PREDICTION OF ULTIMATE TORSIONAL STRENGTH OF SPANDREL BEAMS USING ARTIFICIAL NEURAL NETWORKS

Prof. Dr. Nabeel A. Jasim*, and Meyyada Y. Mohammed*

*****_Dept. of civil engineering, College of engineering, University of Basrah, Basrah, Iraq._

Abstract: A spandrel beam is a structural member lies at the edge of a frame and is connected by a joint to the floor beam extending into the slab. The spandrel beams are primarily responsible for transferring forces from a slab to the supporting edge columns. This work investigates the possibility of using the artificial neural networks to model the complicated nonlinear relationship between the various input parameters associated with reinforced concrete spandrel beams and the actual ultimate strength of them. The descent gradient backpropagation algorithm was employed for predicting the ultimate strength of the reinforced concrete spandrel beams. The optimum topology (which gives least mean square error for both training and testing with fewer number of epochs) is presented. Effects of parameters such as, number of hidden layer(s), number of nodes in the input layer, output layer and hidden layer(s), initialization weight factors and selection of the learning rate and momentum coefficient on the behaviour of the neural network have been investigated. Because of the slow convergence of results when using descent gradient backpropagation, another algorithm which is faster called "resilient backpropagation algorithm" has been used. The neural network trained with the resilient backpropagation RPROP algorithm gives better results than that trained with the steepest descent algorithm with momentum GDM algorithm.

تقييم مقاومة اللي القصوى لعتبات الحافة الخرسانية المسلحة باستخدام تقنية الشبكات العصبية الصناعية

د. نبيل عبد الرزاق جاسم*، ميادة يحيى محمد*

*****قسم الهندسة المدنية، كلية الهندسة، جامعة البصرة، البصرة، العراق.

الملخص: عتبة الحافة عبارة عن عضو إنشائي يقع عند حافة المنشأ ويُربط مع عتبة الأرضية والسقف, لذلك فأن عتبات الحافة تكون مسئولة بصورة رئيسية عن نقل القوى من السقف إلى الأعمدة الساندة للحافات. يتحرى هذا البحث إمكانية استخدام الشبكات العصبية الصناعية لتشكيل العلاقة اللاخطّية المعقّدة بين المتغيرات المختلفة الخاصة بعتبة الحافة الخرسانية المسلحة والمقاومة القصوى الفعلية لهذه العتبة. أستخدم مفهوم الانحدار العكسي في تدريب نموذج الشبكة العصبية لتقييم المقاومة القصوى لعتبات الحافة الخرسانية المسلحة, وولدت الشبكة المثلى (التي تعطي اقل معدل مربع الخطأ لنماذ ج التدريب والفحص بأقل عدد من الدورات). وكذلك تم فحص تأثير المتغيرات المختلفة للشبكة مثل عدد الطبقات المخفية وعدد العقد في طبقة الإدخال والطبقات المخفية والفرض الأولي لمعاملات الأوزان و اختيار معدل التعلم ومعامل الزخم على سلوك وأداء الشبكة العصبية. ونتيجة لبطئ الإنجاز عند استخدام مفهوم الانحدار العكسي تم استخدام مفهوم آخر أسرع من المفهوم الأول يسمى "الإرجاع العكسي المرن" لتدريب الشبكة العصبية معطية نتائج أفضل من الطريقة السابقة وهذا وجد من خلال تخفيض وقت التدريب (عدد الدورات) وتقارب أفضل بين نماذج التدريب ونماذج الاختبار (والتي ليست ضمن نماذج التدريب).

Introduction

Artificial Neural Networks (ANNs) are a computational tool that attempt to simulate the architecture and internal features of the human brain and nervous system. They have been widely used for prediction and classification problems.

Neural Network consists of a number of interconnected processing elements, commonly referred to as neurons or nodes as shown in Fig. (1).

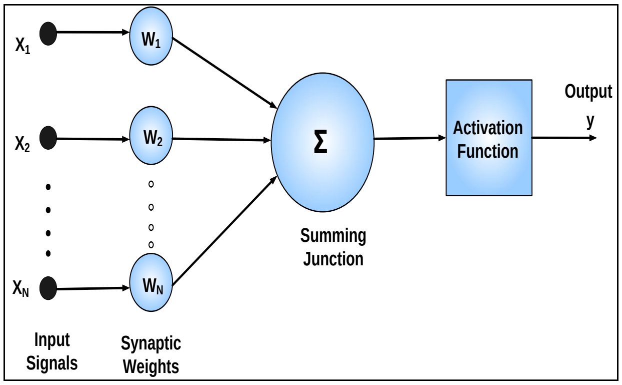

Fig. (1) A Processing Unit [1]

The nodes are logically arranged into two or more layers and interact with each other via weighted connections. These scalar weights determine the nature and strength of the influence between interconnected nodes. Each node is connected to all the nodes in the next layer. There is an input layer where data are presented to the neural network and an output layer that hold the response of the network to the input. Between them there are intermediate layers, also known as the hidden layers, which enable these networks to represent and compute complicated associations between patterns.

Each hidden and output node processes its inputs by multiplying each input by its weight, summing the product, and then passing the sum through a nonlinear activation function to produce a result [2]. Figure (2) shows Architecture of neural network.

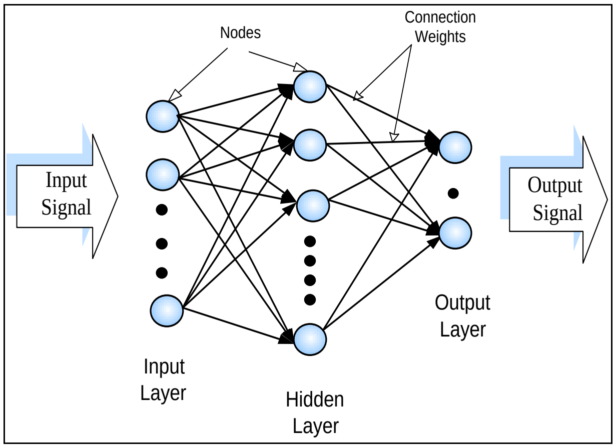

Fig. (2) Architecture of Neural Network [1]

The present study investigates the possibility of using the artificial neural networks to model the complicated nonlinear relationship between the various inputs parameters associated with a reinforced concrete spandrel beam and the actual ultimate strength of the spandrel beam.

Behaviour of Reinforced Concrete Spandrel Beams

A spandrel beam is a structural member lies at the edge of a frame, and is connected by a joint to the floor beam extending into the slab. The loads applied to the slab are carried by the floor beams, which transfer certain amount of these loads to the spandrel beams through the joint, causing the spandrel beams to twist under those eccentric loads [3], so that they are primarily responsible for transferring forces from a slab to the supporting edge columns and as a result, they are subjected to a combination of torsion, shear and bending [4].

Figure (3) shows a portion of a floor–spandrel beam assembly within a frame lies between the points of inflection, which can be simulated by hinges. In these points the bending moment equals zero, as shown in Fig. (4).

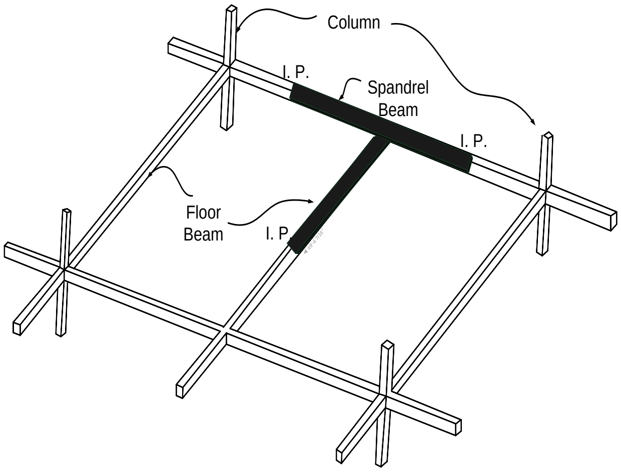

Fig. (3) Spandrel Beam Within a Structural Frame

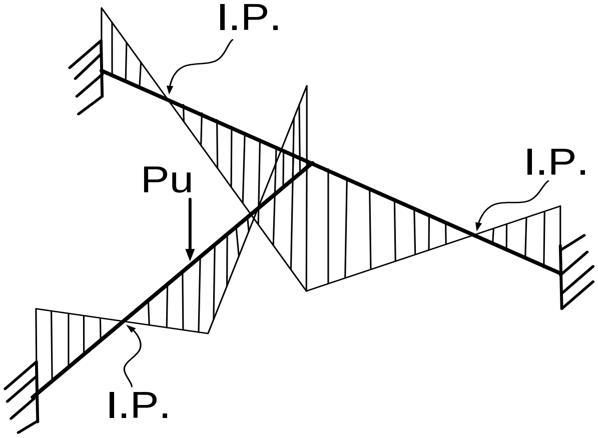

Fig. (4) Bending Moment Diagram for the Spandrel Beam

The design of a floor–spandrel beam assembly is entirely dependent on the value of the torsional moment (Tu). Different assumptions were made [3,6] for the value of (Tu) which was given by the equation:

$$Tu=\varphi\sqrt{fc'}\ b^2h/3$$

where (b) is the width of beam section, (h) is its overall depth, and (fc') is the cylinder concrete compressive strength (MPa).

The value of the coefficient (φ) was taken as zero, 0.25, 0.33, and 0.44.

Selection of the Training and Testing Patterns

Multilayered feedforward backpropagation neural networks are used for this research, which are implemented using neural network toolbox that is available in MATLAB version 7.0.0 (2004). This program implements several different neural network algorithms, including backpropagation algorithm

The experimental results used to construct the neural network are those obtained from available literature [3, 6, 7, 8, and 9].

The total experimental data are divided into two sets: a training set and a testing set. The training set is used for computing the gradient and updating the network weights and biases to diminish the training error, and thus finding the relationship between the input and output parameters. Hence, the learning process is a crucial phase in NN modeling. The testing set is used to evaluate the generalisation ability of the learning process. In this study the testing set contains approximately (15)% of total database. The parameters used in this study are shown in Table (1). Dimensions of the spandrel beam are shown in Fig. (5).

Table (1) Input and Output Parameters

Item	*Parameters*	*Range of Parameters*		*Units*
		From	*To*	
Input Parameters	(bs)	120	180	mm
	(hs)	200	300	mm
	(ls)	1200	3000	mm
	(As_t)	0	226.2	mm2
	(fy_1)	no rein. & 289	560.8	MPa
	(As_b)	157.1	628.3	mm2
	(fy_2)	413.4	560.8	MPa
	(ρ)	0	1	%
	(fy_3)	no stirrup & 344.5	540	MPa
	(fc')	20	49.1	MPa
Output Parameter	(Tu)	1.3	14.95	kN.m

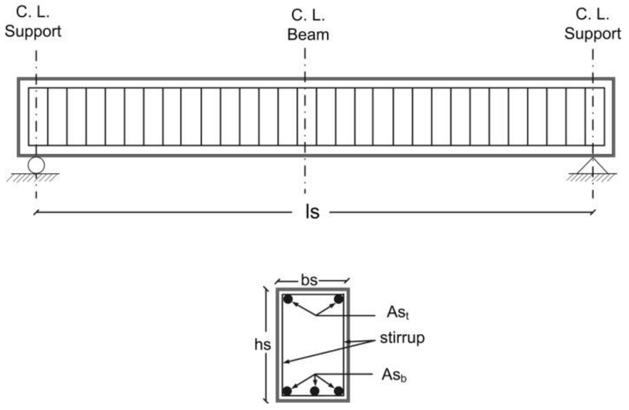

Fig. (5) Dimensions of Spandrel Beam

The total number of (48) test beams were utilized. The training set contains (41) beams and the testing set comprises of (7) beams. Neural networks interpolate data very well. Therefore, the training set should be selected in such a way that it includes data from all regions of desirable operation.

Input and Output Nodes

The main difficulty in the structural identification of a complex nonlinear system arises from the huge amount of possible relationships among variables. The selection of outputs is straightforward and depends on the modeling goal. However, informed input-variable selection is critical in achieving efficient model performance. In this study, the parameters which may be introduced as the components of the input vector consist of the total depth of spandrel beam cross section (hs), the width of spandrel beam cross section (bs), the length of spandrel beam (ls), area of compression steel reinforcement (As_t), yielding stress of compression steel reinforcement (fy_1), area of tension steel reinforcement (As_b), yielding stress of tension steel reinforcement (fy_2), the ratio of transverse stirrup (ρ), yielding stress of transverse reinforcement (fy_3), and concrete cylinder compressive strength (fc'). The output vector is the ultimate torsional moment of spandrel beams (Tu). Therefore, the nodes in the input layer and output layer are (10) and (1), respectively.

Weight Initialization

The first step in the neural network computation is the initialization of the weight factors between any two nodes within the network. Because no prior information about the system being modeled is available, therefore in this study different initialization functions are used. These include Widrow-Hoff initialization function which changes the weight after each run, zero initialization function, and random initialization function with ranges [(-0.25 to +0.25), (-0.5 to +0.5), (-0.75 to +0.75), and (-1 to +1)]. Figure (6) shows the effect of using these initialization functions on the performance of network. The performance is determined as the difference between the target output which is known from the experimental work and the output of the network, and it is calculated as the square of this difference and denoted by the mean square error (MSE). This figure shows that the Widrow-Hoff gives better performance than other functions. Therefore Widrow-Hoff initialization function is used in this study.

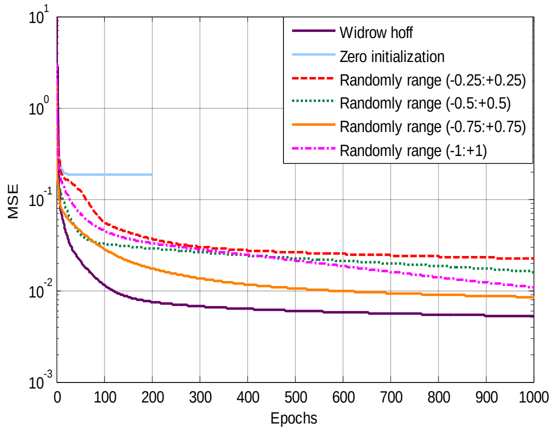

Fig. (6) Effect of Initialization Weight Functions

Number of Hidden Layers and Nodes in Each Hidden Layer

The number of hidden layers depends on the nonlinearity of the problem. The choice of the number of hidden units is therefore a tradeoff between the necessary flexibility of the model and the time it takes to train the network. It is usually to start with a relatively small number of hidden units and increase it until the approximation quality of the network becomes acceptable. Unfortunately, the network needs to be fully retrained after each modification of its structure.

The number of nodes in a hidden layer(s) drastically affects the outcome of the network training. If the number of nodes is small, the network becomes unable to perform the problem satisfactorily as shown in Fig. (7) in which (2) nodes are used in a network of one hidden layer. This is because reducing the number of hidden units will reduce the interconnection of the network. On the other hand, if the number of nodes selected is too large the network may give good training but the performance of testing data is poor (over-fitting) as shown in Fig. (8) in which (25) nodes are used in a network of one hidden layer. This is because the neural networks developed complex relationship between input parameters and outputs.

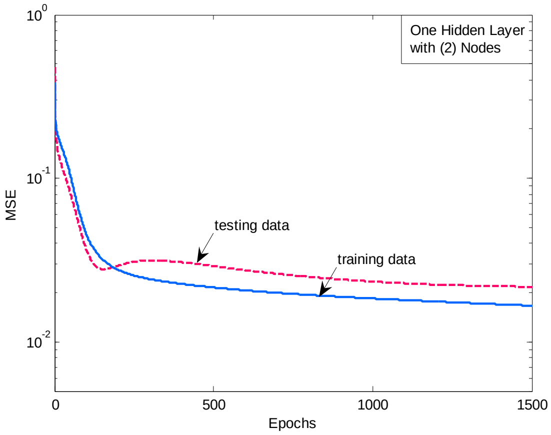

Fig. (7) Performance of Network (Under Fitting)

Over-fittings and predictions in training and outputs of neural networks are commonly influenced by the number of hidden layers and nodes in each hidden layer. Therefore, trial-and-error approach is carried out to choose an adequate number of hidden layers and number of nodes in each hidden layer.

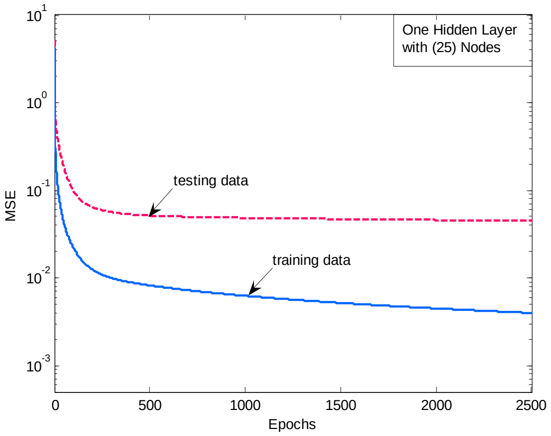

Fig. (8) Performance of Network (Over Fitting)

The number of nodes in the hidden layer is selected according to the following rules [1]:

1. The maximum error of the output network parameters should be as small as possible for both training patterns and testing patterns.
2. The training epochs should be as few as possible.

In this study the network is tested with one and two hidden layer configurations with an increasing number of nodes in each hidden layer(s). The optimal topology is determined first by using one hidden layer with activation function as hyperbolic tangent (tansig) function in hidden layer and linear (purelin) function in output layer. Different numbers of nodes from (2 to 15) are investigated and the performance of these topologies for both training and testing are shown in Fig. (9). From this figure the network with (6) nodes in the hidden layer gives the best performance for both training and testing than other (MSE=0.00373). Then two hidden layers are used with activation functions as hyperbolic tangent (tansig) function in first hidden layer and linear (purelin) function in both second hidden and output layer. Different numbers of nodes in each hidden layer from (2 to 12) nodes are used. The performance of these topologies of network for both training and testing is shown in Fig. (10). From this figure the network with (6-9) nodes in the first and second hidden layers gives the best performance for both training and testing (MSE=0.00254). The results show that a network with two hidden layers is significantly better than that with one hidden layer. Also the number of nodes in two hidden layers affects the number of epochs required for the results to converge. This is clearly shown in Fig. (11). From this figure it is found that the network with (6-9) gives best performance with small number of epochs.

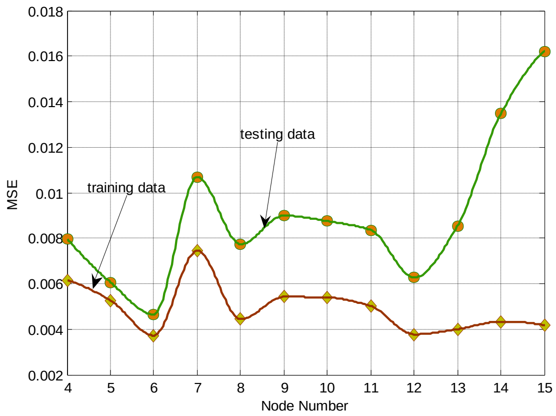

Fig. (9) Performance of Network with One Hidden Layer

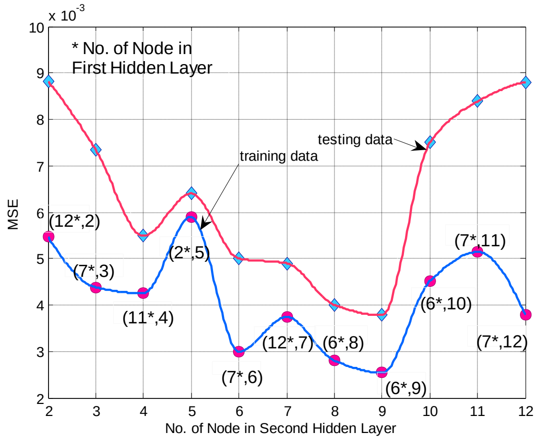

Fig. (10) Performance of Network with Two Hidden Layers

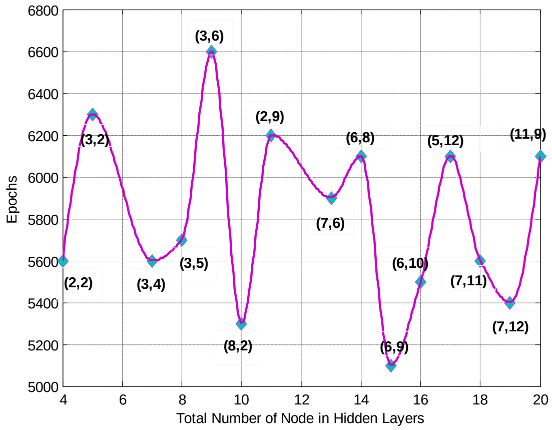

Fig. (11) Effect Number of Nodes on Number of Epochs

Table (2) shows how the number of nodes in the two hidden layers affects the response of network with different types and arrangements activation functions.

Table (2) MSE for The Networks with Different Types and Arrangements of Activation Functions

Network type	*(tansig, purelin, purelin)*	*(tansig, tansig, purelin)*	*(tansig, tansig, tansig)*	*(purelin , tansig, tansig)*	*(tansig, purelin, tansig)*
6-8	0.0028	0.0034	0.00338	0.0045	0.00493
6-9	0.00254	0.00388	0.00559	0.00366	0.00411
6-10	0.00452	0.00283	0.00276	0.00423	0.00508
7-6	0.003	0.00323	0.00308	0.00668	0.00466
7-11	0.00514	0.0033	0.00286	0.00268	0.00505
7-12	0.00378	0.00457	0.00431	0.00356	0.00484
11-4	0.00425	0.00385	0.00305	0.00368	0.00485
12-7	0.0026	0.00312	0.00286	0.00335	0.00335

From the Table above, it can be seen that the networks with [[6-9] (tansig, purelin, purelin), [6-10] (tansig, tansig, tansig), [7-11] (purelin, tansig, tansig)] give the best performance. Figure (12) shows the convergence history of these networks for both training and testing data. It can be seen that the network (6-9) gives the best performance for both training and testing. From the above analysis, the node number of (6-9) in the hidden layers [(6) nodes in the first hidden layer and (9) nodes in the second hidden layer] is chosen. The configuration of the neural network is shown in Fig. (13).

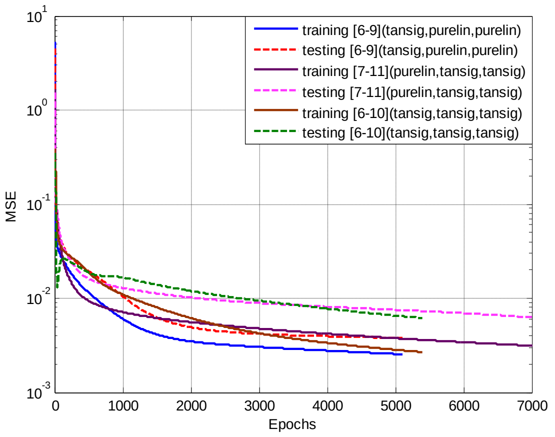

Fig. (12) Convergence History for Both Training and Testing

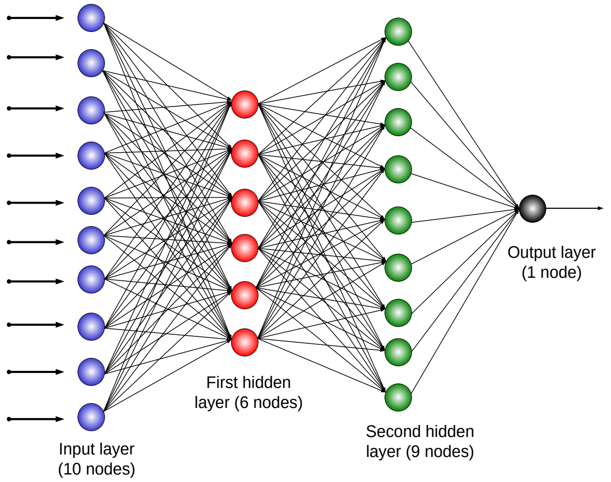

Fig. (13) Configuration of Neural Network (6-9)

Selection of Learning Rate (α) and Momentum Coefficient (μ)

The learning rate and momentum coefficient are two important parameters that control the effectiveness of the training algorithm. 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 (μ). The effect of learning rate (α) and momentum (μ) on the behaviour of neural network is studied by using the combination of (α) [from 0.05 to 0.9 with a step of 0.05] and (μ) [from 0.0 to 0.9 with a step of 0.1]. Each combination is trained with the selected network (two hidden layers 6,9) and with same set of data, and initial weight to (5100) epochs. Training results are shown in Fig. (14). From this figure it can be seen that for α=0.05 and μ =0.4 the network gives the best performance with (MSE=0.00254) and they are chosen for the proposed network. In addition, the learning rate has a considerable effect on convergence of results. If it is high the algorithm may oscillate and paralyzed (NaN), on the other hand if it is small the algorithm will take a long time to converge.

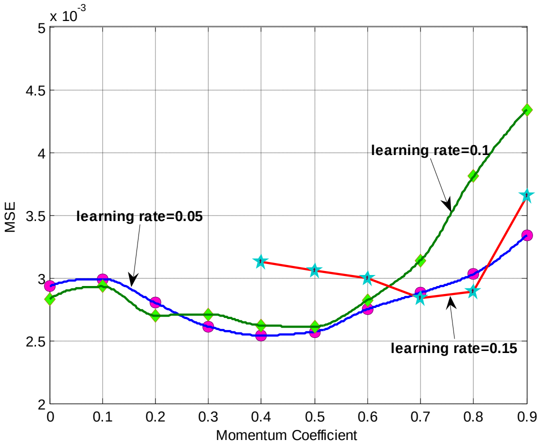

Fig. (14) Effect of Combination of Learning Rate and Momentum

The convergence history of this network is shown in Fig. (15), and Table (3) shows the properties of this network.

Table (3) Properties of The Proposed Network

Network	***Epochs***	***MSE Training***	***MSE Testing***
(6-9-1)	5100	0.00254	0.00378

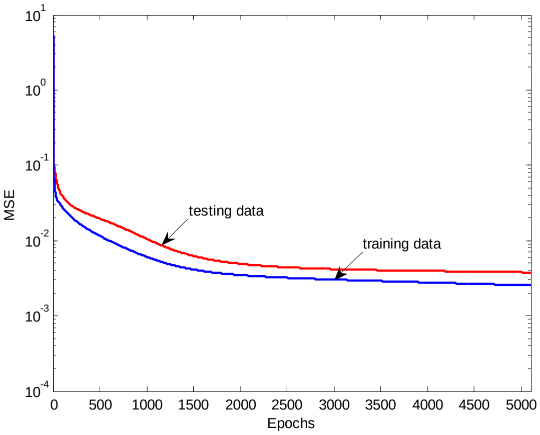

Fig. (15) Convergence History for Both Training and Testing Data

Generalization of Neural Network

The performance of a trained network can be measured to some extent by the errors on the training and testing sets, but it is often useful to investigate the network response in more detail. One option is to perform a regression analysis between the network response and the corresponding targets. The routine (postreg) is designed to perform this analysis.

The format of this routine is [m,c,r] = postreg (a,t), where (m) and (c) correspond to the slope and the y-axis intercept of the best linear regression that relates the targets to the network outputs. If the fit is perfect (outputs exactly equal to targets), the slope would be (1), and the intercept with the y-axis would be (0). The third variable, (r), is the correlation coefficient between the outputs and targets. It is a measure of how well the variation in the output is explained by the targets. If this number is equal to (1), then there is perfect correlation between targets and outputs.

Figures (16) and (17) show the regression analysis between the output of neural network and the corresponding target for training and testing data respectively. In the figures, outputs are plotted versus the targets as open circles. The solid line indicates the best linear fit and the broken line indicates the perfect fit (output equals targets). The values of the slope are (0.986) and (1.1) respectively, interceptions with y-axis are (0.0571) and (-0.207) respectively, and correlation coefficient is (0.993). These values indicate that the mapping of neural network for the both training and testing data is very good.

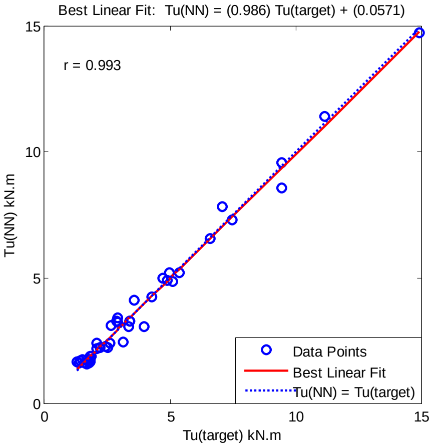

Fig. (16) Regression Analysis Based on (GDM) for Training Data

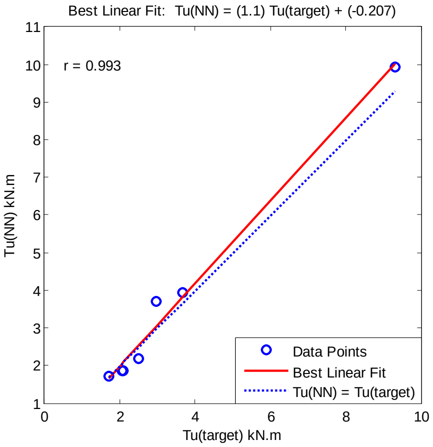

Fig. (17) Regression Analysis Based on (GDM) for Testing Data

Backpropagation Training Algorithms

In this study the standard backpropagation algorithm with momentum (steepest descent with momentum) is first used to adjust both weights and biases and the results of this method are explained in above. Normally, the multilayer networks use sigmoid and hyperbolic tangent activation functions, which are characterized by the fact that their slope must approach zeros as the input becomes large. This causes a problem when using the steepest descent to train a neural network, since the gradient can have a very small magnitude, and therefore cause small changes in the weights and biases, even though the weights and biases are far from their optimal values. The purpose of the Resilient PROPagation (RPROP) algorithm is to eliminate these harmful effects. 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 [1].

The training and testing sets are treated with the resilient backpropagation algorithm similarly as in the gradient descent backpropagation. Compared to the gradient descent backpropagation, the resilient backpropagation algorithm produces a smaller (MSE) for the two phases of training and testing as shown in Fig. (18). From this figure and Fig. (15), it is found that the resilient gradient gives convergence faster (small number of epochs) than the gradient descent backpropagation.

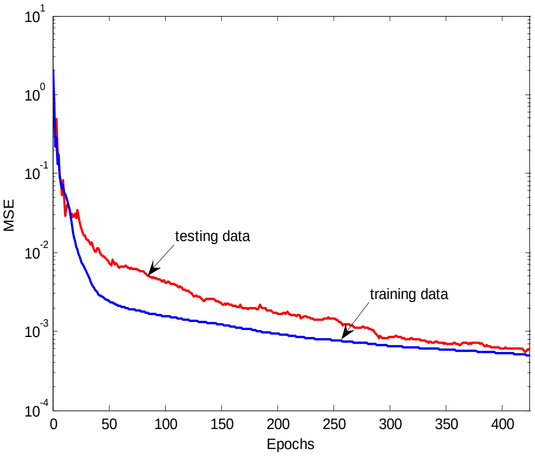

Fig. (18) Convergence History for Both Training and Testing Data for (6-9) Net Using (RPROP)

The resilient backpropagation algorithm requires exactly (425) epochs for MSE (for training set) to drop to a value of (0.0005), compared to (5100) epochs required to reach a value of (0.00254) MSE for gradient descent backpropagation method. The comparison between the results of both algorithms related to the performance of neural network is shown in Fig. (19), and is summarized in Table (4).

Table (4) Performance of Two Different Algorithms for Network Model

Algorithm	***Epochs***	***MSE training***	***MSE testing***
GDM	5100	0.00254	0.00378
RPROP	425	0.0005	0.0006

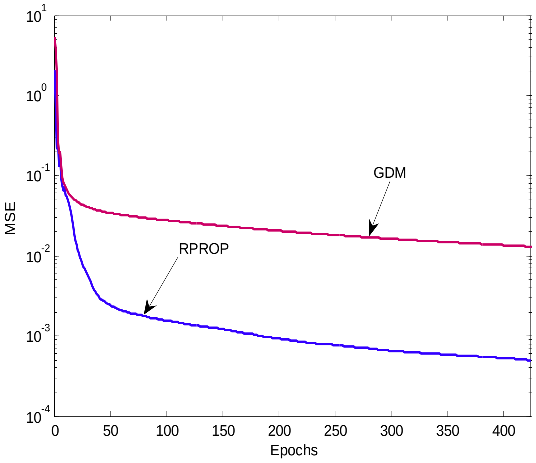

Fig. (19) Comparison Between (GDM) and (RPROP) Algorithms

The performance of resilient backpropagation algorithm is tested by the regression analysis between the output of this network and the corresponding targets for both training and testing data as shown in Figs. (20) and (21). In the figures m= 0.997, 1.03, c= 0.0108, -0.139, and r= 0.999, 0.998 for training and testing data respectively. These values indicate an excellent agreement between the predicted values and the target values.

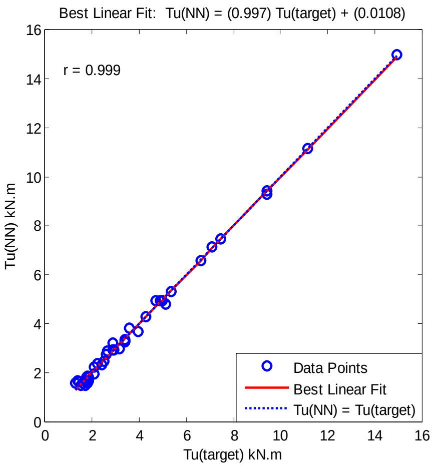

Fig. (20) Regression Analysis Based on (RPROP) for Training Data

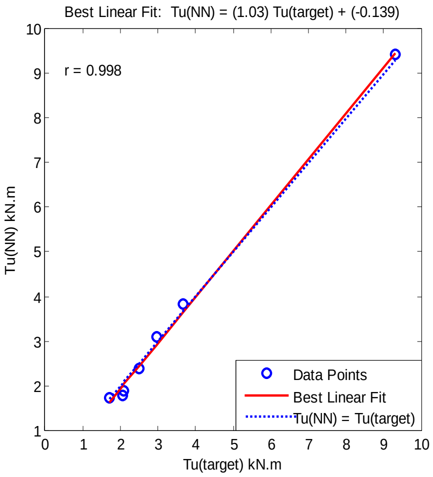

Fig. (21) Regression Analysis Based on (RPROP) for Testing Data

Based on the above analysis, the optimal network architecture (6-9) with activation functions (tansig, purelin, purelin) with (RPROP) is used for this study.

Parametric Analyses Based on Artificial Neural Network

The influence of main parameters on the ultimate torsional moment of reinforced concrete spandrel beams is studied using the proposed network and experimental results in the database.

Influence of Concrete Compressive Strength

Figure (22) shows the effect of concrete compressive strength on the ultimate torsional moment of reinforced concrete spandrel beams. It can be seen from this figure that as the concrete compressive strength increases, the ultimate torsional moment increases. For an increase in compressive strength from (30 to 48) MPa, the increase in the ultimate torsional moment is (40.87) %.

As compared with the ACI limit design which is proposed by Hsu and Burton [8], it can be seen from Fig. (22) that the increase of ultimate torsional moment, due to the increase of concrete compressive strength, is always less than that given by the neural network. For an increase in compressive strength from (30 to 48) MPa, the increase in the ultimate torsional moment according to ACI limit design is (20.95)%.

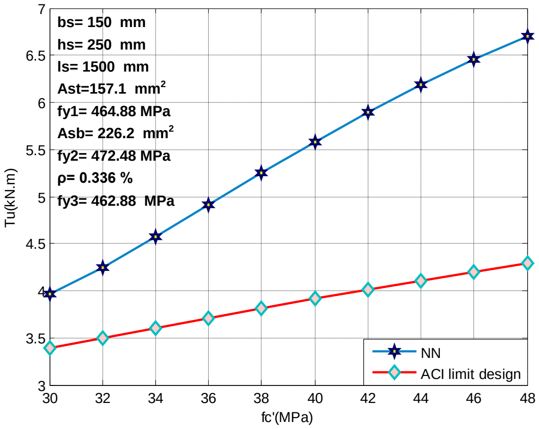

Fig. (22) Variation of Ultimate Torsional Moment with Variation of Cylinder Concrete Compressive Strength

Size Effect (Influence of Beam Width & Depth)

In Figs. (23) and (24), the ultimate torsional moment of reinforced concrete spandrel beams is plotted versus the width of spandrel beam (bs). It can be seen from the figures that the increase in width of spandrel beam leads the ultimate torsional moment to increase. From Fig. (23) for an increase in width from (130 to 165) mm, the increase in the ultimate torsional moment is (41.25) %.

The ACI limit design gives the same trend of relationship between the ultimate torsional moment and spandrel beam width, as shown in Fig. (23). However for an increase in width from (130 to 165) mm, the increase in the ultimate torsional moment according to ACI limit design is (37.9) %.

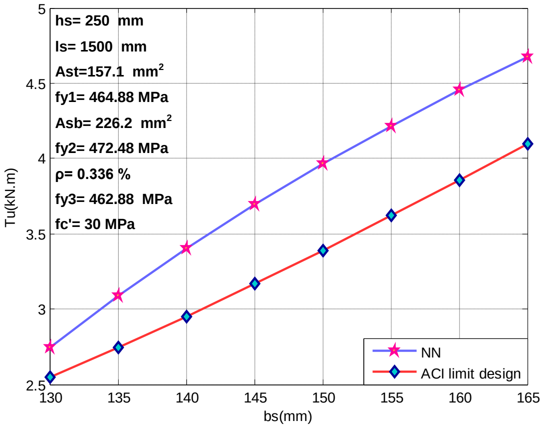

Fig. (23) Variation of Ultimate Torsional Moment with Variation of Spandrel Beam Width

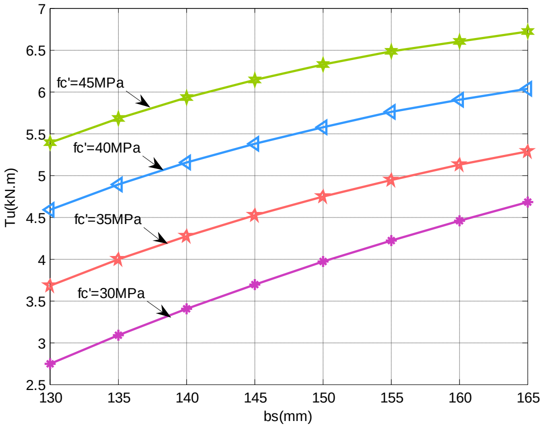

Fig. (24) Variation of Ultimate Torsional Moment with Variation of Spandrel Beam Width for Different Value of (fc´)

In Figs. (25) and (26), the ultimate torsional moment of reinforced concrete spandrel beams is plotted versus the total depth of spandrel beam (hs). It can be seen from the figures that the increase in depth of spandrel beam leads the ultimate torsional moment to increase. From Fig. (25) for an increase in depth from (240 to 275) mm, the increase in the ultimate torsional moment is (42.24)%.

The results of the ACI limit are also shown in Fig. (25). The ultimate torsional moment increases with the increase of spandrel beam depth. However, for an increase in depth from (240 to 275) mm, the increase in the ultimate torsional moment is (12.74) %.

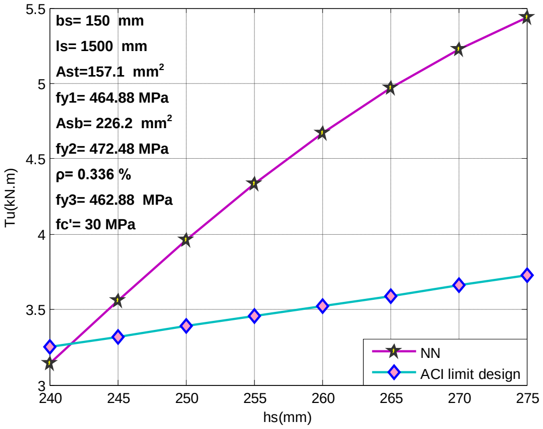

Fig. (25) Variation of Ultimate Torsional Moment with Variation of Spandrel Beam Total Depth

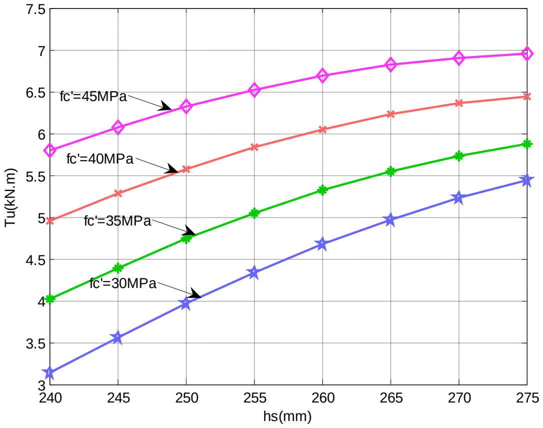

Fig. (26) Variation of Ultimate Torsional Moment with Variation of Spandrel Beam Total Depth for Different Value of (fc´)

Influence of Ratio of Web Reinforcement

Figure (27) shows the effect of the ratio of web reinforcement on the ultimate torsional moment of reinforced concrete spandrel beams. It can be seen from this figure that as the ratio of web reinforcement increases, the ultimate torsional moment increases. For an increase in ratio of web reinforcement from (0.244 to 0.915) %, the increase in the ultimate torsional moment is (17.36) %.

The artificial neural network predicts a non-linear response of spandrel beams with the amount of web reinforcement. However, the ACI limit design gives a linear response as it can be seen in Fig. (27).

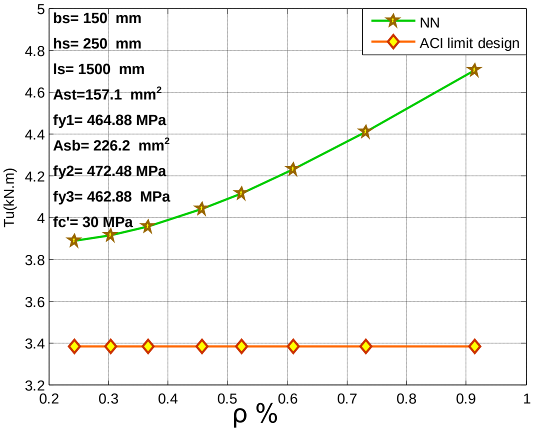

Fig. (27) Variation of Ultimate Torsional Moment with Variation of Ratio of Web Reinforcement

Conclusions

This study investigates the ability of using the artificial neural networks to evaluate the ultimate strength of reinforced concrete spandrel beam. The neural network is particularly usefulness for evaluating systems with multitude of variables. The backpropagation neural network, which is a multi-layered feedforward neural network, has been proved to accurately predict the ultimate torsional moment of reinforced concrete spandrel beams. The most important conclusions that can be drawn from the present study are the followings:

1. When using steepest descent algorithm with momentum algorithm the learning rate and momentum coefficient are effective in the training process performance and generalization capability. The convergence of the training becomes faster when the learning rate and momentum coefficient are (0.05) and (0.4) respectively.
2. Using two hidden layers in the neural networks, rather than single hidden layer, significantly improves the performance of network. The configuration (6-9) (6 nodes in the first hidden layer and 9 in the second hidden layer) is proved to be very efficient for predicting the ultimate torsional strength of spandrel beams. The final number of nodes in each hidden layer is determined by the consideration of the training time, the mapping of the neural network for the training pattern, and generalization of the neural network monitored by the test patterns.
3. The Widrow-Hoff method for initializing the weight factors and biases is found to give a minimum mean square error.
4. The neural network trained with the resilient backpropagation RPROP algorithm exhibits better behaviour than that trained with the steepest descent algorithm with momentum GDM algorithm. This was found from the reduced training time (No. of epoch) and better mapping of the neural network for the training patterns and generalization for the testing patterns.

References

1. Lu, W., *"Neural Network Model for Distortional Buckling Behaviour of Cold-Formed Steel Compression Members"*, Ph.D. Thesis, Helsinki University of Technology, June 2000, 76 pp.

2. Yeh, C., "*Analysis of Strength of Concrete Using Design of Experiments and Neural Networks"*, Journal of Materials in Civil

Engineering, Vol. 18, No. 4, August 2006, pp.597-604.

3. **Jawad, N. A. M.,** *"Strength and Behaviour of Reinforced Concrete Spandrel Beams",* M. SC. Thesis, University of Basrah, Iraq, 1988, 141 pp.
4. **Jirsa, J. O., Baumgartner, J. L. and Mogbo, N. C.,** *"Torsional Strength and Behaviour of Spandrel Beams",* ACI Structural Journal, Vol. 66, No. 11, November 1969, pp. 926-932.
5. **Nilson, A. H., Darwin, D. and Dolan, C. W.,** *"Design of Concrete Structures",* Thirteen Edition, International Edition, 2003, 779pp.
6. **Abul Mansur, M. and Rangan, B. V.,** *"Torsion in Spandrel Beams"* , Journal of the Structural Division, Vol. 104, No. ST7, July 1978, pp. 1061-1075.
7. **Muherdeen, E. H.,** *"Failure Surface of Reinforced Concrete Spandrel Beams",* M.SC. Thesis, University of Basrah, Iraq, 1991, 112 pp.
8. **Hsu, T. T. C. and Burton, K. T.,** *"Design of Reinforced Concrete Spandrel Beams",* Journal of the Structural Division, Vol. 100, No. ST1, January 1974, pp. 209-229.
9. **Strack, A. S.,** *"Strength and Behaviour of Reinforced Concrete Spandrel Beams",* M.SC. Thesis, University of Basrah, Iraq, 1987, 151 pp.

Notations

a= neural network output.
As_b= area of tension steel reinforcement.
As_t= area of compression steel reinforcement.
b= width of beam section.
bs= width of spandrel beam.
c= y-axis intercept of the best linear regression that relates the targets to the network outputs.
ds= effective depth of spandrel beam.
fc´= compressive strength of concrete cylinder.
fy_1=yielding stress of compression steel reinforcement.
fy_2=yielding stress of tension steel reinforcement.
fy_3=yielding stress of transverse reinforcement.
GDM= steepest descent with momentum.
h= total depth of beam section.
hs= total depth of spandrel beam.
I.P= inflection point.
ls= length of spandrel beam.
m= slop of the best linear regression that relates the targets to the network outputs.
MSE= mean square error.
Pu= ultimate applied load.
r= correlation coefficient.
RPROP= resilient backpropagation.
t= target.
Tu= ultimate torsional moment.
α= learning rate.
μ= momentum coefficient.
ρ= the ratio of transverse stirrup.