TẠP CHÍ KHOA HỌC & CÔNG NGHỆ ĐẠI HỌC DUY TÂN
DTU Journal of Science and Technology 5(60) (2023) 10-15

TRƯỜNG ĐẠI HỌC DUY TÂN
DUYTAN UNIVERSITY

Interval estimation of compressive strength of concrete using artificial neural network developed with Python

Dự báo theo khoảng cường độ chịu nén của bê tông sử dụng mạng nơ-ron thần kinh nhân tạo được phát triển với Python

Hoang Nhat Duc[a,b*], Nguyen Quoc Lam[a,b] , Pham Quang Nhat[a,b]
Hoàng Nhật Đức[a,b*], Nguyễn Quốc Lâm[a,b] , Phạm Quang Nhật[a,b]

[a]Institute of Research and Development, Duy Tan University, Da Nang, 550000, Vietnam
[a]Viện Nghiên cứu và Phát triển Công nghệ Cao, Đại học Duy Tân, Đà Nẵng, Việt Nam
[b]Faculty of Civil Engineering, Duy Tan University, Da Nang, 550000, Vietnam
[b]Khoa Xây dựng, Trường Đại học Duy Tân, Đà Nẵng, Việt Nam

(Ngày nhận bài: 12/8/2023, ngày phản biện xong: 20/8/2023, ngày chấp nhận đăng: 21/9/2023)

Abstract

The compressive strength (CS) of concrete mixes is a crucial parameter. This paper aims to construct an artificial neural network (ANN) model for interval estimation of the CS of concrete blended with ground granulated blast furnace slag (GGBFS). The nonlinear regression based method is employed to derive the prediction intervals. A historical dataset, including 533 samples, is used to train and test the ANN approach. The contents of cement, GGBFS, water, plasticizer, coarse aggregate, and fine aggregate are input variables which are used for estimating the CS values at different ages. The model is developed in Python by the authors and deposited in Github repository. Experimental results show that the ANN model with 8 neurons is able to achieve good prediction performance with a coefficient of determination of 91%. In addition, the prediction interval with a 90% confidence level can capture roughly 91% of the testing data samples.

Keywords: Concrete strength, Blast furnace slag, Artificial neural network, Prediction interval.

Tóm tắt

Cường độ chịu nén là một thông số rất quan trọng của hỗn hợp bê tông. Nghiên cứu của chúng tôi xây dựng một mô hình mạng thần kinh nhân tạo để ước lượng theo khoảng cường độ chịu nén của hỗn hợp bê tông trộn với xỉ hạt lò cao nghiền. Phương pháp dựa trên hồi quy phi tuyến tính được sử dụng để tính toán các khoảng dự đoán. Một bộ dữ liệu, bao gồm 533 mẫu, được sử dụng để huấn luyện và kiểm chứng phương pháp được đề xuất. Hàm lượng xi măng, GGBFS, nước, chất hóa dẻo, cốt liệu thô và cốt liệu mịn là các biến đầu vào, được sử dụng để ước tính các giá trị cường độ chịu nén ở các độ tuổi khác nhau. Mô hình đã được các tác giả phát triển bằng Python và được lưu trữ trong Github. Kết quả thực nghiệm cho thấy mô hình mạng thần kinh nhân tạo với 8 nơ-ron có khả năng dự báo tốt với hệ số xác định là 91%. Ngoài ra, khoảng dự đoán với mức độ tin cậy 90% có thể bao phủ được khoảng 91% mẫu dữ liệu thử nghiệm.

Từ khóa: Cường độ bê tông, Xỉ lò cao, Mạng thần kinh nhân tạo, Dự báo theo khoảng.

* *Corresponding Author:* Hoang Nhat Duc
Email: hoangnhatduc@duytan.edu.vn

1. Introduction

Artificial neural network (ANN)-based models have been widely employed as a nonlinear regression tool for the prediction of concrete compressive strength (CS) [1]. However, the vast majority of ANN-based models dealt with point estimations of the CS, without the concern of expressing the degree of confidence associated with them. To better support the decision making in concrete mix design, construction engineers and researchers should be provided with the uncertainties associated with the point estimations. These uncertainties are often characterized by interval forecasting techniques, which are used to derive a prediction interval (PI). A PI refers to a range of values in which an estimated random variable is expected to fall with a specific coverage probability (e.g., 95%). This probability is often known as the confidence level (CL) [2]. In addition, the width of the PI also expresses information regarding the uncertainty of the prediction. Hence, for a certain CL, the narrower the width of the PI is, the more accurate the prediction outcome is.

This study first constructs an ANN model in Python and then incorporates the nonlinear regression-based approach for estimating the PI. The constructed model is trained and verified by a historical dataset that includes 533 samples. This dataset was compiled in [3]. The contents of cement, GGBFS, water, plasticizer, coarse aggregate, and fine aggregate are independent variables, which are employed for predicting the CS values at different ages. The developed model is uploaded to the Github repository and can be openly downloaded. The subsequent sections of the paper are organized as follows: The second section reviews the research method; the third section reports the experimental results; the last section summarizes the study with several concluding remarks.

2. Research method

2.1. Prediction intervals

Prediction intervals (PIs) are bounds of an estimated variable (e.g., the CS of concrete mixes) constructed from a collected dataset. PIs enclose a point estimation within a lower bound (LB) and an upper bound (UB) with a given probability or CL [2]. Thus, it is able to express the probability of an estimated value as follows:

$$\Pr(LB \le Y \le UB) = CL \tag{1}$$

To assess the quality of the constructed PI, Prediction Interval Coverage Probability (PICP) is used. This index measures the percentage of the predicted variables which lie within the LB and UB of the PI [4]. PICP is computed as follows:

$$PICP = \frac{1}{N}\sum_{i=1}^{N} \chi_i \tag{2}$$

where N is the number of data samples; $\chi_i = 1$ if $y_i \in [LB_i, UB_i]$ and $\chi_i = 0$ otherwise.

For a given CL, the narrower the PIs are, the more useful they are. It is because narrow PIs contain less uncertainty and convery more information about the predicted value of the CS of concrete. Therefore, mean width of prediction interval (MWPI), which computes the average width of the PIs, is also used for evaluating the prediction results [5]. MWPI is given by:

$$MWPI = \frac{1}{N}\sum_{i=1}^{N}(UB_i - LB_i) \tag{3}$$

2.2 Artificial neural network for regression analysis

For point estimation of a target variable, an ANN model with an input, a hidden layer, and an output layer can be used [6]. At the 1st hidden layer, the input $(z_i^{(1)})$, weighted sum of input $(v_i^{(1)})$, and output $(y_i^{(1)})$ can be computed as follows [3]:

$$z_i^{(1)} = x_i, i = 1,2,...,D \text{ and } z_{D+1}^{(1)} = 1 \quad (4)$$

$$v_i^{(1)} = z_1^{(1)} \times w_{i,1}^{(1)} + z_2^{(1)} \times w_{i,2}^{(1)} + ... + z_{D+1}^{(1)} \times w_{i,D+1}^{(1)} = \sum_{k=1}^{D+1} z_k^{(1)} \times w_{i,k}^{(1)} \quad (5)$$

$$y_i^{(1)} = f_A(v_i^{(1)}) \quad (6)$$

where f_A is an activation function (e.g. sigmoid function); $W^{(1)} = [w_{i,k}^{(1)}]$ (with $i = 1,2,...,M$ and $k = 1,2,...,D+1$) denotes the matrix of the synaptic weights connecting an input node to a neurons in the hidden layer. M denotes the number of neurons in the hidden layer.

At the output layer, the predicted variable is given by:

$$y = z_1^{(M)} \times w_1^{(M)} + z_2^{(M)} \times w_2^{(M)} + ... + z_{N_{M-1}+1}^{(M)} \times w_{N_{M-1}+1}^{(M)} = \sum_{k=1}^{N_{M-1}+1} z_k^{(M)} \times w_k^{(M)} \quad (7)$$

where M is the number of neuron in the last hidden layer. $W^{(2)} = [w_k^{(2)}]$ (k = 1,2,…, M+1) is the matrix of the synaptic weights connecting a neuron in the hidden layer to the output neuron.

The squared error loss (SEL) is often used for training the ANN model, this loss function is given by:

$$L = \frac{1}{2}(t-y)^2 \quad (8)$$

where t denotes the actual value of the depended variable.

To adapt $W^{(1)}$ and $W^{(2)}$, the gradient descent and error backpropagation algorithms can be used. These algorithms requires the computation of the derivative of L with respect to each synaptic weight $\partial L / \partial w_{i,k}^{(1)}$ and $\partial L / \partial w_{i,k}^{(2)}$. To calculate those derivatives, the generalized delta rule can be employed [7]. In general, element of the two weight matrices are revised as follows:

$$w = w - \xi \times \frac{\partial L}{\partial w} \quad (9)$$

where ξ denotes the learning rate parameter.

2.3. Interval estimation with artificial neural network regression

The construction of PIs for the ANN regression model in this study relies on the method of [8], which is based on the theory of nonlinear regression. The review work of [9] is highly useful for explaining the method used in [8]. The $(1-\alpha)$% PI for an estimated y_i is computed as follows:

$$y_i \pm t_{n-p}^{1-0.5\times\alpha} s_\varepsilon \sqrt{1 + g_i^T (F^T F)^{-1} g_i} \quad (10)$$

where $t_{n-p}^{1-0.5\times\alpha}$ denotes the (0.5α) quantile of a cumulative t-distribution function with $n-p$ degrees of freedom [9]. F is the Jacobian matrix of the ANN model with respect to its synaptic weights.

An asymptotically unbiased estimate of the variance of random errors (ε) is given by [8]:

$$s_\varepsilon = \sqrt{\sum_{i=1}^{N}(t_i - y_i)/(N-P)} \quad (11)$$

where P is the number of the model's parameters.

3. Results and discussions

In this study, to evaluate the performance of the ANN model in point estimation, the indices of RMSE, MAPE, and R^2 are used. These indices are given by:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - t_i)^2} \quad (12)$$

$$MAPE = \frac{100}{N} \times \sum_{i=1}^{N} \frac{|y_i - t_i|}{y_i} \quad (13)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(t_i - y_i)^2}{\sum_{i=1}^{N}(t_i - \bar{t})^2} \quad (14)$$

where t_i and y_i are the actual and predicted variable of the i^{th} data point, respectively. N denotes the number of the data points.

```
class ArtificialNeuralNetworkRegression:
    def __init__(self):
        self.W0 = np.zeros((1,1))
        self.W1 = np.zeros((1,1))
        self.M = 8
        self.alpha = 0.1
        self.MaxEp = 10
        self.name = 'Artificial Neural Network Regression Model'

    def Train(self, Xtr, Ttr):
        W0 = CreateGaussRandMatrix(self.M, D_x+1, 0, 1) # M x D_x+1
        W1 = CreateGaussRandMatrix(1, self.M+1, 0, 1) # 1 x M+1
        ...
    def Predict(self, Xte):
        Nte = Xte.shape[0]
        D_x = Xte.shape[1]
        W0 = self.W0.copy()
        W1 = self.W1.copy()
        Yte = np.zeros(Nte)
        for i in range(Nte):
            # forward
            x_0 = Xte[i,:].reshape(D_x, 1)
            z_0 = np.vstack((x_0, 1))
            v_0 = np.matmul(W0, z_0)
            y_0 = np.zeros((v_0.shape[0], 1))
        ...
```

Fig. 3.1 The model's class coded in Python

```
def Compute_Bound(Xtr, Ttr, FtFinv, s, st, W0, W1, Level = 0.95):
    RowSize = W0.size + W1.size
    Ntr = Xtr.shape[0]
    D_x = Xtr.shape[1]
    Bounds = np.zeros(Ntr)
    for i in range(Ntr):
        x = Xtr[i, :]
        t = Ttr[i]
        dW_x = compute_one_sample_gradient(x, t, W0, W1)
        gi = np.copy(dW_x).reshape((RowSize,1))
        giT = gi.transpose()
        A = np.matmul(giT, FtFinv)
        B = np.sqrt(1 + np.matmul(A, gi))

        Bound = st * s * B
        Bounds[i] = Bound
    return Bounds
```

Fig. 3.2 The function used for computing prediction intervals

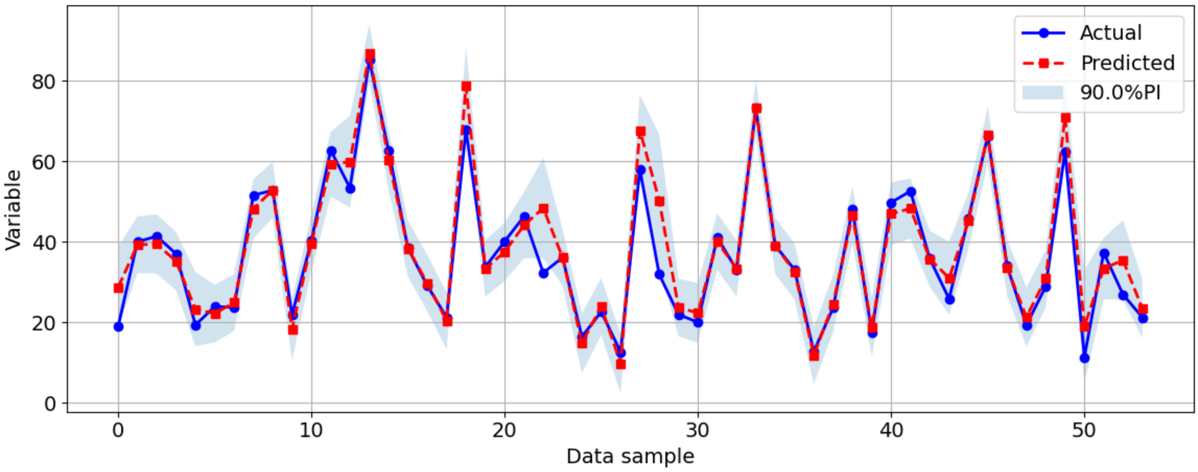

Fig. 3.3 Interval estimation of concrete blended with GGBFS

The general steps for estimating the CS of concrete in this study are as follows: (i) The dataset is randomly divided into a training (90%) and testing set (10%); (ii) an ANN model is trained by the samples in the training set; the performance of the trained model is evaluated via the indices of RMSE, MAPE, and R^2; (iii) the index of s_ε is computed by **Eq. 11**. The CL is selected (e.g., 90%). The PI for the training and testing sets are calculated; and (iv) the quality of the PI is evaluated by the metrics of PICP and MWPI. It is noted that the model has been developed in Python and stored in Github repository. The illustration of the model's class is provided in **Fig. 3.1**. The function employed for computing the PI is demonstrated in **Fig. 3.2**.

A dataset, that includes 533 samples and seven predictor variables, is used [3]. The contents of cement, GGBFS, water, plasticizer, coarse aggregate, and fine aggregate are utilized for predicting the CS values at different ages. In this study, the ANN model includes 8 neurons in the hidden layer; it has been trained during 100 epochs and the learning rate is set to be 0.1. The training performance of the model is as follows: RMSE = 3.89; MAPE = 8.05%; and R^2 = 0.96. The testing performance of the model is as follows: RMSE = 5.00; MAPE = 10.66%; and R^2 = 0.91. The CL for computing PIs of the model is set to be 90%. The PICP values of the training and testing phases are 95.62% and 90.74%, respectively. In addition, the MWPI values for the training and testing data samples are 14.90 MPa and 15.93 MPa, respectively. These outcomes demonstrate that ANN is a capable tool for interval estimation of the CS of concrete mixtures.

4. Conclusions

The CS of concrete mixes is a crucial parameter, required to be estimated during the mixture design phase. This study employed ANN and nonlinear regression-based method for constructing the PI of the CS values. A historical data, including 533 samples and seven features, was used to train and test the built model. Experimental results show that the ANN model is capable of explaining roughly 91% variation in the CS of GGBFS-blended concrete mixes. In addition, the ANN model also yielded reliable PIs of the CS values with PICP of 90.74% for the CL of 90%. The future extensions of the current work include the use of other advanced methods for constructing PI for ANN regressor and the employment of other state-of-the-art optimizers for training the ANN models.

Supplementary material

The Python code and the dataset used to support the findings of this study have been deposited in the Github repository at: https://github.com/NHDDTUEDU/ANNR_PI_CS.

References

[1] Ben Chaabene W, Flah M, Nehdi ML. (2020). Machine learning prediction of mechanical properties of concrete: Critical review. *Construction and Building Materials* (260), 1-18. doi:https://doi.org/10.1016/j.conbuildmat.2020.119889

[2] Taormina R, Chau K-W. (2015). ANN-based interval forecasting of streamflow discharges using the LUBE method and MOFIPS. *Engineering Applications of Artificial Intelligence* (45), 429-440. doi:https://doi.org/10.1016/j.engappai.2015.07.019

[3] Hoang N. D. (2023). Estimation of the compressive strength of concretes containing ground granulated blast furnace slag using a novel regularized deep learning approach. *Multiscale and Multidisciplinary Modeling, Experiments and Design* (6), 415-430. doi:10.1007/s41939-023-00154-z

[4] Khosravi A, Nahavandi S, Creighton D. (2010). Construction of Optimal Prediction Intervals for Load Forecasting Problems. *IEEE Transactions on Power Systems* (25), 1496-1503. doi:10.1109/TPWRS.2010.2042309

[5] Cheng M-Y, Hoang N-D. (2014). Interval Estimation of Construction Cost at Completion Using Least Squares Support Vector Machine. *Journal of Civil Engineering and Management* (20), 223-236. doi:https://doi.org/10.3846/13923730.2013.801891.

[6] Tran T-H, Hoang N-D. (2016). Predicting Colonization Growth of Algae on Mortar Surface with Artificial Neural Network. *Journal of Computing in Civil Engineering* (30), 1-8. doi: 10.1061/(ASCE)CP.1943-5487.0000599

[7] Hoang N-D, Nguyen O-L, Pham Q-N. (2023). Training artificial neural network regression based on the generalized delta rule: a case study in modeling the compressive strength of concrete. *DTU Journal of Science & Technology* (2), 23-30.

[8] De VlEaux RD, Schumi J, Schweinsberg J, Ungar LH. (1998). Prediction Intervals for Neural Networks via Nonlinear Regression. *Technometrics* (40), 273-282. doi:10.1080/00401706.1998.10485556

[9] Khosravi A, Nahavandi S, Creighton D, Atiya AF. (2011). Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances. *IEEE Transactions on Neural Networks* (22), 1341-1356. doi:10.1109/TNN.2011.2162110