



OPEN

Study of PLSR-BP model for stability assessment of loess slope based on particle swarm optimization

Bin Gong

The assessment of loess slope stability is a highly complex nonlinear problem. There are many factors that influence the stability of loess slopes. Some of them have the characteristic of uncertainty. Meanwhile, the relationship between different factors may be complicated. The existence of multiple correlation will affect the objectivity of stability analysis and prevent the model from making correct judgments. In this paper, the main factors affecting the stability of loess slopes are analyzed by means of the partial least-squares regression (PLSR). After that, two new synthesis variables with better interpretation to the dependent variables are extracted. By this way, the multicollinearity among variables is overcome preferably. Moreover, the BP neural network is further used to determine the nonlinear relationship between the new components and the slope safety factor. Then, a new improved BP model based on the partial least-squares regression, which is initialized by the particle swarm optimization (PSO) algorithm, is developed, i.e., the PLSR-BP model. The network with global convergence capability is simplified and more efficient. The test results of the model show satisfactory precision, which indicates that the model is feasible and effective for stability evaluation of loess slopes.

The slope stability analysis can not only provide basis for economical and reasonable slope design, but also help to make judgments about the stability state and evolution trend of no matter artificial or natural slopes, prevent the potential risks and guide the slope treatment. Due to the economic development and population expansion, the safety and security of the transport networks and residential areas may be threatened by the potential slope instabilities in many countries. Slope instabilities are complex natural hazards that may result in disastrous consequences¹. Therefore, the slope stability assessment is a critical research area in civil engineering². In order to ensure the safety of economic construction and prevent the potential economic losses and casualties, slope stability analyses are required, and appropriate assessment methods are of practical need. Currently, the expert evaluation, analytical methods and machine learning are three common methods employed for the slope stability analysis^{3,4}. The first method is mainly used to analyze the reasons and developing processes of slope deformation according to the expertise and engineering geological survey. The essence is to apply the previous practical experience into the similar slope engineering projects⁵. Based on the experts' experiences and knowledge, the relative factors which may trigger the slope collapses can be identified and the safety and stability of a slope can be evaluated. However, the major disadvantage of the expert evaluation techniques is the subjectivity and the decisions may contain the bias of researchers⁶. The analytical methods are mainly used to analyze the slope system characteristics by establishing appropriate mathematical models. Based on this approach, the dangerous sliding surface and safety factor can be identified. However, it is actually difficult to determine the calculation parameters accurately, which may lead to misleading results. In fact, this kind of methods are only appropriate for evaluating slope stability in small areas⁷. Recently, based on the intelligent statistical learning theory, machine learning has been introduced into the slope collapse prediction. The machine learning models are generally established on the basis of the artificial intelligence techniques and historical data⁸. Yan and Li⁹ built a prediction model to evaluate the stability of an open pit slope based on the Bayes discriminant analysis (BDA). Samui and Kothari¹⁰ applied the least squares support vector machine (LSSVM) to explore the mapping function between the input pattern and the safety factor of slopes. Zhao et al.¹¹ developed the nonlinear relationship between the slope stability and

¹State Key Laboratory of Coastal and Offshore Engineering, Dalian University of Technology, Dalian 116024, China. ²Department of Civil and Environmental Engineering, Brunel University London, London UB8 3PH, UK. email: gongbin@dlut.edu.cn

influence factors using the relevance vector machine (RVM). Wang et al.¹² constructed a method to evaluate the stability of complex slope systems based on the projection pursuit algorithm. Liu et al.¹³ applied the improved particle swarm optimization (PSO) algorithm to analyze some critical factors affecting saturated rock slope slip in numerical simulation. Himanshu et al.¹⁴ used the unified particle swarm optimization (UPSPO) to assess optimum location of non-circular failure surface in soil slope. And Moayed et al.¹⁵ compared the feasibility of the artificial neural network (ANN), adaptive neuro-fuzzy inference system (ANFIS), and hybrid particle swarm optimization (HPSO) for assessing the safety factor of cohesive slopes.

In terms of loess slope stability evaluation, many internal and external factors should be considered. However, some of them show the obvious features of fuzziness, randomness and variability. Simultaneously, there is a complex nonlinear relationship between the evaluation indices and the influencing factors, which cannot be described by simple mathematical formula. Therefore, the stability assessment of loess slopes is a dynamic, nonlinear, uncertain and systematic problem. Besides, the correlation may exist between different parameters of geomaterials. Actually, it is impossible to take the effects of all the influencing factors into account fully and properly. However, a multi-variable system containing some main factors will be affected by the overlapping information inevitably because the multicollinearity among variables will exaggerate certain characteristics of the analysis system, which will definitely affect the objectivity and chock the decision process¹⁶.

The artificial neural network (ANN) has been successfully used to solve the slope stability problems by many scholars^{17–20} in recent years. Among them, the back propagation (BP) neural network is a kind of widely used neural networks. BP neural network shows good performance in knowledge learning, experience storage, computational efficiency and fault tolerance. It has the ability to extract features and acquire knowledge from the dynamic uncertain multi-factor systems, and also to approximate any complex nonlinear functional relationship. Meanwhile, in consideration of the explicit error back-propagation strategy and strict weight correction procedure from a mathematical point of view, it is reasonable to evaluate the slope stability with the BP neural network. Partial least-squares regression (PLSR) is a new multivariate statistical data-analysis method which can realize the multiple linear regression, canonical correlation analysis and principal component analysis. Through extracting the representative synthesis variables of a given system, PLSR can reduce the dimension of the independent variable system, simplify the network structure and improve the modeling efficiency. Meanwhile, the adverse effects of the variable multicollinearity can be overcome in this way. Moreover, many evolutionary algorithms were proposed for global optimization and employed to improve the performance of other algorithms, such as the dragonfly algorithm²¹, multi verse optimizer²², robust optimization²³ and cooperative meta-heuristic algorithm²⁴. As a population-based global optimization technique, the particle swarm optimization (PSO) has arisen extensive attention from the optimization community because of the simple structure, clear parameter meaning, high convergence rate and little manual intervention. Hence, PSO is adopted to solve the local convergence problem of BP neural network using its excellent global optimization capability, which means the satisfactory global optimal solution can be obtained with a relatively high speed. In this study, by taking advantage of the partial least-squares regression, particle swarm optimization and BP neural network, the PLSR-BP model for loess slope stability assessment is established. And the test results of the model show satisfactory precision and performance.

Partial least-squares regression

The partial least-squares regression (PLSR)²⁵, as the second-generation regression analysis method, was developed for the global data treatment. PLSR can be employed to find the hidden structure of the dataset and extract the meaningful information based on the dimensional reduction of data and inverse calibration technology²⁶. Through PLSR, several new representative synthesis variables can be extracted from the variable system by removing the redundant information. By this way, the adverse effects of multicollinearity among variables on the accuracy and reliability of modeling can be overcome effectively²⁷.

Extract principal components using PLSR. Hypothetically, there are p independent variables $r = \{x_1, x_2, \dots, x_p\}$ and q dependent variables $s = \{y_1, y_2, \dots, y_q\}$ (where p and q are two positive integers). There are n groups of independent and dependent variable data, respectively ($X = [r_1, r_2, \dots, r_n]_{n \times p}^T$ and $Y = [s_1, s_2, \dots, s_n]_{n \times q}^T$). n is the number of the selected samples. t_i and u_i are extracted from X and Y , respectively. Namely, t_i is a linear combination of x_1, x_2, \dots, x_p and u_i is a linear combination of y_1, y_2, \dots, y_q . Meanwhile, the following requirements must be met: (1) t_i and u_i should carry the variation information of their own data table as much as possible and (2) the correlation degree between t_i and u_i should be highest. After extracting the first principal components (t_1 and u_1), the partial least-squares regression will be carried out further to get the regression relations of X and t_1 as well as Y and u_1 . The algorithm will be terminated when the precision of the regression equation is considered to be satisfactory by testing. Otherwise, the residual information of X and Y after being explained with t_1 and u_1 will be further used to extract the second components (t_2 and u_2). This process repeats until a satisfactory precision is reached.

Determine principal components using the cross validation (CV). The principal components of a variable system can be determined by judging whether the prediction ability of the model will be improved significantly when adding the components. The cross-validation method is usually used for the determination of the principal components.

Hypothetically, y_j represents the sample data and t_1, t_2, \dots, t_A are the principal components extracted by PLSR. \hat{y}_{hij} is the fitted value of y_j at the i th sample point using the regression model established with h principal components (t_1, t_2, \dots, t_h). These principal components are extracted using all the sample points. $\hat{y}_{h(-i)}$ is the fitted value of y_j at the i th sample point using the regression model established with h principal components ($t'_1, t'_2, \dots,$

t_h^2). These principal components are extracted using all the sample points except the i th sample point. Moreover, the sum of squared errors ss_{hj} of y_j and the sum of squared prediction errors $press_{hj}$ of y_j are defined as follows²⁸:

$$ss_{hj} = \sum_{i=1}^n (y_{ij} - \hat{y}_{hji})^2 \tag{1}$$

$$press_{hj} = \sum_{i=1}^n (y_{ij} - \hat{y}_{hj(-i)})^2 \tag{2}$$

Furthermore, the sum of squared errors ss_h of Y which is described by h principal components extracted from all the sample points and the sum of squared prediction errors $press_h$ of Y are defined as follows²⁸:

$$ss_h = \sum_{j=1}^q ss_{hj} \tag{3}$$

$$press_h = \sum_{j=1}^q press_{hj} \tag{4}$$

In general, $press_h > ss_h$ and $ss_h < ss_{h-1}$. ss_{h-1} is the sum of squared errors of Y which is described by $h-1$ principal components extracted from all the sample points. Compared with ss_{h-1} , $press_h$ reflects not only the role of the principal component t_h , but also the disturbance error of the sample data. Hence, it is always expected that the value of $press_h$ could be smaller than ss_{h-1} to a certain extent (*i.e.*, the value of $press_h/ss_{h-1}$ is considered to be the smaller the better). Thus, the cross validation of the principal component t_h can be defined as²⁸:

$$Q_h^2 = 1 - press_h/ss_{h-1} \tag{5}$$

Typically, when $Q_h^2 \geq 0.0975$, the addition of the principal component t_h will benefit the system; otherwise, there is no need to add the principal component t_h .

Extraction algorithm of PLSR. The extraction algorithm of PLSR can be summarized as follows:

(1) Complete the standardization process.

The standardization formula is shown as follows²⁸:

$$\hat{z}_{ij} = (z_{ij} - \bar{z}_j) / sd_j \tag{6}$$

where \hat{z}_{ij} is the standardized value, z_{ij} is the real value, and \bar{z}_j and sd_j are the arithmetic mean and standard deviation of the data in the j th column of the data matrix, respectively.

According to Eq. (6), the standardized data matrices of X and Y can be obtained and expressed as $E_0 = [E_{01}, E_{02}, \dots, E_{0p}]_{n \times p}$ and $F_0 = [F_{01}, F_{02}, \dots, F_{0q}]_{n \times q}$. Hypothetically, t_1 and u_1 are the first principal components of E_0 and F_0 , respectively.

(2) Extract the principal components.

Calculate the unit eigenvector w_1 corresponding to the largest eigenvalue of the covariance matrix $E_0^T F_0 F_0^T E_0$. Note that w_1 is the first axis of E_0 and as a unit eigenvector, *i.e.*, $\|w_1\| = 1$. Simultaneously, calculate the unit eigenvector c_1 corresponding to the largest eigenvalue of the covariance matrix $F_0^T E_0 E_0^T F_0$. Note that c_1 is the first axis of F_0 and as a unit eigenvector, *i.e.*, $\|c_1\| = 1$. After determining the vectors w_1 and c_1 , the first principal components can be obtained as $t_1 = E_0 w_1$ and $u_1 = F_0 c_1$. After that, the two regression equations of E_0 and F_0 about t_1 can be determined, respectively²⁸.

$$E_0 = t_1 a_1^T + E_1 \tag{7}$$

$$F_0 = t_1 b_1^T + F_1 \tag{8}$$

where the regression coefficient vectors are as follows²⁸:

$$a_1 = E_0^T t_1 / \|t_1\|^2 \tag{9}$$

$$b_1 = F_0^T t_1 / \|t_1\|^2 \tag{10}$$

(3) Test the cross validation.

If $Q_h^2 \geq 0.0975$, it means that the next principal components should be extracted. After replacing the residuals matrices E_0 and F_0 with E_1 and F_1 , the second principal components t_2 and u_2 can be calculated in the same way. This process will repeat until $Q_h^2 < 0.0975$. If the rank of X is A , we have the following equations²⁸:

$$E_0 = t_1 a_1^T + \dots + t_A a_A^T + E_A \tag{11}$$

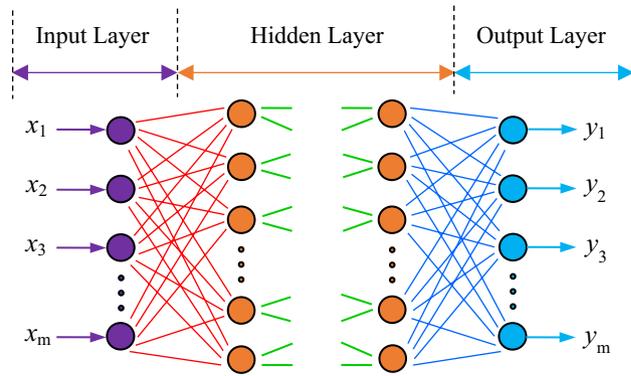


Figure 1. Typical architecture of BP neural network (MS Office 2019; www.microsoft.com/en-us/microsoft-365/get-started-with-office-2019).

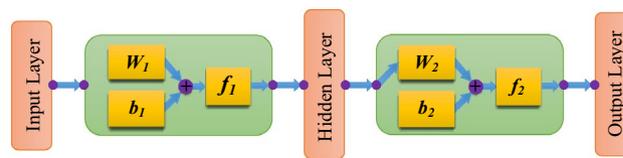


Figure 2. Flow chart of BP neural network (MS Office 2019; www.microsoft.com/en-us/microsoft-365/get-started-with-office-2019).

$$F_0 = t_1 b_1^T + \dots + t_A b_A^T + F_A \quad (12)$$

BP neural network

Artificial neural network²⁹ can be seen as a connected parallel architecture consisting of several layers of neurons. For ANN, the knowledge can be gained from the sample sets and be represented as ‘weights’ and ‘thresholds’ in the connections of the neural network. Through the weight and threshold matrices, the influence of the input variables on the output variables can be determined. On the other hand, the appropriate mathematical methods can be chosen to adjust the weights and thresholds to realize specific functions. The BP neural network is used in this study because of its reliability and applicability. According to the randomly initialized weight and threshold matrices, the error between the network output and the target values can be calculated. Then, based on a weight correction procedure, the error is propagated backward and used to update the weight and threshold matrices of the previous layers using the back-propagation algorithm³⁰. In this way, the mapping function between the system input variables and output variables can be modelled by the BP neural network step by step.

The architecture of a standard BP neural network is shown in Fig. 1. Generally, it has one input layer composed of neurons corresponding to the input variables, no less than one hidden layer and one output layer composed of neurons corresponding to the output variables.

In this study, the number of neurons (m) in the input layer is the same as the number of physical–mechanical parameters to be considered, and the number of neurons (n) in the output layer is the same as the number of evaluation indices. The evaluation index of the slope stability is the safety factor in this paper. Hence, the number n is 1. The number of neurons (p) in a hidden layer can be specified either manually or by an optimization method³⁰. The training samples are used to update the weight and threshold matrices by making the summed squared error between the safety factor values and the output of the BP network a minimum using the back propagation algorithm.

The computing process of a three-layer BP neural network is shown in Fig. 2. W_1 and b_1 are the weight and threshold matrices between the input and hidden layers, respectively; W_2 and b_2 are the weight and threshold matrices between the hidden and output layers, respectively; f_1 and f_2 are the transfer functions between two adjacent layers. Tan-Sigmoid transfer function (*tansig*), Log-Sigmoid transfer function (*logsig*) and linear transfer function (*purelin*) are the three common transfer functions for multilayer artificial neural networks.

However, the summed squared error between the target values and the output values of the BP network depends on the randomly initialized weight and threshold matrices. The limitations such as slow convergence, local convergence and poor generalization ability hamper the performance of ANN seriously³¹. Therefore, an appropriate optimization algorithm with global optimization capability is necessary for the initial assignment of the weights and thresholds of BP neural network.

Particle swarm optimization

The particle swarm optimization algorithm (PSO) was proposed by Eberhart and Kennedy³². As a new intelligent swarm optimization algorithm, PSO is an important branch of the evolutionary algorithms. The velocity-position model is applied in PSO. The position of each particle represents a candidate solution in the solution space. The quality of the particle solution is measured by the previously defined fitness function.

The position and velocity of Particle i in the n -dimensional space can be set as $\mathbf{o}_i = \{o_{i1}, o_{i2}, \dots, o_{in}\}$ and $\mathbf{v}_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}$. PSO searches the optimal solution through iterating. Firstly, a group of particles are initialized randomly in the n -dimensional space. The velocity decides the displacement of a particle during one iteration in the solution space. Then, the particle velocity and position are adjusted dynamically according to the individual extremum $\mathbf{p}_i = \{p_{i1}, p_{i2}, \dots, p_{in}\}$ and global extremum $\mathbf{g} = \{g_1, g_2, \dots, g_n\}$ using the following formulae³²:

$$v_{ij}^{k+1} = wv_{ij}^k + c_1r_1(p_{ij}^k - o_{ij}^k) + c_2r_2(g_j^k - o_{ij}^k) \quad (13)$$

$$o_{ij}^{k+1} = o_{ij}^k + v_{ij}^{k+1} \quad (14)$$

where w is the inertia weight coefficient, c_1 and c_2 are the learning factors, r_1 and r_2 are the random numbers between (0, 1), v_{ij}^k and o_{ij}^k are the j th components of the velocity vector and position vector of Particle i in the k th epoch, where $j = 1, 2, \dots, n$ (n is the dimension of the solution space). The first item of the velocity-updating formula reflects the inheritance of the previous velocity, which makes particles maintain inertial motion; the second item is usually called the cognition term. This item is only related to the particle's own experience and reflects the thinking on behalf of itself; the third item is usually called the social item, which reflects the information sharing and cooperation between particles. By learning from itself and other particles, the particle is targeted to obtain more effective information from its ancestors. This process ensures that the optimal solution can be obtained in a short time³³.

The PLSR-BP model for loess slope stability assessment

The stability of loess slopes depends on a lot of internal and external factors involving the soil properties, structure characteristics, groundwater, climate change, weathering effects, seismicity, human activities and so on. The key of assessing the loess slope stability appropriately is to select the influence factors correctly. However, there are no widely accepted theories guiding the selection so far. A common approach is to analyze the practical conditions as well as refer to the experience of the geological experts and engineers. On the basis of analyzing the previous study³⁴ and data availability comprehensively, seven parameters are determined as the independent variables affecting the loess slope stability in this study, i.e., density γ (x_1), cohesion c (x_2), internal friction angle ϕ (x_3), slope height h (x_4), slope ratio s (x_5), pore water pressure coefficient γ_u (x_6) and seismic intensity q (x_7). Because the main means analyzing the slope stability quantitatively is to calculate the safety factor, the safety factor y_1 is selected as the dependent variable. The 23 representative loess slopes in Northwest China described in the literature³⁴ are selected as the analysis samples, as shown in Table 1.

Correlation between independent variables. The correlation analysis is carried out on MATLAB platform and the correlation coefficients between different independent variables are shown in Table 2.

It can be seen in Table 2 that the multicollinearity exists among independent variables. Especially, the linear correlations between some variables, such as x_1 , x_2 , x_3 , and x_4 are significant. Thus, it is necessary to overcome the multicollinearity by extracting principal components using the partial least-squares regression.

Extract principal components by PLSR. The partial least-squares regression is implemented for the system $y_1 = f(x_1, x_2, x_3, x_4, x_5, x_6, x_7)$ on MATLAB platform. The cross validation shown in Table 3 demonstrates that the first two principal components t_1 and t_2 are acceptable. The mathematical expression is shown as follows:

$$\begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} -0.0492 & 0.0542 \\ 0.0567 & -0.0260 \\ 0.0599 & 0.0337 \\ 0.0117 & -0.1203 \\ 0.0438 & 0.0266 \\ -0.0566 & -0.0759 \\ -0.0508 & -0.1117 \end{bmatrix}^T \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \hat{x}_3 \\ \hat{x}_4 \\ \hat{x}_5 \\ \hat{x}_6 \\ \hat{x}_7 \end{bmatrix} \quad (15)$$

where $[\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7]^T$ is the standardized matrix of $[x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T$.

PLSR-BP model based on the particle swarm optimization algorithm. According to the cross-validation testing, a stability assessment model with good performance can be obtained by only choosing the first two principal components t_1 and t_2 . Actually, if the following principal components could not offer more meaningful information for explaining the variable system \mathbf{Y} , choosing too many principal components may mislead the understanding about the statistical trend and result in incorrect prediction conclusions.

In this study, the three-layer BP neural network is applied. The two principal components t_1 and t_2 are treated as the input of the neural network, and \hat{Y} (the standardized value of \mathbf{Y}) is treated as the output of the neural network, i.e., there are two neurons at the input layer and one neuron at the output layer. It has been proved that a three-layer BP neural network with M neurons at the input layer, $2M + 1$ neurons at the hidden layer and N

Serial Num	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_1
1	17.3	105.8	27.1	81.3	3.2	-1	0	2.271
2	17.9	72.1	24.1	93.2	3	-1	7	1.522
3	18.1	63.7	27.5	41.8	2.5	-1	8	1.657
4	18.7	42.6	22.8	49.7	2.8	-1	8	1.284
5	18.4	51.1	23.3	34	1.3	0	0	1.445
6	17.3	110	22.8	85.3	2.1	0	7	1.743
7	18.1	63.7	25	81.3	2	0	8	1.384
8	17.5	93.2	27.5	77.4	2.5	0	9	1.611
9	17.7	80.5	24.5	105	2.5	0.25	0	1.519
10	17.7	84.7	24.5	77.4	2	0.25	7	1.379
11	18.3	55.3	24.5	73.4	1.6	0.25	8	1.002
12	18	67.8	24.1	93.2	1.6	0.25	9	0.842
13	17.4	101.6	29.2	65.5	1.6	0.5	0	1.301
14	18.4	51.1	24.5	34	2	0.5	7	1.153
15	18.2	59.5	22.8	49.7	2.3	0.5	8	0.997
16	18.1	63.7	22.4	73.4	2.5	0.5	9	0.77
17	17.5	89.6	25	91.3	1.6	-1	0	1.28
18	18.7	42.6	22	53.7	1.3	0	8	0.99
19	18.3	70.3	25.7	63.4	2.8	0.25	0	1.59
20	17.3	105.8	28.9	81.3	3.2	-1	9	1.58
21	18.5	51.1	27.3	34	1.3	0	0	1.73
22	17.7	81.7	24.5	77.4	2	0.25	7	1.36
23	18.2	59.5	22.8	49.8	2.3	0.5	8	0.99

Table 1. Sample data of loess slopes (Gao et al.³⁴). $x_1, x_2, x_3, x_4, x_5, x_6, x_7$ and y_1 are the density, cohesion, internal friction angle, slope height, slope ratio, pore water pressure coefficient, seismic intensity and safety factor, respectively.

Variable	x_1	x_2	x_3	x_4	x_5	x_6	x_7
x_1	1	-0.979**	-0.492*	-0.673**	-0.325	0.211	0.144
x_2		1	0.521*	0.627**	0.355	-0.213	-0.195
x_3			1	0.058	0.200	-0.291	-0.314
x_4				1	0.288	-0.163	-0.025
x_5					1	-0.457*	0.166
x_6						1	0.058
x_7							1

Table 2. Correlation coefficients between independent variables. The sign ‘*’ or ‘**’ indicates that the correlation is significant at 0.05 level or 0.01 level.

Component	t_1	t_2	t_3
Q_{fi}^2	0.554	0.110	0.015
Threshold	0.0975	0.0975	0.0975

Table 3. Cross validation.

neurons at the hidden layer can express any continuous function accurately³⁵. Hence, the structure of the neural network is set to be 2–5–1. Simultaneously, the *logsig* function is applied as the transfer function of the hidden layer and the *purelin* function is applied as the transfer function of the output layer.

Based on MATLAB platform, the neural network is established by learning knowledge from the 23 groups of sample data listed in Table 1. The initial weights and thresholds are optimized using PSO. Through trial calculation, the parameters of PSO are set as: the total number of particles $n = 23$, particle dimension $d = 21$, inertia weight coefficient w decreasing from 1.15 to 0.45 linearly, learning factors $c_1 = 2.2$ & $c_2 = 2.0$ and maximum evolution number $m = 200$. At the end of evolution, the mean square error (MSE) of the network drops to 0.0787. Then,

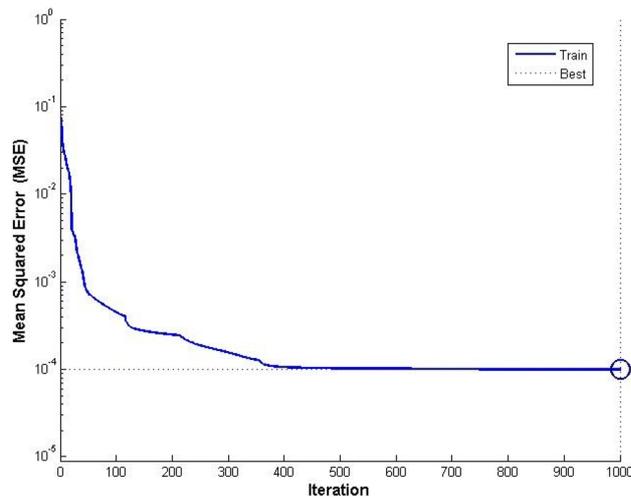


Figure 3. Training process of the neural network.

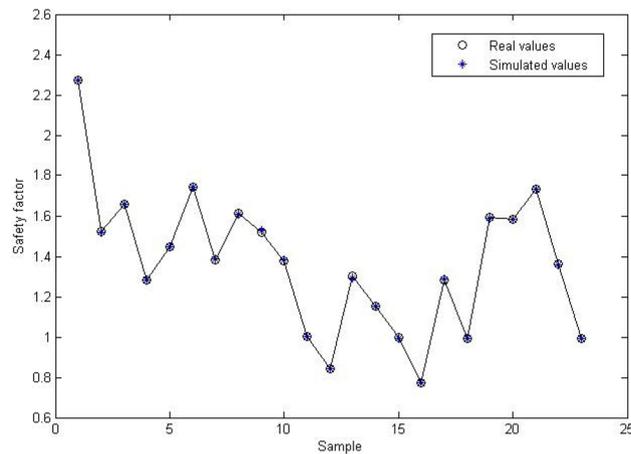


Figure 4. Simulated results of the neural network.

the neural network is initialized with the optimal particle position and trained with the Levenberg-Marquard algorithm. During this process, the training target is 1×10^{-5} , maximum iteration number is 1000, learning rate is 0.05, and display interval is 200. At the end of training, the final MSE of the neural network is 9.9667×10^{-5} . The training process and results are displayed in Figs. 3 and 4. Figure 3 shows that the MSE gradually drops from around 0.8×10^{-2} to around 1×10^{-4} after 400 iterations and reaches 9.9667×10^{-5} after 1000 iterations, which indicates that the convergence is steady and fast. The total runtime is 30.50893 s on the computer with a i7-10510U CPU and 16 GB RAM. The time complexity of the code is $O(m)$, in which m is the maximum evolution number of PSO. From Fig. 4, we can see that the simulated results of the trained BP neural network almost coincide with the real values, which indicates that the established analysis model can precisely describe the complex nonlinear relationship between the influencing factors and the safety factor and successfully capture the main features of the loess slope stability evaluation system.

Then, four new loess slope samples are used to verify the precision of the established neural network model and the related parameter values are listed in Table 4. Meanwhile, Table 5 shows the comparison of the calculated safety factors by the proposed PLSR-BP model, the PSO-BP model without the partial least-squares regression analysis and the traditional BP neural network. From the evaluation results shown in Table 5, it can be seen that the forecasting model has been significantly improved by extracting the new synthesis variables and overcoming the multicollinearity among variables, and the performance of the established PLSR-BP model is obviously superior to the other two models. Moreover, the maximum absolute error of the output safety factor of the PLSR-BP model is less than 0.040 and the relative error never exceeds 5.0%. The high precision further indicates that the established PLSR-BP model based on the partial least-squares regression is feasible and reliable. Simultaneously, from Table 5, we can see that the relative error of Sample 2 predicted by the PLSR-BP model is largest among all the test samples. That is because the real safety factor of this slope is only 0.790 which is a value lower than 95% of the training samples. Namely, such a low safety factor is not common, and the proposed model hasn't learned much knowledge about low safety factor because of data availability.

Serial num	Density /(kN/m ³)	Cohesion /kPa	Internal friction angle /($^{\circ}$)	Slope height/m	Slope ratio	Pore water pressure coefficient	Seismic intensity	Safety factor
1	17.91	71.35	22.17	52	1.0	0.0	0	1.229
2	17.94	70.13	22.16	73.42	2.48	0.5	9	0.790
3	16.5	100.0	29	45	0.7	0.0	0	1.127
4	18.6	93.0	20	13.2	3.0	0.2	0	1.567

Table 4. Parameters of four new loess slope samples.

Serial num	Safety factor	BP			PSO-BP			PLSR-BP		
		Calculated value	Absolute error	Relative error/%	Calculated value	Absolute error	Relative error/%	Calculated value	Absolute error	Relative error/%
1	1.229	1.358	0.129	10.50	1.141	0.088	7.16	1.220	0.009	0.73
2	0.790	1.055	0.265	33.54	0.819	0.029	3.67	0.824	0.034	4.30
3	1.127	1.422	0.295	26.18	1.057	0.070	6.21	1.152	0.025	2.22
4	1.567	1.111	0.456	29.10	1.460	0.107	6.83	1.571	0.004	0.26

Table 5. Comparison between network outputs and practical values.

Conclusion

The assessment of loess slope stability is a highly complex nonlinear problem. Some of the factors affecting the slope stability exhibit the characteristics of fuzziness, randomness and variability. Meanwhile, there is a complex nonlinear relationship between the influencing factors and the safety factor. In this study, by taking advantage of the artificial neural network and intelligent swarm optimization algorithm, the improved BP model for the stability assessment of loess slopes is developed based on the partial least-squares regression, i.e., the PLSR-BP model.

Although the stability assessment of loess slopes is a dynamic, nonlinear, uncertain and systematic problem, it has been proved that the BP neural network has the ability to approach the complex nonlinear relationship. It is appropriate and effective to evaluate the loess slope safety and stability using the BP neural network. Moreover, this study focuses on the multicollinearity problem. The correlation analysis indicates that the multicollinearity exists in the variable system. The existence of multiple correlation will affect the objectivity of stability analysis and prevent the model from making correct judgments. Therefore, the partial least-squares regression is carried out and two new synthesis variables with better interpretation to the dependent variables are extracted. In this way, the adverse effects of the variable multicollinearity are overcome. Simultaneously, the neurons at the input layer of BP neural network are also reduced to two, which simplifies the network structure and improves the modeling efficiency. Additionally, with the aim of converging to the global optimal solution more quickly, the BP neural network is initialized by PSO because of its global optimization ability. The test results show satisfactory precision, which indicates that the proposed model is feasible and reliable for the stability evaluation of loess slopes.

Combining the advantages of the particle swarm optimization, BP neural network and partial least-squares regression, the proposed assessment model can not only tackle with the variable correlation, local convergence and nonlinearity problems, but also present more extensive applicability. It can be used to determine the stability state and calculate the safety factor of loess slopes. Meanwhile, more influencing factors, such as rainfall density, groundwater level, weathering degree of geomaterials, etc., can be considered in the developed model based on data availability to conduct parameter sensitivity analysis and form a specific model reflecting the actual situation in a certain area. However, for such a specific model, the quality of the training samples may affect the effectiveness of the established model, i.e., the accurate parameter values should be ensured for the samples. Additionally, the developed model cannot give satisfactory results if the parameter values of an evaluated slope exceed the parameter ranges of the training sample. Although the loess slope stability assessment involves many aspects and a few challenges may be encountered, the proposed model has proved to be an effective and efficient approach for engineers in the field of loess slopes, and it shows potential in a variety of slope engineering applications.

Data availability

The datasets generated and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Received: 16 February 2021; Accepted: 23 August 2021

Published online: 09 September 2021

References

- Lu, P. & Rosenbaum, M. S. Artificial neural networks and grey systems for the prediction of slope stability. *Nat. Hazards* **30**(3), 383–398 (2003).

2. Hoang, N. D. & Pham, A. D. Hybrid artificial intelligence approach based on metaheuristic and machine learning for slope stability assessment: A multinational data analysis. *Expert Syst. Appl.* **46**, 60–68 (2016).
3. Ahangar-Asr, A., Faramarzi, A. & Javadi, A. A new approach for prediction of the stability of soil and rock slopes. *Eng. Comput.* **27**(7), 878–893 (2010).
4. Leshchinsky, D. Slope stability analysis: Generalized approach. *J. Geotech. Eng.* **116**(5), 851–867 (1990).
5. Raghuvanshi, T. K., Ibrahim, J. & Ayalew, D. Slope stability susceptibility evaluation parameter (SSEP) rating scheme—An approach for landslide hazard zonation. *J. Afr. Earth Sci.* **99**, 595–612 (2014).
6. Fall, M., Azzam, R. & Noubactep, C. A multi-method approach to study the stability of natural slopes and landslide susceptibility mapping. *Eng. Geol.* **82**(4), 241–263 (2006).
7. Song, Y. Q. *et al.* Susceptibility assessment of earthquake-induced landslides using Bayesian network: A case study in Beichuan. *China. Comput. Geosci.* **42**, 189–199 (2012).
8. Esmaili, M. *et al.* Multiple regression, ANN and ANFIS models for prediction of back break in the open pit blasting. *Eng. Comput.* **30**(4), 549–558 (2014).
9. Yan, X. & Li, X. Bayes discriminant analysis method for predicting the stability of open pit slope. In *Proceedings of the International Conference on Electric Technology and Civil Engineering (ICETCE)*, Lushan, China (2011).
10. Samui, P. & Kothari, D. P. Utilization of a least square support vector machine (LSSVM) for slope stability analysis. *Sci. Iran.* **18**(1), 53–58 (2011).
11. Zhao, H., Yin, S. & Ru, Z. Relevance vector machine applied to slope stability analysis. *Int. J. Numer. Anal. Methods Geomech.* **36**, 643–652 (2012).
12. Wang, K. & Xu, F. Slope stability evaluation based on PSO-PP. *Appl. Mech. Mater.* **580–583**, 1708–1713 (2014).
13. Liu, B. W., Wang, Z. W. & Zhong, X. Y. Particle swarm optimization algorithm in numerical simulation of saturated rock slope slip. *Math. Problems Eng.* **2021**, 6682659 (2021).
14. Himanshu, N., Burman, A. & Kumar, V. Assessment of optimum location of non-circular failure surface in soil slope using unified particle swarm optimization. *Geotech. Geol. Eng.* **38**(2), 2061–2083 (2020).
15. Moayedi, H. *et al.* The feasibility of three prediction techniques of the artificial neural network, adaptive neuro-fuzzy inference system, and hybrid particle swarm optimization for assessing the safety factor of cohesive slopes. *ISPRS Int. J. Geo-inf.* **8**(9), 391 (2019).
16. Wang, H. W. Elementary introduction to multiple correlation causing harm to principle component analysis. *J. Beijing Univ. Aeronaut. Astronaut.* **22**(1), 60–65 (1996).
17. Wang, H. B., Xu, W. Y. & Xu, R. C. Slope stability evaluation using back propagation neural networks. *Eng. Geol.* **80**, 302–315 (2005).
18. Zhou, K. P. & Chen, Z. Q. Stability prediction of tailing dam slope based on neural network pattern recognition. In *Proceedings of the Second International Conference on Environmental and Computer Science ICECS'09* (IEEE Computer Society, 2009).
19. Das, S. K. *et al.* Classification of slopes and prediction of factor of safety using differential evolution neural networks. *Environ. Earth Sci.* **64**(1), 201–210 (2011).
20. Jiang, J. P. BP neural networks for prediction of the factor of safety of slope stability. In *Proceedings of International Conference on Computing, Control and Industrial Engineering (CCIE)*, Wuhan, China (2011).
21. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **27**(4), 1053–1073 (2016).
22. Mirjalili, S., Mirjalili, S. M. & Hatamlou, A. Multi-verse optimizer: A nature-in-spired algorithm for global optimization. *Neural Comput. Appl.* **27**(2), 495–513 (2016).
23. Mirjalili, S., Lewis, A. & Mostaghim, S. Confidence measure: A novel metric for robust meta-heuristic optimization algorithms. *Inf. Sci.* **317**, 114–142 (2015).
24. Abd Elaziz, M. *et al.* Cooperative meta-heuristic algorithms for global optimization problems. *Expert Syst. Appl.* **176**, 114788 (2021).
25. Geladi, P. & Kowalski, B. R. Partial least-squares regression: A tutorial. *Anal. Chim. Acta* **185**, 1–17 (1986).
26. Romera-Fernández, M. *et al.* Feasibility study of FT-MIR spectroscopy and PLS-R for the fast determination of anthocyanins in wine. *Talanta* **88**, 303–310 (2012).
27. Wang, H. W. & Zhu, Y. H. PLS regression in the function of eliminating multi-correlation. *Appl. Stat. Manag.* **15**(6), 48–52 (1996).
28. Wang, H. W. *The Method and Application of Partial Least Squares Regression* (National Defense Industry Press, 1999).
29. Fausett, L. V. *Fundamentals of Neural Networks* 1st edn. (Prentice Hall, 1994).
30. Naghadehi, M. Z. *et al.* A new open-pit mine slope instability index defined using the improved rock engineering systems approach. *Int. J. Rock Mech. Min. Sci.* **61**, 1–14 (2013).
31. Li, B. *et al.* Slope stability analysis based on quantum-behaved particle swarm optimization and least squares support vector machine. *Appl. Math. Model.* **39**, 5253–5264 (2015).
32. Eberhart, R. C. & Kennedy, J. A new optimizer using particle swarm theory. In *Proceedings of Sixth International Symposium on Micro Machine and Human Science. Nagoya, Japan*, 66–73 (1995).
33. Zhang, Q. K. *et al.* Vector coevolving particle swarm optimization algorithm. *Inf. Sci.* **394–395**, 273–298 (2017).
34. Gao, J. Y., Xing, Y. C. & Chen, Y. X. Prediction model for stability of high loess slopes. *Chin. J. Geotech. Eng.* **33**(s1), 163–169 (2011).
35. Jiao, L. C. *The Theory of Neural Network System* (Xidian University Press, 1990).

Acknowledgements

This work is financially supported by the China Postdoctoral Science Foundation (Grant Number 2020M680950), for which the author is grateful.

Author contributions

Bin Gong conducted the research and wrote the manuscript.

Competing interests

The author declares no competing interests.

Additional information

Correspondence and requests for materials should be addressed to B.G.

Reprints and permissions information is available at www.nature.com/reprints.

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



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021