Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9896258, 1 page
https://doi.org/10.1155/2023/9896258

Retraction

Retracted: An Automatic Driving Control Method Based on Deep Deterministic Policy Gradient

Wireless Communications and Mobile Computing

Received 12 December 2023; Accepted 12 December 2023; Published 13 December 2023

This article has been retracted by Hindawi, as publisher, following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of systematic manipulation of the publication and peer-review process. We cannot, therefore, vouch for the reliability or integrity of this article.

Please note that this notice is intended solely to alert readers that the peer-review process of this article has been compromised.

Wiley and Hindawi regret that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] H. Zhang, J. Xu, and J. Qiu, "An Automatic Driving Control Method Based on Deep Deterministic Policy Gradient," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 7739440, 9 pages, 2022.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7739440, 9 pages
https://doi.org/10.1155/2022/7739440

Research Article

An Automatic Driving Control Method Based on Deep Deterministic Policy Gradient

Haifei Zhang[1], **Jian Xu**[2] **and Jianlin Qiu**[1,2]

[1]*School of Computer and Information Engineering, Nantong Institute of Technology, Yongxing Road 211, Nantong 226002, China*
[2]*School of Information Science and Technology, Nantong University, Seyuan Road 9, Nantong 226019, China*

Correspondence should be addressed to Haifei Zhang; 46462490@qq.com

Received 15 November 2021; Accepted 24 December 2021; Published 24 January 2022

Academic Editor: Ali Kashif Bashir

The traditional automatic driving behavior decision algorithm needs to manually set complex rules, resulting in long vehicle decision-making time, poor decision-making effect, and no adaptability to the new environment. As one of the main methods in the field of machine learning and intelligent control in recent years, reinforcement learning can learn reasonable and effective policies only by interacting with the environment. Firstly, this paper introduces the current research status of automatic driving technology and the current mainstream automatic driving control methods. Then, it analyzes the characteristics of convolutional neural network, reinforcement learning method (Q-learning), and deep Q network (DQN) and deep deterministic policy gradient (DDPG). Compared with the DQN algorithm based on value function, the DDPG algorithm based on action policy can well solve the continuity problem of action space. Finally, the DDPG algorithm is used to solve the control problem of automatic driving. By designing a reasonable reward function, deep convolution network, and exploration policy, the intelligent vehicle can avoid obstacles and, finally, achieve the purpose of avoiding obstacles and running the whole process in a 2D environment.

1. Introduction

The traditional automatic driving technology involves the composition of perception, planning, decision-making, control, and other modules. Through the perception module, the relevant information of the road and environment is obtained, the overall driving route is planned, and then, the planning and perceived information are continued to be used for future driving goals. Such a design may be associated with many task modules. For some complex task systems, the number of modules will be particularly large, and the maintenance cost will be relatively high. At present, some supervised learning methods can achieve their goals through learning and training, but such learning methods need a large amount of learning data as the basis of network training. Through the historical data, some feature points are obtained to act on the experience pool of target decision-making. Such methods need a large amount of labeled data, which cannot achieve the goal of self-help learning and online decision-making. This paper uses the technology of the deep reinforcement learning-deep deterministic policy gradient method to train the policy network, so that the network can control the intelligent vehicle to avoid obstacles and, finally, achieve the purpose of avoiding obstacles and running the whole process in the 2D environment.

2. Related Work

The automatic driving technology in China started later than in other countries. In the real sense, the automatic driving vehicle equipped with some sensors is the beginning of the study of automatic driving in China. After entering the 21st century, the emergence of autonomous driving of unmanned vehicles has created the highest driving speed record in China, reaching 76 kilometers per hour. After that, some other scientific research institutions developed an automatic driving vehicle platform, which has a certain impact in China. The domestic Baidu company is a leader in the IT field and has invested a lot of money and R&D strength in automatic driving. In terms of automatic driving,

fully automatic driving under mixed road conditions has been perfectly realized, and relevant research and development tests are further promoted [1, 2]. By the end of November 2016, the number of patent applications for Baidu automatic driving technology had reached 605. Relying mainly on the accumulation of artificial intelligence and deep learning, Baidu is engaged in the development of ten technologies related to driverless vehicles, including ten technologies of environmental perception, behavior prediction, planning and control, operating system, intelligent interconnection, on-board hardware, human-computer interaction, high-precision positioning, high-precision map, and system security [3]. The methods based on reinforcement learning and deep reinforcement learning have achieved good results in automatic driving and have great application significance in training efficiency and driving strategies. However, further research and development are still needed in complex problems and considering pedestrians [4–7]. In terms of deep learning automatic driving, scholars from Tsinghua University have improved the robustness and accuracy of algorithm recognition through the research on CNN-related algorithms and achieved good results in multitarget recognition, but there are too many redundant results and low efficiency [8]. The automatic driving technology using machine learning mainly studies how computers acquire knowledge or optimize their own skills through experience or exploring the environment to improve learning and computing efficiency. This is a technical field used to solve automatic driving in the current development [9]. Transfer trajectory planning, reinforcement learning, deep reinforcement learning, and machine learning are widely used to solve the problem of automatic driving [10–12]. At present, although some research results have been achieved in automatic driving technology, there are still some problems in many aspects. Therefore, it is very meaningful to use the deep deterministic policy gradient method to study automatic driving technology.

Foreign automatic driving technology, from the beginning of the unmanned carrier, automatic driving handling equipment first appeared in the United States. It was used to transport goods in the grocery warehouse with arranged wires [3]. After that, someone successfully developed an autonomous robot, which can drive automatically on low speed and flat roads. Some automatic driving competitions abroad have also promoted the development of automatic driving technology. The foreign Google company is a leader in the technology of automatic driving in the industry. Since the preparation, it participated in the driving test on time urban roads with the developed automatic driving vehicle in 2010 [13–15]. Bojarski et al. [16] proposed an end-to-end learning automatic driving mode, which can learn the steering wheel control policy from the data captured by the vehicle camera through the convolutional neural network. However, this method needs to input the data of human driving into the training network, and the cost of data acquisition and annotation is large. Chae et al. [17] proposed a brake control system based on deep reinforcement learning. Based on the DQN algorithm, the system judges whether braking is required and the braking force through the data captured by the sensor, so as to avoid hitting obstacles and pedestrians. However, this method only applies deep reinforcement learning to the brake control system, which has great limitations. Sallab et al. proposed an end-to-end reinforcement learning policy [18] for lane auxiliary maintenance, comparing the DQN algorithm of discrete policy with the DDAC algorithm of discrete policy, and achieved good results. Sallab et al. proposed a deep reinforcement learning framework for automatic driving [19], which divides automatic driving into three stages: identification, prediction, and planning. The framework uses a deep neural network for identification and a cyclic neural network for prediction and uses the method of deep reinforcement learning to train the planning network segment. Chen et al. [20] proposed a new autonomous driving mode based on direct perception, which uses the deep ConvNet architecture to estimate the enlightenment of driving behavior, rather than analyzing the whole scene (intermediary perception method) or blindly mapping the image directly to driving commands (behavior reflection method). In May 2016, Google announced its cooperation with Fiat Chrysler Automobiles (FCA). FCA produced 100 Pacifica hybrid vans for Google, equipped with a complete set of sensors, telematics, and computing units. In October, the test vehicle equipped with the new automatic driving system was tested in many places with extreme weather in the United States. At the same time, automobile enterprises including Japan and Germany have also joined the research of automatic driving and are jointly committed to the research of automatic driving technology.

3. Reinforcement Learning

3.1. Principles of Reinforcement Learning. Reinforcement learning is a process in which agents learn how to take a series of actions in the environment, so as to maximize the cumulative reward. The basic framework of reinforcement learning algorithm is shown in Figure 1. The agent in the algorithm represents the subject of problem solving. For example, the agent in this study is an autonomous vehicle. The agent tries to make an action in the environment, which will lead to the environment being updated, and the agent will transition to a new state. In such a process, the agent can get the reward corresponding to the previous action at the same time. Repeating this process will produce a large number of training sample sets. Using these data to continuously optimize the behavior of the agent, after a long time of training, we will get an optimal policy to complete the task.

The theoretical basis of reinforcement learning is the Markov decision process (MDP). The most basic form of MDP is the Markov chain, which must conform to the Markov property, that is, the conditional probability distribution P of the future state S of the system only depends on the current state and has nothing to do with the past state, which will make the observed state conditionally independent.

The Markov decision process includes the following steps: give the agent the initial state s_1, the agent is in the

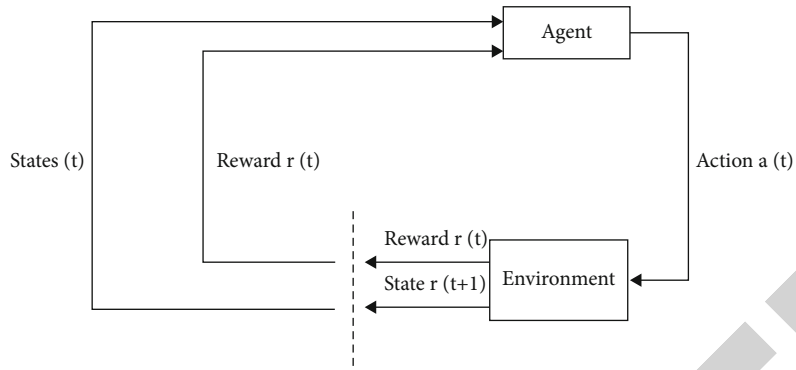

Figure 1: Basic framework of reinforcement learning.

s_1 state, select the action a_1 from the action space A, reach the next state s_2, get the reward r_1, continue to select the action a_2, get the reward r_2, and enter the state s_3, and so on until the agent reaches the maximum number of iterative steps T. The process from any time t to the ending state is called an episode, and the reward obtained in the episode is expressed as the following equation:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots\cdots + \gamma^T r_{t+T+1} = \sum_{k=0}^{T} \gamma^k r_{t+k+1}. \quad (1)$$

Among them, γ is a number between $[0, 1)$; γ^k means that the larger the time step t is, the less influence the reward will have on the current action. Since the reward obtained fluctuates greatly, the expectation of reward is introduced as the state-value in the following equation:

$$V(s) = E[G \mid S_t = s]. \quad (2)$$

The learning process of reinforcement learning is the process of optimizing the policy by maximizing the state value function. The policy is the control rule of the agent, which can be expressed as the probability distribution function of the actions that can be taken in a certain state. That is,

$$\pi(a \mid s) = P[A_t = a \mid S_t = s]. \quad (3)$$

If you want to maximize the reward in the whole stage, you can achieve it by selecting the maximum reward action in each state in the agent and introducing the Bellman equation to define the state value in the following equation:

$$\begin{aligned} V(s) &= \max_{a \in A} E_{s' \in S}[r_{s,a} + \gamma V_{s'}] \\ &= \max_{a \in A} \sum_{s' \in S} p_{a,s \longrightarrow s'} \left(r_{s,a} + \gamma V\left(s'\right) \right), \end{aligned} \quad (4)$$

where $V(s)$ and $V(s')$ represent the value of the current state and the target state, respectively; $p_{a,s \longrightarrow s'}$ represents the probability that the agent reaches the target state s' after an action a is selected in the state s. According to this formula, the value expression of the action can be extracted as the following equation:

$$Q(s, a) = E_{s' \in S}[r_{s,a} + \gamma V_{s'}] = \sum_{s' \in S} p_{a,s \longrightarrow s'} \left(r_{s,a} + \gamma V\left(s'\right) \right). \quad (5)$$

Thus, the theoretical basis of Q-learning is obtained as the following equation:

$$Q(s, a) = r_{s,a} + \gamma \max_{a' \in A} Q\left(s', a'\right). \quad (6)$$

3.2. Q-Learning Algorithm. The Q-learning algorithm is an algorithm based on value function, which belongs to the model-free learning method. The learning algorithm establishes a "state action" Q table, learns the value of a specific state and a specific action, records the action value function obtained by the action taken in the current state, and updates the Q table through the reward brought by each action. The update method is expressed in the following equation:

$$Q(s, a) = (1 - \lambda_t) Q(s, a) + \lambda_t \left[r + \gamma \max_{a \in A} Q\left(s', a'\right) \right], \quad (7)$$

where A is the collection of a series of actions, a is the action taken in the current state, s' is the next state, and a' is the next predicted action. γ is the discount factor, and λ is the learning rate. The larger the λ value, the faster the learning convergence. If it is too large, it is easier to overconverge rather than to arrive at the optimal solution. The flow chart of the Q -learning algorithm is shown in Figure 2.

4. Deep Reinforcement Learning

Deep reinforcement learning is the combination of deep learning and reinforcement learning. Classical algorithms include the DQN algorithm and DDPG algorithm. The DDPG algorithm is further developed on the basis of the DQN algorithm. It is a model-free and off-policy algorithm.

4.1. DQN Algorithm. The traditional DQN [21] is a method combining Q-learning and the deep neural network. Deep Q network has advantages in dealing with continuous state space, but it cannot solve the problem of continuous action

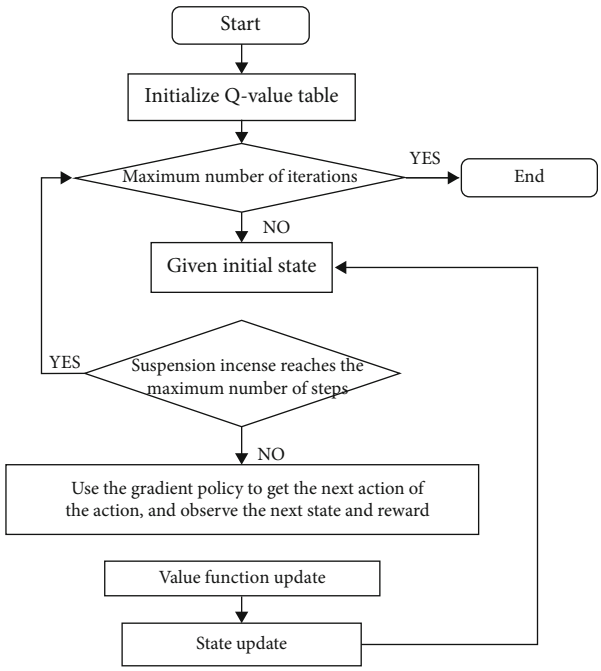

Figure 2: Flow chart of Q-learning algorithm.

space. Figure 3 shows the structure of the DQN algorithm, and Figure 4 shows the flow chart of the DQN algorithm.

4.2. DDPG Algorithm. The DDPG algorithm [22] is an off-policy and model-free deep reinforcement learning algorithm, combining deep learning and reinforcement learning, integrating the advantages of the DQN algorithm and Actor-Critic (AC) algorithm. The DDPG algorithm is the same as the AC algorithm framework, but its neural network division is finer. The DQN algorithm has good performance in discrete problems. The DDPG algorithm uses the experience of DQN for reference to solve the problem of continuous control and realize end-to-end learning. The algorithm flow of DDPG is shown in Figure 5, in which the actor network accepts the input state, makes action selection, and outputs action variables; the critic network evaluates the quality of the selected action and calculates the reward value. The detailed steps of the DDPG algorithm are as follows:

(1) Initialize the parameters of the neural network. The actor selects an action according to the behavior policy, adds noise N_t to the action output by the policy network to increase exploration, and transmits it to the environment to execute the action a_t:

$$a_t = \mu(s_t \mid \theta^{\mu}) + N_t. \tag{8}$$

(2) After the environment is executed a_t, return to reward r_t and new state s_{t+1}

(3) Actor stores the state transition (s_t, a_t, r_t, s_{t+1}) into the replay memory as the training set of the online network

(4) DDPG creates two copies of neural networks for the policy network and the Q network, respectively, the online network and the target network. The update method of the policy network is as follows:

$$\begin{cases} \text{online} : Q(s, a \mid \theta^{\mu}), & \text{gradient update } \theta^{\mu}, \\ \text{target} : Q\left(s, a \mid \theta^{\mu'}\right), & \text{soft update } \theta^{\mu'}. \end{cases} \tag{9}$$

The Q network update method is as follows:

$$\begin{cases} \text{online} : Q\left(s, a \mid \theta^{Q}\right), & \text{gradient update } \theta^{Q}, \\ \text{target} : Q\left(s, a \mid \theta^{Q'}\right), & \text{soft update } \theta^{Q'}. \end{cases} \tag{10}$$

N transition data are randomly sampled from replay memory as minibatch training data of the online policy network and online Q network. Single transition data in minibatch is represented by (s_i, a_i, r_i, s_{i+1}).

(5) In critical, calculate the Q gradient of the online Q network:

The loss of the Q network is defined as

$$\begin{aligned} L &= \frac{1}{N}\sum_i \left(y_i - Q\left(s_i, a_i \mid \theta^{Q}\right)\right)^2, \\ y_i &= r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1} \mid \theta^{\mu'}\right) \mid \theta^{Q'}\right). \end{aligned} \tag{11}$$

The gradient for L and θ^{Q} can be obtained: $\nabla_{\theta^{Q}} L$, where the calculation uses the target policy network μ' and target Q network Q'.

(6) Update online Q: update θ^{Q} with the Adam optimizer

(7) In the actor, calculate the policy gradient of the policy network:

$$\nabla_{\theta^{\mu}} J_{\beta}(\mu) \approx \frac{1}{N} \bullet \left(\nabla_{\alpha} Q\left(s, a \mid \theta^{Q}\right)\Big|_{s=s_i, a=w(s_i)} \bullet \nabla_{\theta^{\mu}} \mu(s \mid \theta^{\mu})\Big|_{s=s_i}\right). \tag{12}$$

(8) Update online policy network: update θ^{μ} with the Adam optimizer

(9) The parameters of the target network adopt the method of soft update:

$$\begin{cases} \theta^{Q'} \longleftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}, \\ \theta^{\mu'} \longleftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}. \end{cases} \tag{13}$$

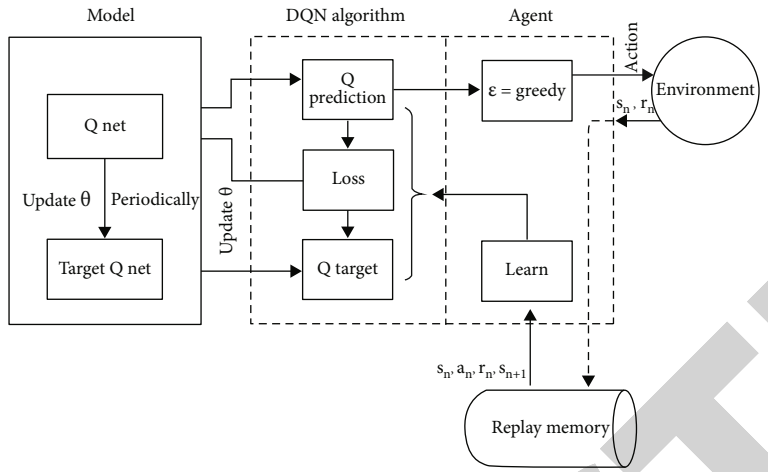

Figure 3: DQN algorithm flow structure.

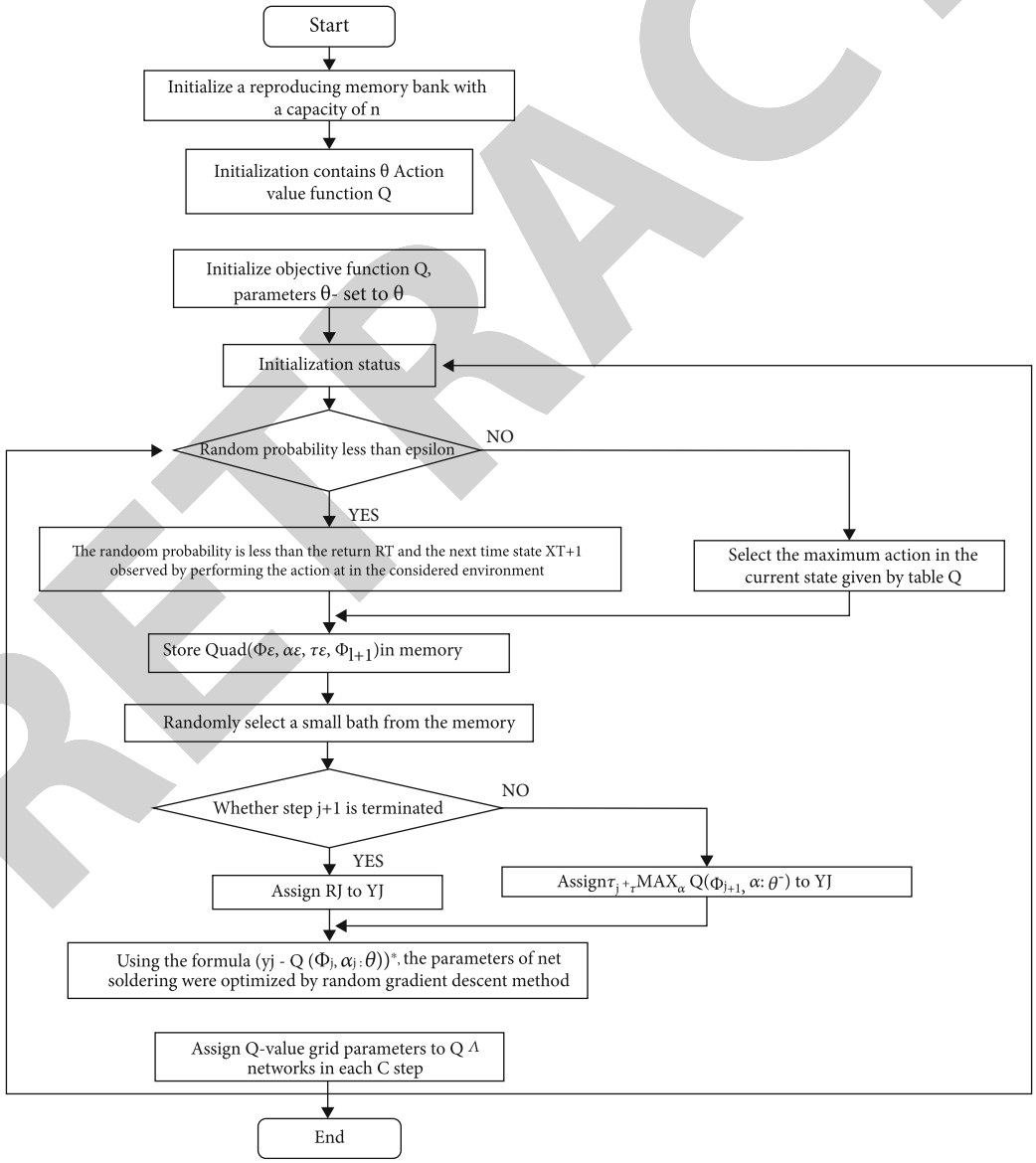

Figure 4: The flow chart of DQN algorithm.

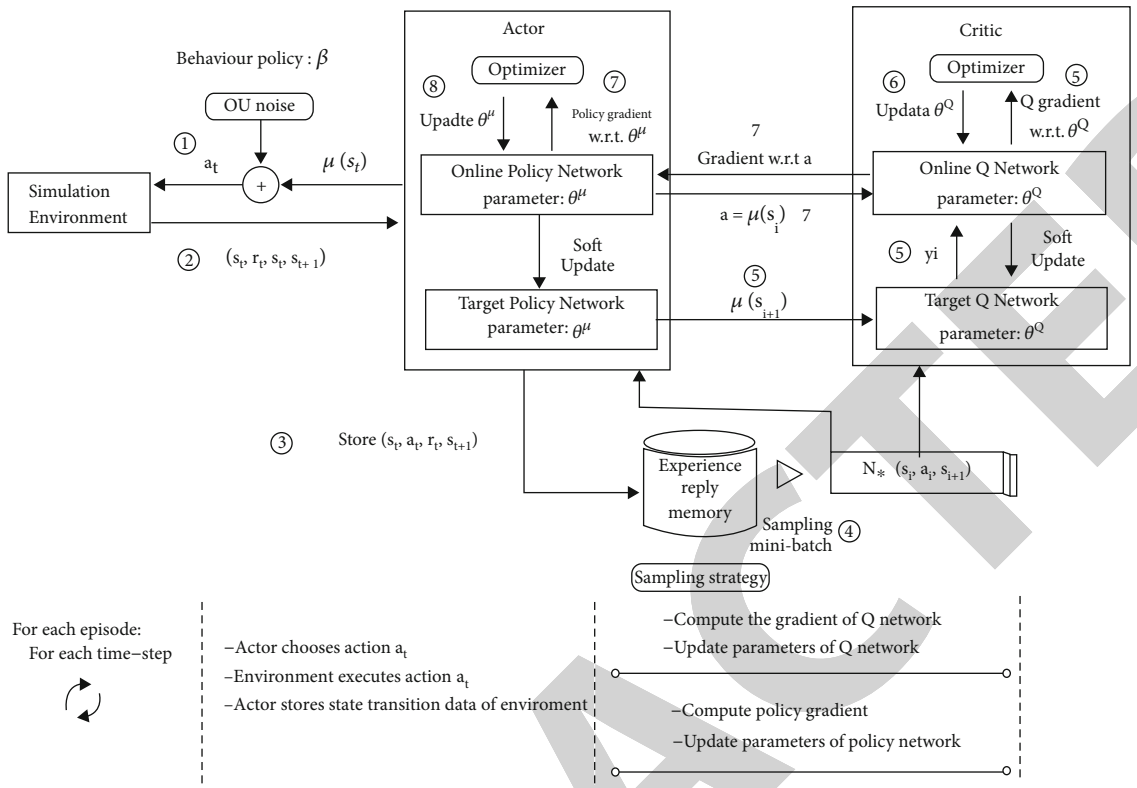

Figure 5: The algorithm flow chart of DDPG.

In general, the DDPG algorithm uses the Actor-Critic framework to iterate the training of the policy network and Q network through the interaction among the environment, actor, and critic.

5. Automatic Driving Control Method Based on DDPG

5.1. System Structure. The model of automatic driving is mainly divided into two parts, including the DDPG algorithm and the experimental simulation. By using the DDPG algorithm to train the neural network of the automatic driving model, the network can control the motor vehicle to avoid obstacles and drive normally on the road.

The experimental simulation part mainly includes the motor vehicle and the environment of the motor vehicle. After receiving the control sensor, the information of the environment is continuously transmitted to the DDPG algorithm, so that the algorithm can obtain the state variables and reward values. Through the continuous training of the network and the continuous updating of network parameters, the target reward value is also continuously improved. The structure diagram of the automatic driving control system model is shown in Figure 6. It mainly reflects that in the motor vehicle motion model, according to the control command action execution of the environment, the new state value is obtained and then the obtained reward value; the relevant parameters of this motion are trained in the neural network; and the trained network is continuously updated

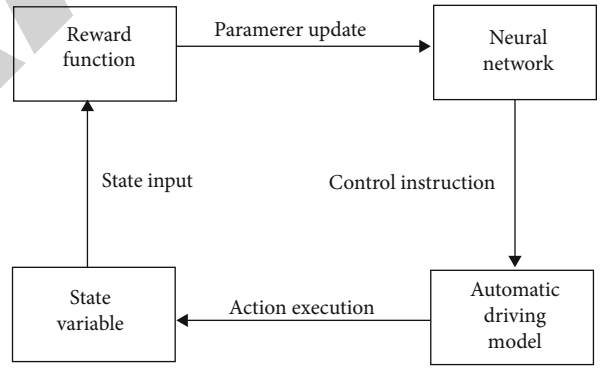

Figure 6: Structure of automatic driving control system.

until the end. Such a process ends with the maximum number of iterations.

5.2. Automatic Driving Control Model. The automatic driving model involved in this design uses the two-dimensional $500 * 500$ pixel space to control the automatic driving motor vehicle, in which the range of obstacles is the pixel space of the middle area $260 * 260$, and the parts (120, 120), (380, 120), (380, 380), and (120, 380) surrounded by the following four points and the areas beyond $500 * 500$ are the range of obstacles. The motor vehicle adopts the mode of fixed speed, with 5 sensors; the farthest detectable distance is 150; and the position coordinate of the motor vehicle starting training is (450, 300). The sensors are located in the middle and front of both sides of the vehicle, 45 degrees in front of the left and

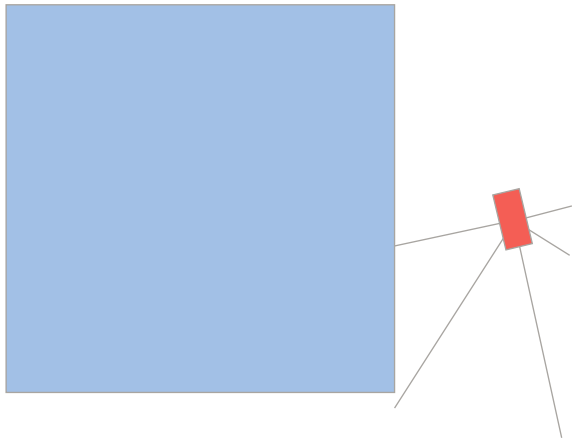

Figure 7: Automatic driving model.

45 degrees in front of the right. A total of five sensors are used to detect the operation of the current vehicle. The automatic driving control model is shown in Figure 7. The unmanned vehicle is marked as a $20 * 40$ pixel coordinate area. The sensor data mainly includes the distance from the obstacle and its coordinates. During the training process, the straight and turning directions of the agent are controlled by the network.

5.3. Automatic Driving Control Method

5.3.1. Reward. The design of the reward function of the unmanned vehicle under this model is relatively simple. It mainly detects whether the unmanned vehicle encounters obstacles during each movement. If the target is -1, otherwise it is 0.

$$r = \begin{cases} 0, & \text{No\_collision}, \\ -1, & \text{Have\_collision}. \end{cases} \tag{14}$$

5.3.2. State. According to the current model design, the relevant parameters of the unmanned vehicle are selected as the training parameters of the DDPG algorithm, which mainly includes the five distance parameters detected by the sensor. When the minimum distance between the motion sensor and the obstacle is less than half of the unmanned vehicle, it means that the unmanned vehicle has a collision, and the reward is -1.

6. Simulation Experiment and Result Analysis

6.1. Experimental Setup. The parameters for solving the automatic driving problem using the DDPG algorithm are as follows: the maximum number of iterations is 500, the maximum number of steps per iteration is 600; the reward discount factor is 0.9; the learning rate of the actor and critical is 0.0001; batch-size, that is, the number of samples obtained in one training, is 16; and the number of neurons is 120. Simulation training for this problem was done in Python language and observation on the model training was done according to the model diagram, which can better reflect the current training degree of the model.

6.2. Result Analysis. The training of the algorithm at the beginning of model training is randomly intercepted. Through the visual diagram and the training process, it can be seen that the learning ability of the model at the beginning of training is not strong, the model can only be explored at will, and it is easy to collide at the beginning. Occasionally, automatic driving can be carried out briefly in a single direction. According to the diagram, we can only learn at the initial stage, mainly to explore some movement directions, and the uncertainty in the movement direction is not high.

In the middle of model training, a model diagram in the training process is randomly intercepted. As can be seen from the figure, the automatic driving model of the unmanned vehicle has been able to avoid obstacles for turning or straight operation and can better avoid obstacles in the process of turning. The process of quickly avoiding obstacles and driving forward has been basically realized. The automatic driving control model can still gradually find a better motion planning direction through random straight or turning. In the process of this training, when encountering some places that have not been explored, there is still the possibility of collision, but with the continuation of the iteration, the driving can be basically completed.

In the later stage of model training, a model diagram in the training process is randomly intercepted. As can be seen from the figure, the unmanned vehicle has been able to drive better in the control model, avoid obstacles perfectly, and hardly encounter obstacles. In the later stage of training, the network has basically been trained and formed, which can quickly judge whether the next action is straight or turning, and basically reaches the maximum steps of each iteration. From this aspect, it also reflects that the learning ability of the model is very good in the later stage and can well control the movement of the unmanned vehicle. The agent driverless model already has a relatively formed network model. The network parameters are optimized and the loss value is low, so this good effect can be achieved. Through this training, it is more fully explained that the DDPG algorithm is feasible and effective in solving the unmanned control problem.

In this model simulation experiment, the change of reward value in each iteration is shown in Figure 8. It can be seen from the figure that after about 300 iterations, the reward value obtained can basically be stable at 0, indicating that collision-free motion has been basically realized at this time. The 500 iterations set in this simulation basically converge to about 300 times, and the relative number of times to reach convergence is still relatively small. This shows that the DDPG method based on the deep deterministic policy gradient has fast convergence speed and obvious effect in solving the unmanned vehicle automatic driving problem, which shows that the depth reinforcement learning algorithm has better advantages in this problem. Through this experiment, we can know that the method based on deep reinforcement learning makes the model have better self-

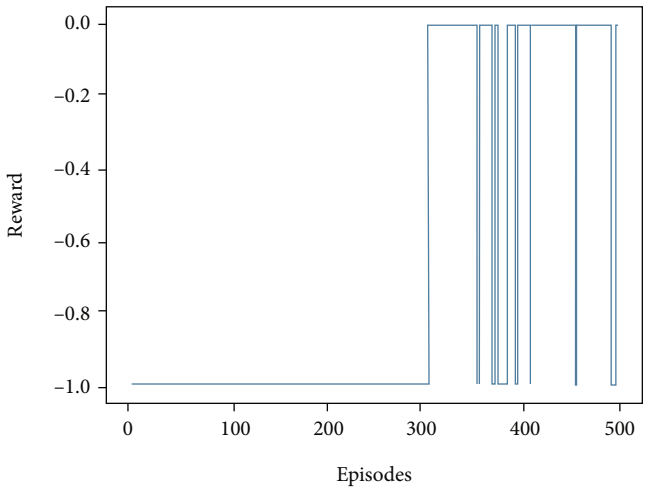

Figure 8: The reward value of each iteration of automatic driving model.

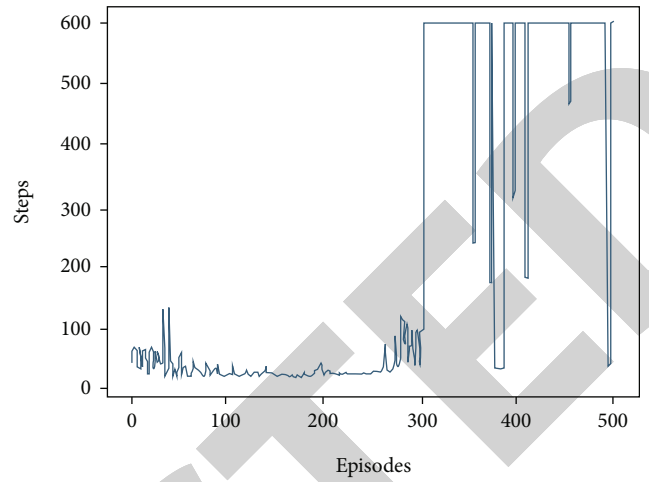

Figure 9: The number of steps of each episode of automatic driving model.

learning ability. Compared with the general supervised learning method, the trained model agent has obvious advantages. The agents in the model generate a large number of training data to train the network according to the environmental changes, which is flexible and convenient for network training. The network can be trained and analyzed without too much label data.

It can be seen from Figure 9 that at the initial stage of iteration, the unmanned vehicle automatic driving model performs fewer steps, and the model easily collides with obstacles. After 300 iterations, the model can basically achieve the maximum number of steps each time. It can also be seen that due to the continuous training and the continuous optimization of the network, the network can better judge the selection decision of execution action. At the later stage of training, the collision-free driving in the model can be basically realized. This better shows that the agent algorithm model has strong applicability, strong robustness, and high stability. At the same time, the test mode is used to test the network. The trained network can well control the automatic driving of the unmanned vehicle and avoid obstacles.

7. Conclusions

This paper introduces the current research status of automatic driving technology and analyzes the current mainstream automatic driving control methods. Then, it analyzes the characteristics of the convolutional neural network, reinforcement learning method (Q-learning), and deep Q network (DQN) and deep deterministic policy gradient (DDPG). Compared with the DQN algorithm based on value function, the DDPG algorithm based on action policy can well solve the continuity problem of action space. Finally, the DDPG algorithm is used to solve the control problem of automatic driving. Data are collected through training, and the neural network training is carried out on the automatic driving model, so that the network can control the intelligent vehicle to avoid obstacles and finally achieve the goal that the intelligent vehicle can avoid obstacles and run the whole process in 2D environment. In terms of automatic driving model design, this design simply uses the unmanned vehicle driving in two-dimensional space for control, and the driving action considered is only limited to straight ahead and turning. It contains less automatic driving control information and does not carry out training and testing in real three-dimensional space or mature automatic driving model. After that, the automatic vehicle control of three-dimensional and multidimensional model can be considered for training simulation test.

Data Availability

The experiments involved in this paper do not require any raw/processed data. The model is automatically trained by the DDPG algorithm.

Conflicts of Interest

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

Acknowledgments

This work is sponsored by the following: (1) the Training Project of Top Scientific Research Talents of Nantong Institute of Technology under Grant No. XBJRC2021005; (2) the Science and Technology Planning Project of Nantong City under Grant Nos. JC2021132, JCZ20172, JCZ20151, JCZ20148, and MS22021028; (3) the Scientific Research Backbone Training Project of Nantong Institute of Technology under Grant No. ZQNGG109; (4) the Key Projects of Innovation and Entrepreneurship Training Program for College Students in Jiangsu Province in 2021 under Grant No. 202112056003Z; and (5) the Innovation and Entrepreneurship Training Program of Nantong Institute of Technology in 2021 under Grant No. XDC2021036.

References

[1] P. Ke, Z. Yanxin, and Y. Chenkun, *A decision-making method for self-driving based on deep reinforcement learning [M.S. thesis]*, Beijing Jiao Tong University, Beijing, China, 2020.

[2] L. Lingyun, *Research on end-to-end automatic driving technology based on deep reinforcement learning [M.S. thesis]*, University of Chinese Academy of Sciences, Beijing, China, 2020.

[3] H. Jia, R. Hui, W. Wen-yang, T. Xiaodi, G. Song, and G. Peng, "Development summary of Baidu and Google driverless car," *Auto Electric Parts*, vol. 12, no. 12, pp. 19–21, 2017.

[4] Z. Bin, H. Ming, W. Chen Xiliang, L. B. Chunxiao, and Z. Bo, "Self-driving via improved DDPG algorithm," *Computer Engineering and Applications*, vol. 55, no. 10, pp. 264–270, 2019.

[5] W. Bingchen, *Research on autonomous driving decision control based on deep reinforcement learning [M.S. thesis]*, Dalian University of Technology, Dalian, China, 2020.

[6] S. Nan, *Research on driverless control policy based on deep reinforcement learning [M.S. thesis]*, Harbin Institute of Technology, Harbin, China, 2020.

[7] P. Feng and B. Hong, "Research progress of automatic driving control technology based on reinforcement learning," *Journal of Image and Graphics*, vol. 26, no. 1, pp. 28–35, 2021.

[8] Z. Xinyu, G. Hongbo, Z. Jianhui, and Z. Mo, "Overview of deep learning intelligent driving methods," *Journal of Tsinghua University (Science and Technology)*, vol. 58, no. 4, pp. 438–444, 2018.

[9] M. Yuchen, "Research on key technologies of automatic driving," *Practical Electronic*, vol. 14, pp. 74–76+63, 2019.

[10] Y. Lingli, S. Xuanya, L. Ziwei, W. Yadong, and Z. Kaijun, "Intelligent land vehicle model transfer trajectory planning method of deep reinforcement learning," *Control Theory & Applications*, vol. 36, no. 9, pp. 1409–1422, 2019.

[11] L. Si, "Research on autonomous driving based on deep reinforcement learning," *Automation Application*, vol. 5, pp. 57–59, 2020.

[12] L. Zhihang, "Autonomous driving strategy based on deep recursive reinforcement learning," *Industrial Control Computer*, vol. 33, no. 4, pp. 61–63, 2020.

[13] Z. Miao, Z. Qi, L. Wentao, and Z. Boyuan, "A policy-based reinforcement learning algorithm for intelligent train control," *Journal of the China Railway Society*, vol. 42, no. 1, pp. 69–75, 2020.

[14] J. Markoff, "Google cars drive themselves in traffic," *New York Times*, vol. 10, pp. 1–5, 2010.

[15] H. Zhiqiu and C. Zhang, "Evolution summarization of automated guided vehicles (AGV)," *Machine Design and Manufacturing Engineering*, vol. 39, no. 1, pp. 53–59, 2010.

[16] M. Bojarski, D. Del Testa, D. Dworakowski et al., "End to end learning for self-driving cars," 2016, https://arxiv.org/abs/1604.07316.

[17] H. Chae, C. M. Kang, B. Kim, J. Kim, C. C. Chung, and J. W. Choi, "Autonomous braking system via deep reinforcement learning," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, Yokohama, Japan, 2017.

[18] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "End-to-end deep reinforcement learning for lane keeping assist," 2016, https://arxiv.org/abs/1612.04340.

[19] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 19, pp. 70–76, 2017.

[20] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deep driving: learning affordance for direct perception in autonomous driving," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2722–2730, 2015, https://ieeexplore.ieee.org/document/7410669.

[21] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Proximal policy optimization algorithms," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[22] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al., "Continuous control with deep reinforcement learning," 2020, https://arxiv.org/abs/1509.02971.