



A new reliability analysis approach with multiple correlation neural networks method

Shangjie Li¹ · Xianzhen Huang^{1,2} · Xingang Wang¹ · Yuxiong Li¹

Accepted: 18 November 2022 / Published online: 12 December 2022

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

Structural reliability is defined as the safety probability of the structure under the influence of uncertain factors over a given period of time. The direct integration method from the definition of reliability is closer to the true value. However, the complex integration domain and the calculation of multiple integrals bring difficulties for reliability calculation. This study proposes a multiple correlation neural networks (MCNN) method to solve these problems. In the first step, a set of artificial neural networks (ANNs) are built to approximate the safety domain of structural reliability. Then, the trained ANNs are connected with improved dual neural networks to form the MCNN method. Using the correlation of ANNs in MCNN, the solution of multiple definite integrals with the implicit integral domain is obtained. The proposed method has significant accuracy and efficiency compared with other existing methods and increases the computational stability of ANN-based reliability calculation methods. The performances of the newly proposed method in calculation accuracy and efficiency are analyzed for several structural reliability problems.

Keywords Structural reliability · Direct integral approach · Multiple definite integrals · Multiple correlation neural networks

1 Introduction

It is well known that uncertainties in engineering structures may cause severe accidents and should be reasonably analyzed and controlled. Reliability calculation takes these uncertainties into account when evaluating structural system safety. From the middle of the last century, many methods have been developed for structural reliability calculation, mainly can be summarized into three categories. (1) Analytical methods. The first-order reliability (Liu and Kiureghian 1991) and second-order reliability methods (Tvedt 1990; Zhang and Du 2010; Liu and Peng 2012; Zhang et al. 2015; Zhang and He 2018) are

considered to be the most popular in the past few decades. These methods require an iterative process. With the non-normal variables changed into standard variables by non-linear transformation, the most probable point is located by the sensitivity of the performance function. Although the method can give acceptable accuracy in solving some realistic problems, the most probable point is difficult to locate as the nonlinearity of the performance function increases. (2) Simulation-based methods. The direct Monte Carlo simulation (MCS) methods (Rubinstein and Kroese 2007) and other improved MCS methods (Nie and Ellingwood 2000; Niederreiter and Spanier 2000; Au and Beck 2001; Papadopoulos et al. 2012; Shayanfar et al. 2018) are the most widely used simulation methods, and the calculation results can be regarded as the exact value and used to verify the accuracy of other methods. However, the computation of the required large number of samples is very time-consuming. (3) Direct integration methods. Starting from the definition of reliability, the direct integration methods (Genz and Malik 1980; Zhang and Cui 1997) calculate the integral formula of reliability. Even though some numerical integration methods (Place and

✉ Xianzhen Huang
xzhhuang@mail.neu.edu.cn

¹ School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, People's Republic of China

² Key Laboratory of Vibration and Control of Aero Propulsion Systems Ministry of Education of China, Northeastern University, Shenyang 110819, People's Republic of China

Stach 1999; Allahviranloo 2005; Simos 2009) have been developed, there are still difficulties in solving the complex multiple integrals.

In recent years, various machine intelligence methods, particularly artificial neural network (ANN) methods, have been widely used in reliability calculation (Jin and Liu 2008; Chojaczyk et al. 2015; Dey et al. 2020; Li et al. 2022). ANN is an adaptive nonlinear dynamic system that can approximate highly nonlinear function accurately (Liao et al. 2015; Wang and Li 2017; Aljarah et al. 2018; Asteris et al. 2019). ANN is composed of a large number of simple neurons which are interconnected. Training an ANN to approximate the limit state function (LSF) and performing MCS in terms of the trained ANN can reduce the number of finite element calculations in reliability calculation (Papadrakakis et al. 1996; Papadrakakis and Lagaros 2002; Cardoso et al. 2008; Chojaczyk et al. 2015; Yoon et al. 2020). However, these methods are still time-consuming for complex LSF. Alternatively, ANN is combined with a gradient-based method like FORM or SORM for reliability calculation (Chojaczyk et al. 2015; Goh and Kulhawy 2003; Yan and Lin 2016), which can reduce the convergence problem. However, the inherent limitations of gradient-based methods limit the computational accuracy. Several studies (Gomes and Awruch 2004; Bucher and Most 2008; Dai et al. 2015; Nezhad et al. 2019) show that the ANN-based response surface methods are efficient in some specific reliability calculation problems. The key to the direct integration methods for reliability calculation is the solution of multiple definite integrals. Zeng et al. (2006) proposed an ANN-based integration solution method by training weights with cosine basis function to fit the integrand, resulting in satisfactory computational accuracy. However, the variable must be within a fixed interval, which seriously affects the use in science and engineering practice. Lloyd et al. (2020) proposed a novel numerical integration technique that uses a shallow neural network design to approximate the integrand function within a bounded set. Experiments show that the method is feasible and works best on predictable integral functions, but is less effective on singular and non-smooth functions.

Li et al. (2018) proposed a dual neural networks (DNN) method to calculate the multiple definite integrals in reliability calculation. The DNN consists of neural network A (NETA) and neural network B (NETB), which is constructed by training NETA to approximate the integrand. The NETB corresponding to the primitive function is directly obtained with the functional relations between the two neural networks. However, this method is effective only when the integral concerns a single definite integral or the integral domain of multiple definite integrals is a hypercube. Then, Li et al. (2019) solved multiple definite

integrals for the arbitrary integral domain by repeatedly applying DNN. This method has the characteristics such as numerical stability and does not need to know the integrand. The method has been successfully applied in the reliability calculation of several engineering problems (Li et al. 2018; Li et al. 2019; Du and Li 2019). These advantages have made DNN quickly attracts extensive attention.

This paper proposes a multiple correlation neural network (MCNN) method for reliability calculation based on DNN. The integral domain is difficult to determine for most reliability calculation problems because of the implicit performance function. In MCNN, a set of ANN models are trained to approximate the reliability integral domain. Through a novel transformation process, the interval domain of each variable is determined by the trained ANN and then combined with the DNN to form the MCNN. In this paper, the integrand is the joint probability density function of all random variables. The integrand is a continuous, differentiable, real-valued function, and the training accuracy of the NETA can be fully guaranteed. The DNN with a new activation function pair is used for more efficient reliability calculation. By iteratively computing the definite integral of each variable, the solution of the multiple definite integrals corresponding to the reliability is obtained. Several representative examples show that the proposed method is an efficient and accurate reliability calculation method and can be used as a paradigm. This paper is organized as follows: In Sect. 2, the basic integral form of structural reliability calculation is briefly presented. Section 3 presents the fundamental theories of DNN for calculating multiple definite integrals. Section 4 details the newly proposed method for reliability calculation. Illustrative numerical examples and results are discussed in Sect. 5. In Sect. 6, the conclusion of this work is outlined.

2 The integral form of structural reliability calculation

Structural reliability analysis frequently involves the calculation of a component probability of safety:

$$\mathbf{R} = \mathbf{R}\{G(\mathbf{X}) > 0\} = \int \cdots \int_{G(\mathbf{X}) > 0} f(x) d\mathbf{X} \quad (1)$$

where $\mathbf{X} = \{x_1, x_2, \dots, x_N\}^T$ represents N dimensional basic input random variables, $G(\mathbf{X})$ is a performance function and $G(\mathbf{X}) > 0$ represents the safety event. $f(x)$ is the joint probability density function of variables \mathbf{X} , typically representing the loads, material properties and geometry.

It is challenging to directly solve the integrals in Eq. (1) due to the implicit performance function. An indicator

function $I(G(X))$ is introduced. If $G(X) > 0$, let $I(G(X)) = 1$, otherwise, $I(G(X)) = 0$. Then Eq. (1) becomes:

$$R = R\{G(X) > 0\} = \int_{-\infty}^{+\infty} \dots \int_{-\infty}^{+\infty} I(G(X)) \cdot f(x) dX \tag{2}$$

Let $y(X) = I(G(X))f(x)$, the integrand is a piecewise function. The explicit expression of integrand $y(X)$ is not available, because the performance function $G(X)$ is always expressed implicitly. The multiple definite integrals in Eq. (2) are also difficult to solve with the traditional numerical methods.

3 Review of DNN and direct integration method

3.1 Introduction of DNN and direct integration method

The DNN (Li et al. 2018) and direct integration method is proposed to calculate the multiple definite integrals with implicit integrand, which provides a new thought for reliability calculation with ANN method. The main thought is training a three-layered NETA to approximate the integrand in integral Eq. (2), and another three-layered NETB is analytically derived to approximate the primitive function. The derived primitive function is taken as the integrand in the next step, the solution of multiple definite integrals is obtained by repeating this step.

The DNN consists of two interrelated forward-type BP neural networks A and B, the NETA is used to approximate the integrand in Eq. (2), and the NETB corresponds to the primitive function. The structure of the NETA is shown in Fig. 1:

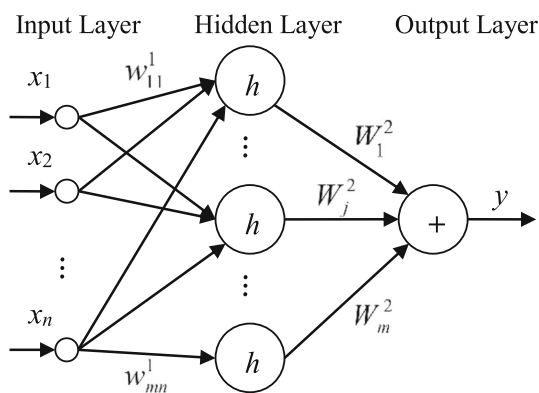


Fig. 1 The structure of NETA

The functional relation between the outputs and inputs of NETA can be expressed as:

$$y = \sum_{j=1}^m h \left(\sum_{i=1}^n w_{ji}^1 x_i + \vartheta_j \right) W_j^2 \tag{3}$$

The structure of the NETB is shown in Fig. 2:

The functional relation between the outputs and the inputs of NETB can be written as:

$$Y = \sum_{j=1}^m g \left(\sum_{i=1}^n w_{ji}^1 x_i + \vartheta_j \right) w_j^2 + b \tag{4}$$

The relations of NETA and NETB are, respectively, integrand and primitive function of the variable x_i , their function relations of weight coefficients and activation functions are as below:

$$W_j^2 = w_j^2 \cdot w_{ji}^1, \quad \frac{\partial g(x)}{\partial x} = h(x) \tag{5}$$

where $h(x)$ and $g(x)$ are called a set of activation function pairs.

According to the fundamental theories of DNN, the solution of the reliability integral in Eq. (2) can be achieved by using DNN repeatedly (Li et al. 2019). The flowchart is shown in Fig. 3.

3.2 Remarks on the DNN and direct integration method

The three-layer ANN structure is shown in Fig. 4. DNN has a broader range of applications than traditional numerical integration methods. The performance function $G(X)$ is always an implicit expression, only some sample data points that can reflect the function relation between the input and output of the integrand in Eq. (2) are known. The traditional numerical integration methods are challenging to calculate under this situation, and the DNN can be directly used to calculate reliability integral; when the

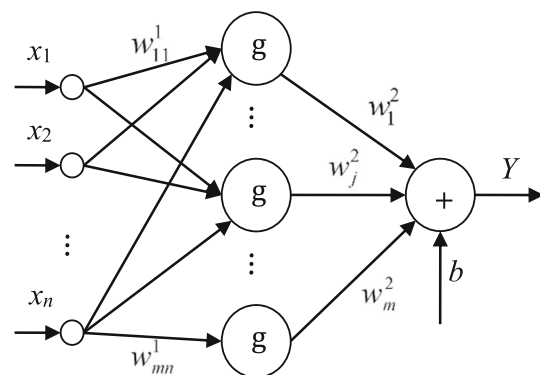


Fig. 2 The structure of NETB

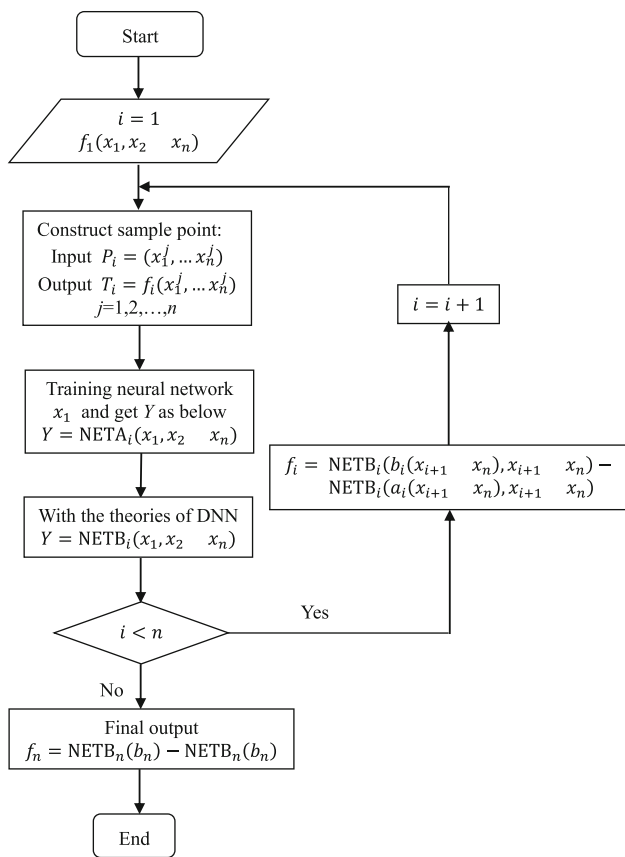


Fig. 3 Flowchart of DNN method for solving multiple definite integrals

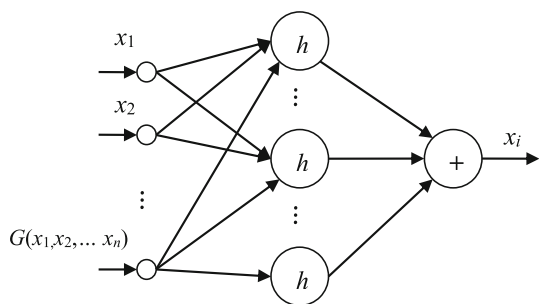


Fig. 4 Structure of the neural network

interval domain of a variable is a function concerning other variables, the DNN shows excellent performance; the accuracy of reliability calculation can be controlled by increasing the number of hidden neurons and training steps in the DNN. However, the integrand in Eq. (2) is a piecewise function, which may influence the approximation accuracy of NETA. The integrand in Eq. (2) is different for different reliability calculation problems, so all the ANNs in DNN need to be rebuilt for different reliability calculation problems. Therefore, the reliability calculation of DNN deserves further study.

This paper firstly constructs a set of ANNs to approximate the reliability integral domain. The trained ANN can represent the interval domain of each variable without the need to know the expression of the performance function. Unlike the DNN and direct integration method for reliability calculation, the NETA is used to approximate the joint probability density function of $f(x)$ in Eq. (1). $f(x)$ is a continuous, differentiable and real-valued function, and the model accuracy of the trained NETA can be guaranteed. For different reliability problems with the same distribution of random variables, the previously trained ANNs can be used directly without needing to reset network parameters and be trained again. Therefore, the newly proposed MCNN method is significant for computational accuracy, stability and efficiency of reliability.

4 The proposed MCNN method

4.1 Reliability integral domain estimation with ANN

The expression of the performance function $G(x_1, x_2, \dots, x_n)$ is always implicit, and only some sample points are available, so the reliability integral domain is difficult to estimate and perform in function form. This section constructs a set of ANNs to achieve high precision approximation of the integral domain.

For the first integral $\int_{x_i} \{x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_n, G(x_1, x_2, \dots, x_n)\}$ are selected as the inputs and x_i as the output. The three-layered ANN structure is as follow:

The functional relation between the outputs and inputs can be written as:

$$x_i = ch(k_1x_1 + \dots + k_{i-1}x_{i-1} + k_{i+1}x_{i+1} \dots + k_{n-1}x_n + k_nG(X) + b_2) \tag{6}$$

where k is the weight vector from the input layer to the hidden layer, b_2 represents the bias in the hidden layer, c is the weight vector from the hidden layer to the output layer, and function sigmoid() is selected as the activation function $h()$.

After training the ANN, the weight and bias vectors in Eq. (6) are obtained. Set the performance function in the limit state $G(X) = 0$, the relationship between the input random variable x_i and other variables $\{x_1, x_2, x_{i-1}, x_{i+1}, x_n\}$ in the limit state can be expressed as:

$$x_i = ch(k_1x_1 + \dots + k_{i-1}x_{i-1} + k_{i+1}x_{i+1} \dots + k_{n-1}x_n + b_2) \tag{7}$$

Then the integral interval of the first variable x_i is successfully determined by other $n - 1$ random variables. For

the second integral of another random variable, also repeat the same step as the first integral. Set the performance function $G(\mathbf{X}) = 0$ and $x_i = 0$, the integral interval of the second variable expressed by other $n - 2$ random variables is obtained. Finally, the integral intervals of all variables are derived.

4.2 The MCNN method for structural reliability calculation

This section elaborates on the newly proposed MCNN for structural reliability calculation. To illustrate the MCNN method more clearly, the reliability calculation with two variables is considered. The definition of reliability can be expressed as:

$$R = \mathbf{R}\{G(x_1, x_2) > 0\} = \int \int_{G(x_1, x_2) > 0} f(x_1, x_2) dx_1 dx_2 \tag{8}$$

The detailed calculation steps of the MCNN method are as follows, and Fig. 5 illustrates the flowchart of the calculation process.

Step 1 Set the network parameters. The initial network weights and bias are random values in the interval $[0, 1]$,

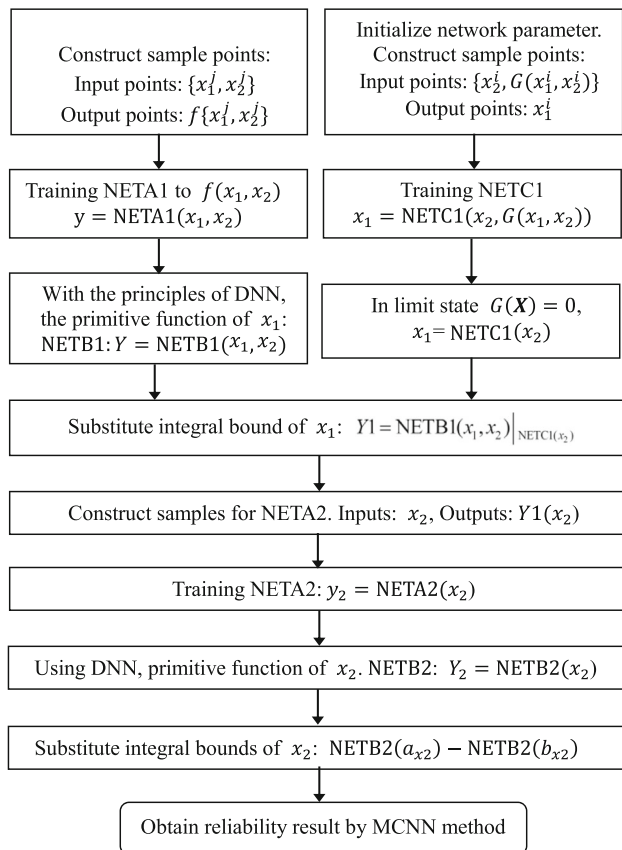


Fig. 5 The flowchart of the MCNN method

and the Levenberg-Marquardt algorithm (Hagan and Menhaj 1994) is used to update the network weights and bias of all ANNs.

Step 2 Construct the input points $\{x_1^i, x_2^i\}$ and output points $f(x_1^i, x_2^i)$, and train a three-layered neural network A1(NETA1) to approximate the joint probability density function $f(x_1, x_2)$. The function sigmoid() is selected as the activation function $h(x)$ in the hidden layer. The functional relation between the outputs and inputs of NETA1 can be expressed as:

$$y = Wh(w_1x_1 + w_2x_2 + \vartheta) \tag{9}$$

Step 3 According to the functional relations of DNN in Eq. (5), the three-layered network B1(NETB1) corresponding to the primitive function of integrand $f(x_1, x_2)$ in x_1 is analytically derived. The functional relation between the outputs and inputs of NETB1 can be expressed as:

$$Y = W_1g(w_1x_1 + w_2x_2 + \vartheta) \tag{10}$$

where $W_1 = W/w_1$ and $h(x) = \partial g(x)/\partial x$, the weight vectors w_1, w_2 and bias ϑ are same with the values in Eq. (9), the activation function $g(x)$ is function softplus().

Step 4 Construct a three-layered neural network C1 (NETC1) to estimate reliability integral domain, set $\{x_2, G(x_1, x_2)\}$ as the inputs and x_1 as the output. The expression of the trained NCEC1 is:

$$x_1 = ch(k_1x_2 + k_2G(x_1, x_2) + b_1) \tag{11}$$

Set the performance function in limit state $G(\mathbf{X}) = 0$, the functional relation between variables x_1 and x_2 can be expressed as:

$$x_1 = ch(k_1x_2 + b_1) \tag{12}$$

Step 5 With the maximum and minimum values of the variable ranges, the NETC1 can accurately determine the variable interval. The computational result of the first definite integral in x_1 has been completed, the final result can be expressed as:

$$Y1 = W_1g(w_1x_1 + w_2x_2 + \vartheta)|_{NETC1} \tag{13}$$

Step 6 Construct a three-layered neural network A2 (NETA2) to approximate the function in Eq. (13). The function in Eq. (13) is just about variable x_2 , the NETA2 is trained with x_2 as the input and $Y1(x_2)$ as output. The functional relation of the input and output variables is expressed as follow:

$$y2 = W_2h(w_3x_2 + b_2) \tag{14}$$

Step 7 With the functional relations of $W_3 = W_2/w_3$ and $h(x) = \partial g(x)/\partial x$, the neural network B2(NETB2) corresponding to the primitive function is directly obtained. The functional relation of input and output variables can be expressed as:

$$Y2 = W_3g(w_3x_2+b_2) \tag{15}$$

where the weight and bias vectors are the same as those of NETA2.

Step 8 Substitute the upper bound a_{x_2} and lower bound b_{x_2} of variable x_2 into NETB2, the calculation of the second definite integral in x_2 is completed. The upper and lower bounds are the maximum and minimum values of variable x_2 . The final result is as follow:

$$Y3 = W_3g(w_3a_{x_2}+b_2) - W_3g(w_3b_{x_2}+b_2) \tag{16}$$

Step 9 Get the result of the reliability calculation.

5 Numerical example

5.1 Example 1

In this example, the usability reliability of cantilever beam with rectangular section under uniform distributed load is estimated. Usability failure will occur when maximum deflection at the free exceeds $L/325$, where L is the length of beam. The young modulus E and length L are set as $L = 600\text{mm}$ and $E = 2.6104\text{MPa}$, the performance function is (Rajashekhar and Ellingwood 1993):

$$G(X) = 0.01846154 - 74.76923 \frac{x_1}{x_2^3}$$

The intensity x_1 of distributed uniform load and the height x_2 of the beam follow mutually independent normal distributions, their statistics are listed in Table 1.

The reliability is calculated by solving the following multiple integrals.

$$R = R\{G(X) > 0\} = \int \int_{G(X) > 0} \frac{1}{2\pi \times 200 \times 37.5} \cdot e^{-\frac{1}{2} \left[\left(\frac{x_1 - 1000}{200} \right)^2 + \left(\frac{x_2 - 250}{37.5} \right)^2 \right]} dx_1 dx_2 \tag{17}$$

The variables of x_1 and x_2 are transformed into two standard normal distributed variables to generate the training samples. The ranges of $[\mu_{y_1} \pm 4\delta_{y_1}]$ and $[\mu_{y_2} \pm 4\delta_{y_2}]$ are, respectively, divided into 40 parts, two variables cross each other forming the input sample points of NETA1. The joint probability density function of y_1 and y_2 is taken as the output. After considering the accuracy and efficiency, the number of neurons in the hidden layer and

Table 2 The network parameters of all ANNs

Method	NETA1	NETB1	NETA2	NETB2	NETC1
Hidden neuron	100	100	30	30	30
Training step	500	0	1000	0	1000

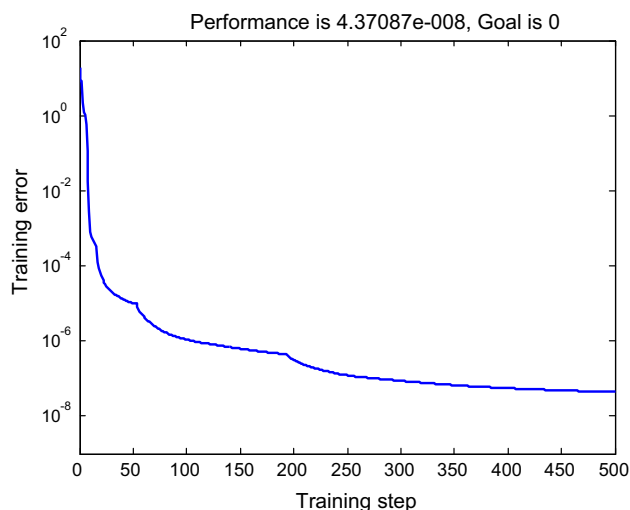


Fig. 6 Training error curve of NETA1 in example 1

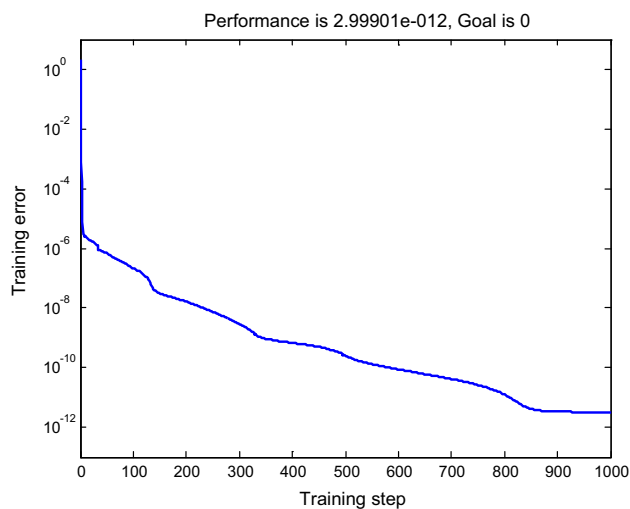


Fig. 7 Training error curve of NETA2 in example 1

maximum training steps are determined and listed in Table 2.

Table 1 The statistics of random variables for example 1

Variable	Mean μ	Standard deviation δ	Dimension	Distribution
Intensity x_1	1000.0	200.0	MPa	Normal
Height x_2	250	37.5	mm	Normal

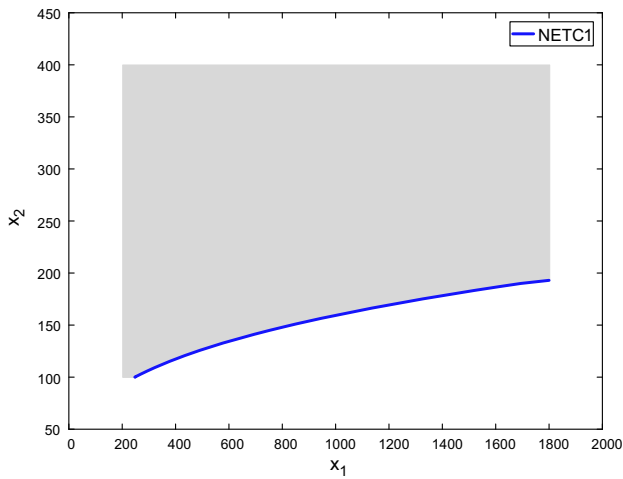


Fig. 8 The determined reliability integral domain by NETC1

The error convergence curves of NETA1 and NETA2 are shown in Figs. 6 and 7, which show that all ANNs have good training performance under the set network parameters. NETC1 is trained to construct the integral domain, and some relevant network parameters are shown in Table 2. The number of sample points along each variable axis is 40 to generate the training samples and calculate the corresponding performance function values. Variable x_2 and performance function $G(x_1, x_2)$ are used as the inputs of NETC1, while x_1 is taken as the output. Set the performance function in limit state $G(x_1, x_2) = 0$, the trained NETC1 becomes a function of x_1 with respect to x_2 . The reliability integral domain is easily determined based on the trained NETC1 and the variable ranges, as shown in Fig. 8.

Following the calculation process in Sect. 4.2, we get the reliability result by the newly proposed MCNN method. Using the same neural network parameters as the proposed method, we also calculate the reliability by DNN and Optimized DNN (ODNN) methods. The approximate solutions obtained by the ANN-based response surface method (ANNRSM) (Ren and Bai 2011), uniform design method with artificial neural network based genetic algorithms (UDM-ANN-GA) (Cheng and Li 2008), DNN method, ODNN method, MCS method (Rajashekhar and Ellingwood 1993) and newly proposed MCNN method are compared in Table 3.

Table 3 Reliability calculation results by different methods for Example 1

Method	ANNRSM	UDM-ANN-GA	DNN	ODNN	MCNN	MCS
Reliability	0.990305	0.990405	0.990786	0.990498	0.990393	0.990393
Relative error (%)	0.0089	0.0012	0.0397	0.0106	0	0

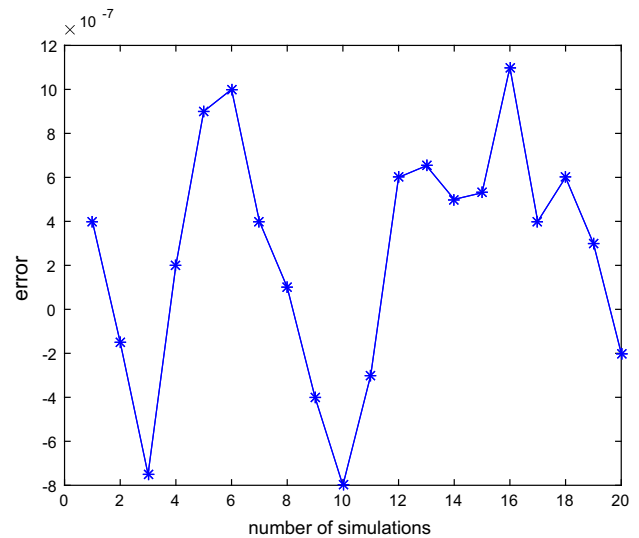


Fig. 9 The absolute error of each simulation

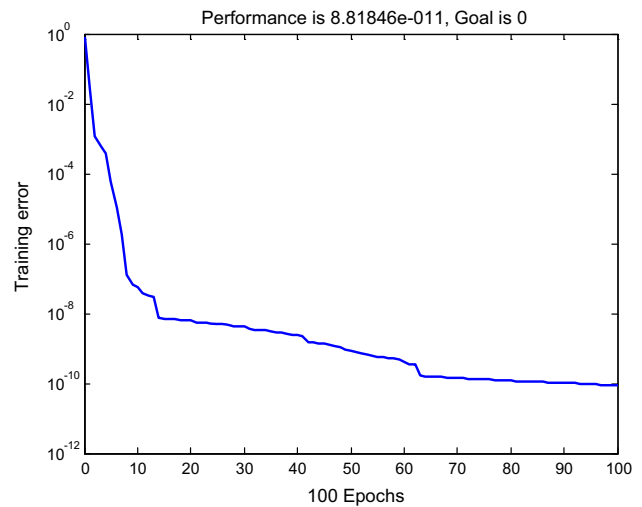


Fig. 10 Training error curve of NETA2 in example 2

As shown in Table 3, the proposed method has higher calculation accuracy of reliability than other methods when compared to the MCS result, and the calculation error is less than $1e-6$. Only finite training steps show the high running efficiency of the newly proposed method. The trained NETA1 in this example can be used in other reliability calculation problems when the random variables

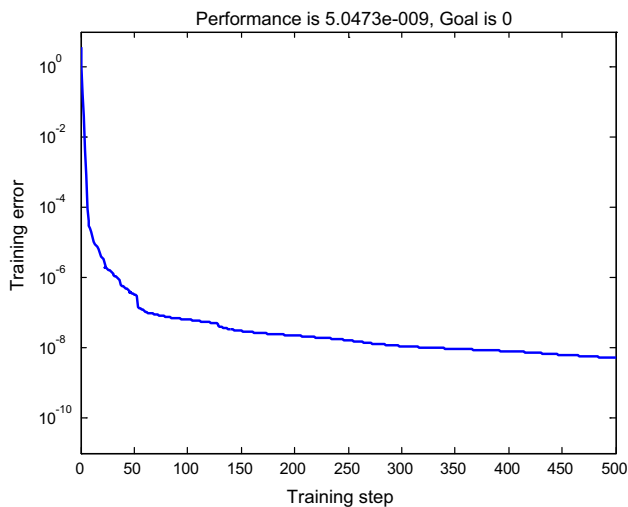


Fig. 11 Training error curve of NETC in example 2

have identical probability distributions, which can significantly improve the computational efficiency of reliability.

To illustrate the stability of the method in calculation accuracy of reliability, we conducted 20 independent experiments and the absolute errors of reliability results calculated for each experiment are shown in Fig. 9. It is seen from Fig. 9 that the absolute error of each simulation is less than $1e-6$, which shows that the proposed method is stable with high accuracy in reliability calculation.

5.2 Example 2

A nonlinear exponential performance function (Kim and Na 1997; Kaymaz and McMahon 2005; Roussouly et al. 2013) is considered in this example, the function can be expressed as:

$$G(X) = \exp[0.4(x_1+2)+6.2] - \exp[0.3x_2+5] - 200$$

where variables x_1 and x_2 are assumed to be independent and have a standard normal distributions with zero mean and unit standard deviation.

The variables in this example have a similar distribution to those in example 1, so the NETA1 trained in example 1 can be used directly. The network parameters listed in Table 2 are used in this example. The convergence curve error of NETA2 is shown in Fig. 10, which shows the high training efficiency. The NETC is constructed to approximate the reliability integral domain. The ranges of $[\mu_{x_1} \pm 4\delta_{x_1}]$ and $[\mu_{x_2} \pm 4\delta_{x_2}]$ are, respectively, divided into 40

parts, two variables cross each other to form the sample points, and the corresponding $G(x_1, x_2)$ is calculated. Variable x_2 and $G(x_1, x_2)$ are selected as the input variables and x_1 as the output variable. After adjustment, the number of neurons in hidden layer is set as 30, the maximum number of training steps is set as 500. The convergence curve of the training errors is shown in Fig. 11.

Figure 11 shows that the NETC has good training performance under the set network parameters, which ensures the accuracy of the reliability interval domain expressed by NETC. After training NETC, the performance function is set to the limit state $G(x_1, x_2) = 0$. The NETC is equivalent to a single input variable x_2 and a single output variable x_1 , which reflects the functional relationship between the two variables in the limit state. Combining the variable ranges, the reliability integral domain determined is determined by the trained NETC.

Following the calculation process in Sect. 4.2, we get the reliability result of the example. The reliability is calculated using the same training steps and network parameters by DNN and ODN methods, respectively. The approximate solutions obtained by improved sequential response surface method (ISRS) (Kim and Na 1997), doubly weighted moving least squares (DWMLS) method (Li et al. 2012), response surface using downhill simplex algorithm (DSA-RSM) (Su et al. 2019), DNN method, ODN method, MCS (Li et al. 2012) and the newly proposed MCNN method are compared in Table 4.

As shown in Table 4, the proposed method has higher accuracy of reliability calculation than other methods when compared to the MCS result. Only finite training steps show the high efficiency of the newly proposed method.

6 Conclusion

IN this paper, a newly multiple correlation neural network (MCNN) method is developed for the direct integration method of reliability calculation. The computational accuracy and efficiency of the method have been tested by representative examples and compared with some existing methods. The results show that the proposed reliability calculation method has higher accuracy and efficiency, and the computational results are stable at high accuracy. This paper details the proposed ANN-based reliability calculation method, which can be used as a paradigm for many other numerical applications.

Table 4 Reliability calculation results by different methods for Example 2

Method	ISRS	DWMLS	DNN	ODNN	DSA-RSM	MCNN	MCS
Reliability	0.99644	0.99639	0.99463	0.99653	0.99661	0.99629	0.99632
Relative error (%)	0.0120	0.0070	0.1696	0.02108	0.0291	0.0030	0

Acknowledgments The work was supported by National Natural Science Foundation of China (51975110), Liaoning Revitalization Talents program (XLYC1907171), and Fundamental Research Funds for the Central Universities (N2003005).

Funding The authors have not disclosed any funding.

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest Authors Shangjie Li, Xianzhen Huang, Xingang Wang, Yuxiong Li declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

- Aljarah I, Faris H, Mirjalili S (2018) Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Comput* 22(1):1–15. <https://doi.org/10.1007/s00500-016-2442-1>
- Allahviranloo T (2005) Romberg integration for fuzzy functions. *Appl Math Comput* 168(2):866–876. <https://doi.org/10.1016/j.amc.2004.09.036>
- Asteris PG, Nozhati S, Nikoo M, Cavaleri L, Nikoo M (2019) Krill herd algorithm-based neural network in structural seismic reliability evaluation. *Mech Adv Mater Struct* 26(13):1146–1153. <https://doi.org/10.1080/15376494.2018.1430874>
- Au SK, Beck JL (2001) Estimation of small failure probabilities in high dimensions by subset simulation. *Probab Eng Mech* 16(4):263–277. [https://doi.org/10.1016/S0266-8920\(01\)00019-4](https://doi.org/10.1016/S0266-8920(01)00019-4)
- Bucher C, Most T (2008) A comparison of approximate response functions in structural reliability analysis. *Probab Eng Mech* 23(2–3):154–163. <https://doi.org/10.1016/j.probangmech.2007.12.022>
- Cardoso JB, Almeida JRD, Dias JM, Coelho PG (2008) Structural reliability analysis using monte carlo simulation and neural networks. *Adv Eng Softw* 39(6):505–513. <https://doi.org/10.1016/j.advengsoft.2007.03.015>
- Cheng J, Li QS (2008) Reliability analysis of structures using artificial neural network based genetic algorithms. *Comput Methods Appl Mech Eng* 197(45–48):3742–3750. <https://doi.org/10.1016/j.cma.2008.02.026>
- Chojaczyk AA, Teixeira AP, Neves LC, Cardoso JB, Soares CG (2015) Review and application of artificial neural networks models in reliability analysis of steel structures. *Struct Saf* 52(3):78–89. <https://doi.org/10.1016/j.strusafe.2014.09.002>
- Dai H, Zhang H, Wang W (2015) A multiwavelet neural network-based response surface method for structural reliability analysis. *Comput Aided Civ Infrastruct Eng* 30(2):151–162. <https://doi.org/10.1111/mice.12086>
- Dey A, Miyani G, Sil A (2020) Application of artificial neural network (ANN) for estimating reliable service life of reinforced concrete (RC) structure bookkeeping factors responsible for deterioration mechanism. *Soft Comput* 24:2109–2123. <https://doi.org/10.1007/s00500-019-04042-y>
- Du J, Li H (2019) Direct integration method based on dual neural networks to solve the structural reliability of fuzzy failure criteria. *Proc Inst Mech Eng Part C J Mech Eng Sci* 233(19–20):7183–7196. <https://doi.org/10.1177/0954406219868498>
- Genz AC, Malik AA (1980) Remarks on algorithm 006: an adaptive algorithm for numerical integration over an N-dimensional rectangular region. *J Comput Appl Math* 6(4):295–302. [https://doi.org/10.1016/0771-050X\(80\)90039-X](https://doi.org/10.1016/0771-050X(80)90039-X)
- Goh AT, Kulhawy FH (2003) Neural network approach to model the limit state surface for reliability analysis. *Can Geotech J* 40(6):1235–1244. <https://doi.org/10.1139/t03-056>
- Gomes HM, Awruch AM (2004) Comparison of response surface and neural network with other methods for structural reliability analysis. *Struct Saf* 26(1):49–67. [https://doi.org/10.1016/S0167-4730\(03\)00022-5](https://doi.org/10.1016/S0167-4730(03)00022-5)
- Hagan MT, Menhaj MB (1994) Training feedforward networks with the Marquardt algorithm. *IEEE Trans Neural Netw* 5(6):989–993. <https://doi.org/10.1109/72.329697>
- Jin N, Liu D (2008) Wavelet basis function neural networks for sequential learning. *IEEE Trans Neural Netw* 19(3):523–528. <https://doi.org/10.1109/TNN.2007.911749>
- Kaymaz I, McMahon CA (2005) A response surface method based on weighted regression for structural reliability analysis. *Probab Eng Mech* 20(1):11–17. <https://doi.org/10.1016/j.probangmech.2004.05.005>
- Kim SH, Na SW (1997) Response surface method using vector projected sampling points. *Struct Saf* 19(1):3–19. [https://doi.org/10.1016/S0167-4730\(96\)00037-9](https://doi.org/10.1016/S0167-4730(96)00037-9)
- Li J, Wang H, Kim NH (2012) Doubly weighted moving least squares and its application to structural reliability analysis. *Struct Multidisciplinary Optim* 46(1):69–82. <https://doi.org/10.1007/s00158-011-0748-2>
- Li H, He Y, Nie X (2018) Structural reliability calculation method based on the dual neural network and direct integration method. *Neural Comput Appl* 29(7):425–433. <https://doi.org/10.1007/s00521-016-2554-7>
- Li H, Li Y, Li S (2019) Dual neural network method for solving multiple definite integrals. *Neural Comput* 31(1):208–232. https://doi.org/10.1162/neco_a_01145
- Li SJ, Huang XZ, Wang DH (2022) Stochastic configuration networks for multi-dimensional integral evaluation. *Inf Sci* 601:323–339. <https://doi.org/10.1016/j.ins.2022.04.005>
- Liao SH, Hsieh JG, Chang JY, Lin CT (2015) Training neural networks via simplified hybrid algorithm mixing nelder–mead and particle swarm optimization methods. *Soft Comput* 19(3):679–689. <https://doi.org/10.1007/s00500-014-1292-y>
- Liu P, Kiureghian AD (1991) Optimization algorithms for structural reliability. *Struct Saf* 9(3):161–177. [https://doi.org/10.1016/0167-4730\(91\)90041-7](https://doi.org/10.1016/0167-4730(91)90041-7)
- Liu D, Peng Y (2012) Reliability analysis by mean-value second-order expansion. *J Mech Des* 134(6):061005. <https://doi.org/10.1115/1.4006528>
- Lloyd S, Irani RA, Ahmadi M (2020) Using neural networks for fast numerical integration and optimization. *IEEE Access* 8:84519–84531. <https://doi.org/10.1109/ACCESS.2020.2991966>
- Nezhad HB, Miri M, Ghasemi MR (2019) New neural network-based response surface method for reliability analysis of structures. *Neural Comput Appl* 31(3):777–791. <https://doi.org/10.1007/s00521-017-3109-2>
- Nie J, Ellingwood BR (2000) Directional methods for structural reliability analysis. *Struct Saf* 22(3):233–249. [https://doi.org/10.1016/s0167-4730\(00\)00014-x](https://doi.org/10.1016/s0167-4730(00)00014-x)
- Niederreiter H, Spanier J (2000) Monte Carlo and quasi-Monte Carlo methods. Springer, Heidelberg

- Papadopoulos V, Giovanis DG, Lagaros ND, Papadrakakis M (2012) Accelerated subset simulation with neural networks for reliability analysis. *Comput Methods Appl Mech Eng* 223:70–80. <https://doi.org/10.1016/j.cma.2012.02.013>
- Papadrakakis M, Lagaros ND (2002) Reliability-based structural optimization using neural networks and Monte Carlo simulation. *Comput Methods Appl Mech Eng* 191(32):3491–3507. [https://doi.org/10.1016/S0045-7825\(02\)00287-6](https://doi.org/10.1016/S0045-7825(02)00287-6)
- Papadrakakis M, Papadopoulos V, Lagaros ND (1996) Structural reliability analysis of elastic-plastic structures using neural networks and Monte Carlo simulation. *Comput Methods Appl Mech Eng* 136(1–2):145–163. [https://doi.org/10.1016/0045-7825\(96\)01011-0](https://doi.org/10.1016/0045-7825(96)01011-0)
- Place J, Stach J (1999) Efficient numerical integration using Gaussian quadrature. *SIMULATION* 73(4):232–237. <https://doi.org/10.1177/003754979907300405>
- Rajashekhar MR, Ellingwood BR (1993) A new look at the response surface approach for reliability analysis. *Struct Saf* 12(3):205–220. [https://doi.org/10.1016/0167-4730\(93\)90003-J](https://doi.org/10.1016/0167-4730(93)90003-J)
- Ren Y, Bai G (2011) New neural network response surface methods for reliability analysis. *Chin J Aeronaut* 24(1):25–31. [https://doi.org/10.1016/S1000-9361\(11\)60004-6](https://doi.org/10.1016/S1000-9361(11)60004-6)
- Roussouly N, Petitjean F, Salaun M (2013) A new adaptive response surface method for reliability analysis. *Probab Eng Mech* 32:103–115. <https://doi.org/10.1016/j.probengmech.2012.10.001>
- Rubinstein RY, Kroese DP (2007) *Simulation and the Monte-Carlo method*. Wiley, New York
- Shayanfar MA, Barkhordari MA, Barkhori M, Barkhori M (2018) An adaptive directional importance sampling method for structural reliability analysis. *Struct Saf* 70:14–20. <https://doi.org/10.1016/j.strusafe.2017.07.006>
- Simos TE (2009) Closed Newton-cotes trigonometrically-fitted formulae of high order for long-time integration of orbital problems. *Appl Math Lett* 22(10):1616–1621. <https://doi.org/10.1016/j.aml.2009.04.008>
- Su H, Lan F, He Y, Chen J (2019) A modified downhill simplex algorithm interpolation response surface method for structural reliability analysis. *Eng Comput* 37(4):1423–1450. <https://doi.org/10.1108/EC-03-2019-0085>
- Tvedt L (1990) Distribution of quadratic forms in normal space—application to structural reliability. *J Eng Mech* 116(6):1183–1197. [https://doi.org/10.1061/\(ASCE\)0733-9399\(1990\)116:6\(1183\)](https://doi.org/10.1061/(ASCE)0733-9399(1990)116:6(1183))
- Wang DH, Li M (2017) Stochastic configuration networks: fundamentals and algorithms. *IEEE Trans Cybern* 47(10):3466–3479. <https://doi.org/10.1109/TCYB.2017.2734043>
- Yan F, Lin Z (2016) New strategy for anchorage reliability assessment of GFRP bars to concrete using hybrid artificial neural network with genetic algorithm. *Compos B Eng* 92:420–433. <https://doi.org/10.1016/j.compositesb.2016.02.008>
- Yoon S, Lee YJ, Jung HJ (2020) Accelerated monte carlo analysis of flow-based system reliability through artificial neural network-based surrogate models. *Smart Struct Syst* 26(2):175–184. <https://doi.org/10.12989/sss.2020.26.2.175>
- Zeng ZZ, Wang YN, Wen H (2006) Numerical integration based on a neural network algorithm. *Comput Sci Eng* 8(4):42–48. <https://doi.org/10.1109/MCSE.2006.73>
- Zhang W, Cui W (1997) Direct integration method for structural reliability calculation. *J Shanghai Jiao Tong Univ* 31(2):114–116
- Zhang J, Du X (2010) A second-order reliability method with first-order efficiency. *J Mech Des* 132(10):101006. <https://doi.org/10.1115/1.4002459>
- Zhang T, He D (2018) An improved high-order statistical moment method for structural reliability analysis with insufficient data. *Proc Inst Mech Eng Part C J Mech Eng Sci* 232(6):1050–1056. <https://doi.org/10.1177/0954406217694662>
- Zhang Z, Jiang C, Wang GG, Han X (2015) First and second order approximate reliability analysis methods using evidence theory. *Reliab Eng Syst Saf* 137:40–49. <https://doi.org/10.1016/j.res.2014.12.011>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.