The Use of an Artificial Neural Network in the Prediction the Compressive Strength of Concrete

Vanderci F. Arruda[1], Gray F. Moita[1], Eliene P. Carvalho[1], Priscila F. S. Silva[1]

[1] *Centro Federal de Educação Tecnológica de Minas Gerais*
Programa de Pós-Graduação em Modelagem Matemática e Computacional
Av. Amazonas, 7675, Nova Gameleira, 30.510-000, Belo Horizonte, MG, Brazil.
vanderci-engcivil@hotmail.com; gray@cefetmg.br; eliene@cefetmg.br; 201422800040@aluno.cefetmg.br;

Abstract. This study proposes the use of an artificial neural network (ANN) in a mechanical characteristic prediction of a material widely explored in the design of structures, the concrete. The conventional way to obtain the mechanical characteristics of such a material is by means of expensive and costly laboratory tests and, as an alternative, the use of an intelligent simulating system can be proposed. ANNs are based on a bio-inspired model of the biological neuron, which processes data from simple units. In this study, a well-known (and established in the academic literature) database was used. The artificial neural network tested used the supervised learning method and the networks were trained based upon the following algorithms: classical backpropagation, backpropagation with momentum, backpropagation with learning rate, backpropagation with momentum and learning rate, and Levenberg-Marquardt. This work includes the use of a preprocessing strategy on the input data and different backpropagation training algorithms. The main objective of this work was to obtain reliable results to estimate the compressive strength of concrete by using machine learning. Among the training algorithms tested, the one that presented the best performance was the Levenberg-Marquardt, which proved to be effective in predicting the compressive strength of concrete at twenty-eight days, obtaining, as performance metrics, the RMSE of 4.39 MPa and the coefficient of determination of 0.93. From these results it is possible to verify that this method proved to be reliable for the calculation of the compressive strength of concrete by reducing possible errors and amplifying the reliability of the application of computational technology in engineering projects.

Keywords: Artificial Neural Networks, Intelligent Systems, Computational Methods, Machine Learning, Computational Intelligence.

1 Introduction

The resistance shown by high performance concrete is the reason why it is being widely used in the concrete industry. Being composed of the basic components of conventional concrete (Portland cement, fine aggregate, coarse aggregate and water), it has complementary cementitious materials such as fly ash, blast furnace slag, chemical additives and superplasticizer. This concrete is also used in chemically aggressive environments and in extreme environmental conditions such as marine structures, bridges, and tunnels. Additives can be used for the purpose of increasing strength and improving characteristics such as workability and durability. These materials can also be used with the purpose of reducing the final price of concrete, since in the concrete mix Portland cement is the most expensive component. According to Chou [1], the partial replacement of cement with combined pozzolanic industrial by-products decreases energy consumption and the release of hazardous air emissions.

Among the problems of materials science is the prediction of mechanical characteristics of construction materials, Sobhani [2]. As an essential item in structural engineering, the compressive strength of concrete at twenty-eight days is a parameter determined by specific standards. The concrete compressive strength is considered the most important quality of high-performance concrete; thus, the fast and reliable prediction of concrete compressive strength is a fundamental item for pre-design and quality control. For this purpose, experimental tests, batching diagrams and, alternatively, the use of computer methods can be used. Experimental destructive testing is an activity that demands time, planning, and financial resources. According to Silva et al. [3], the compressive strength of concrete is one of the most used parameters in structural calculation and its determination is the result of several studies; however, this mechanical characteristic of concrete can be affected by nonlinear factors when, for example, compression tests that involve a destructive process of the concrete specimens are used. Several studies show that the compressive strength of concrete is not only affected by the water/cement factor (Abrams law), but also by the other components. Understanding the relationship between the components is necessary to optimize concrete mixtures, Popovics[4].

One of the alternatives to obtain the relationship between the constituents and the compressive strength of concrete is to obtain a regression equation with empirical basis; however, besides being a difficult task, the analysis of different factors that affect the compressive strength of high-performance concrete are not seen in conventional concrete. To combat the limitations of conventional methods, intelligent systems can be used. In this way, machine learning can be used not only to generate knowledge, but also as a means of modeling general information using statistical parameters that approximate the relationships between inputs and outputs based on a measured set of data, Reich [5].

This study proposes to determine the compressive strength of concrete by means of artificial neural networks. Similar work has been developed to accurately predict the compressive strength of concrete, as in Zarandi et al. [6]; Topçu and SarIdemir [7]; and Yeh and Lien [8].

2 Data Description and Preprocessing

The dataset used in this paper was chosen from an intensive database search. Thus, the dataset available in the studies by Yeh [9] was used. The concrete compressive strength is characterized by the presence and quantity of the following constituents: coarse and fine aggregate, cement content, fly ash, blast furnace slag, water, superplasticizer content and age.

The dataset contains 1030 concrete compressive strength test results based on various proportions of constituents. In order to assess the data dispersion, the data range is shown in Table 1.

The compressive strength of concrete is a mechanical property that depends on the degree of cement hydration over time, so each sample has a specific strength associated with a specific age. Thus, there is no pattern that represents all the values of the input variable concrete age. It is emphasized that, for this study, the computational model was developed to predict compressive strength values at ages ranging from 1 to 365 days.

Variable	Data range	
Cement (kg/m³)	540.00	102.00
Blast-furnace slag (kg/m³)	359.40	0.00
Fly ash (kg/m³)	200.10	0.00
Water(kg/m³)	247.00	121.80
Superplasticizer (kg/m³)	32.20	0.00
Coarse aggregate (kg/m³)	1145.00	801.00
Fine aggregate (kg/m³)	992.60	594.00
Age (days)	365.00	1.00
Concrete compressive strength (MPa)	82.60	2.33

Table 1 - Basic statistics of dataset

The datasets may have values that have different scales, which causes difficulty for visualization and may even worsen the predictive performance of some of the computational methods. Unstandardized data can reduce the speed of implementation or even prevent the convergence of estimators, Wang, Wang and Alexander [10].

Some models are built by the assumption that their parameters take on values close to zero, i.e., at comparable scales. Estimators using metrics and gradient assume that the data are standardized [10]. However, preprocessing techniques prove effective and can improve the performance of computational models. Thus, the dataset was scaled to comparable scales using Normalizer as a technique that standardizes features by removing the mean and scaling to unit variance.

3 Performance Metrics

In this study, the mean square error (MSE), given by Equation (1), the root mean square error (RMSE), as per Equation (2), and the coefficient of determination (R²), given by Equation (3), are used as performance metrics:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \tag{1}$$

$$RMSE = \sqrt{MSE} \tag{2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{3}$$

In the above equations, y_i is the observed value, $\bar{y}$ is the mean of the observations, $\hat{y}$ is the predicted value of y_i and N is the total number sampling in the data set.

4 Artificial Neural Networks

One of the main characteristics of an ANN is the gradual improvement of its performance, either in classification or regression. Learning is an iterative process of parameter adjustments, and can occur in a supervised, unsupervised, semi-supervised, or reinforcement manner. Supervised learning occurs when predefined input patterns are passed to the network in conjunction with output results and error-based adjustment are used for the predictions. Conversely, when there are no pre-set patterns, the learning is unsupervised.

In supervised learning the error is calculated at each iteration, where the obtained values are checked against the target outputs. This error is returned to the learning system and the weights applied to the neurons are adjusted. The aim is to obtain the smallest difference between the obtained response and the desired response.

According to Hagan et al. [11], there are different possibilities of training algorithms, and these can be based on gradient or Jacobian methods. It is essential to use training algorithms compatible with the case under study. The descending gradient is widely used in solving machine learning problems and was used in the implementation of some of the training algorithms proposed in this paper. In an iterative way, the algorithm obtains the parameter values that minimize a given function of interest.

Several training algorithms have been evidenced in the last decades [11], among those used in this study are gradient descent backpropagation, gradient descent backpropagation with momentum, gradient descent backpropagation with learning rate, gradient descent backpropagation with momentum and learning rate and Levenberg-Marquardt backpropagation. The main difference in these methods is in the way the weights are updated. Items such as momentum rate and learning rate can be included in the formulation.

The formula used for training and updating the weights for each epoch is as follows:

$$w_{ji}(h) = w_{ji}(h-1) + \Delta w_{ji}(h). \tag{4}$$

The change in $\Delta w_{ji}(h)$ is:

$$\Delta w_{ji}(h) = \eta\delta_{pi}x_{pi} + \alpha\Delta w_{ji}(h-1). \tag{5}$$

In Equation (5), η is the learning rate parameter, δ_{pi} is the propagated error, x_{pi} is the output of neuron i for record p, α is the momentum parameter, and $\Delta w_{ji}(h-1)$ is the change in w_{ji} in the previous cycle.

The use of ANNs depends on the existence of a relationship between the input and output variables, since neural networks have the ability to learn from the knowledge extracted from a set of data, being able to interpolate and extrapolate what they have learned. Thus, the quantity and quality of the data set is an essential factor for the simulations. In the case of ANNs, a previous step before moving on to the training phase is to divide the database into 3 distinct sets: training, validation, and test.

Among the best-known neural models is the MultiLayer Perceptron Backpropagation (MLP). The first step in the operation of this ANN is called feedforward, that is, the forward propagation of data, where the values are updated in each layer. At first, the input is multiplied by the neuron weights and added to the bias, according to Equation (6). After this procedure, the result obtained is then used in the activation function, according to Equation (7), determining the output parameter.

$$u_k = \sum_i W_{ki} x_i + b_k \tag{6}$$

$$y_k = f(u_k) \tag{7}$$

The second step involves the term backpropagation, backtracking. The predicted value is compared to the target value and thus the error of the iteration is determined, according to Equation (1). After the error is calculated, neuron weights are updated until performance parameters or stopping criteria are met. Thus, the MultiLayer Perceptron Backpropagation network makes forward and backward calculations, Koivo [12].

The implementation of the Artificial Neural Networks used in this work was done using the MatLab software, specifically the graphical interface toolbox (NNtool). The MLP network was the alternative chosen for having the capacity to solve pattern recognition problems as well as to perform regression analysis.

The methodology applied in this study was as follows:

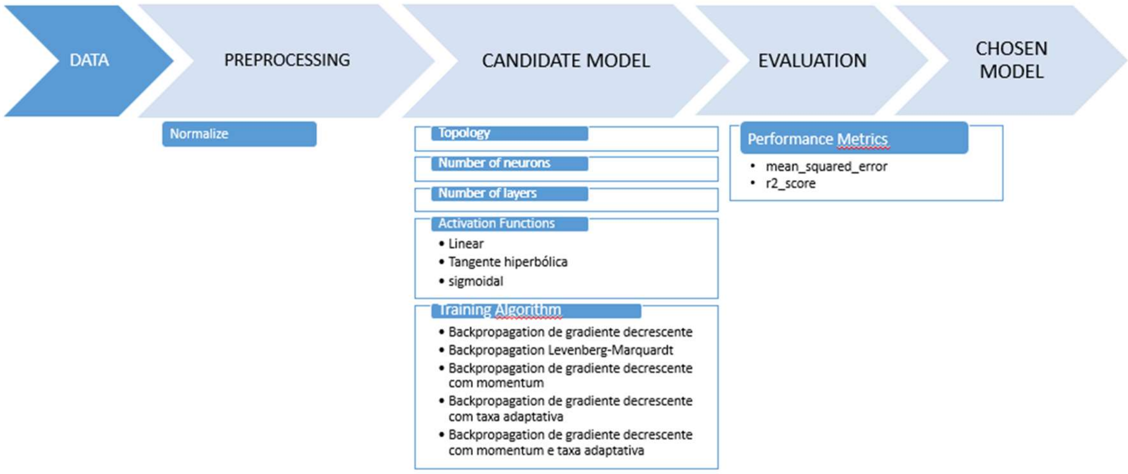

Figure 1 – the methodology used

The first step in creating an ANN is defining the input data. The dataset was partitioned into training, validation, and test data in the ratio 70/15/15. The raw input data was put into comparable scales using Normalizer as a preprocessing strategy. No preprocessing task was performed on the output data.

Once the data entry was performed, the network type was chosen, and the feedforward backpropagation network was used. The implementation carried out took into consideration the training algorithms.

A further step was to define the network architecture. Two topology options were employed in this work: the first with one hidden layer and the other with two hidden layers.

Another essential item in the development of the artificial neural network is the choice in the number of neurons used. The current study proposes the trial-and-error method to obtain the number of neurons of the "optimal" network architecture. Thus, in the topology with a hidden layer, 4 to 13 neurons were used. In the case of the second topology, the number of neurons with the best performance in the first topology was used for each training algorithm, varying the number of neurons in the second layer from 4 to 13 neurons.

Selecting parameters for training

One of the problems in building an ANN is in defining its parameters, and each item has its function. The number of epochs defines the number of training cycles, that is, the number of times that the training set is presented to the network. The desired final error is to stop training after the Mean Squared Error falls below a given value. The learning rate influences the training process; a low rate slows down the learning process, while a high-rate causes oscillations in training and prevents convergence in the learning process. The purpose of the momentum rate is to increase the network's training speed and decrease the risk of instability. To train the network, the convergence and stopping criteria shown in Table 2 were used.

Training Data or Stop Criteria	
number of epochs	**1000.00**
Goal	**0.00**
learning rate	**0.01**
Momentum Rate	**0.90**

Table 2 - convergence and stopping criteria

Once the initial data was defined, the implementation of the neural networks was done using the MatLab toolbox (NNtool), and RMSE and R^2 were used as performance metrics.

Identifying the best result (Error and Correlation)

Next, the best results are ordered using the training algorithms and the description of the topologies used, and the average of 10 simulations is performed to obtain the performance metrics, as defined in Table 3. The values of the performance metrics reflect the data of training, validation, and testing.

Number of hidden layers	Activation function type	Number of neurons	Type of training role									
			Backpropagation								Levenberg Marquardt	
			Classic		Momentum		Taxa adaptativa		Mom./ tx. Adap.			
			RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2	RMSE	R^2
1	logsig/logsig	6	15.65	0.46								
	logsig/logsig	4			17.17	0.26						
	purelin/purelin	8					9.98	0.62				
	tansig/tansig	4							9.14	0.70		
	tansig/tansig	13									4.77	0.93
2	purelin/purelin/purelin	6_5	34.51	0.20								
	logsig/logsig/tansig	4_10			23.43	0.07						
	purelin/purelin/purelin	8_7					11.57	0.51				
	tansig/tansig/tansig	4_10							8.96	0.69		
	tansig/tansig/tansig	13_7									4.39	0.93

Table 3 - Results obtained in the implementation

Among the topologies tested, the one that presented the best performance was obtained with the Levenberg-Marquardt training algorithm, and this was the topology with two intermediate layers, having 13 neurons in the first hidden layer, and 7 neurons in the second hidden layer.

As an activation function, the one that showed the best response was the hyperbolic tangent function in all layers, as presented in Table 3, obtaining an RMSE of 4.39 MPa and a correlation coefficient of 0.93.

Figure 1 shows a portion of the test data and its targets, that is, the original values and the values predicted by the Artificial Neural Network implemented in this study. The Figure 1 presents the data referring to the test plot of the database.

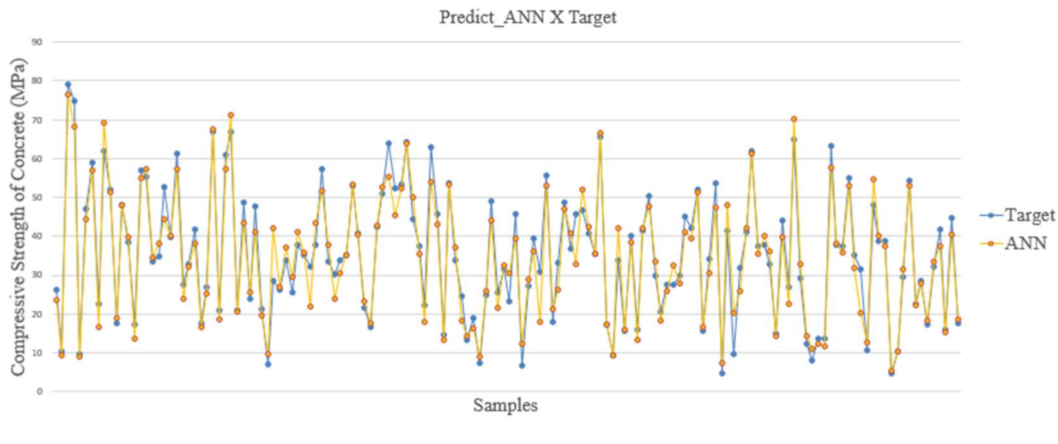

Figure 2 - Original versus predicted results for ANN

5 Comparison between the results

The scatterplot of the predicted value and the actual value of the training, validation and test (100% of the data) are shown in figure 3.

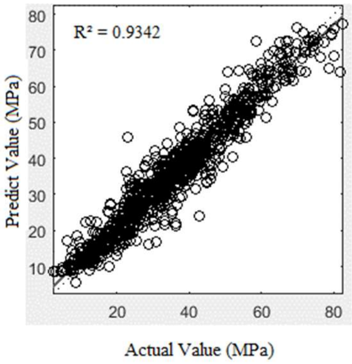

Figure 3 - Scatter diagram of training, validation, and testing data

Table 4 shows the comparison between the results obtained in this study and the results in the literature. As shown in Table 4, the performance results measured by RMSE obtained in this study are in the same order of magnitude of those obtained in previous studies. However, in general terms, the ANN now developed presented the best result compared to other studies, comparing the root mean square error values. The R^2 values presented by the ANN were higher than those obtained in the listed studies.

Research	Algorithm	R^2	RMSE (MPa)
Chou et al. [13]	RNA	0.91	-
	SVM	0.88	-
Chou et al.[14]	RNA	-	7.95
	SVM	-	5.59
Young et al. [15]	RNA	0.82	6.30
	SVM	0.83	6.40
Proposed method	RNA	0.93	4.39

Table 4 - Comparison between present and literature results

Based on the results above, it is possible to conclude that computational methods can be used as an alternative way to complement other tools in this area, such as the batching diagram and the concrete compressive strength test on specimens, to determine the concrete strength at twenty-eight days.

6 Conclusion

The general objective of this work was to present an analysis of a computational method to predict the compressive strength of concrete at twenty-eight days using Artificial Neural Networks, employing the dataset given by Yeh [9].

The results obtained show that the ANN presented the best performance with an RMSE value of 4.39 MPa and $R^2 = 0.93$. The ANN had an overall error rate that can be considered low compared to studies available in the literature. It is worth noting that the dataset used in the work contains most of the samples with twenty-eight days. Hence, values of this order of magnitude will have a better agreement between the predicted and the experimental values, while other values will present greater discrepancies.

It is possible to conclude that the use of this technique to obtain the compressive strength of concrete proved satisfactory compared to studies available in the literature. Therefore, the use of this method should be encouraged because it is an alternative way to determining the mechanical characteristic of concrete, in this case, the compressive strength.

7 Acknowledgments

CAPES and CEFET-MG supported the work described in this paper.

References

[1] J. S. Chou et al. Optimizing the prediction accuracy of concrete compressive strength based on a comparison of data-mining techniques. *Journal of Computing in Civil Engineering*, v. 25, n. 3, p. 242-253, 2011.
[2] J. Sobhani et al. Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and ANFIS models. *Construction and Building Materials*, v. 24, n. 5, p. 709-718, 2010.
[3] P. S. Silva; G. F. Moita; V. F. Arruda. Machine learning techniques to predict the compressive strength of concrete. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*, v. 36, n. 4, 2020.
[4] S. Popovics . "Analysis of the concrete strength versus water-cement ratio relationship." *ACI Mater. J.*, 87(5), 517–529 1990.
[5] Y. Reich. Machine learning techniques for civil engineering problems. *Computer-Aided Civil and Infrastructure Engineering*, v. 12, n. 4, p. 295-310, 1997.
[6] F. Zarandi et .al. "Fuzzy polynomial neural networks for approximation of the compressive strength of concrete." *Appl. Soft Comput.*, 8(1), 488–498. 2008
[7] I. B Topçu,.; M. Saridemir . "Prediction of compressive strength of concrete containing fly ash using artificial neural networks and fuzzy logic." *Comput. Mater. Sci.,* 41(3), 305–311. 2008
[8] I. C.Yeh; L. C.Lien, . "Knowledge discovery of concrete material using genetic operation trees." *Expert Syst. Appl.*, 36(3), 5807–5812. 2009
[9] I. C. Yeh. Modeling concrete strength with augment-neuron networks. *Journal of Materials in Civil Engineering*, v. 10, n. 4, p. 263–268, 1998.
[10] L. Wang; G. Wang; C. A. Alexander. Big data and visualization: methods, challenges and technology progress. *Digital Technologies*, v. 1, n. 1, p. 33–38, 2015.
[11] Hagan et al. Neural networks in a softcomputing framework. *Neural Networks in a Softcomputing Framework*, p. 1–566, 2006.
[12] H. N. Koivo. NEURAL NETWORKS: Basics using MATLAB Neural Network Toolbox By. *Neural Networks*, p. 1–59, 2008.
[13] J. S. Chou et al. Optimizing the Prediction Accuracy of Concrete Compressive Strength Based on Optimizing the Prediction Accuracy of Concrete Compressive Strength Based on a Comparison of Data-Mining Techniques. *JOURNAL OF COMPUTING IN CIVIL ENGINEERING*, 2011.
[14] J. S. Chou et al. Machine learning in concrete strength simulations: Multi-nation data analytics. *Constr. Build. Mater.*, 73:771–780, 2014.
[15] B.A. Young et al. Can the compressive strength of concrete be estimated from knowledge of the mixture proportions? New insights from statistical analysis and machine learning methods. *Cem. Concr. Res.*, 115:379–388, 2019.