

---

**Joint particle size calculation model (JCM): a neural network approach for efficient rock and soil simulation****Jichao Sun**

Sun, J. 2025. Joint particle size calculation model (JCM): a neural network approach for efficient rock and soil simulation. *Baltica* 38 (1), 32–44. Vilnius. ISSN 0067-3064.

Manuscript submitted 6 October 2024 / Accepted 11 April 2025 / Available online 2 May 2025

© Baltica 2024

**Abstract.** The numerical simulation of rock and soil mass in hydropower engineering will consume a lot of human resources and electric power required for simulation. A rapid and efficient establishment of a rock and soil numerical model is urgent for numerical simulation. The equivalence of the material grading curve is an essential criterion for the accurate simulation of geotechnical material by the discrete element, and particle size calculation is the premise of the grading curve. The discrete element of joint particles can accurately simulate the geotechnical particles in reality, but it is difficult and time-consuming to calculate the particle size of joint particles. This paper presents a neural network calculation model for joint particle size, that is, the joint particle size calculation model (JCM), quickly estimating particle size. The results show that the neural network model with 3–5 circle balls and 11–13 hidden neurons can obtain the gradation curve with a good coincidence degree. The model significantly reduces the calculation time. This paper establishes a joint particle database, which provides standard data for simulation and substantially improves simulation efficiency. The database enhances the comparability and standardization of research and makes research results more universal. The research significantly reduces computers' human resources and power consumption in numerical simulation.

**Keywords:** soil simulation; rock simulation; simulation efficiency; particle simulation; particle database; particle model database; discrete element method

✉ Jichao Sun ([sunjc@cugb.edu.cn](mailto:sunj@cugb.edu.cn); [jichao@email.com](mailto:jichao@email.com)),  <https://orcid.org/0000-0002-7561-1223>

Key Laboratory of Groundwater Conservation of MWR,

School of Water Resources and Environment, China University of Geosciences, Beijing, 100083, China

---

**INTRODUCTION**

Using numerical simulation methods to study geotechnical engineering and geological engineering problems encountered in hydropower engineering construction can significantly reduce research costs compared with field and laboratory tests. Using the corresponding numerical simulation methods to build a numerical model can quickly and dramatically reduce computing time and the consumption of power resources in computer operation, as well as further realize green development. Therefore, a rapid and efficient establishment of a numerical model of rock and soil requires geotechnical and geological engineering numerical simulation.

The primary purpose of numerical simulation is to use the simulated rock and soil mass to be equivalent

to the natural rock and soil mass. The stress-strain and deformation-displacement can reflect the field by applying force on the simulated rock and soil material.

Numerical simulation methods include the finite element method (Fakher, Hosseini, Hashemi 2020), the boundary element method (Palade *et al.* 2020; Coox *et al.* 2017), and the discrete element method. The discrete element simulation method (Ghodki *et al.* 2019; Izard *et al.* 2020; Garcia-Archilla, Novo 2020) can reasonably simulate a significant deformation problem of granular bodies, cracks (Nitka, Tejchman 2020), and other structures under stress. Because of the flexible and fast characteristics of circular particle discrete elements, many researchers have widely used circular particle discrete elements.

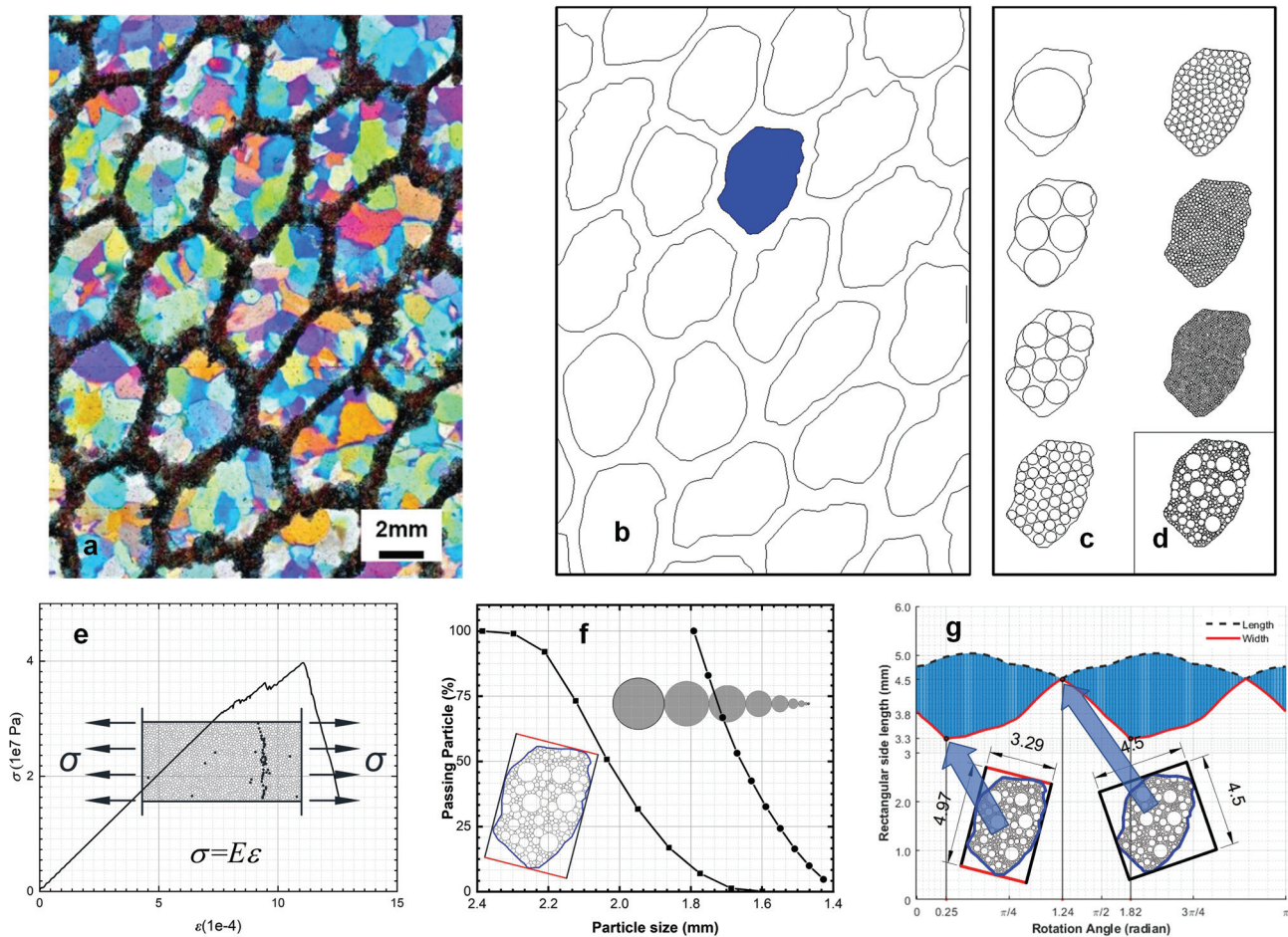
Circular particle discrete elements can simulate various hydraulic structures, geological bodies, soils

and rock, and seepage calculation (Fakhimi 2009; Sun 2016). A comparison was conducted between the experimental and numerical simulation results to evaluate the model's accuracy and reliability. Rocks and soil are made up of many particles mixed (Fig. 1a and b). A joint particle can simulate a complex particle (Fig. 1c). Joint particles (Fig. 1c) are non-circular particles formed by combining multiple elementary particles (typically circular particles) through specific geometric relationships. These circular particles can be arranged in various configurations to create joint particles with diverse and complex shapes. By adjusting the size, arrangement, and geometric relationships of the elementary particles, researchers can more flexibly construct a wide range of granular systems (Sun, Huang 2022), enabling the generation of soil structures that meet specific requirements (Fig. 1a and b). This approach allows for a more accurate representation of real soil characteristics. By employing realistic joint particles in numerical simulations, the precision and applicability of modelling soil mechanical

behaviour can be significantly enhanced. The more particles are used, the higher the simulation accuracy is (Sun 2018).

A joint particle simulates field materials by needing strain and strain consistency in the mechanics' process (Fig. 1e). The consistency of the results is achieved by adjusting the mechanical parameters in past research. However, the last result consistency, for example, stress/strain and displacement/deformation, is not factual consistency and does not mean the consistency of the materials, because the result is obtained through a complex mechanical operation. The factual consistency means that all simulation processes are precisely equivalent to reality. The materials equivalence is the first step, including the particle size and particle grain size distribution.

The particle size is of great significance in the behaviour of particulate materials and numerical simulations. Adesina *et al.* (2025) found that the particle size ratio significantly affects the mechanical properties of gap-graded granular assemblies. It behaves



**Fig. 1** Joint particle simulating rock/soils and the equivalence principle: (a) coral rock: dark places are coral skeletons, and the coloured ones are fillings between them; (b) skeletons and fillings boundary, the blue part is chosen to be studied; (c) different numbers of spherical particles simulating one filling, the quantity is 1, 5, 10, 50, 100, 500 and 1000; (d) different diameter ball filling; (e) ball discrete element simulation of soil tensile crack failure to obtain strain and stress; (f) gradation difference of the ball and joint particle discrete element simulates particle; (g) the rotation process of calculating the particle size

differently under different fine particle contents and has different effects on internally stable and unstable assemblies. This is consistent with the conclusions of Fatemeh *et al.* (2024) regarding the influence of the particle size ratio on shear strength when studying binary granular soils. Mohammad and Abouzar (2025) conducted direct simple shear tests through discrete element simulations and pointed out that the particle size and specimen size can affect the stress path, peak and post-peak shear strengths (Mohammad, Abouzar 2025). Reinecke *et al.* (2023) studied and found that the particle size affects the focusing and separation effects in serpentine microchannels. The equilibrium trajectories of particles with different sizes are different, indicating that accurate particle size calculation is crucial for the accuracy of simulation results. In discrete element simulations, particle size calculation is a prerequisite for obtaining the equivalence of material grading curves and is indispensable for accurately simulating the characteristics of geotechnical materials. These studies provide important bases for a deeper understanding of particulate material behaviour and the optimization of numerical simulations.

The gradation of the site rock and soil mass is the same as that of the simulated rock and soil mass, the essential standards to ensure that the material simulation results are consistent with the actual rock and soil mass. To achieve helpful equivalence between the two, the gradation curves of the two should coincide.

Much research uses circle diameter as the joint particle size to calculate the gradation curve, which is seriously wrong, as shown in Fig. 1f. The two gradation curves of circle diameter and joint particle width are different. There is no suitable method to calculate the particle size of joint particles, and the rotary particle size calculation process is very complex, time-consuming, and challenging. The research aims to get particle size quickly by the neural network.

Mohammadkhanifard, Zad (2024) used the discrete element method to simulate the interaction between dynamic anchors and particles of different shapes and sizes and analyzed the pullout capacity of the anchors. However, this method faces certain challenges. For example, the computational load is high, and the processing of particle shapes and sizes affects the accuracy of the results. When Reinecke *et al.* (2023) simulated the behaviour of particles in microchannels, they needed to consider the interaction between particles and the fluid as well as the influence of the channel geometry, resulting in complex calculations. The particle size calculation of composite particles (Sun, Huang 2022) is a crucial prerequisite for DEM calculation. It has been possible to accurately calculate the particle size of composite particles (Sun, Huang 2022). The method used to calculate the joint particle size is the Rotation Calculation Model

(RCM) (Sun, Huang 2022). However, when dealing with a large number of particles using RCM model, a significant amount of calculation time is required. Therefore, new calculation methods for the particle size of composite particles need to be considered. When simulating geotechnical engineering problems, multiple factors should be comprehensively considered, and the discrete element simulation method should be optimized to improve simulation efficiency and accuracy. Kishida *et al.* (2025) and others developed a machine learning-based surrogate model RNNSR to address a high computational cost of discrete element simulations and extended it to simulate the mixing and separation of binary-sized particles, improving the calculation speed.

Therefore, the primary purpose of this paper is to reduce the calculation difficulties and save the calculation time of the particle size of joint particles. Machine learning (Samalavicius *et al.* 2024) and neural network computational methods (Karamut, Binal 2024; Timurkutluk *et al.* 2023) are essential approaches for rapidly solving complex parameters and addressing engineering challenges (Bilgilioglu 2023). This study proposes a neural network model to directly calculate the particle size of joint particles by the parameters of compositing ball particles. A standard database of joint particles was established, including parameters such as particle size of joint particles. At the same time, we keep the database open. Future researchers can calculate the particle size of joint particles directly according to elementary particles' parameters if they need more involved and different particles. The database and the neural network model will significantly simplify the workload of scientific researchers. Moreover, they enhance the comparability and scientificity of varying calculation models and research.

## ESTABLISHMENT OF NEURAL NETWORK MODEL AND DATABASE: JOINT PARTICLE SIZE CALCULATION MODEL (JCM)

### Database establishment

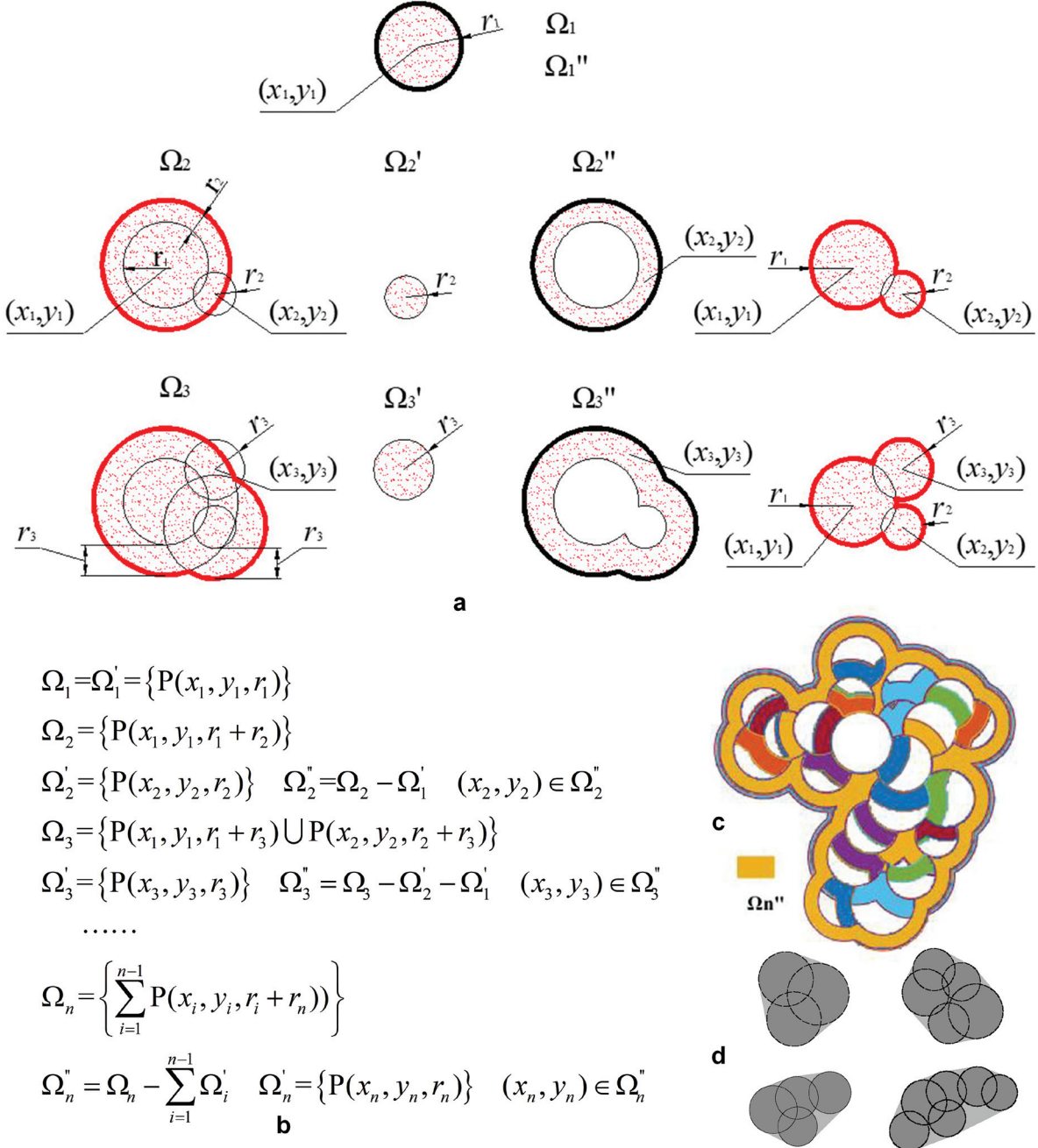
A spherical discrete element is an essential branch of discrete element simulation, and many research results have emerged. We use many particles to simulate soils and rock. The greater the number is, the closer the external geometric shape to the particles in reality is (Jarrar *et al.* 2020). The digital approach to particle packing is established by digitizing particle shapes and pack space. The DEM (discrete element method) particles can accurately simulate the actual three-dimensional sand particles' shape, carry out the crushing simulation experiment and obtain numerical results agreeing with the experimental results (Jarrar *et al.* 2020).

However, the more simulated balls the researchers used, the more time it took. The current calculation cannot use a too large number of spheres to simulate particles.

The joint particles are composed of multiple circular particles. The parameters of each circular particle include the centre and the radius, and these particles need to overlap. The usual method randomly generates these parameters in a specific area to determine the superposition. Consider forming a joint particle if

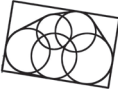
any particle co-occurs with at least one other particle. Because complete randomness cannot guarantee particle superposition, circular balls are generated continuously. It takes much time to judge and develop continually, and it is challenging to generate a large number of joint particles.

The steps of joint particle generation are as follows: (1) generate random parameters to generate a random circle 1. The parameters are the centre and the radius of the circle,  $(x_1, y_1, r_1)$ , and the circle area



**Fig. 2** Flow chart of joint particles generation: (a) The formation process of joint particles; (b) An example of superimposed particle centre zone; (c) Symbolic representation of each region; (d) Examples of joint particles composed of 3, 4, 5, and 6 circles.  $(x_i, y_i, r_i)$  is a circular particle whose radius is  $r_i$  at the centre  $(x_i, y_i)$ ,  $i = 1, 2, 3, \dots, n$ .  $\Omega_1$  and  $\Omega_1'$  is the first circular area.  $\Omega_2$  is the circular area with  $(x_1, y_1)$  as the centre and  $r_1 + r_2$  as the radius.  $\Omega_2'$  is a circular area with  $(x_n, y_n)$  as its centre and  $r_n$  as its radius.  $\Omega_n''$  is the ring of the centre  $(x_n, y_n)$  of the  $n$ th particle.  $\Omega_n$  is the sum of circular areas with  $(x_i, y_i)$  as the circle and  $r_i + r_n$  as the radius,  $i = 1, 2, \dots, n-1$

**Table 1** Three examples of the joint in the database

| No. in database | Particle image                                                                    | Initial parameters |        |         | Result |        |         |
|-----------------|-----------------------------------------------------------------------------------|--------------------|--------|---------|--------|--------|---------|
|                 |                                                                                   | $x$                | $y$    | $r$     | Width  | Height | Angle   |
| 3-100           |  | 0                  | 0      | 2.39294 | 5.605  | 11.783 | 125.478 |
|                 |                                                                                   | 2.068              | -2.147 | 2.640   |        |        |         |
|                 |                                                                                   | 5.291              | -4.113 | 2.695   |        |        |         |
| 4-1000          |  | 0                  | 0      | 2.549   | 6.461  | 14.498 | 24.637  |
|                 |                                                                                   | -2.635             | -2.635 | 2.616   |        |        |         |
|                 |                                                                                   | -2.858             | -6.410 | 2.471   |        |        |         |
| 5-10000         |  | 0.933              | 2.621  | 2.238   | 8.138  | 12.442 | 99.122  |
|                 |                                                                                   | 0                  | 0      | 2.619   |        |        |         |
|                 |                                                                                   | 2.493              | 2.815  | 2.337   |        |        |         |
|                 |                                                                                   | 0.019              | 2.796  | 2.022   |        |        |         |
|                 |                                                                                   | 3.846              | -0.313 | 2.927   |        |        |         |
|                 |                                                                                   | -3.039             | 0.611  | 2.571   |        |        |         |

is  $\Omega_1$ , as shown in Fig. 2a and b. (2) Generate the radius  $r_2$  of the following random circle 2 and randomly determine a circle centre  $(x_2, y_2)$  in the region  $\Omega_2$ , that is, generate the joint particles composed of two superimposed circles. (3) Generate the radius  $r_3$  of the third random circle 3 and randomly determine a circle centre  $(y_3)$  in the region  $\Omega_3$ , that is, generate a joint particle composed of three superimposed circles. Subsequently, generate joint particles consisting of multiple particles by this analogy.

This paper establishes the joint particle database composed of 3, 4, 5 spherical particles. The database consists of 310000 particle data. Each particle parameter includes the centre and the diameter of the circular particle, the minimum width of the composite particle covering the length and width of the rectangular, and the rotation angle of the rectangular. Table 1 shows some examples of three composite particles in the database.

### Joint particle size calculation model (JCM)

Currently, the computational cost of discrete element methods is exceedingly high. During the modelling phase of granular flow, it is necessary to calculate a vast number of particles, particularly when determining the particle size distribution of composite particles. When dealing with tens of thousands of particles, this process becomes extremely time-consuming. Therefore, it is essential to explore and propose novel methodologies to address this issue.

Neural network models have achieved significant research breakthroughs in the field of Earth sciences. They are widely applied in earthquake prediction, where they learn from historical seismic data to identify precursors and predict the likelihood of seismic events. By analyzing seismic waveform data, these models enhance the accuracy of earthquake early warning systems (Mousavi *et al.* 2020). In climate prediction, neural networks are capable of handling

complex nonlinear climate systems. For instance, deep learning-based models have been used to predict extreme weather events, such as typhoons and heavy rainfall, as well as long-term climate change trends (Ham *et al.* 2019). In hydrology, neural network models are employed for river flow prediction, flood forecasting, and water resource management, significantly improving prediction accuracy (Kratzert *et al.* 2018).

This study leverages neural network models to establish a neural network model for predicting the particle size distribution of composite particles. This paper presents a neural network machine learning computing model. A neural network calculation model for joint particle size, that is, joint particle size calculation model (JCM) is established.

Each neuron node in the neural network accepts the output value of the upper layer neuron as the input value. The neuron node of the input layer will directly transfer the attribute value to the next layer (hidden layer or output layer) (Adil *et al.* 2022).

In a multi-layer neural network, there is a functional relationship between the output of the upper node and the next node's input, called the activation function (also known as the excitation function) (Olimov *et al.* 2020).

The particle size of joint particles is studied in this paper, shown in Eq. (1). The input data are the centre and the radius of each circular particle as  $X$  in Eq. (1). The final output data are the particle size  $Y$  of joint particles, and the radius of joint particles is a number. Therefore, the dimensions of matrix  $A_m$  and  $B_m$  are  $1 \times k_{m-1}$  and  $1 \times 1$ , and the dimensions of intermediate matrix  $A_j$  and  $B_j$  are  $k_j \times k_{j-1}$  and  $k_m \times 1$ .

$$Y = \sigma(C \times E_{m-1} + D) \quad (1)$$

where  $C = [c_{m-1} c_{m-2} \dots c_m]_{1 \times k_m}$ ,  $C$  is the calculation result of each step;  $E_M = \sigma(A_M \times E_{M-1} + B_M)$ ,  $A$  is the weight matrix;  $D = [d]$ ;  $A_M = [w_{IJ}^M]$ ,  $I = 1, 2, 3, \dots, k_M$ ,  $J = 1, 2, 3, \dots, k_{M-1}$ ;  $w$  is the weight;  $X = [x_1 x_2 \dots x_n]^T$ ,

$X$  is the original input data,  $n$  is the number of initial data,  $E_0=X$ ;  $B_M=[b_1^M b_2^M \dots b_{kM}^M]^T$ ;  $M=1, 2, 3, \dots, m$ ;  $B$  and  $D$  are the linear terms;  $\sigma$  is the activation function, and  $m$  is the neuron number of the hidden layer.

It is the multiple multiplications of matrices and the weight adjusted continuously through the feedback of the results. Finally, minimize the mean square error. Save the weight matrix with the smallest mean square error. Directly calculate the result data according to the weight matrix and the input data (Carpenter, Grossberg 1988; Rivas *et al.* 2021), not through training again.

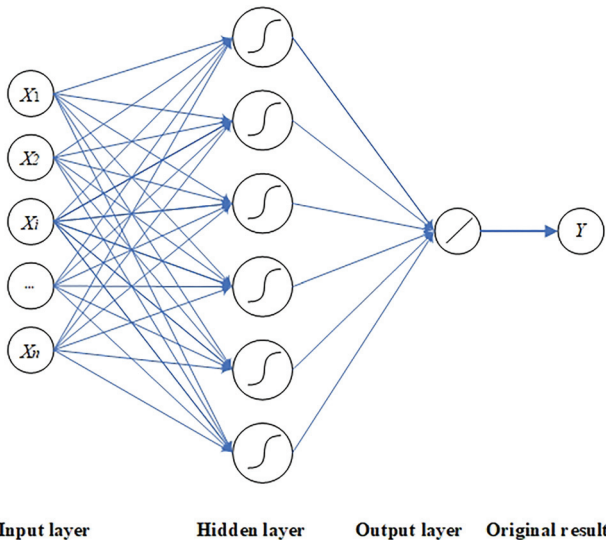
The two-layer neural network model is selected in the calculation, including a hidden layer and an output layer. The centre of the first circle is (0, 0), the input data are Eq. (2), the particle size formula is Eq. (3), and the activation function is Eq. (4). Figure 3 shows the neural network model. The number of neurons in the hidden layer ranges from 8 to 40. The particle size formula for a joint particle composed of 4 balls with 13 hidden layer neurons is Eq (5).

$$\begin{bmatrix} (x_1, y_1, r_1) \\ \vdots \\ (x_i, y_i, r_i) \\ \vdots \\ (x_n, y_n, r_n) \end{bmatrix} \rightarrow \begin{bmatrix} (0, 0, r_1) \\ (x_2, y_2, r_2) \\ \vdots \\ (x_i, y_i, r_i) \\ \vdots \\ (x_n, y_n, r_n) \end{bmatrix} \rightarrow X = [r_1, x_2, y_2, r_2, \dots, x_n, y_n, r_n]^T \quad (2)$$

$$Y_{1 \times 1} = C_{1 \times m} [\sigma(A_{m \times q} \times X_{q \times 1} + B_{m \times 1})] + D_{1 \times 1} \quad (3)$$

$$\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

$$Y_{1 \times 1} = C_{1 \times 13} [\sigma(A_{13 \times 10} \times X_{10 \times 1} + B_{13 \times 1})] + D_{1 \times 1} \quad (5)$$



**Fig. 3** Neural network model for joint particle size

where  $m$  is the neuron number of the hidden layer;  $n$  is the number of circles making up the joint particle.

After training, obtain the model. Calculate the particle size of the joint particles by inputting the centre coordinates and radius into the model. There is a specific error between the particle size calculation results of the model and the actual particle size. So, the mean square error evaluates the model. Eq. (6) is the formula of the mean square error. The smaller the mean square error is, the higher the calculation accuracy of the training model is. The mean square error is

$$M_{SE} = \frac{\sum_{i=1}^s (X_i - x_i)^2}{s} \quad (6)$$

where  $X_i$  is the  $i$ th value calculated by the model,  $x_i$  is the  $i$ th actual value, and  $s$  is the amount of data.

## CALCULATION PROCESS, RESULTS, AND DISCUSSION

In this paper, many parameters are used to study the establishment of the database and obtain relevant laws. The specific steps are as follows: (1) Select the parameters needed in this paper as input data according to the database's parameters. (2) Establish the neural network model of joint particle size. (3) Study the established model, including three convergence modes, the number of circle particles of joint particles, and the number of hidden neurons. Finally, obtain the optimal convergence mode, the number of circle particles, and the number of hidden neurons.

### Parameters

The calculation parameters are shown in Table 2.

Three kinds of convergence modes are adopted. Levenberg-Marquardt backpropagation convergence mode (Moayedi *et al.* 2020; Khatti *et al.* 2019), called Lm., Bayesian regularization backpropagation con-

**Table 2** Parameters of the neural network model for joint particle size

|                                                            |                              |        |
|------------------------------------------------------------|------------------------------|--------|
| Types of composite particles                               | Number of circle particles   | 3–12   |
|                                                            | Number of joint particles    | 10     |
| Hidden layer                                               | Number of hidden neurons     | 8–40   |
|                                                            | Number of hidden layer types | 33     |
| Convergence control parameters                             | Timestep of learning (s)     | 0.05   |
|                                                            | Maximum training time (s)    | 3000   |
|                                                            | Minimum mean square error    | 0.0001 |
|                                                            | Maximum training time (s)    | 60     |
| Number of neural network models of a single joint particle |                              | 20     |
| Number of convergence mode                                 |                              | 3      |
| Total number of neural network models                      |                              | 19800  |

vergence mode (Pinzolas *et al.* 2006), called Br., and scaled conjugate gradient backpropagation convergence mode (Khadse *et al.* 2017; Nataj, Lui 2020), called Scg.

The input parameters are the centre and the radius of circle balls compositing the joint particle, and the output result is the particle size of the joint particle.

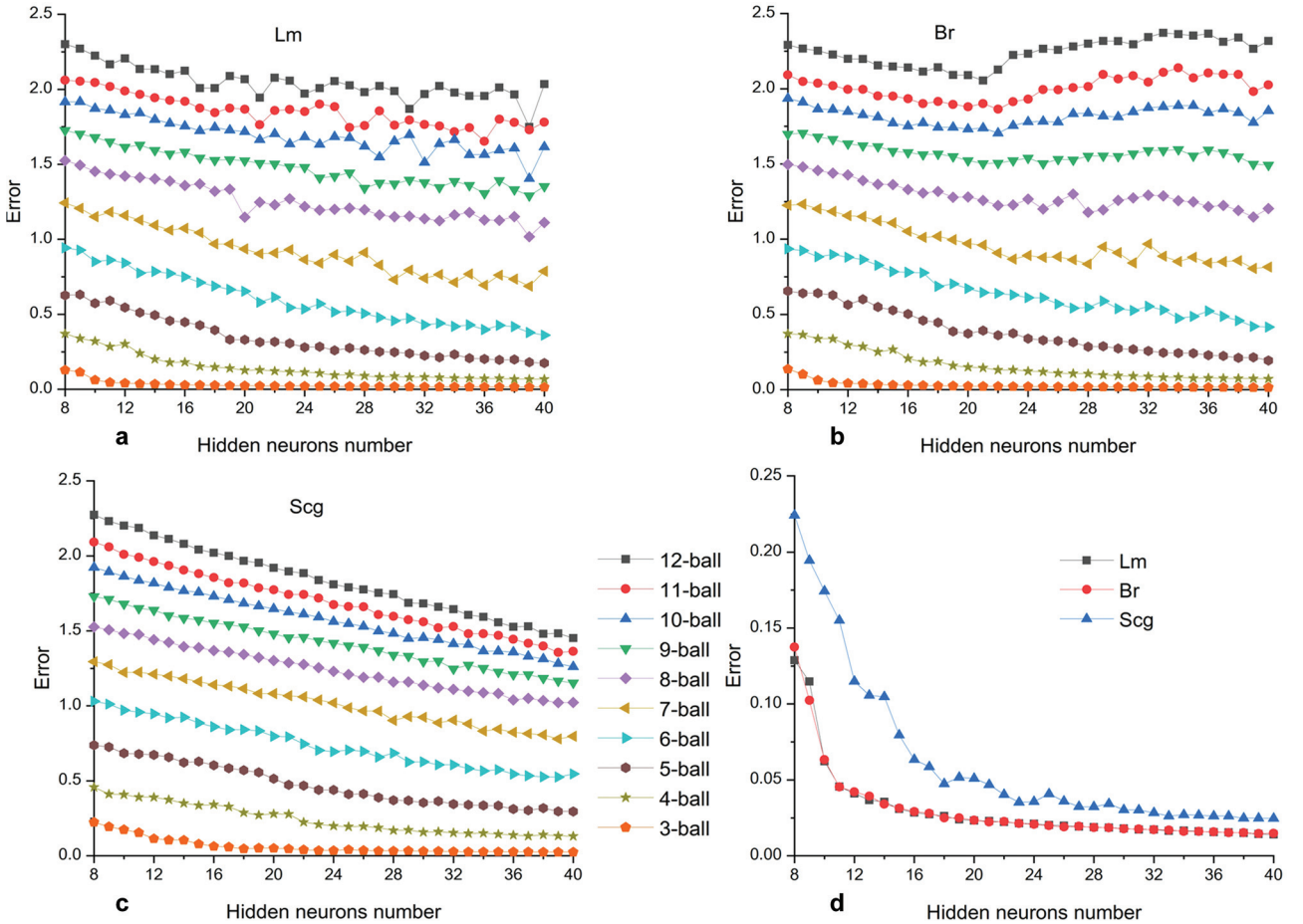
### Convergence mode and number of circle particles

The initial weights are randomly assigned when establishing the neural network training model. Terminate the training when reaching the minimum error or the total number of training times or training time. The initial weights affect the result accuracy, and the artificial assignment of the initial weights may not be the best. Therefore, this paper's solution is that each model carries out many assignment pieces of training to find the minimum error model and initial weights. At last, obtain an excellent training model as the final one.

In this paper, from the 20 training models of each joint particle under three convergences, the model with the minimum mean square error is selected as the joint particles' output model. According to Eq. (6), the final calculated mean square error results are listed in Fig. 4.

The mean square error decreases as the number of hidden neurons increases, as shown in Fig. 4a, b, c. The mean square error of 3–6 circle particles decreases gradually with the rise in the number of hidden neurons when the Lm mode is adopted, as shown in Fig. 4a. From seven circular particles, the error fluctuates.

Under the Br convergence mode, some mean square errors keep a relatively stable enormous value (10-ball curve shown in Fig. 4b) with the increase of the number of hidden neurons, as shown in Fig. 4b. Some mean square errors even tend to increase (12-ball and 11-ball curves shown in Fig. 4b). The mean square error increases with the increase of circle balls, and from six circle particles, the error fluctuates.



**Fig. 4** Mean square error of joint particles with 8-40 neurons in three convergence modes: (a) Mean square error of particle size under Lm convergence mode; (b) Mean square error of particle size under Br convergence mode; (c) Mean square error of particle size under Scg convergence mode; (d) Mean square error of three-ball joint particle size under Lm, Br, and Scg convergence modes. The mean square error in the Figure is from Eq. (6). Lm is the Levenberg-Marquardt backpropagation. Br is the Bayesian Regularization backpropagation. Scg is the Scaled conjugate gradient backpropagation

From Fig. 4c we can see that the mean square error decreases with the increase of the number of hidden neurons and the decrease of circle particles when the Scg convergence model is adopted. This rule is well maintained. From Fig. 4d we can see that the error decreases with the number of hidden neurons in the three modes when the three circles are combined. The mean square error in Lm and Br modes is almost coincident, and the mean square error in the Lm mode is slightly smaller than that in the Br mode, and the mean square error in the Scg mode is larger.

It indicates that the increase in the number of neurons under the same training parameters does not necessarily bring better training results. As the number of circles increases, the mean square error increases gradually. When the number of balls is more than 6, the error is relatively large. So, the number of circle balls in [3, 6] is acceptable. If the number of circle balls is within [3, 5], Lm and Br convergence modes perform better.

Therefore, when the number of circle balls is in the range of [3, 5] and the Lm convergence mode is applied, the neural network model can reasonably calculate the joint particles' particle size.

### Number of hidden neurons

From the previous part we see that the model's accuracy is the best when the number of circle balls is in [3, 5] and in the convergence mode of Lm. Therefore, it is necessary to further study the optimal number of hidden neurons. From Fig. 4a, b, and c we see that the number of different circle balls in [3, 5] shows consistent accuracy characteristics. Therefore, the joint particles of three circle balls are selected for the study in this part. The models with 8-40 neurons train 20 times, and the smallest mean square error is selected. The accuracy of the calculation results is shown in Fig. 5. The joint particles' size is taken as the horizontal coordinate-axis  $X$ , and the particle size from the model is taken as the vertical coordinate-axis  $Y$ .

The scatters diagram is in Fig. 5b, c, d, e, and f, and each diagram has 10000 points.

From Fig. 5a we see that the mean square error decreases when the number of neurons increases. The mean square error decreases significantly when the number of neurons changes from 10 to 11. At the number of 11–19 neurons, the mean square error decreases gradually, weakening the decrease. However, when the number of neurons is greater than 19, the decrease in the mean square error is not significant. From Fig. 5b, c, d, e, and f we see that as the number of neurons increases, the scatter points are more concentrated to the diagonal, indicating that the error gradually decreases (as shown in Fig. 5a).

### Model accuracy under the gradation curve

When the neural network model calculates the particle size of joint particles, there is a specific error between the results and the actual particle size, as shown in Table 3. Table 3 shows the first 10 joint particles composed of 4 circle balls in the database supplied by this research. Error histograms normal distribution curve of 10000 joint particles under three convergence modes is shown in Fig. 6. We can find the error distribution conforms to the normal distribution. Many joint particles constitute the real rock and soil mass. In the simulation process, the evaluation standard of the simulation effect is the maximum coincidence of the simulation material gradation curve and the natural material gradation curve. Therefore, the particle size error of many single joint particles may not affect the final simulated soil gradation curve.

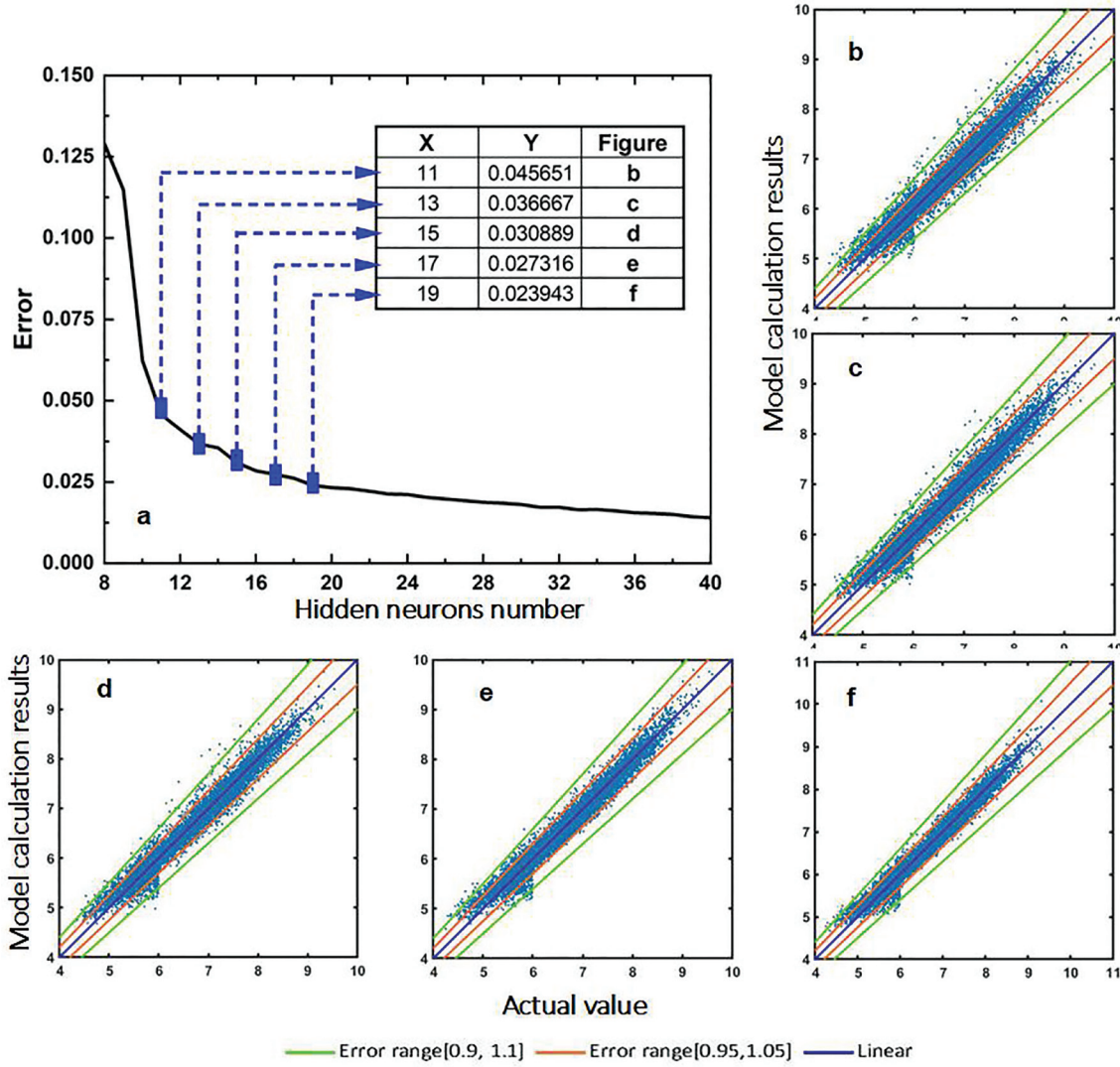
The formula of soil particle gradation is:

$$\delta_i = \frac{\sum_{j=1}^i q_j}{Q} \times 100\% \quad (7)$$

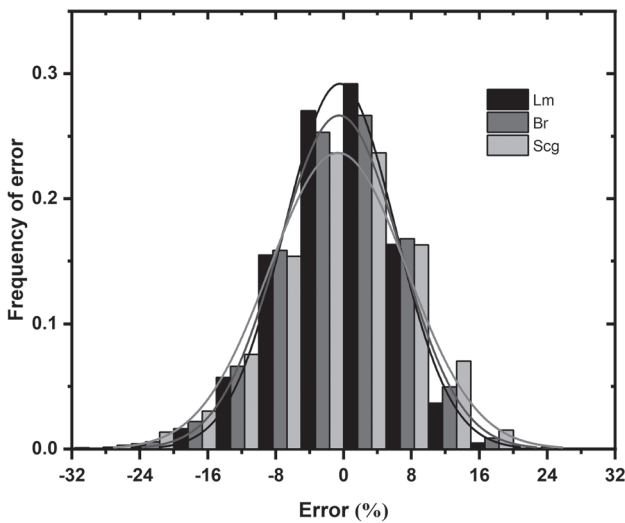
where  $\delta_i$  is the percentage of the cumulative mass of particles whose particle size is less than  $D_i$  in the total mass,  $q_j$  is the quality of the particle whose size is in

**Table 3** Particle size and error calculated by neural network models under three convergence modes of joint particles composed of four circle balls

| No. in database | The real value of particle size | 34-13-1-10 |         | 35-13-2-12 |         | 36-13-3-18 |         |
|-----------------|---------------------------------|------------|---------|------------|---------|------------|---------|
|                 |                                 | Lm         | Error % | Br         | Error % | Scg        | Error % |
| 10              | 6.75                            | 6.69       | 0.84    | 6.35       | 5.89    | 6.57       | 2.58    |
| 11              | 6.92                            | 6.90       | 0.26    | 6.39       | 7.58    | 6.87       | 0.62    |
| 12              | 9.38                            | 8.99       | 4.18    | 9.16       | 2.39    | 9.80       | 4.48    |
| 13              | 7.39                            | 7.41       | 0.25    | 7.72       | 4.38    | 7.89       | 6.77    |
| 14              | 7.23                            | 7.05       | 2.47    | 6.79       | 6.13    | 6.46       | 10.63   |
| 15              | 7.65                            | 7.64       | 0.07    | 7.41       | 3.11    | 7.52       | 1.63    |
| 16              | 6.02                            | 6.05       | 0.44    | 6.21       | 3.07    | 6.18       | 2.62    |
| 17              | 9.29                            | 8.61       | 7.30    | 8.87       | 4.49    | 7.92       | 14.69   |
| 18              | 7.79                            | 7.79       | 0.01    | 7.48       | 4.00    | 7.43       | 4.58    |
| 19              | 6.67                            | 6.70       | 0.36    | 6.91       | 3.54    | 6.62       | 0.86    |
| 20              | 6.86                            | 7.48       | 9.00    | 7.57       | 10.23   | 7.47       | 8.80    |



**Fig. 5** The curve of the mean square error and neuron number of joint particles composed of three circle balls: (a) The mean square error curve between the actual particle size and the particle size calculated by the neural network model with 8-40 neurons in the Lm convergence mode; (b), (c), (d), (e), and (f) The scatter diagram of actual particle size and particle size calculated by the neural network model obtained from 11, 13, 15, 17, and 19 neurons, where the range error between red lines is between [0.95, 1.05] and the range error between green lines is between [0.9, 1.1]

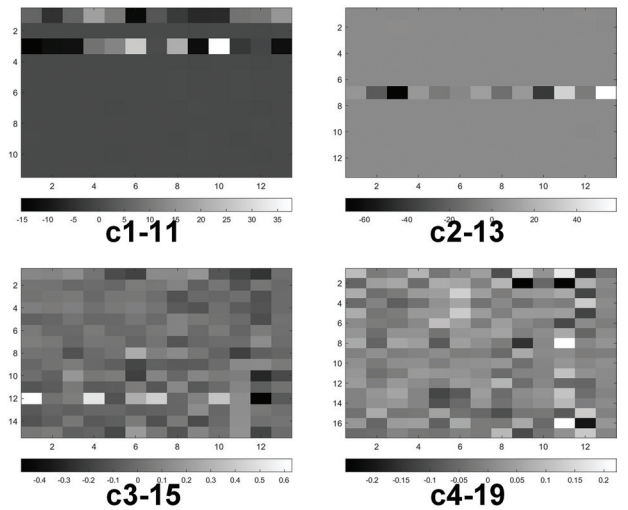
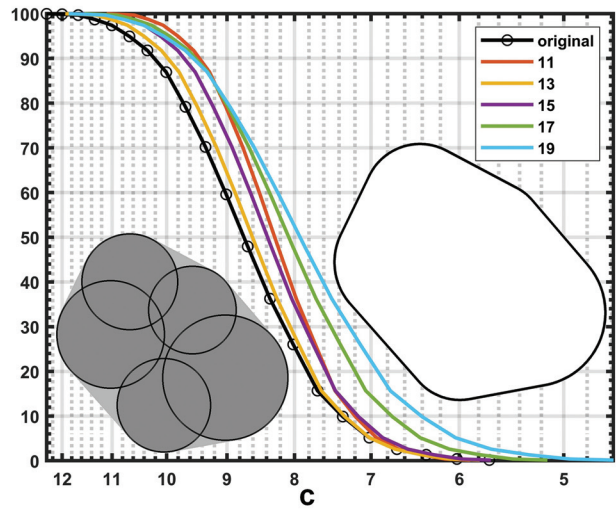
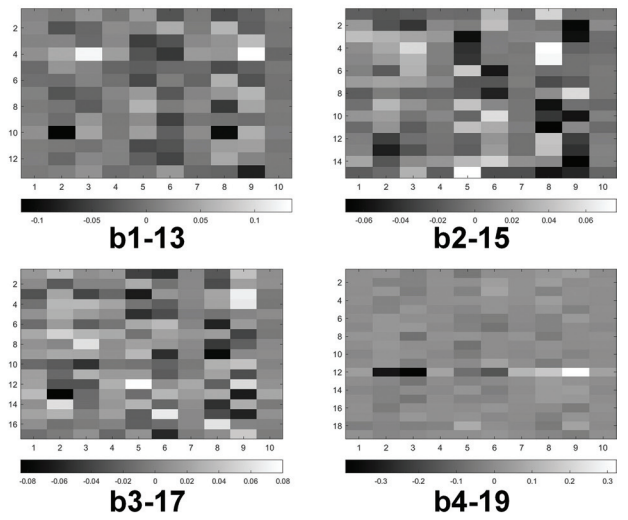
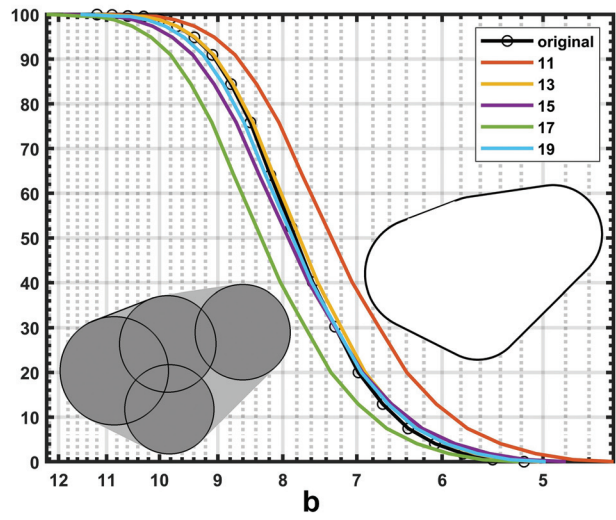
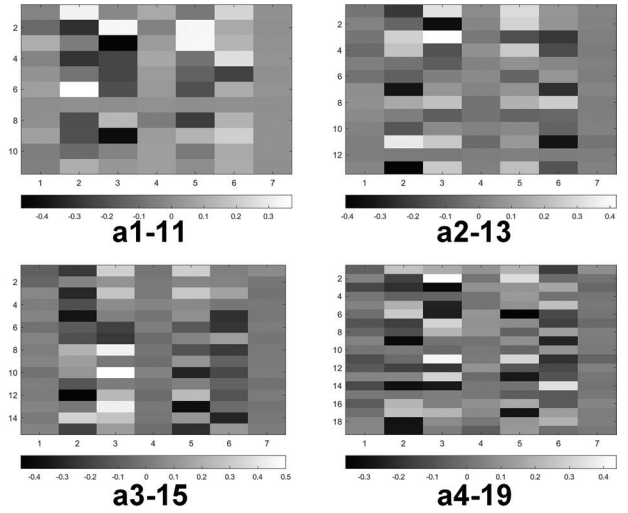
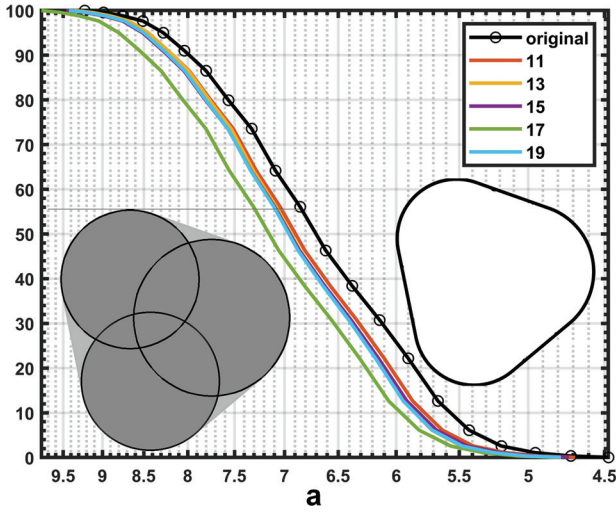


**Fig. 6** Error histograms normal distribution curve under three convergence modes

( $D_{j-1}$ ,  $D_j$ ], and  $Q$  is the total quality of the particle.

The calculation steps of the gradation curve are as follows: (1) 2000 joint particles are randomly selected from 10000 joint particles composed of 3-5 circle balls. Calculate the gradation according to Eq. (7). The particle size was the actual particle size. (2) In the selected 2000 joint particles, calculate the particle sizes of five training models with the neuron numbers 11, 13, 15, 17, and 19. (3) Using the 2000 particles' size calculated by the model, obtain the gradation according to Eq. (7). (4) Draw the two distribution curves calculated by the actual particle size and the calculated size from the neural network model, as shown in Fig. 7, and draw the model weights in the form of grey patches in a1-c4.

The following weight range is small: the 7th and 10th neurons in Fig. 7a1-11, the 5th and 12th neurons in Fig. 7a2-13, the 4th and 11th neurons in Fig. 7a3-15,



**Fig. 7** The gradation curve and grey patches of weight: (a), (b) and (c) are the grading curves of 2000 particles consisting of 3, 4, and 5 balls, respectively. The horizontal axis is the joint particle size. The vertical axis is the gravitation percentage (%). The curve of the factual size of 2000 joint particles is the ‘original’ calculated by the RCM model (Sun, Huang 2022). Other curves are the results of calculations using the JCM model (this research). Figure number explanation: xi-j is Figure No. of grey patches of model weight with j neurons. For example, a1-11 is the weight grey patches of the 3 circle balls model with 11 neurons, and b1-13 is the weight grey patches of the 4 circle balls model with 13 neurons. In the grey patches, the horizontal axis represents  $q$  in Eq.(3), and the vertical coordinate represents the number of neurons

the 4th and 15th neurons in Fig. 7a4-19, and the 1st and 11th neurons in Fig. 7b1-13. The neurons' weights in Fig. 7c2-13 are small except for the 7th neuron. Some hidden neurons have a short range of weights, indicating that the neurons have little effect on the results, but the specific meaning of the neurons is not exact.

The following weight range is enormous: the 2nd and 6th neurons in Fig. 7a1-11, the 3rd and 11th neurons in Fig. 7a2-13, the 4th and 10th neurons in Fig. 7b1-13, and the 7th neuron in Fig. 7c2-13. Similarly, the weight range of some neurons is extensive, indicating that these neurons have an essential significance, and the specific meaning represented by these neurons is not exact.

From the grey patches of weights, the colour difference of columns 1, 4, 7, 10, 13, and 16 is small, indicating that the weight is near 0, and the corresponding input data of this column is the radius of a single circle ball. The results show that the circle ball's radius has a lower effect on the joint particle size than the circle ball's centre position. With the increase of the number of circle balls, the weight shows a relatively significant change, shown as follows: the volatility increases with the absolute value of the weight increase, and the influence of the radius of the single circle on the radius of the joint particles also increases. Therefore, the number of circle balls cannot be too large, and between 3 and 5 is best.

From Fig. 7a, b, and c we can see that the points concentrated with the circle ball number increase, more concentrated in one same size. The gradation curve gradually becomes vertical. The coincidence degree between the joint particle gradation curve composed of four circle balls and the original is the best. The fitting degree of the model containing 13 neurons in four circle balls is the best, and the difference between the absolute values of the weights is also small.

Based on the above analysis, it can be concluded that the fitting degree of the model with 13 neurons and 4 circle balls is relatively good, and the difference between the absolute values of the weights is small.

### A neural network calculation model

The neural network calculation model with 4 circle balls and 13 neurons meets the requirements of a better and faster calculation for simulating joint particles. The weight of the neural network model is as follows, referring to Eqs. (2)–(5):

$$Y_{1 \times 1} = C_{1 \times 13} [\sigma (A_{13 \times 10} \times X_{10 \times 1} + B_{13 \times 1})] + D_{1 \times 1}$$

The four circles form a joint particle. Every circle has a centre  $(x, y)$  and a radius  $r$ . The first circle has a circle of  $(0,0)$  and is not used as input data. One joint particle has 10 raw numbers, and  $X$  is  $10 \times 1$ .

$$A = 0.0001 \times$$

|      |       |      |      |      |      |      |       |      |      |
|------|-------|------|------|------|------|------|-------|------|------|
| 144  | 32    | 281  | 14   | 334  | -329 | 30   | -394  | -193 | 44   |
| 184  | -338  | -135 | 206  | 127  | 519  | 73   | 617   | -220 | -4   |
| 125  | 145   | -114 | 179  | -538 | -339 | 189  | 280   | 691  | 2    |
| 248  | 509   | 1244 | 392  | -331 | -599 | 357  | 306   | 1343 | -134 |
| -101 | 307   | 145  | -151 | -231 | -108 | -92  | -407  | -189 | 13   |
| -56  | -76   | 127  | -48  | -436 | 68   | -153 | 562   | -336 | -28  |
| 108  | -315  | -152 | 162  | 349  | -293 | 151  | 312   | 616  | -40  |
| 52   | -3    | -245 | 98   | 592  | 50   | 210  | -592  | 469  | 55   |
| 117  | -84   | 304  | 35   | 206  | -513 | 0    | -38   | -76  | 36   |
| 245  | -1153 | 450  | 43   | 391  | -220 | 212  | -1033 | 548  | 80   |
| 75   | 22    | -25  | 121  | -412 | -326 | 83   | 385   | 414  | 6    |
| 91   | -133  | 173  | 109  | -210 | -473 | 13   | 485   | 141  | 38   |
| 42   | 99    | 267  | -79  | -8   | 334  | -126 | -168  | -816 | 32   |

$$B' = 0.0001 * [6200 \ 5776 \ 5682 \ 6180 \ -5926 \ -6164 \ 5931 \ 5962 \ 6189 \ 6484 \ 6077 \ 6123 \ 7300]$$

$$C = 0.0001 * [7786 \ 2829 \ 4654 \ -84 \ 8825 \ 9092 \ 4266 \ 4499 \ -8674 \ -78 \ -11379 \ 7065 \ -2706]$$

$$D = 570.6192$$

### Calculation time effect comparison

Use the established neural network model in this research, namely the JCM model, to calculate the size of the joint particles. The original calculation approach was the RCM model as proposed in the literature (Sun, Huang 2022). The number of spherical balls ranged from 3 to 12. The computation time of both methods was monitored. The number of neurons selected is shown in Fig. 7 as follows: 11 neurons are for the 3 circle balls model (Fig. 7a1-11), 13 neurons are for the 4 circle balls model, 11 neurons are for the 5 circle balls model (Fig. 7c1-11), 19 neurons are for the 6 circle balls model, and 13 neurons are for 7–12 circle balls model. Two methods are used to calculate the size of 2000 joint particles. The time taken is shown in Fig. 8. With the increase of circle balls, the calculation time of the original calculation method is gradually increasing.

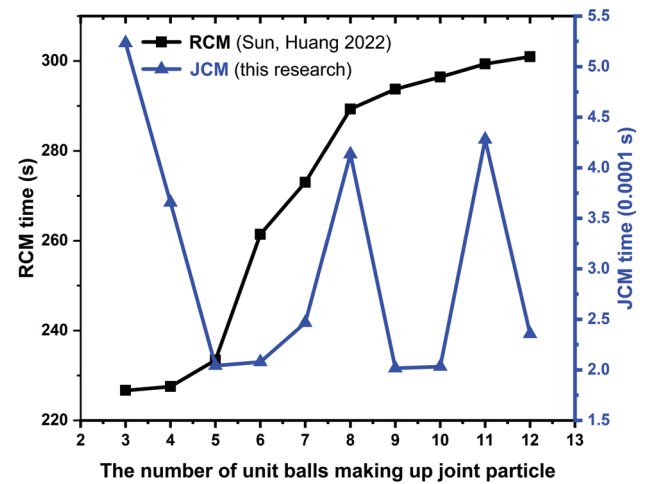


Fig. 8 The calculation time of 2000 joint particle sizes by neural network model method and direct calculation method

The calculation time of JCM is not directly related to the number of circle balls, and the time consumed is minimal and stable. The calculation time ratio of the two methods is [430000, 1450000]. The neural network model method has a good time efficiency.

## CONCLUSIONS

The neural network model with the 3–5 circle balls and 11 and 13 neurons can take very little time to obtain a better coincidence effect of the gradation curve. The neural network model with the best fitting degree is the model of 4 circle balls and 13 hidden neurons.

We established a joint particle size database of 3–12 circle balls. Simultaneously, the neural network calculation model with 4 circle balls and 13 neurons meets the requirements of a better and faster calculation for simulating joint particles.

The radius of circle balls has less influence on the particle size of joint particles than the centre of circle balls. With the increase of single-circle balls, the time consumed by the original calculation method of joint particles gradually increases. The calculation time of the neural network model is not affected by the number of circle balls and maintains a low constant value. The established database and neural network model can significantly reduce human resources in research and the power cost consumed by computer operations.

The specific meaning of hidden neurons with relatively large and small weights is not exact and is requiring further research.

## ACKNOWLEDGMENTS

The National Natural Science Foundation of China (No. 51179177) supported this work. We sincerely thank all the reviewers and editors for their professional and constructive revision suggestions and guidance. They have significantly enhanced the theoretical depth and argumentative rigor of this paper, helped the author further refine the research ideas, and will surely have a positive impact on future research. Here, we express our most earnest gratitude to you.

## REFERENCES

- Adesina, P., O'Sullivan, C., Wautier, A. 2025. The influence of particle size ratio on the mechanical behaviour of gap-graded granular assemblies under drained triaxial compression: A DEM study. *Computers and Geotechnics* 179, 19. <http://dx.doi.org/10.1016/j.compgeo.2024.106987>
- Adil, M., Ullah, R., Noor, S., Gohar, N. 2022. Effect of number of neurons and layers in an artificial neural network for generalized concrete mix design. *Neural Computing & Applications* 34, 8355–8363. <http://dx.doi.org/10.1007/s00521-020-05305-8>
- Bilgilioglu, H. 2023. A comparison of different machine learning models for landslide susceptibility mapping in Rize (Türkiye). *Baltica* 36(2), 115–132. <http://dx.doi.org/10.5200/baltica.2023.2.3>
- Carpenter, G.A., Grossberg, S. 1988. The art of adaptive pattern-recognition by a self-organizing neural network. *Computer* 21(3), 77–88. <http://dx.doi.org/10.1109/2.33>
- Coox, L., Atak, O., Vandepitte, D., Desmet, W. 2017. An isogeometric indirect boundary element method for solving acoustic problems in open-boundary domains. *Computer methods in applied mechanics and engineering* 316, 186–208. <http://dx.doi.org/10.1016/j.cma.2016.05.039>
- Fakher, M., Hosseini-Hashemi, S. 2020. On the vibration of nanobeams with consistent two-phase nonlocal strain gradient theory: exact solution and integral nonlocal finite-element model. *Engineering with computers* 38, 2361–2384. <http://dx.doi.org/10.1007/s00366-020-01206-5>
- Fakhimi, A. 2009. A hybrid discrete-finite element model for numerical simulation of geomaterials. *Computers and Geotechnics* 36(3), 386–395. <http://dx.doi.org/10.1016/j.compgeo.2008.05.004>
- Fatemeh, V.-N., Hamed, B., Mahdi, K. 2024. Effect of particle size ratio and fines content on drained/undrained behavior of binary granular soil according to confining pressure and packing density: a DEM simulation. *Computational Particle Mechanics* 11(1), 141–167. <http://dx.doi.org/10.1007/s40571-023-00614-w>
- Garcia-Archilla, B., Novo, J. 2020. Error analysis of fully discrete mixed finite element data assimilation schemes for the Navier-Stokes equations. *Advances in Computational Mathematics* 46, 61. <http://dx.doi.org/10.1007/s10444-020-09806-x>
- Ghodki, B.M., Patel, M., Namdeo, R., Carpenter, G. 2019. Calibration of discrete element model parameters: soybeans. *Computational Particle Mechanics* 6(1), 3–10. <http://dx.doi.org/10.1007/s40571-018-0194-7>
- Ham, Y.-G., Kim, J.-H., Luo, J.-J. 2019. Deep learning for multi-year ENSO forecasts. *Nature* 573 (7775), 568–572. <http://dx.doi.org/10.1038/s41586-019-1559-7>
- Izard, E., Hamouda, H.B., Voorde, J.V. 2020. High-stress impact-abrasion test by discrete element modeling. *Computational Particle Mechanics* 8, 1061–1073. <http://dx.doi.org/10.1007/s40571-020-00377-8>
- Jarrar, Z.A., Alshibli, K.A., Al-Raoush, R.I. 2020. Three-Dimensional Evaluation of Sand Particle Fracture Using Discrete-Element Method and Synchrotron Microcomputed Tomography Images. *Journal of Geotechnical and Geoenvironmental Engineering* 146, 06020007. [http://dx.doi.org/10.1061/\(asce\)gt.1943-5606.0002281](http://dx.doi.org/10.1061/(asce)gt.1943-5606.0002281)
- Karamut, R.O., Binal, A. 2024. Artificial neural networks-based ternary charts for predicting strength and frost heaving in mountain soils. *Baltica* 37(2), 151–169. <http://dx.doi.org/10.5200/baltica.2024.2.6>

- Khadse, C.B., Chaudhari, M.A., Borghate, V.B. 2017. Electromagnetic Compatibility Estimator Using Scaled Conjugate Gradient Backpropagation Based Artificial Neural Network. *Ieee Transactions on Industrial Informatics* 13(3), 1036–1045. <http://dx.doi.org/10.1109/tii.2016.2605623>
- Khatti, T., Naderi-Manesh, H., Kalantar, S.M. 2019. Application of ANN and RSM techniques for modeling electrospinning process of polycaprolactone. *Neural Computing & Applications* 31(1), 239–248. <http://dx.doi.org/10.1007/s00521-017-2996-6>
- Kishida, N., Nakamura, H., Ohsaki, S., Watano, S. 2025. Surrogate model of DEM simulation for binary-sized particle mixing and segregation. *Powder Technology* 455, 13. <http://dx.doi.org/10.1016/j.powtec.2025.120811>
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M. 2018. Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology And Earth System Sciences* 22(11), 6005–6022. <http://dx.doi.org/10.5194/hess-22-6005-2018>
- Moayedi, H., Aghel, B., Vaferi, B., Foong, L.K., Bui, D.T. 2020. The feasibility of Levenberg-Marquardt algorithm combined with imperialist competitive computational method predicting drag reduction in crude oil pipelines. *Journal of Petroleum Science and Engineering*, 185, 106634. <http://dx.doi.org/10.1016/j.petrol.2019.106634>
- Mohammad, Z.-S., Abouzar, S. 2025. A DEM study on the effects of specimen and particle sizes on direct simple shear tests. *Granular Matter* 27(2), 21. <http://dx.doi.org/10.1007/s10035-025-01513-y>
- Mohammadkhanifard, H., Zad, A.A. 2024. DEM modeling of dynamic anchors with particles of different shapes and sizes. *Marine Georesources & Geotechnology* 16, 1–16. <http://dx.doi.org/10.1080/1064119x.2024.2405176>
- Mousavi, S., Mostafa, E., William L., Zhu, W., Chuang, L.Y., Beroza, G.C. 2020. Earthquake transformer – an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature communications* 11(1), 3952. <http://dx.doi.org/10.1038/s41467-020-17591-w>
- Nataj, S., Lui, S.H. 2020. Superlinear convergence of non-linear conjugate gradient method and scaled memoryless BFGS method based on assumptions about the initial point. *Applied mathematics and computation* 369, 124829. <http://dx.doi.org/10.1016/j.amc.2019.124829>
- Nitka, M., Teichman, J. 2020. Comparative DEM calculations of fracture process in concrete considering real angular and artificial spherical aggregates. *Engineering fracture mechanics* 239, 1–19. <http://dx.doi.org/10.1016/j.engfracmech.2020.107309>
- Olimov, B., Karshiev, S., Jang, E., Din, S., Paul, A., Kim, J. 2020. Weight initialization based-rectified linear unit activation function to improve the performance of a convolutional neural network model. *Concurrency and Computation-Practice & Experience* 33, e6143. <http://dx.doi.org/10.1002/cpe.6143>
- Palade, V., Petrov, M.S., Todorov, T.D. 2020. Neural network approach for solving nonlocal boundary value problems. *Neural Computing & Applications* 32, 14153–14171. <http://dx.doi.org/10.1007/s00521-020-04810-0>
- Pinzolas, M., Toledo, A., Pedreno, J.L. 2006. A neighborhood-based enhancement of the Gauss-Newton Bayesian regularization training method. *Neural Computation* 18(8), 1987–2003. <http://dx.doi.org/10.1162/neco.2006.18.8.1987>
- Reinecke, S.R., Blahout, S., Zhang, Z., Rosemann, T., Hussong, J., Kruggel-Emden, H. 2023. Effects of particle size, particle density and Reynolds number on equilibrium streaks forming in a square wave serpentine microchannel: A DEM-LBM simulation study including experimental validation of the numerical framework. *Powder Technology* 427, 22. <http://dx.doi.org/10.1016/j.powtec.2023.118688>
- Rivas, A., Gonzalez-Briones, A., Hernandez, G.P., Javier, C.P. 2021. Artificial neural network analysis of the academic performance of students in virtual learning environments. *Neurocomputing* 423, 713–720. <http://dx.doi.org/10.1016/j.neucom.2020.02.125>
- Samalavicius, V., Vanhala, E. K.-M., Lekstutyte, I., Gadeikienė, S., Gadeikis, S., Žaržojus, G. 2024. Hydraulic conductivity determination of Lithuanian soils using machine learning. *Baltica* 37(2), 137–150. <http://dx.doi.org/10.5200/baltica.2024.2.5>
- Sun, J. 2016. Ground sediment transport model and numerical simulation. *Polish Journal of Environmental Studies* 25(4), 1691–1697. <http://dx.doi.org/10.15244/pjoes/60766>
- Sun, J. 2018. Permeability of particle soils under soil pressure. *Transport in Porous Media* 123(2), 257–270. <http://dx.doi.org/10.1007/s11242-018-1038-x>
- Sun, J., Huang, Y. 2022. Modeling the Simultaneous Effects of Particle Size and Porosity in Simulating Geo-Materials. *Materials* 15(4), 1576. <http://dx.doi.org/10.3390/ma15041576>
- Timurkutluk, B., Ciflik, Y., Sonugur, G., Altan, T., Genc, O., Colak, A.B. 2023. Microstructural design of solid oxide fuel cell electrodes by micro-modeling coupled with artificial neural network. *Powder Technology* 425, 118551. <http://dx.doi.org/10.1016/j.powtec.2023.118551>

## Database

The database is established by the author: Jichao Sun (jichao@email.com) Any researcher can download it from the website: [https://sunjc.eu.org/other/JCM\\_database/index.htm](https://sunjc.eu.org/other/JCM_database/index.htm)