World Academy of Science, Engineering and Technology
International Journal of Civil and Environmental Engineering
Vol:8, No:9, 2014

Modeling and Analysis of Concrete Slump Using Hybrid Artificial Neural Networks

Vinay Chandwani, Vinay Agrawal, Ravindra Nagar

***Abstract*—**Artificial Neural Networks (ANN) trained using back-propagation (BP) algorithm are commonly used for modeling material behavior associated with non-linear, complex or unknown interactions among the material constituents. Despite multi-disciplinary applications of back-propagation neural networks (BPNN), the BP algorithm possesses the inherent drawback of getting trapped in local minima and slowly converging to a global optimum. The paper present a hybrid artificial neural networks and genetic algorithm approach for modeling slump of ready mix concrete based on its design mix constituents. Genetic algorithms (GA) global search is employed for evolving the initial weights and biases for training of neural networks, which are further fine tuned using the BP algorithm. The study showed that, hybrid ANN-GA model provided consistent predictions in comparison to commonly used BPNN model. In comparison to BPNN model, the hybrid ANN-GA model was able to reach the desired performance goal quickly. Apart from the modeling slump of ready mix concrete, the synaptic weights of neural networks were harnessed for analyzing the relative importance of concrete design mix constituents on the slump value. The sand and water constituents of the concrete design mix were found to exhibit maximum importance on the concrete slump value.

***Keywords*—**Artificial neural networks, Genetic algorithms, Back-propagation algorithm, Ready Mix Concrete, Slump value.

I. Introduction

READY Mix Concrete (RMC) has emerged as the preferred choice among contractors and builders for reinforced concrete construction, primarily due to its customized combination of constituents resulting in an engineered premium quality concrete mix. With the adaptability to be transported to congested sites and better conditions of quality control, RMC has given an impetus to the infrastructure growth providing both reliability and durability of construction. Regardless of the sophistication of the mix design procedure used and other considerations, such as cost, a concrete mixture that cannot be placed easily or compacted fully is not likely to yield the expected strength and durability characteristics [1]. The ease, with which concrete can be placed, compacted and finished at site with sufficient resistance to segregation, is defined as the workability of concrete.

Vinay Chandwani is a Ph.D Research Scholar in the Department of Civil Engineering, Malaviya National Institute of Technology Jaipur, JLN Marg, Jaipur, Rajasthan, India (mobile: (+91) 9828022916, e-mail: chandwani2@yahoo.com).

Vinay Agrawal is working as Assistant Professor in the Department of Civil Engineering, Malaviya National Institute of Technology Jaipur, JLN Marg, Jaipur, Rajasthan, India (e-mail: agrawal_vinay_2000@yahoo.com).

Ravindra Nagar is working as Professor in the Department of Civil Engineering, Malaviya National Institute of Technology Jaipur, JLN Marg, Jaipur, Rajasthan (e-mail: ravindranagar@hotmail.com).

The difficulty in measuring the mechanical work defined in terms of workability, the composite nature of the fresh concrete, and the dependence of the workability on the type and method of construction makes it impossible to develop a well-accepted test method to measure workability [2]. However, consistency of concrete is measured in quantitative terms using a widely used test called the Slump Test. Determination of concrete slump, is an important quality assurance parameter in RMC industry. It not only helps in assessing the shelf life of the RMC and the maximum transit time that a RMC can safely undertake without loss of flowability and pumpability, but also helps in customizing concrete to the type and need of construction activity and maintaining uniformity of concrete from batch to batch.

The empirical formula in the form of regression equations based on experimental results, are commonly in use to judge the property of concrete based on its design mix constituents. However, these empirical relationships do not provide the desired prediction accuracy when there are number of variables influencing the concrete property whose interactions are non-linear, complex or unknown in nature. Artificial Neural Networks (ANN) inspired by the learning mechanism of the human brain, present a simplified approach for modeling unstructured material behavior problems. Modeling slump of concrete based on its design mix proportion is one such unstructured problem. In past decade ANN has been harnessed to predict the slump and strength of ready mix concrete containing retarders and high strength concrete containing silica fume and plasticizers [3], slump of Fly ash and slag concrete (FSC) [4], for modeling compressive strength and slump of high strength concrete (HSC) [5], to model the slump of high performance concrete and comparison of ANN with second order regression models [6] and for modeling slump of concrete based on its mix constituents using laboratory test results [7].

Apart from the studies mentioned above, recent studies conducted for modeling the material behavior of concrete [8]-[13], have harnessed the back-propagation (BP) algorithm trained artificial neural network. Despite back-propagation neural network's (BPNN) popularity and wide range of applications, it is always faced with an inherent drawback of getting trapped at local minima even though there is much deeper minimum nearby and slow convergence rate. This drawback is attributed to the random draw of initial neural network weights and biases, which compels ANN to behave differently during each re-run of the network training incapable of finding satisfactory solutions. The stochastic search ability of genetic algorithms (GA) inspired by the evolutionary processes, namely, natural selection and genetic variation, allows simultaneous search for optimal solutions in different directions, minimizing the chance of getting trapped in a local minimum. Recent studies harnessing genetic algorithms (GA) for evolving the initial weights and biases for neural networks [14]-[18], have shown that hybrid ANN-GA approach outperforms the prediction performance of the conventional BP training of neural networks.

Despite a wide range of applications of hybridizing GA with ANN for evolving optimal initial weights and biases, GA has not been amalgamated with ANN for modeling the slump of concrete. The hybrid methodology presented in the paper harnesses the genetic algorithms (GA) stochastic global search for evolving the initial weights and biases and the local search BP algorithm for fine tuning of GA evolved weights and biases for developing a robust ANN model. The hybrid ANN-GA methodology has been used for modeling the slump of ready mix concrete based on its design mix constituents viz., cement, pulverized fly ash, sand, coarse aggregate (20 mm), coarse aggregate (10 mm), admixture and water. The synaptic weights of the trained ANN-GA model have been used for assessing the relative importance of each constituent of the RMC on the slump value.

The research paper has been divided into sections. Data collection is dealt in Section II. Section III describes the methodology of determining ANN architecture, its subsequent training using GA evolved weights and biases and statistical performance metrics. Section IV and V present the results and their discussions. Section VI deals with evaluating relative importance of concrete's design mix proportion on its slump value and Section VII elaborately deals with the conclusions of the study.

II. Material

A. Exemplar Data for Neural Network Training, Validation and Testing

ANN is a learning paradigm inspired by the approach in which information is processed by the human brain. The neural networks imbibe the subtle relationships between and input and output data pairs and are able to automatically construct a relationship based on the flow of information through the network. The neural networks are therefore data intensive and rely heavily on the quality and quantity of data/information. The exemplar patterns for neural network modeling of concrete slump were collected from a local RMC plant. The data comprised of concrete design mix proportions and their corresponding slump test values. The data consisted of 565 concrete design mixes having different proportions of cement, pulverized fly ash (PFA), sand, coarse aggregate (20 mm), coarse aggregate (10 mm), admixture and water content in kg/m^3. The corresponding slump values were measured and reported in mm. The data were randomized and separated into three disjoint datasets viz., training, validation and test data-sets. The training dataset comprised of 70% data. The remaining 30% data were divided equally to form the validation and test datasets.

III. Methods

In conducting the study, the Neural Network Toolbox and Global Optimization Toolbox included in the commercially available software MATLAB R2011b (Version 7.13.0.564) was used to implement the BPNN and GA respectively.

A. Pre-processing of Data

As discussed in the previous section that, exemplar data for modeling slump of concrete comprised of cement, PFA, sand, coarse aggregate 20 mm and 10 mm, admixture, water and slump value. Since the data consists of different constituents having different material properties and maximum-minimum range, it requires standardization of data through normalization in the range -1 to +1 or 0 to +1. This pre-processing of data is preferably done to remove any inherent bias towards any variable, thereby ensuring the equal attention of the network toward all variables. Moreover, this facilitates the learning speed, as these values fall in the region of sigmoid transfer function where the output is most sensitive to the variations of the input values [19]. Linear scaling in the range -1 to +1 has been used in the present study having function

$$x_{norm} = \frac{2*(x - x_{\min})}{(x_{\max} - x_{\min})} - 1 \quad (1)$$

where x_{norm} is the normalized value of the variable x, $x_{\max}$ and $x_{\min}$ are the minimum and maximum values of variable x respectively.

B. Neural Network Architecture and Training Parameters

ANN presents a computational counterpart to the human brain, with nodes representing the neurons and weighted connections between the nodes synonymous to the synapses between the biological neurons. Nodes or artificial neurons are the simple processing elements and are arranged in layers. A neural network can consist of several layers. Multi layer Feed-forward neural networks (MFNN) that are commonly used for tasks associated with prediction and forecasting, consist of an "input layer", "output layer" and a number of intermediate "hidden layer/s". The feed-forward neural network is a fully connected network, allowing only connections in forward direction between the inter-layer neurons.

The "input layer" consists of neurons which receive input information. In the present study, the "input layer" consisted of seven neurons viz., cement, PFA, sand, coarse aggregate (20 mm), coarse aggregate (10 mm), admixture and water content in kg/m^3. The "output layer" consisted of only one neuron i.e. concrete slump value in mm. The number of hidden layers and hidden layer neurons depend of the complexity of the function to be approximated. Reference [20] claimed that a single hidden layer neural network with sufficient number of hidden layer neurons can approximate any functional relationship. Certain studies in the past have suggested the "thumb rules" for deciding the number of hidden layer neurons [21]-[24]. Nevertheless, the number of hidden layers and hidden layer neurons is decided by a trial and error process. In the present study, seven single hidden layer neural network architectures of different complexities with hidden neurons varying from five to eleven have been trained and validated to select the optimal architecture. The neural network architecture with five hidden layer neurons for modeling slump of concrete is shown in Fig. 1.

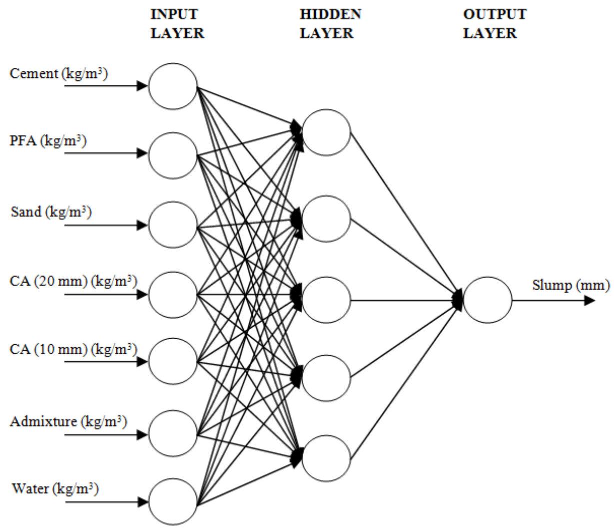

Fig. 1 Neural Network Architecture with five hidden layer neurons

The input neurons receive the signal and passes on the weighted sum of all signals arriving at a single neuron through a transfer function. Introduction of transfer function adds non-linearity into the network and helps it to learn the complex or nonlinear functional relationships among the input-output data pairs. In this study tangent hyperbolic transfer function is used to transfer the output at input neurons to the hidden layer neurons. For comparison of actual values with ANN predicted output, a linear transfer function has been adopted to transfer the information from hidden layer to the output layer.

The training of MFNN is undertaken by back-propagation (BP) algorithm. This type of neural network is commonly known as back-propagation neural network (BPNN). The BP algorithm is a gradient descent algorithm which attempts to minimize the error between the actual values and predicted outputs by sequential updating of neural network weights and biases. A suitable learning rate and momentum coefficient for BP algorithm helps in the convergence of the algorithm by iterative updating of weights and biases during each learning cycle. A large value of the learning rate speeds up the convergence but may lead the network to overshoot the global minima. To allow larger learning rate to speed up convergence without producing weight oscillations, momentum coefficient is incorporated in updating of synaptic weights. The momentum term effectively filters out the high frequency variations of the error surface in the weight space, since it adds the effect of the past weight changes on the current direction of movement in the weight space [25]. A suitable combination of learning rate and momentum coefficient helps in faster convergence of the BPNN. In the present study Lavenberg-Marquardt back-propagation algorithm has been used along with learning rate 0.45 and momentum coefficient 0.85. Lavenberg-Marquardt back-propagation training algorithm is the fastest converging algorithm preferred for supervised learning. It can be regarded as a blend of steepest descent and Gauss–Newton method, combining the speed of Newton algorithm with the stability of the steepest descent method [26]. The algorithm has a dual way of approaching the solution to a function, behaving as steepest descent when the solution is far away from the local minimum and Gauss–Newton when the solution is near to the local minimum.

C. Training and Determination of ANN Using BP Algorithm

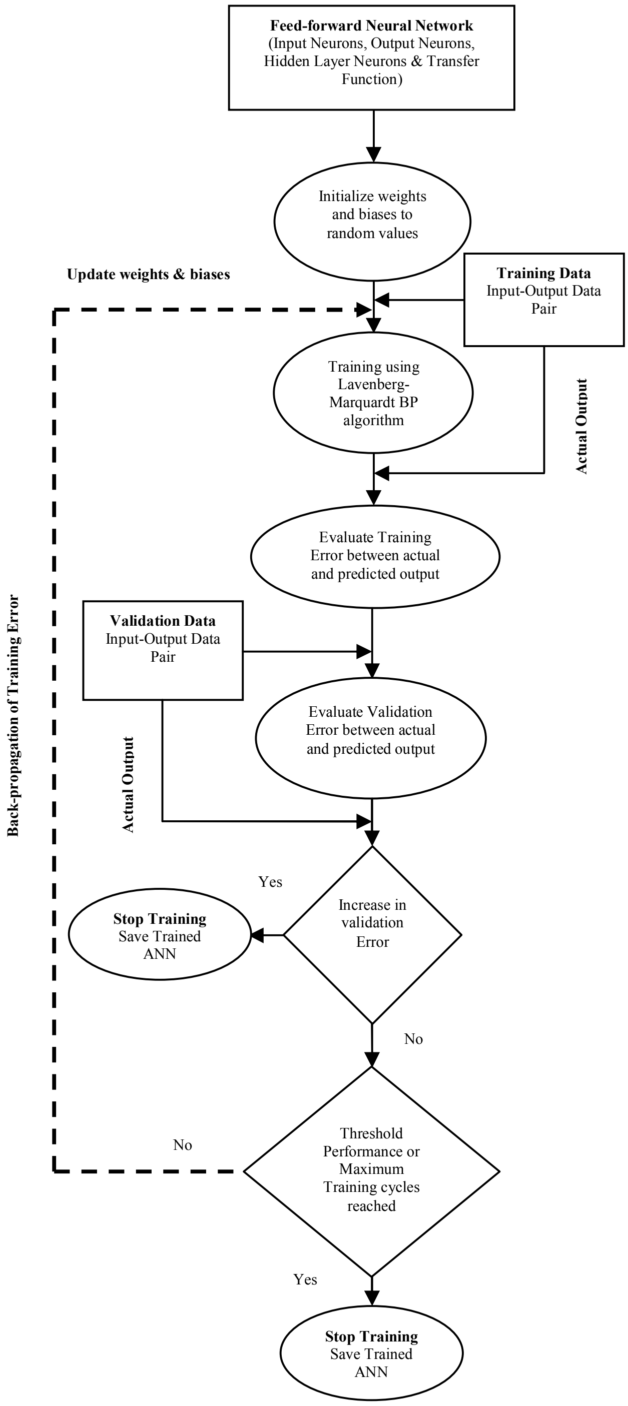

Fig. 2 Neural Network training and validation using BP algorithm

The multiple neural network architectures created by varying the number of hidden layer neurons and its training parameters discussed in the previous section, were trained using Lavenberg-Marquardt BP training algorithm. Prior to training, the initial weights and biases were randomized and initialized in the range -0.5 to +0.5. The training dataset comprising of input-output data pairs were presented to the neural network and ANN predicted outputs were compared with actual outputs for computing the training error. In order to avoid over-fitting of the training data-set, early stopping technique was employed. Apart from monitoring the training error at each training cycle, validation error is also evaluated by presenting the validation data-set to the trained neural network. The training of neural network is terminated when the validation error tends to increase, although training error may continue to decrease. The ANN with least validation error is selected as the neural network model for modeling slump of concrete. Flow-chart of neural network training and validation using BP algorithm is shown in Fig. 2. In the present study the neural network architecture (7-9-1) with nine hidden layer neurons yielded the least validation error and is therefore chosen as the neural network model for modeling slump of RMC.

D. Evolving Initial Neural Network Weights and Biases Using GA and Subsequent Training Using BP Algorithm

Genetic algorithms (GA) are population based stochastic search and optimization algorithms inspired by Darwin's "Survival of the Fittest" heuristic. Based on the evolutionary ideas of natural selection and genetics, they present a perfect blend of exploration and exploitation of the search space, to direct the search to the regions where there is maximum probability of finding a better solution. Compared with other traditional searching or optimization techniques such as hill-climbing methods, which depend solely on local information to decide the best direction along which the next step should move, GAs use global information, perform parallel search and do not require local gradient information, which enable it to find globally optimal or near globally optimal solutions [27]. The gradient-free and parallel nature of search employed by GA gives it an edge over BP algorithm's gradient decent technique to avoid falling into local minima and accelerated progress towards global optimum.

The process of hybridizing GA with ANN constitutes of two stages viz., evolution of neural network initial weights and biases using GA and using these optimal weights and biases for training of ANN using BP algorithm. The weights and biases of neural network are initialized as genes of the chromosomes. The ANN is constructed and fitness of each chromosome is evaluated by presenting the ANN with the training input-output data pair and evaluating the root mean square error (RMSE) between the actual and the predicted outputs. The fitness function acts as measure of distinguishing optimal solution from numerous sub-optimal solutions by evaluating the ability of the possible solutions to survive or biologically speaking it test's the reproductive efficiency of chromosomes. GA performs stochastic operations on the chromosomes through genetic operations viz., crossover and mutation and evolution operation viz., selection. By employing the selection operator, the fitter chromosomes are segregated from the chromosomes having less fitness, thereby improving the population fitness over successive generations.

Crossover is a recombination operator that helps in producing new offspring by incorporating the strengths of the parent chromosomes thereby, enriching the population with better individuals. Crossover produces clones of good chromosomes by randomly choosing a cross site along the length of chromosome and following this cross site, swapping the genes across the two parent chromosomes to produce a better off-spring. Mutation introduces genetic diversity into the current population by randomly modifying its building blocks. It allows exploration of the entire solution space and prevents the algorithm to be trapped in local minima. The crossover and mutation operator operating together, create the next generation of population. The process is repeated till maximum generations or stalling of fitness function at a particular value is achieved. In the present study an initial population size of 50 chromosomes with roulette wheel selection strategy, scattered crossover operator with probability of crossover 0.9, uniform mutation with probability of mutation 0.01 and maximum number of generations 100 has been used.

The neural network weights and biases evolved using GA are subsequently harnessed for training the ANN using BP algorithm. The ANN is initialized using GA evolved weights and biases and further fine tuned through BP algorithm training of ANN. The flow-chart of evolving weights and biases using GA and subsequent use of these weights and biases for ANN training is exhibited in Fig. 3.

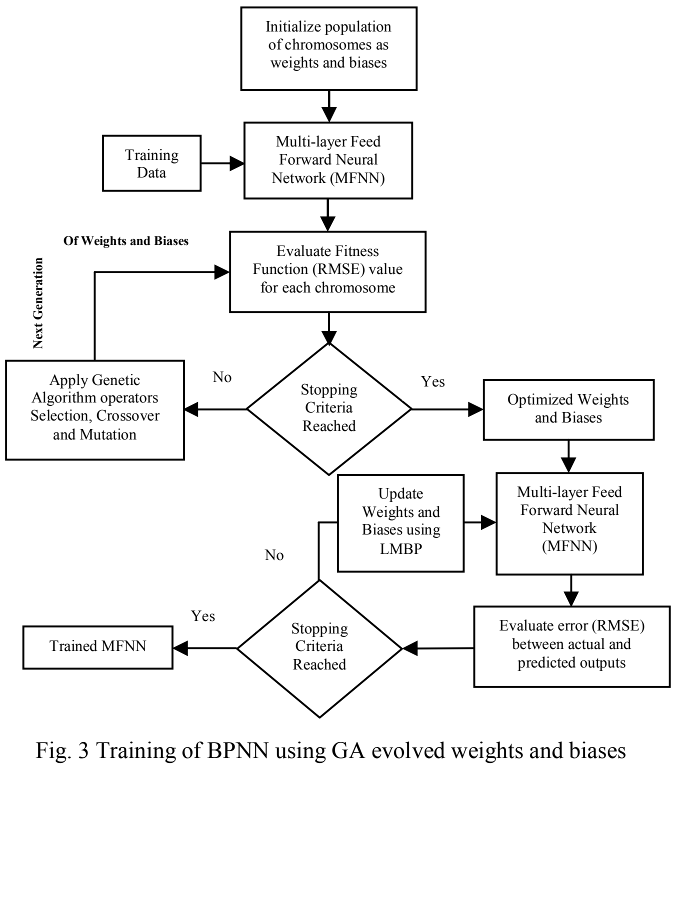

Fig. 3 Training of BPNN using GA evolved weights and biases

E. Statistical Analysis

The performance of the trained BPNN and hybrid ANN-GA models were evaluated using five different statistical parameters. The statistical performance metrics include: root mean square error (RMSE), coefficient of correlation (R), Nash-Sutcliffe efficiency (E), mean absolute percentage error (MAPE) and normalized mean bias error (NMBE) given by:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(T_i - P_i\right)^2} \tag{2}$$

$$R = \frac{\sum_{i=1}^{N}\left(\left(T_i - \overline{T}\right)\left(P_i - \overline{P}\right)\right)}{\sqrt{\sum_{i=1}^{N}\left(T_i - \overline{T}\right)^2 \sum_{i=1}^{N}\left(P_i - \overline{P}\right)^2}} \tag{3}$$

$$E = 1 - \frac{\sum_{i=1}^{N}\left(T_i - P_i\right)^2}{\sum_{i=1}^{N}\left(T_i - \overline{T}\right)^2} \tag{4}$$

$$MAPE(\%) = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|T_i - P_i\right|}{T_i} \times 100 \tag{5}$$

$$NMBE(\%) = \frac{1/N \sum_{i=1}^{N}\left(P_i - T_i\right)}{1/N \sum_{i=1}^{N} T_i} \times 100 \tag{6}$$

where T_i and P_i denote the target or observed values and ANN predicted values and $\overline{T}$ and $\overline{P}$ represent the mean observed and mean ANN predicted values, respectively. N represents the total number of data.

RMSE statistics computes the root mean square error by comparing the target or actual values with the predicted outputs. A lower RMSE indicates good prediction, but this statistic is biased towards to high error values. Coefficient of correlation (R) measures the degree of association between the two variables. The value of this statistic close to 1.0 indicates an almost perfect linear relationship between the actual and predicted values. The coefficient of efficiency (E) or Nash Sutcliffe efficiency [28] is a ratio of residual error variance to measured variance in observed data. A value close to unity indicates the accuracy of the model. MAPE statistics measure the mean of the relative absolute error divided by the observed value. A lower value of this statistic indicates better prediction accuracy. NMBE measures the ability of the model to predict a value which is situated away from the mean value. A positive NMBE indicates over-prediction and a negative NMBE indicates an under-prediction of the model [29]. A combined use of the performance metrics narrated above can provide an unbiased estimate for the prediction ability of the neural network models.

IV. Results

As discussed in previous section that, hybridization of GA with ANN consisted of two stages. In the first stage, the evolutionary heuristic GA was harnessed for determining the optimal weights and biases for training ANN using BP algorithm. The GA performed 2000 function evaluations and took 39.7179 seconds to converge to optimal weights and biases in 39 generations (Fig. 4).

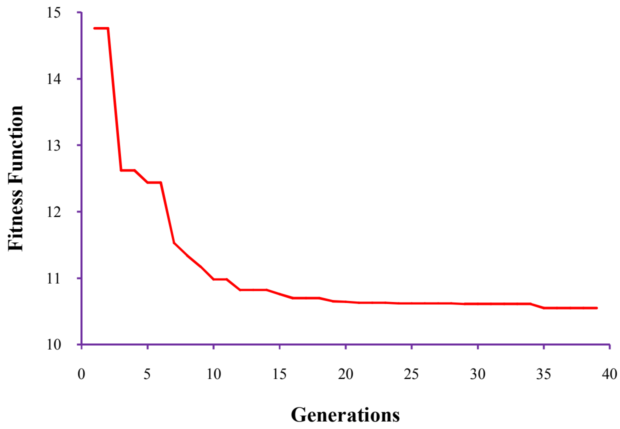

Fig. 4 Fitness function versus generations

In the second stage, the neural network architecture selected for modeling slump of concrete (7-9-1) was initialized with GA evolved weights and biases and trained using the BP algorithm. The hybrid ANN-GA was able to reach the desired performance goal 0.003 in 63 epochs taking 2.0124 seconds (Fig. 5). The same neural network architecture trained using BP algorithm initialized with a random draw of weights took 1374 epochs and 50.903 seconds to reach the desired performance goal (Fig. 6).

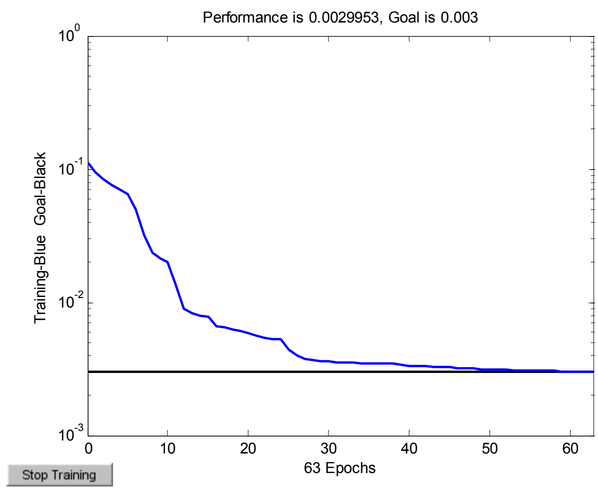

Fig. 5 Training of hybrid ANN-GA model

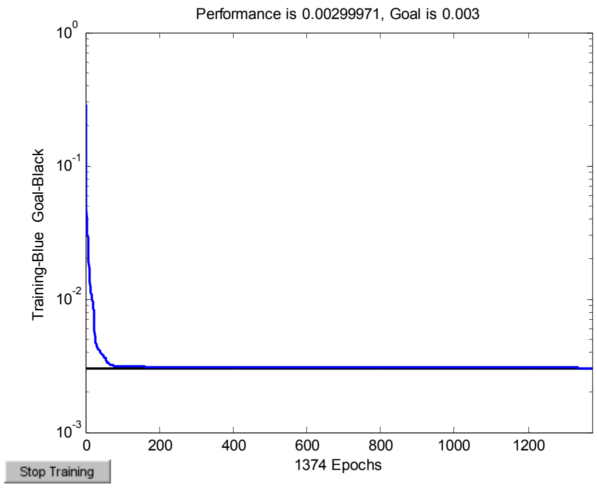

Fig. 6 Training of BPNN model

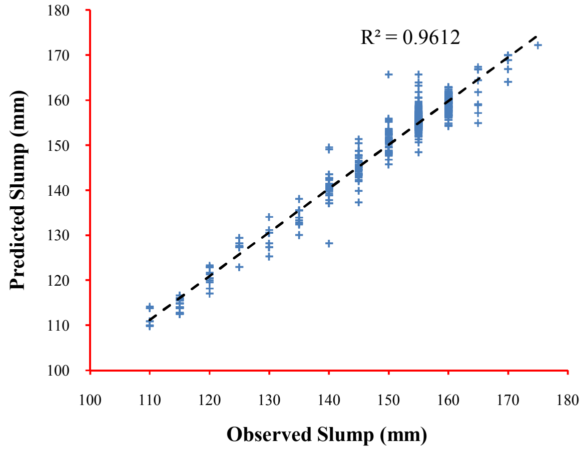

Fig. 8 Regression plot for ANN-GA model

TABLE I
STATISTICAL PERFORMANCE OF ANN MODELS FOR TRAINING, VALIDATION AND TEST DATA-SETS

Model	RMSE (mm)	R	E	MAPE (%)	NMBE (%)
Training					
BPNN	2.4942	0.9816	0.9625	1.2720	0.2497
ANN-GA	1.9439	0.9885	0.9772	1.0013	0.0004
Validation					
BPNN	3.6037	0.9453	0.8906	1.6755	0.1950
ANN-GA	2.7176	0.9692	0.9378	1.2939	0.0138
Testing					
BPNN	3.3873	0.9499	0.8948	1.5862	-0.5249
ANN-GA	2.7436	0.9659	0.9370	1.3042	0.0531

The trained BPNN and ANN-GA models were validated and tested. The results in terms of statistical performance metrics are exhibited in Table I.

The trained models were also tested using the entire RMC data. Regression plots were developed for BPNN and ANN-GA models between the actual and the predicted slump values and are shown in Figs. 7 and 8 respectively. The statistical performance for the entire data-set is tabulated in Table II.

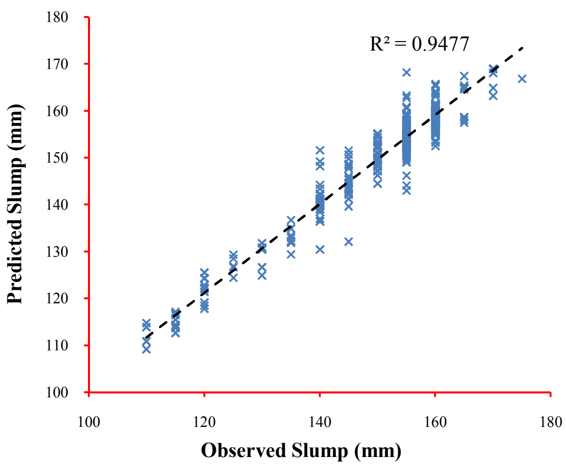

Fig. 7 Regression plot for BPNN model

TABLE II
STATISTICAL PERFORMANCE OF ANN MODELS FOR THE ENTIRE DATA-SET

Model	RMSE (mm)	R	E	MAPE (%)	NMBE (%)
BPNN	2.8326	0.9735	0.9467	1.3797	-0.2570
ANN-GA	2.4197	0.9804	0.9611	1.1346	0.0129

V. DISCUSSIONS

On analyzing the results it can be inferred that by hybridizing GA with ANN, the drawback of the BP algorithm getting trapped at local minima and slow convergence can be easily avoided. In comparison to ANN trained using the BP algorithm (BPNN) which took 1374 epochs and 50.903 seconds to reach the desired level of performance, the hybrid ANN-GA took only 63 epochs and 41.7293 seconds (including GA time to evolve weights and biases) to achieve the same performance.

The ANN-GA model provided a good prediction during training, validation and testing of the trained model. This is proved by higher values of statistics R, E and lower values of statistics MAPE and RMSE. In comparison to BPNN model, which provided NMBE values 0.2497%, 0.1950% and -0.5249%, the ANN-GA model gave 0.0004%, 0.0138% and 0.0531% values during training, validation and testing phases respectively. This indicates the consistency of prediction provided by the hybrid ANN-GA model.

The performance statistics computed for the entire dataset using the trained ANN-GA model, showed a lower RMSE, MAPE value of 2.4197 mm and 1.1346% respectively and a higher E and R value of 0.9611 and 0.9804 respectively. The value of NMBE statistics -0.2570% and 0.0129% for BPNN and ANN-GA models respectively, indicates that BPNN model is under predicting the slump values whereas ANN-GA achieved a near optimal prediction of slump values. Overall, the statistical analysis shows that, ANN-GA has consistently outperformed the prediction accuracy of BPNN models.

VI. Relative Importance of Concrete Design Mix Proportions on Slump Value

In the neural network, the connection weights between neurons are the links between the inputs and the outputs, and therefore are the links between the problem and the solution [30]. The connection weights can be used to interpret the influence of the input variables and understand the role played by each neuron in the hidden layer [31]. The procedure given by Garson [32] called the "Weights Method", involves partitioning the hidden-output connection weights of each hidden neuron into components associated with each input neuron [33]. The product of input-hidden layer neuron weights w_{ij} (i represents the input neuron and j represents the hidden neuron) and hidden-output layer neuron weights v_{jk} (j represents hidden neuron and k represents the output neuron) are summed across all hidden neurons. The relative contributions of the variables are calculated by dividing the absolute value of each variable contribution by the grand sum of all absolute contributions. Equation (7) gives percentage impact Q_{ik} of the input variable xi on the output y_k [34].

$$Q_{ik} = \frac{\sum_{j=1}^{N}\left(\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk}\right)}{\sum_{i=1}^{N}\left(\sum_{j=1}^{L}\frac{w_{ij}}{\sum_{r=1}^{N} w_{rj}} v_{jk}\right)} \times 100 \tag{7}$$

where $\sum_{r=1}^{N} w_{rj}$ denotes the sum of connection weights between the input neurons N and the hidden neuron j. Fig. 9 shows the relative importance of concrete's design mix constituents on the slump value.

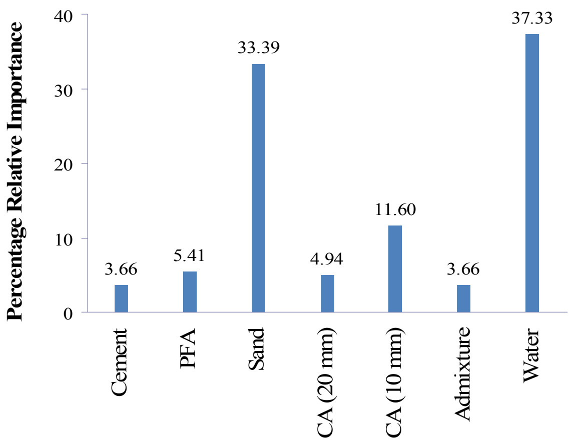

Fig. 9 Relative importance of concrete constituents on slump value

The results evaluated using "Weights Method" reveal that water content in the concrete has a predominant effect on the workability contributing 37.33% relative importance to the slump value since it improves fluidity of the fresh concrete by lubricating the particles through formation of a water film. Sand imparted 33.39% importance due to its higher volumetric percentage in the concrete paste. Sand influences workability by reducing the inter-particle friction thereby increasing workability. But very fine sands require more paste for a given consistency leading to mixes that are harsh and unworkable. Also a higher fine to coarse aggregate ratio decreases consistency and increases cohesiveness. The relative importance of coarse aggregates (CA) of size 20 mm and 10 mm was cumulatively evaluated as 15.54%. Due to smaller size and light weight of CA (10 mm), it needs small effort to overcome the inter-particle frictional resistance. For the same volume of aggregates, CA (20 mm) due to their larger size, has smaller surface area demanding smaller coating of cement paste. Hence a large proportion of cement paste is left for aiding the workability of concrete.

PFA (Pulverised Fly Ash) is generally used as a partial replacement for cement. In the present study PFA imparted an importance of 5.41% to the concrete slump since its finer particles covers the interstitial pores in concrete microstructure thereby making more water available for lubrication. Increase of cement content at a particular water content leads to excellent cohesiveness, but these mixes tend to be sticky. A lower cement content will lead to harsh mixes. In case of RMC, PFA is mostly used as partial replacement for cement from strength and economy point of view. Hence, due to incorporation of PFA in the concrete design mix, cement content showed a smaller relative importance of only 3.66% on the slump value. The addition of admixture causes delayed setting of concrete allowing it to remain in green state for longer duration of time. Due to small dosage of admixture in the design mix considered in the study, the admixture constituent is shown to contribute 3.66% importance on the slump value.

VII. Conclusions

Modeling slump of concrete based on concrete's design mix proportions forms a highly non-linear problem which is difficult to be modeled using conventional mathematical techniques. ANN with BP training algorithm has been traditionally used for modeling material behavior because of its simple implementation. BP algorithm is based on gradient descent technique and its convergence probability depends primarily on the initial weights and biases, which introduces a drawback significantly affecting the performance of ANN. The hybridization of ANN with GA not only covers up the drawback of BP algorithm to converge at suboptimal points, but this amalgamation of two distinct approaches helps in deriving the best from global search ability of GA and local search ability of BP algorithm. The proposed hybrid technique harnessed GA to evolve the optimal set of initial neural network weights and biases which were further fine tuned using LMBP algorithm.

The study showed that in comparison to often used BPNN approach, the hybrid ANN-GA model gave consistent predictions during training, validation and testing phases, indicating the robustness of the hybrid modeling approach. Moreover, the ANN-GA model took less time in converging to the desired level of performance than the BPNN model. It can therefore be inferred that GA evolved initial neural network weights and biases enable faster convergence of BP algorithm. The proposed hybrid model can be used as a decision support tool, aiding the technical staff to easily predict the slump value for a particular concrete design mix. This technique will considerably decrease the effort and time to design a concrete mix for a customized slump without undertaking multiple trials.

Due to composite nature of concrete, the relative importance of each design mix constituent on the concrete slump cannot be ascertained directly. The "Weights method" based on the value of the synaptic weights of ANN offered a simplified technique of assessing the relative importance of neural network inputs. Using this technique the relative importance of concrete mix constituents on slump value was conveniently evaluated and inferences regarding the effect of each constituent on slump value were deduced. The study showed that water content, sand and coarse aggregates are the key ingredients in concrete design mix which impart maximum influence on the slump value of concrete.

REFERENCES

[1] P.K. Mehta and P.J.M. Monteiro, *Concrete: Structure, Properties and Materials*. 3rd ed. New York: McGraw Hill, 2006.

[2] Z. Li, *Advanced Concrete Technology*. 1st ed. New Jersey: John Wiley & Sons, Inc, 2011.

[3] W.P.S. Dias and S.P. Pooliyadda, "Neural Networks for predicting properties of concrete with admixtures," *Construction and Building Materials*, vol. 15, no. 7, pp. 371-379, 2001.

[4] I.-C. Yeh, "Exploring concrete slump model using artificial neural networks," *Journal of Computing in Civil Engineering*, vol. 20, no. 3, pp. 217-221, 2006.

[5] A. Oztas, M. Pala, E. Ozbay, E. Kanca, N. Caglar and M.A. Bhatti, "Predicting the compressive strength and slump of high strength concrete using neural network," *Construction and Building Materials*, vol. 20, no. 9, pp. 769-775, 2006.

[6] I.-C.Yeh, "Modeling slump flow of concrete using second-order regressions and artificial neural networks," *Cement and Concrete Composites*, vol. 29, pp. 474-480, 2007.

[7] A. Jain, S.K. Jha and S. Misra, "Modeling and analysis of concrete slump using artificial neural networks," *Journal of Materials in Civil Engineering*, vol. 20, no. 9, pp. 628-633, 2008.

[8] M. Saridemir, "Prediction of compressive strength of concretes containing metakaolin and silica fumes by artificial neural networks," *Advances in Engineering Software*, vol. 40, no. 5, pp. 350-355, 2009.

[9] S.J. Kwon and H.W. Song, "Analysis of carbonation behaviour in concrete using neural network algorithm and carbonation modeling," *Cement and Concrete Research*, vol. 40, no. 1, pp. 119-127, 2010.

[10] R. Siddique, P. Aggarwal and Y. Aggarwal, "Prediction of compressive strength of self compacting concrete containing bottom ash using artificial neural networks," *Advances in Engineering Software*, vol. 42, no. 10, pp. 780-786, 2011.

[11] M.I. Khan, "Predicting properties of high performance concrete containing composite cementitious materials using Artificial Neural Networks," *Automation in Construction*, vol.22, pp. 516-524, 2012.

[12] O.A. Hodhod and H.I. Ahmed, "Developing an artificial neural network model to evaluate chloride diffusivity in high performance concrete," *HBRC Journal*, vol.9, no. 1, pp. 15-21, 2013.

[13] A.M. Diab, H.E. Elyamany, A.E.M.A. Elmoaty and A.H. Shalan, "Prediction of concrete compressive strength due to long term sulphate attack using neural networks," *Alexandria Engineering Journal*, vol.53, pp. 627-642, 2014.

[14] C.L. Su, S.M. Yang and W.L. Huang, "A two stage algorithm integrating genetic algorithms and modified Newton method for neural network training in engineering systems," *Expert Systems with Applications*, vol. 38, no. 10, pp. 12189-12194, 2011.

[15] A. Johari, A.A. Javadi and G. Habibagahi, "Modelling the mechanical bahaviour of unsaturated soils using a genetic algorithm based neural network," *Computers and Geotechnics*, vol. 38, no. 1, pp. 2-13, 2011.

[16] H. Karimi and F. Yousefi, " Application of artificial neural network-genetic algorithm (ANN-GA) to correlation of density in nanofluids," *Fluid Phase Equilibria*, vol. 336, pp. 79-83, 2012.

[17] R. Wang,C. Zhou, Z. Deng, B. Ni and Z. Zhao, "Predicting f_oF_2 in China region using the artificial neural networks improved by the genetic algorithms," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 92, pp. 7-17, 2013.

[18] Y. Xue, L. Cheng, J. Mou and W. Zhao, "A new fracture prediction method by combining genetic algorithm with neural network in low-permeability reservoirs," *Journal of Petroleum Science and Engineering*, vol. 121, pp. 159-166, 2014.

[19] M.M. Alshihri, A.M. Azmy and M.S. El-Bisy, "Neural Networks for predicting compressive strength of structural light weight concrete," *Construction and Building Materials*, vol. 23, no. 6, pp. 2214-2219, 2009.

[20] K. Hornik, M. Stinchcombe and H. White, "Multilayer feed forward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.

[21] S. Tamura and M. Tateishi, "Capabilities of four layered feedforward neural network: four layer versus three," *IEEE Trasactions on Neural Networks,* vol. 8, no. 2, pp. 251-255, 1997.

[22] P.C. Pendharkar and J.A.Rodger, "Technical efficiency based selection of learning cases to improve the forecasting efficiency of neural networks under monotonicity assumption,", *Decision Support Systems*, vol. 36, no. 1, pp. 117-136, 2003.

[23] K. Jinchuan and L. Xinzhe, "Empirical analysis of optimal hidden layer neurons in neural network modeling for stock prediction," *in Proceedings of Pacific-Asia Workshop on Computational Intelligence and Industrial Applications,* vol. 2, pp. 828-832, Dec. 2008.

[24] D. Hunter, Y. Hao, M.S. Pukish, J. Kolbusz and B.M Wilamowski, "Selection of proper Neural Network sizes and architectures-A comparative study," *IEEE Transaction on Industrial Informatics*, vol. 8, no. 2, pp. 228-240, 2012.

[25] S. Rajasekaran and G.A.V. Pai, "*Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications*," New Delhi: Prentice-Hall of India Private Limited, 2003..

[26] B.M. Wilamowski, Y. Chen, A. Malinowski, "Efficient Algorithm for Training Neural Networks with One Hidden Layer," *in Proceedings of International Joint Conference on Neural Networks IEEE*, pp. 1725-1728, 1999.

[27] K. Wang, *Computational Intelligence in Agile Manufacturing Engineering,* in: Gunasekaran A, editor. Agile Manufacturing The 21st Century Competitive Strategy, Oxford, UK: Elsevier Science Ltd, 2001, pp. 297-315.

[28] J.E. Nash and J.V. Sutcliffe, "River flow forecasting through conceptual models Part I – a discussion of principles," *Journal of Hydrology*, vol. 10, no. 3, pp. 282–290, 1970.

[29] S. Srinivasulu and A. Jain, "A comparative analysis of training methods for artificial neural network rainfall-runoff models," *Applied Soft Computing*, vol. 6, pp. 295-306, 2006.

[30] J.D. Olden and D.A. Jackson, "Illuminating the "Black Box": a randomization approach for understanding variable contributions in artificial neural networks." Ecological Modeling ,vol. 154, pp. 135-150, 2002.

[31] G. Acciani, E. Chiarantoni and G. Fornarelli, *A neural network approach to study O_3 and PM_{10} concentration,* in: Kollias S, Staflopatis A, Duch W, Oja E, editors. ICANN'06 Proceedings of the 16th international conference on Artificial Neural Networks - Volume Part II, Berlin, Germany: Springer Verlag, 2006, pp. 913-922.

[32] G.D. Garson, "Interpreting neural network connection weights,"*Artificial Intelligence Expert* , vol. 6, pp. 47-51, 1991.

[33] M. Gevrey, I. Dimopoulos and S. Lek, "Review and comparison of methods to study the contribution of variables in artificial neural network models," *Ecological Modeling* , vol. 160, pp. 249-264, 2003.

[34] J.J. Montano and A. Palmer, "Numeric sensitivity analysis applied to feedforward neural networks," *Neural Computing and Applications*, vol. 12, pp. 119-125, 2003.