

A Study of Unmanned Path Planning Based on a Double-Twin RBM-BP Deep Neural Network

Xuan Chen^{1,*}, Zhiping Wan¹ and Jiatong Wang²

¹Zhejiang Industry Polytechnic College, Shaoxing, 312000, China

²Kunsan National University, Jeollabuk-do, 54150, Korea

*Corresponding Author: Xuan Chen. Email: chenxuan1979@sina.com

Received: 26 May 2020; Accepted: 27 August 2020

Abstract: Addressing the shortcomings of unmanned path planning, such as significant error and low precision, a path-planning algorithm based on the whale optimization algorithm (WOA)-optimized double-blinking restricted Boltzmann machine-back propagation (RBM-BP) deep neural network model is proposed. The model consists mainly of two twin RBMs and one BP neural network. One twin RBM is used for feature extraction of the unmanned path location, and the other RBM is used for the path similarity calculation. The model uses the WOA algorithm to optimize parameters, which reduces the number of training sessions, shortens the training time, and reduces the training errors of the neural network. In the MATLAB simulation experiment, the proposed algorithm is superior to several other neural network algorithms in terms of training errors. The comparison of the optimal path under the simulation of complex road conditions shows the superior performance of this algorithm. To further test the performance algorithm introduced in this paper, a flower bed, computer room and other actual scenarios were chosen to conduct path-planning experiments for unmanned paths. The results show that the proposed algorithm has obvious advantages in path selection, reducing the running time and improving the running efficiency. Therefore, it has definitive practical value in unmanned driving.

Keywords: Twin RBM-BP; WOA; path planning

1 Introduction

In recent years, driverless driving has been the focus of research in the automotive industry in various countries [1–3]. Yi et al. [4] proposed a design framework for unmanned driving trajectory planning based on real-time maneuvering decisions, dividing the trajectory space into homotopic regions, and linearizing the trajectory. The simulation experiment takes extreme conditions as the research object, which can effectively avoid accidents. Rajurkar et al. [5] proposed an optimal path-planning scheme for automatic vehicles based on a genetic algorithm to optimize the fuzzy controller. An automatic driving model based on neural networks has been proposed by NA. Neural networks have been widely used in many fields [6–9]. Li et al. [10] introduced deep learning into unmanned driving path planning and predicted the optimal path through a deep learning algorithm to guide the vehicle forward. Sallab et al. [11], Isele et al.



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

[12], Pan et al. [13], Xia et al. [14], Xiong et al. [15] and others used deep reinforcement learning for unmanned driving and achieved good results. The problem with this type of research is that the number of iterations is vast. Generally, it can require more than 1000 sessions to obtain better training results, which increases the algorithm complexity. The training process also requires substantial data support.

For these problems, the paper has proposed a double twin RBP-BP deep learning neural network model based on WOA optimization. In this algorithm, a twin RBP model is used to process the feature map of the unmanned driving path, and another twin RBM model is used to process the path similarity. The WOA algorithm is used to optimize the twin RBM parameters in the deep learning network so that the deep learning neural network training error and the optimal parameters are obtained, which offers better prediction effects.

2 Related Knowledge

2.1 Restricted Boltzmann Machine

The restricted Boltzmann machine [16] is a neural network model based on statistical mechanics and an energy model. It uses energy to represent the stable state of the entire system. The smaller the energy usage is, the more stable the system is. Otherwise, it shows that the system is in a certain state of fluctuation and instability. It has the characteristics of a fast learning rate and strong learning ability. It is an important model in the field of deep learning. The structure is shown in Fig. 1.

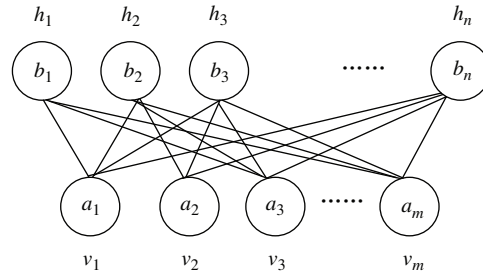


Figure 1: RBM structure

In Fig. 1 v refers to the visible layer. Data samples are imported from the visible layer. h refers to the hidden layer. It represents the characteristics of the extracted data. Suppose that the RBM network has m visible units and n hidden units. The vectors $v = (v_1, v_2, \dots, v_m)^T$ and $h = (h_1, h_2, \dots, h_n)^T$ refer to the status of the unit at the visible layer and the hidden layer. Suppose $a = (a_1, a_2, \dots, a_m)^T$ is the offset vector at the visible layer, and a_i refers to the offset of the i visible unit at the visible layer. Suppose $b = (b_1, b_2, \dots, b_n)^T$ is the offset vector at the hidden layer, and $W \in R^{m \times n}$ is the weighted matrix between the visible layer and the hidden layer. $w_{i,j}$ is the weight of the j node and the i node at the visible layer. The process of RBM training is to learn to obtain the parameter $\theta = \{a_i, b_j, W_{ij}\}$:

$$E(v, h) = - \sum_{i \in v} a_i v_i - \sum_{i \in h} b_i h_i - \sum_{i,j} v_i h_j w_{ij} \quad (1)$$

In Eq. (1), $E(v, h)$ refers to the energy function of RBM. The network assigns a probability to each pair of visible and hidden vectors through the energy transfer function:

$$p(v, h) = \frac{1}{Z} e^{-E(v, h)} \quad (2)$$

where Z is the partition function and its expression is:

$$Z = \sum_{v, h} e^{-E(v, h)} \quad (3)$$

Eq. (3) uses the following Eq. (4) to express the logarithmic gradient of weights:

$$\frac{\partial \log p(v, h)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \quad (4)$$

where $\langle v_i h_j \rangle_{data}$ is the expected data, and $\langle v_i h_j \rangle_{model}$ is the expected model. Then, the RBM learning rules can be obtained as follows:

$$\Delta w_{ij} = \varepsilon \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \right) \quad (5)$$

For data expectation, since there is no direct connection between RBM hidden layer units, an unbiased sample of data distribution can be quickly obtained. Suppose randomly given training images are v , the probability of the binary state of the hidden layer unit being is:

$$p(h_j = 1|v) = \sigma \left(b_j + \sum_i v_i w_{ij} \right) \quad (6)$$

For the same reason, the probability of the binary state of the visible layer unit is set to 1:

$$p(v_i = 1|h) = \sigma \left(a_i + \sum_j h_j w_{ij} \right) \quad (7)$$

In summary, the main objective of RBM is to calculate $\theta = \{a_i, b_j, W_{ij}\}$. This is mainly obtained by calculating the maximum log likelihood of RBM on the learning set. The calculation Eq. (8) is as follows:

$$\theta^* = \arg \max_{\theta} \left(\sum_{t=1}^T \log \left(P(v^{(t)}|\theta) \right) \right) \quad (8)$$

In order to obtain the optimal value of Eq. (8), it is calculated by the gradient ascent method [17]. The calculation process is as follows:

$$L(\theta) = \sum_{t=1}^T \log \left(P(v^{(t)}|\theta) \right) = \sum_{t=1}^T \left(\log \sum_h \exp \left[-E(v^{(t)}, h|\theta) \right] - \log \sum_v \sum_h \exp \left[-E(v, h|\theta) \right] \right) \quad (9)$$

For the calculation of Eq. (9), the literature [18] is used to calculate the derivative, and the calculation process is as follows:

$$\begin{aligned}
\frac{\partial L}{\partial \theta} &= \sum_{t=1}^T \frac{\partial}{\partial \theta} \left(\log \sum_h \exp[-E(v^{(t)}, h|\theta)] - \log \sum_v \sum_h \exp[-E(v, h|\theta)] \right) \\
&= \sum_{t=1}^T \left(\sum_h \frac{\exp[-E(v^{(t)}, h|\theta)]}{\sum_h \exp[-E(v^{(t)}, h|\theta)]} \times \frac{\partial[-E(v^{(t)}, h|\theta)]}{\partial \theta} - \sum_v \sum_h \frac{\exp[-E(v, h|\theta)]}{\sum_v \sum_h \exp[-E(v, h|\theta)]} \times \frac{\partial[-E(v, h|\theta)]}{\partial \theta} \right) \\
&= \sum_{t=1}^T \left(\left\langle \frac{\partial[-E(v^{(t)}, h|\theta)]}{\partial \theta} \right\rangle_{P(h|v^{(t)}, \theta)} - \left\langle \frac{\partial[-E(v, h|\theta)]}{\partial \theta} \right\rangle_{P(h|v, \theta)} \right) \quad (10)
\end{aligned}$$

In Eq. (10), $P(h|v^{(t)}, \theta)$ means the probability distribution of the hidden layer when the visible unit is set to uniform learning sample $v^{(t)}$, $P(h|v, \theta)$ means the joint distribution of visible units and hidden layer units. According to the two distributions, the offset number distribution of parameter $\theta = \{a_i, b_j, W_{ij}\}$ is:

$$\frac{\partial P(v|\theta)}{\partial W_{ij}} = \langle v_i h_j \rangle_{P(h|v^{(t)}, \theta)} - \langle v_i h_j \rangle_{P(h|v, \theta)} \quad (11)$$

$$\frac{\partial P(v|\theta)}{\partial a_i} = \langle v_i \rangle_{P(h|v^{(t)}, \theta)} - \langle v_i \rangle_{P(h|v, \theta)} \quad (12)$$

$$\frac{\partial P(v|\theta)}{\partial b_j} = \langle h_j \rangle_{P(h|v^{(t)}, \theta)} - \langle h_j \rangle_{P(h|v, \theta)} \quad (13)$$

Eqs. (11)–(13) are RBM learning rules.

2.2 Whale Optimization Algorithm

The whale optimization algorithm [19] uses a set of search agents to determine the global optimal solution of the optimization problem. The search process for a given problem begins with a set of random solutions, and the candidate solutions are updated through optimization rules until the end conditions are met. The whale algorithm is divided into three stages: predation, bubble attack and food search.

(1) Surround predator

In the initial stage of the algorithm, humpback whales do not know where the food is; they all obtain the position information of the food through group cooperation. Other whale individuals will approach this position and gradually surround the food, so the following mathematical model is used:

$$\vec{D} = |\vec{C} \bullet \vec{X}^*(t) - \vec{X}(t)| \quad (14)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \bullet \vec{D} \quad (15)$$

In the Eq. (14), \vec{D} refers to the distance vector from the search agent to the target food, and t refers to the current iteration times. \vec{C} and \vec{A} are coefficient vectors, X^* is the local optimal solution, \vec{X} is the place vector, and \vec{C} and \vec{A} are expressed as follows:

$$\vec{C} = 2 \bullet \vec{r} \quad (16)$$

$$\vec{A} = 2 \vec{a} \bullet \vec{r} - \vec{a} \quad (17)$$

In the Eq. (17), \vec{a} refers to the linearly decreasing vector from 2 to 0, and r is a random number between 0 and 1.

(2) Bubble attack

In this stage, the humpback whale was used to attack the bubble, and the behavior of the whale preying and spitting out the bubble was designed by shrinking the surroundings and spirally updating the position to achieve the goal of local whale optimization.

1) Shrinking and surrounding principle

When $|A| < 1$, the individual whale approaches the best whale at its current location, and the larger $|A|$ is, the faster the pace of the whale.

2) Spiral update position

The individual humpback whale first calculates the distance from the current optimal whale and then swims in a spiral. When searching for food, the mathematical model of the spiral is:

$$\vec{X}(t+1) = \vec{D} \bullet e^{lb} \bullet \cos(2\pi l) + \vec{X}^*(t) \quad (18)$$

In the Eq. (18), b is the constant coefficient, and l is a random vector between 0 and 1.

3) Food searching stage

Humpback whales obtain good results by controlling the $|A|$ vector. When $|A| > 1$, individual humpback whales have approached the position of the reference humpback whale, and the individual whale has updated its position toward the randomly selected humpback whale. The model is expressed as follows:

$$\vec{D} = |\vec{C} \bullet \vec{X}_{rand} - \vec{X}(t)| \quad (19)$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \bullet \vec{D} \quad (20)$$

In the Eqs. (19) and (20), \vec{X}_{rand} is a randomly obtained position vector of the reference humpback whale.

3 WOA-Optimized Dual Twin RBM-BP Neural Networks' Unmanned Driving Path-Planning Algorithm

3.1 Theoretical Analysis of WOA Optimized Double-Twin RBM-BP

Because RBM parameter design is an extremely complicated task, RBM has no rules to follow during design, and it is difficult to ensure the optimization of the network. The parameter learning rate in the double-twin RBM-BP deep learning neural networks, the number of visible layers v , the number of hidden layers h , and the parameter set jointly determine the performance of the double-twin RBM-BP neural networks. Therefore, the parameter design complexity is more complicated than the traditional single RBM. Optimizing these six parameters reasonably is the key to the effectiveness of the model in this paper. Using genetic algorithms, particle swarm optimization and other biomimetic algorithms to optimize BP neural network parameters can improve its performance. Thus, this paper uses the more powerful WOA algorithm for parameter optimization of the twin RBM-BP deep learning neural networks. The parameters v , h , and a total of 12 sets of parameters in the twin RBM-BP are optimized by the WOA algorithm.

3.2 Training Model Based on the Double Twin RBM-BP Neural Network

The RBM-BP combines the advantages of RBM and BP and targets complex and high-dimensional network traffic data. We use RBM's strong feature learning ability and unsupervised learning of high-dimensional data to remove redundant features and reduce data complexity. This can reduce the training

complexity of the data and improve the recognition accuracy of the deep learning network. However, the RBM-BP networks require huge calculations and have shortcomings regarding the training sample library. In this paper, we will use double twin RBM-BP networks to achieve the use of a small number of layers to reduce the number of training sessions. The structure of twin RBM-BP is shown in Fig. 2.

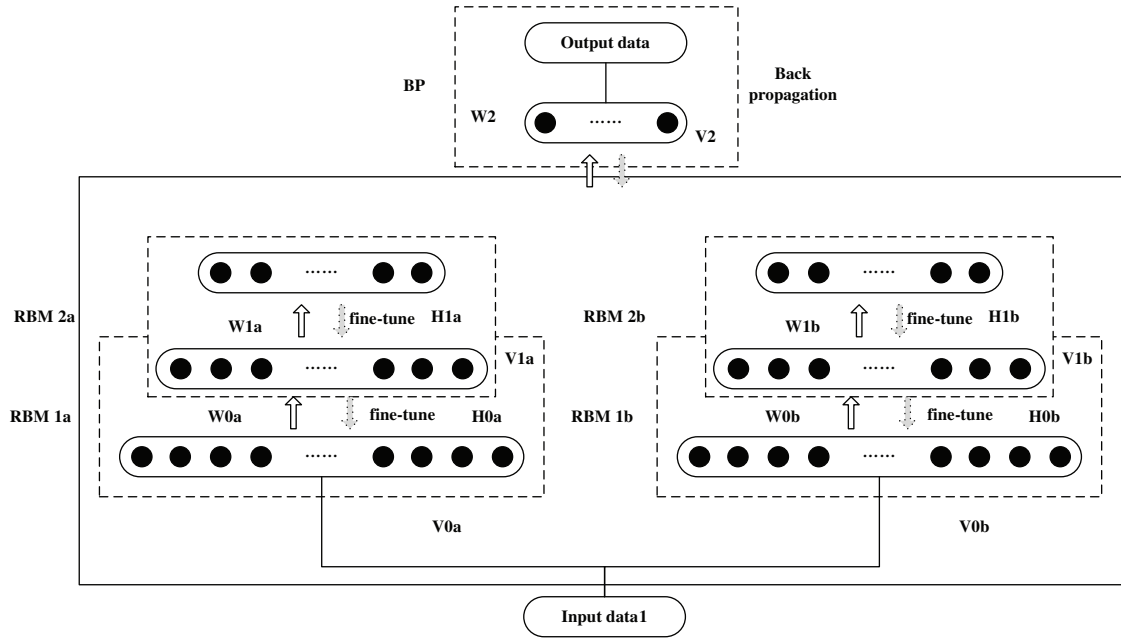


Figure 2: Double twin RBM-BP deep learning neural networks

3.3 WOA-Optimized RBM-BP Parameter

During the predation phase, \vec{X} refers to the location vector. In this paper, the variables that need to be optimized include the corresponding vectors of the parameter learning rate ε , number of visible layers v , number of hidden layers h and parameter set $\theta = \{a_i, b_j, W_{ij}\}$. Therefore, \vec{X} in Eq. (15) can be expressed as follows:

$$\vec{X} = [\varepsilon_1, v_1, h_1, \{a_i, b_j, W_{ij}\}_1, \varepsilon_2, v_2, h_2, \{a_i, b_j, W_{ij}\}_2] \quad (21)$$

In the spiral phase, $\vec{X} = [\varepsilon_1, v_1, h_1, \{a_i, b_j, W_{ij}\}_1, \varepsilon_2, v_2, h_2, \{a_i, b_j, W_{ij}\}_2]$ the optimization objective function involved in the calculation is the training error of the twin RBM deep learning neural networks, so it is expressed as:

$$f_{obj} = f(\vec{X}) = f(\varepsilon_1, v_1, h_1, \{a_i, b_j, W_{ij}\}_1, \varepsilon_2, v_2, h_2, \{a_i, b_j, W_{ij}\}_2) \quad (22)$$

Through optimization, the parameters $v_{1,opt}$, $h_{1,opt}$, $\varepsilon_{1,opt}$ and $\theta = \{a_i, b_j, W_{ij}\}_{1,opt}$ and $v_{2,opt}$, $h_{2,opt}$, $\varepsilon_{2,opt}$ and $\theta = \{a_i, b_j, W_{ij}\}_{2,opt}$ can be obtained. Apply the value of the two groups of the optimal parameters $[\varepsilon, v, h, \{a_i, b_j, W_{ij}\}]$ to the twin RBM deep learning neural networks to complete the optimization of the neural networks. Optimize two sets of parameters of twin RBM networks through WOA $[\varepsilon, v, h, \{a_i, b_j, W_{ij}\}]$ to minimize the training errors.

3.4 Path Planning

After optimizing the RBM-BP deep learning neural network parameters through the WOA algorithm, the following two sets of RBM models and their corresponding learning rules can be obtained:

$$E(v, h) = - \sum_{i \in v_{1,opt}} a_{i,1,opt} v_{i,1,opt} - \sum_{i \in h_{1,opt}} b_{i,1,opt} h_{i,1,opt} - \sum_{i,j} v_{i,1,opt} h_{i,1,opt} w_{ij,1,opt} \quad (23)$$

$$E(v, h) = - \sum_{i \in v_{2,opt}} a_{i,2,opt} v_{i,2,opt} - \sum_{i \in h_{2,opt}} b_{i,2,opt} h_{i,2,opt} - \sum_{i,j} v_{i,2,opt} h_{i,2,opt} w_{ij,2,opt} \quad (24)$$

$$\Delta w_{ij} = \varepsilon_{1,opt} \left(\langle v_{i,1,opt} h_{j,1,opt} \rangle_{data} - \langle v_{i,1,opt} h_{j,1,opt} \rangle_{model} \right) \quad (25)$$

$$\Delta w_{ij} = \varepsilon_{2,opt} \left(\langle v_{i,2,opt} h_{j,2,opt} \rangle_{data} - \langle v_{i,2,opt} h_{j,2,opt} \rangle_{model} \right) \quad (26)$$

Eqs. (23) and (25) represent the model and learning rules of the first RBM in the twin RBM, which is mainly used for feature extraction of the input map data. After entering the map data matrix, neural networks are trained according to the obstacle area and non-obstacle area in the map. The training goal is to train with the shortest distance.

Eqs. (24) and (26) represent the model and learning rules of the second RBM in the twin RBM because there may be many different routes under the shortest distance between the starting point and the end point given at random. Therefore, it is necessary to analyze the similarity of these routes. In the route selection, routes with a small number of turns should be considered as much as possible, even if the number of turns is the same, as far as possible, and the angle of the turns is small. Therefore, the closeness between the optimal driving decision made by the twin RBM, and the actual target point is fed back to the neural network system as a feedback value.

3.5 Algorithm Process

According to the process described above, the design process of the algorithm in this paper is shown in Tab. 1:

Table 1: Design process of the algorithm in this paper

Input: RBM-BP related parameters, whale algorithm related parameters, route start and end points, number of iterations, map data matrix
Output: Route planning of the model
Step 1: Initialize the whale population
$\vec{X} = [\varepsilon_1, v_1, h_1, \{a_i, b_j, W_{ij}\}_1, \varepsilon_2, v_2, h_2, \{a_i, b_j, W_{ij}\}_2]$
Step 2: Calculate the corresponding fitness value of each individual whale
$f_{obj} = f(\vec{X}) = f(\varepsilon_1, v_1, h_1, \{a_i, b_j, W_{ij}\}_1, \varepsilon_2, v_2, h_2, \{a_i, b_j, W_{ij}\}_2)$
Step 3: Save X * as the best whale
$\vec{X}^* = [\varepsilon_1, v_1, h_1, \{a_i, b_j, W_{ij}\}_1, \varepsilon_2, v_2, h_2, \{a_i, b_j, W_{ij}\}_2]^*$
Step 4: For loop iteration optimization.
Step 5: For every whale, calculate the corresponding fitness value
$f_{obj} = f(\vec{X}) = f(\varepsilon_1, v_1, h_1, \{a_i, b_j, W_{ij}\}_1, \varepsilon_2, v_2, h_2, \{a_i, b_j, W_{ij}\}_2)$

(Continued)

Table 1 (continued).

Step 6: If the newly calculated fitness value $fitness < \text{current minimum value}$
 $f_{obj,new} < \min\{f_{obj}\}$
Step 7: Replace the best whale
 $X = X_{new}$
End if. End for.
Step 8: Spawn new individuals. Encircle
 $\vec{D} = |\vec{C} \cdot \vec{X}_{rand} - \vec{X}(t)|$
 $\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D}$
Step 9: Conduct bubble attack
 $\vec{X}(t+1) = \vec{D} \bullet e^{lb} \bullet \cos(2\pi l) + \vec{X}^*(t)$
Step 10: Search food. Update the location of all whales
Step 11: After WOA optimization, obtain optimal parameters of twin RBM model
 $E_1(v, h) = - \sum_{i \in v_{1,opt}} a_{i,1,opt} v_{i,1,opt} - \sum_{i \in h_{1,opt}} b_{i,1,opt} h_{i,1,opt} - \sum_{i,j} v_{i,1,opt} h_{i,1,opt} w_{ij,1,opt}$
 $E_2(v, h) = - \sum_{i \in v_{2,opt}} a_{i,2,opt} v_{i,2,opt} - \sum_{i \in h_{2,opt}} b_{i,2,opt} h_{i,2,opt} - \sum_{i,j} v_{i,2,opt} h_{i,2,opt} w_{ij,2,opt}$

4 Simulation Experiment

To further illustrate the route planning effect of the algorithm in this paper, the basic RBM-BP deep neural networks, CNN networks, Q-Learning network and the improved Q-Learning network are compared with the algorithm in this paper (as shown in Tab. 2). The content of the comparison includes the comparison of training error, optimal route length under complex road conditions and route planning effect. In the experiment part, the hardware platform is CPU i7-9700, memory 32GDDR3, Wins 10, simulation software MATLAB 2019. Take the unmanned driving smart car in Fig. 3 as the research object. The car consists of an FPGA chip of model zynq7020 (as shown in Fig. 4), a CCD camera, four infrared sensors, eight ultrasonic sensors, eight hall sensors (for detecting driving speed), four angle sensors (detection of steering), and an acceleration sensor. The circuit is shown in Fig. 5. To improve the effect of the simulation, the movement direction of the car is expressed as up, down, left, top left, bottom left, top right, bottom right, and eight directions. Set the motion space of the trolley to a two-dimensional plane and simulate the environment area as a 20×20 grid (as shown in Fig. 6).

Table 2: The main relevant parameters of the algorithm and comparison algorithm in this paper

Parameter	Algorithm in this paper	RBM-BP algorithm	CNN	Q-learning	Advance Q-learning
Number of visible layers v	[3,3]	4	4	4	4
Number of hidden layers h	[3,3]	4	4	4	4
Learning rate ε	[0.1330,0.1453]	0.1	0.1	0.1	0.1
$\{a_i, b_j, W_{ij}\}$	A/W:0 matrix B: random matrix	Random initial value	Random initial value	Random initial value	Random initial value

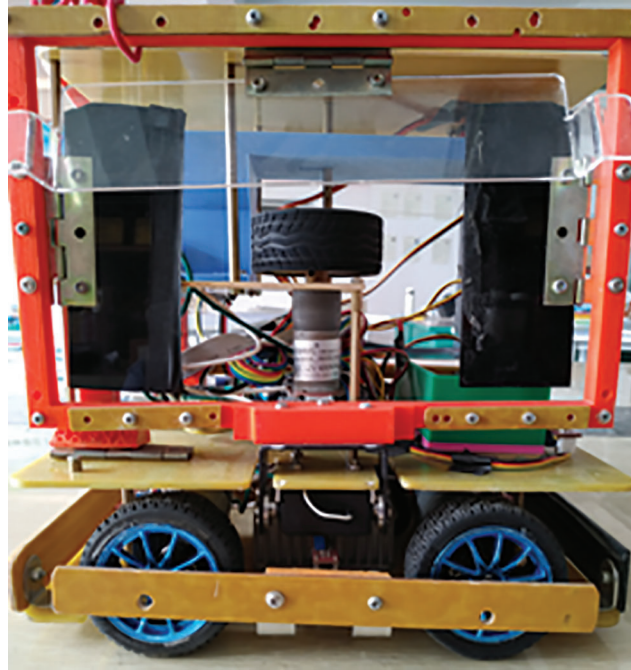


Figure 3: Unmanned driving intelligent car

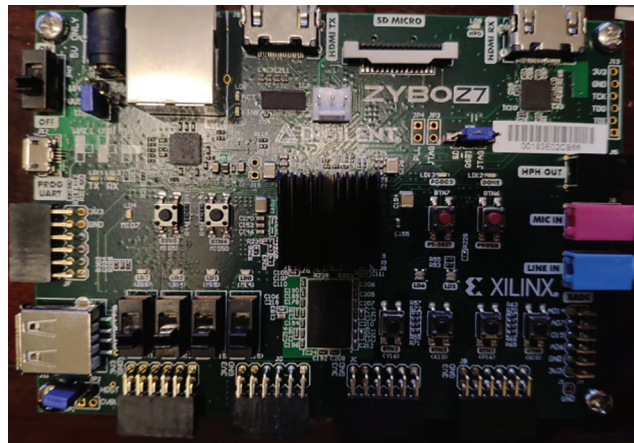


Figure 4: zynq7020 chip

4.1 Comparison of Optimization Indexes of Neural Networks under Different Algorithms

To verify the advantages of this algorithm in optimizing the parameters of double-stacked RBM-BP neural networks, this paper optimizes and contrasts the double-stacked RBM-BP neural networks using the genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO), and WOA algorithm. The source of the dataset is UCI wine quality [20]. Randomly select 1000 data points as the training dataset and select 100 data points from the remaining data points as the test dataset. All data of the training dataset and the test dataset are normalized between $[-1,1]$. Adopt the average variance mean square error (MSE) and mean absolute percentage error (MAPE). N is the total number of samples, y_n is the predicted value, and t_n is the actual value. The Eq is as follows:

reaches convergence, while the basic RBM-BP only reaches the convergence value when the number of iterations reaches approximately 380, and CNN reaches convergence when the number of iterations is 396. Q-Learning reached convergence at 392 iterations, and improved Q-Learning reached convergence at 386 iterations. This shows that the algorithm in this paper uses twin RBM-BP to train the input data and WOA optimizes the neural network parameters to effectively reduce the number of training sessions, reduce the error rate, and achieve better training goals.

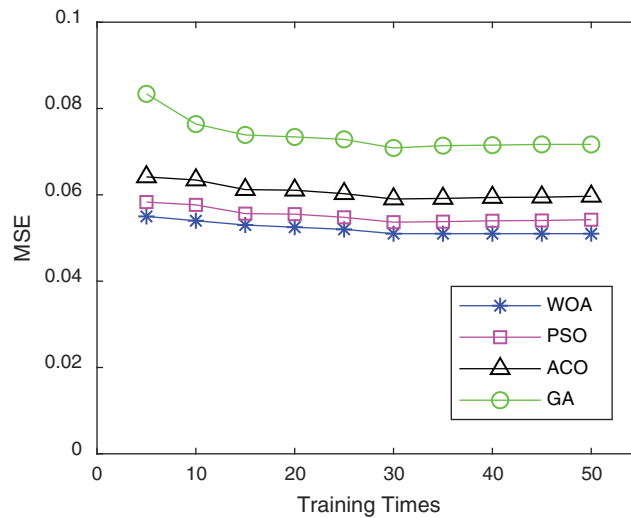


Figure 7: MSE comparison

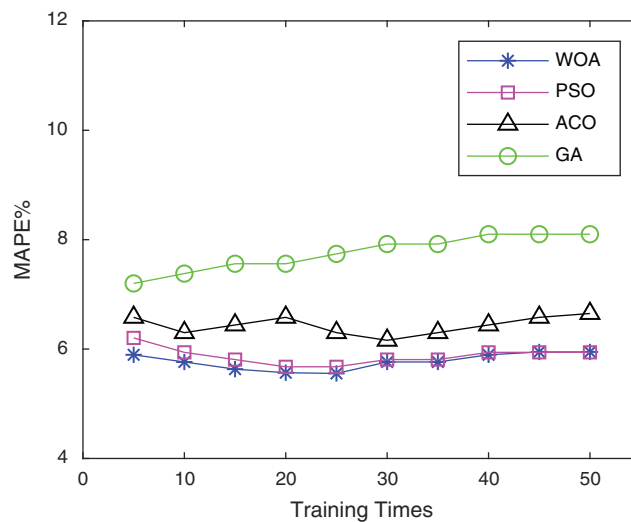


Figure 8: MAPE comparison

4.3 Comparison of Path Lengths under Complex Road Conditions

Fig. 10 shows a schematic diagram of simulating a complex path, and Fig. 11 shows the path-planning effect of five algorithms in simulating complex road conditions. From the trajectory shown in the figure, the path length corresponding to the algorithm in this paper has obvious advantages. Compared with the RBM-BP, CNN networks, Q-Learning network and improved Q-Learning network are reduced by 16%, 28%, 16%

and 8%, respectively. Fifty different sets of starting and ending points are randomly selected in the same scene, and the result of the average path is shown in Fig. 12. It can be found that the algorithm in this paper has certain advantages under the conditions of complex road conditions, mainly because the whale algorithm is optimized, so that the optimization effect of the parameters of the double-stacked RBM-BP networks is significantly improved. Therefore, it has a good planning effect under complex road conditions.

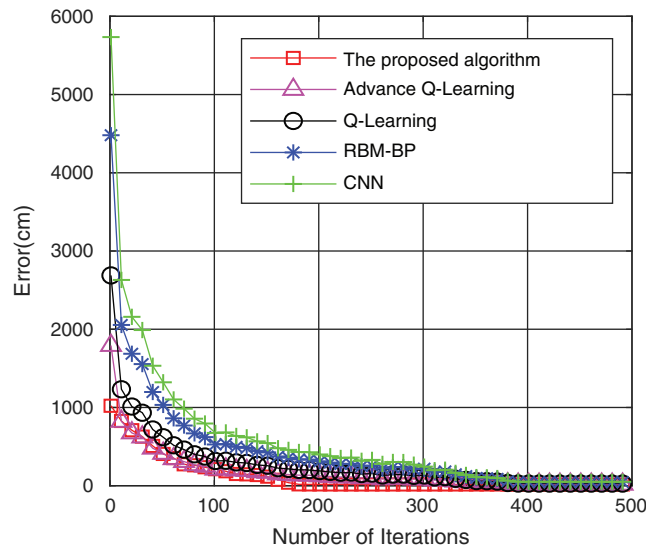


Figure 9: Comparison of the training errors of the five algorithms

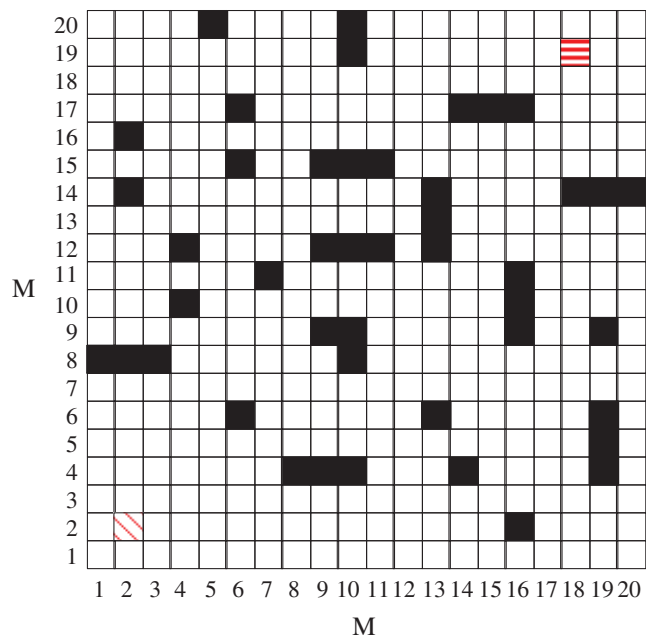


Figure 10: Schematic diagram of the complex path

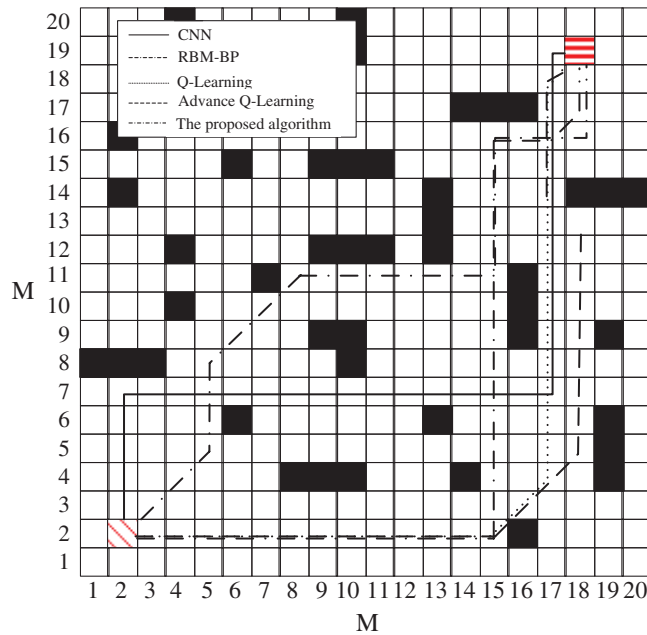


Figure 11: Schematic diagrams of five algorithm complex path trajectories

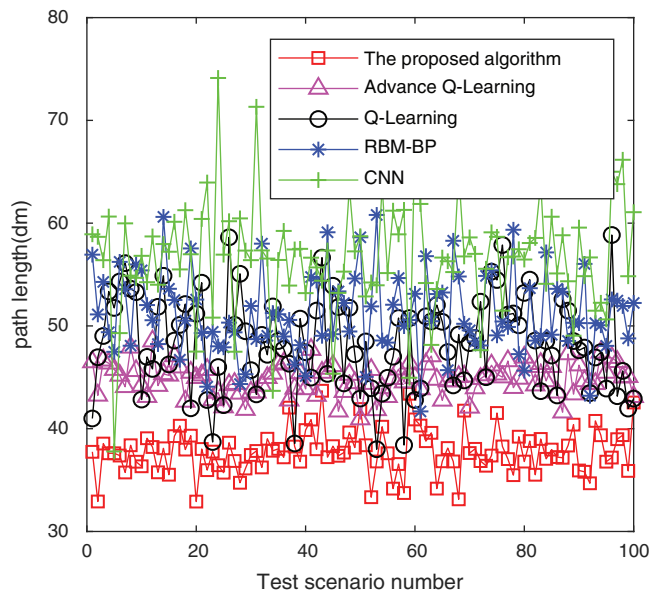


Figure 12: Comparison of 100 random starting and ending paths

4.4 Comparison of Path Lengths under Real Road Conditions

To further illustrate the effect of the algorithm in the real environment in this paper, the flower bed and the machine room were selected as comparative scenes, as shown in Figs. 13 and 14. The flower bed consists of 6 cylindrical flower ponds, which can imitate simple path conditions, and the machine room can imitate complex path conditions. The floor plans in the two scenes traveled under the five algorithms are shown in Figs. 15 and 16.



Figure 13: Garden scene



Figure 14: Computer room scene

It is found from Figs. 15 and 17 that the simulated route of the algorithm in this paper almost matches the actual route traveled, which shows that the algorithm of this paper has a good result in path planning in actual scenarios. Figs. 16 and 18 show the comparison of the actual routes traveled by the five algorithms in the two scenarios. In Fig. 15, the algorithm of this paper circumvents the obstacle and goes directly to the end point. The trajectory does not show too much during the driving process. The curve is complete, and the RBM-BP route trajectory fluctuates greatly. The CNN curve has experienced two obstacles. The curve effect of Q-Learning and improved Q-Learning is slightly worse than that of the algorithm in this paper. The algorithm in Fig. 17 can circumvent obstacles and rush to the end in time, and the trajectory alters the least in the whole process, which shows that the planning effect of the algorithm in this paper is good. The path lengths of the CNN, RBM-BP and the improved Q-Learning are significantly greater than those of the algorithm in this paper. From the comparison of the results in the above figures, it has been shown that the algorithm in this paper has a better route planning effect in actual scenarios.

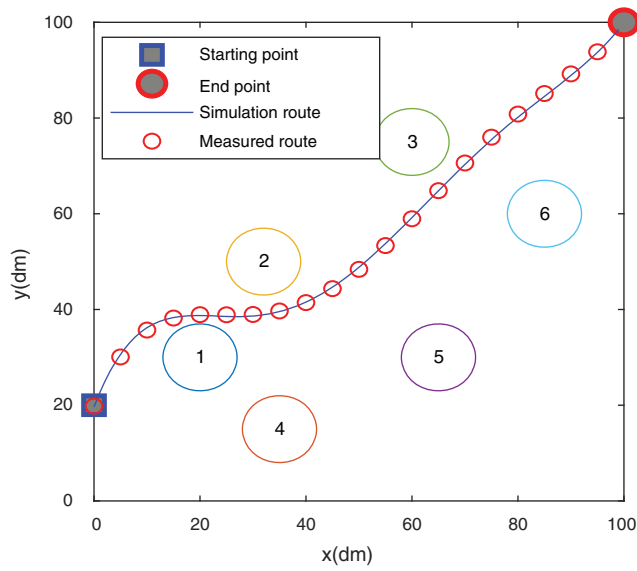


Figure 15: The trajectory in the flower bed

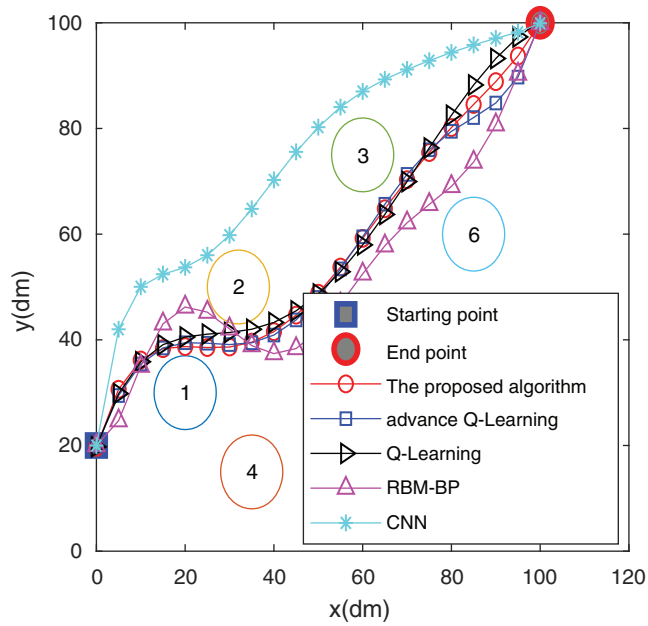


Figure 16: The trajectory in the flower bed

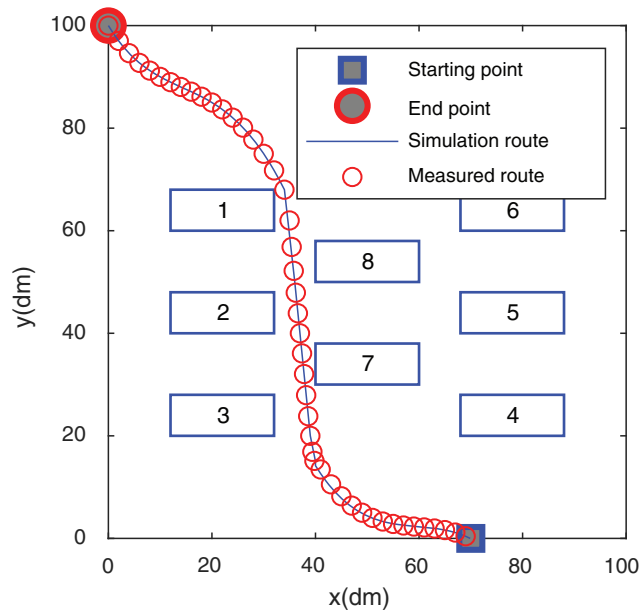


Figure 17: The trajectory in the computer room

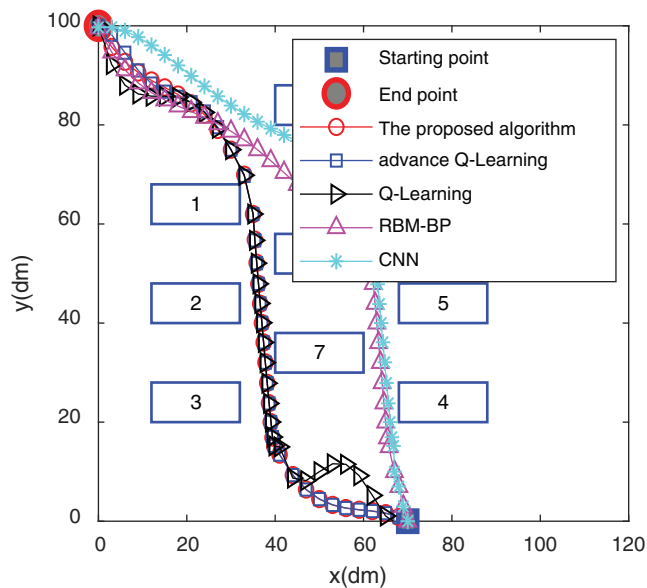


Figure 18: The trajectory in the computer room

5 Conclusion

In view of the problems of large error and long running time in the path planning of unmanned driving, the RBM-BP deep neural network algorithm based on double superposition is proposed. The algorithm calculates the characteristics and similarity of the path map through two superimposed RBMs. Then, the output feature vectors in the hidden layers of the two superimposed RBM networks are used as input feature vectors of BP neural networks for training and learning again. Finally, the whale algorithm is used

to optimize multiple parameters, such as the number of visible layers, hidden layers and the learning rate, which reduces the training time and training errors. Simulation experiments show that the algorithm has a good path-planning result.

Funding Statement: This work is supported by the Public Welfare Technology Research Project of the Science and Technology Department of Zhejiang Province and Research on Key Technologies for Power Lithium-ion Battery Second Use in Energy Storage System (LGG19E070006).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] R. Hoogendoorn, B. van Arerm and H. Serge, "Automated driving, traffic flow efficiency, and human factors: Literature review," *Transportation Research Record*, vol. 2422, no. 1, pp. 113–120, 2014.
- [2] K. Berntorp, "Path planning and integrated collision avoidance for autonomous vehicles," in *2017 American Control Conf.*, IEEE, Seattle, WA, USA, pp. 4023–4028, 2017.
- [3] D. González, J. Pérez, V. Milanés and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.
- [4] B. Yi, P. Bender, F. Bonarens and C. Stiller, "Model predictive trajectory planning for automated driving," *IEEE Transactions on Intelligent Vehicles*, vol. 14, no. 1, pp. 24–38, 2019.
- [5] S. D. Rajurkar, A. K. Kar, S. Goswami and N. K. Verma, "Optimal path estimation and tracking for an automated vehicle using GA optimized fuzzy controller," in *2016 11th Int. Conf. on Industrial and Information Systems*, IEEE, Roorkee, India, pp. 365–370, 2016.
- [6] A. Maamar and K. Benahmed, "A hybrid model for Anomalies detection in AMI system combining K-means clustering and deep neural network," *Computers, Materials & Continua*, vol. 60, no. 1, pp. 15–39, 2019.
- [7] D. Zeng, Y. Dai, F. Li, W. Jin and S. Arun Kumar, "Aspect based sentiment analysis by a linguistically regularized cnn with gated mechanism," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 5, pp. 3971–3980, 2019.
- [8] W. Wei, Y. Jiang, Y. Luo, J. Li and X. Wang *et al.*, "An advanced deep residual dense network (DRDN) approach for image super-resolution," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1592–1601, 2019.
- [9] J. Zhang, W. Wang, C. Lu and S. Arun Kumar, "Lightweight deep network for traffic sign classification," *Annals of Telecommunications*, vol. 75, no. 7–8, pp. 1–11, 2019.
- [10] G. Li and Y. Ma, "Deep path planning algorithm based on CNNs for perception images," in *2018 Chinese Automation Congress*, Xi'an, China: IEEE, pp. 2536–2541, 2018.
- [11] A. E. L. Sallab, M. Abdou, E. Perot and S. K. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 19, no. 19, pp. 70–76, 2017.
- [12] D. Isele, R. Rahimi and A. Cosgun, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," in *2018 IEEE Int. Conf. on Robotics and Automation*, Brisbane, QLD, Australia: IEEE, pp. 2034–2039, 2018.
- [13] X. L. Pan, Y. You, Z. Wang and C. Lu, "Virtual to real reinforcement learning for autonomous driving," *Arxiv preprint arxiv: 1704. 03952*, 2017.
- [14] W. Xia, H. Li and B. Li, "A control strategy of autonomous vehicles based on deep reinforcement learning," in *9th Int. Sym. on Computational Intelligence and Design*, IEEE, Hangzhou, China, pp. 198–201, 2016.
- [15] X. Xiong, J. Wang, F. Zhang and K. Li, "Combining deep reinforcement learning and safety based control for autonomous driving." *Arxiv preprint arxiv: 1612. 00147*, 2016.
- [16] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in: D. E. Rumelhart, J. L. McClelland (eds.), vol. 1, Cambridge: MIT Press, 1986, pp. 194–281.

- [17] R. Salakhutdinov, A. Mnih and G. E. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proc. of the Twenty-Fourth Int. Conf. on Machine Learning*, New York, NY, USA: ACM, pp. 791–798, 2007.
- [18] G. E. Hinton, "A practical guide to training restricted Boltzmann machines, " in *Neural Networks: Tricks of the Trade*, vol. 7700. Berlin, Heidelberg: Springer, 2012, pp. 599–619.
- [19] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [20] UCI: winequality-red. 2017. [Online]. Available: <http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality>.