International Journal of the Physical Sciences Vol. 6(5), pp. 975-981, 4 March, 2011
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.023
ISSN 1992 - 1950 ©2011 Academic Journals

Full Length Research Paper

Application of artificial neural networks to predict compressive strength of high strength concrete

Seyed Jamalaldin Seyed Hakim[1]*, Jamaloddin Noorzaei[2], M. S. Jaafar[2], Mohammed Jameel[1] and Mohammad Mohammadhassani[1]

[1]Department of Civil Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia.
[2]Department of Civil Engineering, Faculty of Engineering, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia.

Accepted 18 February, 2011

A method to predict 28-day compressive strength of high strength concrete (HSC) by using MFNNs is proposed in this paper. The artificial neural networks (ANN) model is constructed trained and tested using the available data. A total of 368 different data of HSC mix-designs were collected from technical literature. The data used to predict the compressive strength with ANN consisted of eight input parameters which include cement, water, coarse aggregate, fine aggregate, silica fume, superplasticizer, fly ash and granulated grated blast furnace slag. For the training phase, different combinations of layers, number of neurons, learning rate, momentum and activation functions were considered. The training was terminated when the root mean square error (RMSE) reached or was less than 0.001 and the results were tested with test data set. A total of 30 architectures were studied and the 8-10-6-1 architecture was the best possible architecture. The results show that the relative percentage error (RPE) for the training set was 7.02% and the testing set was 12.64%. The ANNs models give high prediction accuracy, and the research results demonstrate that using ANNs to predict concrete strength is practical and beneficial.

Key words: Multilayer feedforward neural networks (MFNNs), artificial neural networks (ANNs), relative percentage error (RPE), high strength concrete (HSC), root mean square error (RMSE).

INTRODUCTION

High strength concrete (HSC) is defined as a concrete that has higher durability and strength as compared to the conventional concrete. Addition of the mineral and chemical admixture makes the HSC become a highly complex material resulting in a difficultly to model its behavior. The compressive strength of concrete is a major and important mechanical property, which is generally obtained by measuring concrete specimens after a standard curing of 28 days. Conventional methods of predicting 28-day compressive strength of concrete are basically based upon statistical analysis by which many linear and nonlinear regression equations have been constructed to model such a prediction problem (Hakim, 2006).

*Corresponding author. E-mail: jamalhakim@siswa.um.edu.my. Tel: 0060123551940.

Obviously, obtaining test values of the early strength concrete takes time and results in a delay of time in forecasting the 28-day strength. Furthermore, choosing a suitable regression equation involves technique and experience and is not a simple task. Such traditional prediction models have been developed with a fixed equation form based on a limited number of data and parameters. If the new data is quite different from the original data, then the model should update to include its coefficients and also its equation form.

ANNs do not need such a specific equation form. Instead of that, it needs sufficient input-output data. Also, it can continuously re-train the new data, so that it can conveniently adapt to the new data. ANN has been investigated to deal with problems involving incomplete or imprecise information (Noorzaei et al., 2007).

Several authors have used ANNs in structural engineering. For example, Yeh (1998), Kasperkiewicz et al. (1995), Lai and Sera (1997) and Lee (2003) applied

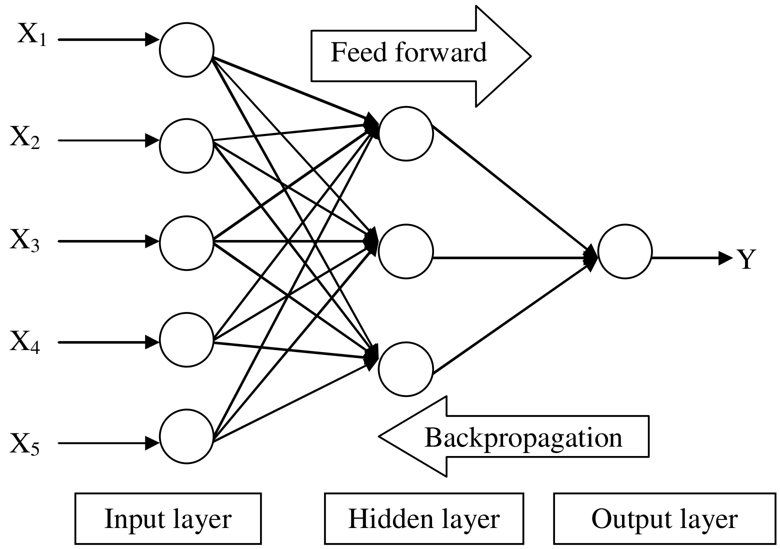

Figure 1. Architecture of a typical multilayer feed forward neural network.

the NN for predicting properties of conventional concrete and high performance concretes.

Bai et al. (2003) developed neural network models that provide effective predictive capability with respect to the workability of concrete incorporating metakaolin (MK) and fly ash (FA). Guang and Zong (2000) proposed a method to predict 28-day compressive strength of concrete by using multilayer feed forward neural networks. Dias and Pooliyadda (2001) used back propagation neural networks to predict the strength and slump of ready mixed concrete and high strength concrete, in which chemical admixtures and mineral additives were used.

Artificial neural networks

Artificial neural networks (ANNs) are data processing systems consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the central cortex of the brain. They have the ability to learn from experience in order to improve their performance and to adapt themselves to changes in the environment (Hola and Schabowicz, 2005; Mansour et al., 2004).

ANNs can provide meaningful answers even when the data to be processed include errors or are incomplete and can process information extremely rapidly when applied to solve real world problems.

As shown in Figure 1, a typical neural network has three layers: The input layer, the hidden layer and the output layer. The MFNN model is one of the most commonly used ANN models, whose application stretches to almost every field. Each neuron in the input layer represents the value of one independent variable. The neurons in the hidden layer are only for computation purpose. Each of the output neurons computes one dependent variable. Signals are received at the input layer, pass through the hidden layer, and reach the output layer.

Problem presentation

High-strength concrete is specified where reduced weight is important or where architectural considerations call for small support elements. By carrying loads more efficiently than normal-strength concrete, high-strength concrete also reduces the total amount of material placed and lower the overall cost of the structure.

HSC is a complicated mixture, the influencing material parameters are cement, aggregate, water, mineral and chemical admixture. For example, cements have different types, chemical compounds, fineness and strength. Coarse aggregates have natural, crushed and uncrushed; fine aggregates may have different size, quality and mine sources. Admixtures used in high strength concrete also have different types of chemical compounds. Moreover, the methods of mixing, testing age, loading rate during tests and transportation can also affect the durability and strength of concrete.

Pozzolans, such as fly ash and silica fume, are the most commonly used mineral admixtures in high-strength concrete. It would be difficult to produce high-strength concrete mixtures without using chemical admixtures. A common practice is to use a superplasticizer in combination with a water-reducing retarder.

The superplasticizer gives the concrete adequate workability at low water-cement ratios, leading to concrete with greater strength. The water-reducing retarder slows the hydration of the cement and allows workers more time to place the concrete.

The prediction of high strength concrete has become more complicated because several parameters and characteristics need to be considered. Determining the major parameters and the most important characteristics that influence high strength concrete, is an important discussion point highlighted in this research. In addition, focus is given to the potentials and suitability of ANN within this application.

Table 1. Range of component of data sets.

Input parameter	Minimum (kg/m^3)	Maximum (kg/m^3)
Cement	60	950
Water	72	270.8
Coarse aggregate	409	1464.6
Fine aggregate	172.2	1296
Silica fume	0	97
Superplasticizer	0	33
Fly Ash	0	280
Granulated graded blast furnace slag	0	320

Data set for neural network training and testing

368 different high strength concrete mix design data were collected from the laboratory by Professor Janusz Kasperkiewicz (1995 and 2000) in the Institute of Fundamental Technology Research of Poland through direct communication and some from the laboratory of concrete technology of the University Putra Malaysia (UPM) together with other technical papers (Yeh, 1998; Lee, 2003; Lai and Sera, 1997; Pala et al., 2005; Hola and Schabowicz, 2004, 2005). Test data were assembled for high strength concrete containing cement, coarse aggregate, fine aggregate, water, fly ash, silica fume, granulated graded blast furnace slag, and superplasticizer. These data were gathered for compressive strength of high strength concrete at 28 days and the range of compressive strength is from 40 to 140 MPa.

A neural network model was conducted, trained and tested using these available test data. Out of the 368 specimen outputs, 256 were used as training examples, and 112 were used as testing examples. Based on literature review (Mansour et al., 2004; Kim et al., 2001; Eldin and Senouci, 1994; Oztas et al., 2005; Hadi, 2003; Ashour and Alqedra, 2005). These numbers of specimens were enough for training and testing of ANN. Division of the data was carried out randomly between the two sets. The ranges of input parameters are shown in Table 1.

It is meaningful to mention that in ANNs, learning is better if the data collected is from many different fields. The scattering of input information for the training phase will affect the accuracy of a neural network. Therefore, classification of the input information is very important in the training phase.

RESULTS AND DISCUSSION

A total of 30 different architecture networks were trained in order to obtain the final developed ANN architecture in this study. In this paper, the WinNN software is used to predict the compressive strength of high strength concrete. Table 2 shows the architecture network with different conditions.

From Table 2, networks N_1- N_4 have a bigger error as compared to networks N_5-N_8 and the other two hidden layer networks. The decreased error from network N_1 to network N_8 is due to the increased connection weight (number of hidden neurons). On the other hand, the increased connection weights will cause long computational time. This is because the network needs more hidden neurons to learn the mistake and store the knowledge in the neurons.

However, too much connection weight in one hidden layer will only make the network produced over fitting the network output. Furthermore, the percentage of good patterns achieved by one hidden layer is less than the two hidden layers. For the one hidden layer network, the iteration is higher than the two hidden layers network. Therefore, the N_1-N_8 was not suitable. For the N_{27}-N_{30}, the RMSE is small but it needs a higher number of hidden neurons as compared to the other networks.

As shown in Table 2, networks N_9-N_{24} have a higher RMSE and Iteration. These networks took a longer computational time which made it more complicated. Table 2 shows that network N_{25} has the best possible result. Obviously, the network consists of two hidden layers which have a better result as compared to the one hidden layer.

This network architecture consists of 8-10-6-1; there are eight neurons (eight parameters) in the input layer, ten neurons in the first hidden layer, six neurons in the second hidden layer and one neuron in the output layer which represents the compressive strength.

This network has been chosen as the most suitable network for generalization due to its small root mean square error (RMSE) and a high percentage in good patterns as compared to other different architecture networks. Furthermore, the sigmoid of activation function in this architecture network has effectively limited the amplitude of the output neurons. This 8-10-6-1-architecture network is shown in Figure 2. After selecting the best possible architecture, the network was trained to reduce the error between the neural network output and the target output. The aim of training is to find a set of connection weights that will minimize the mean squared error forecasting error in the shortest possible training time (Kim et al., 2004). Training data is a process to minimize the RMSE between actual and estimated output values with a set of suitable connection weights.

In this study, the training is carried out using 256 data sets. It is meaningful to mention that the target error for this research will be set to 0.001. Error backpropagation training algorithm is used to minimize the output error by updating the weights.

The training process weights and biases are modified and converge towards values representing a solution of

Table 2. Comparison training result between specifications of different architectures.

R. M. S. E	No. of iteration	No. of C.W	Good pattern (100%)	A. F	M	L. R	A	N
0.013736	18073	41	66.4	Linear	0.60	0.010	8-4-1	N_1
0.007094	14990	51	78.5	Sig(x)	0.64	0.020	8-5-1	N_2
0.007811	15030	81	81.6	Tanh(x)	0.60	0.054	8-8-1	N_3
0.006216	9345	101	83.2	Sig(x)	0.65	0.040	8-10-1	N_4
0.004982	17080	131	86.7	Tanh(x)	0.54	0.060	8-13-1	N_5
0.002973	11460	151	91.8	Sig(x)	0.60	0.040	8-15-1	N_6
0.002581	19430	161	93.4	Sig(x)	0.50	0.041	8-16-1	N_7
0.002484	16390	171	93	Sig(x)	0.58	0.055	8-17-1	N_8
0.005326	8820	88	85.5	Sig(x)	0.65	0.052	8-5-6-1	N_9
0.003325	15330	116	90.6	Sig(x)	0.68	0.063	8-5-10-1	N_{10}
0.003818	11835	87	88.7	Sig(x)	0.65	0.060	8-6-4-1	N_{11}
0.003442	7920	151	91.4	Sig(x)	0.65	0.065	8-6-12-1	N_{12}
0.003148	7930	127	93.8	Sig(x)	0.67	0.069	8-7-7-1	N_{13}
0.003415	11420	145	90.6	Sig(x)	0.60	0.063	8-7-9-1	N_{14}
0.004058	10770	123	88.3	Sig(x)	0.70	0.042	8-8-5-1	N_{15}
0.001578	7470	153	96.9	Sig(x)	0.68	0.053	8-8-8-1	N_{16}
0.002505	6890	163	93.8	Sig(x)	0.66	0.070	8-8-9-1	N_{17}
0.003679	11725	126	92.2	Sig(x)	0.67	0.067	8-9-4-1	N_{18}
0.003318	16650	126	92.2	Tanh(x)	0.69	0.063	8-9-4-1	N_{19}
0.002820	12370	148	93	Sig(x)	0.62	0.055	8-9-6-1	N_{20}
0.003493	10705	159	92.2	Sig(x)	0.66	0.067	8-9-7-1	N_{21}
0.003800	7400	127	92.6	Sig(x)	0.65	0.068	8-10-3-1	N_{22}
0.004132	11830	151	91.8	Sig(x)	0.66	0.067	8-10-5-1	N_{23}
0.003412	11987	163	91.8	Tanh(x)	0.70	0.068	8-10-6-1	N_{24}
0.002988	10079	163	93	Sig(x)	0.70	0.068	8-10-6-1	N_{25}
0.005425	14635	152	88.7	Tanh(x)	0.76	0.060	8-11-4-1	N_{26}
0.002682	3840	178	92.2	Sig(x)	0.68	0.070	8-11-6-1	N_{27}
0.002912	7065	179	94.9	Sig(x)	0.60	0.130	8-12-5-1	N_{28}
0.003169	6160	178	93.8	Sig(x)	0.55	0.080	8-13-4-1	N_{29}
0.002608	8470	175	93.4	Sig(x)	0.56	0.110	8-14-3-1	N_{30}

N: Network; A: Architecture; L. R: Learning rate; M: Momentum; A. F: Activation functions for hidden and output layers; C. W: Connectivity weights; R. M. S .E: Root mean square error.

the problem. Training stops when the RMSE achieves an equal or less than value of 0.001 or the percentage of good patterns achieved is close to 100%.

The networks become unstable and oscillation occurs when the learning rate is higher than 0.10 and the momentum rate is between 0.85 and 1.0. The RMSE becomes bigger when the higher learning rate is added to the network due to that network being unable to learn or store the knowledge as the learning rate is too fast.

It is worth mentioning that, the learning rate in a parameter determines the size of the weights adjustment each time the weights are changed during training. Small values for the learning rate cause small weight changes and large values cause large changes. The learning rate has to be chosen as high as possible to allow fast learning without leading to oscillations (Yeh et al., 1992; Kim, 2001). The value of learning rate ranges between “0.0” and “1.0” where a value closer to 1 indicates significant modification in weight while a value closer to 0 indicates little modification (Okine and Fekpel, 1996).

For the momentum, it is added to the network to achieve a higher percentage in good patterns. Moreover, it helps avoid oscillatory entrapment in the local minima and achieve to global minima. Learning rate and momentum interact with each others, so several different conditions of networks are run to check the accuracy.

The network with the learning rate in the range 0.01 to 0.05 and momentum in the range 0.40 to 0.60, give a bigger RMSE and also the highest iteration. Therefore, this range is not suitable for the network generalization. The best possible range found was 0.6 to 0.8 for the learning rate and the momentum range fall within 0.65 to 0.75. Besides this, the activation function is another important parameter for the layers. The aim of an activation function is to limit the amplitude of the output neurons. The nonlinearity degree of an activation function

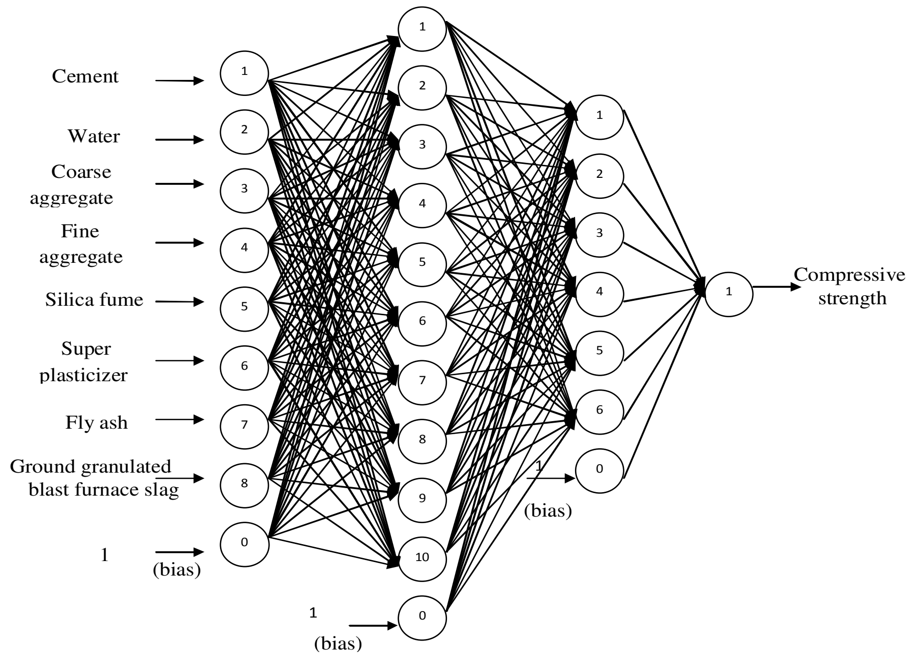

Figure 2. Final architecture of the developed artificial neural network.

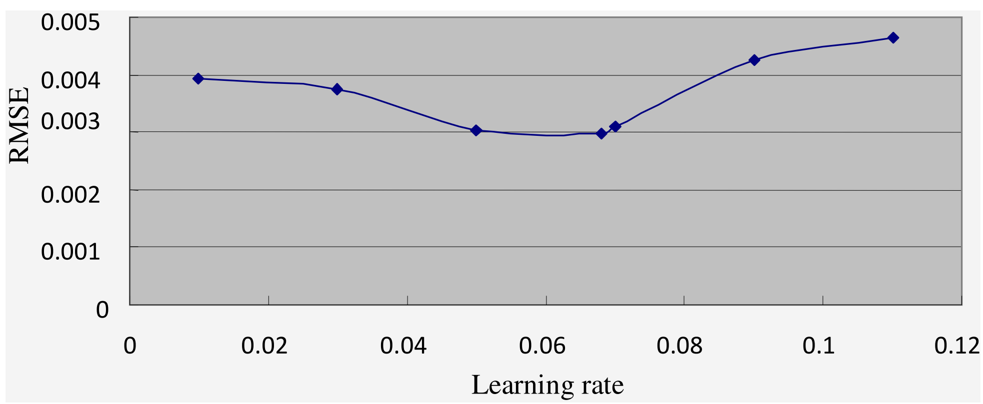

Figure 3. Variation of RMSE versus rates of learning.

is valuable for most ANN applications. In Table 2, networks N_1, N_3, N_5, N_{19}, N_{24} and N_{26} have the worse result with a large RMSE, long computational time and a lower percentage in good patterns than the other networks.

To make sure the 8-10-6-1 architecture network is accurate, another different condition network was run to check the variation of the RMSE versus learning rate. The result in Figure 3 shows that the range 0.05 to 0.075 of learning rate is more accurate due to its smaller RMSE.

Furthermore, a variation of RMSE versus rates of momentum is also plotted in Figure 4; it shows that the lower and higher momentum rate is gives a larger RMSE. A suitable momentum is in the range 0.65 to 0.75 which gives a smaller RMSE.

Finally, this network is trained with the 0.068 rate of learning, 0.7 for momentum, 20000 iteration and sigmoid as the activation function for both the hidden layer and output layer. In the ANNs, the training data is the most important source to determinate the suitability of the network for the generalization. The accuracy of the training data will decrease the RMSE. Besides that, the testing data also can not be neglected because the testing set is used to avoid over-training and to evaluate the confidence in the performance of the trained network.

As Figure 5 shows, the predicted and experimental data in the training process seem very accurate. The relative percentage error (RPE) of the training process is 7.02%. This result shows that the artificial neural network was very successful in predicting the compressive strength.

After the training process, the neural networks will be tested with another data set, which has not been used for

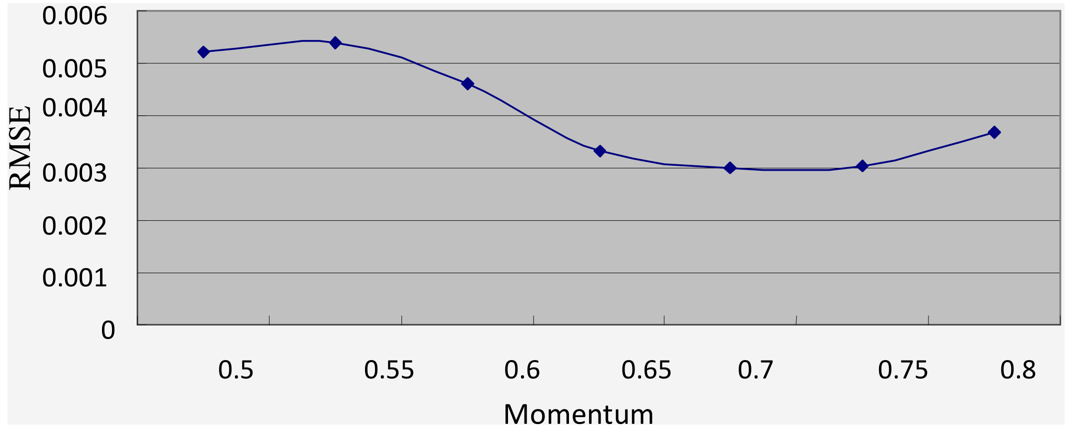

Figure 4. Variation of RMSE versus rates of momentum.

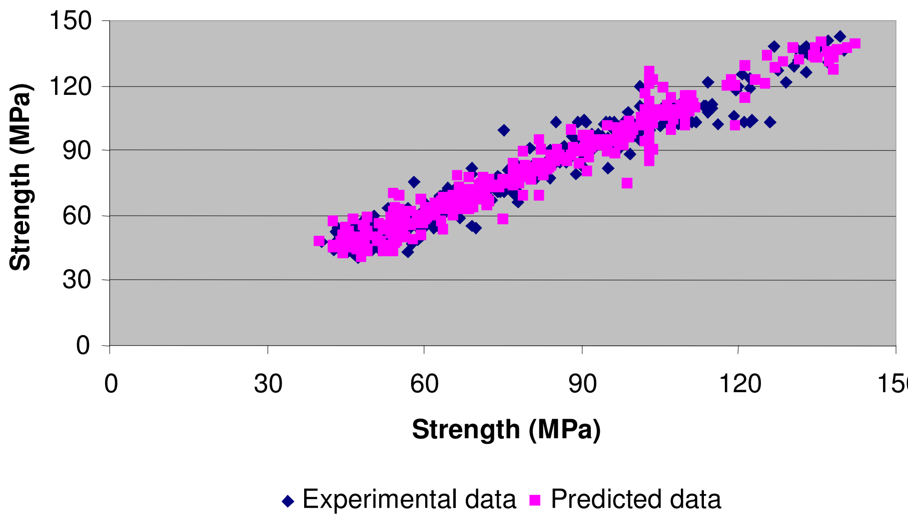

Figure 5. Comparison of experimental and predicted data in training process.

the previous training process. In the testing phase, 112 data sets are carried out. The aim of this verification is to guard against overtraining, where the ANN has memorized or over-fitted the connection weights to the training patterns. The testing set is used to evaluate the confidence in the performance of the trained network. The result of the testing process is shown in Figure 6. The predicted data has a more difference in percentage of RMSE with the experimental data. This is maybe because the data collected comes from a limited area study field or source.

Furthermore, it might be due to some of the testing patterns that did not fall in the class of the training space. Also, the testing process results to bad output (compressive strength with big error) usually occurring form the bad data set within the range (100 to 140 Mpa). An explanation for these results could be that there is an insufficient amount of training data around this range. For the testing process the RPE is 12.64%, this percentage is considered to be a small RMSE. Based on the available literatures, the value of error achieved (12.64%) for testing data is acceptable; however, the aim was to achieve better results as close as possible to the training results. So, these results can be accepted for the prediction of the compressive strength of high strength concrete.

Conclusions

The main conclusions drawn from this study are as follows:

1. The 8-10-6-1 architecture network is better than other architecture networks where there are eight neurons in the input layer, ten neurons in the first hidden layer, six

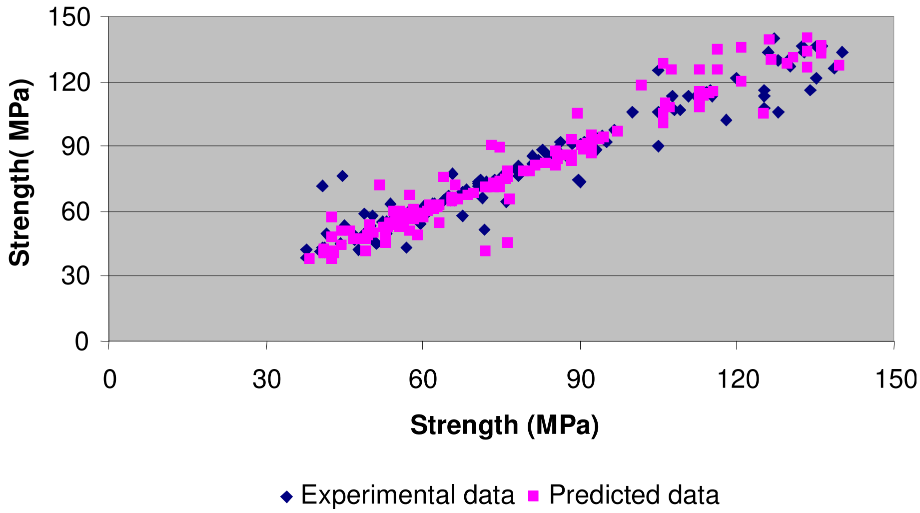

Figure 6. Comparison of experimental and predicted data in testing process.

neurons in the second hidden layer and one neuron in the output layer.

2. Calculation of the mean percentage relative error for training and testing set data shows that the ANN predicted the high compressive strength of concrete with an error of 7.02 and 12.64%, respectively; these are acceptable in concrete technology.

3. The results prove that ANNs can work efficiently in predicting the high compressive strength of concrete and is more accurate than the model using the regression analysis and conventional methods. Also, from the results obtained, it can be concluded that the ANNs can save a lot of computational effort compared to conventional methods significantly. The use of these networks will help in solving more complex problems.

REFERENCES

Ashour AF, Alqedra MA (2005). Concrete breakout strength of single anchors in tension using neural networks. J. Adv. Eng. Software, 36: 87-97.

Bai J, Wild S, Ware JA, Sabir BB (2003). Using neural networks to predict workability of concrete incorporating metakaolin and fly ash. J. Adv. Eng. Software, 34: 663–669.

Dias WPS, Pooliyadda SP (2001). Neural networks for predicting properties of concretes with admixtures. J. Constr. Build. Mater., 15: 371–379.

Eldin NN, Senouci AB (1994). Measurement and prediction of the strength of rubberized concrete. J. Cem. Concrete Comp., 16: 287-298.

Guang NH, Zong WJ (2000). Prediction of compressive strength of concrete by neural networks. J. Cem. Concrete Res., 30: 1245-1250.

Hadi M (2003). Neural networks applications in concrete structures. J. Comp. Struct., 81: 373-381.

Hakim SJS (2006). Development and applications of artificial neural network for prediction of ultimate bearing capacity in soil and compressive strength of concrete. Master Thesis, Department of Civil Engineering, University Putra Malaysia, Malaysia.

Hola J, Schabowicz K (2005). Application of artificial neural networks to determine concrete compressive strength based on non-destructive tests. J. Civil Eng. Manage., 11(1): 23-32.

Hola J, Schabowica K (2004). New technique of nondestructive assessment of concrete strength using artificial intelligence. J. NDT&E Int., 38: 251-259.

Kasperkiewicz J (2000). The application of ANNs in certain materials analysis problems. J. Mater. Process. Tech., 106: 74-79.

Kasperkiewics J, Racz J, Dubrawski A (1995). HPC strength prediction using ANN. ASCE. J. Comp. Civil Eng., 4: 279–284.

Kim JI, Kim DK, Feng MQ, Yazdani F (2004). Application of neural networks for concrete strength. J. Mater. Civil Eng., 16(3): 257-264.

Kim CY, Bae GJ, Hong SW, Park CH, Moon HK, Shin HS (2001). Neural network based prediction of ground surface settlements due to Tunneling. J. Comp. Geotech., 28: 517–547.

Lai S, Sera M (1997).Concrete strength prediction by means of neural network. J. Constr. Build. Mater., 11(2): 93–98.

Lee SC (2003). Prediction of concrete strength using artificial neural networks. J. Eng. Struct., 25: 849–857.

Mansour MY, Dicleli M, Lee YJ, Zhang J (2004). Predicting the shear strength of reinforced concrete beams using artificial neural networks. J. Eng. Struct,. 26: 781–799.

Noorzaei J, Hakim SJS, Jaafar MS, Abang AAA, Thanon WAM (2007). An optimal architecture of artificial neural network for predicting compressive strength of concrete. J. Indian Concrete, 81(8): 17-24.

Okine NOA, Fekpel ESK (1996). Strength characteristics modeling of lateritic soils using adaptive neural networks. J. Constr. Build. Mater., 10(8): 577-582.

Oztas A, Pala M, Ozbay E, Kanca E, Caglar N, Bhatti MA (2005). Predicting the compressive strength and slump of high strength concrete using neural network. J. Constr. Build. Mater., 20(9): 769-775.

Pala M, Ozbay E, Oztas A, Yuce MI (2005). Appraisal of long-term effects of fly ash and silica fume on compressive strength of concrete by neural networks. J. Constr. Build. Mater., 21: 384-394.

Yeh IC (1998). Modeling of strength of HPC using ANN. Cem. Concrete Res. J., 28(12): 1797–1808.

Yeh YC, Kuo YH, Hsu, DS (1992). Building an expert system for debugging FEM input data with artificial neural networks. J. Expert Syst. Appl., 5: 59-70.