Prediction Of Concrete Strength Using Artificial Neural Network

Ogbodo Munachiso C, Dumde Dinebari K

Department of Civil University of Port Harcourt
Rivers State, Nigeria, PH-(+234) 803 719 3188
munachiso.ogbodo@uniport.edu.ng

Department of Civil University of Port Harcourt
Rivers State, Nigeria, PH-(+234) 806 811 5323
enterdumde@gmail.com

Abstract : This report presents the prediction of concrete mix ratio using Artificial Neural Network mode (ANN)l. An artificial neural network model was developed, trained and tested with 259 concrete mix data sets. These data sets were gotten from concrete companies, sorted and used, for which 70%, 15% and 15% were used for training, validation and testing phases respectively. A 3-layered feed-forward neural network model with a back-propagation algorithm was adopted. Input layer comprises of 4 nodes representing the Fineness Modulus, Coarse Aggregate ratio, Water cement ratio, and Maximum aggregate size and five output parameters which are compressive strength, water content, fine aggregate content, coarse aggregate content and cement contents all in (grams) which are the expected output. The ANN model result was compared with other approach of concrete mix design and was considered adequate. The absolute error between the output from conversional mix design and the Artificial Neural Network predicted data was 0.00083. The results indicate the utility, reliability and usefulness of the artificial neural network for accurately predicting concrete mix ratio.

Keywords: Artificial Neural Network, Concrete, Mix proportion.

1. Introduction

Concrete is a composite material that consists essentially of a binding medium within which are embedded particles or fragments of aggregate. In hydraulic cement concrete, the binder is formed from a mixture of hydraulic cement and water [1]. Concrete is the most widely used construction material because of its flowability in most complicated form. i.e. its ability to take any shape while wet, and its strength development characteristics when it hardens. Each concrete constituent influences the characteristics of the concrete and must be controlled as to composition and quantity if the end product is to be within acceptable limits of uniformity, workability, and strength. Proportioning of a concrete mix comprises of determining the relative quantities of materials to be used in production of concrete for a given purpose. The process of selecting proportions of these suitable ingredients of concrete and determining their most optimum proportions which would produce, as economically as possible, concrete that satisfies the job requirements, that is, the concrete having a certain minimum compressive strength, the desired workability and durability is called "Concrete Mix Design". Different researches have been done on the prediction of concrete mix design. Some of which include; D.O. Onwuka et. al. presented two distinct computer programs in VISUAL BASIC Language to predict concrete mix. The first model program was based on simplex function while the second model program was based on modified regression function. For both programs, the concrete compressive strength can be obtained by imputing into the computer the mix proportions of the concrete components. On the other hand, the input of mix proportions of the constituent concrete materials into the computer also gives the compressive strength as output [2]. Ken W. Day presented a system that enables the designer to input any desired set of figures constituting an ideal grading. Two further numbers are to be input to constitute inner and outer limits on the entered ideal. The software was to produce a more economical and consistent concrete. This software system can enable concrete practitioners with limited access to computers to make full use of permission to vary mix proportions [3].

2. Artificial Neural Network

Artificial Neural Network (ANN) is a uniform processing model that is inspired by the way biological nervous systems such as the brain, process information [4]. Artificial Neural Network models have the ability to learn and generalize the problems even when input data contain error or incomplete. ANN like people learns by example. The human brain can be described as a biological neural network; an interconnected web of neurons transmitting elaborate patterns of electrical signals. Dendrites receive input signals and, based on those inputs, fire an output signal via an axon [5]. In its most general form, a neural network is a machine that is designed to model the way in which the brain performs a particular task or function of interest; the network is usually implemented using electronic components or simulated in software on a digital computer [6]. Our interest will be confined largely to neural networks that perform useful computations through a process of learning. To achieve good performance, neural networks employ a massive interconnection of simple computing cells referred to as neurons or processing units. We may thus offer the following definition of a neural network viewed as an adaptive machine. An artificial neuron is composed of five main parts: inputs, weights, sum function, activation function and outputs. Inputs are information that enters the neuron from other neurons from external world. Weights are values that express the effect of an input set or another process element in the previous layer on this process element. Sum function is a function that calculates the effect of inputs and weights absolutely on this process element. This function computes the net input that comes to a neuron. The weighted sums of the input components (net)j are calculated using Eq. (1) as follows:

$$(\mathrm{net})_j = \sum_{i=1}^{n} w_{ij}\, x_i + b \qquad [1]$$

Where $(net)_j$ is the weighted sum of the j[th] neuron for the input received from the previous layer with n neurons, w_{ij} is the weight between the j[th] neuron in the previous layer, x_i is the output of the i[th] neuron in the previous layer. Activation function is a function that processes the net input obtained from sum function and determines the neuron output. In general, for multilayer feed forward models as the activation function sigmoid activation function is used. The output of the j[th] neuron $(out)_j$ is computed using Eq. (2) with a sigmoid activation function as follows;

$$(out)_j = f(net)_j = \frac{1}{1+e^{-\propto(net)}} \qquad [2]$$

Where α is a constant used to control the slope of the semi-linear region. The sigmoid nonlinearity activates in every layer except in the input layer. The sigmoid activation function represented by Eq. (2) gives outputs in (0, 1). Because its derivatives can be determined easily with regard to the parameters within $(net)_j$ variable. It is apparent that a neural network derives its computing power through, first, its massively parallel distributed structure and, secondly, its ability to learn. Generalization refers to the neural network producing reasonable outputs from inputs not encountered during training (learning). These two information processing capabilities make it possible for neural networks to solve complex (large-scale) problems that are currently intractable. In practice, however, neural networks cannot provide the solution working by themselves alone. Rather, they need to be integrated into a consistent system engineering approach. Specifically, a complex problem of interest is decomposed into a number of relatively simple tasks, and neural networks are assigned a subset of the tasks (e.g. pattern recognition, associative memory, control) that match their inherent capabilities [7]. Neural networks and conventional algorithmic computers are not in competition but complement each other. There are tasks that are most suited to an algorithmic approach like arithmetic operations and some tasks are more suited for neural networks. Even more, a large number of tasks require systems that use a combination of the two approaches (normally a conventional computer is used to supervise the neural network) in order to perform at maximum efficiency [8].

3. Neural Network Design and Training

In this study, industry based concrete mix designs were obtained from concrete companies. 259 sets of data were assessed and confirmed to be well controlled. Table 1 shows a sample of the sourced data set showing; the slump (s), fineness modulus (fm), Max. aggregate size (A), water-cement ratio (wc), comprehensive strength (f_{cu}), water (w), cement (c), fine aggregate (fa), and coarse aggregate (ca) contents. The data set were sub-divided into input and output data. The input parameters are the design stipulations and material specifications. These subsets of data are the initial parameters required for concrete mix design. The output parameters are proportions of concrete ingredients like cement content, fine aggregate content, coarse aggregate content, and water content and compressive strength for this particular work.

Table 1: *Sample of sourced data*

S	fm	A	wc	f_{cu}	w	c	fa	ca
90	2.64	20	0.87	18.2	200	230	973	969
95	2.64	20	0.7	15.5	195	275	948	977
100	2.64	20	0.7	15.4	195	275	948	977
100	2.64	20	0.7	15.6	195	275	948	977
100	2.64	20	0.7	15.8	195	275	948	977
100	2.64	20	0.7	15.9	195	275	948	977
100	2.64	20	0.7	18.4	195	275	948	977
100	2.64	20	0.6	20.9	190	315	948	955
100	2.64	20	0.6	22.6	190	315	948	955

A 3-layered feed-forward neural network model with a back-propagation algorithm was adopted. The ANN was developed using the popular MATLAB software package, (MATLAB R2012a). Where the sigmoid transform functions were adopted as the activation function to process the net input obtained and determines the neuron output. The sorted data gotten from mix design tests was randomly divided into three sets; 70% (181 samples) for model training, 15% (39 samples) for model validation and the remaining 15% (39 samples) was used to test the correctness of the developed model. The model has four (4) neurons representing input parameters which are Fineness Modulus of fine aggregate, slump, Water cement ratio and Maximum size of coarse Aggregate and five output parameters which are comprehensive strength, weights in (kg/m of mix components (water, fine aggregate, coarse aggregate and cement). The training function adopted in the construction of the ANN model was Levenberg-Marquardt (LM). This training function is one of the conjugate gradient algorithms that start training by searching in the steepest descent direction (negative of the gradient) on the first iteration. The learning and weight and bias adjustments were done automatically. In general, the net input to each node is calculated as:

$$N_j^l = \sum_{i=1}^{n} W_{ji}^l . X_i^{l-1} + \beta_j^l \qquad (3)$$

Where W_{ji}^l is the weight that connects node j in layer l to node i in layer $l-1$; n is the number of nodes in layer $l-1$; $\beta_j^l$is a threshold value assigned to node j in layer l; and $X_i^{l-1}$is the input coming from node i in layer $l-1$ to node j in layer l. The net input, N_j^l is then modified by an activation function, f, to generate an output value, Y_j^l given by:

$$Y_j^l = f(N_j^l) \qquad (4)$$

Where f is a nonlinear activation function (sigmoid activation function) assigned to each node in the network. The learning mechanism of this back-propagation network is a generalized delta rule that performs a gradient-descent on the error space, in order to minimize the total error between the calculated (target) and desired values at the output layer during modification of the connection weights. The implementation of this algorithm updates the network weights and biases in the direction in which the error decreases most rapidly. Training was accomplished in an iterative manner. Each iteration cycle involves the feed-forward computation followed by an error-backward propagation to modify the connection weights. Convergence depends on the number of hidden layer nodes, learning rate parameters and the size of the data set required to create the proper results. The training procedure was carried out by presenting the network with 70% of the data set as earlier mentioned in a patterned format. The network is presented with the variables in the input vector of the first training pattern, followed by an appropriate computation through the nodes in the hidden layers and prediction of the appropriate outputs. The error between the predicted output and target value is calculated and stored. The network is then presented with the second training pattern and so on until the network has gone through all the data available for training the network. The Root-Mean-Square (RMS) of the error was calculated and back propagated to the network. The connection strength between nodes (Biases and weights) are modified during the back-propagation phase such that the (RMS) errors are reduced. The iterations were done continuously until convergence was achieved at 147th iteration. The training process and the associated ANNs analyses were carried out with an optimal value of learning rate of 0.00064955 at 141 iterations with an error goal of 0.0000. Mean squared error (MSE) during the training cycles was also monitored by calculating the error against each parameter. The overall performance of the trained networks was obtained by comparing the outputs. 78 sets of data were used to evaluate the confidence (test) and validate the performance of the trained network.

4. Results and Discussions

Figures 1 illustrates the distribution of the network outputs versus the target values for the training, validation, testing and for all the data sets. All output data points are distributed along the optimal agreement line, with the training, validation, testing for the best Root-Mean-Square (RMS) errors of -0.000087, 0.003, 0.0045, and 0.00027 respectively. The correlation coefficient between the predicted and target components was found to be 1 for all the phases which is satisfactory.

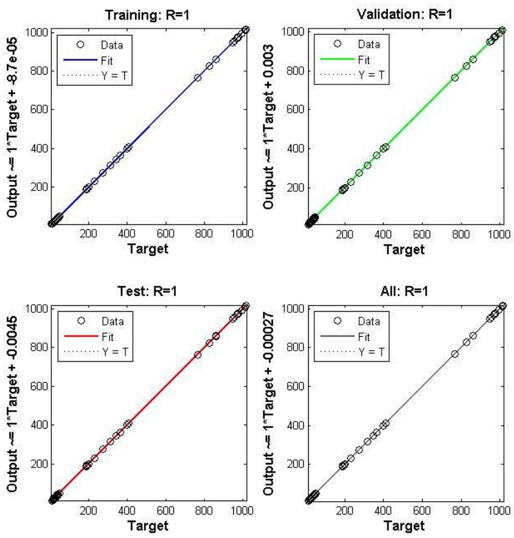

Figure 1: *Regression Analysis*

The relatively larger prediction error and less correlation parameters may, therefore, be associated to high variability in the mixture development rather than the prediction method and may be related to the different types of such materials used in the training set targets. Figure 2 show the convergence characteristics of the ANN model during the training, validation and testing phases, respectively. Also, it shows the number of epochs (147 iterations) and the best validation performance which is 0.00064955 at 141 iterations. The plot (mean squared error against epochs) also indicates the rate of training. After the best validation point (at epoch 141), the 3 plot phases maintain a constant mean squared error rate, showing that more training on the model has no effect.

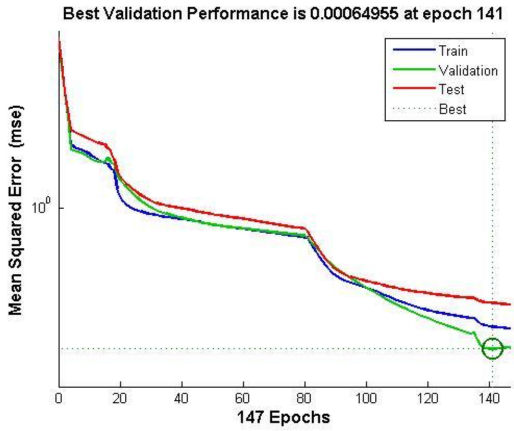

Figure 2: *Performance Plot*

To test the accuracy of the ANN model, concrete mixes, unfamiliar to the network in the range of training data sets, were presented to the ANN model and the network was required to predict the output associated with each mix design. The actual and predicted values are listed in table 2. From the error margin, the mean predicted error was 0.0009, which shows the accuracy of the model.

Table 2: *Actual and ANN values*

f_{cu}		w		c		fa		ca	
Actual	ANN	Actual	ANN	Actual	ANN	Actual	ANN	Actual	ANN
17.9	17.89456	200	200.0193	230	230.0221	973	973.0181	969	969.0062
18.2	18.17144	200	200.0126	230	230.0241	973	973.0143	969	968.9954
17.5	17.49135	200	200.0171	230	230.0206	973	973.0167	969	969.0045
10.4	10.40939	200	199.9912	230	229.9896	973	972.9937	969	968.9767
10.6	10.60421	200	200.0009	230	229.9907	973	972.9996	969	968.9857
12.9	12.88239	200	200.0091	230	230.0022	973	973.0088	969	968.9987
10.5	10.50596	200	200.0006	230	229.9902	973	972.9992	969	968.985
10.5	10.50226	200	200.0104	230	229.9905	973	973.0047	969	968.9932
10.5	10.50743	200	199.9915	230	229.9901	973	972.9942	969	968.9773
10.1	10.11538	200	199.9818	230	229.9882	973	972.988	969	968.9677
10.3	10.31084	200	199.9952	230	229.9891	973	972.9958	969	968.9799
18.1	18.05665	195	194.9685	275	274.9844	948	947.9742	977	977.0053

The errors for the test results falls close to the zero error, as showed on figure 4. The mean error for the compressive strength, water, cement, fine aggregate and coarse aggregate are 0.002781, -0.00289, -0.00115, -0.00155, and -0.00229 respectively. The mean error from different parameters indicates that learning rate has a minimum error. The compressive strength has a positive and least error different from the other output parameters because it was part of the input parameter. While the mean error for the entire output parameters is -0.00083, indicating a minimum error rate showing that the model has undergone learning process effectively.

5. Conclusion

Based on the findings of this investigation, the following conclusions can be drawn:

- The model performed quite well in predicting, not only the output parameters used in the training process, but also those of test mixtures that were unfamiliar to the neural network.
- The testing of the model by un-used data within the range of input parameters shows that the mean error for the compressive strength, water content, cement content, fine aggregate content and coarse aggregate content are 0.002781, -0.00289, -0.00115, -0.00155, and -0.00229 respectively, which indicates that learning rate has a minimum error. The compressive strength has a positive and least error different from the other output parameters because it was part of the input parameter. Also, the mean absolute maximum error for the model as about 27%. While the mean error for the entire output parameters is -0.00083, indicating a minimum error rate showing that the model has undergone learning process effectively.
- The results obtained from the developed Artificial Neural Network model were compared with results from experimental studies. From the results obtained, the mean predicted error was 0.0009 which indicate good agreements between both data.
- The average prediction to the experimental data were closed when compared, indicating the ability of the model to predict concrete mix ratio accuracy and effectively.

6. Recommendation

Artificial Neural Network can be used in the prediction of concrete mix design because of its accuracy, speed and effectively in concrete mix ratio prediction. Although the prediction capability of any ANN model is limited to data located within the boundaries of the training range, the proposed model can be retrained to include a wider range of input variables by providing additional training sets covering the new range.

References

[1]. P. Kumar and J.M. Paulo, Concrete: microstructures, properties and materials, McGraw-Hill Education, 2006.

[2]. D.O. Onwuka, C.E. Okere, O.M. Ibearugbulem, S.U. Onwuka "Computer-Aided Design of Concrete Mixes" International Journal of Computational Engineering Research Vol. 3(2), pp. 67-81, 2013.

[3]. Ken W. Day, Tailor Made Concrete Structures, CRC Press, 2008.

[4]. W.Mc Cullock and W. Pitts "A logical calculus of the ideas immanent in neural network", Bull Math Biophys,Vol 5,pp 115-133, 1943.

[5]. David Shittman, The Nature of Code, D. Shiffman, 2012.

[6]. David Kriesel, "A Brief Introduction to Neural Networks", http://www.dkriesel.com. 2007.

[7]. M. Hajek, Neural Network, University of KwaZulu-Natal, 2005.