Mechanical Parameter Identification of Hydraulic Engineering with the Improved Deep Q-Network Algorithm

Ji, Wei; Liu, Xiaoqing; Qi, Huijun; Liu, Xunnan; Lin, Chaoning; Li, Tongchun

DOI
10.1155/2020/6404819

Publication date
2020

Document Version
Final published version

Published in
Mathematical Problems in Engineering

Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6404819, 20 pages
https://doi.org/10.1155/2020/6404819

Research Article

Mechanical Parameter Identification of Hydraulic Engineering with the Improved Deep Q-Network Algorithm

Wei Ji,[1] **Xiaoqing Liu**,[1] **Huijun Qi**,[2] **Xunnan Liu**,[1] **Chaoning Lin**,[1,3] **and Tongchun Li**[1]

[1]*College of Water Conservancy and Hydropower Engineering, Hohai University, Nanjing 210098, Jiangsu, China*
[2]*College of Computer and Information, Hohai University, Nanjing 210098, Jiangsu, China*
[3]*Faculty of Technology, Policy, and Management, Delft University of Technology, Delft 2628 BX, Netherlands*

Correspondence should be addressed to Xiaoqing Liu; lxqhhu@163.com and Huijun Qi; qihuijun@hhu.edu.cn

Received 30 July 2020; Accepted 29 November 2020; Published 28 December 2020

Academic Editor: Zheng-zheng Wang

During the long-term operating period, the mechanical parameters of hydraulic structures and foundation deteriorated gradually because of the environmental factors. In order to evaluate the overall safety and durability, these parameters should be calculated by some accurate analysis methods, which are hindered by slow computational efficiency and optimization performance. The improved deep Q-network (DQN) algorithm combined with the deep neural network (DNN) surrogate model was proposed in this paper to ameliorate the above problems. Through the study cases of different zoning in the dam body and the actual engineering foundation, it is shown that the improved DQN algorithm has a good application effect on inversion analysis of material mechanical parameters in this paper.

1. Introduction

The premier task is to monitor the safe status of structures during the operating period. There have been catastrophes of engineer crash from time to time around the world due to the lack of overall monitoring methods and the low analysis accuracy of calculating methods. A disastrous example is that the dam Edenville broke, and the leaking flood shattered both Smallwood dam and Sanford dam subsequently in the downstream position, which caused serious damage to surrounding cities.

The hydraulic project crashes happen mainly because of the collapse of the dam body and the sliding of the foundation or abutment. During the operating term, the concrete dam is affected by environmental factors obviously. At the microlevel, there are physical and chemical reactions in the parameters of the dam body material and foundation material, so their mechanical parameters deteriorated gradually, leading to the increase of structure displacement or leakage at the macrolevel. Both the deformation of the dam body or foundation and the leakage of the concrete structure are key monitoring targets. The deformation monitoring includes forward analysis and inversion analysis. The former is to map the linear or nonlinear relation between environmental loads and displacement by establishing a regress model [7–9], whose target is predicting the status of the engineering and environment nearby in the future. The latter is to check the strength and the stability according to the mechanical parameters of structures or foundations by calculating the data of structural operating state combined with the data of the environmental variation [10].

Because the constitutive models of practical engineering are all nonlinear, it is impossible to work out the problems directly. By calculating the maximum or minimum value of target functions, the heuristic algorithms became the main methods to optimize parameters in the feasible region. Particle swarm optimization (PSO) algorithm and genetic algorithm were applied to optimize the structural parameters in the early time [11]. Kang introduced the artificial bee algorithm in 2013 [12]. And he optimized the models by combining heuristic algorithm with machine learning algorithm in 2016 [13–15]. After that, he improved firework algorithm and obtained better effect in identifying parameters [16]. Besides, Lin carried out inversion calculation with

wolf pack algorithm, and the resultant accuracy was higher than whale optimization algorithm.

There are two main problems existing in the inversion analysis of hydraulic structures. The first one is that the current displacement inversion method is based on the finite element method (FEM). Under the combination of different mechanical parameters and environmental loads, the nodes' displacements are calculated by the finite element model. With the growing number of parameters, the calculating dimensions rise synchronously. Besides, the time complexity of the finite element model increases sharply with more grids. The two factors could lead to the result that the calculating convergence time is so long that the feasibility of application in the practical project is low. The second one is that although so many heuristic algorithms provide the possibility to implement global search in the feasible region, these methods calculate and compare the target values after sampling practical points in the parameter space, so they could not guarantee the best consequence in the multidimensional parameter space and have poor convergence in practice.

Recently, machine learning algorithm with a positive developing trend includes three parts, which are supervised learning, unsupervised learning, and reinforcement learning (RL) [17]. As the cutting-edge branch, RL differs from the other ones. It is a learning algorithm with delay effect, seeking the best policy with dynamic programming [17]. The core idea is that the agent tries different policies to select corresponding actions under diverse state from the environment during the interactive process between the agent and the environment, so the agent could find the best action to maximize the reward when facing different states after the learning stage [18]. RL adopts the way of exploring from the beginning time and then utilizing the exploratory experience to complete the trail-and-error process [19]. Bellman proposed a dynamic method to deal with the value function based on the information from the systematic state [20], but the curse of dimensionality occurred when the method was applied, which was solved effectively by Mes and Rivera [21]. Some scholars introduced the function approximation method to access the value when the state and action were consecutive, such as the linear function and artificial neural network [23, 24]. With the gradual development of RL theory, these relative technologies had made great progress in the industry. Zhiang Zhang et al. reduced the indoor energy consumption by 16.7% by optimizing the HVAC system with deep reinforced learning algorithm [25]. Zhe Wang and Hong discussed the contribution and current obstacles when RL was adopted in controlling buildings [26]. The industry of robot employed RL to control the mechanical action accurately [27–30]. Fangyuan Chang et al. achieved the goal of reducing cost in the charging battery by combining RL and LSTM [31].

To improve two inversion problems with machine learning mentioned above in the paper, the DNN surrogate model and reinforcement learning are introduced into the structural inversion calculation for the first time. The deep neural network completes the learning stage with training samples which are the calculating results from the FEM, which makes the DNN model replace the finite element model to map the target points' displacements approximately and improve the convergence efficiency greatly under the premise of ensuring the calculating accuracy. The basic theory of reinforcement learning guarantees the convergence of the algorithm. The inversion calculation of structural material parameters with monitoring data is a Markov process. Its core is working out the best value of a nonlinear function in the global parameter space. Taking the monitoring data as a part of the observable environmental state, the inversion calculation and optimization of structural material parameters can be realized through reinforcement learning combined with the engineering's deep learning surrogate model. This paper adopts the punitive idea which is a negative reinforcement mode to form deep reinforcement learning algorithm by combining the target of inversion calculation and the DNN surrogate model with reinforcement learning. Besides, the interactive mode of information between the agent and the environment is improved to adapt to the optimization of material parameters of engineering structures and the surrounding foundation. The last part is to employ a new mode to express the displacement relativity among different monitoring points from the same structural sections to make the deep reinforcement learning algorithm adapt to the inversion calculation of multiple zones, to ensure the coordination among the parameters among all zones in the same section, so that this algorithm could get a wider application to introduce a new mode for the hydraulic inversion analysis.

2. Theory

2.1. The Inversion Theory of Mechanical Parameters. The elastic modulus is calculated inversely by the relation between the monitoring data of dam deformation and those of the environment. According to the monitoring theory [32], the displacement along the river of the dam body, *disp*, consists of the water pressure component δ_H, time-dependent component δ_T, and temperature component δ_θ.

The water pressure component δ_H is strongly related to the upstream water head, mechanical parameters of the structure and foundation, and the coordinates of target points. The constitutive model of the concrete dam reads

$$u_c = \mathbf{F}(\mathbf{E}, \mathbf{H}, x, y), \tag{1}$$

$$\mathbf{E} = [E_1, E_2, \ldots, E_n]. \tag{2}$$

F maps the relation between the displacements of finite element nodes and the state combined by different material parameters and environmental loads. **E** is a vector consisting of different material parameters in every zone from the finite element model. **H** is the water head. (x, y) is a group of nodes' coordinate. u_c is the displacement of target finite element nodes calculated by the constitutive model F with the target mechanical parameter E when facing different environmental loads.

The inversion analysis is to seek the suitable mechanical parameters to minimize the error f_e which is produced by

the displacement series of target nodes and water pressure component δ_H separated from displacement monitored by measuring instruments. The error f_e reads

$$f_e = \|u_c - \delta_H\|_2^2. \tag{3}$$

2.2. DNN Surrogate Model. A three-layer network structure with a suitable activation function could approximate any function infinitely in theory with reasonable number of iterative epochs [33]. According to equation (1), the factors affecting the displacement of nodes u_c are the material mechanical parameters E, water level H, and the coordinate of nodes (x, y), so the form of the sample is $[E, H, x, y \vdots u_c]$, which indicates that the input vector is $[E, H, x, y]$. and the calculating target of output node O is u_c, shown in Figure 1 and equation (4). f represents the mapping relation between input data and output data. After the input layer and output layer are determined, the number of layers and nodes in the hidden layer need to be determined by trail calculation according to the specific demand. The output error J results from equations (4) and (5). W and b are weights and biases, respectively, connecting these layers.

$$O = f(E, H, x, y), \tag{4}$$

$$J(E, H, x, y, W, b) = \frac{1}{2}\|u_c - O\|_2^2. \tag{5}$$

The neural network adopts the gradient descent method to minimize the output error J and upgrade the network parameters W and b. By selecting reasonable number of samples each time, the minibatch gradient descent method could not only ensure the representativeness of each group of samples to reduce the negative impact of noise points on network and ensure the convergence but also prompt the velocity of network convergence to reach a better learning model, so this method meets the requirements of this paper.

The process of producing DNN samples needs the following steps:

Input: constitutive model F, m groups of material mechanical parameters E, and n groups of reasonable water level H

Output: sample of finite element node displacement $[E, H, x, y \vdots u_c]$

start:

for $i = 1$ to m:

for $j = 1$ to n:

calculate displacement of nodes by model F

$$u_c = F(\mathbf{E}, H, x, y)$$

store sample $[E, H, x, y \vdots u_c]$

output all samples.

end

The process of constructing the DNN surrogate model is shown in the following steps:

Input: the number of network layers, the node number of each layer, activation function, loss function, the maximum epoch N, and m samples from each batch

Output: DNN model with fixed weights W and biases b

start:

initialize the weight coefficient matrix W and bias vector b randomly

for iter = 1 to N:

input vector $[E, H, x, y]$ into nodes of input layer

forward propagation calculation

$O = f(E, H, x, y)$

calculating the loss

$J(E, H, x, y, W, b) = (1/2)\|u_c - O\|_2^2$

back propagation calculation according to the loss of the last step

upgrade weight matrix W and bias vector b

output the DNN model with fixed network structure and parameters

end

The fixed DNN model could map the relation between input nodes and output nodes in a very short time which overcomes the problem that the calculating velocity of the finite element model is too slow, so replacing the constitutive model with the network is reasonable for inversion analysis.

2.3. Optimization Capability of DQN. According to Figure 2, the framework of reinforcement learning consists of five parts: agent, environment, state, action, and reward (the abbreviations are Agent, Env, s, a, and r). Env provides a current state s as an input of the agent. Agent selects action a corresponding to s according to policy π. Env accepts and assesses a to calculate the reward r and then produces the next state ($state'$ in Figure 2). The value of policy π in the current Env is determined by accumulating r of all time steps in each epoch. The calculating formulas are shown as follows:

$$\begin{aligned} G_t &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{i=0}^{T} \gamma^i r_{t+i+1}, \quad (t < T), \\ V_\pi(S) &= E_\pi[G_t | s_t t = ns], \\ q_\pi(s, a) &= E_\pi[G_t | s_t t = nsq, ha_{t=} xa]. \end{aligned} \tag{6}$$

γ is the discounted factor for the reward of future time steps, whose range is $[0, 1]$. $V_\pi(S)$ is the value function of the state, and $q_\pi(s, a)$ is the value function of state-action.

The target of reinforcement learning is to seek the best policy π_* when facing different states in Env. Under the guidance of the best policy π_*, the accumulative reward G_t reaches the highest value. The corresponding value function

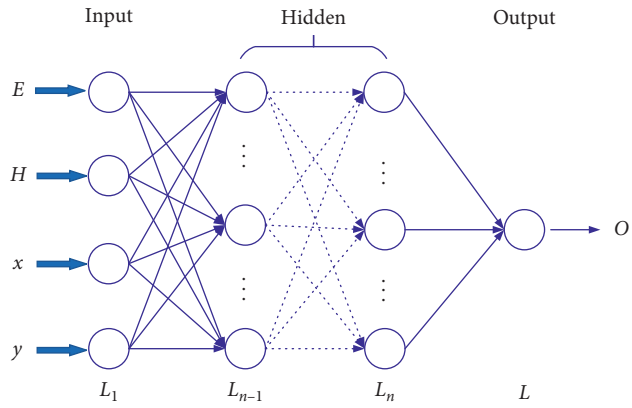

Figure 1: The structure of the DNN surrogate model.

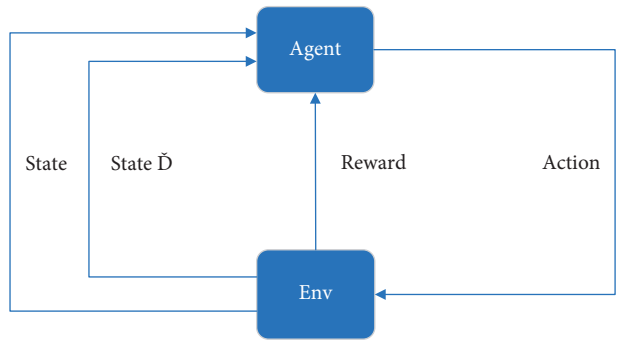

Figure 2: The framework of reinforcement learning.

of state-action is the optimal one $q_*(s,a)$. S is the collection of s, and A is the collection of a.

$$q_*(s,a) = \max_{\pi} q_\pi(s,a), \quad (s \in S,\ a \in A). \tag{7}$$

Q-learning is mainly adopted to work out $q_\pi(s,a)$, which was proved that the convergence can be guaranteed in theory by Watkins in 1992 [34]. It is a value-based method, upgrading $q_\pi(s,a)$ between different time steps with the time difference method in one epoch. The calculating method is shown in the following under a certain policy π:

$$q_t(s,a) = q_t(s,a) + \alpha\left(r + \gamma \max_{a'} q_{t+1}(s',a') - q_t(s,a)\right), \tag{8}$$

where $\max_{a'} q_{t+1}(s',a')$ means the value corresponding to a', which maximizes the q value among the optional actions, under the state s' in $t+1$th time step. α is the learning rate, which is used to control the update rate of each time step.

Agent selects a from A using two contradictory ways named exploitation and exploration. The former selects a by exploiting the past experience to solve the current state s, while the latter abandons the past experience and selects a at random to extend the action space A when facing the current state s. If RL only carries out the exploitation policy, the optimization would usually fall into local extremum because of the lack of full exploration in the action space. However, if RL gives up the exploitation policy, a thorough exploration process would make the algorithm lose the definite objective and fail to search for a better policy π. Two search means are balanced by the ε – greedy method, whose flowchart is shown in Figure 3, where the range of the threshold ε_0 is [0, 1].

The main idea of the ε – greedy method is that, in the initial period, exploring the action space A is the first choice because of short of experience. After being trained with suitable time steps, the model learned how to select better actions to accumulate experience when facing different states. The past memory is gradually used to promote the total reward. During this process, the model transits from the exploration stage to the exploitation stage by degrees, which means that the probability of random selection action shrinks correspondingly, shown with equation (9). t_{step} indicates the current time step.

$$\varepsilon = \max\left(0.01,\ \frac{(\varepsilon_0/2)}{t_{\text{step}}}\right). \tag{9}$$

The original reinforcement learning usually adopts linear transformation or look-up table method, which could not solve multidimension or nonlinear problems. DQN algorithm that combines deep learning algorithm and reinforcement learning not only obtains excellent characterization capability of deep learning to transform the data features into the state as the input of the Agent but also selects the proper action a by calculating all feasible state-action values $q_t(s,a)$. In the past, Env demanded relevance

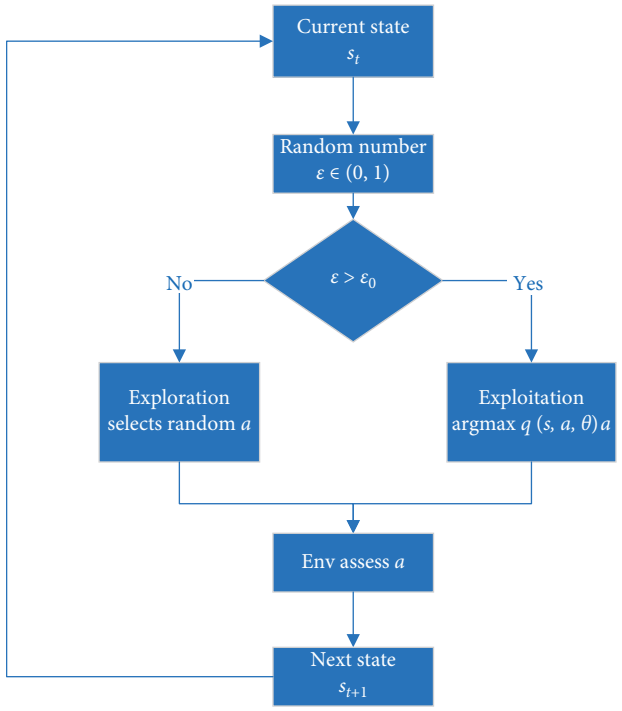

Figure 3: The flowchart of ε – greedy.

between two successive time steps to a certain degree, which did not meet the demand for independence among samples when deep learning was applied. In 2013, Mnih proposed experience replay technology to deal with this obstacle, with another advantage that data could be used repeatedly to effectively increase the input samples. This method contained two main steps [23]:

(1) Storage: store the past data $[s_t, a_t, r_t, s_{t+1}]$ in the memory zone as samples

(2) Sample and replay: extract multiple samples $[s_t, a_t, r_t, s_{t+1}]$ from each batch as the input data of the deep network

During the iterative process of Q-learning, the parameters of the state-action value function operating in the t time step are the same as those operating in the $t+1$ time step, which results in synchronous ups and downs of the q value in two time steps, enhancing the probability of model divergence. So, the actor-critic framework was introduced. The actor is expressed as $q(s, a, \theta)$, and the critic is expressed as $q(s, a, \theta^-)$, which indicate that two models share the same structure with different network parameters. The former is used to assess value of the current state. The latter is applied in the next sate to evaluate the result of the current network and guides the update of the actor network. The update mode of the q value is shown in equation (10). θ of actor is copied to θ^- of critic at a certain interval of time steps.

$$q_t(s, a, \theta) = q_t(s, a, \theta) + \alpha\left(r + \gamma \max_{a'} q(s', a', \theta^-) - q_t(s, a, \theta)\right). \tag{10}$$

2.4. Combination of the Improved DQN and Inversion Calculation. The target of mainstream RL is to develop the best policy to guide Agent to select proper a when facing different states from Env and obtain the highest accumulated reward, while the task of inversion calculation is to select an elastic modulus that is suitable for the deformation of the engineering structure and foundation. So, the interactive mode of information between Agent and Env is improved: after Agent selects a proper action a according to state s, Env assesses this a, and in the meantime, this a improves the parameters of the state to search the best material mechanical parameters.

2.4.1. Construction of the Inversion Agent. The DNN surrogate model, established according to Section 2.2, is used to calculate the agent displacement u_{cal}, as one part of Agent. Subtract u_{cal} from the displacement of target samples u_{true}, and the difference guides Agent to select a. After that, the corresponding state-action value would be evaluated. The flowchart is shown in Figure 4, where $p(a)$ means the probability of action a.

2.4.2. Improvement of the Interactive Mode between Agent and Env. How the reward r is produced by Env assessing the action a from Agent is shown in equations (11) and (12):

$$\text{error} = u_{cal} - u_{true}, \tag{11}$$

$$r = -|\text{error}|. \tag{12}$$

The target of DQN is to seek a proper elastic modulus E. The less the absolute value of reward r is, the closer the calculated elastic modulus is to the actual parameter in Env, which indicates that E in state s ceaselessly approaches the real value in Env during the iteration process.

The improvement of the interactive mode between Agent and Env in the DQN framework is that the action a selected by Agent adjusts the parameters in Env. The difference error has positive or negative states. Based on this, two kinds of action, 0 and 1, are adopted. The former indicates that E in state s is smaller than the actual one, so the positive increment ΔE could enlarge E in state s. The latter indicates that E in state s is bigger, so the negative increment ΔE could shrink E in state s. And there is a linear relation, to a certain extent, between the degree of shrinkage and expansion and the absolute value of reward r, so the mode that different actions adjust E in the state is shown with equations (13) and (14):

$$\Delta E = -r * E_{step} * (a - 0.5), \tag{13}$$

$$E = E + \Delta E, \tag{14}$$

where E_{step} is an adjustment factor, controlling the degree of adjusting E and ensuring the model could converge.

The overall process is presented as follows:

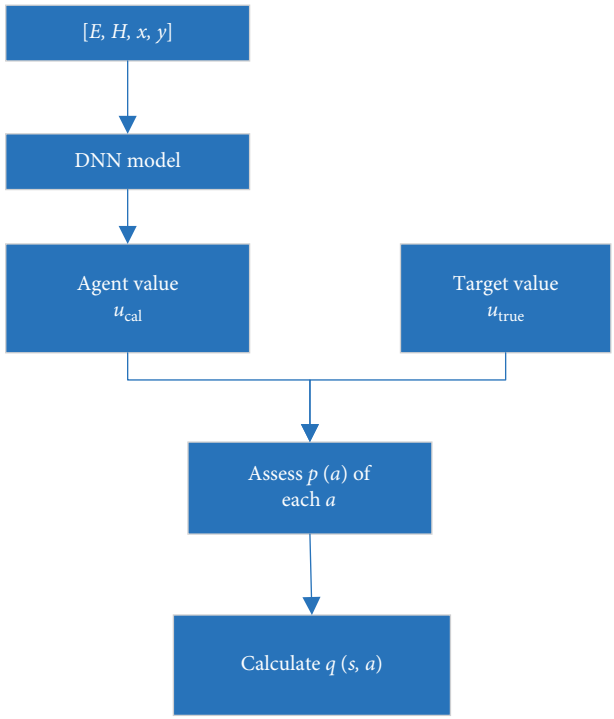

Figure 4: The structure of Agent.

Initialize memory zone D, the maximum epochs, discounted factor γ, adjustment factor E_{step}, and random probability ε_0

Initialize the actor network parameter θ, the critic network parameter $\theta^- = \theta$

for epoch = 1 to epochs:

initialize state s and the corresponding water pressure component disp_t

for $t = 1$ to T:

select action a_t from A randomly or actor network according to ε – greedy

update the random probability $\varepsilon = \max(0.01, (\varepsilon_0/2)/t)$

Env evaluate a_t and get r_t

update E in Env: $E = E - r_t * E_{step} * (a_t - 0.5)$

get $s_{t+1}, \text{disp}_{t+1}$ from Env

store data $[s_t, \text{disp}_t, a_t, r_t, s_{t+1}, \text{disp}_{t+1}]$ in memory zone D as the sample

extract a batch of samples from memory zone D

Actor calculates $q(s_t, \text{disp}_t, a_t, \theta)$

Critic calculates $\max_{a_{t+1}} q(s_{t+1}, \text{disp}_{t+1}, a_{t+1}, \theta^-)$

output = if $t \geq T - 1$ then r_t else $r_t + \gamma * \max_{a_{t+1}} q(s_{t+1}, \text{disp}_{t+1}, a_{t+1}, \theta^-)$

calculate loss = $(q(s_t, \text{disp}_t, a_t, \theta) - \text{output})^2$

optimize actor network parameter θ with Adam algorithm

copy θ to θ^- every N time steps

output: the optimum E in Env

The DQN framework is shown in Figure 5.

In summary, this paper adopts the improved DQN algorithm embedded with the DNN surrogate model. Agent completes the task to adjust E in the state from Env to minimize the absolute error (maximize the reward) calculated by agent u_{cal} from Agent and actual displacement u_{true} from the target sample, which could evaluate the quality of the optimizing result.

2.4.3. Relation of Inversion in Multizones. In different zones in the dam section, relevance among the displacements of nodes, to a certain degree, exists without causality. So, it is unsuitable to adopt equation (16) to adjust parameters in all zones by identical adjustment extent, and it is also unreasonable to adjust the parameter only in one zone corresponding to the current sample, ignoring the relevance among deformation of all zones. With the action of upstream water pressure, the whole section of the dam body demands for the deformation coordination. For example, in Figure 6, the displacement of node P_A in the upper zone is related to not only the mechanical parameters in zone Ω_1 but also those in zone Ω_2. The relevance is expressed with the following equation:

$$E_{other} = E_{other} - r_t * (a_t - 0.5) * (\text{randnum} * 0.1 * E_{step} + 0.01). \quad (15)$$

When a sample adjusts the mechanical parameters in other zones, the adjustment factor is $(\text{randnum} * 0.1 * E_{step} + 0.01)$, where *randnum* is a random number belonging to (0, 1). The random number is used to control the adjustment amplitude. Besides, 0.01 is added into equation (18) to ensure that the relevance is positive. On the contrary, when the sample adjusts the parameter in its own zone, the adjustment increment is still calculated by equation (13).

3. Case Study

3.1. Inversion Calculation of the Single Dam Zone: Case A. This case A is to minimize the cumulative absolute error of the agent displacement u_{cal} and the sample displacement u_{true} to optimize the DQN model and search an elastic modulus suitable for the whole dam section. The target displacement u_{true} is the displacement of target node u_c calculated by the constitutive model.

Step 1: establish the finite element model. The finite element model is shown in Figure 7, containing two components, dam and foundation. The horizontal direction x is along the river, and the vertical direction y is the elevation. The dam height is 107.5 m, the length of dam bottom is 88 m, and the length and width of the dam foundation are 488 m and 300 m, respectively. All mechanical parameters of the model are listed in Table 1. E_A indicates the elastic modulus

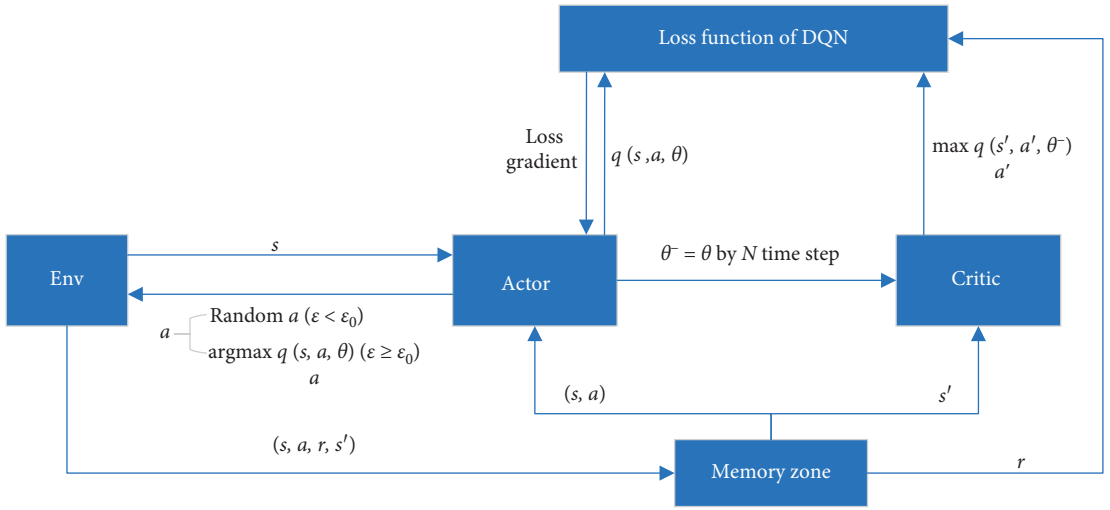

Figure 5: The framework of DQN.

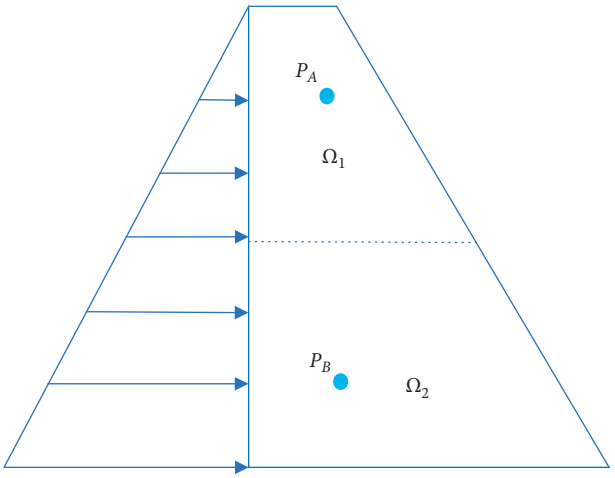

Figure 6: The diagram of multiple zones in the dam section.

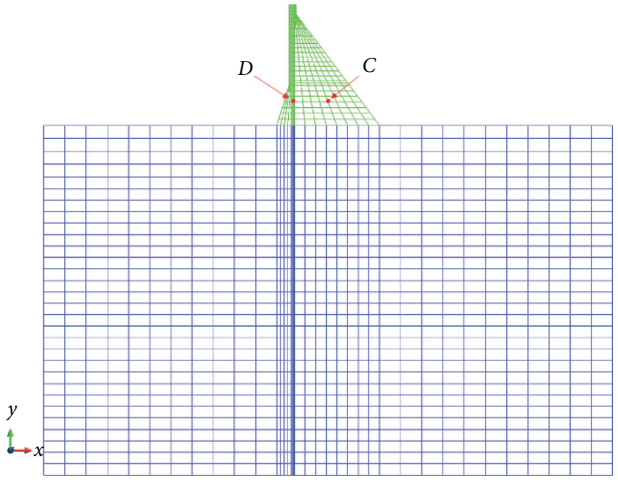

Figure 7: The finite element model of case A.

Table 1: The mechanical parameters of case A.

Component	Density (kg/m^3)	Elastic modulus (GPa)	Poisson's ratio
Dam body	2400	E_A	0.167
Foundation	2400	9	0.167

of the dam section. The nodes of foundation bottom are fixed in the horizontal and vertical direction, and the nodes at both sides of the foundation are fixed in the vertical direction.

Step 2: select the sample. 259 different water levels were extracted randomly from 86.5 m to 104.3 m, and 200 different elastic moduli E_A were randomly extracted from 5 GPa to 20 GPa evenly, not containing 10.3 GPa (the target parameter). There were 51,800 groups of combination states of mechanical parameter and water pressure. The model was calculated using software GeHoMadrid developed by Hohai University and Universidad Politécnica de Madrid to get the node displacement of all states. The result $[E, H, x, y, u_{\text{true}}]$ was stored as samples to train and verify the DNN model.

Step 3: construct the DNN surrogate model. The model had four layers, established by Keras. The first layer had four input nodes, named input-EHXY, and the form of the input vector is $[E_A, H, x, y]$. The second and third layers were fully connected layers, named Sec-layer and Third-layer, with 8 and 10 nodes, respectively. The fourth layer was the output layer named dispout with 1 node, and the calculating target was u_{true}. The specific structure is shown in Figure 8. The loss function was "mean_squared_error," optimized by Adam. The learning rate was 0.001, the maximum iterative epoch was 1000, and the activation function of all nodes adopted "relu."

The samples from step 2 were shuffled randomly, and all data were normalized to [0, 1] according to the data features. Training samples occupied 80%, and the rest were verifying samples. The changing horizontal section and deformable foundation have an effect on the displacement in the dam. And the increase of altitude weakens the nonlinear effect. The location of node C is in the lower zone and near the foundation, so this zone could illustrate nonlinear deformation more clearly than those nodes in the higher area. Besides, the closer the node is to the foundation, the smaller the displacement is, so node C was selected. The predicting samples were the displacement along the river of node C in Figure 7 calculated under the state that the elastic modulus was 10.3 GPa with 259 water levels above.

The iterative process of the training error and verifying error is shown in Figure 9, where it indicated that, during the former 100 epochs, the two errors decreased sharply to the level close to 0. After 200 epochs, the network parameters were nearly stable. When the training stage was completed, the fixed DNN model was stored to replace the finite model in the later steps.

The displacement of different nodes is related to water level elevation and the elastic modulus of the dam body. According to the monitoring theory [32], u_{true} could be calculated by the multivariable linear regression (MLR) model shown in the following equation:

$$u_{\text{true}} = \sum_{i=1}^{3} \beta_i H^i + \beta_4 E + \beta_5 x + \beta_6 y + \tau. \tag{16}$$

Training samples and predicting samples are the same as those of the DNN model. The calculating results of predicting samples are shown in Table 2.

From Table 2, the mean relative errors of DNN and MLR were lower than 3%. Furthermore, the accuracy of the DNN model in both mean relative error and maximum relative error was an order of magnitude higher than that of the MLP model. The possible reason was that the MLP model constructed regression factors based on the plane cross-section assumption and complete elastomer assumption, but node C was near the dam foundation, which meant during the calculation, the deformation of the dam body and foundation did not meet the first assumption. The displacement of node C was not completely linear. DNN model was nonlinear, which represented that the neural network could map the relation between environmental load and displacement more efficiently. The maximum relative error was lower than 2%. From this, it was reasonable for the DNN, after being well trained, to replace the finite element model.

Step 4: construct Agent. The Agent included three parts. The first one was the DNN model stored in step 3, which received the state s, $[E_A, H, x, y]$ and produced agent displacement u_{cal}; the second part was the target displacement u_{true} corresponding to the current state, named disp_value; the third part was two optional actions, named actions_input. u_{cal} minus u_{true} in the layer named subtrac_1 was the error, which was used to select action a, combining the layer actions_input to calculate the state-action value q. The specific structure is shown in Figure 10.

Step 5: calculation with the DQN algorithm. The predicting samples normalized in step 3 were target samples in this step. This maximum number of epoch was 100, and each epoch had 100 time steps. The initial value of probability ε_0 was determined as 0.2. With the increase of time step t, ε decreased with a linear trend and would be stable at 0.01 eventually. The sample volume of the memory zone was 512, the discounted factor γ was 0.5, the learning rate α was 0.5, the adjustment factor E_{step} was 0.01, and the replay size of samples in each time step was 32. The initial modulus could be selected randomly, whose range was from 5 GPa to 20 GPa. The target displacement was the value of node C calculated by FEM with 259 water levels when the elastic modulus was 10.3 GPa. The iterative process and result are shown in the following.

3.1.1. Process Analysis. Figures 11 and 12 show that, in the initial period, the model was in the exploration stage, selecting actions randomly, resulting in the fluctuation of the reward. Then, the DQN model moved into the exploitation stage, with the increase of epoch and selecting the right action when facing different states. The absolute value of the reward decreased smoothly, and the searching parameters

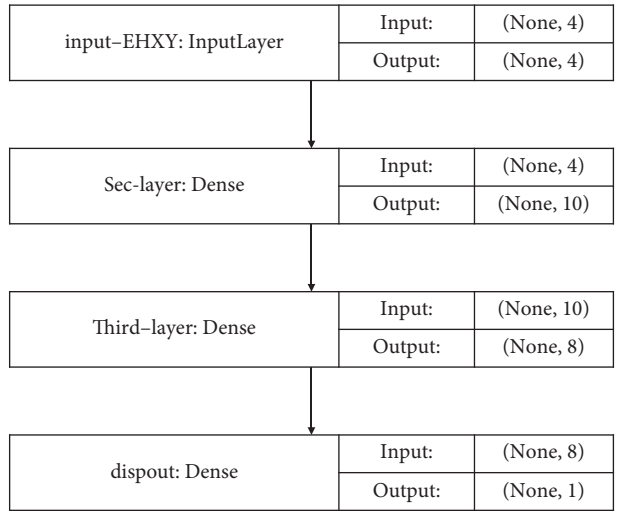

Figure 8: Structure of the DNN model in case A.

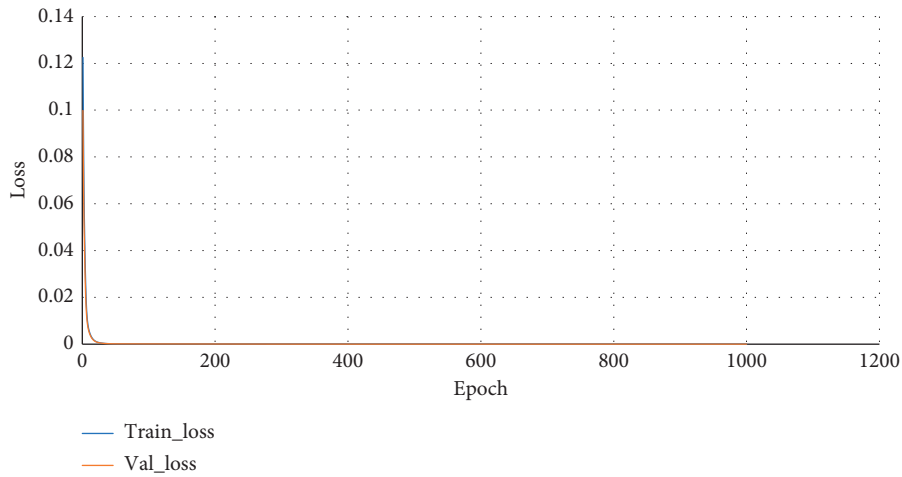

Figure 9: The iterative process of the model error in case A.

kept approaching the target value in the former 40 epochs before the model was generally stable. The result of inversion calculation reached the optimal status of the model.

3.1.2. Result Analysis. Figure 13 shows that the blue line that represented the agent displacement calculated by Agent almost coincided with the orange line that represented the actual displacement, which indicated that the values of two lines were very close in the same water level. Figure 14 shows the absolute error between two displacement lines, where the mean absolute error was 0.015 mm, and the standard deviation (SD) was 0.0085 mm. The error values were mainly concentrated in (0, 0.02) mm. Thus, the error value remained at a low level.

When the interactive process between Agent and Env was completed, the eventual elastic modulus E_A was 10.3187 GPa, and the actual target was 10.3 GPa. So, the absolute error was 0.0187 GPa, and the relative error was 0.18%. Two possible reasons of the error were as follows: the first one was that the DNN surrogate model had a mean error of 0.372% relative to the finite element model, and its accuracy could determine the accuracy of DQN; the second reason was that the search method of DQN was not perfect. The error level indicated that the inversion consequence calculated by DQN algorithm was very close to the actual value in case A, which meant the method of this paper had a fine effect on the inversion analysis of the whole dam section.

3.2. Inversion Calculation of Double Dam Zones: Case B. This case B is to minimize the cumulative absolute error of the agent displacement u_{cal} and the sample displacement u_{true} to optimize the DQN model and search two elastic moduli suitable for the upper and lower dam zones. The target displacement u_{true} is the displacement of target node u_c calculated by the constitutive model.

Step 1: establish the finite element model. The finite element model is shown in Figure 15, containing three components: two zones in the dam section and foundation.

Table 2: Error of predicting samples in 10.3 GPa.

Relative error (%)	DNN	MLR
Mean	0.372	2.723
Maximum	1.833	16.212

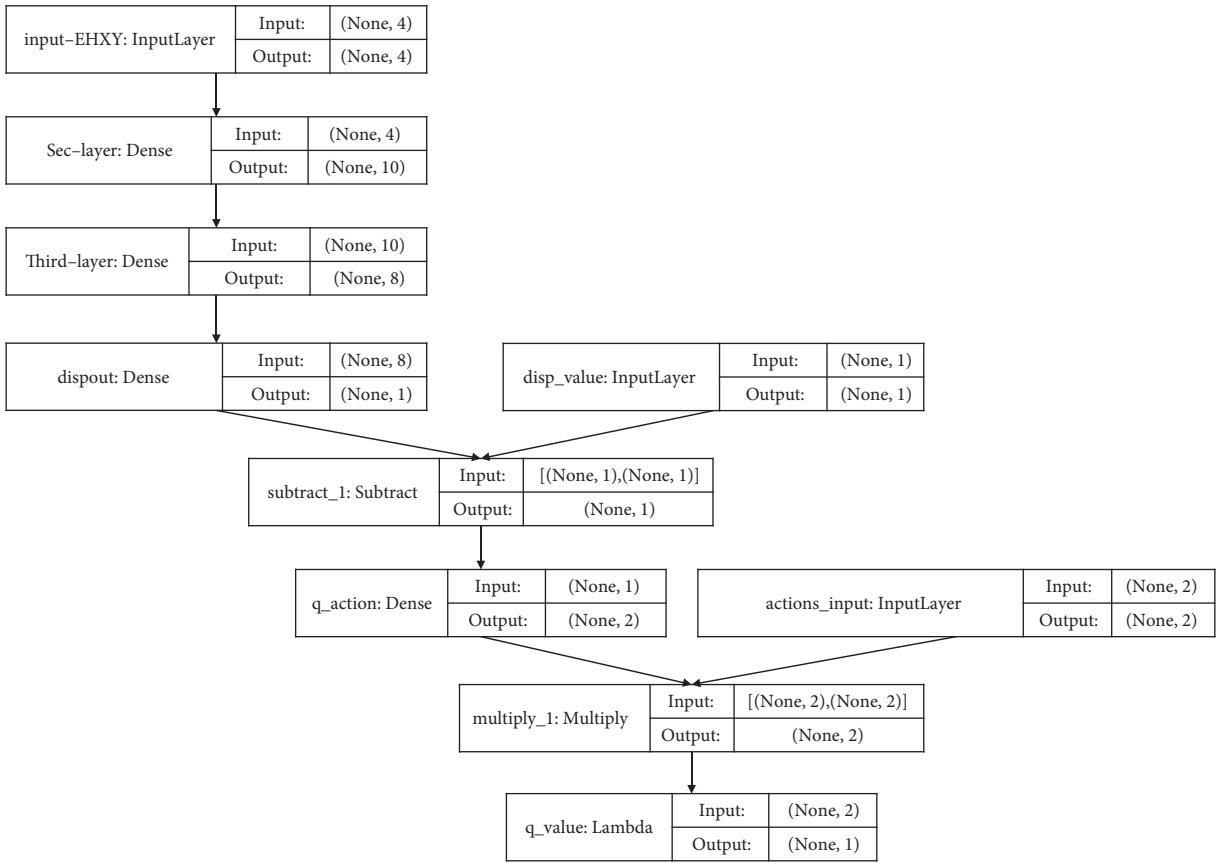

Figure 10: The structure of Agent in case A.

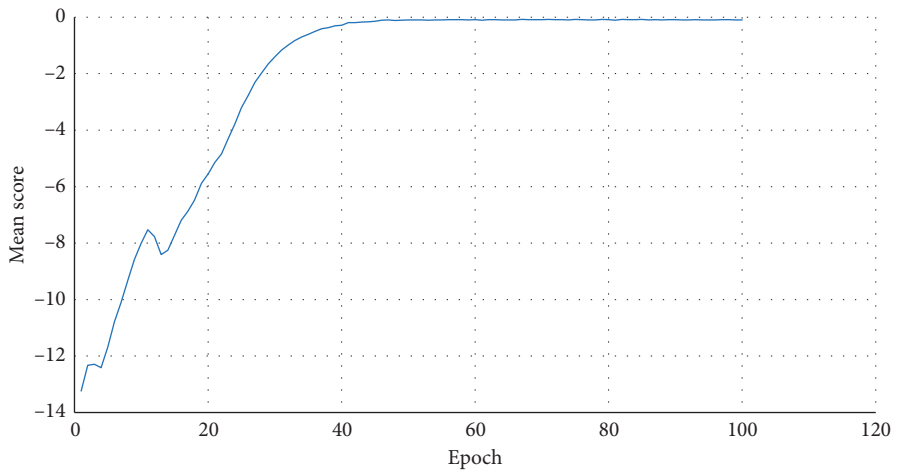

Figure 11: The iterative process of the reward in case A.

The horizontal direction x is along the river, and the vertical direction y is the elevation. The dam height is 50 m, the width of the dam crest is 5 m, the length of dam bottom is 5 m, and the length and width of the dam foundation are 190 m and 100 m, respectively. All mechanical parameters of the model are listed in Table 3. E_{B1} indicates the elastic modulus of the

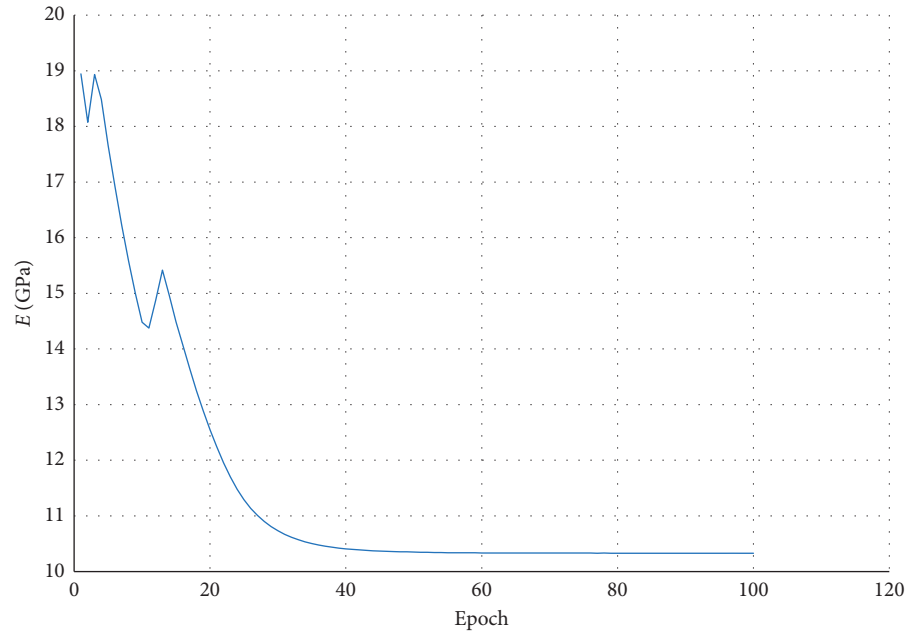

Figure 12: The iterative process of E of Env in case A.

Figure 13: Contradiction of the calculating value and actual value.

upper zone, and E_{B2} indicates the elastic modulus of the lower zone. The nodes of foundation bottom are fixed in the horizontal and vertical direction, and the nodes at both sides of the foundation are fixed in the vertical direction.

Step 2: select the sample. 140 different water levels were extracted randomly from 36.0 m to 50.0 m, and 230 groups of different combinations of elastic moduli, E_{B1} and E_{B2}, were randomly extracted. The range of modulus in the upper zone, E_{B1}, was 9.5 GPa~22.5 GPa, not containing 18.0 GPa, while that in the lower zone, E_{B2}, was 15 GPa~25 GPa, not containing 22.0 GPa because the elastic module in the lower zone was larger than that in the upper zone in order to reduce the engineering cost. During the calculation of the finite element model, the elastic modulus in the green zone including node A remained smaller than that in the yellow zone including node B. There were 32,200 groups of combination states of the mechanical parameter and water pressure. The model was calculated using software GeHoMadrid to get the node displacement of all states. The result $[E_{B1}, E_{B2}, H, x, y, u_{true}]$ was stored as samples to train and verify the DNN model.

Step 3: construct the DNN surrogate model. Different from case A, the input layer of the DNN surrogate model in case B had 5 nodes, and the input vector was $[E_{B1}, E_{B2}, H, x, y]$. The rest of the hyperparameters were identical to those of the DNN model in case A. The specific structure of the DNN model in case B is shown in Figure 16.

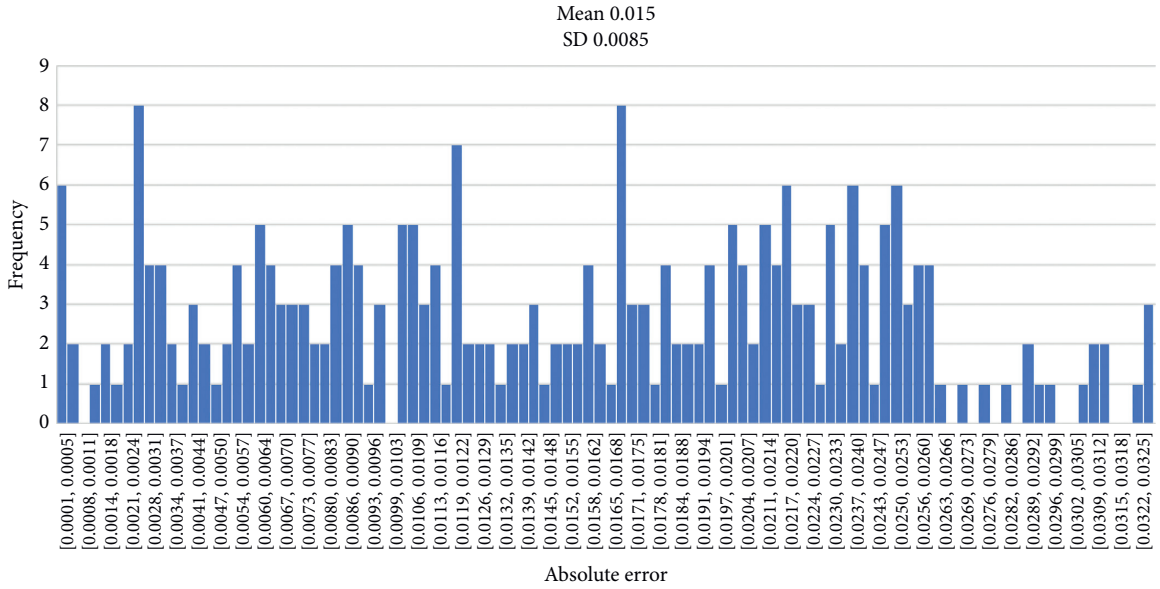

Figure 14: Histogram of the absolute error of distribution.

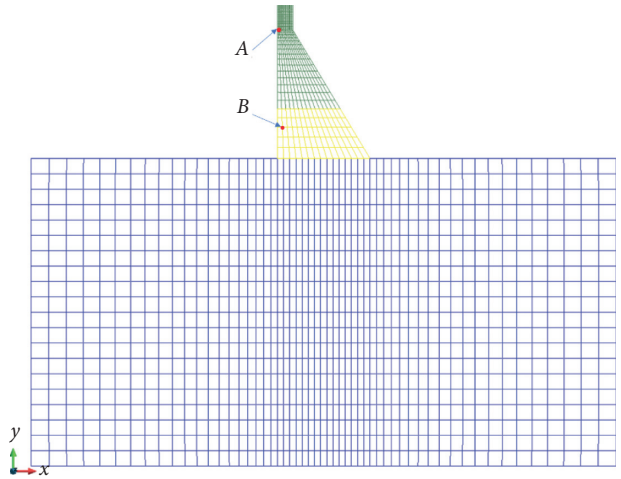

Figure 15: The finite element model of case B.

Table 3: The mechanical parameters of case B.

Component	Density (kg/m^3)	Elastic modulus (GPa)	Poisson's ratio
Upper zone	2400	E_{B1}	0.167
Lower zone	2400	E_{B2}	0.167
Foundation	2400	5	0.167

The samples from step 2 were shuffled randomly, and all data were normalized to [0, 1] according to the data features, where the first independent variable E_{B1} and the second one E_{B2} were normalized with the same scale. Training samples occupied 80%, and the rest were verifying samples. The predicting samples were the displacements along the river of nodes A and B in Figure 15 calculated under the state that the upper elastic modulus was 18.0 GPa and the lower one was 22.0 GPa with 140 water levels above.

The iterative process of the training error and verifying error is shown in Figure 17, where it indicated that, during the former 100 epochs, the two errors decreased sharply to the level close to 0. After the former 200 epochs, the network parameters were nearly stable. After the training stage, the DNN model was stored to replace the finite model in the later steps.

The maximum relative error was 3.56%, and the mean relative error was 0.59%, which indicated that the overall

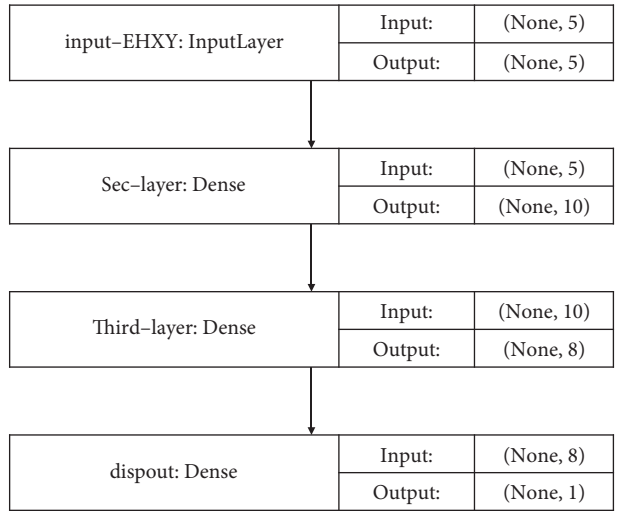

Figure 16: Structure of the DNN model in case B.

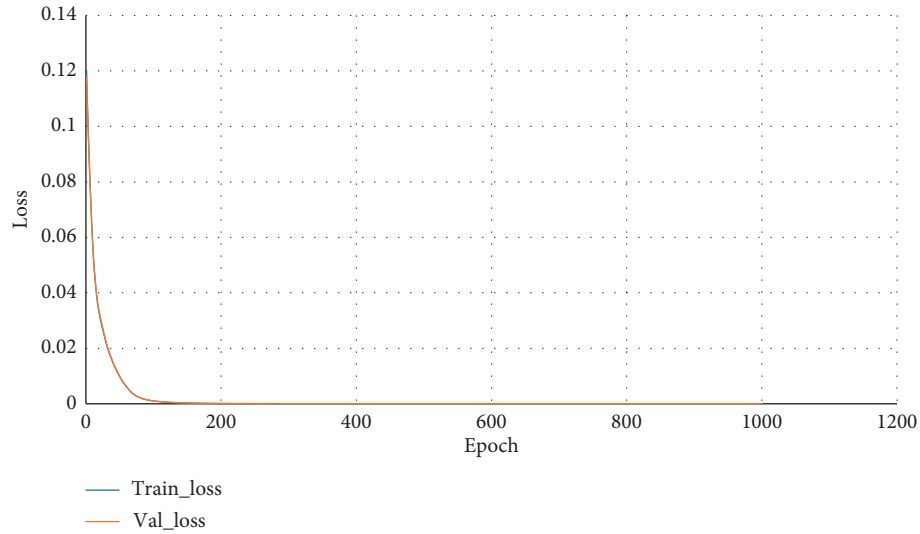

Figure 17: The iterative process of the model error in case B.

relative error was low. It was reasonable for DNN, after being well trained, to replace the finite element model according to the accuracy.

Step 4: construct Agent. The structure and parameters in Agent were the same as those in case A except that the input layer of the fixed DNN model had 5 nodes.

Step 5: calculation with DQN algorithm. The predicting samples normalized in step 3 were calculated as target samples in this step. This maximum number of epoch was 200, and each epoch had 100 time steps. The variation of random probability ε was identical to that in case A. The sample volume of the memory zone was 512, the discounted factor γ was 0.5, the learning rate α was 0.5, the adjustment factor E_{step} was 0.03, and the replay size of samples in each time step was 64. The initial modulus could be selected randomly in a reasonable range. In case B, the initial values in both the green zone and in the yellow part were determined to be 25 GPa. The target displacements were the values of node C calculated by FEM with 140 water levels when the elastic moduli were 22.0 GPa and 18.0 GPa. The iterative process and result are shown in the following.

3.2.1. Process Analysis. Different from Figure 11, Figure 18 shows that the zoning reward had been increasing with constant fluctuation during the negative reinforcement stage and then was stable in (−0.2~0), which indicated that the change of one zone would lead to the fluctuation of another zone. As a result, the agent displacement could not remain steady completely, but the overall trend was increasing, representing that the absolute value of the reward was decreasing, which meant that the penalty from Env was lower and lower and got stable in a certain range. Figure 19 shows the searching parameters kept approaching the target parameters and then tended to be stable. The result of

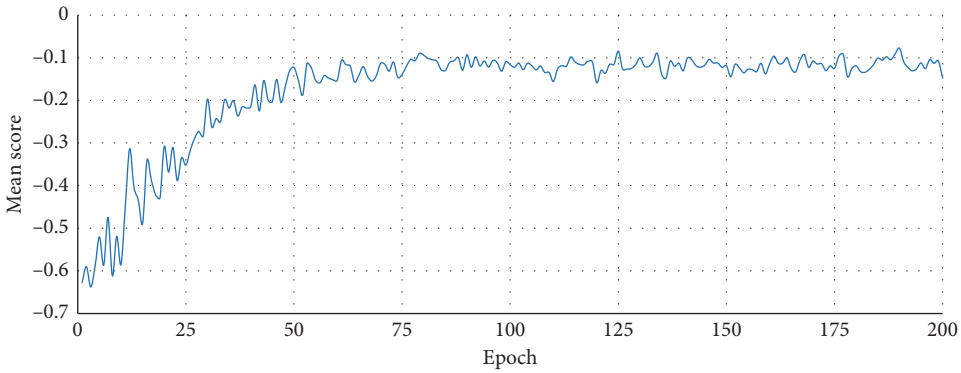

Figure 18: The iterative process of the reward in case B.

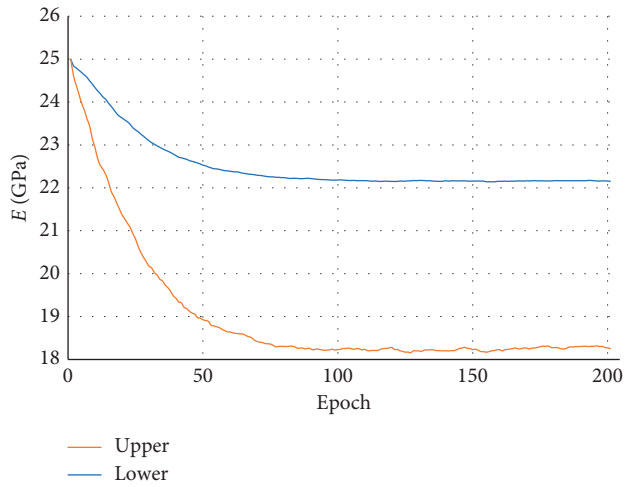

Figure 19: The iterative process of E in Env in case B.

inversion calculation reached the optimal status of the model.

3.2.2. Result Analysis. The results of the absolute error are shown in Figures 20 and 21. Both value and distribution of the error related to node A were better than those of node B. The possible reason was that node A was near the dam crest, so the water level elevation had a stronger effect on its displacement, and the foundation had a weaker influence on node A, which represented the deformation of node A had a better regularity.

The calculating result of DQN algorithm is listed in Table 4, which showed the relative error in the upper zone was 1.29%, and the other one in the lower zone was slightly smaller, 0.86%. The error level indicated that the inversion consequence calculated by DQN algorithm was very close to the actual parameter values in case B, meaning the method of this paper had a fine effect on the inversion analysis of the dam with multiple zones.

3.3. Verification with Actual Engineering: Case C. The engineering is a RCC dam on the main stream of a river in Cambodia, with 10 dam sections. The elevation of the dam crest is at 153.00 m, and the bottom surface is at 41.00 m, with a maximum dam height of 112.00 m. The width of the dam crest is 6.00 m. The top elevation of the upstream break slope is 84.0 m, and the slope is 1 : 0.3, and the downstream slope is 1 : 0.75. The mechanical parameters of the rock in the dam foundation are shown in Table 5. Under the long-term action of dam gravity and groundwater, the displacement along the river of the project showed a slow upward trend during the operating period, so the material parameters of the dam foundation should be paid attention to. The target of case C is the elastic modulus of the foundation of the project.

Step 1: establish the finite element model. This case selected one section of the dam, where the foundation was at 45.5 m, and the dam height was 107.5 m. The length of the dam foundation was 88.0 m, and the size of the dam foundation was 488 m ∗ 300 m. Some scholars [35, 36]

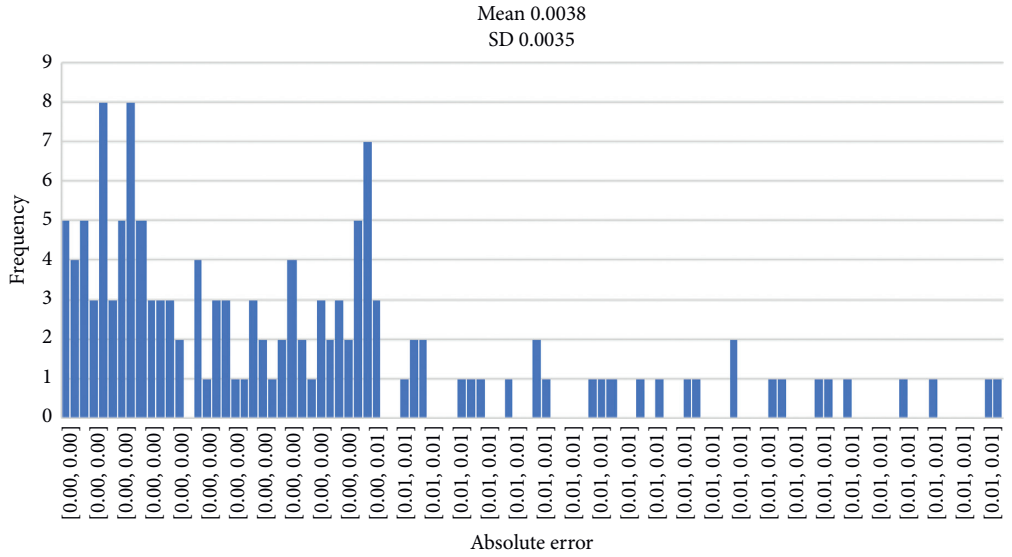

Figure 20: The distribution of the absolute error of node *A*.

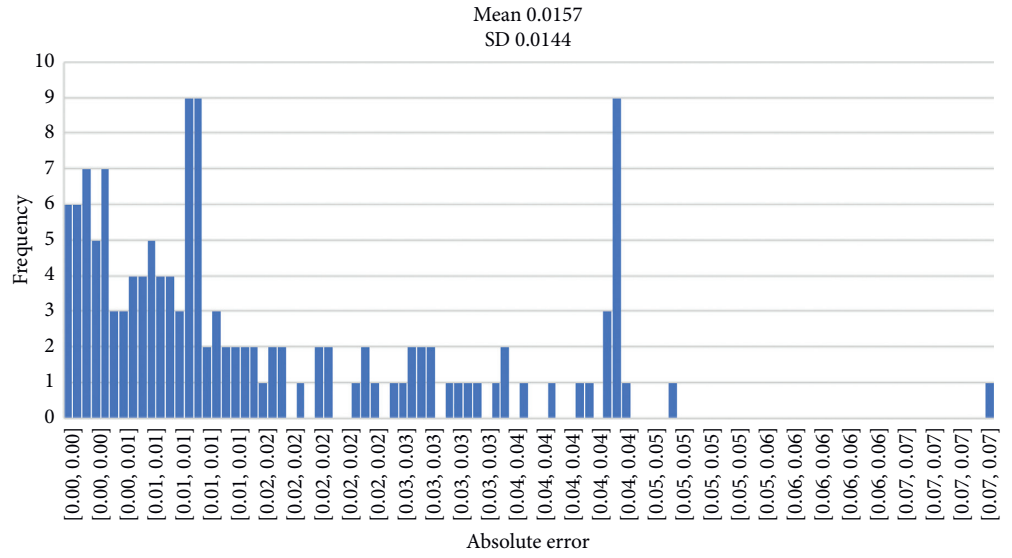

Figure 21: The distribution of the absolute error of node *B*.

Table 4: The result of inversion calculation in two zones.

Zone	Upper	Lower
Target (GPa)	18.0	22.0
Result (GPa)	18.2325	22.1893
Absolute error (GPa)	0.2325	0.1893
Relative error (%)	1.29	0.86

proposed that the mechanical parameters of the layer between structure and foundation were inferior to those of the surrounding rock mass because of the excavation technology or earthquake. However, the calculating model is based on the static load. Besides, the calculating depth of the foundation in this model is 300 m, so the weak layer is so thin to be ignored to reduce the complexity of this model. The finite element model was identical to the one in case A. The monitoring displacement series, 221 data along the river from July 25, 2014, to Oct 31, 2019, came from the inverted plumb line, node *D* in Figure 7, located near the upstream side of the dam body. Mechanical parameters of the model are listed in Table 6. E_C indicated the elastic modulus of the foundation. Because the gravity dam is usually built on the fresh base rock, the main foundational material is quartz sandstone.

Table 5: The mechanical parameters of the rock in the dam foundation.

Rock type	Young's modulus (GPa)	Poisson's ratio	Shearing strength	Compressive strength (MPa)	Bulk density (kg/m^3)
Quartz sandstone	5~10	0.15~0.23	C = 4.7~8.4 MPa φ = 38.2°~45.5°	60~80	2520
Fine sandstone	7~8	0.18~0.25	C = 3~5 MPa φ = 35°~45°	45~55	2510
Silty mudstone	2~3	0.28~0.30	C = 0.8~1.0 MPa φ = 35°~38°	10~20	2530
Mudstone	1~2	0.30~0.35	C = 0.6~0.8 MPa φ = 30°~35°	1~3	2350

Table 6: The mechanical parameters of the actual project.

Component	Density (kg/m^3)	Elastic modulus (GPa)	Poisson's ratio
Dam body	2400	25	0.167
Foundation	2520	E_C	0.200

Step 2: select the sample. 221 water-level data, from 125.33 m to 145.96 m, were selected on the dates when the inverted plumb line measured displacement. Because of the unknown actual parameter in the dam foundation, in order to make the training samples contain the possible target, 200 different elastic moduli E_C were selected from 3 GPa to 10 GPa according to the values in Table 5. There were 44,200 groups of combination states of the mechanical parameter and water pressure. The model was calculated using software GeHoMadrid to get the node displacement of all states. The result $[E_C, H, x, y, u_c]$ was stored as samples to train and verify the DNN model in step 4.

Step 3: withdraw the water pressure component. The multivariable linear regression model is shown in the following equation:

$$\text{disp} = \sum_{i=1}^{3} \beta_i H^i + \sum_{j=1}^{2} \left(\beta_{1j} \sin \frac{2\pi jt}{365} + \beta_{2j} \cos \frac{2\pi jt}{365}\right) + C_1 D + C_2 \ln(1 + D) + C_3 \frac{D}{D+1} + C_4\left(1 - e^{-D}\right) + \tau, \tag{17}$$

$$D = \frac{(t - t_0)}{100}. \tag{18}$$

β and C are regression coefficients. H is the water level, while H_0 is the initial value. τ is the random error. t represents the current monitoring date, and t_0 represents the initial monitoring date. The water pressure component δ_H calculated by the MLP model above is the orange line in Figure 22. And it was used as the target displacement u_{true} in the samples calculated in DQN, $[E_C, H, x, y, \delta_H]$, where the initial value of E was determined randomly, H was the actual water level, and (x, y) was the coordinate of node D.

Step 4: construct the DNN surrogate model. The structure and parameters were the same as those of the DNN model in case A. The samples from step 2 were shuffled randomly, and all data were normalized to [0, 1] according to the data features. After that, training samples occupied 70%, 15% of samples were used to verify the DNN model, and the rest were predicting samples.

The iterative process of the training error and verifying error is shown in Figure 23, where it indicated that, during the former 100 epochs, the two errors decreased sharply to the level close to 0. After the 200 epochs, the network parameters were nearly stable. After the training stage, the DNN model was stored to replace the finite element model in the later steps.

Step 5: construct Agent. The structure and parameters in Agent were the same as those in case A.

Step 6: the calculating target was searching the elastic modulus of the dam foundation to minimize the difference between the inversion result and the actual water pressure component. This maximum number of epoch was 200, and each epoch had 100 time steps. The variation of random probability ε was identical to that in case A. The sample volume of the memory zone was 400, the discounted factor γ was 0.5, the learning rate α was 0.5, the adjustment factor E_{step} was 0.02, and the replay size of samples in each time step was 32. 10 GPa which was selected as the initial modulus. The iterative process and result are shown in the following.

3.3.1. Process Analysis. Figures 24 and 25 show that, in the initial period, the model was in the exploration stage, selecting actions randomly, resulting in the fluctuation of the reward. After that, the DQN model moved into the exploitation stage. With the increase of epoch and selecting the right action when facing different states, the absolute value of the reward was decreasing consistently, and the searching parameters kept approaching the target from the initial value 10 GPa in the former 50 epochs before the model was generally stable.

3.3.2. Result Analysis. After the interactive process between Agent and Env, the elastic modulus E_C of the dam foundation was 5.1549 GPa. All calculating results are shown in Figures 22 and 26. The former displayed that the blue line

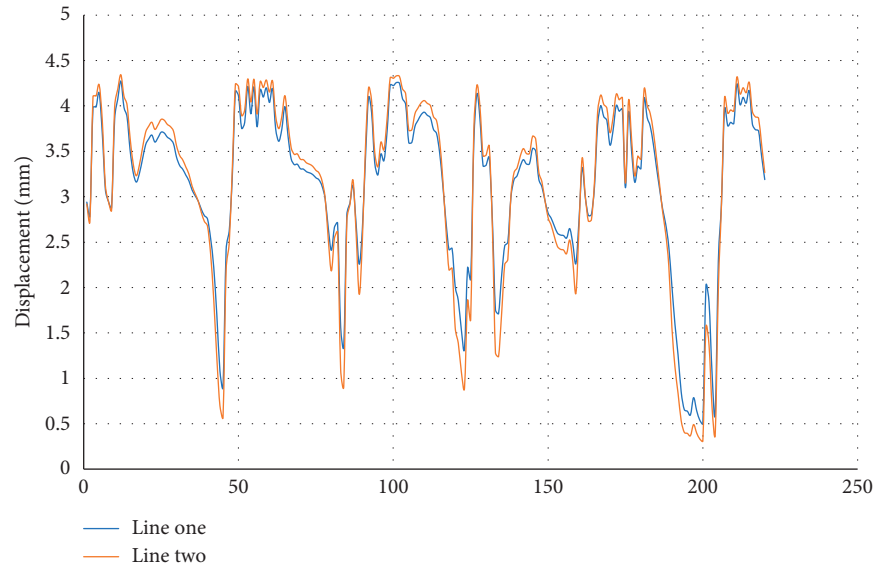

Figure 22: The contradiction between inversion displacement (blue) and water pressure component (orange).

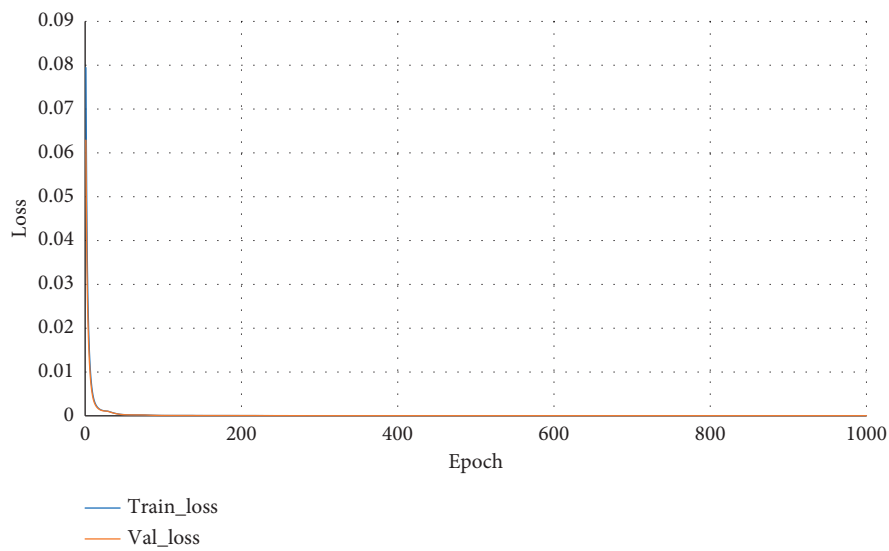

Figure 23: The iterative process of the model error of the engineering model.

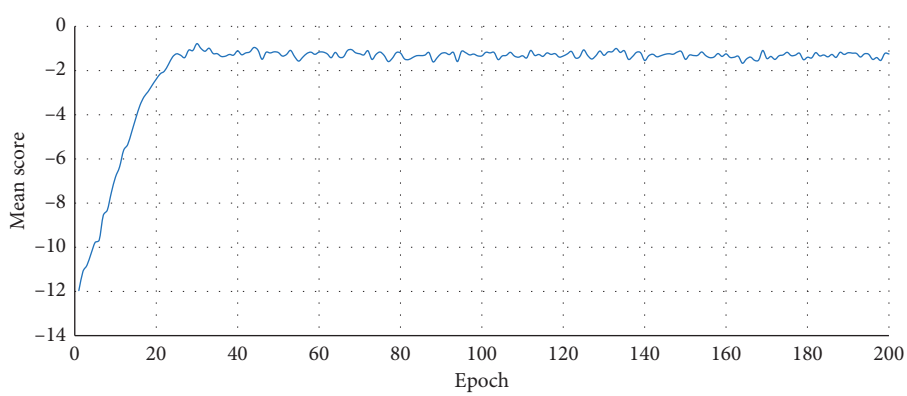

Figure 24: The iterative process of the reward in actual engineering.

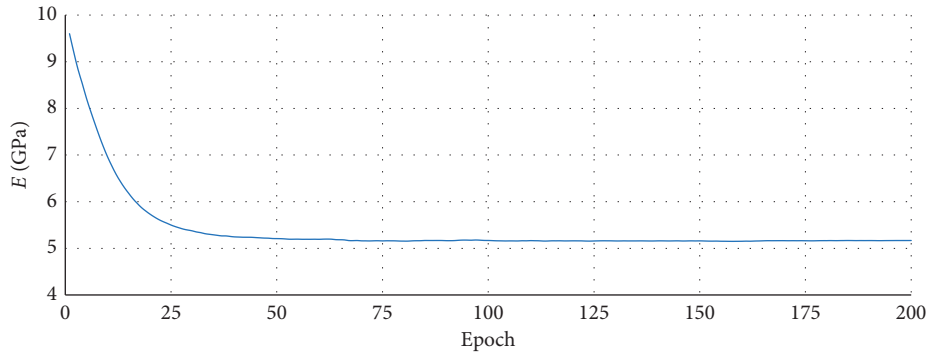

Figure 25: The iterative process of E in actual engineering.

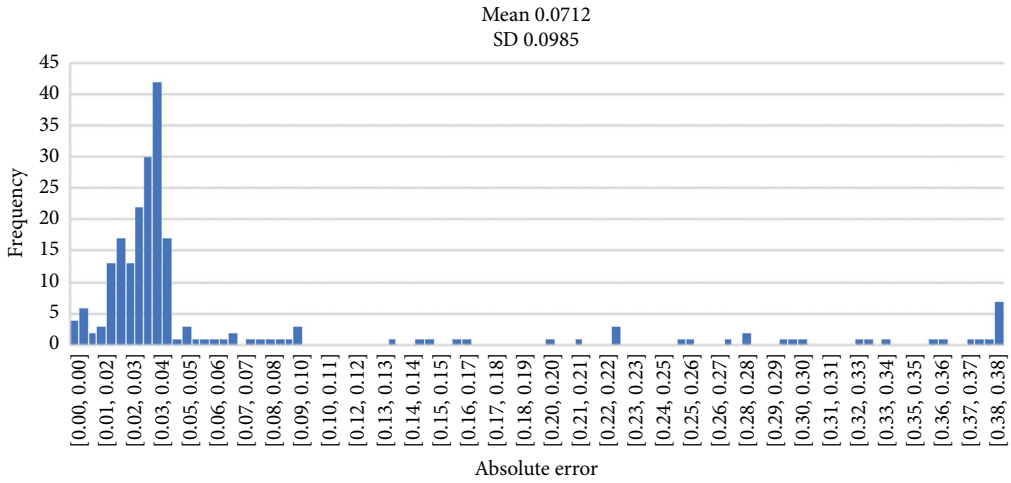

Figure 26: The distribution of the absolute error in the actual project.

indicating the inversion displacement series fitted well with the orange line representing the water pressure component, except a few points with obvious errors, which meant that the displacement values of two lines were close at the same water level on the whole. The latter was the distribution of the absolute error calculated by two displacement series, whose mean value was 0.0712 mm and standard deviation was 0.0985 mm. These errors were mainly concentrated on 0 mm~0.1 mm. A few values reached 0.3 mm~0.4 mm. The error level was low in the mass, which indicated that this method in the paper was suitable to be applied in actual engineering.

4. Conclusion

The accurate calculation of mechanical parameters in the engineering structure and foundation is dependent on detailed monitoring data of the structure and environment, reasonable constitutive model, and excellent searching algorithm. In this paper, the DNN model with a suitable structure replaced the finite element model and was embedded in the agent of the reinforcement algorithm to form the DQN, which was used to optimize the mechanical parameters in engineering in the global space. The conclusions are as follows:

(1) According to the mechanical parameters and environmental loads of engineering, the corresponding DNN surrogate model was established to replace the finite element model. After the network model was verified, the mean relative error of predicting samples calculated by the DNN model with suitable hyperparameters and a regular training stage was lower than 1%, and the calculating efficiency of the DNN was much higher than that of the constitutive model, which indicated that it was advantageous for a reasonable DNN model to map the relation between the target displacement and the state of different mechanical parameters combining with variable environmental loads.

(2) The DNQ algorithm improving the interactive mode between Env and Agent combined with the DNN surrogate model completed the inversion calculation of the structural mechanical parameter. After the improved framework calculated target values in examples, the maximum relative error and the minimum one of the elastic moduli after searching process were 1.29% and 0.18%, respectively. After the improved algorithm was used in actual engineering, the inversion displacement series fitted well with the water pressure component on the whole. Thus, the

DQN algorithm had a good effect in the inversion analysis of mechanical parameters in the hydraulic structure.

(3) The method to express the displacement relation among different dam zones was introduced to ensure the relevance and coordination during the process of optimizing parameters from multizones. This improvement extended the FEM from a single region in case A to a double region in case B, providing a new path for inversion analysis in multiple structural zones.

(4) The research focus is to combine the DNN surrogate model and the improved DQN algorithm and then apply the new model to the inversion calculation of mechanical parameters in the hydraulic structure and foundation with single or multiple zones. In future, the framework could be developed to improve the optimization method applied to inversion analysis in multiple monitoring points and several kinds of mechanical parameters.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Wei Ji contributed to conceptualization, data curation, formal analysis, methodology, software, visualization, writing, reviewing, and editing. Xiaoqing Liu contributed to funding acquisition, investigation, project administration, supervision, writing, review, and editing. Huijun Qi contributed to conceptualization, methodology, software, visualization, writing, review, and editing. Chaoning Lin contributed to investigation and formal analysis. Xunnan Liu contributed to data curation and software. Tongchun Li contributed to resources and project administration.

Acknowledgments

This research was greatly supported by the National Key Research and Development Plan (no. 2018YFC0407102), the Fundamental Research Funds for the Central Universities (SN: B200202180), and the National Natural Science Foundation of China (SN: 52009035).

References

[1] W. Ge, Z. Li, R. Y. Liang, W. Li, and Y. Cai, "Methodology for establishing risk criteria for dams in developing countries, case study of China," *Water Resources Management*, vol. 31, no. 13, pp. 4063–4074, 2017.

[2] A. J. Wiley, "The St. Francis dam failure," *Journal-American Water Works Association*, vol. 20, no. 3, pp. 338–342, 1928.

[3] P. Habib, "The Malpasset dam failure," *Engineering Geology*, vol. 24, pp. 295–329, 1987.

[4] R. Ardito, G. Maier, and G. Massalongo, "Diagnostic analysis of concrete dams based on seasonal hydrostatic loading," *Engineering Structures*, vol. 30, no. 11, pp. 3176–3185, 2008.

[5] R. Ardito, P. Bartalotta, L. Ceriani, and G. Maier, "Diagnostic inverse analysis of concrete dams with statical excitation," *Journal of the Mechanical Behavior of Materials*, vol. 15, no. 6, pp. 381–390, 2004.

[6] C. Lin, T. Li, X. Liu et al., "A deformation separation method for gravity dam body and foundation based on the observed displacements," *Structural Control and Health Monitoring*, vol. 26, no. 2, Article ID e2304, 2019.

[7] S. Chen, C. Gu, C. Lin et al., "Safety monitoring model of a super-high concrete dam by using RBF neural network coupled with kernel principal component analysis," *Mathematical Problems in Engineering*, vol. 2018, Article ID 1712653, , 2018.

[8] C. Lin, T. Li, S. Chen, X. Liu, C. Lin, and S. Liang, "Gaussian process regression-based forecasting model of dam deformation," *Neural Computing and Applications*, vol. 31, no. 12, pp. 8503–8518, 2019.

[9] S. Chen, C. Gu, C. Lin et al., "Multi-kernel optimized relevance vector machine for probabilistic prediction of concrete dam displacement," *Engineering with Computers*, 2020.

[10] C. Lin, T. Li, S. Chen et al., "Structural identification in long-term deformation characteristic of dam foundation using meta-heuristic optimization techniques," *Advances in Engineering Software*, vol. 148, Article ID 102870, 2020.

[11] R. Perera, S.-E. Fang, and A. Ruiz, "Application of particle swarm optimization and genetic algorithms to multiobjective damage identification inverse problems with modelling errors," *Meccanica*, vol. 45, no. 5, pp. 723–734, 2010.

[12] F. Kang, J. Li, and H. Li, "Artificial bee colony algorithm and pattern search hybridized for global optimization," *Applied Soft Computing*, vol. 13, no. 4, pp. 1781–1791, 2013.

[13] F. Kang, J.-s. Li, and J.-j. Li, "System reliability analysis of slopes using least squares support vector machines with particle swarm optimization," *Neurocomputing*, vol. 209, pp. 46–56, 2016.

[14] F. Kang, Q. Xu, and J. Li, "Slope reliability analysis using surrogate models via new support vector machines with swarm intelligence," *Applied Mathematical Modelling*, vol. 40, no. 11-12, pp. 6105–6120, 2016.

[15] F. Kang and J. Li, "Artificial bee colony algorithm optimized support vector regression for system reliability analysis of slopes," *Journal of Computing in Civil Engineering*, vol. 30, no. 3, 2016.

[16] S. Dou, J. Li, and F. Kang, "Parameter identification of concrete dams using swarm intelligence algorithm," *Engineering Computations*, vol. 34, no. 7, 2017.

[17] R. Nian, J. Liu, and B. Huang, "A review on reinforcement learning: Introduction and applications in Industrial process control," *Computers & Chemical Engineering*, vol. 139, Article ID 106886, 2020.

[18] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, Massachusetts, USA, 1998.

[19] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, The MIT Press, Cambridge, Massachusetts, USA, 2018.

[20] R. Bellman, "A Markovian decision process," *Indiana University Mathematics Journal*, vol. 6, no. 4, pp. 679–684, 1957.

[21] M. R. K. Mes and A. P. Rivera, "Approximate dynamic programming by practical examples," *Markov Decision Processes in Practice*, Springer, Cham, Switzerland, 2017, ISBN 978-3-319-47764-0.
[22] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31 st International Conference on Machine Learning*, Beijing, China, 2014.
[23] V. Mnih, "Playing atari with deep reinforcement learning," 2013, http://arxiv.org/abs//1312.5602.
[24] V. Mnih, K. Kavukcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, 2015.
[25] Z. Zhang, A. Chong, Y. Pan, C. Zhang, and K. P. Lam, "Whole building energy model for HVAC optimal control: a practical framework based on deep reinforcement learning," *Energy and Buildings*, vol. 199, pp. 472–490, 2019.
[26] Z. Wang and T. Hong, "Reinforcement learning for building controls: the opportunities and challenges," *Applied Energy*, vol. 269, Article ID 115036, 2020.
[27] Z. Bing, C. Lemke, and L. Cheng, "Energy-efficient and damage-recovery slithering gait design for a snake-like robot based on reinforcement learning and inverse reinforcement learning," *Neural Networks*, vol. 129, pp. 323–333, 2020.
[28] I. Carlucho, M. De Paula, and G. Acosta, "An adaptive deep reinforcement learning approach for MIMO PID control of mobile robots," *ISA Transactions*, vol. 102, 2020.
[29] J. García and D. Shafie, "Teaching a humanoid robot to walk faster through safe reinforcement learning," *Engineering Applications of Artificial Intelligence*, vol. 88, Article ID 103360, 2020.
[30] F. Li, Q. Jiang, S. Zhang, M. Wei, and R. Song, "Robot skill acquisition in assembly process using deep reinforcement learning," *Neurocomputing*, vol. 345, pp. 92–102, 2019.
[31] F. Chang, T. Chen, W. Su, and Q. Alsafasfeh, "Control of battery charging based on reinforcement learning and long short-term memory networks," *Computers & Electrical Engineering*, vol. 85, Article ID 106670, 2020.
[32] Z. Wu, *Safety Monitoring Theory & its Application of Hydraulic Structures*, Higher Education Press, Beijing, China, 2003.
[33] K. Hornik, M. Stinchcombe, H. White et al., "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
[34] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.
[35] J. Yang, J. Dai, C. Yao, S. Jiang, C. Zhou, and Q. Jiang, "Estimation of rock mass properties in excavation damage zones of rock slopes based on the Hoek-Brown criterion and acoustic testing," *International Journal of Rock Mechanics and Mining Sciences*, vol. 126, Article ID 104192, 2020.
[36] Z. Z. Wang, Y. J. Jiang, and C. A. Zhu, "Seismic energy response and damage evolution of tunnel lining structures," *European Journal of Environmental and Civil Engineering*, vol. 23, no. 6, pp. 758–770, 2019.