

Prediction of Ultimate Moment Capacity of Steel-Concrete Composite Beams Using Artificial Neural Networks

*Rana A. Mtashet, *Dr. Nabeel A. Jasim, and +Dr. Majid A. Alwan

* Department of Civil Engineering, Basrah university, Basrah, Iraq.

+ Department of Computer Engineering, Basrah university, Basrah, Iraq.

Abstract: The paper deals with neural networks identification of ultimate moment capacity of steel-concrete composite beams on base of experimental results. Basic information on artificial neural networks and its parameters suitable for analysis of experimental results are given. Two types of neural network algorithms are used. Results of identification are reported. The results show that artificial neural networks are highly suitable for assessing the ultimate moment capacity of composite section. The proposed neural network was also used to explore the effect of the various parameters on the behaviour of composite beams.

التنبؤ بمقدار العزم الأقصى للعتبات المركبة (حديد - خرسانة) باستخدام الشبكات العصبية

الصناعية

رنا عوده مطشر* ، دنيل عبد الزراق جاسم* ، د. ماجد عبد النبي علوان+

* قسم الهندسة المدنية ، جامعة البصرة ، البصرة ، العراق

+ قسم هندسة الحاسبات ، جامعة البصرة ، البصرة ، العراق

الخلاصة: أن هذا البحث يتعامل مع استخدام الشبكات العصبية الصناعية في إيجاد العزم الأقصى للعتبات المركبة من فولاذ وخرسانة بالاعتماد على نتائج الفحوصات العملية . وضحت الدراسة الحالية المعلومات الأساسية عن مفهوم الشبكات العصبية الصناعية وكذلك بعض العوامل المناسبة لتحليل نتائج الفحوصات العملية . تم استخدام مفهومين من مفاهيم الشبكات العصبية. أظهرت النتائج بأن الشبكات العصبية الصناعية مناسبة لتخمين العزم الأقصى للمقاطع المركبة. الشبكات المقترحة استخدمت في دراسة تأثير العوامل المختلفة على سلوك العتبات المركبة.

Introduction

The steel-concrete composite beams define a system consisting of steel beam, on top of which a concrete slab is cast. The interaction between concrete slab and steel beam is accomplished by means of mechanical

devices called shear connectors Fig.(1).

For design purposes, a single steel beam is assumed to act compositely with an effective width of the concrete slab, which is limited by the influence of shear lag [1]. Many recent codes (like

BS 5950-1990) of practice permit ultimate load design in which the calculation of the ultimate moment of resistance is based on the assumption of simple rectangular compressive stress blocks within the top concrete slabs. The required number of shear connectors is calculated simply by equating the total ultimate design shear capacity of shear connectors between sections of zero and maximum moment to the horizontal force that to be transmitted at failure [1].

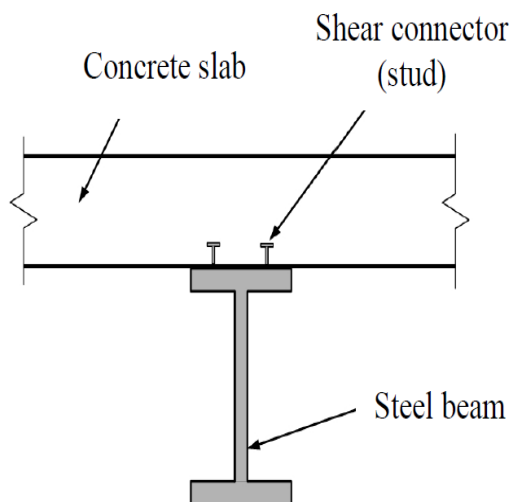


Fig. (1) Composite T-beam

Many research programs, both experimental and analytical have led to an understanding of the behaviour of composite beams and evaluating of the ultimate capacity of them. Chapman and Balakrishnan [1964] [2] studied the behaviour of seventeen simply supported composite T-beams under static concentrated and distributed loading. Ansourin and Rodrick [1978] [3] analyzed the behaviour of composite

beams including the effects of interface slip, yielding of steel, residual stresses, and non-linearity of the concrete. Ansourian [1982] [4] studied the sagging rotation capacity of composite beams. Razaqpur and Nofal [1990] [5] described various aspects of general finite element program for nonlinear analysis of composite structures. Abu-amer and Nassif [2003] [6] presented results of investigation to determine the effective slab width in composite beams from results of an experimental and analytical method.

In all previous works the analysis of composite beams were based on mathematical models which are complex for practical use. In this paper an attempt is made to use an alternative approach known as artificial neural network in which no mathematical model is required to analyze composite beams. Artificial neural networks are useful computing systems which can be trained to produce expected results from specified input data. The trained network can then be used indefinitely to produce results from fresh data not seen during the training phase. An artificial neural network has found wide application in all fields of science including structural engineering. The first application of neural network in structural engineering goes back only to the end 1980. Since then wide range of

applications have emerged. Vanluchen and Sun [1990] [7] used neural network to model simple reinforced concrete beam behaviour subject to bending moment. Jankins [1997] [8] applied a neural network based method to the approximate analysis of grillage structures. Lu [2000] [9] used two layer backpropagation neural network to predict the local and distortional buckling behaviour of cold-formed steel compression member. Pathak and Gupta [2006] [10] used neural networks for preliminary design of tubular girder bridge decks.

Artificial Neural Network

An artificial neural network can be represented as a simplified model of the nervous system. It consists of highly interacted simple processing elements [11]. The elements are called artificial neurons or nodes. The structure of artificial neural networks is as a rule, layered. Three functional groups can be distinguished in the artificial neural network. The set of input nodes that receives information from external sources and sends signals to the subsequent layers is called input layer. The set of output nodes that receive processed information and sends output signals is called output layer. Other layer(s) that receive(s) information from input layer and processes them in a hidden way to output layer is (are)

called hidden layer(s) as shown in Fig.(2). The lines in the diagram between the nodes indicate the flow of information from one node to next. This particular type of neural network known as feedforward neural network, the information flows only from input to output (that is, from left to right). Normally input and output layers have number of nodes equal to the number of input and output variables correspondingly, and the number of hidden layer(s) and number of nodes in hidden layer(s) depends upon the complexity of problem [12].

A model of the artificial neuron is shown in Fig.(3). The model includes n inputs, one output, a summation block, and an activation block.

The network can be trained for different types of function. The training process requires a set of data. The total data are divided into two groups; training data, and testing data. The training data are used to train the network to find the relationship between the input and output parameters [12].

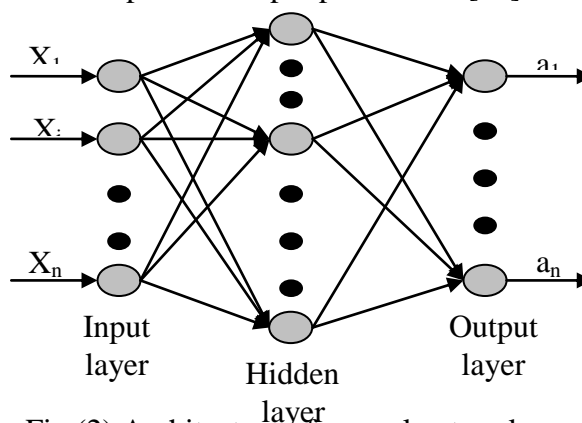


Fig.(2) Architecture of neural network

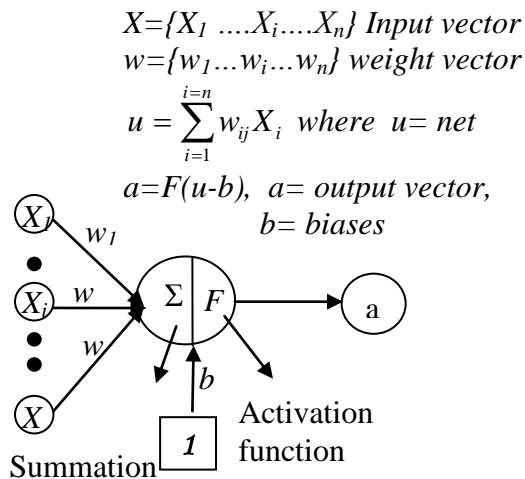


Fig.(3) Architecture of neural network

The neural networks interpolate data very well, but the extrapolation of data has not the same confidence. Therefore, the training data should be selected in such a way that it includes data from all regions of interest. After training network, an important aspect of developing neural networks is determining how well the network performs. Checking the performance of a trained network involves two main criteria; first run the network by using the training data to see whether the network produces good approximation to the known output for these data. Second, evaluating the network performance for a set of testing data that were not used in the training phase but desired outputs are available for comparison between the network output and the desired output. This property of network is called generalization. The

number of testing data are taken randomly as (20%) of total data [12].

Model Development and Optimization

A number of factors has an influence on the performance of the neural network, which can be described as the speed of learning and the generalization capacities of neural networks. In the following, several main factors are discussed in detail.

(1) Training Algorithm

There are different optimization techniques to be used in the training of neural networks. They have a variety of computation and storage requirements, and no one is best suited to all locations. In the following, the outlines of the two training optimization techniques that are used for building the neural network are defined[13]:

i) Steepest Descent with Momentum (GDM)

The standard backpropagation algorithm implements the steepest descent method (also called the gradient descent method). At each step of the steepest descent method the weights are adjusted in the direction in which the error function decreases most rapidly.

ii) Resilient Backpropagation (RPROP)

Resilient backpropagation is high performance algorithms that can converge from ten to one hundred times

faster than the algorithms discussed previously. In this algorithm only the sign of the derivative is used to determine the direction of the weight update, the magnitude of the derivative has no effect on the weight update.

(2) Pre- Processing and Post-Processing of Data

Data scaling is another essential step for network training. The network can be more speedy and efficient if the input and target are scaled to fall in specific range. The training is impale if the problem region is relatively narrow in some dimensions and elongated in others [14]. Moreover, the sigmoid (logsig, tansig) transfer function is usually used within the network, and upper and lower limits of output from these functions are (0 to 1, -1 to +1) respectively. Three normalization methods are used in this study.

i) Maximum and minimum method

$$p_n = 2 \frac{p - p_{\min}}{p_{\max} - p_{\min}} - 1$$

where, p_n is scale of the variable p , and p_{\min} , p_{\max} are minimum and maximum values of the variable p , respectively, [13].

ii) Mean and standard deviation method

$$p_n = \frac{p - p_{mean}}{p_{std}}$$

where :

$$p_{mean} = \frac{1}{z} \sum_{i=1}^z p_i$$

$$p_{std} = \left(\frac{1}{z-1} \sum_{i=1}^z (p_i - p_{mean})^2 \right)^{1/2}$$

z : number of training data [13].

iii) A simple linear normalization method

$$p_n = 0.2 + 0.6 \frac{p - p_{\min}}{p_{\max} - p_{\min}} \quad \text{for input}$$

values

$$t_n = \left(\frac{t}{10^g} \right)^{1/2} + k \quad \text{for output values}$$

where, t : target , t_n : scale of variable t , k : is a constant between 0.25 to -0.25 used to insure that the value of output fall between 0.2 to 0.8, and g : is the number of digit of integer value .Values of k and g used in this study are 0.08 and 3, respectively.

(3) Initializing Weight Factor

Prior to training a neural network, initial values for the weights, between the nodes of the various layers must be set. Typically, the weight factors are initialized to small and random values by using either Random or Widro-hoff method. When the nodes are connected by a large weight value, the neural network might become paralyzed [15]. This phenomenon occurs since, at the high output values corresponding to the high weight value, the derivative of the transfer function approaches zero, and accordingly the

weight change approaches to zero. Thus, the training of neural networks approaches to a halt.

(4) Learning Rate (α) and Momentum Coefficient (mc)

The learning rate and the momentum coefficient are two important parameters that control the effectiveness of the training algorithm. When using the steepest descent algorithm with momentum (GDM), the network performance can be improved by finding optimal values for learning rate (α) and the momentum coefficient (mc).

Application of Neural Network Model to Composite Beams:-

1) Simply Supported Beams Under Sagging Bending Moment

The standard backpropagation neural network with momentum is firstly used to determine the ultimate capacity of simply supported composite beams under sagging moments.

The method of trial and error was carried out to define the configuration of artificial neural network.

The nodes in input and output layers are usually determined by the engineering problem requirement. In this study the dimensions and properties

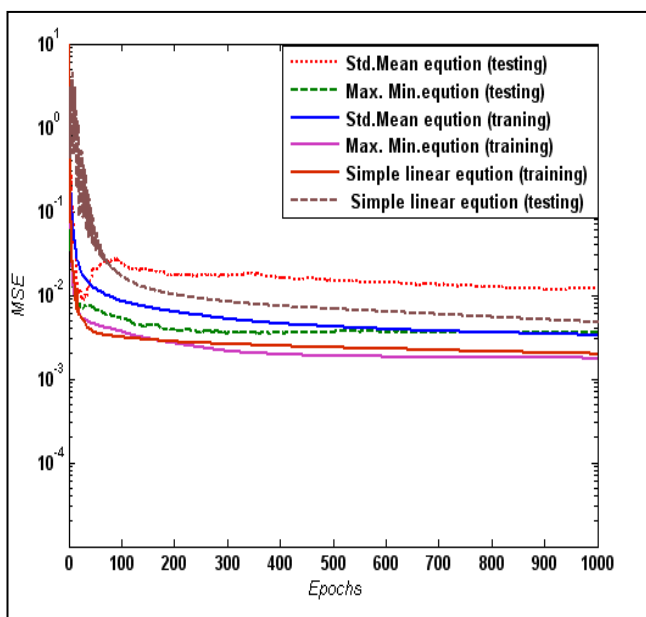
of concrete and steel are chosen as the components of input vector. The component of output vector is only ultimate moment capacity. The input vector contains:-

for concrete b_c, t_c, f_c

for steel A_s, f_y, d

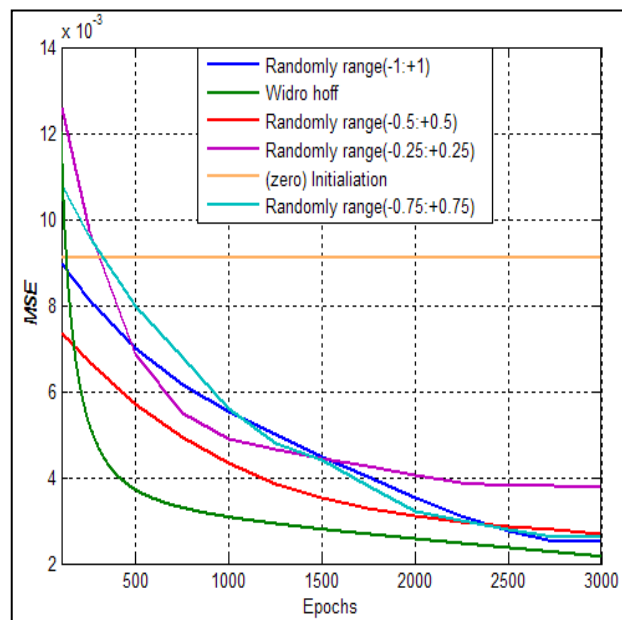
where, b_c is total concrete slab width, t_c is thickness of concrete slab, f_c is cylinder compressive strength, A_s is steel section area, f_y is steel yielding stress, and d is the total depth of steel section.

The three mentioned above normalization methods are used in this study. The performance of the network for both training and testing is shown in Fig.(4). From this figure it is found that the normalization by using Max. and Min. method gives best performance than the other one. The system being modeled is available so that in this study different initialization functions are used: Widro-Hoff initialization function, Random initialization function with ranges $[(-1 \text{ to } 1), (-0.75 \text{ to } 0.75), (-0.5 \text{ to } 0.5), \text{ and } (-0.25 \text{ to } 0.25)]$, and zero initialization function. Figure (5) shows the effect of these initialization functions on performance of the network. The Widro-Hoff function gives better performance than other functions.



Fig(4) Comparison between different normalization methods for training and testing

The number of hidden layers and number of nodes in each hidden layer depend on the network application. Although using a single hidden layer is sufficient in solving many functional approximation problems, some problems might be easier to solve with a two-hidden layer. So that in the present study the network is trained with one and two hidden layer(s). Eighty six data sets are used with the step descent and momentum rule for learning rate of 0.3 and momentum coefficient of 0.5. The transfer functions used are [tansig, pureline] for the network with one hidden layer and [tansig, purelin, purelin] for the neural network with two hidden layers. Figures (6) and (7) show the performance of the network with one and two hidden layers, respectively for both training and testing.



Fig(5) Effect of the initialization weight

The optimal configuration having minimum a verge square error is 6:4 (i.e. with 6 nodes in first hidden layer and 4 nodes in second hidden layer) and the configuration of this network is shown in Fig.(8).

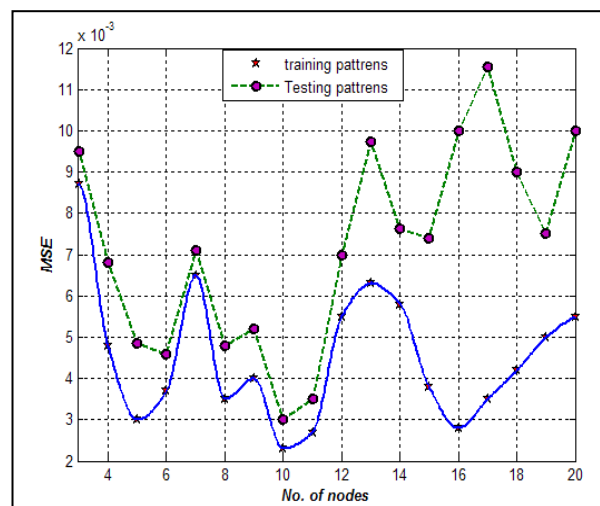


Fig.(6) Performance of network with one hidden layer

Effects of the learning rate (α) and the momentum coefficient (mc) on the behaviour of neural network is studied by using combinations of (α) [from 0.1 to 1.0 with a step of 0.1] and

(mc) [from 0.0 to 0.9 with a step of 0.1]. Each combination is trained with the selected network (two hidden layers 6-4) and with the same set of data, to 2000 epochs. Training results are shown in Table (1). From this table it can be seen that the momentum rate (from 0 to 0.7) improves the convergence of results in training process (i.e. decreasing the time of convergence). But for $\alpha=0.1$ and 0.2 the effect of (mc) is very small. In addition, the learning rate has a considerable effect on convergence of results. If it's high the algorithm may oscillate and become unstable. On other hand if it's small the algorithm will take a long time to converge. Table (1) shows that the effective values of learning rate and momentum coefficient are 0.3 and 0.7 respectively.

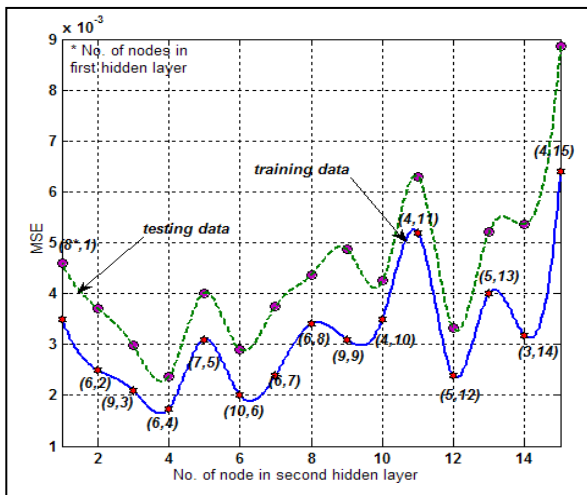


Fig.(7) Performance of network with two hidden layers

The relationship between the target (experimental) values and the output of neural network is shown in

Figs.(9) and (10) for training and testing data respectively. In figures, outputs are plotted versus the targets as open circles. The broken line indicates the best line fit and the solid line indicates the perfect fit (output equals target).

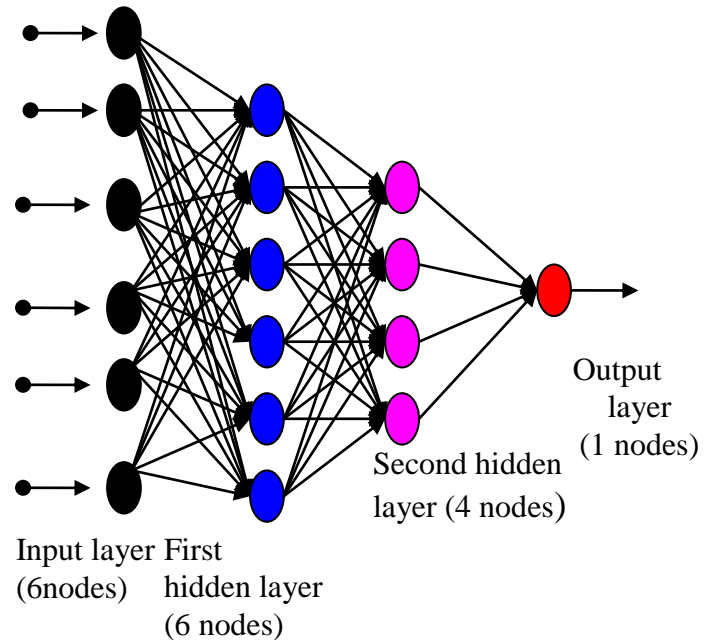


Fig. (8) Configuration of neural network (6-4)

The results show that the neural network correctly maps the training data and correctly identifies the testing data. This is evidenced by the fact that the broken line lie closely to solid line as well as by the high correlation coefficient (r) values.

Since the sigmoid activation functions are characterized by the fact that their slope approaches zero as the input becomes large, this may cause a problem when using the steepest descent algorithm to training a neural network. The gradient can have a very small magnitude, and therefore causes small changes in weight. Therefore, resilient backpropagation

(RPROP) algorithm is used to eliminate these harmful effects and improve the Performance of network. To make a comparison, the same training and testing sets are treated with the resilient algorithm as they are previously treated by (GDM). Compared to the (GDM), the RPROP produce a smaller mean square error (MSE) for the two phases training and testing as shown in Fig. (11). The comparison between the results of both algorithms related to the performance of neural network is summarized in Table (2).

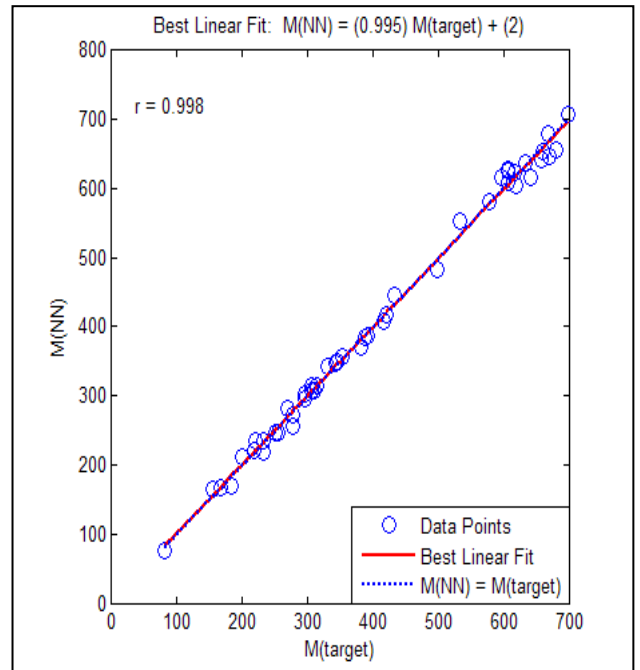


Fig.(9) Output versus target for training data

Table (1) Mean square error (MSE) for the network with different learning rates and momentum coefficients

Learning Rate (a)	Momentum (mc)									
	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	0.0035	0.00345	0.00346	0.00345	0.00343	0.00343	0.00346	0.00337	0.00338	0.00554
0.2	0.00245	0.00244	0.00243	0.00242	0.0024	0.00239	0.0024	0.00237	0.00238	0.004
0.3	0.00204	0.00193	0.00185	0.00174	0.00169	0.00163	0.00153	0.00154	0.00205	0.00263
0.4	0.00354	0.00274	0.00244	0.00206	0.00194	0.00137	0.0018	0.00175	0.00204	0.00334
0.5	0.0047	0.0040	0.0035	0.0030	0.00234	0.00226	0.00184	0.00174	0.0022	0.003
0.6	NaN	NaN	NaN	NaN	0.0045	0.0041	0.00384	0.0034	0.00554	0.00625
0.7	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.8	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.9	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

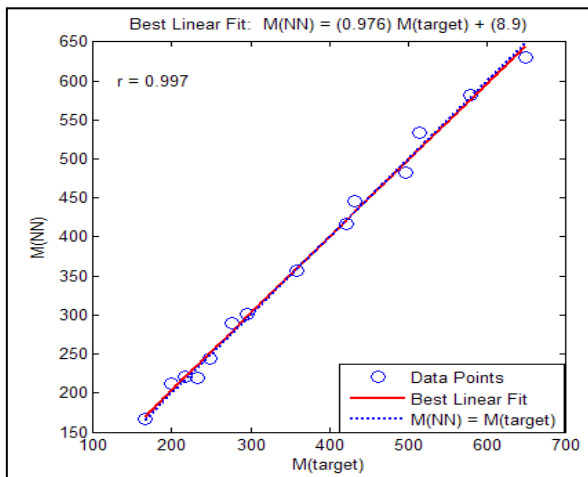


Fig.(10) Output versus target for testing data

Table (2) Performance of network for two different algorithms

Algorithm	Epochs	MSE training	MSE testing
GDM	2000	0.00154	0.00198
RPROP	456	0.0009	0.00099

Thus from the above analysis the network (6:4) topology with transfer function [tansig, purelin, purelin] with (RPROP) can be used to determine the ultimate moment capacity of composite section.

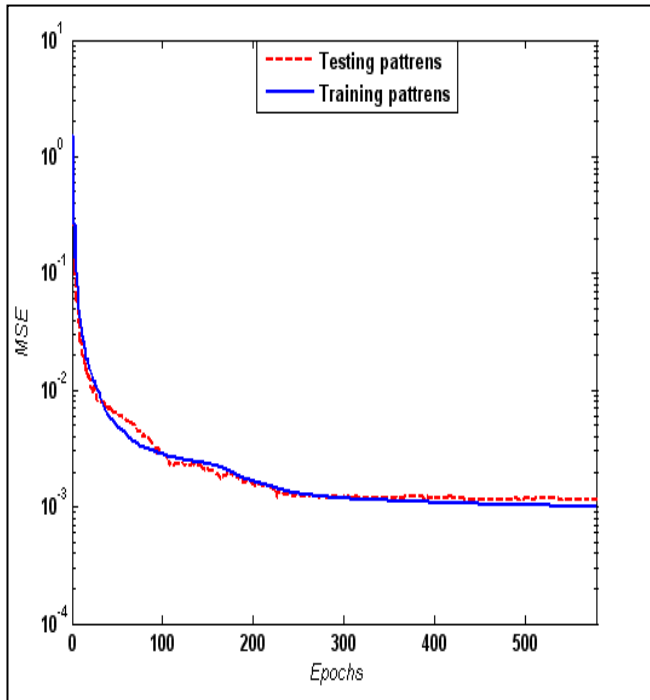


Fig. (11) Convergence history for both training and testing data

Parametric Study

Since the neural network is capable to generalization, parametric study can be carried out to evaluate the effect of variation of input parameters on the behaviour of composite beams. The results are shown in Figs. (12) to (14).

(1) Simply Supported Composite Beams Under sagging Moments

Figure (12) shows the variation of the ultimate moment capacity with the variation of concrete slab compressive strength (f_c). This figure

indicates that the ultimate moment capacity increases with the increase of (f_c).

Figure (13) shows the variation of ultimate moment capacity with yielding stress, f_y of the steel I-beam. The figure illustrates that the yield stress has an influence on the ultimate moment capacity. It increases as the yield stress increases.

Slab dimensions also affect the beam behaviour. The slab thickness has a marked effect on the beam strength but the slab width is less effective. Fig.(14) illustrates the effect of slab thickness on the ultimate moment capacity. Beam with deeper concrete slab have higher ultimate capacity. A small increase in ultimate moment capacity of composite beams occurs with increasing width of concrete slab. This can be seen in Fig.(15).

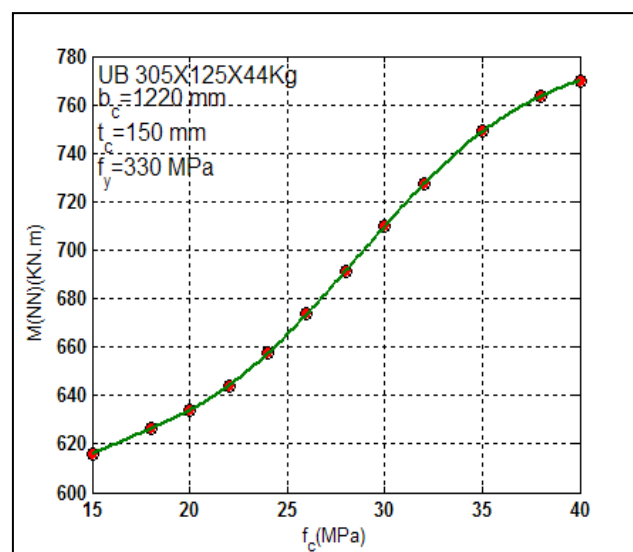


Fig.(12) Variation of ultimate moment capacity with variation of cylinder compressive strength (f_c)

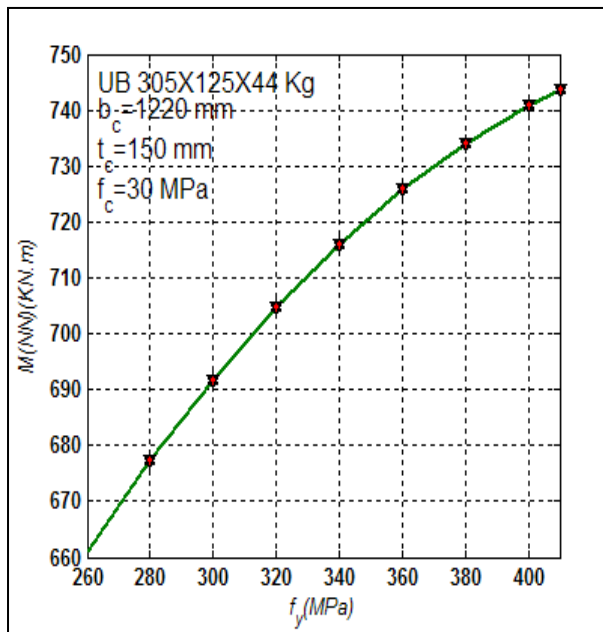


Fig.(13) Variation of ultimate moment capacity with variation of yielding stress of the steel I-beam

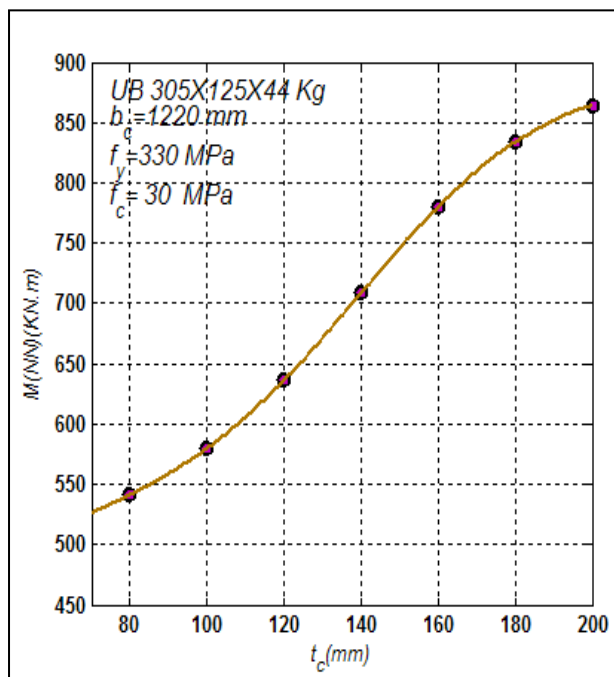


Fig.(14) Variation of ultimate moment capacity with variation of concrete slab thickness

When taking the effect of strain-hardening of steel beam by adding strain-hardening modulus (E_{sh}) to input parameters the MSE reduces from 0.00099 to 0.0006 with a new topology consisting of 9 and 4 nodes in first and

second hidden layer respectively. Accordingly, the proposed model with the inclusion of strain hardening of steel gives better convergence to the experimental results.

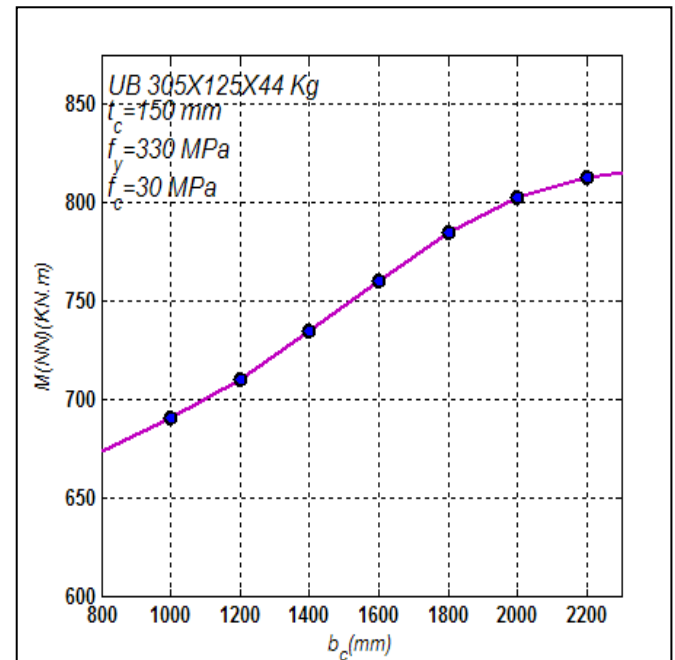


Fig.(15) Variation of ultimate moment capacity with variation of concrete slab width

(2) Simply Supported Composite Beams Under Hogging Moments

The backpropagation neural network is also used to investigate the behaviour of simply supported composite beams under hogging moments by calculating their ultimate moment capacity. The eight input parameters are: Slab longitudinal reinforcement (A_f), yielding stress of longitudinal reinforcement (f_{sk}), distance from centroid of longitudinal reinforcement to top steel flange (d_s), width of steel flange (b_f), thickness of steel flange (t_f), depth of web (d_w), thickness of web (t_w), yielding stress of

steel section (f_y). The output parameter is the ultimate hogging moment capacity. The neural network model is trained based on the experimental results.

Networks with both one and two hidden layers (and varying number of nodes in each layer) are investigated. From this investigation the best topology is found to be the one with (6) and (5) nodes in the first and second hidden layers, respectively (i.e. a 8-6-5-1 topology). It produces a minimum MSE, therefore this topology is used in this part of study. Figure (16) shows the convergence history of this neural network with the resilient backpropagation training algorithm and (tansig, purelin, purelin) activation functions. The suitability of this neural network model is tested by comparing the target (experimental) value and the output of the neural network for both training and testing data, respectively. The results show that the neural network correctly maps the training data and correctly identifies the testing data.

The proposed network is used to investigate the effectiveness of the slab longitudinal reinforcement on behaviour of the composite beam as shown in Fig.(17) where the increase of the slab longitudinal reinforcement leads to an increase in the moment capacity.

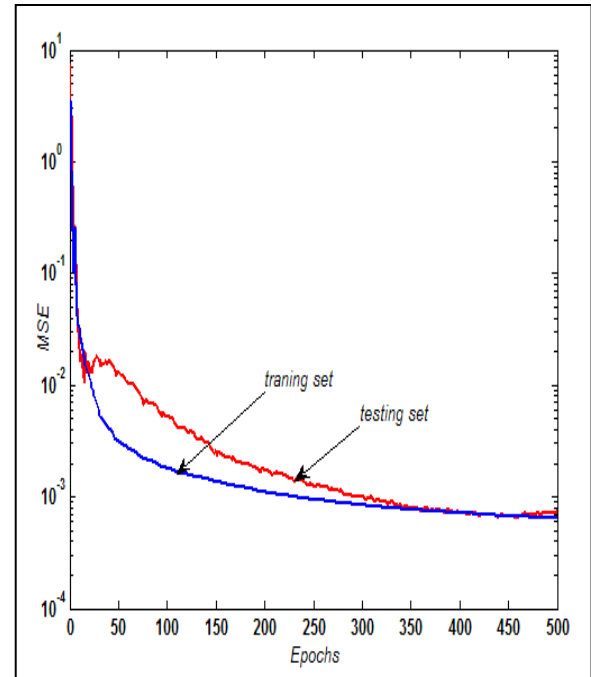


Fig.(16) Convergence history for the hogging moment case

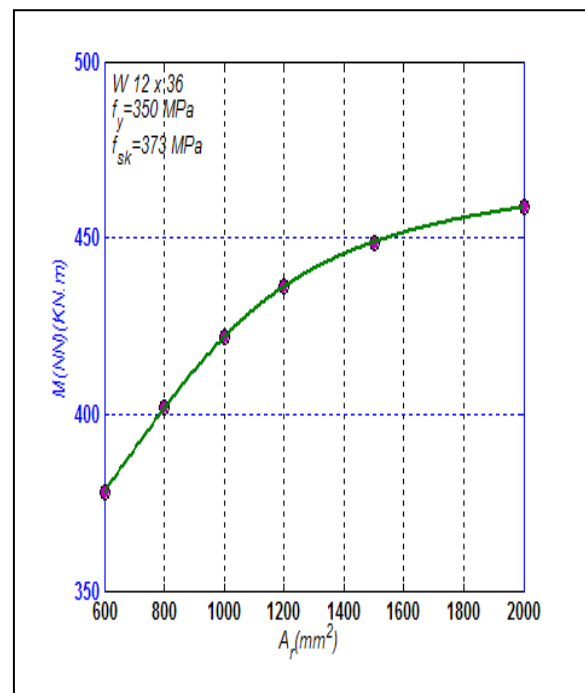


Fig.(17) Variation of ultimate moment with longitudinal reinforcement for the hogging moment case

Figure (18) shows the effect of the yielding stress f_y of steel section. This figure shows that the ultimate moment capacity increases with increasing of f_y .

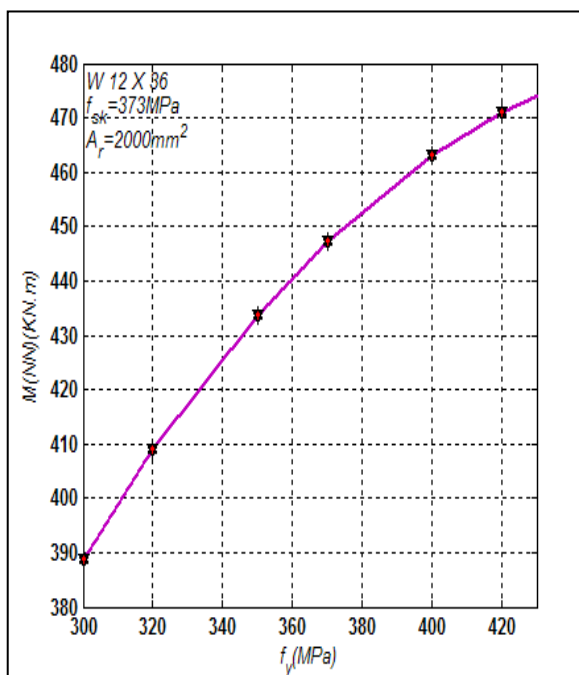


Fig.(18) Variation of the ultimate moment with steel yielding stress

Conclusions

- 1) The neural network model has been proved to be very effective in the analysis of composite beams.
- 2) The Max. Min. normalization method, Widro-Hoff initialization weight, and [tansig, purelin, purelin] transfer functions are found to give best performance for both training and testing phases.
- 3) When using (GDM) algorithm the convergence of the training process becomes more effective with learning rate and momentum coefficient of 0.3 and 0.7, respectively.

- 4) The neural network trained with the RPROP algorithm exhibits better behaviour than that trained with the GDM algorithm. This is characterized by reducing training time by 77.2% and better mapping of the neural network for the training data by 41.55% and generalization for the test data by 52.63%.

Notation and Abbreviations

- a neural network output.
- A_r area of slab longitudinal reinforcement in the effective cross section.
- A_s cross section area of steel beam.
- b biases.
- b_c concrete slab width.
- b_f flange width of steel beam.
- d total depth of steel section.
- d_r distance between the centroid of longitudinal reinforcement and top of concrete slab.
- d_w web depth of steel beam.
- F neural network transfer function.
- f_c compressive strength of concrete cylinder.
- f_{sk} yielding stress of slab longitudinal reinforcement.
- f_y yielding stress of steel section.
- M_c theoretical moment capacity of composite beam.
- r correlation coefficient.
- t target.
- t_c thickness of concrete slab.

t_f flange thickness of steel beam.
 t_w web thickness of steel beam.
 w weight factor.
 X neural network input.
 α learning rate.
 GDM steepest descent with momentum.
 M(NN) neural network moment capacity of composite beam.
 MSE mean square error.
 RPROP resilient backpropagation.

References

- [1] **Johnson, R. P.**, "Composite Structures of Steel and Concrete", Vol. 1, Beams, Colmun, Frames and Application in Building, Crosby-lockwood Staples, London, 1975, 190 pp.
- [2] **Chapman, J. C., and Balakrishnan, S.**, " *Experiments on Composite Beams*", The Structural Engineer, Vol.42, No.11, Nov. 1964, pp.369-383.
- [3] **Ansourian, P. and Roderick, J. W.**, " *Analysis of Composite Beams*", Journal of the Structural Division, Vol.104 ,No. ST10, October 1978, pp.1631-1645.
- [4] **Ansourian, P.**, " *Plastic Rotation of Composite Beams* ", Journal of Structure Division, Vol .108, No.ST 3, March 1982, pp. 643-659.
- [5] **Razaqpur, G. A., and Nofal, M.**, " *Analysis Modeling of Nonlinear Behavior of Composite Bridges* ", Journal of Structural Engineering, Vol.116, No. 6, June, 1990, pp.1715-1733.
- [6] **Abu-Amer, T. F., and Nassif, H.**, " *Effect of Flange Width of the Deflection of Steel Composite Beams* ", Proc. of the 9th Arab Structural Engineering Conf. Nov. 29, 2003, Abu-Dhabi.
- [7] **Vanluchen, R., and Sun, R.**, " *Neural Networks in Structural Engineering* ", Microcomp. Civ. Engrs, 1990, pp. 207 -215.
- [8] **Jenkins, W.**, " *Approximate Analysis of Structural Grillages Using a Neural Network* ", Proc. Inst. Civ. Engrs. Structs & Buildings, Aug. 1997, pp. 355-363.
- [9] **Lu, W.**, " *Neural Network Model for Distortional Buckling Behaviour of Cold-Formed Steel Compression Members*", Ph.D. Thesis, Helsinki University of Technology, 2000, 70 pp.
- [10] **Pathak, R., and Gupta, A.**, " *Analysis & Design of Tubular Girder for Bridge Application Using Artificial Intelligence*", Advance in Bridge Engineering, March 2006, pp 653-668.
- [11] **Jerzy, H., and Krzysztof, S.**, " *Application of Artificial Neural Network to Determine Compressive Strength Based on Non-Destructive Test*", Journal of Civil Engineering and

- Management, Vol. XI, No.1, 21 Feb. 2005, pp. 23-32.
- [12] **Lau, C.**, "*Neural Networks: theoretical foundation and analysis*", New York: IEEE Press, 1992.
- [13] **Neural Network Toolbox User's Guide:** for Use with MATLAB 2004.
- [14] **Flood, and Kartam, N.**, "*Neural Networks in Civil Engineering. I: Principles and Understanding*", Journal of Computing in Civil Engineering, Vol.8, No.2, April 1994, pp.131-148.
- [15] **Hegazy, T., Fazio, P. and Moselhi,** "*Developing Practical Neural Network Applications Using Back-Propagation*", Microcomputers in Civil Engineering, Vol.9, No. 2, pp.145-159.