DEVELOPMENT OF ARTIFICIAL NEURAL NETWORKS FOR PREDICTING CONCRETE COMPRESSIVE STRENGTH

J. Noorzaei, S.J.S. Hakim, M.S. Jaafar, and W.A.M.Thanoon
Department of Civil Engineering, Universiti Putra Malaysia, 43400 UPM-Serdang, Malaysia
Email: jamal@eng.upm.edu.my

ABSTRACT

This research work focuses on development of Artificial Neural Networks (ANNs) in prediction of compressive strength of concrete after 28 days. To predict the compressive strength of concrete six input parameters that are cement, water, silica fume, super plasticizer, fine aggregate and coarse aggregate are identified. A total of 639 different data sets of concrete was collected from the technical literature. Training data sets comprises 400 data entries, and the remaining data entries (239) are divided between the validation and testing sets. Different combinations of layers, number of neurons, activation functions, different values for learning rate and momentum were considered and the results were validated using an independent validation data set. A detailed study was carried out, considering two hidden layers for the architecture of neural network. The performance of the 6-12-6-1 architecture was the best possible architecture. The MSE for the training set was 5.33% for the 400 training data points, 6.13% for the 100 verification data points and 6.02 % for the 139 testing data points. The results of the present investigation indicate that ANNs have strong potential as a feasible tool for predicting the compressive strength of concrete.

Keywords**:** *artificial neural networks (ANNs), back propagation (BP), compressive strength, mean squared error (MSE).*

INTRODUCTION

Artificial neural networks (ANNs) are a family of massively parallel architectures that are capable of learning and generalizing from examples and experience to produce meaningful solutions to problems even when input data contain errors and are incomplete. This makes ANNs a powerful tool for solving some of the complicated engineering problems. Basically, the processing elements of a neural network are similar to the neuron in the brain, which consists of many simple computational elements arranged in layers.

The basic strategy for developing a neural network-based model for material behavior is to train a neural network on the results of a series of experiments using that material. If the experimental results contain the relevant information about the material behavior, then the trained neural network will contain sufficient information about material's behavior to qualify as a material model (Hakim, Mesri and Selaru) [1,2,3]. Such a trained neural network not only would be able to reproduce the experimental results, but also it would be able to approximate the results in other experiments through its generalization capability.

A compressive strength of concrete is a major and important mechanical property, which is generally obtained by measuring concrete specimen after a standard curing of 28 days. Concrete strength is influenced by lots of factors. Some of these parameters include quality of aggregate, strength of cement, water content and water-to-cement ratio. The traditional approach used in modeling the effects of these parameters on the compressive strength of concrete starts with an assumed form of analytical equation and is followed by a regression analysis using experimental data to determine unknown coefficients in the equation, Dias [4].

Unfortunately, rational and easy-to-use equations are not yet available in design codes to accurately predict the compressive strength of concrete. Also, the current empirical equations presented in the codes and standards for estimating compressive strength are based on tests of concrete without supplementary cementitious materials,Yeh, [5].The validity of these relationships for concrete with supplementary cementitious materials (fly ash, silica fume, super plasticizer, etc.) should be investigated.

Guang and Zong, [6] proposed a method to predict 28-day compressive strength of concrete using multilayer feedforward neural networks. Dias ,[4] presented an artificial neural network model for predicting the strength and slump of ready mixed concrete. Wang [7] developed an automatic knowledge acquisition system based on neural networks to design concrete mix. Application of neural networks for estimation of concrete strength was presented by Kim, [8].

Eldin and Senouci, [9] employed a neural network for measuring and predicting of the strength of rubberized concrete. Employing the artificial neural network method in modeling of strength of high performance concrete is shown by Yeh, [5]. Hola [10] determined concrete compressive strength based on non-destructive tests using artificial neural network. Lee, [11] developed the intelligent prediction system of concrete strength that provides in-place, information on compressive strength of concrete. Mansour [12] applied the ANNs for predicting the shear strength of reinforced concrete beams. Lai [13] predicted the mechanical properties of concrete by ANNs.

Predicting of compressive strength of concrete using ANNs is the aim of this study. For this aim, a computer program was developed in Qbasic. Using this program, a neural network model can be constructed, trained and tested using the available test data of 639 different concrete mix-designs gathered from the technical literature. The data used in neural network model are arranged in a format of six input parameters that cover the cement, water, silica fume, super plasticizer, fine aggregate and coarse aggregate. The proposed neural network model predicts the compressive strength of concrete.

DEVELOPMENT OF ANN FOR CONCRETE STRENGTH

Data Selection

In order to develop ANN architecture, 639 samples of concrete data on 28th day of compressive strength of concrete were collected. In the present work, training data set comprises 400 data entries, and the remaining data entries (239) are divided between the validation and testing sets. To test the reliability of the neural network model, 139 samples were randomly selected as the test set and 100 samples as the validation set. The dividing process was carried out randomly between the three sets and each dataset has been statistically examined to ensure that it covers the range of input parameters.

These data was collected from laboratory of concrete of University Putra Malaysia , different papers, Dias [4], Yeh [5], Hola [10], Lee, [11], Lai [13], Hola [14] and Pala [15] and some from laboratory data by Kasperkiewicz and co-workers [16, 17] in institute of fundamental technology research of Poland. These data was collected for compressive strength of concrete after 28 days including weight per m^3 of each concrete component. In a neural network if the area for data is more, learning is better. The accuracy of a neural network depends on the scattering of input information for training of the network. For this reason, classification of input information is very important in training. Therefore the input information is classified in six cases and in each case classification is based on one of the concrete components. The ranges of input parameters are in Table 1.

Construction of Neural Network Model and Parameters

The architecture of a network describes how many layers a network has, the number of neurons in each layer, each layer's activation function, and how the layers connect to each other. The best architecture to use depends on the type of problem to be represented by the network Ince [18], Bazier [19]. Selecting an optimal ANN architecture is an open problem of investigation and depends on the application domain. In the present study there are six inputs and compressive strength of concrete is output. For this reason, the initial structure of neural network is illustrated Figure 1.

The architecture of neural network was determined by training, testing and validating of 25 networks having different conditions as tabulated in Table 2. As it is seen in this table, several architecture of neural network models were examined by varying the number of hidden layers, number of neurons in each hidden layer, type of activation function, value of learning rate and value of momentum term.

It is clear from Table 2 that each network has been trained with both one hidden layer and two hidden layers and it were shown significant difference in terms of accuracy and computational time required for learning. It is found that in ANN with two hidden layer, the time for computing reduced to arrive an error of 0.007. It is obvious in columns 2 and 9 of Table 2, that ANN model with one hidden layer i.e., networks N_1 to N_4, MSE will be very high and in networks N_5 to N_{10}, MSE is smaller, but number of connectivity is too big and computational time is long, accordingly, the neural network did not give good generalization of the network, indicating that more neurons are required. For example in network N_{10}, sixty hidden neuron for connecting of

481 weights is needed. However, adding more neurons to the only one hidden layer produced over fitting of the network output.

It is seen in columns 2, 7, 8 and 9 of Table 2, in architectures with two hidden layers MSE is less than when there is one hidden layer. However the iterations taken by one hidden layers networks were more than those by two hidden layer networks. This problem was finally overcome by introducing a second hidden layer with six neurons connected to the first hidden layer. Eventually, the back propagation in this study is restricted to two hidden layers, which yields a total of four layers. So, the networks N_1 to N_{10} in Table 2 were not acceptable. Moreover it is clear from this table that in networks N_{17} to N_{25} error is small, but numbers of hidden neurons and number of iteration are too high. In these networks numbers of connectively are too much and computational are very complicated and take long computational time. For examples networks N_{17}, N_{24} and N_{25} have 361, 851 and 691 connectivity weights, respectively. To arrive to acceptable convergence number of iteration were recorded 25000, 100000 and 70000, respectively.

It can be seen in this table, which networks N_{11} to N_{16}, have acceptable error with logical iterations and small hidden neurons. Table 2 shows that network N_{15} have the best possible architecture. In this network, after 10000 iterations, the required convergence is arrived.

The architecture of this network is 6-12-6-1, i.e. there are six neurons in the input layer corresponding to the six factors (six components of an input vector), two hidden layers that twelve in the first hidden layer and six neurons in the second hidden neurons and one neuron in the output layer corresponding to 28th day compressive strength. The final architecture for this network is presented in Figure 2.

Training Phase

Training of the neural network is carried out using 400 data sets. It is worth mentioning that in this study, the training process was terminated when any of the following conditions are satisfied:

i) The maximum number of iterations is reached to 10000.
ii) The mean square error of the training set is reached to 0.007**.**
iii) The mean square error of the training data sets starts diverges after 5000 number of iteration**.**

One of the important variables in network design is the learning rate coefficient. Each time a pattern is presented to the network, the weights leading to a neuron are modified slightly during learning in the direction required to produce a smaller error at the outputs the next time the same pattern is presented. The amount of weight modification is proportional to the learning rate. 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 Al-khaleefi, [20].

However, the learning rate in a parameter that 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 best learning rate is not obvious. If the learning rate is 0.0, the network will not learn. The learning rate is very important in identifying over-learning and when to stop training Al-khaleefi [20].

In process of training, it was seen that for the range of (0.05-0.1) the learning rate, convergence was faster and number of iterations was less than other range. However, when the learning rate increased, the iterations number jumps to a divergence point and training doesn't converge even in 10000 training iterations. The effect of the learning rate on the total number of iterations and on the speed of convergence was examined.

The ANN model trained against different values of learning rate. Each value has been given to the network and MSE value was determined. The results for learning rate are shown in Table 2.It is clear from Table 2 that to achieve low MSE when learning rate is in range (0.01-0.06), number of iterations increased drastically, which indicates the ANN model, is not computationally efficient. In this research work, also for learning rate value more than 0.2, network was unstable and has highly oscillations. As it is obvious from Table 2 that learning rate in range of (0.06-0.08) has a minimum error and minimum number of iteration for convergence without any oscillation. Furthermore the value of 0.06 for learning rate has minimum error as compared of to learning rate 0.07 and 0.08, but this difference is not significant. Hence in this research, for avoiding any oscillation the best value for learning rate that have above conditions, is 0.08. Effect of different learning rates against MSE is plotted in Figure 3.

For preventing unstable and oscillation network, is added in back propagation algorithm a value that is called momentum. The momentum term adds inertia to the training procedure, and helps avoid oscillatory entrapment in local minima [20].In this present, the network first checked for range of (0.8-1) for momentum and was seen over oscillations and very big error under high momentum range. After that, under low momentums ranges (0.3-0.5), with considering 0.07 for learning rate is needed to very long computational times. However, as it is clear in Table 2, rang (05-0.7) for momentum, give good results. Variations of MSE versus rates of momentum, is illustrated in Figure4. Based on the results tabulated in Table 2 and plotted via Figure 4, value for momentum rate is selected 0.65. It can be seen in this Figure that value of error in momentum rate of 0.6 is less than value of 0.65 for momentum, but this difference in error isn't considerable. Hence, for preventing of oscillation, value of 0.65 is selected.

The summary of the evaluation learning rate, momentum and number of iterations versus MSE are given in Table 3. It is obvious from this table that for the learning rate above 0.2, the ANN model is unstable. In summary, a learning factor equal to 0.08 and momentum rate equal to 0.65 was set for the training of the network. An activation function is used for producing the neuron output and limiting the amplitude of the output of a neuron. It determines the relationship between inputs and outputs of a neuron and a network. In general, the activation function introduces a degree of nonlinearity that is valuable for most ANN applications Al-khaleefi [20]. Decision on activation function for layers is another important parameter.

To arrive at optimum activation functions, the constructed ANN system executed with different activation functions. In this investigation, as shown in Table 4 the value of mean square error in the last of 500^{th} epoch and 10 times, is compared for sigmoid, tangent hyperbolic and linear activation functions. It is worth mentioning that in this present work, determination of activation function by keeping the sigmoid function in output layer and assumes different activation function for hidden layers. The result shows that on average, the sigmoid function performs better than the other two activation function.

Sigmoid function on average reaches lower MSE rates than the other activation functions. For this reason the sigmoidal transfer function is implemented between the input and hidden layers and also is selected in output layer. Hence, with 0.08 and 0.65 for learning rate and momentum, respectively and selection of sigmoid activation function in hidden and output layers, the developed ANN is trained for the 400 sets of data the predicted values are compared.

Figure 5 shows the comparison of compressive strength of concrete predicted by the developed ANN and that of actual compressive strength of concrete. It is obvious from this plot that there is a good agreement between the predicted and actual compressive strength of concrete. It was found that the forecasting error was 5.33% in 10000 iterations for these particular trial runs. The results shown in this Figure are fairly reasonable, since both the test and validation set errors have very similar characteristics, and no significant over fitting has occurred. In the training process, weights and biases are constantly adjusted to minimize the error between the actual and the predicted outputs of the unit in the output layer.

Testing Phase

Next step after training process in the development of the ANN is to test the developed ANN model. After the network was trained in the 400 training cases, is used the testing set to avoid over-training and to evaluate the confidence in the performance of the trained network with a training set for a 10000 number of iterations. The testing process has been carried out for a total 139 data sets. Figure 6 exhibits the comparison between predicted compressive strength of concrete against the experimental evidence, which highlighted that there is a good agreement between the predicted values and that of experiment data.

The results are shown that the artificial neural network was very successful in predicting of compressive strength of error with MSE of 6.02 percent. Also the ANN predicts the compressive strength of concrete in testing stage reasonably well the 6-12-6-1 neural models in general performs better than the others and it is able to give accurate prediction of compressive strength of concrete. In testing, result showed that, bad output (compressive strength with big error) usually occurred form the bad data set in range (0-30). One explanation for these results could be that there is an insufficient amount of training data around this range.

Validation Phase

The validation set is used to as a further check for the generalization of the Neural Network, but do not have any effect on the training. In the validation phase, the ANN accuracy is examined using the validation set.
The plot of predicted compressive strength of concrete in validation sets (100) against experimental data depicted is shown in Figure7. It is obvious from this plot that is reasonably good agreement between the results predicted and target results. These results show that the artificial neural network was successful in training the relationship between the input and output data with the MSE of 6.13 Percent (MSE for validation set).

Comparison of Training, Testing and Validation

The progress of the training was checked by plotting the training, validation and test mean square error versus the performed number of iterations, as presented is shown in Figure 8. The results in this Figure indicate that the neural network was successful in learning the relationship between the different input parameters and output (compressive strength).

CONCLUSION

In this study, a neural network model for prediction of compressive strength of concrete was developed. The study suggests that the use of neural networks has several significant advantages over other conventional methods. The following summarizes the findings of the study.

(i) The performance of the 6-12-6-1 architecture was better than other architectures. That means, there are six neurons in the input layer corresponding to the six factors, two hidden layers that twelve in the first hidden layer and six neurons in the second hidden neurons and one neuron in the output layer corresponding to 28^{th} day compressive strength.
(ii) The learning parameters as 0.08 and momentum parameter as 0.65 gave a best possible results for training of network.
(iii) The MSE for the training set was 5.33% for the 400 training data points, 6.13 for the 100 verification data points and 6.02 % for the 139 testing data points.
(iv) The results obtained from the developed computer program were compared with results from experimental studies. The comparisons of results indicate good agreements. From the results obtained artificial neural network, it can recognize the concrete in term of ‘strength’ with a confidence level of about 95%, which is considered as satisfactory from an engineering point of view.

REFERENCE

[1] Hakim SJS (2006) Development of artificial neural network with application to some civil engineering problems. Master thesis, Civil Engineering Department, University Putra Malaysia,.

[2] Masri SF, Chassiakos AG, and Caughey TK (1993) Identification of nonlinear dynamic systems using neural networks. Journal of Applied Mechanics; **60**: 123-33.

[3] Selaru FM, Xu Y, Yin J, Zou T, Liu TC, Mori Y, Abraham JM, Sato F, Wang S, Twigg C, Olaru A, Shustova V, Leytin A, Hytiroglou P, Shibata D, Harpaz N, Meltzer SJ (2002) Artificial neural networks distinguish among subtypes of neoplastic colorectal lesions. Journal of Gastroenterol; **122**: 606-613.

[4] Dias WPS and Pooliyadda SP (2001) Neural networks for predicting properties of concretes with Admixtures. Journal of Construction and Building Materials; **15**: 371-379.

[5] Yeh IC (1998) Modeling of strength of high-performance concrete using artificial neural networks. Journal of Cement and Concrete Research; **28**(12): 1797–1808.

[6] Guang NH, Zong W.J (2000). Prediction of compressive strength of concrete by neural networks. Journal of Cement and Concrete Research; **30**: 1245-1250.

[7] Wang JZ, Ni HG, He JY (1999).The application of automatic acquisition of knowledge to mix design of concrete. Journal of Cement Concrete Research; **29**: 1875-1880.

[8] Kim JI, Kim DK, Feng MQ, Yazdani F (2004). Application of neural networks for concrete strength. Journal of Materials in Civil Engineering; **16**(3): 257-264.

[9] Eldin NN and Senouci AB (1994) Measurement and prediction of the strength of rubberized concrete. Journal of Cement and Concrete Composites; 16:287-298.

[10] Hola J and Schabowicz K. (2005) Application of artificial neural networks to determine concrete compressive strength based on Non-destructive tests. Journal of Civil Engineering and Management; **11**(1): 23-32.

[11] Lee SC (2003). Prediction of concrete strength using artificial neural networks. Journal of Engineering Structure; **25**: 849-857.

[12] Mansour MY, Dicleli M, Lee YJ, Zhang J (2004). Predicting the shear strength of reinforced concrete beams using artificial neural networks. Journal of Engineering Structures; **26**: 781–799.

[13] Lai S, Serra M (1997) Concrete strength prediction by means of neural network. Journal of Construction and Building Materials; **11**(2): 93-98.

[14] Hola J, Schabowicz K. (2004) New technique of nondestructive assessment of concrete strength using artificial intelligence. Journal of NDT&E International 2004.

[15] 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. Journal of Construction and Building Materials 2005.

[16] Kasperkiewicz J, Racz J, Dubrawski A (1995). HPC strength prediction using artificial neural network. Journal of Computer in Civil Engineering. **9**(4): 279-284.

[17] Kasperkiewicz J (2000) The application of ANNs in certain materials analysis problems", Journal of Materials Processing Technology, **106**: 74-79.

[18] Ince R (2004). Prediction of fracture parameters of concrete by artificial neural networks. Journal of Engineering Fracture Mechanics. **71**: 2143-2159.

[19] Baziar MH, Ghorbani A (2005). Evaluation of lateral spreading using artificial neural networks. Journal of Soil Dynamics and Earthquake Engineering. **25**: 1-9.

[20] Al-khaleefi AM, Terro MJ, Alex AP, Wang Y (2002). Prediction of fire resistance of concrete filled tubular steel columns using neural networks. Fire Safety Journal; **37**: 339-352.

Table1: Range of Input Parameters in database

Input Parameters	Minimum(MPa)	Maximum(MPa)
Cement	94	900
Water	50	650
Silica fume	0	300
Super plasticizer	0	40
Fine aggregate	0	1600
Coarse aggregate	0	2000

Table 2: Comparison between specifications of different architectures

Network	Architecture	L.R	M	A.F for Hidden Layers	A.F for Output Layer	No. of C.W	No. of Iteration	M.S.E
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
N_1	6-10-1	0.005	0.55	Linear	Linear	81	50000	0.084973
N_2	6-15-1	0.03	0.6	Sig(x)	Sig(x)	121	10000	0.012291
N_3	6-20-1	0.04	0.6	Sig(x)	Sig(x)	161	10000	0.011574
N_4	6-25-1	0.06	0.6	Tanh(x)	Tanh(x)	201	10000	0.013439
N_5	6-30-1	0.1	0.7	Sig(x)	Sig(x)	241	10000	0.009273
N_6	6-35-1	0.08	0.7	Sig(x)	Sig(x)	281	10000	0.008824
N_7	6-40-1	0.08	0.65	Tanh(x)	Tanh(x)	321	10000	0.011354
N_8	6-45-1	0.1	0.7	Sig(x)	Sig(x)	361	10000	0.009073
N_9	6-50-1	0.08	0.7	Sig(x)	Tanh(x)	401	30000	0.008745
N_{10}	6-60-1	0.1	0.7	Sig(x)	Sig(x)	481	20000	0.008276
N_{11}	6-5-5-1	0.06	0.65	Tanh(x)	Sig(x)	71	10000	0.009846
N_{12}	6-6-6-1	0.04	0.6	Sig(x)	Sig(x)	101	10000	0.009248
N_{13}	6-8-8-1	0.08	0.7	Sig(x)	Sig(x)	137	10000	0.009073
N_{14}	6-10-10-1	0.05	0.7	Sig(x)	Sig(x)	181	10000	0.007692
N_{15}	**6-12-6-1**	**0.08**	**0.65**	**Sig(x)**	**Sig(x)**	**169**	**10000**	**0.007531**
N_{16}	6-12-12-1	0.08	0.7	Sig(x)	Sig(x)	253	15000	0.007339
N_{17}	6-15-15-1	0.06	0.7	Sig(x)	Sig(x)	361	25000	0.007103
N_{18}	6-16-8-1	0.01	0.5	Tanh(x)	Tanh(x)	257	30000	0.008247
N_{19}	6-8-16-1	0.04	0.6	Tanh(x)	Sig(x)	217	20000	0.007648
N_{20}	6-10-5-1	0.06	0.65	Sig(x)	Sig(x)	131	10000	0.007987
N_{21}	6-14-7-1	0.08	0.7	Sig(x)	Sig(x)	211	10000	0.007029
N_{22}	6-18-18-1	0.08	0.7	Sig(x)	Sig(x)	487	30000	0.006893
N_{23}	6-20-20-1	0.06	0.7	Sig(x)	Sig(x)	581	40000	0.006732
N_{24}	6-25-25-1	0.1	0.75	Sig(x)	Sig(x)	851	100000	0.006624
N_{25}	6-30-15-1	0.02	0.5	Sig(x)	Sig(x)	691	70000	0.006849

L.R=Learning Rate, M=Momentum, A.F=Activation Function,
CW=Connectivity Weights, M.S.E=Mean Square Error

Table 3: Comparison between learning parameters and MSE

Learning Rate	Momentum Rate	MSE	Iterations
0.01	0.45	0.007451	50000
0.03	0.55	0.007492	40000
0.04	0.60	0.007431	35000
0.06	0.65	0.007758	35000
0.08	0.70	0.007963	20000
0.10	0.75	0.009124	10000
0.20	0.80	0.0112749	5000
0.30	0.75	Unstable	
0.40	0.70		
0.50	0.70		

Table 4: Influence type of activation function on value of network error in training

Method	Run										
	1	2	3	4	5	6	7	8	9	10	Mean
Sigmoid	.035	.039	.159	.033	.053	.039	.029	.035	.030	.031	.0483
Tanh(x)	.028	.062	.032	.040	.036	.034	.030	.048	.346	.181	.0838
linear	.066	.085	.252	.183	.102	.061	.264	.058	.067	.145	.1287

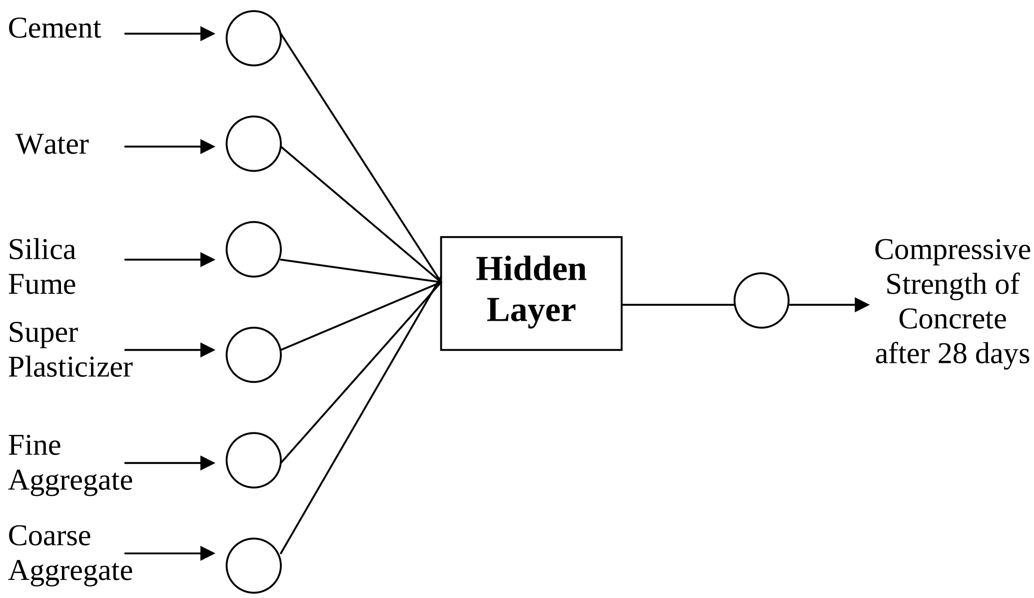

Figure 1: Initial Neural Network Structure

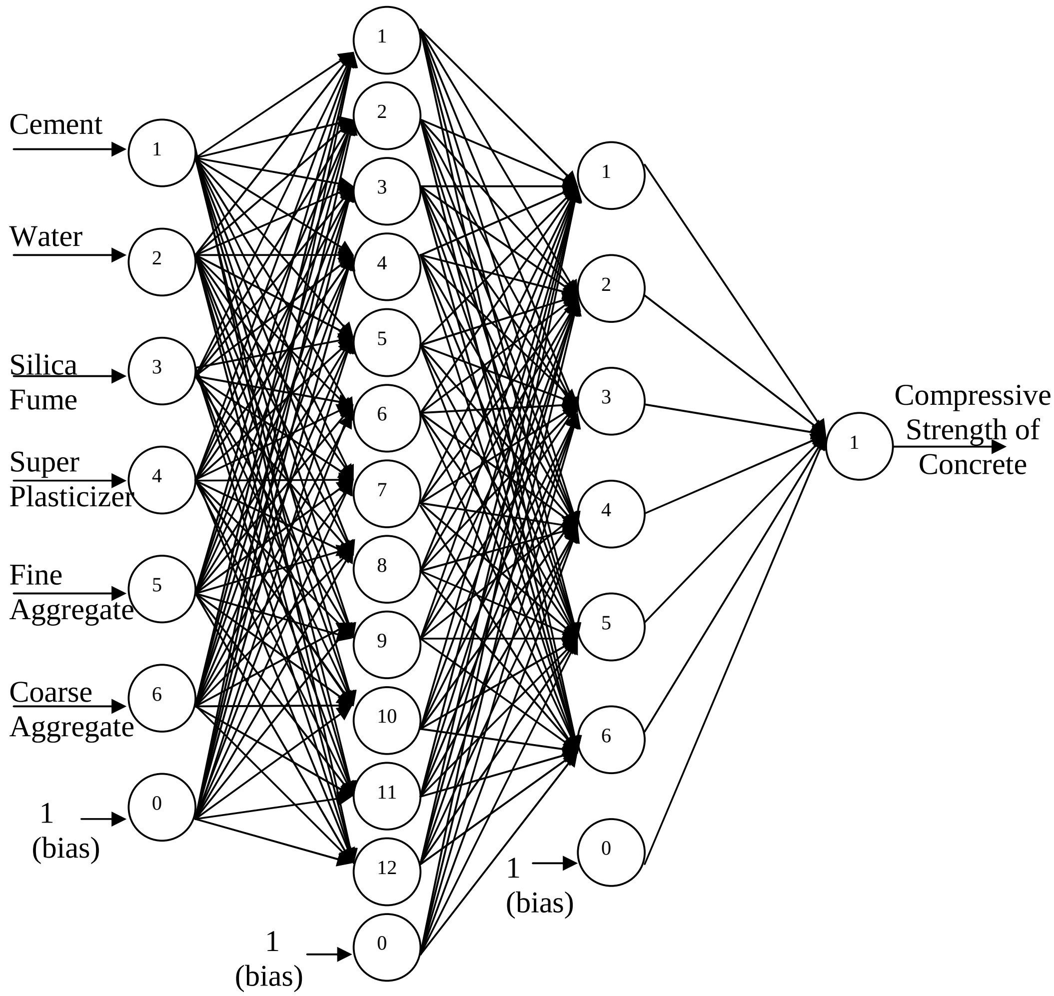

Figure 2: Final Architecture of the Developed Artificial Neural Network

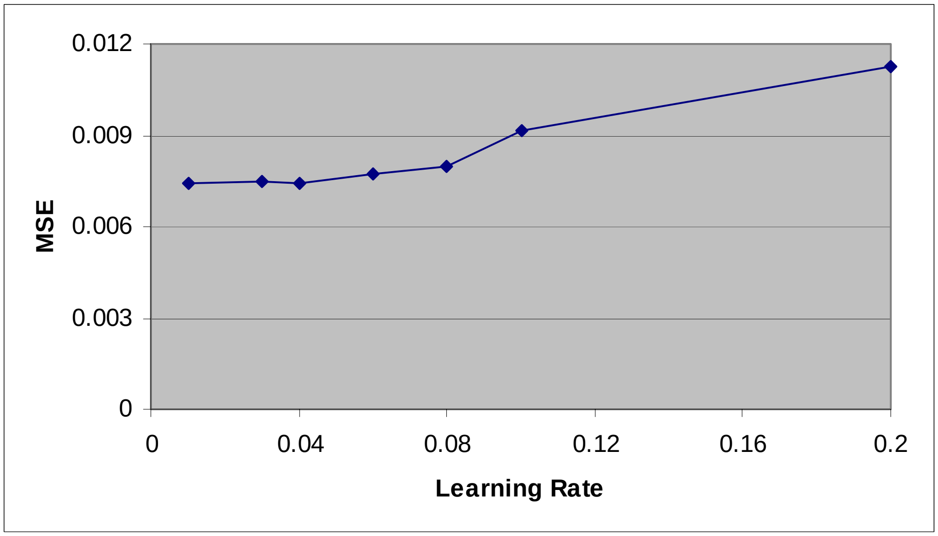

Figure3: Variation of MSE versus Rates of Learning

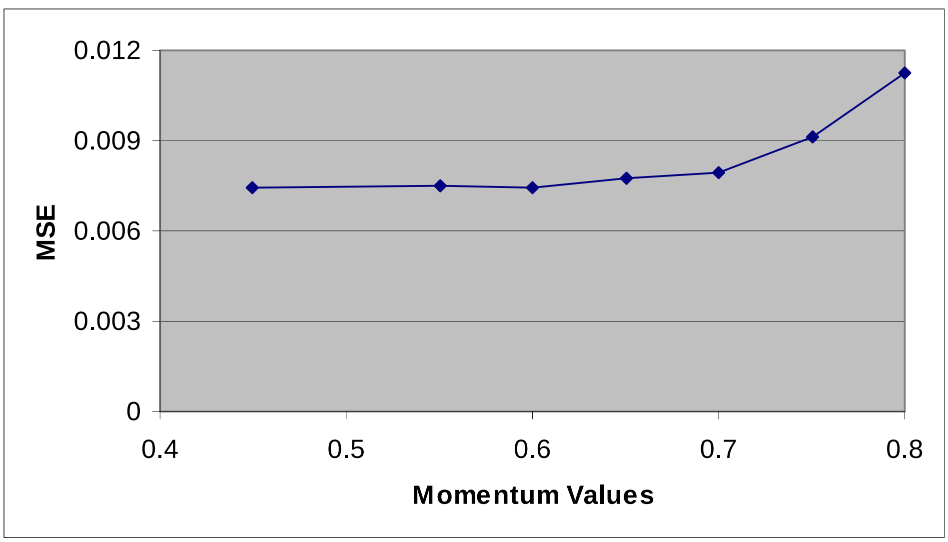

Figure 4: Variation of MSE versus Rates of Momentum

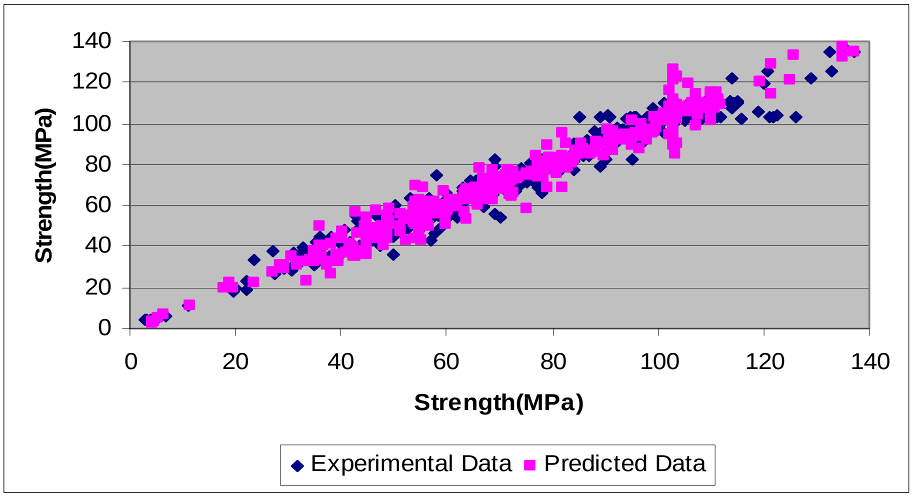

Figure5: Comparison of Experimental and Predicted data in Training Process

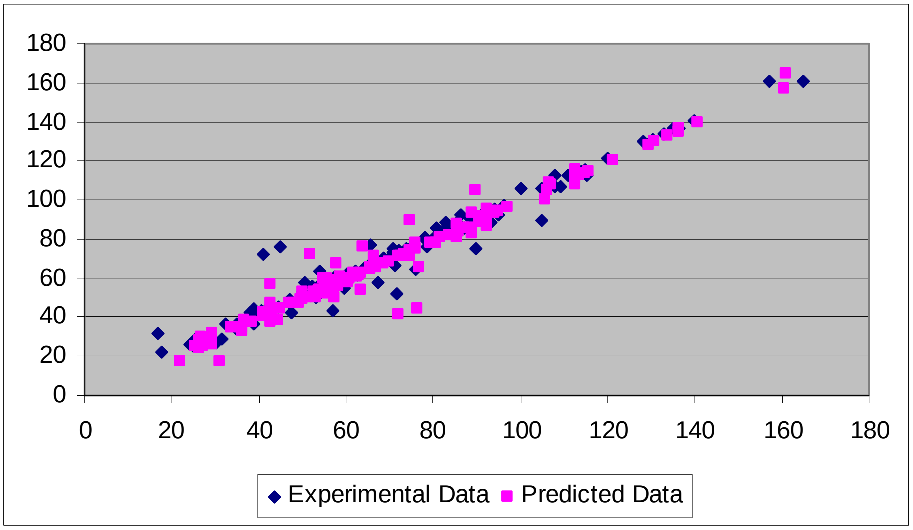

Figure 6: Comparison of Experimental and Predicted data in Testing Process

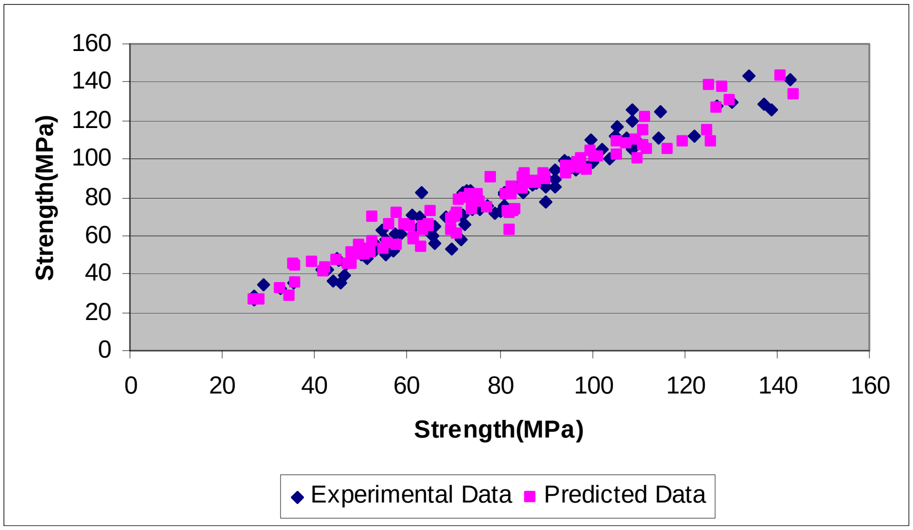

Figure 7 Comparison of Experimental and Predicted data in Validation Process

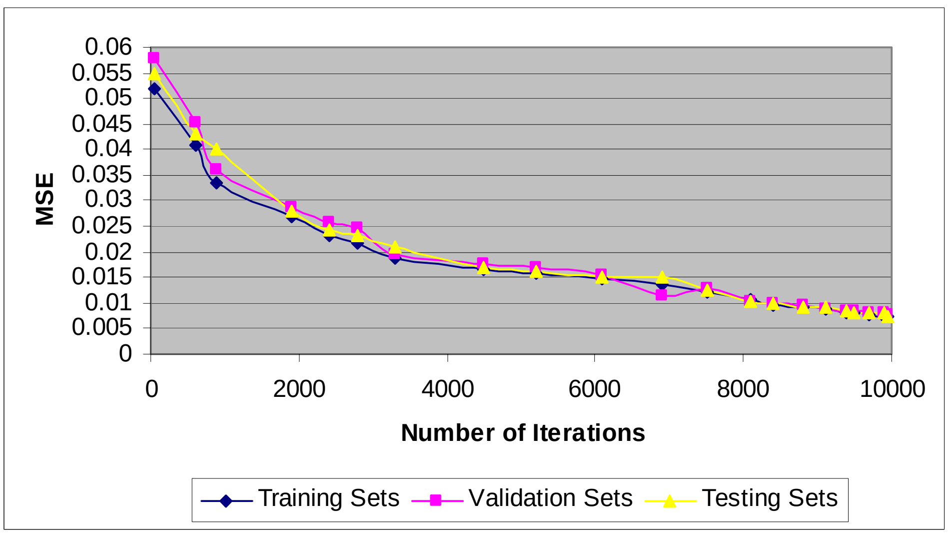

Figure 8: Comparison of MSE of the Training, Testing and Validation Sets versus the Number of Iterations