Cable supported bridge damage location based on modified BP neural networks

Yang Jie [1,2] **and Li Aiqun** [1]

[1] *College of Civil Engineering, Key Laboratory of Concrete and Prestressed Concrete Structures of Ministy of Education, Southeast University, Nanjing, China*

[2] *Dept. of Civil Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China*

Email: yangjie@nuaa.edu.cn

ABSTRACT :

In the recent years, there has been increasing literatures focusing on the applications of ANNs in structural damage detection. Some researchers have done fundamental works separately in damage detection based on Back Propagation networks. However, adequate importance has not been attached to the some defects of BP neural networks. This paper integrates Bold Driver technology, momentum item, simulated annealing algorithm and stochastic hill-climbing algorithm to modify the traditional BP neural networks. The new modified BP networks have several advantages, such as quick convergence, escaping from local minimum and auto-optimizing network topology. Finally the developed method based on modified BP neural networks will be applied in Runyang Yangzi suspension bridge. The results show that the method can effectively extract hanger's damage pattern and effectively locate the hanger damage.

KEYWORDS: long span bridge, damage location, neural networks

1. INTRODUCTION

In the recent years, there has been increasing literatures focusing on the applications of ANNs in structural damage detection. Among them, Back Propagation networks are the mostly used networks. Some researchers, e.g. (Wu et al. 1992), (Elkordy et al. 1994), (Pandey 1995), (Hanagud and Luo 1997), (Yun and Bahng 2000), (Masri et al. 2000), (Lam et al. 2006), have done fundamental works separately.

A Back-Propagation neural network consists of one input layer, one or more hidden layers and one output layer. Every node in each layer is connected to every node in the adjacent layer. The trained network can serve as a nonlinear mapping function between the input set and the output set. The training process consists of two stages, feed-forward and back-propagation. In the feed-forward stage, the current layer output, which is a nonlinear function of summarized former layer's output, are multiplied with weights and transferred to the next layer. In back-propagation stage, the weights of cells are modified according to the error signals to minimize the target error. Back-propagation learning rules usually adopt the gradient descent arithmetic. The original and detail description refers to (Rumelhart et al. 1986).

However, there are some drawbacks limits its engineering application, e.g. slow convergence, local minima, bad extendibility, subjectivity in network architecture. This paper presents a modified BP learning algorithm, which integrates Bold Driver technology, adding momentum item, simulated annealing algorithm and stochastic hill-climbing algorithm to overcome the drawbacks of traditional BP networks.

2. MODIFIED BP LEARNING ALGORITHM

The learning algorithm includes two stages, shown in Figure 1. One is the forward calculation process, where the input data are multiplied by the weights and transferred to the hidden layer cells, then to the output layer cells. The other is the error Back-Propagation process, where the error is assigned to the cells and the weights are updated along the inverse route.

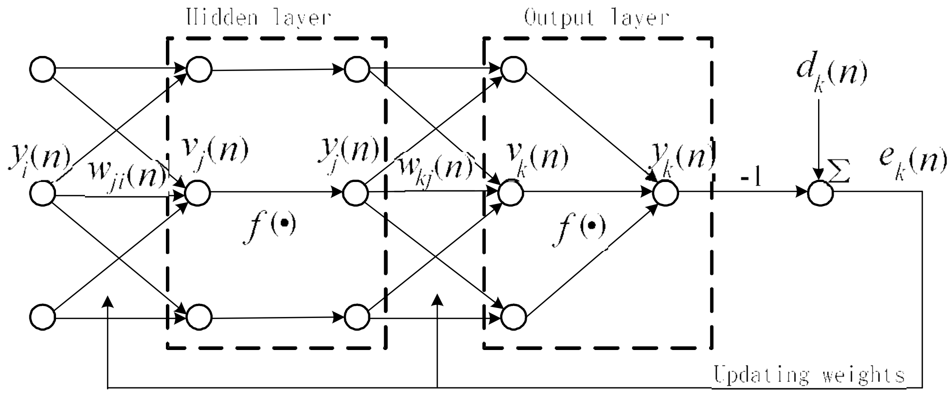

Figure 1. BP calculation process

In the forward process,

$$v_j(n) = \sum_i w_{ji}(n) y_i(n) \quad (1)$$

$$y_j(n) = f(v_j(n)) \quad (2)$$

$$v_k(n) = \sum_j w_{kj}(n) y_j(n) \quad (3)$$

$$y_k(n) = f(v_k(n)) \quad (4)$$

The error of the k-th cell is,

$$e_k(n) = d_k(n) - y_k(n) \quad (5)$$

So the global error energy function can be written as,

$$E(n) = \frac{1}{2}\sum_k [e_k^2(n)] \quad (6)$$

The learning objective can be set as minimizing the global error energy function. For output layer cells, the differential of energy function with respect to the weights can be deduced as follows.

$$\begin{aligned}\frac{\partial E(n)}{\partial w_{kj}(n)} &= \frac{\partial E(n)}{\partial e_k(n)}\frac{\partial e_k(n)}{\partial y_k(n)}\frac{\partial y_k(n)}{\partial v_k(n)}\frac{\partial v_k(n)}{\partial w_{kj}(n)} \\ &= -e_k(n)f^{'}(v_k(n))y_j(n)\end{aligned} \quad (7)$$

Taken,

$$\begin{aligned}\delta_k(n) &= -\frac{\partial E(n)}{\partial e_k(n)}\frac{\partial e_k(n)}{\partial y_k(n)}\frac{\partial y_k(n)}{\partial v_k(n)} \\ &= e_k(n)f^{'}(v_k(n))\end{aligned} \quad (8)$$

So,

$$\frac{\partial E(n)}{\partial w_{kj}(n)} = -\delta_k(n)y_j(n) \quad (9)$$

According to the gradient descend rule, the increment of weight can be written as,

$$\Delta w_{kj}(n) = \eta\delta_k(n)y_j(n) \quad (10)$$

where, η is the learning rate.

Then, the updated weights can be computed,

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n) = w_{kj}(n) + \eta\delta_k(n)y_j(n) \quad (11)$$

For hidden layer cells, similarly, the updated weights can be obtained,

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) = w_{ji}(n) + \eta\delta_j(n)y_i(n) \quad (12)$$

where,

$$\begin{aligned}\delta_j(n) &= f^{'}(v_j(n))\sum_k e_k(n)f^{'}(v_k(n))w_{kj}(n) \\ &= f^{'}(v_j(n))\sum_k \delta_k(n)w_{kj}(n)\end{aligned} \quad (13)$$

In formulae (11) and (12), the learning rate η is a constant value. A small η means slow learning and inefficient convergence, while a large η means quick learning and possible divergence. So a balance should be kept between efficiency and convergence. Here adopts the Bold Driver method to dynamically keep η in a rational area.

$$\eta(n+1) = \begin{cases}\alpha\eta(n) & (when\ \Delta E(n) < 0) \\ \beta\eta(n) & (when\ \Delta E(n) > 0)\end{cases} \quad (14)$$

where, α , β are adjusting coefficient, and $\alpha > 1, \beta < 1$.

There is plateau usually in the error energy surface, where the gradient $\frac{\partial E(w)}{\partial w}$ is too small to efficiently change the weights. A momentum factor θ ($0<\theta<1$) will be introduced into the weight updating formulae to consider the contribution of the last weights updating.

For hidden cells,

$$\Delta w_{ji}(n) = \eta\delta_j(n)y_i(n) + \theta\Delta w_{ji}(n-1) \quad (15)$$

For output cells,

$$\Delta w_{kj}(n) = \eta\delta_k(n)y_j(n) + \theta\Delta w_{kj}(n-1) \quad (16)$$

The weights updating algorithm based on the gradient decent rule is a global learning algorithm, which usually encounters the local minimum problem. Using Simulated Annealing algorithm (Ceranic 2001) can help the learning process to escape from the local minimum by the cell's stochastic transition. The pseudo codes of the algorithm are listed in the Figure 2.

```
for n←1 to N do
{
        T←T_0*EXP(-n)
        if T<T_end then return current_weights
        compute next_weights and ΔE
        if ΔE<0_end then current_weights←next_weights
        else
        {
                P←EXP(-ΔE/T)
                if rand(0,1)<P then
                {
                        current_weights←next_weights
                }
        }
}
return current_weights
```

Figure 2. Simulated Annealing in BP learning

```
while topset≠null do
{
        i←1
        net←topset[i]
        for j←1 to n do
        {
                stochastic weights
                train net
                if train_fail then
                {
                        cancel topset[i]
                        restart
                }
                compute valid_error[j]
        }
        average valid_error
        if valid_error<BestE then
        {
                BestE←valid_error
                Bestnet←net
        }
        cancel topset[i]
}
return Bestnet
```

Figure 3 Stochastic hill-climbing algorithm in topology optimization

The topology structure of BP networks consist of input layer, hidden layer and output layer, of which input layer and output layer can be determined by the problem to be solved. While how to fix the numbers of hidden layers and hidden cells remains a hot potato. Stochastic hill-climbing (Dunn 1998) is a heuristic search algorithm and is used to find optimal topology structure in this paper. The pseudo codes are listed in Figure 3.

3. DAMAGE LOCATION OF A LONG SPAN SUSPENSION BRIDGE

The Runyang Yangtse River Bridge consists of a suspension bridge and a cable-stayed bridge. The suspension bridge has a length of 1490m, and a width of 33.9m. The steel box girder is sliding-supported at the towers. The concrete tower has a height of 210.28m. Figure 4 shows the 3-d FEM model of the suspension bridge.

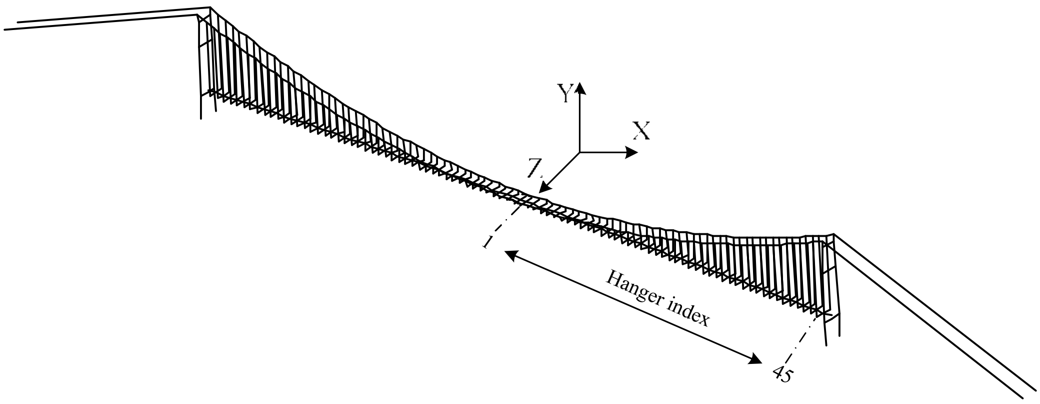

Figure 3. FEM model of Runyang suspension bridge

Taking the geometrical symmetry of the bridge into account, the 45 hangers at one quarter of the bridge are selected as damage elements. The damage scenarios are categorized in five groups according to the increasing damage extent, listed in Table 1.

Table 1. Damage scenarios

Damage group	Numbers of scenario	Damage extent（%）	Simulation method
Dg 1	45	5	$E=0.95E_0$
Dg 2	45	10	$E=0.90E_0$
Dg 3	45	15	$E=0.85E_0$
Dg 4	45	20	$E=0.80E_0$
Dg 5	45	25	$E=0.75E_0$

The normalized nature frequency of each damage scenario with respect to the intact bridge can be calculated. The 1~5 and 6~10 order normalized frequencies of bending mode of damage group Dg1 are plotted in Figure 4 and Figure 5. It's found that the damage pattern of each frequency is different. If we only focus on the first order frequency, then the influence of damage extent can be obtained, shown in Figure 6. It is found that damage extent doesn't alter the nature of the damage pattern curves and it only changes the amplitude of the curves in linear proportion.

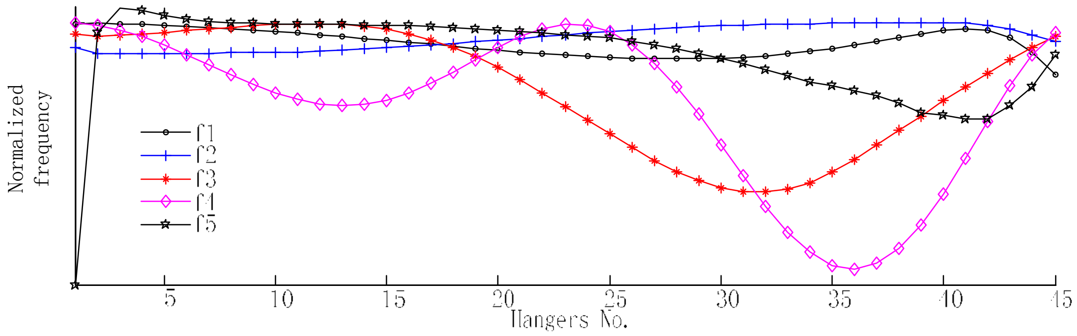

Figure 4. damage pattern of 1-5 frequencies

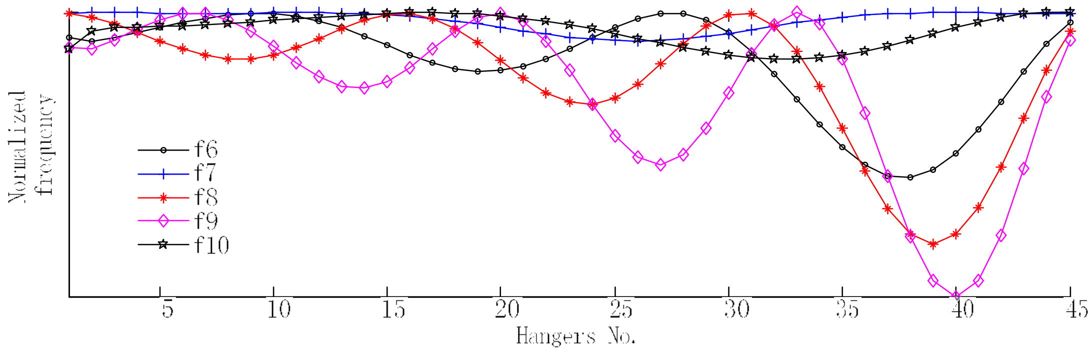

Figure 5. damage pattern of 6-10 frequencies

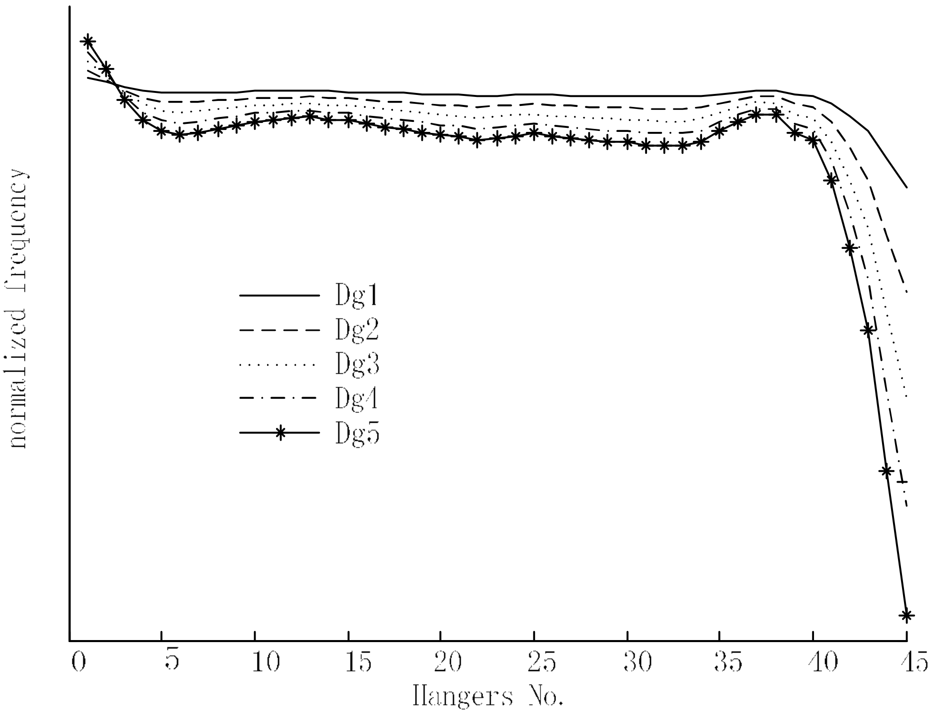

Figure 6. damage pattern of Dg1~Dg5

So the damage data of Dg1 and Dg5 are integrated as training data set, and damage data of Dg4 and Dg3 are chosen as validation set and test set separately. The network has 10 input cells and 45 output cells, and hidden cells are selected by stochastic hill-climbing algorithm. The optimal topology turns out to be one hidden layer with 17 hidden cells.

Based on the presented learning algorithm, the trained network has memorized the damage pattern about hanger location. For the 45 damage scenario in Dg 3, the trained network can output its identification results. The results, listed in Table 2, shows that the network makes 42 right decision, except for the hanger 8, 15 and 22, and the correct rate of identification is 93.3%.

Table 2. Identification results for Dg3

Location	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
results	√	√	√	√	√	√	√	×	√	√	√	√	√	√	×
Location	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
results	√	√	√	√	√	√	×	√	√	√	√	√	√	√	√
Location	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
results	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√

4. CONCLUSIONS

A modified BP learning algorithm, integrated with Bold Driver technology, momentum item, simulated annealing algorithm and stochastic hill-climbing algorithm, is presented. And the application in the hanger's damage location of a suspension bridge proves that the method can effectively extract hanger's damage pattern and effectively locate the hanger damage.

REFERENCES

Ceranic, B. (2001). An application of simulated annealing to the optimum design of reinforced concrete retaining structures. *Computers & Structures*, **79:17**, 1569-1581.

Dunn, S. A. (1998). The use of genetic algorithms and stochastic hill-climbing in dynamic finite element model identification. *Computers & Structures*, **66:4**, 489-497.

Elkordy, M. F., Chang, K. C., and Lee, G. C. (1994). Application of neural networks in vibrational signature analysis. *Journal of Engineering Mechanics*, **120:2**, 251-264.

Hanagud, S., and Luo, H. (1997). Damage detection and health monitoring based on structural dynamics. *Structural Health Monitoring*, 715-726.

Lam, H. F., Yuen, K. V., and Beck, J. L. (2006). Structural health monitoring via measured Ritz vectors utilizing artificial neural networks. *Computer-Aided Civil and Infrastructure Engineering*, **21:4**, 232-241.

Masri, S. F., Smyth, A. W., Chassiakos, A. G., Caughey, T. K., and Hunter, N. F. (2000). Application of neural networks for detection of changes in nonlinear systems. *Journal of Engineering Mechanics-Asce*, **126:7**, 666-676.

Pandey, P., Barai, SV. (1995). Multilayer perceptron in damage detection of bridge structures. *Computers & Structures*, **54:4**, 597-608.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, **323:6088**, 533-536.

Wu, X., Ghaboussi, J., and Garrett, J. H. (1992). Use of neural networks in detection of structural damage. *Computers & Structures*, **42:4**, 649-659.

Yun, C. B., and Bahng, E. Y. (2000). Substructural identification using neural networks. *Computers & Structures*, **77:1**, 41-52.