Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **xx**, No. x, pp. 1-27

DOI: 10.4208/aamm.OA-2022-0127
xxx 2024

Compressive Strength Prediction of High-Performance Hydraulic Concrete using a Novel Neural Network Based on the Memristor

Jun Lu[1], Lin Qiu[2], Yingjie Liang[3], Ji Lin[3,*]

[1] *Nanjing Hydraulic Research Institute, Nanjing, Jiangsu 210029, China*
[2] *College of Mechanical and Electrical Engineering, National Engineering Research Center for Intelligent Electrical Vehicle Power System, Qingdao University, Qingdao, Shandong 266071, China*
[3] *Key Laboratory of Coastal Disaster and Defence (Ministry of Education), College of Mechanics and Materials, Hohai University, Nanjing, Jiangsu 210098, China*

Received 5 May 2022; Accepted (in revised version) 8 February 2023

Abstract. This paper proposes the memristor-memristor (M-M) and the memristor-gradient descent (M-GD) neural networks based on the classical back propagation neural network. The presented models are employed to predict the compressive strength of high-performance hydraulic concrete (HPC), and are tested by well-fitting and accurate predictions with the experimental data. The developed algorithms are also evaluated through comparisons with the classical learning algorithms including the gradient descent method, the gradient descent with momentum, the gradient descent with adaptive learning rate, the elastic gradient descent, and the Levenberg-Marquardt algorithm. It is observed that the established M-GD generally outperforms the classical algorithms and M-M. The constructed M-M neural network has a quite high convergence speed, and the strength prediction error induced by it can roughly satisfy the demands in construction engineering. This work extends the nonlinear memristor to a brand-new field, and provides an effective methodology for forecasting the compressive strength prediction of HPC.

AMS subject classifications: 92B20, 65K05, 60G25

Key words: Artificial neural network, memristor, high-performance concrete, compressive strength.

1 Introduction

As one of the most widely used building materials in modern engineering structures in the world, the concrete plays an important role in the building construction and civil en-

*Corresponding author.
Email: linji@hhu.edu.cn (J. Lin)

gineering. Driven by the development of modern capital construction and new construction technology, the special demand of concrete performance in construction industry becomes critical. The high-performance concrete (HPC) [1–3] is a kind of construction material with many special properties which include high workability, high strength, high volume stability, and high durability [4]. The HPC has gained increasing popularity and been extensively studied over the last decade. It has potential to be used in a large number of huge projects [5] such as nuclear reactor, sea-crossing bridge, nuclear waste containers, and submarine tunnel.

Compressive strength is a critical mechanical property to measure the quality of HPC which is a hot topic in the HPC research recently. Compared with strength prediction of conventional concrete, evaluation of HPC strength is relatively difficult. Chou et al. [6] demonstrated that the compressive strength of HPC shows a highly non-linear relationship with each component of concrete, and so some characteristics of the HPC are not completely understood. This unfortunately leads to the inability of traditional predicting methods [7, 8], which are generally based on maturity concept [9, 10] of concrete or the value of fresh concrete test [11] to be used.

In the last decade, the artificial neural network (ANN) [12–15] has been extensively employed in determining the concrete strength due to its self-organization, self-adaptability, reasoning ability, and self-learning ability [16]. Kasperkiewicz et al. [17] stated that the ANN could be applied to forecast the compressive strength of HPC and it also shows promising potentials in optimizing concrete mixes. Nehdi et al. [18] illustrated that the ANN is able to accurately estimate the concrete slump, the segregation, and the filling capacity. It is also proved by Kim et al. [19] that the ANN is a powerful tool in forecasting the compressive strength of concrete based on the mix proportions. Furthermore, combined with other techniques, the neural network has been improved to some extent [20–22]. The results show that the neural network model has high accuracy and strong predictive ability, and its superiorities in the strength prediction of HPC with complex internal rules are more obvious.

Considering the complexity of the neural network itself, its weight needs to be constantly adjusted and updated. For the circuit implementation of neural networks, the circuit design of artificial neural network synapses has always been a thorny problem. The integrated circuits and ultra-large-scale integrated circuits have been tried to complete the design of artificial synapses. However, due to the complexity of circuits and traditional devices, its large size and high power consumption make it difficult for such artificial synaptic circuits to be highly integrated, and also make its density difficult to achieve the requirements of biological neural networks. Currently, transistors are used to construct hardware circuits of neural networks which are mainly for the simulation of neurons and synapses. However, this technical settlement derived from the Very Large Scale Integration Circuit is restricted by the huge number of transistors required for the simulation and its high power consumption. In addition, synaptic transmission characteristics are affected by time and space. Therefore, it is a tough work to construct the hardware circuit of the neural network with the existing components. Fortunately, the emergence of memristor provides the possibility to resolve this problem. The memristor is the fourth fundamental element, which was originally envisioned by circuit theorist Chua [23] in 1971. In recent years, it has been fully demonstrated its usefulness in the integrated circuit design [24], the large-capacity non-volatile memory [25], the artificial intelligence [26] and the neural network [27,28]. This paper makes the first attempt to employ the memristor-based neural network to the compressive strength prediction of HPC. Based on the classical back propagation (BP) neural network, the memristor-memristor (M-M) neural network and the memristor-gradient descent (M-GD) neural network are firstly developed in this paper. These neural networks are a step closer to the circuit implementation, which occupy less computer resources and can achieve high efficiency. With the data mined from other references, the proposed models are adopted and validated to the compressive strength fitting and prediction. From the results, it is confirmed that the built M-M neural network has a quite high convergence speed, and the strength prediction error can roughly meet the demands in construction engineering. In addition, the neural network combining the classical learning algorithms and memristor are firstly developed and applied. It can be convinced from the results that the established M-GD generally outperforms the classical algorithms and M-M. It is expected to expand a brand-new field for the application of memristor. The main contribution of this paper is that we make the first attempt to apply the memristor to construct the neural networks. Compared with the traditional neural networks, the proposed neural networks are built relying on the memristor. It is easy to develop the corresponding hardware circuits. The input data is represented by the voltage signal. By changing the voltage values at both end of the memristor, the weights and biases can be directly updated, so that the training and testing of the neural network can be realized. The whole process does not need computer simulation which makes it possible to break through the development bottleneck of traditional neural networks.

A brief outline of this paper is arranged as follows. Section 2 gives an introduction to memristor. In Section 3, the classical BP neural network is briefly presented first, then two kinds of memristor-based neural network are introduced. In Section 4, the memristor-based neural networks are utilized to predict the HPC strength in the first example, the fitting and prediction effect of the proposed models, and the results of comparisons with different algorithms are displayed. Furthermore, the another example further verifies the effectiveness and practicality of the M-GD. Finally, some conclusions are drawn in Section 5.

2 Memristor

A simplified device of the memristor [29,30] is considered and the diagram with a simplified equivalent circuit is shown in Fig. 1. Strukov et al. [29] developed the physical model of the memristor as follows:

$$\begin{cases} v(t)=M(t)i(t), \\ M(t)=R_{ON}x(t)+R_{OFF}[1-x(t)], \\ \dfrac{dx(t)}{dt}=ki(t), \end{cases} \tag{2.1}$$

where $M(t)$ is the effective resistance of memristor, $i(t)$ the input current, $v(t)$ the output voltage, $k=\mu_v R_{ON}/D$ the memristor constant and $\mu_v=10^{-14}\mathrm{m}^2\cdot\mathrm{s}^{-1}\cdot V^{-1}$, $x(t)$ the ratio between the length of doped region to the total length of the memristor, i.e., $x(t)=w/D$ in Fig. 1(a). When $w=D$, the resistance of the memristor achieves the minimum value R_{ON}. When $w=0$, the resistance of the memristor reaches the maximum value R_{OFF}.

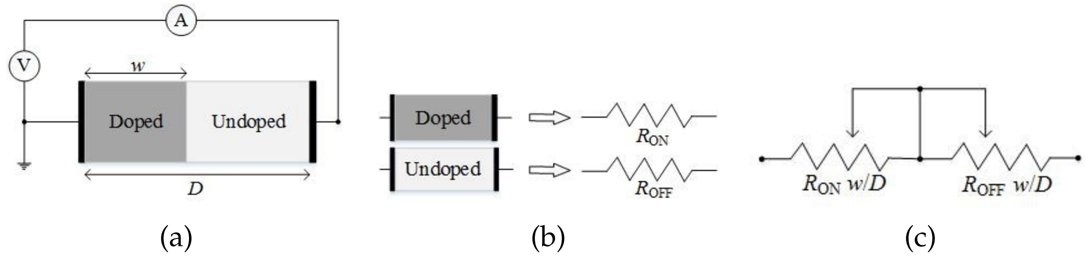

Figure 1: Diagram with a simplified equivalent circuit.

According to the Bernoulli dynamics analysis, the flux control model of memristor can be obtained. The relationship between the charge and the magnetic flux is as follows [30]:

$$q(t)=\begin{cases} \dfrac{\varphi(t)-c_1}{R_{OFF}}, & \varphi(t)<c_3, \\ \dfrac{\sqrt{2h\varphi(t)+M^2(0)}-M(0)}{k}, & c_3\le\varphi(t)<c_4, \\ \dfrac{\varphi(t)-c_2}{R_{ON}}, & \varphi(t)\ge c_4, \end{cases} \tag{2.2}$$

in which $M(0)$ represents the initial resistance of the memristor, $\varphi(t)$ stands for the magnetic flux, $q(t)=\int_{-\infty}^{t}i(t)dt$ is the charge, and

$$h=(R_{ON}-R_{OFF})\mu_v R_{ON}/D^2, \tag{2.3a}$$

$$c_1=-[R_{OFF}-M(0)]^2/(2h), \tag{2.3b}$$

$$c_2=-[R_{ON}-M(0)]^2/(2h), \tag{2.3c}$$

$$c_3=\left[R_{OFF}^2-M^2(0)\right]/(2h), \tag{2.3d}$$

$$c_4=\left[R_{ON}^2-M^2(0)\right]/(2h). \tag{2.3e}$$

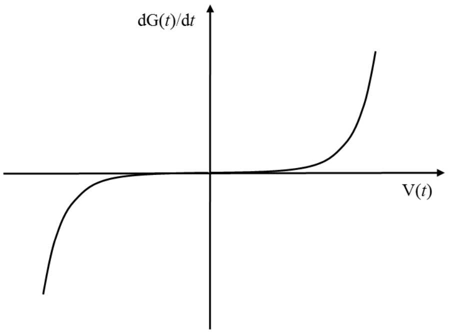

Figure 2: A representative curve for the relationship between the derivative of conductance and the voltage of the memristor.

Inserting Eq. (2.2) into the second and third equations in Eq. (2.1), the conductance of memristor can be obtained as

$$G(t)=\frac{1}{M(t)}=\begin{cases} \dfrac{1}{R_{OFF}}, & \varphi(t)<c_3, \\ \dfrac{1}{\sqrt{2h\varphi(t)+M^2(0)}}, & c_3\le\varphi(t)<c_4, \\ \dfrac{1}{R_{ON}}, & \varphi(t)\ge c_4. \end{cases} \tag{2.4}$$

Then, we can obtain the following equation:

$$\frac{dG(t)}{dt}=\begin{cases} \left[2h\varphi(t)+M^2(0)\right]^{-3/2}\times(-h)\times\dfrac{d\varphi(t)}{dt}, & c_3\le\varphi(t)<c_4, \\ 0, & \text{else}. \end{cases} \tag{2.5}$$

By Eq. (2.5), we can easily plot a representative curve for the relationship between the derivative of conductance of the memristor and the voltage at both ends of the memristor, as shown in Fig. 2.

It can be found that the curve is "sinh-like", so HP Laboratory proposed a model of voltage-controlled memristor [28] as follows:

$$\frac{dG(t)}{dt}=A\sinh(BV), \tag{2.6}$$

in which A and B are the constants to be determined.

It can be observed from Fig. 2 that this kind of voltage-controlled memristor describes the relationship between voltage and conductance. It is noted that small voltages trigger very little change in device conductance, but when the memristor voltage reaches a certain value, it will induce drastic changes. Due to this important feature of memristor, this model connects with the BP neural network, and memristor-based neural network models can be constructed.

3 Memristor-based neural network

3.1 Classical BP neural network

In this section, the BP neural network will be briefly introduced firstly. It should be mentioned that the main factors affecting the strength of HPC are water to binder ratio (W/B, %), water content (W, kg/m^3), fine aggregate ratio (s/a, %), fly ash replacement ratio (FAR, %), air-entraining agent content (AE, kg/m^3), and superplasticizer content (SP, kg/m^3). The six factors are selected as input variables and the compressive strength value as output variables. Therefore, we introduce a BP neural network with 6 inputs and 1 output directly. The classical BP neural network has the structure as described in Fig. 3.

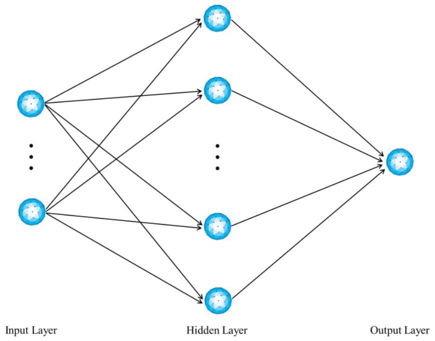

Figure 3: The structure of BP neural network.

It can be seen that three layers are contained in this BP neural network, i.e., input layer, hidden layer, and output layer. The basic idea of BP neural network is the method of mean square error, which can be employed to modify the connection weight of network when combined with gradient descent method (GD) [31]. The modification of weight aims to minimize the final output error or to achieve the desired value. The learning algorithm is an extremely important part of BP neural network. In order to improve the performance of neural network, many other advanced algorithms have been consecutively proposed, such as the gradient descent with momentum (GDM) [32], the gradient descent with adaptive learning rate (GDA) [33], the elastic gradient descent (EGD) [34], and the Levenberg-Marquardt algorithm (L-M) [35]. These five classical algorithms are also utilized to compare with the proposed model in this study.

3.2 Memristor-based neural network

3.2.1 M-M neural network

Unlike the traditional neural networks, we regard the memristor as the learning algorithms in the newly developed networks. The function tanh is employed as an activation

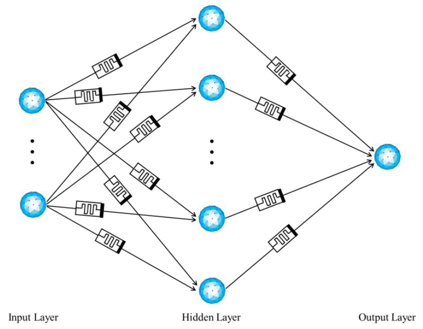

Figure 4: M-M neural network architecture.

function. We adopt an M-M network structure, as depicted in Fig. 4. Assuming that the input variable is x_j, $j=1,\cdots,m$ and the output value is y, the specific algorithm of the M-M neural network is as follows:

(1) Input layer:

$$Inp_j = x_j, \tag{3.1}$$

(2) Input of hidden layer:

$$Hid_i^I = \sum_{j=1}^{m} w_{ij}^{(1)} Inp_j + b_i^{(1)}, \tag{3.2}$$

$w_{ij}^{(1)}$ and $b_i^{(1)}$ are the weights and biases between the input layer and the hidden layer, respectively.

(3) Output of hidden layer:

$$Hid_i^O = \tanh(Hid_i^I), \tag{3.3}$$

(4) Output layer:

$$y = \sum_{i=1}^{n} w_i^{(2)} Hid_i^O + b^{(2)}, \tag{3.4}$$

in which n is the number of neurons in the hidden layer, $w_i^{(2)}$ and $b^{(2)}$ are the weights and biases between the hidden layer and the output layer, respectively.

(5) Weight updating rules: According to Eq. (2.6), for a very small Δt, dG can be approximately replaced by ΔG. Then, we can write

$$\Delta G = A\sinh(BV) \times \Delta t. \tag{3.5}$$

Corresponding the conductance of the memristor to the weight of the neural network, we obtain

$$\Delta w_k^{(1)} = A\sinh[B(M_k - y_k)] \times \alpha^{(1)} \times Inp, \tag{3.6a}$$

$$w_k^{(1)}(i+1) = w_k^{(1)}(i) + \Delta w_k^{(1)}(i), \tag{3.6b}$$

and

$$\Delta w_k^{(2)} = A\sinh[B(M_k - y_k)] \times \alpha^{(2)} \times Hid^O, \tag{3.7a}$$

$$w_k^{(2)}(i+1) = w_k^{(2)}(i) + \Delta w_k^{(2)}(i), \tag{3.7b}$$

in which M_k is the kth measured value, y_k the kth value obtained by M-M neural network, $\alpha^{(1)}$ the learning rate between the input layer and the hidden layer, $\alpha^{(2)}$ the learning rate between the hidden layer and the output layer.

(6) Bias updating rules: Similar to (5), the bias updating rules can be written as follows:

$$\Delta b_k^{(1)} = A\sinh[B(M_k - y_k)] \times \alpha^{(1)}, \tag{3.8a}$$

$$b_k^{(1)}(i+1) = b_k^{(1)}(i) + \Delta b_k^{(1)}(i). \tag{3.8b}$$

and

$$\Delta b_k^{(2)} = A\sinh[B(M_k - y_k)] \times \alpha^{(2)}, \tag{3.9a}$$

$$b_k^{(2)}(i+1) = b_k^{(2)}(i) + \Delta b_k^{(2)}(i). \tag{3.9b}$$

3.2.2 M-GD neural network

Unlike the construction of the M-M neural network, we adopt a combination of classical learning algorithm and memristor in this network, as shown in Fig. 5. The GD is used for learning and updating, the memristor is also arranged in the network, and the activation function is tanh. Assuming that the input variable is x_j, $j=1,\cdots,m$, and the output value is y, the procedures of the M-GD neural network are as follows:

(1) Input layer:

$$Inp_j = x_j, \tag{3.10}$$

(2) Input of the hidden layer:

$$Hid_i^I = \sum_{j=1}^{m} w_{ij}^{(1)} Inp_j + b_i^{(1)}, \tag{3.11}$$

(3) Output of the hidden layer:

$$Hid_i^O = \tanh(Hid_i^I), \tag{3.12}$$

(4) Output layer:

$$y=\sum_{i=1}^{n} w_i^{(2)} Hid_i^O + b^{(2)}, \tag{3.13}$$

(5) Updating rules between the input layer and hidden layer: The GD is employed for network learning, and optimize weight and bias.

(6) Updating rules between the hidden layer and output layer: The memristor is considered and the rules are the same as that in Eqs. (3.7) and (3.9).

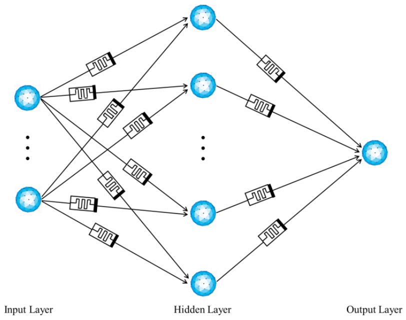

Figure 5: M-GD neural network architecture.

4 Memristor-based neural network for predicting the HPC strength

4.1 Example 1

4.1.1 The dataset

The experimental dataset was obtained from [16,36,37]. In this example, a total of 104 sets of the data were invoked, and the data is summarized in detail in "Appendix A". To train the established neural network method, the learning results need to be tested. To this end, the 12 sets (Numbers 2, 10, 17, 30, 40, 48, 55, 63, 75, 83, 93, and 101. For convenience, we numbered them 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 in turn) of the data were selected as the prediction samples, and the remaining 92 sets were used as training samples.

4.1.2 Tuning parameters

Parameters of classical BP neural network. In our analysis, the learning rate is assumed to be 0.05, and the activation functions are tansig and purelin. To determine the number of neurons, we tested the performance of the classical BP neural network under different numbers of neurons in the hidden layer shown in Table 1. Considering the

Table 1: Comparison of BP neural networks under different numbers of neurons.

Number of neurons	4	5	6	7	8	9	10	11	12
MSE of training results	0.6017	0.4297	0.3310	0.3224	0.3180	0.3109	0.3096	0.3047	0.3096
MSE of predictive results	3.3856	3.5315	2.9261	2.7787	2.0828	2.3130	2.4094	2.7457	2.8962

trends of both training and prediction errors, the number of neurons in the hidden layer is set to 8 for the BP neural network in the following study.

Parameters of M-M neural network and M-GD neural network. In our methods, the learning rate between the input layer and the hidden layer is assumed to be 0.008, and the learning rate between the hidden layer and output layer is 0.007. The activation function is tanh. After testing, the number of neurons in the hidden layer in the M-M neural network can be set to be very small, which is set to 3. However, the number of neurons in the hidden layer in the M-GD neural network needs to be larger, which is fixed at 15.

4.1.3 Performance measurement

To measure, compare, and evaluate the training and predictive results of the developed methods comprehensively, this study employs the following six performance-measurement equations:

Absolute error:

$$AE=|M_i-P_i|, \tag{4.1}$$

Relative error:

$$RE=\frac{|M_i-P_i|}{M_i}\times 100, \tag{4.2}$$

Mean absolute error:

$$MAE=\frac{1}{N}\sum_{i=1}^{N}|M_i-P_i|, \tag{4.3}$$

Mean relative error:

$$MRE=\frac{1}{N}\sum_{i=1}^{N}\left(\left|\frac{M_i-P_i}{M_i}\right|\times 100\right), \tag{4.4}$$

Mean square error:

$$MSE=\frac{1}{N}\sum_{i=1}^{N}(M_i-P_i)^2, \tag{4.5}$$

Correlation coefficient:

$$R=\frac{\sum_{i=1}^{N}(M_i-\bar{M}_i)(P_i-\bar{P}_i)}{\sqrt{\sum_{i=1}^{N}(M_i-\bar{M}_i)^2\sum_{i=1}^{N}(P_i-\bar{P}_i)^2}}, \tag{4.6}$$

where N is the number of data samples, M_i the measured value, and P_i the predictive value, $\bar{M}_i$ the mean measured value, and $\bar{P}_i$ the mean predictive value.

4.1.4 Results and discussion

Results of M-M neural network. The training process of the M-M neural network is displayed in Fig. 6, in which the abscissa indicates the number of iterations in the training process, and the ordinate represents the MSE of the training results. It can be observed that the proposed method has a quite high convergence speed. The fitting effect of training samples is shown in Fig. 7. As can be seen from Fig. 7, the fitting results are not very ideal, and many data points cannot be accurately caught, but the overall trend of the fitting curve is consistent with the measured strengths.

The trained M-M neural network can be used for strength prediction. The 12 sets of prediction samples prepared in Section 4.1 are tested on the trained network, and the prediction effect of the samples of the M-M is displayed in Fig. 8. It can be found that the consistency between the predictive and measured strength values of the proposed network is not high, which indicates that this network cannot estimate the compressive strength of the samples very accurately.

The absolute errors and relative errors of the predictive value obtained by the M-M are displayed in Table 2. As reflected in Table 2, except for the first set of samples, the absolute error of the prediction samples reaches 4.92MPa, the maximum absolute error

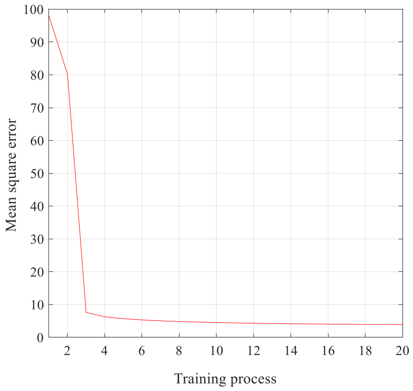

Figure 6: The training error curve of the M-M neural network.

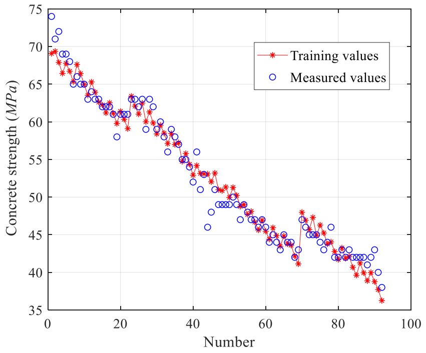

Figure 7: The fitting effect for the number of training samples of the M-M neural network.

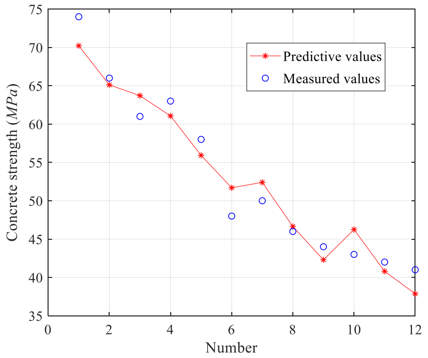

Figure 8: The prediction effect for the number of prediction samples of the M-M neural network.

of the other 11 sets prediction samples is 3.69MPa, and the maximum relative error is 7.69%. The requirements of construction engineering are [38],

$$\text{Maximum absolute error} \leq 4MPa,$$

and

$$\text{Maximum relative error} \leq 10\%.$$

Based on the above prediction results, the strength prediction errors of the M-M neural network can roughly satisfy the requirements in construction engineering.

In order to improve the accuracy of the proposed M-M neural network, we use 8 number of neurons in the hidden layer in the M-M neural network. Fig. 9 displays the fitting effect of training samples. Compared with the Fig. 7, we can see that much more number of neurons used in the hidden layer will improve the accuracy significantly. Furthermore, the prediction effect of the samples of the M-M is displayed in Fig. 10. It can be found that the consistency between the predictive and measured strength values of the proposed network is high compared with Fig. 8.

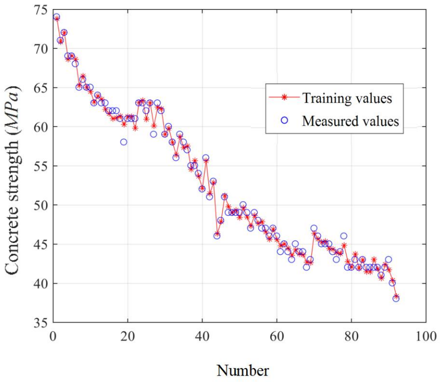

Figure 9: The fitting effect for the number of prediction samples of the M-M neural network with 8 neurons in the hidden layer.

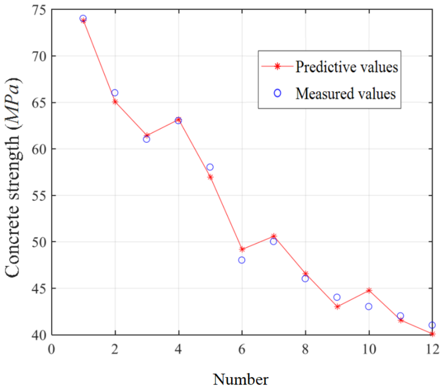

Figure 10: The prediction effect for the number of prediction samples of the M-M neural network with 8 neurons in the hidden layer.

Results of M-GD neural network. For the M-GD neural network, the training process is given in Fig. 11, the abscissa in it indicates the number of iterations in the training process, and the ordinate represents the MSE of the training results. It is evident that the convergence rate of the proposed scheme is fast. The fitting effects of training samples and the prediction effect of prediction samples are presented in Figs. 12 and 13, respectively. As can be observed from Fig. 12, the fitting results are quite ideal, only a few data points cannot be accurately fitted, and the general trend of the fitting curve is consistent with the measured strength. It can be seen from Fig. 13 that the prediction curve is very close to the measured strength, which shows that the M-GD neural network has the ability to predict the strength of prediction samples accurately, and the predictive strength of the network is in good agreement with the measured strength. Fig. 14 presents the scatter diagram of the measured and predictive value, in which the regression coefficient (RC)

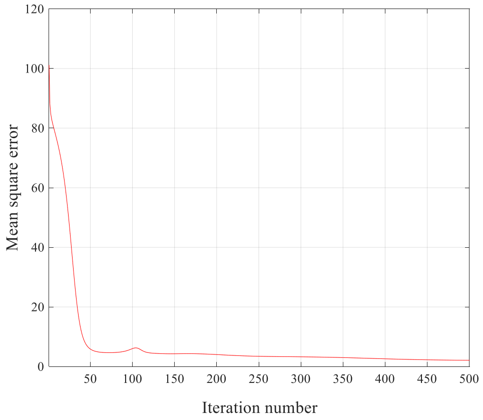

Figure 11: The training error curve of the M-GD neural network.

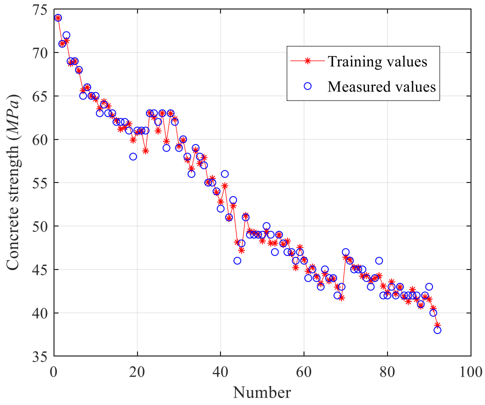

Figure 12: The fitting effect of training samples of the M-GD neural network.

is 0.9828. It shows that the predictive strength and measured strength have a very high fitting precision.

4.1.5 Comparison

This section presents the results of comparing the proposed models M-M and M-GD to other learning algorithms including the well-studied GD, GDM, GDA, EGD and L-M. Table 3 presents the comparison results obtained by various methods after setting appropriate parameters for each of them. It is easy to see that the MSEs of GD are 1.1136 and 3.6075, respectively, which perform poorest in the classical algorithms. However, when the memristor is combined, the established M-GD generally outperforms the classical algorithms and M-M. Especially for predictive results, the MSEs obtained by the M-GD are far less than those obtained by other algorithms.

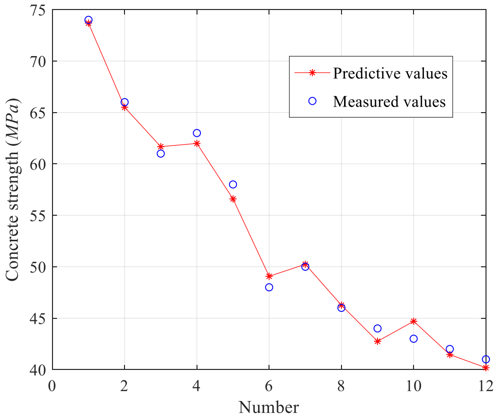

Figure 13: The prediction effect of prediction samples of the M-GD neural network.

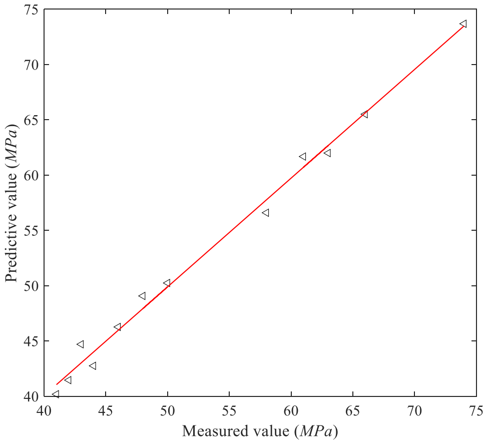

Figure 14: Scatter diagram of the measured and predictive values.

It can also be observed from Table 3 that the L-M has obvious advantages over all classical algorithms. In order to further explore the accuracy of the results from the proposed M-M and M-GD, the absolute errors and relative errors of 12 prediction samples are presented in Figs 15 and 16 and comparison is made with those from L-M. As shown in Figs. 15 and 16, the M-M undoubtedly enforces poorly compared with the L-M and M-GD. The curves of the relative and absolute errors obtained by the M-GD are almost below than those obtained by L-M, which means that the M-GD outperforms the L-M in prediction.

In order to clearly illustrate the superiority of the proposed model in prediction, a detailed comparison between the L-M and M-GD is presented in Table 4. In the table, the minimum absolute error, maximum absolute error, MAE, minimum relative error,

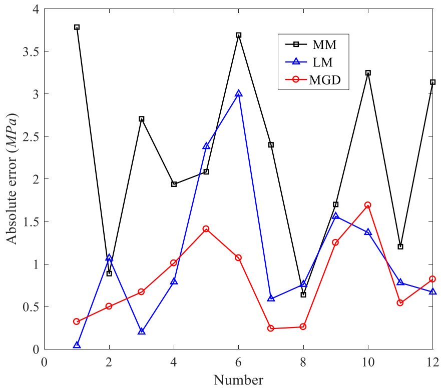

Figure 15: Comparison of absolute errors obtained by the L-M, M-M and M-GD.

maximum relative error, MRE, R and RC of the predictive values obtained by the L-M and M-GD are listed. Among all the error results achieved by L-M, only minimum absolute error and relative error are smaller than those obtained from the M-GD, which shows that the M-GD performs better than L-M in terms of accuracy. Moreover, the R and RC obtained from the L-M are smaller than those from the M-GD, which illustrates that the correlation achieved by M-GD is stronger.

Table 2: Performance measurement results of the M-M.

Number	Measured value (MPa)	Predictive value (MPa)	Absolute error (MPa)	Relative error (%)
1	74	69.08	4.92	6.65
2	66	65.11	0.89	1.35
3	61	63.71	2.71	4.44
4	63	61.06	1.94	3.08
5	58	55.92	2.08	3.59
6	48	51.69	3.69	7.69
7	50	52.40	2.40	4.80
8	46	46.64	0.64	1.39
9	44	42.30	1.70	3.86
10	43	46.25	3.25	7.56
11	42	40.80	1.20	2.86
12	41	37.86	3.14	7.66

Table 3: Performance measurement results of various algorithms.

Algorithm	GD	GDM	GDA	EGD	L-M	M-M	M-GD
MSE of training results	1.1136	0.9543	0.7058	0.3240	0.2997	2.8523	0.4900
MSE of predictive results	3.6075	3.1108	2.5546	2.2321	2.1786	6.2565	0.8743

Table 4: Performance measurement results of L-M and M-GD.

Algorithm	Minimum absolute error (MPa)	Maximum absolute error (MPa)	MAE (MPa)	Minimum relative error (%)	Maximum relative error (%)	MRE (%)	R	RC
L-M	0.04	2.38	1.10	0.05	6.24	2.22	0.9917	0.9600
M-GD	0.24	1.69	0.82	0.43	3.93	1.64	0.9962	0.9828

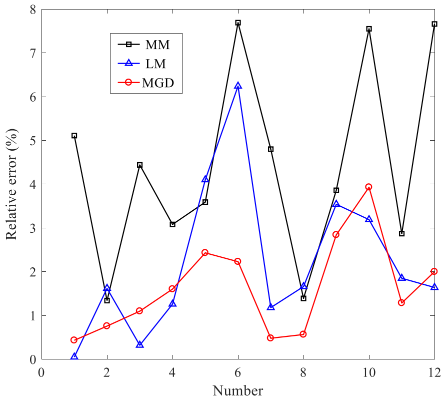

Figure 16: Comparison of relative errors obtained by the L-M, M-M and M-GD.

4.2 Example 2

4.2.1 The dataset

To further verify the effectiveness and practicality of the M-GD neural network, it was examined by another dataset containing a relatively small amount of data. The experimental dataset was mined from [39]. In this example, the eight factors, i.e., type of cement (TC), fine sand (FS, %), coarse sand (CS, %), fine aggregate (FA, %), coarse aggregate (CA, %), cement weight (CW, da Nm^{-3}), w/c ratio (w/c) and superplasticizer (SP', da Nm^{-3}), are employed as input variables. A total of 31 sets of the data attached in "Appendix B" were used. The 23 sets (Numbers 1-23) of the data were selected as the training samples, and the other 8 sets (Numbers 24-31) were employed as testing samples.

4.2.2 Results of M-GD neural network

The relevant tuning parameters are the same as those in the M-GD neural network employed in the previous example. The fitting effects of training samples and the prediction effect of testing samples are described in Figs. 17 and 18, respectively. It is noticeable that the general trend of the fitting curve is consistent with the measured strength. Fig. 18 demonstrates that the predictive strength obtained by the M-GD is in good agreement with the measured strength. In addition, the maximum and minimum relative errors, and mean relative error of training process are 6.65%, 0.023% and 2.68% respectively. The maximum and minimum relative errors, and mean relative error of testing process are 5.59%, 1.34% and 3.15%, respectively. It can be concluded that the M-GD neural network is able to predict the concrete strength accurately, and thus can be considered as a promising technique in this regard.

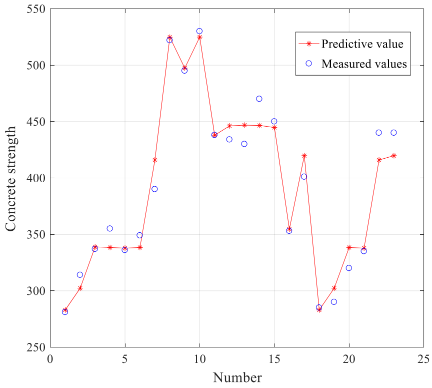

Figure 17: The fitting effect of training samples of the M-GD neural network.

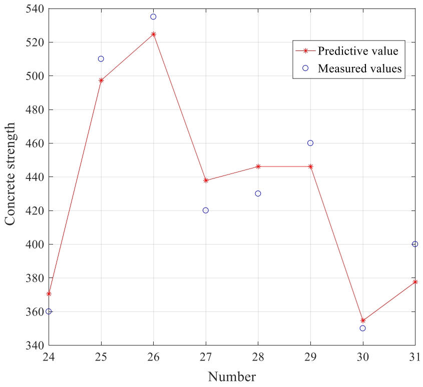

Figure 18: The prediction effect of testing samples of the M-GD neural network.

4.2.3 Extensive model validation

To avoid contingency, the training samples and testing samples were randomly assigned and grouped another six times. The details of the six groups are given as follows:

G1: 4 sets of testing samples are shown in "Appendix C1", the remaining 27 sets were training sample.

G2: 6 sets of testing samples are given in "Appendix C2", the remaining 25 sets were training sample.

G3: 8 sets of testing samples are shown in "Appendix C3", the remaining 23 sets were training sample.

G4: 10 sets of testing samples are given in "Appendix C4", the remaining 21 sets were training sample.

G5: 12 sets of testing samples are shown in "Appendix C5", the remaining 19 sets were training sample.

G6: 14 sets of testing samples are given in "Appendix C6", the remaining 17 sets were training sample.

These groups were trained and tested under the same parameters given in the Section 4.2.2, and the errors obtained by the M-GD for six data groups were presented in Table 5. Obviously the training and testing errors are quite small, which fully confirms the effectiveness and practicality of the proposed method.

Table 5: Errors obtained by the M-GD for six data groups.

Error	Training error (%)			Testing error (%)		
	Maximum relative error	Minimum relative error	MRE	Maximum relative error	Minimum relative error	MRE
G1	6.43	0.45	3.09	3.27	0.55	1.58
G2	7.84	0.04	2.82	6.70	1.35	3.17
G3	7.66	0.18	2.94	6.97	1.28	3.34
G4	6.76	0.33	3.12	6.52	0.32	3.41
G5	5.94	0.29	2.28	9.04	0.76	4.58
G6	5.37	0.03	1.77	9.31	0.58	4.78

5 Conclusions

Based on the classical BP neural network, the M-M neural network and the M-GD neural network are proposed. The presented models are utilized to forecast the compressive strength of HPC, and the fitting and prediction effect of the developed models are displayed. Moreover, the proposed methods are also evaluated from the comparisons with the classical learning algorithms including the GD, GDM, GDA, EGD and L-M. The main conclusions are obtained as follows:

1) The M-GD neural network is newly proposed. It generally outperforms the classical algorithms and M-M, and can be regarded as a promising forecasting tool in the field of civil engineering.

2) It is the first attempt to apply the M-M neural network in predicting the HPC strength. This network has a relatively high convergence speed, and the strength prediction errors of the M-M neural network can roughly meet the demands in construction engineering.

3) This study presents a way to construct the hardware circuit of the neural network with the memristor, and the established neural networks can deal with huge data prediction efficiently in principle.

4) Considering that it is the first attempt to employ memristor in civil engineering, it is desired that the memristor is extended to other fields such as engineering mechanics, environmental engineering, and aerospace.

It should be noted here that the proposed methods in this paper are applied in a specific case of study and show some advantages. The performance of the proposed method should be widely studied before the real applications to other fields.

Appendix A

Table 6: Experimental dataset of Example 1.

Number	f_c(MPa)	W/B(%)	W$(\mathrm{kg/m^3})$	s/a(%)	FAR(%)	AE$(\mathrm{kg/m^3})$	SP$(\mathrm{kg/m^3})$
1	74	30	160	48	10	0.069	8.00
2	74	30	160	48	20	0.069	8.00
3	71	30	160	46	0	0.069	8.00
4	72	30	160	45	10	0.069	8.00
5	69	30	160	44	20	0.069	8.00
6	69	30	160	42	0	0.069	8.00
7	68	30	160	42	10	0.069	8.00
8	65	30	160	41	20	0.069	8.00
9	66	30	170	47	0	0.074	8.50
10	66	30	170	46	20	0.074	8.50
11	65	30	170	44	0	0.074	8.50
12	65	30	170	43	10	0.074	8.50
13	63	30	170	42	20	0.074	8.50
14	64	30	170	41	0	0.074	8.50
15	63	30	170	40	10	0.074	8.50
16	63	30	170	39	20	0.074	8.50
17	61	30	180	45	0	0.078	7.50
18	62	30	180	44	10	0.078	7.50
19	62	30	180	44	20	0.078	7.50
20	62	30	180	42	0	0.078	7.50
21	61	30	180	41	10	0.078	7.50
22	58	30	180	40	20	0.078	7.50
23	61	30	180	38	0	0.078	7.50
24	61	30	180	38	10	0.078	7.50
25	61	30	180	37	20	0.078	7.50
26	63	35	160	51	0	0.059	5.71
27	63	35	160	50	10	0.059	5.71
28	62	35	160	50	20	0.059	5.71
29	63	35	160	48	0	0.059	5.71
30	63	35	160	47	10	0.059	5.71
31	59	35	160	47	20	0.059	5.71
32	63	35	160	45	0	0.059	5.71
33	62	35	160	44	10	0.059	5.71

Number	f_c(MPa)	W/B(%)	W$(\mathrm{kg/m^3})$	s/a(%)	FAR(%)	AE$(\mathrm{kg/m^3})$	SP$(\mathrm{kg/m^3})$
34	59	35	160	43	20	0.059	5.71
35	60	35	170	49	0	0.063	4.86
36	58	35	170	49	10	0.063	4.86
37	56	35	170	48	20	0.063	4.86
38	59	35	170	46	0	0.063	4.86
39	58	35	170	45	10	0.063	4.86
40	58	35	170	45	20	0.063	4.86
41	57	35	170	43	0	0.063	4.86
42	55	35	170	42	20	0.063	4.86
43	55	35	180	48	0	0.067	3.86
44	54	35	180	47	10	0.067	3.86
45	52	35	180	46	20	0.067	3.86
46	56	35	180	44	0	0.067	3.86
47	51	35	180	44	10	0.067	3.86
48	48	35	180	43	20	0.067	3.86
49	53	35	180	41	0	0.067	3.86
50	46	35	180	40	10	0.067	5.14
51	48	35	180	40	20	0.067	5.14
52	51	40	160	52	0	0.040	4.00
53	49	40	160	52	10	0.040	2.57
54	49	40	160	51	20	0.040	4.00
55	50	40	160	49	0	0.040	4.00
56	49	40	160	49	10	0.040	4.00
57	49	40	160	48	20	0.040	4.00
58	50	40	160	46	0	0.040	4.00
59	49	40	160	46	10	0.040	4.00
60	47	40	160	45	20	0.040	4.00
61	49	40	170	51	0	0.043	2.13
62	48	40	170	50	10	0.043	2.13
63	46	40	170	50	20	0.043	2.13
64	47	40	170	48	0	0.043	2.13
65	47	40	170	47	10	0.043	2.13
66	46	40	170	47	20	0.043	2.13
67	47	40	170	45	0	0.043	2.13
68	46	40	170	44	10	0.043	2.13
69	44	40	170	44	20	0.043	2.13
70	45	40	180	49	0	0.045	2
71	44	40	180	49	10	0.045	2.25
72	43	40	180	48	20	0.045	2.25
73	45	40	180	46	0	0.045	2.25
74	44	40	180	46	10	0.045	2.25
75	44	40	180	45	20	0.045	2.25
76	44	40	180	43	0	0.045	2.25
77	42	40	180	42	10	0.045	2.25
78	43	40	180	42	20	0.045	2.25
79	47	45	160	53	0	0.036	3.56
80	46	45	160	53	10	0.036	3.56
81	45	45	160	52	20	0.036	3.56

Number	f_c(MPa)	W/B(%)	W$(\mathrm{kg/m^3})$	s/a(%)	FAR(%)	AE$(\mathrm{kg/m^3})$	SP$(\mathrm{kg/m^3})$
82	45	45	160	50	0	0.036	3.56
83	43	45	160	50	10	0.036	3.56
84	45	45	160	49	20	0.036	3.56
85	44	45	160	47	0	0.036	3.56
86	43	45	160	47	10	0.036	3.56
87	44	45	160	46	20	0.036	3.56
88	46	45	170	52	0	0.038	1.89
89	42	45	170	51	10	0.038	1.89
90	42	45	170	51	20	0.038	1.89
91	43	45	170	49	0	0.038	1.89
92	42	45	170	48	10	0.038	1.89
93	42	45	170	48	20	0.038	1.89
94	43	45	170	46	0	0.038	1.89
95	42	45	170	45	10	0.038	1.89
96	42	45	170	45	20	0.038	1.89
97	42	45	180	51	0	0.040	2.00
98	42	45	180	50	10	0.040	2.00
99	41	45	180	50	20	0.040	2.00
100	42	45	180	47	0	0.040	2.00
101	41	45	180	47	20	0.040	2.00
102	43	45	180	44	0	0.040	2.00
103	40	45	180	44	10	0.040	2.00
104	38	45	180	43	20	0.040	2.00

f_c: compressive strength (MPa).

Appendix B

Table 7: Experimental dataset of Example 2.

Number	f'_c	TC	FS (%)	CS (%)	FA (%)	CA (%)	CW (da Nm^{-3})	w/c	SP' (da Nm^{-3})
1	281	325	19	27	28	26	280	0.55	0
2	314	325	15	28	30	27	310	0.55	1.1
3	337	325	15	28	30	27	350	0.54	1.1
4	355	325	15	28	30	27	350	0.5	1.1
5	336	325	15	28	30	27	350	0.52	1.1
6	349	325	15	28	30	27	350	0.53	1.1
7	390	325	15	28	30	27	350	0.5	4.1
8	522	425	15	28	30	27	370	0.48	4.4
9	495	425	15	28	30	27	370	0.5	4.4
10	530	425	15	28	30	27	370	0.46	4.4
11	438	425	15	28	30	27	370	0.52	4.4
12	434	425	15	28	30	27	370	0.51	1.3
13	430	425	15	28	30	27	370	0.48	1.3
14	470	425	15	28	30	27	370	0.49	1.3
15	450	425	15	28	30	27	370	0.52	1.3
16	353	325	15	28	30	27	370	0.54	1.3
17	401	325	15	28	30	27	370	0.51	4.4
18	285	325	19	27	28	26	280	0.55	0

Number	f_c'	TC	FS (%)	CS (%)	FA (%)	CA (%)	CW (da Nm^{-3})	w/c	SP′ (da Nm^{-3})
19	290	325	15	28	30	27	310	0.55	1.1
20	320	325	15	28	30	27	350	0.5	1.1
21	335	325	15	28	30	27	350	0.52	1.1
22	440	325	15	28	30	27	350	0.5	4.1
23	440	325	15	28	30	27	370	0.51	4.4
24	360	325	15	28	30	27	350	0.54	4.1
25	510	425	15	28	30	27	370	0.5	4.4
26	535	425	15	28	30	27	370	0.48	4.4
27	420	425	15	28	30	27	370	0.52	4.4
28	430	425	15	28	30	27	370	0.51	1.3
29	460	425	15	28	30	27	370	0.51	1.3
30	350	325	15	28	30	27	370	0.54	1.3
31	400	325	15	28	30	27	370	0.52	4.4

f_c': concrete strength.

Appendix C1

Table 8: Testing samples in Group 1.

Number	f_c'	TC	FS (%)	CS (%)	FA (%)	CA (%)	CW (da Nm^{-3})	w/c	SP′ (da Nm^{-3})
5	336	325	15	28	30	27	350	0.52	1.1
11	438	425	15	28	30	27	370	0.52	4.4
21	335	325	15	28	30	27	350	0.52	1.1
30	350	325	15	28	30	27	370	0.54	1.3

f_c': concrete strength.

Appendix C2

Table 9: Testing samples in Group 2.

Number	f_c'	TC	FS (%)	CS (%)	FA (%)	CA (%)	CW (da Nm^{-3})	w/c	SP′ (da Nm^{-3})
5	336	325	15	28	30	27	350	0.52	1.1
6	349	325	15	28	30	27	350	0.53	1.1
11	438	425	15	28	30	27	370	0.52	4.4
15	450	425	15	28	30	27	370	0.52	1.3
21	335	325	15	28	30	27	350	0.52	1.1
30	350	325	15	28	30	27	370	0.54	1.3

f_c': concrete strength.

Appendix C3

Table 10: Testing samples in Group 3.

Number	f_c'	TC	FS (%)	CS (%)	FA (%)	CA (%)	CW (da Nm^{-3})	w/c	SP' (da Nm^{-3})
5	336	325	15	28	30	27	350	0.52	1.1
6	349	325	15	28	30	27	350	0.53	1.1
11	438	425	15	28	30	27	370	0.52	4.4
15	450	425	15	28	30	27	370	0.52	1.3
21	335	325	15	28	30	27	350	0.52	1.1
23	440	325	15	28	30	27	370	0.51	4.4
27	420	425	15	28	30	27	370	0.52	4.4
30	350	325	15	28	30	27	370	0.54	1.3

f_c': concrete strength.

Appendix C4

Table 11: Testing samples in Group 4.

Number	f_c'	TC	FS (%)	CS (%)	FA (%)	CA (%)	CW (da Nm^{-3})	w/c	SP' (da Nm^{-3})
1	281	325	19	27	28	26	280	0.55	0
3	337	325	15	28	30	27	350	0.54	1.1
7	390	325	15	28	30	27	350	0.5	4.1
8	522	425	15	28	30	27	370	0.48	4.4
11	438	425	15	28	30	27	370	0.52	4.4
13	430	425	15	28	30	27	370	0.48	1.3
16	353	325	15	28	30	27	370	0.54	1.3
21	335	325	15	28	30	27	350	0.52	1.1
24	360	325	15	28	30	27	350	0.54	4.1
30	350	325	15	28	30	27	370	0.54	1.3

f_c': concrete strength.

Appendix C5

Table 12: Testing samples in Group 5.

Number	f_c'	TC	FS (%)	CS (%)	FA (%)	CA (%)	CW (da Nm^{-3})	w/c	SP' (da Nm^{-3})
2	314	325	15	28	30	27	310	0.55	1.1
5	336	325	15	28	30	27	350	0.52	1.1
7	390	325	15	28	30	27	350	0.5	4.1
8	522	425	15	28	30	27	370	0.48	4.4
9	495	425	15	28	30	27	370	0.5	4.4
11	438	425	15	28	30	27	370	0.52	4.4
14	470	425	15	28	30	27	370	0.49	1.3
16	353	325	15	28	30	27	370	0.54	1.3
22	440	325	15	28	30	27	350	0.5	4.1

Number	f'_c	TC	FS (%)	CS (%)	FA (%)	CA (%)	CW (da Nm^{-3})	w/c	SP' (da Nm^{-3})
26	535	425	15	28	30	27	370	0.48	4.4
29	460	425	15	28	30	27	370	0.51	1.3
30	350	325	15	28	30	27	370	0.54	1.3

f'_c: concrete strength.

Appendix C6

Table 13: Testing samples in Group 6.

Number	f'_c	TC	FS (%)	CS (%)	FA (%)	CA (%)	CW (da Nm^{-3})	w/c	SP' (da Nm^{-3})
1	281	325	19	27	28	26	280	0.55	0
2	314	325	15	28	30	27	310	0.55	1.1
6	349	325	15	28	30	27	350	0.53	1.1
7	390	325	15	28	30	27	350	0.5	4.1
9	495	425	15	28	30	27	370	0.5	4.4
11	438	425	15	28	30	27	370	0.52	4.4
12	434	425	15	28	30	27	370	0.51	1.3
13	430	425	15	28	30	27	370	0.48	1.3
16	353	325	15	28	30	27	370	0.54	1.3
24	360	325	15	28	30	27	350	0.54	4.1
27	420	425	15	28	30	27	370	0.52	4.4
29	460	425	15	28	30	27	370	0.51	1.3
30	350	325	15	28	30	27	370	0.54	1.3
31	400	325	15	28	30	27	370	0.52	4.4

f'_c: concrete strength.

Acknowledgements

The work described in this paper was supported by the National Key Research and Development Program of China (No. 2021YFB2600700), the National Natural Science Foundation of China (No. 52171272), the Significant Science and Technology Project of the Ministry of Water Resources of China (No. SKS-2022112), the Nanjing Water Science and Technology Project (No. 202203), and the 333 Talent Project of Training and Support in Key Industries.

References

[1] D. Wang, C. Shi, Z. Wu, J. Xiao, Z. Huang, and Z. Fang, *A review on ultra high performance concrete: Part II. Hydration, microstructure and properties,* Constr. Build. Mater., 96 (2015), pp. 368–377.

[2] A. Neville, and P.-C. Aitcin, *High performance concrete-An overview,* Mater. Struct., 31 (1998), pp. 111–117.

[3] K. Ozawa, K. Maekawa, and H. Okamura, *Development of high performance concrete,* J. Fac. Eng., 41 (1992), pp. 149–157.

[4] R. Parichatprecha, and P. Nimityongskul, *Analysis of durability of high performance concrete using artificial neural networks,* Constr. Build. Mater., 23 (2009), pp. 910–917.

[5] Z. J. He, and Y. P. Song, *Triaxial strength and failure criterion of plain high-strength and high-performance concrete before and after high temperatures,* Cem. Concr. Res., 40 (2010), pp. 171–178.

[6] J. S. Chou, and C. F. Tsai, *Concrete compressive strength analysis using a combined classification and regression technique,* Autom. Constr., 24 (2012), pp. 52–60.

[7] S. Popovics, *History of a mathematical model for strength development of Portland cement concrete,* ACI Mater. J., 95 (1998), pp. 593–600.

[8] M. Nataraja, M. Jayaram, and C. Ravikumar, *Prediction of early strength of concrete: a fuzzy inference system model,* Int. J. Phys. Sci., 1 (2006), pp. 047–056.

[9] F. A. Oluokun, E. G. Burdette, and J. H. Deatheridge, *Early-age concrete strength prediction by maturity–another look,* ACI Mater. J., 87 (1990), pp. 565–572.

[10] C. Guo, *Maturity of concrete: method for predicting early-stage strength,* ACI Mater. J., 86 (1989), pp. 341–353.

[11] R. B. Ardalan, A. Joshaghani, and R. D. Hooton, *Workability retention and compressive strength of self-compacting concrete incorporating pumice powder and silica fume,* Constr. Build. Mater., 134 (2017), pp. 116–122.

[12] J. Dong, Q. H. Qin, and Y. Xiao, *Nelder–mead optimization of elastic metamaterials via machine-learning-aided surrogate modeling,* Int. J. Appl. Mech., (2020), 2050011.

[13] S. Czarnecki, Ł. Sadowski, and J. Hoła, *Artificial neural networks for non-destructive identification of the interlayer bonding between repair overlay and concrete substrate,* Adv. Eng. Software, 141 (2020), 102769.

[14] K. Mathiyalagan, R. Sakthivel, and S. M. Anthoni, *New robust passivity criteria for stochastic fuzzy BAM neural networks with time-varying delays,* Commun. Nonlinear Sci. Numer. Simul., 17 (2012), pp. 1392–1407.

[15] M. Pakdaman, A. Ahmadian, S. Effati, S. Salahshour, and D. Baleanu, *Solving differential equations of fractional order using an optimization technique based on training artificial neural network,* Appl. Math. Comput., 293 (2017), pp. 81–95.

[16] M. A. Kewalramani, and R. Gupta, *Concrete compressive strength prediction using ultrasonic pulse velocity through artificial neural networks,* Autom. Constr., 15 (2006), pp. 374–379.

[17] J. Kasperkiewicz, J. Racz, and A. Dubrawski, *HPC strength prediction using artificial neural network,* J. Comput. Civil Eng., 9 (1995), pp. 279–284.

[18] M. Nehdi, Y. Djebbar, and A. Khan, *Neural network model for preformed-foam cellular concrete,* ACI Mater. J., 98 (2001), pp. 402–409.

[19] J. I. Kim, D. K. Kim, M. Q. Feng, and F. Yazdani, *Application of neural networks for estimation of concrete strength,* J. Mater. Civ. Eng., 16 (2004), pp. 257–264.

[20] J. S. Chou, and A. D. Pham, *Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength,* Constr. Build. Mater., 49 (2013), pp. 554–563.

[21] H. I. Erdal, O. Karakurt, and E. Namli, *High performance concrete compressive strength forecasting using ensemble models based on discrete wavelet transform,* Eng. Appl. Artif. Intell., 26 (2013), pp. 1246–1254.

[22] J. Sobhani, M. Najimi, A. R. Pourkhorshidi, and T. Parhizkar, *Prediction of the compressive strength of no-slump concrete: A comparative study of regression, neural network and ANFIS models,* Constr. Build. Mater., 24 (2010), pp. 709–718.

[23] L. Chua, *Memristor-the missing circuit element,* IEEE Trans. Circuit Theory, 18 (1971), pp. 507–519.

[24] J. Kengne, Z. Njitacke Tabekoueng, V. Kamdoum Tamba, and A. Nguomkam Negou, *Periodicity, chaos, and multiple attractors in a memristor-based Shinriki's circuit,* Chaos, 25 (2015), 103126.

[25] Y. F. Pu, X. Yuan, and B. Yu, *Analog circuit implementation of fractional-order memristor: arbitrary-order lattice scaling fracmemristor,* IEEE T Circuits-I, 65 (2018), pp. 2903–2916.

[26] M. Teimoory, A. Amirsoleimani, J. Shamsi, A. Ahmadi, S. Alirezaee, and M. Ahmadi, *Optimized implementation of memristor-based full adder by material implication logic,* in: 2014 21st IEEE International Conference on Electronics, Circuits and Systems (ICECS), IEEE, pp. 562–565.

[27] A. Afifi, A. Ayatollahi, and F. Raissi, *Implementation of biologically plausible spiking neural network models on the memristor crossbar-based CMOS/nano circuits,* in: 2009 European Conference on Circuit Theory and Design, IEEE, pp. 563–566.

[28] G. S. Snider, *Spike-timing-dependent learning in memristive nanodevices,* in: 2008 IEEE international symposium on nanoscale architectures, IEEE, pp. 85–92.

[29] D. B. Strukov, G. S. Snider, D. R. Stewart, and R.S. Williams, *The missing memristor found,* Nature, 453 (2008), pp. 80–83.

[30] L. Wang, E. Drakakis, and S. Duan,et al., *Memristor model and its application for chaos generation,* Int. J. Bifurcat. Chaos, 22 (2012), 1250205.

[31] S. I. Amari, *Backpropagation and stochastic gradient descent method,* Neurocomputing, 5 (1993), pp. 185–196.

[32] R. Yu, X. An, B. Jin, J. Shi, O. A. Move, and Y. Liu, *Particle classification optimization-based BP network for telecommunication customer churn prediction,* Neural Comput. Appl., 29 (2018), pp. 707–720.

[33] H. Liu, H. Q. Tian, X. F. Liang, and Y. F. Li, *Wind speed forecasting approach using secondary decomposition algorithm and Elman neural networks,* Appl. Energy, 157 (2015), pp. 183–194.

[34] D. Ding, *Visual neural network model for welding deviation prediction based on weld pool centroid,* Int. J. Pattern Recognit Artif Intell., 32 (2018), 1859014.

[35] C. J. Jane, *An improved grey back propagation neural network with Levenberg-Marquardt algorithm and genetic algorithm,* J. Grey Syst., 20 (2017), pp. 71–77.

[36] A. Baykasoğlu, A. Öztaş, and E. Özbay, *Prediction and multi-objective optimization of high-strength concrete parameters via soft computing approaches,* Expert Syst. Appl., 36 (2009), pp. 6145–6155.

[37] C. H. Lim, Y. S. Yoon, and J. H. Kim, *Genetic algorithm in mix proportioning of high-performance concrete,* Cem. Concr. Res., 34 (2004), pp. 409–420.

[38] X. Chen, J. Fei, and X. Yuan, *High-performance concrete strength prediction model based on the radial basis function neural network of human cerebral cortex,* NeuroQuantology, 16 (2018).

[39] S. Lai, and M. Serra, *Concrete strength prediction by means of neural network,* Constr. Build. Mater., 11 (1997), pp. 93–98.