SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3 ISSUE 4
ISSN 2054-7390

Optimal Design of Continuous Reinforced Concrete Beams Using Neural Networks

[1]Jiin-Po Yeh and [2]Ren-Pei Yang
Department of Civil and Ecological Engineering, I-Shou University, Taiwan
[1]jpyeh@isu.edu.tw, [2]yunglnpei@gmail.com

ABSTRACT

This paper aims to build a neural network model to optimally design two-span continuous reinforced concrete beams. The training and checking data of the neural network are obtained by genetic algorithms, whose constraints are built according to the ACI Building Code and objective function is to find the minimum cost of longitudinal reinforcement, stirrups and concrete. The neural network adopted in this paper is the feed forward back propagation network, whose input vector consists of the span, width and effective depth of the beam, dead load, compressive strength of concrete as well as yield strength of steel and the output vector the positive and negative steel ratios and minimum total cost. The correlation coefficients between the target and network output of the testing data can reach as high as 0.9992, 0.9980 and 0.9999, respectively for the positive and negative steel ratios and minimum cost. Compared with the adaptive neuro-fuzzy inference system, the neural network shows almost the same accuracy but is much easier implemented.

Keywords: Continuous reinforced concrete beams, Genetic algorithms, Feedforward backpropagation networks, Correlation coefficients.

1 Introduction

Genetic algorithms are search procedures which mimic biological evolution. They can solve the optimization problems by the evolution theory of "survival of the fittest," whose constraints can be in the form of linear equality or inequality with bounds on the optimization variables. In 1970s, Professor John Holland first formally introduced its basic concept [1]. In 1989, Goldberg described in more detail the theory, terminology and its applications on machine learning and optimization [2]. From then on, the genetic algorithms became more attractive and widely used. Genetic algorithms have a variety of applications in fields like engineering, chemistry, economics, manufacturing and so on. Take civil engineering as an example. Many publications have been seen, such as multiobjective optimization of trusses [3], reliability analysis of structures [4], global optimization of grillages [5], global optimization of trusses with a modified genetic algorithm [6], optimization of pile groups using hybrid genetic algorithms [7], locating the critical slip surface in slope stability analyses [8-10], optimal design of reinforced concrete beams [11], etc.

Neural networks, as used in artificial intelligence, have traditionally been viewed as simplified models of neural processing in the brain. They are a computational tool based on the properties of biological neural systems, which are simplified models of neural processing in the brain and could capture and represent complex input/output relationships. The motivation for the development of neural network technology

DOI: 10.14738/tmlai.34.1303
Publication Date: 10th August, 2015
URL: http://dx.doi.org/10.14738/tmlai.34.1303

arose from the desire to have an artificial system to perform intelligent tasks similar to those done by humans. The artificial neural network was originated by McCulloch and Pitts in 1943 [12], which paved the way for neural network research. Rosenblatt [13] created the perceptron, an algorithm for pattern recognition based on a two-layer learning computer network using simple addition and subtraction. Also key later advance was the backpropagation algorithm by Werbos in 1975 [14]. In 1986, Rumelhart et al. [15] proposed the theory of parallel distributed processing that computed through the parallel cooperative and competitive interactions of a large number of simple neuron-like processing units in contrast to conventional programs that computed through the sequential application of stored commands. They developed the most famous learning algorithm to modify the weights on connections between these units so that the global error minimum could be achieved, which marked a milestone in the current artificial neural networks. Since then, a huge proliferation in the ANN methodologies and applications have been published, such as control of chaotic pendulum [16], flood forecasting [17], structural optimization [18-20], facilitating the accurate estimation of probabilistic constraints in optimization problem [21], frame optimization [22], traffic sign classification [23], modeling the financial market with multiple prices [24], etc.

2 Genetic Algorithms and Neural Networks

Because genetic algorithms could deal with high nonlinear constraints and neural networks could build complicated nonlinear relationships between inputs and outputs, this paper combines these two techniques to optimally design two-span continuous reinforced concrete beams and compares the results with the previous work [11], which uses the adaptive neuro-fuzzy inference system [25].

2.1 Genetic Algorithms

The genetic algorithm belongs to the field of artificial intelligence, which is a heuristic and multi-point search that imitates the course of natural selection. Without the need to find the derivatives of the objective function, it can solve optimization problems by the skills similar to natural evolution. To start the algorithm, the initial population of candidate solutions is generated at random across the search space, and then a sequence of new populations is produced. First, evaluate the fitness value of each member; Secondly, select members called parents according to their fitness, where fitter members are more likely to be chosen; Thirdly, retain a specified number of individuals, called elites, which have fittest values. They will pass to the next population unchanged; Fourthly, produce children by combining the vector entries of a pair of parents, i.e., crossover, where offspring under “crossover” will combine parental traits and will not be identical to its parents; Fifthly, make random changes to a single parent, i.e., mutation, which is more like walking randomly in the vicinity of a candidate solution; Finally, replace the current population with the crossover and mutation children and elites to form the next generation. The above steps are repeated until a terminating condition is met.

The optimization problem for two-span continuous reinforced concrete beams discussed in this paper can be formulated as follows:

Minimize $f(\boldsymbol{x})$

to find x

subject to

$$C_i(\boldsymbol{x}) \le 0,\ i=1,\ldots, m$$
$$C_j(\boldsymbol{x}) = 0,\ j=1,\ldots, n \quad (1)$$
$$\boldsymbol{LB} \le \boldsymbol{x} \le \boldsymbol{UB}$$

where $\boldsymbol{x}$ is the vector of design variables, $C_i(\boldsymbol{x})$ and $C_j(\boldsymbol{x})$ represents the nonlinear inequality and equality constraints, respectively, $f(\boldsymbol{x})$ is the fitness function, which is the total cost of concrete, longitudinal reinforcement and stirrups, and ***LB*** and ***UB*** are the vectors of lower and upper bounds of design variables, respectively.

2.2 Artificial Neural Networks

The neural network used in this paper is the two-layer feedforward backpropagation network, whose structure is shown in Figure 1. The transfer function of the single hidden layer is the tan-sigmoid function

$$a_i = f(net_i) = \frac{2}{1+e^{-2net_i}} - 1 \quad , \quad i = 1,2,3,\ldots,k \quad (2)$$

where k is the number of the neurons in the hidden layer, $net_i = w_{i1}P_1 + w_{i2}P_2 + \ldots + w_{iR}P_R + b_i$, $P_1, P_2, \ldots P_R$ are the inputs, R is the number of elements in the input veator, $w_{i1}, w_{i2}, \ldots, w_{iR}$ are the weights connecting the input vector and the ith neuron, and b_i is the bias of the ith neuron in the hidden layer. The output layer uses the linear transfer function

$$O_i = f(Net_i) = Net_i \quad , \quad i = 1, 2, \ldots, q \quad (3)$$

where $Net_i = W_{i1}a_1 + W_{i2}a_2 + \ldots\ldots + W_{ik}a_k + B_i$, $W_{i1}, W_{i2}, \ldots, W_{ik}$ are the weights connecting the neurons in the hidden layer and the ith neuron of the output layer, B_i is the bias of the ith output neuron and q is the number of the network outputs. The square error between the network outputs and the targets can be expressed as

$$E_P = \sum_{j=1}^{q} (t_j - a_j)^2 \quad (4)$$

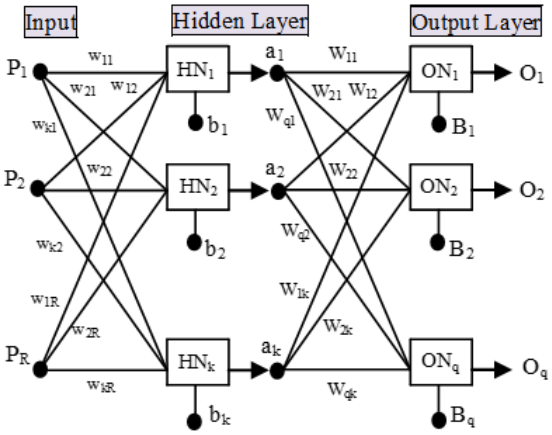

Figure 1 The structure of two-layer feed forward back propagation neural networks

where t_j and a_j are the jth target and network output, respectively, and p is a particular input-output pair. For the whole training set, the mean square error is given by

$$MSE = \frac{1}{m}\sum_{P=1}^{m} E_P \tag{5}$$

where *m* is the total number of training pairs. The Levenberg-Marquardt algorithm [26-28] is chosen as the training function to minimize the mean square error (i.e., the network performance function). If a tentative step increases the performance function, this algorithm will act like the gradient descent method, while it shifts toward Newton's method if the reduction of the performance function is successful. It interpolates between the quasi-Newton's algorithm and the gradient descent method; therefore, the mean square error in Eq. (5) will always be reduced at each iteration.

In order to generalize the network, the validation set is also presented to the network during the training process. When the network begins to overfit the training data, the error on the validation set typically begins to rise. Once the validation error increases for a specified number of iterations, e.g., 6 iterations, the training terminates and the weights and biases corresponding to the minimum of validation error are returned.

3 Design of Two-Span Beams and the Constraints

Based on the provisions of the ACI Building Code Requirements for Structural Concrete and Commentary [29], the constraints for the two-span continuous beam are formulated, taking into account the strength requirements of the maximum positive and negative moments, shear, the service requirement of deflection as well as the development length of flexural reinforcement. Design variables are the width b, effective depth d, positive steel ratio ρ1 and negative steel ratio ρ2 of two-span continuous reinforced concrete beams. The beams considered in this paper are subjected to a uniformly distributed load wu =1.2 wD+1.6 wL , as shown in Figure 2, where wD and wL are dead load and live load. The beams with span length L are singly reinforced and vertical stirrups are used. Top reinforcement in the negative moment region will be cut off, while there are no cutoffs for the bottom reinforcement to simplify the design, as shown in Figure 3. Force and length are measured in the units of kgf (=9.81N) and cm, respectively, for the following formulas.

3.1 Shear

Suppose that V_c is the shear capacity of the plain web concrete and V_u is the factored shear force. The shear diagram of the two-span continuous beam is shown in Figure 2, where V_A=0.375w_u L=V_E and V_C^+ =0.625w_u L=V_C^-. First, consider the segment AB in Figure 2. The design of web steel for shear may be considered by dividing the shear diagram into four regions: (1) Region I: If $V_u \le 0.5\phi V_c$, where ϕ=0.75 is the strength reduction factor, there is no need for shear reinforcement; (2) Region II: If $\phi V_c \ge V_u > 0.5\phi V_c$, a minimum web steel area

$$A_v \ge Max(0.2\sqrt{f_c'}\frac{bs}{f_y},\ \frac{3.5bs}{f_y}) \tag{6}$$

is required, where s is the spacing of vertical stirrups, fc′ is the compressive strength of concrete and fy is the yield strength of steel. According to the ACI code, the spacing s must not be larger than Min (d/2, 60) cm. Because the size of stirrups is usually fixed along the span of the beam, the spacing

$$s \le \text{Min}\ (\frac{d}{2},\ 60,\ \frac{A_v f_y}{0.2\sqrt{f_c'}b},\ \frac{A_v f_y}{3.5b}) \tag{7}$$

must be satisfied by combining Eq. (6) and the requirement of *Min* (*d*/2, 60) cm; (3) Region III: If $3\phi V_c \ge V_u > \phi V_c$, web reinforcement

$$V_s = \frac{A_v f_y d}{s} = \frac{V_u}{\phi} - V_c \tag{8}$$

has to be provided to carry the difference and the spacing

$$s = \frac{\phi A_v f_y d}{V_u - \phi V_c} \tag{9}$$

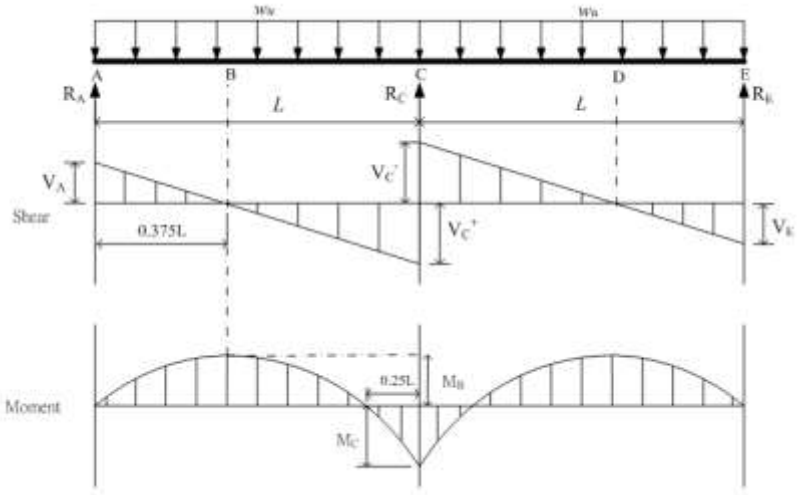

Figure 2 Two-span continuous reinforced concrete beam

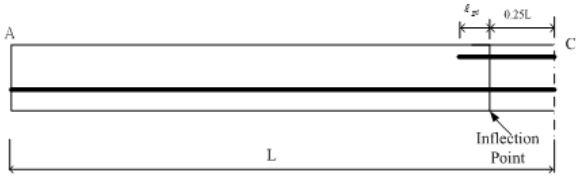

Figure 3 The top and bottom reinforcement in the left span

must not be larger than *Min* (*d*/2, 60, $\frac{A_v f_y}{0.2\sqrt{f_c'}b}$, $\frac{A_v f_y}{3.5b}$) cm; (4) Region IV: If $5\phi V_c \ge V_u > 3\phi V_c$, the web reinforcement in Eq. (8) similarly has to be provided to carry the difference, but the spacing *s* in Eq. (9) must not be larger than Min (d/4, 30, $\frac{A_v f_y}{0.2\sqrt{f_c'}b}$, $\frac{A_v f_y}{3.5b}$) cm.

Web reinforcement for segments BC, CD and DE shown in Figure 2 will be arranged in the same way as the segment AB. Because the reaction in the direction of applied shear introduces compression into the

end regions of a member, the critical section is taken at a distance *d* from the face of support. If the factored shear force V_{ud} at a distance *d* from the face of the support is larger than $5\phi V_c$, the beam section has to be enlarged. Therefore, another constraint is given by

$$V_{ud} \le 5\phi V_c \tag{10}$$

After the spacing for each region is found, the total number of vertical stirrups can be computed right away.

3.2 Bending Moment

The moment diagram can be seen in Figure 2. The maximum positive moment is $M_B = 9w_uL^2/128$ located at 0.375*L* from *A* (or *E*) and the negative moment at the support *C* is $M_c=0.125w_uL^2$. For simplicity, the strain ε_t in the tension reinforcement is assumed to be equal to 0.005; therefore, the section is tension-controlled and the strength reduction factor for moment is fixed at 0.9, not a function of strain in the tension reinforcement any more. The constraint for both positive and negative moment then takes the form

$$M_u \le 0.9M_{n,0.005} \tag{11}$$

where M_u is the factored negative moment M_C or the factored maximum positive moment M_B, and

$$M_{n,0.005} = A_s f_y \left(d - \frac{1}{2} \times \frac{A_s f_y}{0.85 f_c' b} \right) \tag{12}$$

where the area of reinforcement

$$A_s = \frac{0.85 f_c' \beta_1}{f_y} \times \frac{3db}{8} \tag{13}$$

due to the net tensile strain of 0.005 in the extreme tensile reinforcement, and β_1 is the stress block depth factor. To prevent sudden failure with little or no warning when the beam cracks or fails in a brittle manner, the ACI code limits the reinforcement ratio to be between

$$\rho_{max} = \frac{0.85 f_c' \beta_1}{f_y} \left(\frac{3}{7} \right) \tag{14}$$

and

$$\rho_{min} = max \left(\frac{0.8\sqrt{f_c'}}{f_y}, \frac{14}{f_y} \right) \tag{15}$$

where ρ_{max} in Eq. (14) is to make sure that the tensile strain must be greater than or equal to 0.004. Therefore, the constraint for reinforcement ratio is given by

$$\rho_{min} \le \rho \le \rho_{max} \tag{16}$$

where ρ is the steel ratio ρ_1 for positive moment M_B or the steel ratio ρ_2 for negative moment M_c.

3.3 Development of Reinforcement

The ACI Code stipulates that at least one-third of the total tension reinforcement provided for negative bending moment at the support should extend beyond the inflection point (P.I) not less than the effective depth d of the member, $12d_b$, or 1/16 of the clear span, as shown in Figure 3. The inflection point P.I. is located 0.25L from the support. For practical purposes, let span length $L \approx$ clear spam. Therefore, the length of top reinforcement in Figure 3

$$\ell_{top} = 0.25L + \ell_{pi} = 0.25L + Max\ (12d_b, d, \frac{L}{16}) \geq \ell_d \tag{17}$$

where ℓ_d is development length of tension reinforcement and d_b is the nominal diameter of the horizontal longitudinal bar.

3.4 Immediate and Long-Term Deflections

Because the deflection of the beam can be magnified by creep and shrinkage, both immediate and long-term deflection must be considered according to the ACI code. The creep and shrinkage deflection under sustained load can be evaluated using a multiplying factor

$$\lambda = \frac{\xi}{1+50\rho'} \tag{18}$$

where ρ' is the compression reinforcement ratio at midspan for simple and continuous beams and ξ is a time factor that is taken as 1.0 for loading time duration of 3 months, 1.2 for 6 months, 1.4 for 12months and 2.0 for 5 years or more, respectively. Suppose that the beam considered in this paper will support partitions and other construction likely to be damaged by large deflections. Hence, the long-term deflection

$$\Delta_{LT} = \Delta_{iL} + \lambda\Delta_{iD} \leq \frac{L}{480} \tag{19}$$

where Δ_{iL} is immediate live-load deflection and Δ_{iD} is immediate dead-load deflection under service loading.

4 Numerical Results

Given the span length, uniformly distributed dead and live loads, compressive strength of concrete and yield strength of steel of the two-span continuous singly reinforced concrete beams with rectangular cross-section, the optimal design is accomplished by using the genetic algorithm, whose design variables are the width b, effective depth d, positive steel ratio ρ_1 and negative steel ratio ρ_2, and the objective function is to find the minimum cost of concrete, longitudinal reinforcement and stirrups in New Taiwan Dollars. The unit prices are 1800 NT\$/m^3 and 19.5 NT\$/kgf for concrete and steel, respectively, in Taiwan. The concrete cover for the reinforcement is 4 cm and No. 3 vertical stirrups are used. On the basis of materials often used in Taiwan, this paper uses three kinds of yield strength f_y of the tension reinforcement: 2800 kgf/cm^2 (40 ksi), 3500 kgf/cm^2 (50 ksi) and 4200 kgf/cm^2 (60 ksi) as well as three kinds of compressive strength f'_c of the concrete: 210 kgf/cm^2 (3000 psi), 280 kgf/cm^2 (4000 psi) and 350

kgf/cm^2 (5000 psi). Besides, three kinds of span length: 6 m, 8 m and 10 m and four kinds of uniformly distributed dead load: 2100 kgf/ cm, 2300 kgf/cm, 2500 kgf/cm and 2700kgf/cm are adopted; uniformly distributed live load is fixed at 1800 kgf/cm. Hence, there are totally 108 cases to be designed, which will be used as training, validation and testing data for the neural networks.

4.1 Genetic Algorithms

The Global Optimization Toolbox based on MATLAB [30] is employed to execute the genetic algorithm, where some parameters are specified as follows: the population size 20, crossover rate 0.8, and elite number 2. In addition, all the individuals are real-number codes; "Rank" is taken as the scaling function to scale the fitness values; "Roulette" is the selection function to choose parents for crossover; "Two-point crossover" is the strategy to produce offspring; The mutation function "Adaptive Feasible Function" is applied to avoid being trapped into the local minimum. The genetic algorithm is executed 30 times for each case, from which the minimum cost is singled out. Then, the total 108 optimal sets of data are randomly divided into 3 groups: 64 training sets (60 %), 22 validation sets (20 %) and 22 testing sets (20 %).

4.2 Feedforward Backpropagation Neural Networks

This paper applies the Neural Network Toolbox based on MATLAB [31] to build the neural network, whose inputs of the neural network consist of six elements: f_y, f'_c, w_D, L, b as well as d, and targets have three components: the minimum cost, the steel ratios ρ_1 and ρ_2. Because the input vector has six elements, the number of neurons in the hidden layer is first set to be six by the past experience. The Levenberg-Marquardt algorithm is chosen to train the neural networks. The training process is shown in Figure 4. The weights and biases at epoch 64 are returned to the trained network. After the training of the network is completed, the testing set is then used to examine the network performance. The graphs of the network outputs and targets are plotted in Figs. 5-7. The inputs, targets and network outputs of the 22 sets of the testing data are listed Table 1. To further evaluate the performance of the trained network, this paper makes use of a linear regression analysis between the network outputs and the corresponding targets. The scatter plots are shown in Figs. 8-10. The regression results of the steel ratios $\rho 1$ and $\rho 2$ and the minimum cost (103 NT$) are shown in Table 2, where the symbols m, b and r stand for the slope, the y-intercept and correlation coefficient, respectively. The correlation coefficients between the network output and target are 0.9992, 0.9980 and 0.9999, respectively for the positive and negative steel ratios and minimum cost. Besides, the slope m is close to 1 and y-intercept b approximately equals 0. Based on Figs. 5-10 and Table 2, the performance of the feedforward backpropagation networks can be considered excellent, which is as good as the adaptive neuro-fuzzy inference system. The networks with 12, 18 and 24 neurons in the hidden layer are also explored for comparison with 6 neurons. Their linear regression results can be seen in Table 2, which indicates that even the number of neurons used in the hidden layer increases, the network output accuracy doesn't improve significantly.

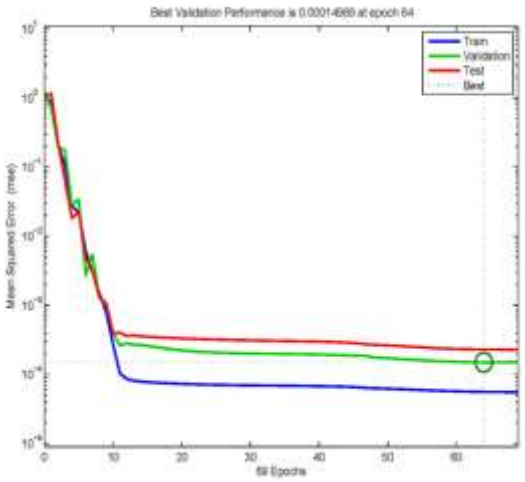

Figure 4 The training process for the neural network with 6 neurons in the hidden layer

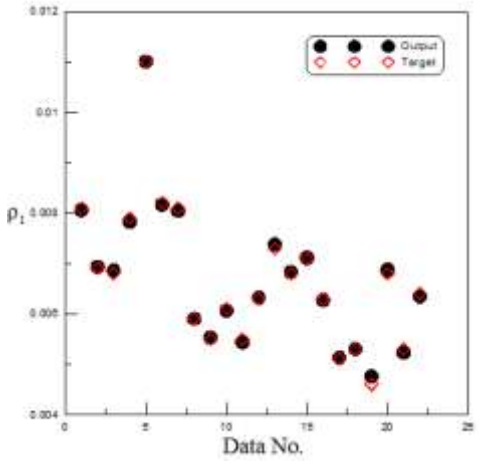

Figure 5 The network outputs and targets of the steel ratio of the positive moment for the testing sets with 6 neurons in the hidden layer

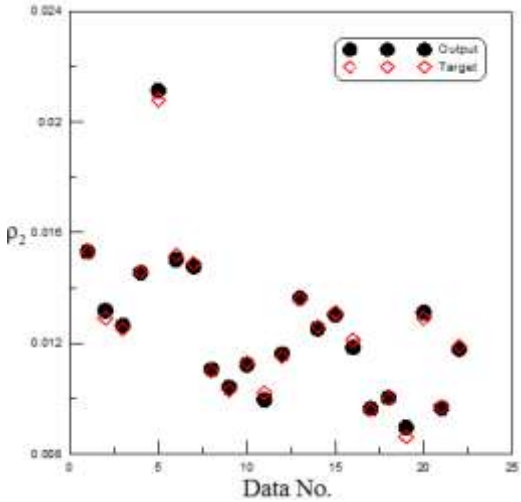

Figure 6 The network outputs and targets of the steel ratio of the negative moment for the testing sets with 6 neurons in the hidden layer

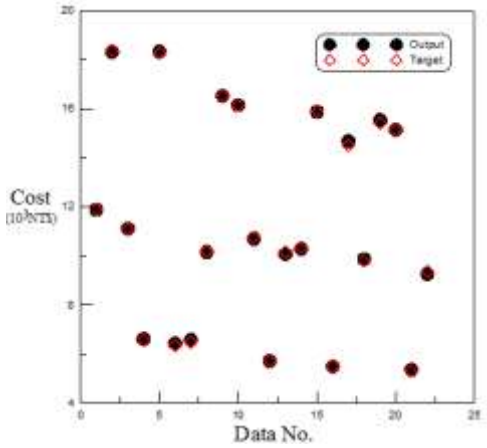

Figure 7 The network outputs and targets of the minimum cost for the testing sets with 6 neurons in the hidden layer

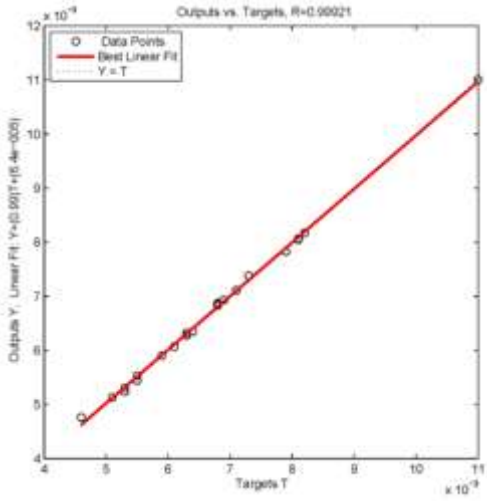

Figure 8 The scatter plot of the steel ratio of positive moment for the testing set with 6 neurons in the hidden layer

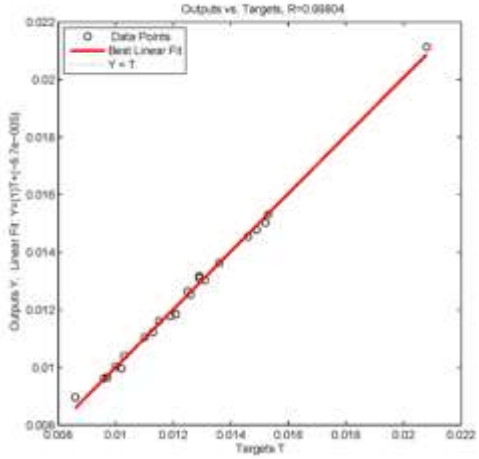

Figure 9 The scatter plot of the steel ratio of negative moment for the testing set with 6 neurons in the hidden layer

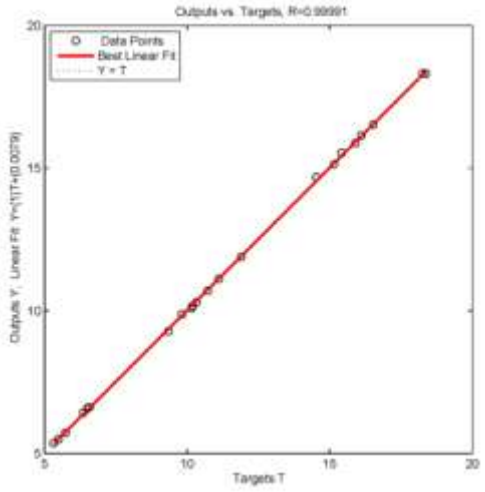

Figure 10 The scatter plot of the minimum cost (103 NT$) for the testing set with 6 neurons in the hidden layer

Table 1 Optimal results and outputs of the 22 sets of testing data for the neural network with 6 neurons in the hidden layer

Inputs and Targets of the network									Network Outputs		
Inputs						Targets					
f_r (ton/cm²)	f'_c (ton/cm²)	w_e (ton/m)	L (m)	b (m)	d (m)	ρ_1	ρ_2	Cost (10^3NT$)	ρ_1	ρ_2	Cost (10^3NT$)
2.8	0.21	2.5	8	0.2005	0.8312	0.0081	0.0153	11.894	0.0081	0.0153	11.885
2.8	0.21	2.5	10	0.2069	1.1012	0.0069	0.0129	18.376	0.0069	0.0132	18.311
2.8	0.28	2.1	8	0.2037	0.8513	0.0068	0.0125	11.109	0.0069	0.0126	11.114
2.8	0.28	2.3	6	0.2000	0.6147	0.0079	0.0146	6.5762	0.0078	0.0146	6.609
2.8	0.28	2.3	10	0.2021	0.8703	0.011	0.0208	18.26	0.0110	0.0211	18.329
2.8	0.35	2.1	6	0.2001	0.5848	0.0082	0.0152	6.3606	0.0082	0.0150	6.429
2.8	0.35	2.3	6	0.2001	0.6024	0.0081	0.0149	6.4939	0.0080	0.0148	6.580
3.5	0.21	2.1	8	0.2000	0.8349	0.0059	0.0110	10.189	0.0059	0.0110	10.165
3.5	0.21	2.5	10	0.2063	1.1032	0.0055	0.0103	16.532	0.0055	0.0104	16.526
3.5	0.28	2.5	10	0.2034	1.0536	0.0061	0.0113	16.109	0.0061	0.0112	16.146
3.5	0.28	2.7	8	0.2008	0.9053	0.0055	0.0102	10.724	0.0054	0.0100	10.697
3.5	0.35	2.1	6	0.2000	0.5986	0.0063	0.0115	5.7371	0.0063	0.0116	5.722
3.5	0.35	2.3	8	0.2002	0.7567	0.0073	0.0136	10.159	0.0074	0.0136	10.083
3.5	0.35	2.5	8	0.2000	0.8003	0.0068	0.0126	10.32	0.0068	0.0125	10.291
3.5	0.35	2.5	10	0.2013	0.9775	0.0071	0.0131	15.901	0.0071	0.0130	15.874
4.2	0.21	2.1	6	0.2001	0.5570	0.0063	0.0121	5.4984	0.0063	0.0118	5.497
4.2	0.21	2.1	10	0.2036	1.0145	0.0051	0.0096	14.527	0.0051	0.0096	14.681
4.2	0.21	2.5	8	0.2000	0.8393	0.0053	0.0100	9.8189	0.0053	0.0100	9.870
4.2	0.21	2.7	10	0.2004	1.1405	0.0046	0.0086	15.417	0.0048	0.0090	15.525
4.2	0.28	2.7	10	0.2006	0.9408	0.0068	0.0129	15.152	0.0069	0.0131	15.139
4.2	0.35	2.1	6	0.2000	0.5978	0.0053	0.0097	5.3274	0.0052	0.0097	5.360
4.2	0.35	2.3	8	0.2000	0.7411	0.0064	0.0119	9.3465	0.0063	0.0118	9.273

Table 2 The regression analysis of the targets with network outputs of the testing data for different number of neurons in the hidden layer

No. of neurons in the hidden layer	Targets vs. network outputs	*m*	*b*	*r*
6	ρ_1	0.9910	0.0001	0.9992
	ρ_2	1.0071	-0.0001	0.9980
	Cost	1.0004	0.0079	0.9999
12	ρ_1	1.0036	0.0000	0.9993
	ρ_2	1.0023	0.0000	0.9982
	Cost	0.9986	0.0351	0.9998
18	ρ_1	1.0185	-0.0001	0.9952
	ρ_2	1.0119	-0.0002	0.9977
	Cost	0.9709	0.2512	0.9972
24	ρ_1	1.0241	-0.0002	0.9981
	ρ_2	1.0341	-0.0005	0.9959
	Cost	0.9962	0.0046	0.9995

5 Conclusions

This paper uses the feedforward backpropagation neural network to form a design model for two-span continuous reinforced concrete beams with rectangular cross-section and compares the results with the adaptive neuro-fuzzy inference system. The training, validation and testing data are obtained from the optimal results of the genetic algorithm. The inputs of these models are the yield strength of steel, compressive strength of concrete, dead load (live load is fixed) and span length, width and effective depth of the beam, while the targets are the minimum cost, the steel ratios for the positive and negative moment. The reason for the inputs of the neural networks to be a little bit different from the given conditions of the genetic algorithm is to make the design model more practical. Numerical results show that the performance of the feedforward backpropagation neural network is excellent with all correlation coefficients being greater than 0.998, the slope and y-intercept of the regression line close to 1 and 0, respectively, which is as good as the adaptive neuro-fuzzy inference system. Furthermore, if more neurons in the hidden layer are used, the effectiveness of the neural network doesn't improve significantly. In addition to having the same accuracy, the feedforward backpropagation neural network is much easier to be implemented than the adaptive neuro-fuzzy inference system.

REFERENCES

[1] Holland, J. H. 1975. Adaptation in natural and artificial systems. The University of Michigan Press, Ann Arbor, MI, USA.

[2] Goldberg, D. E. 1989. Genetic algorithms in search, optimization and machine learning, Addison Wesley, Reading, MA, USA.

[3] Coello, C. A. and Christiansen, A. D. 2000. Multiobjective optimization of trusses using genetic algorithms, Computers and Structures. Vol. 75, pp. 647-660.

[4] Cheng, J and Li, Q. S. 2008. Reliability analysis of structures using artificial neural network based genetic algorithms. Computer Methods in Applied Mechanics and Engineering, Vol. 197, No. 45. pp. 3742-3750.

[5] Belevičius, R. and Šešok, D. 2008. Global optimization of grillages using genetic algorithms, Mechanika. Nr., Vol. 6, No. 74, pp. 38-44.

[6] Šešok, D. and Belevičius, R. 2008. Global optimization of trusses with a modified genetic algorithm, Journal of Civil Engineering Management. Vol. 14, No. 3, pp. 147-154.

[7] Chan, C. M., Zhang, L. M. and Jenny, T. N. 2009. Optimization of pile groups using hybrid genetic algorithms, Journal of Geotechnical and Geoenvironmental Engineering. Vol. 135, Issue 4, pp. 497-505.

[8] Goh, A. T. C. 2000. Search for critical slip circle using genetic algorithms. Civil Engineering and Environmental Systems, Vol. 17, No. 3, pp. 181–211.

[9] Zolfaghari, A.R., Heath, A.C., McCombie, P.F. 2005. Simple genetic algorithm search for critical non-circular failure surface in slope stability analysis. Computer and Geotechnics, Vol. 32, No. 3, pp. 139-152.

[10] Li, Y.-C., Chen, Y.-M., Zhan, T. L. T., Ling, D.-S. and Cleall, P. J. 2010. An efficient approach for locating the critical slip surface in slope stability analyses using a real-coded genetic algorithm. Canadian Geotechnical Journal, Vol. 47, No. 7, pp. 806–820.

[11] Yeh, J-P and Yang R-P, 2014, “Application of the Adaptive Neuro-Fuzzy Inference System for Optimal Design of Reinforced Beams,” Journal of Intelligent Learning Systems and Applications, Vo. 6, pp. 162-175.

[12] McCulloch, W.S. and Pitts, W. 1943. A logical calculus of ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, Vol. 5, pp. 115-133.

[13] Rosenblatt, F. 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. Psychological Review, Vol. 65, No.6, pp. 386–408.

[14] Werbos, P. J. 1975. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Dissertation, Harvard University, Boston, MA, USA.

[15] Rumelhart, D. E., McClelland, J. L. and the PDP Research Group. 1986. Parallel distributed processing: explorations in the microstructure of cognition. volume 1: foundations, MIT Press, Cambridge, MA, USA.

[16] Bakker, R., Schouten, J. C., Takens, F. and van den Bleek, C. M. 1996. Neural network model to control an experimental chaotic pendulum. Physical Review E, 54A, pp. 3545-3552.

[17] Mukerji, A., Chatterjee, C., and Raghuwanshi, N. S. 2009. Flood forecasting using ANN, Neuro-Fuzzy and Neuro-GA models. Journal of Hydrologic Engineering, Vol. 14, No. 6, pp. 647-652.

[18] Cheng, J. and Li, Q. S. 2009. A hybrid artificial neural network method with uniform design for structural optimization. Computational Mechanics, Vol. 44, No. 1, pp. 61-71.

[19] Möller, O., Foschi, R. O., Quiroz, L. M., and Rubinstein, M. 2009. Structural optimization for performance-based design in earthquake engineering: applications of neural networks. Structural Safety, Vol. 31, No. 6, pp. 490–499.

[20] Gholizadeh, S. and Salajegheh, E. 2010. Optimal design of structures for earthquake loading by self organizing radial basis function neural networks. Advances in Structural Engineering, Vol. 13, No. 2, pp. 339-356.

[21] Patel, J. and Choi, S. K. 2012. Classification approach for reliability-based topology optimization using probabilistic neural networks. Structural and Multidisciplinary Optimization, Vol.45, Issue 4, pp. 529–543.

[22] Meon, M. S., Anuar, M. A., Ramli, M. H. M., Kuntjoro, W., and Muhammad, Z. 2012. Frame optimization using neural network. International Journal Advanced Science Engineering Information Technology, Vol. 2, No. 1, pp. 28-33.

[23] Ciresan, D. C., Meier, U., Masci, J. and Schmidhuber, J. 2012. Multi-column deep neural network for traffic sign classification. Neural Networks, Vol. 32, pp. 333-338.

[24] Danting, Z., Tingfeng, T., Zhongwei, J., Huichao, X., Xian, L. and Qing-Guo, W., 2014. Modeling the financial market with multiple prices. Transactions on Machine Learning and Artificial Intelligence, Vol. 2, No. 5, pp. 41-51.

[25] Jang, J.-S. R. 1993. “ANFIS: Adaptive-Network-Based Fuzzy Inference System”, IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, No.3, pp.665-685.

[26] Hagan, M. T. and Menhaj, M. 1994. Training feedforward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks, Vol.5, No.6, pp.989-993.

[27] Hagan, M. T., Demuth, H. B. and Beale, M. H. 1996. Neural network design. PWS Publishing, Boston, MA, USA.

[28] Pujol, J. 2007. The solution of nonlinear inverse problems and the Levenberg-Marquardt method. Geophysics, Vol. 72, No. 4, W1–W16.

[29] ACI 2008. Building code requirements for structural concrete (ACI 318-08) and commentary (ACI 318R-08). American Concrete Institute, Farminton Hills, MI, USA.

[30] The MathWorks. 2012. Global optimization toolbox: user's guide. The MathWorks, Inc., Natick, MA, USA.

[31] Demuth, H., Beale, M., and Hagan, M. 2010. Neural network toolbox user's guide. The MathWorks Inc., Natick, Massachusetts, USA.