtained to make e_j^k minimum. Here S function also is used as activation function. The relationship about input layer a_t^k, the connection weight W from hidden layer to output layer, and the threshold θ_t of each output layer unit is shown in the following formula.

$$b_j = f(S_j) = f(\sum_{i=1}^{n} w_{ij} a_i - \theta_j) \qquad j = 1,2,\cdots,p \quad (3)$$

Suppose $X_r = [w_{1r}, w_{2r}, \cdots, w_{nr}]^T, r = 1,2,\cdots,p$; $A_k = [a_1^k, a_2^k, \cdots, a_n^k]$, then take (3) into (2), so

$$e_r = -\left[\sum_{t=1}^{q} c_t^k \cdot v_{jt}\right] \cdot 1/(1 + e^{-A_k \cdot X_r + \theta_r}) \cdot [1 - 1/(1 + e^{-A_k \cdot X_r + \theta_r})] \quad (4)$$

Quadratic programming optimum method is adopted to make e_r close to minimum, then

$$\min_{X_r} \quad e_r$$

$$\text{sub.to}$$

$$A \cdot X_r \le b$$

$$Aeq \cdot X_r = beq$$

$$lb \le X_r \le ub$$

In the above formulas, θ_r, b, beq, lb and ub are vectors, while A and Aeq are matrixes.

When r from 1 to p, solve the above problems that can totally obtain p vectors, then these p vectors constitute a weight matrix W .

2) *The optimum enhancement arithmetic based on network threshold of generalized least squares.*

a) *The threshold vector γ from hidden layer to output layer*

S function is also used as activation function. Nonlinear generalized least squares method is adopted, which makes y_t close to d_t, while δ_t close to zero.

Suppose $X = \gamma = [\gamma_1, \gamma_2, \cdots, \gamma_t]^T, t = 1,2,\cdots,q$

$$x_1 = \gamma_1$$
$$x_2 = \gamma_2$$
$$\vdots$$
$$x_t = \gamma_t$$

The hidden layer output vector is $B = [b_1, b_2, \cdots, b_p]$;

The above problem is transformed into a standard form of nonlinear least squares programming, which is shown in the following formulas.

$$\min_X \quad g(X) = g_1(x_1)^2 + g_2(x_2)^2 + \cdots + g_t(x_t)^2, t = 1,2,\cdots,q$$

$$g_t(x_t) = 1/(1 + e^{\sum_{j=1}^{p} -b_j \cdot v_{jt} + x_t}) - d_t$$

sub.to

$$A \cdot X \le b$$

$$Aeq \cdot X = beq$$

$$lb \le X \le ub$$

In the above formulas, b, beq, lb and ub are vectors, while A and Aeq are matrixes.

The threshold vector γ can obtain by solving the above problems.

b) *The threshold vector θ from input layer to hidden layer*

S function is also used as activation function. Nonlinear generalized least squares method is adopted to make e_j^k minimum.

Suppose $X = \theta = [\theta_1, \theta_2, \cdots, \theta_t]^T, \quad t = 1,2,\cdots,p$

$$x_1 = \theta_1$$
$$x_2 = \theta_2$$
$$\vdots$$
$$x_t = \theta_t$$

It can obtain the below formula.

$$e_r = -\left[\sum_{t=1}^{q} c_t^k \cdot v_{jt}\right] \cdot 1/(1 + e^{\sum_{i=1}^{n} -a_i \cdot w_{ir} + \theta_r}) \cdot [1 - 1/(1 + e^{\sum_{i=1}^{n} -a_i \cdot w_{ir} + \theta_r})]$$

The above problem is transformed into a standard form of nonlinear least squares programming, which is shown in the following formulas.

$$\min_X \quad g(X) = g_1(x_1)^2 + g_2(x_2)^2 + \cdots + g_t(x_t)^2, \quad t = 1,2,\cdots,p$$
$$g_t(x_t) = e_t$$

sub.to

$$A \cdot X \le b$$

$$Aeq \cdot X = beq$$

$$lb \le X \le ub$$

In the above formulas, b, beq, lb and ub are vectors, while A and Aeq are matrixes.

The threshold vector θ can obtain by solving the above problems.

B. The Calculation Steps of Enhancement Arithmetic

According to these above analysis can obtain the enhancement BP network study process, and the specific steps are shown below.

(1) Initialization, then give a random value among [-1,+1] to connection weights W, V, and thresholds θ, γ.

(2) Randomly select modes of $A_k = [a_1^k, a_2^k, \cdots, a_n^k], D_k = [d_1^k, d_2^k, \cdots, d_q^k]$ to network training.

(3) The input mode $A_k = [a_1^k, a_2^k, \cdots, a_n^k]$, connection weight W and threshold θ calculate the input value sj (activation value) of each intermediate layer's neuronal, then make use of sj calculating the output bj of inter-hidden layers'

unit by activation function $f(x)=\frac{1}{1+e^{-x}}$. The output $b_j = f(s_j)$, in the formula $s_j = \sum_{i=1}^{n} w_{ij} \cdot a_i - \theta_j$.

(4) The output bj of inter-hidden layer, connection weight vjt and threshold calculate the input lt (activation value) of each output layer's unit, then make use of lt calculating the response yt of output layers' unit. The response $y_t = f(l_t)$, in the formula $l_t = \sum_{j=1}^{p} v_{jt} \cdot b_j - \gamma_t \qquad (t=1,2,\cdots,q)$.

(5) The expectable output mode $D_k = [d_1^k, d_2^k, \cdots, d_q^k]$ and network actual output yt calculate the deviation of output layers' unit.

$$\delta_t^k = (d_t^k - y_t^k) \qquad t=1,2,\cdots,q$$

(6) Use v_{jt}, c_t and b_j to calculate the correction errors e_j^k of intermediate layer.

$$e_j^k = [\sum_{t=1}^{q} c_t \cdot v_{jt}] b_j (1-b_j) \qquad (j=1,2,\cdots,p)$$

(7) Make sure the constraint conditions of new connection weights and thresholds between inter-hidden layer and output layer to calculate the new connection weights and thresholds.

(8) Randomly select next learning mode to provide network and return to step (3) until all the m modes are completed.

(9) Randomly select a new mode from m modes and return to step (3) until the network global error function E is less than the pre-set limit value(network can converge) or times of study are more than pre-set values(network cannot converge).

(10) The end of the study.

In the above BP network learning, steps (3)-(6) are forward propagation processes of input learning mode, step (7) is network weight and threshold value optimization process, steps (9) and (10) are completing the training and convergence processes.

VI. Experiment And Analysis

First train a network which has six units in input layer, three units in hidden layer, and one unit in output layer. The input of the input layer are [0.3, 0.7, 0.8, 0.2, 0.4, 0.5]T, and the expectable output is [0]. The initial weights from input layer to hidden layer are [0.2, 0.2, 0.1; -0.2, -0.3, -0.1; 0.3, 0.2, 0.2; -0.1, -0.1, -0.1; 0.1, 0.1, 0.1; -0.1, -0.1, -0.4]; the initial weights from hidden layer to output layer are [-0.3, 0.3, -0.4]T; the initial thresholds of hidden layer are [0.8, 0.7, 0.6]T; the initial threshold of output layer is [0.5]. If the six units of input layer in the network training have these weights [0.03, 0.07, 0.1, 0.15, 0.25, 0.4], and the three units of the hidden layer have these weights [0.1, 0.3, 0.6], when join the weight constraint to the network training and follow the step 6.6.2, finally will obtain the relationship between output error and training times which is shown in figure 2.

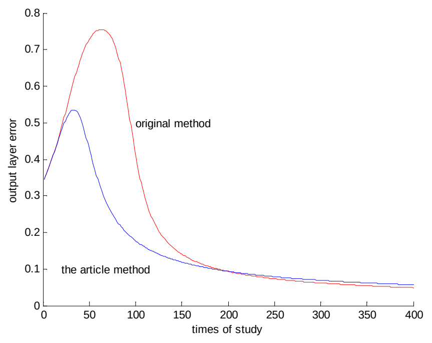

FIGURE II. THE CONTRAST OF METHOD 1

It is obvious that when join the constraint conditions to training, the overall network convergence has little change, however, it will have certain influence to convergent inflexion and convergent trend.

Set the output error 0.05 of the network training and train with the original method and the chapter method, which is used to do the following experiment. Change the network input, then observe the output effect of network. Set a unit input change $\Delta = 0.001$, the variation coefficient of input layer is i, so the variation of the input layer is $-i \cdot \Delta$. Change the first two input units of input layer, when i from 1 increase to 100, the error variation of output layer is shown in figure 3.

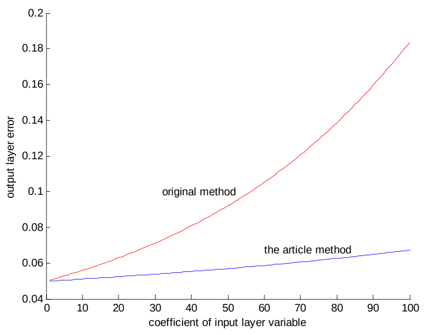

FIGURE III. THE CONTRAST OF METHOD 2

If the input layer changes the last two units, i also from 1 increase to 100, the error variation of the output layer is shown in figure 4.

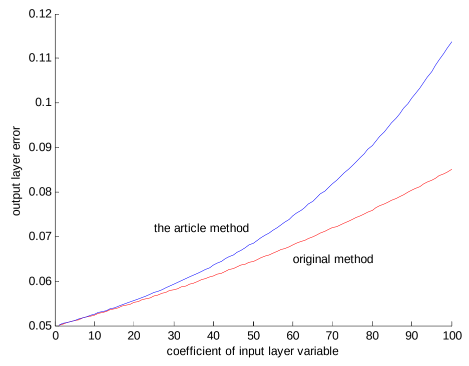

FIGURE IV. THE CONTRAST OF METHOD 3

It is obvious that the network training results are different when the input units have different weights. In the training process when the input changes, the one that has big weights will have a significant influence on network output, and the one that has small weights will have little influence on network output.

VII. CONCLUSIONS

According to the above analysis, it can be conclude that the enhancement arithmetic of BP neural network is a better method.

ACKNOWLEDGMENTS

This research is supported by the fund of Traffic Safety Application Technology Innovation Center of Anhui Sanlian University(No.YJT17002) and National Nature Science Foundation of China(No.51769027).

REFERENCES

[1] Yuan,H.Niemann,Neural networks for appearance-based 3-D object recognition[J], Neurocomputing, 2003, 51:249-264

[2] M.I.Miller,etal.Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition[J]. IEEE transactions on Signal Processing, 1995:vol.43(11), 2678-2690