Journal of Computer and Communications, 2021, 9, 138-157
https://www.scirp.org/journal/jcc
ISSN Online: 2327-5227
ISSN Print: 2327-5219

Double BP Q-Learning Algorithm for Local Path Planning of Mobile Robot

Guoming Liu, Caihong Li*, Tengteng Gao, Yongdi Li, Xiaopei He

School of Computer Science and Technology, Shandong University of Technology, Zibo, China
Email: *lich@sdut.edu.cn

How to cite this paper: Liu, G.M., Li, C.H., Gao, T.T., Li, Y.D. and He, X.P. (2021) Double BP Q-Learning Algorithm for Local Path Planning of Mobile Robot. *Journal of Computer and Communications*, **9**, 138-157.
https://doi.org/10.4236/jcc.2021.96008

Received: May 19, 2021
Accepted: June 20, 2021
Published: June 23, 2021

Abstract

Aiming at the dimension disaster problem, poor model generalization ability and deadlock problem in special obstacles environment caused by the increase of state information in the local path planning process of mobile robot, this paper proposed a Double BP Q-learning algorithm based on the fusion of Double Q-learning algorithm and BP neural network. In order to solve the dimensional disaster problem, two BP neural network fitting value functions with the same network structure were used to replace the two *Q* value tables in Double Q-Learning algorithm to solve the problem that the *Q* value table cannot store excessive state information. By adding the mechanism of priority experience replay and using the parameter transfer to initialize the model parameters in different environments, it could accelerate the convergence rate of the algorithm, improve the learning efficiency and the generalization ability of the model. By designing specific action selection strategy in special environment, the deadlock state could be avoided and the mobile robot could reach the target point. Finally, the designed Double BP Q-learning algorithm was simulated and verified, and the probability of mobile robot reaching the target point in the parameter update process was compared with the Double Q-learning algorithm under the same condition of the planned path length. The results showed that the model trained by the improved Double BP Q-learning algorithm had a higher success rate in finding the optimal or sub-optimal path in the dense discrete environment, besides, it had stronger model generalization ability, fewer redundant sections, and could reach the target point without entering the deadlock zone in the special obstacles environment.

Keywords

Mobile Robot, Local Path Planning, Double BP Q-Learning, BP Neural Network, Transfer Learning

1. Introduction

Path planning is the core part of mobile robot control system and the key of autonomous navigation technology. Where local path planning is one of the important direction of path planning research. In an unknown environment, mobile robot detects and collects map information through the real-time state information acquired by sensors, updates the environment model in real time and plans a collision free path from the start point to the target point [1].

The quality of the path planning result determines whether the mobile robot can efficiently and accurately complete the task [2]. When the pose information of the mobile robot itself is known, it can take the short running path and high smoothness as the optimization objectives [3]. At present, the commonly used local path planning algorithms include artificial potential field method [4], fuzzy control [5], neural network [6] and reinforcement learning [7]. The artificial potential field method has good real-time performance in path planning, but the mobile robot can be affected by the gravitational potential field function and the repulsive potential field function. It is easy to cause local minimum points, shock or stagnation [8]. Multiple restrictions can be added to the function model to eliminate local minimum points [9]. Fuzzy control method can summarize complete control rules and achieve local path planning through fuzzy reasoning and fuzzy judgment. However, simple fuzzification of information will reduce the precision of control [10], but it can be combined with neural network to form a fuzzy neural network to output a higher precision model [11]. Reinforcement learning algorithm utilizes mobile robot to interact with the environment and accumulate rewards to optimize strategies [12]. Literature [13] proposed to use deep reinforcement learning for path planning in an unknown environment, but the input state didn't contain enough feature information and the model generalization was poor. Literature [14] abandoned the Q value table and proposed the method of using neural network to fit the value function. Although it contains enough characteristic information, there is no clear division for the surrounding obstacles, which is easy to fall into the local minimum state and the path planning efficiency is low. Literature [15] used sensors to explore the obstacle information and the returned obstacle information and actions are input into the neural network. Although the local minimum value would not be generated, but the information of the target point would not be considered, so the problem of path redundancy would easily occur. The application of reinforcement learning in local path planning requires multi-step decision and most tasks are correlated. Therefore, the learning of strategies under different scenarios can be accelerated through parameter transfer learning [16]. Transfer learning is a machine learning method that serves as the start point of the second task model for the task developed model [17]. In reinforcement learning, it mainly uses the fixed domain transferring from a single source task to a target task [18]. Literature [19] proposed case-based reasoning (CBR) as a heuristic method to accelerate the process of Q learning algorithm through transfer learning. Literature [20] applied the experience learned in two-dimensional environment to accelerate the learning of strategy in three-dimensional environment. Both References [19] [20] used transfer learning to improve the learning efficiency of reinforcement learning algorithm, but didn't consider strategy learning in two-dimensional complex scenes, and the model generalization ability was poor. Literature [21] used transfer learning to map the corresponding actions output by neural network to another domain, which accelerated the learning process of related but different tasks, but the speed of learning different strategies was slow.

In view of the dimension disaster that the Double Q-learning algorithm cannot deal with and the poor generalization ability problem of the trained model, this paper proposes a Double BP Q-learning algorithm. The fitting value function of two BP neural networks with the same structure is used to replace the Q value table, the mechanism of the prioritized experience replay and parameter transfer are added to accelerate the learning efficiency of the model and improve its generalization ability. The trained model is used to deal with the obstacle avoidance in different obstacles environments.

2. Design of State Space and Action Space

The appropriate state space and action space are designed as the input and output of the neural network. The information of the surrounding environment are returned by the sensors.

Taking the mobile robot as the origin of coordinates and the current running direction of it as the Y-axis direction, the global coordinate system of the mobile robot is established, as shown in Figure 1. Omnidirectional radar sensors are

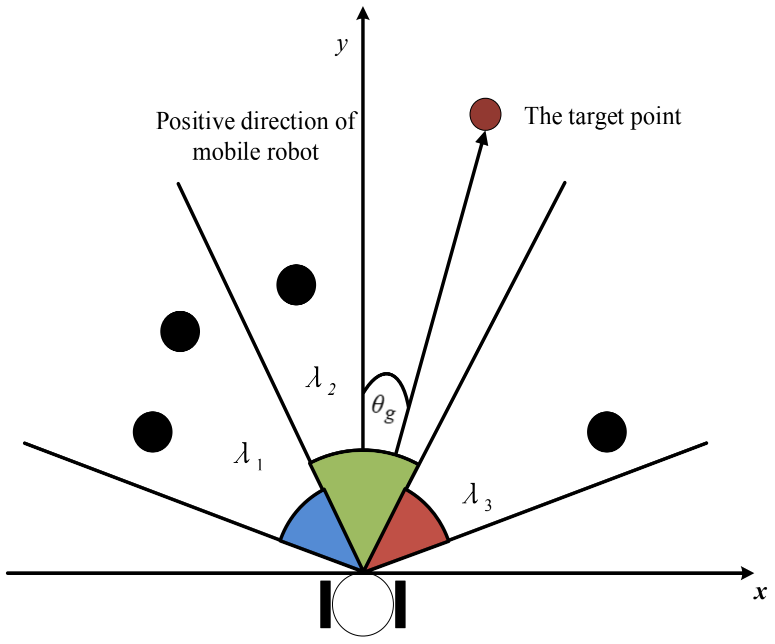

Figure 1. The coordinate system of the mobile robot.

used to detect environmental information [22]. The sector areas in the figure represent the range of detection angles of the sensors. The pose information of the robot includes the distance information d_i of the nearest obstacle returned by the three directional sensors, as well as the distance D_g and angle θ_g between the mobile robot and the target point, so the state space s_j of the mobile robot is defined as:

$$s_j = \left(d_1, d_2, d_3, D_g, \theta_g\right) \tag{1}$$

The angle detection range between 20˚ - 160˚ returned by the radar sensors is discretized into three directions λ_1, λ_2 and λ_3, and the angle range is 50˚, 40˚ and 50˚.

d_1, d_2 and d_3 are the distance information of the nearest obstacle returned from the direction of λ_1, λ_2 and λ_3, respectively. The detection distance is discretized into a maximum of 3 steps. The obstacle information d_i returned by the sensor is defined as follows:

$$d_i = \begin{cases} 0 & \left(d_i = 0\right) \\ 1 & \left(0 < d_i \le 1\right) \\ 2 & \left(1 < d_i \le 2\right) \\ 3 & \left(2 < d_i \le 3\right) \end{cases} \tag{2}$$

D_g is the distance between the mobile robot and the target point, θ_g is the angle between the direction of the mobile robot and the target point. They are calculated as follows:

$$\begin{gathered} D_g = \left(x_r - x_g\right)^2 + \left(y_r - y_g\right)^2 \\ \theta_g = \begin{cases} \arctan \dfrac{y_r - y_g}{x_r - x_g}, \left(x_r \ne x_g\right) \\ \pi/2\left(y_r > y_g, x_r = x_g\right) \\ -\pi/2\left(y_r < y_g, x_r = x_g\right) \end{cases} \end{gathered} \tag{3}$$

where (x_r, y_r) represents the position of the mobile robot at the current moment, and (x_g, y_g) represents the position of the target point.

The action space of the robot is represented by a, $a = \{a_1, a_2, a_3\}$. The current running direction of the mobile robot is taken as the reference direction, and the three actions are defined as:

a_1: move one step 45˚ to the left front;

a_2: move one step forward;

a_3: move one step 45˚ to the front right.

3. Design of Double BP Q-Learning Neural Network Structure

Double BP Q-Learning algorithm is used to learn the obstacle avoidance strategy to plan the path. The state of the mobile robot is taken as the input of the network and outputs the action to be performed. Double BP Q-learning algorithm replaces the Q value table by fitting the value function of the neural network and includes two BP neural networks with the same structure, which are the estimation networks of BP_eval and the target network BP_target [23], respectively.

The input of each network is the state space s_j of the robot. Excessive range of input data in neural network will affect the accuracy of gradient descent [24]. Therefore, the variable in the state s_j of the mobile robot are normalized as follows:

$$s_t = \frac{S_j - S_{\min}}{S_{\max} - S_{\min}} \tag{4}$$

After normalization, the state space $s_j = \left(d_1, d_2, d_3, D_g, \theta_g\right)$ of the mobile robot is converted into $s_t = \left(D\_UL, D\_U, D\_UR, D\_End, Angle\right)$, where s_t is the five input of the neural network. The output of the neural network is the Q value corresponding to the three actions taken by the mobile robot. Nodes of the hidden layer are dynamically set according to the number of nodes of input and output layer. Here, the number is set to 13. The designed structure of a single BP Q-learning neural network design is shown in **Figure 2**.

Where ω_{ij} and b_{ij} represent the weight and bias of the *i*th input layer node to the *j*th hidden layer node, and the input of the *j*th hidden layer is defined as:

$$h_j = \sum\left(\omega_{ij} s_t + b_{ij}\right) \quad \left(i, t = 1 \sim 5; j = 1 \sim 13\right) \tag{5}$$

where φ_{ij} and c_{ij} represent the weight and bias of the *i*th hidden layer node to the *jth* output layer node, and the input of the output layer of the *j*th layer is defined as:

$$o_j = \sum\left(\varphi_{ij} h_i + c_{ij}\right) \quad \left(i = 1 \sim 13; j = 1 \sim 3\right) \tag{6}$$

The parameter ω of the estimation network BP_eval and the parameter ω- of the target network BP_target include the weights and bias of each layer. BP_eval updates the ω parameter by training, BP_target is used to fix the network parameters,

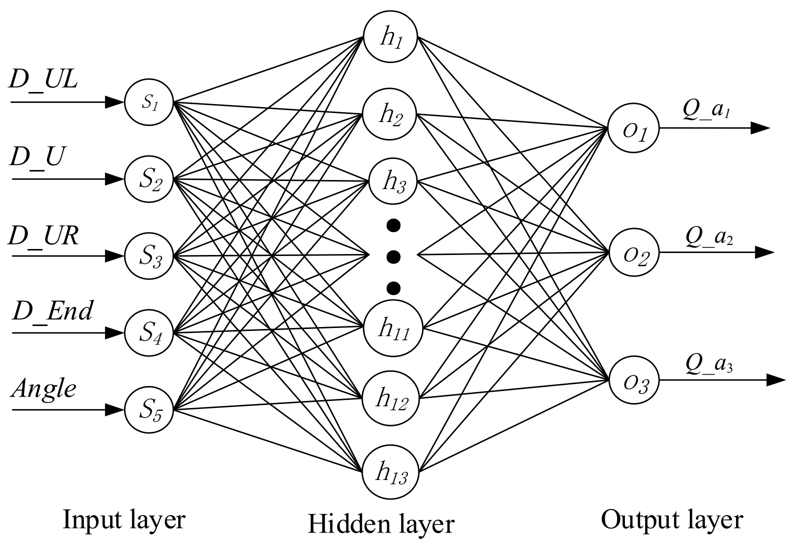

Figure 2. Structure of BP Q-Learning neural network.

and the algorithm copies the ω to the ω-regularly. The Memory part is the memory bank that stores the collected samples during learning. r_t represents the reward and punishment value of the mobile robot when it reaches the next state; BP_eval outputs three estimated Q values: $Q(e\_a_1)$, $Q(e\_a_2)$ and $Q(e\_a_3)$; BP_target outputs three target Q values: $Q(t\_a_1)$, $Q(t\_a_2)$ and $Q(t\_a_3)$. BP_eval outputs $Q(e\_a_t)$ corresponding to the action a_t, and BP_target calculates $Q(t\_a_s)$ corresponding to the action a_s corresponding to the maximum Q value output by BP_eval. Where Loss function is calculated as follows:

$$Loss = \left(r_t + \gamma Q\left(t\_a_s\right) - Q\left(e\_a_t\right)\right)^2 \tag{7}$$

Gradient descent algorithm is used to update the parameters of the network, and the network update block diagram is shown in Figure 3.

4. Double BP Q-Learning Algorithm

Using neural network to fit the value function to solve the dimensional disaster problem, aiming at the problems of slow learning speed and poor model generalization ability in Double BP Q-learning algorithm, reward and punishment function and action selection strategy are redesigned, adding the mechanism of priority experience replay and transfer learning, so that the mobile robot can solve the problem of sparse rewards and accelerate the speed of experience learning.

4.1. Design of Reward and Punishment Functions

The design of reward and punishment functions determines the good or bad degree of actions taken by mobile robot in a certain state [25]. Sparse rewards have always been a problem that affects the convergence of reinforcement learning

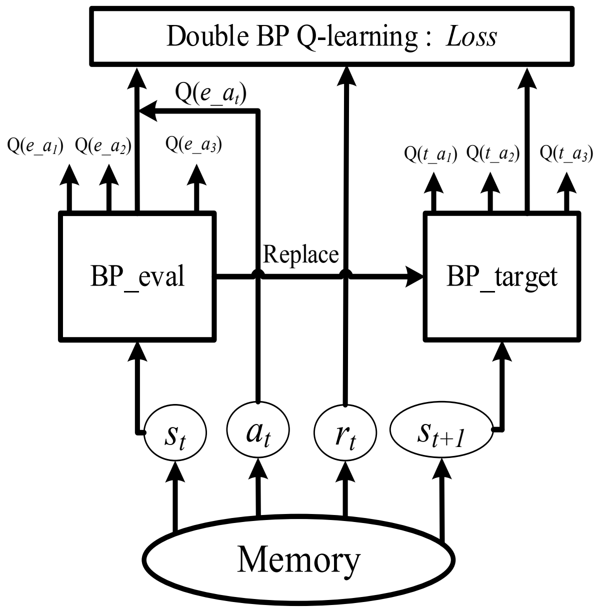

Figure 3. Network update block diagram.

algorithms [26]. A continuous reward and punishment function is designed to solve the problem of sparse rewards. It is defined as follows:

$$r_t = \begin{cases} r\_tar \\ r\_obs \\ \dfrac{D\_before - D\_now}{\mu \cdot D\_now} + \dfrac{d\_now - d\_before}{\eta} \end{cases} \tag{8}$$

where:

r_tar represents the reward for reaching the target point;

r_obs represents the punishment for colliding an obstacle;

D_before represents the distance between the mobile robot and the target point before the action is performed;

D_now represents the distance between the mobile robot and the obstacle after performing the action;

d_now represents the shortest distance between the mobile robot and the obstacle at the current state;

d_before represents the shortest distance between the mobile robot and the obstacle before the action is performed;

μ and η represent normalized discount factors.

After the mobile robot performs the action, whether it is closer to the target point or farther from the obstacle, it will be rewarded according to the calculation Formula (8); if the robot is farther from the target point or closer to the obstacle, it will be punished.

4.2. Design of Dynamic *ε*-Greedy Strategy

Double BP Q-learning algorithm is proposed to improve the exploration-exploitation of *ε*-greedy strategy, and a dynamic *ε*-greedy strategy is designed. The ε-greedy strategy is used to balance the relationship between exploration and exploitation. Its essence is to explore and select actions with *ε* probability, and to learn the existed experience with 1-*ε* probability [27].

Given an initial value of exploitation ε_0 and peak value *ε_end*, with the increase of learning frequency, the exploration rate ε_t decreases from peak value *ε_end* to zero and remains unchanged, while the exploitation rate ε_r increases from the initial value ε_0 to peak value *ε_end* and remains unchanged. The exploration-exploitation rate is computed as follows:

$$\begin{aligned} \varepsilon_r &= \begin{cases} \varepsilon_0 + \varepsilon_i \cdot t\_step, t\_step < step\_max \\ \varepsilon\_end, t\_step \geq step\_max \end{cases} \\ \varepsilon_t &= \begin{cases} \varepsilon\_end - \varepsilon_0 - \varepsilon_i \cdot t\_step, t\_step < step\_max \\ 0, t\_step \geq step\_max \end{cases} \end{aligned} \tag{9}$$

where:

ε_i represents the increase factor of exploration rate;

t_step represents the number of learning rounds;

step_max represents the maximum round of running.

The change of exploration-exploitation rate is shown in **Figure 4**.

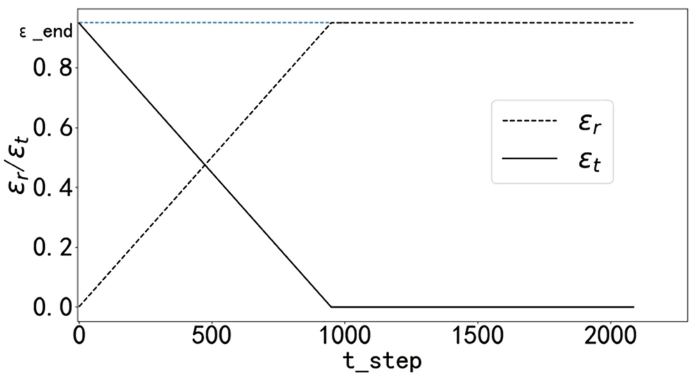

Figure 4. Change of exploration-exploitation rate.

4.3. The Mechanism of Priority Experience Replay

Although random sampling breaks the correlation between samples, it ignores the importance of experience, and introduces the mechanism of priority experience replay to store the priority of samples through the binary tree structure Sumtree. High-priority samples are selected firstly to speed up learning [28].

Double BP Q-learning algorithm collects samples in the training process, which including state s_t, action a_t, reward r_t and state s_{t+1} at the next moment. Samples (s_t, a_t, r_t, s_{t+1}) are stored in the experience pool. TD-error is used to define the sample priority, which is to estimate the difference between the Q values output by the estimation network and the target network. The larger the TD-error, the higher the corresponding priority [29]. The probability defined by the extracted sample is as follows:

$$P(i)=\frac{p_i^\alpha}{\sum_{i=1}^{k} p_i^\alpha} \tag{10}$$

where:

α represents how much priority to add to the sample. If α is 0, it is an uniform sampling:

$$p_i=\frac{1}{rank(i)} \tag{11}$$

where the $rank(i)$ is sorted by TD-error.

The higher the sample priority, the higher the probability of the sample being extracted, and the lower the sample priority is guaranteed to have a certain probability of being selected.

4.4. Transfer Learning

Most tasks in local path planning by reinforcement learning are correlated, and parameters in related tasks are initialized by parameter migration in different map environments to speed up strategy learning of mobile robots in different scenarios [30].

Firstly, the pre-training model is loaded to obtain all the model parameters.

The model parameters ω_s and ω_t to be trained are divided into two parts by using hierarchical Bayes. One part is the common task, such as the function ω_i, which tends to the target point. The other part is unique to each model parameters, such as dense discrete obstacles avoidance strategy v_s and special obstacles avoidance rule v_t. The transfer learning framework of this paper is as follows:

$$\begin{aligned}\omega_s &= \omega_i + v_s \\ \omega_t &= \omega_i + v_t\end{aligned} \tag{12}$$

The model parameter ω_i approaching the target point is obtained through random initialization training, and it is initialized into the model parameter ω_i' of the sparse discrete obstacles environment and the model parameter ω_t of the special obstacles environment. Then, combining with the special state operation rules set in Table 1 and the direction and distance information between the mobile robot and the target point, the obstacle avoidance strategy is learned. Finally, ω_i' is initialized as the model parameter ω_s in the dense discrete obstacle environment to perfect the obstacles avoidance rules.

Table 1. Specific state operation rules.

D_UL	*D_U*	*D_UR*	a_1	a_2	a_3
0	0	1	√	√	
0	1	0			
1	0	0		√	√
0	1	1	√		
1	0	1		√	
1	1	0			√
0	1	2	√		√
1	0	2		√	√
1	2	0		√	√
0	1	3	√		√
1	0	3		√	√
1	3	0		√	√
0	2	1	√	√	
2	0	1	√	√	
2	1	0	√		√
0	3	1	√	√	
3	0	1	√	√	
3	1	0	√		√
1	1	2			√
1	2	1		√	
2	1	1	√		
1	2	2		√	√
2	1	2	√		√
2	2	1	√	√	
1	1	3			√
1	3	1		√	

The structure diagram of transfer learning to local path planning is shown in Figure 5.

4.5. Learning Process of Double BP Q-Learning Algorithm

The mobile robot is trained by the designed Double BP Q-learning neural network and strategy designed above, and use the obtained model to plan local path.

The learning process of the algorithm is as follows.

Step 1. A standard blank map *Q_Map* is generated, and the number and position of starting points, target points and obstacles are randomly set. The initial direction is the connecting direction between the mobile robot and the target point.

Step 2. Initialize the parameters of Double BP Q-learning algorithm;

Step 3. The state s_t is obtained after preprocessing the map environment information returned by the sensors;

Step 4. Neural network parameters are initialized randomly in the initial training; otherwise, parameters are initialized by parameter transfer;

Step 5. The dynamic ε-greedy strategy is used to select the action a_t;

Step 6. The mobile robot performs the action a_t to obtain the reward r_t and the next state s_{t+1};

Step 7. Store $\{s_t, a_t, r_t, s_{t+1}\}$ in Memory bank as a replay unit;

Step 8. Extract small batch of experience for learning;

Step 9. If the next state is the termination state, the Q value of the target network is r_t; otherwise, the Q value is calculated through the target network as follows:

$$Q = r_t + \gamma Q\left(s_{t+1}, \arg\max_a Q\left(s_{t+1}, a_s; \omega_t\right); \omega_t - \right) \tag{13}$$

Gradient descent algorithm is performed to calculate the Loss function as follows:

$$Loss = \left(r_t + \gamma Q\left(s_{t+1}, \arg\max_a Q\left(s_{t+1}, a_s; \omega_t\right); \omega_t - \right) - Q\left(s_t, a_t; \omega_t\right) \right)^2 \tag{14}$$

Step 10. Update the estimation network parameter ω;

Step 11. Update the target network parameter ω- through copying ω to the ω- every C step.

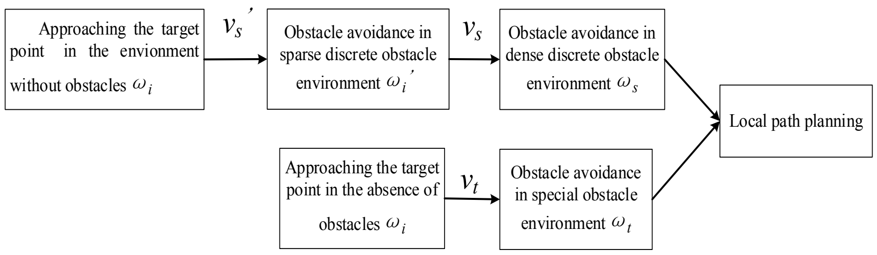
Approaching the target point in the envionment without obstacles ω_i
v_s'
Obstacle avoidance in sparse discrete obstacle environment ω_i'
v_s
Obstacle avoidance in dense discrete obstacle environment ω_s
Local path planning
Approaching the target point in the absence of obstacles ω_i
v_t
Obstacle avoidance in special obstacle environment ω_t

Figure 5. Structure diagram of transfer learning.

5. Simulation Experiment Test

After adding a series of improvement strategies, on PyCharm platform, Double BP Q-learning algorithm and the Double Q-learning algorithm are compared in different map environments, and the optimization goal is to improve the generalization ability of the model under the premise of the same path length. Record the probability of the mobile robot reaching to the target point in the process of parameter update. The simulation environment is a 21 × 21 grid environment, in which the red square grid represents the start point, the yellow circular grid represents the target point, the black square grid represents the obstacle and the black solid line represents the planned path.

5.1. Simulation Test of the Function toward the Target Point

The model parameters toward the target point are randomly initialized, and other parameters are initialized as shown in Table 2.

Using the set parameters, the mobile robot roamed in the environment without obstacles at the initial stage. After reaching the preset roaming times, small batches of experience are selected for learning. After the exploration rate rises to the preset peak value, the exploration rate remains unchanged. Continue training to the pre-training times and output the probability of successfully reaching the target point during parameters update process, the processes are shown in Figure 6.

Figure 7(a) and Figure 7(b) are the comparison of planned path toward the target point by Double Q-learning and Double BP Q-learning algorithms, respectively, where both the length of the paths is 20. The probability of the Double Q-learning algorithm successfully reaching the target point fluctuates greatly after the number of updating rounds *p_step* reaches 40 rounds and the model is unstable, as shown in Figure 6. However, the probability of Double BP Q-learning algorithm reaching the target point tends to be stable and reaches

Table 2. Parameters toward the target point.

Parameter	Initialization values
lr	0.001
ε_0	0
ε_end	0.95
ε_i	0.001
m_size	20000
γ	0.95
batch_size	256
replace_size	20
start_step	3000
learn_step	50
episode	600

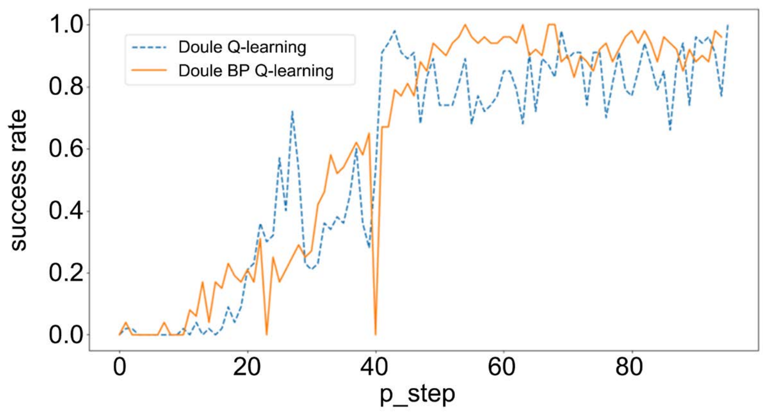

Figure 6. Probability comparison of successfully reaching the target point in each round of parameter update in the environment without obstacle.

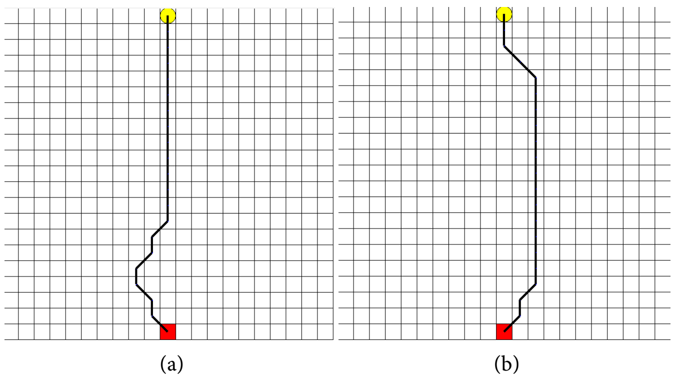

Figure 7. Path comparison toward the target point. (a) Double Q-learning; (b) Double BP Q-learning.

more than 90% in the later period of parameter update, which shows the robot can reach that the target point, with a high probability in the obstacle-free map and the model generalization ability is higher.

The double BP Q-learning algorithm is used to train the model to complete the test of reaching the target point. Figure 8 shows two scenarios that the starting point and target point are arbitrarily set, and the optimal or relatively optimal path toward the target point can be planned through the trained model.

5.2. Simulation in the Sparse Discrete Obstacles Environment

In the sparse discrete obstacles environment, model parameters approaching the target point are used as initialization to carry out model transfer. Some parameters are the same as parameters that approaching the target point, and other parameters are initialized as shown in Table 3.

By increasing the initial exploration rate ε_0 to 0.2, the mobile robot can make use of the learned rules earlier at the early stage of training and have a greater probability to reach the target point and reduce the unnecessary learning. Increase the number of initial roaming times and the memory bank capacity to make better use of the collected experience; after reaching the preset training

Table 3. Parameters of sparse discrete obstacles environment.

Parameter	Initialization values
ε_0	0.2
m_size	30000
start_step	5000
episode	800

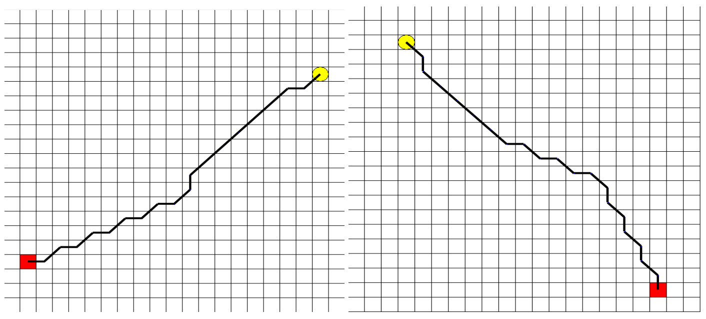
(a) (b)

Figure 8. Simulation results toward the target point. (a) Scenario 1; (b) Scenario 2.

times, the probability of successfully reaching the target point in each round of parameter update in the sparse discrete obstacles environment is output. The training results of the two algorithms are shown in Figure 9.

Figure 10(a) and Figure 10(b) show the path comparison in the sparse discrete obstacles environment between the two algorithms, and the planned paths are all 21. As shown in Figure 9, when the number of parameter update *p_step* reaches 60 rounds, Compared with Double Q-learning algorithm, Double BP Q-learning algorithm has a greater probability of the robot reaching the target point, and the planned path is more smooth and has few redundant sections, which indicates that the Double BP Q-learning algorithm has learned part of the obstacles avoidance rules in the sparse discrete obstacles environment, and the output model has higher generalization ability.

After several tests, under different sparse discrete obstacles environments, the success rate of Double BP Q-Learning algorithm model for path planning can reach 76%, while the success rate of Double Q-Learning algorithm is only 65%. Figure 11 shows partial simulation results of the trained Double BP Q-learning algorithm in the sparse discrete obstacles environments.

5.3. Simulation in Dense Discrete Obstacles Environment

In sparse discrete obstacles environment, due to the insufficient complexity of obstacle map, the model generalization ability is not high, and the strategy learned by the algorithm can't deal with most of the maps, so the sparse discrete obstacles environment model is initialized to model parameters of the dense discrete obstacles environment through parameter transfer, some parameters are the same as parameters of the sparse discrete obstacles environment, other parameters are initialized as shown in Table 4.

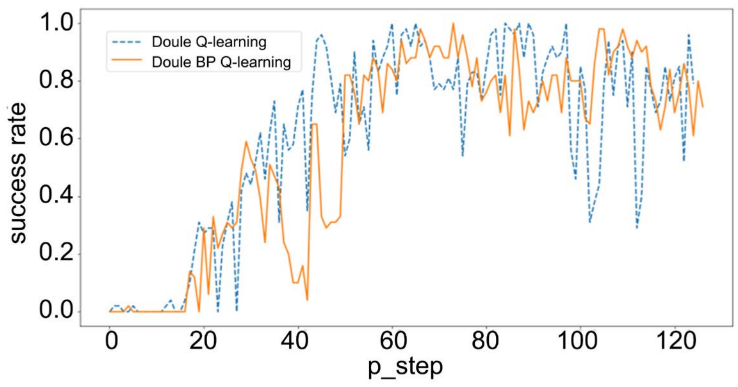

Figure 9. Probability comparison of successfully reaching the target point in each round of parameter update in the sparse discrete environment.

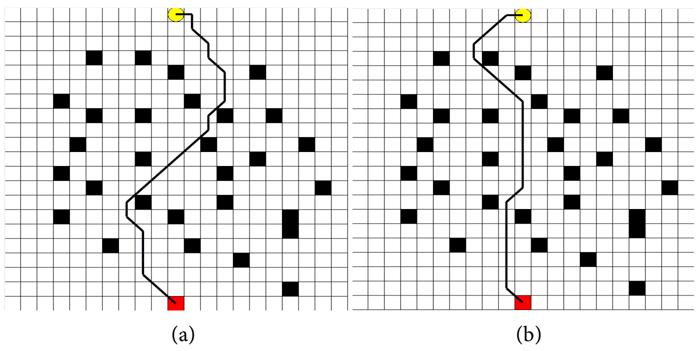

Figure 10. Path comparison in the sparse discrete environment. (a) Double Q-learning; (b) Double BP Q-learning.

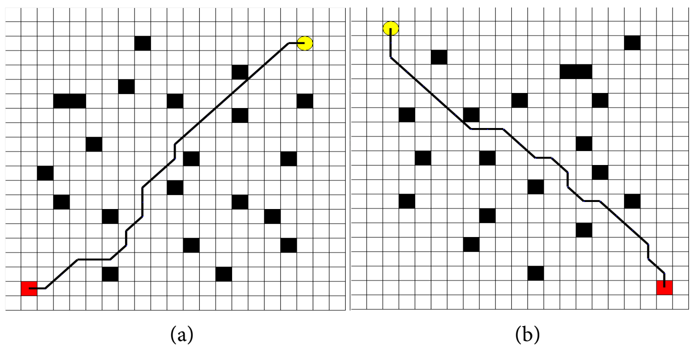

Figure 11. Simulation results of the sparse discrete obstacles environment. (a) Scenario 1; (b) Scenario 2.

Table 4. Environmental parameters of dense discrete obstacles environments.

Parameter	Initialization values
ε_0	0.2
ε_i	0.0005
m_size	40000
start_step	10000
episode	2000

Increase the number of roaming steps in the initial stage and reduce the increase extent of exploration rate. In the initial training process, the mobile robot can find more possible paths and learn more possible scenarios. At the same time, increase the memory bank capacity to store more different experiences. After reaching the preset peak value, the exploration rate remains unchanged and continues to train until reaching the preset training times and outputs the probability of successfully reaching the target point in each round of parameter update process. The simulation results of the two algorithms are shown in Figure 12.

However, as is shown in Figure 12, the probability of the Double Q-Learning algorithm reaching target point fluctuates greatly during the parameter update process, and the model is difficult to converge. However, the Double BP Q-learning algorithm can successfully reach the target point with a high probability after the number of parameter update rounds reaching 60 rounds, and the probability tends to be stable and the smoothness of the planned path is higher, which indicating that the model trained by the algorithm can be applied to most obstacles maps in dense discrete obstacles environments, and the generalization ability is stronger.

Figure 13(a) and Figure 13(b) are the comparison of the planned paths obtained by using the two algorithms respectively in a dense discrete obstacles environment, where the paths length planned by the algorithms are both 21.

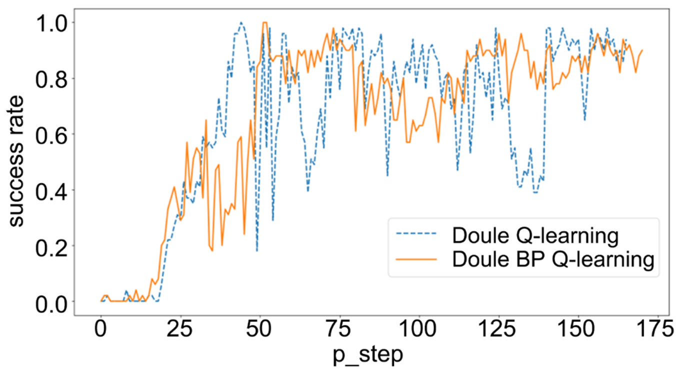

Figure 12. Probability comparison of successfully reaching the target point in each round of parameter update in the dense discrete obstacles environment.

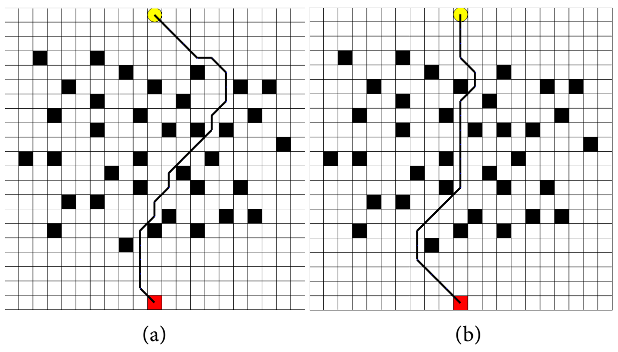

Figure 13. Path comparison of the dense discrete obstacles environment. (a) Double Q-learning; (b) Double BP Q-learning.

After several tests, in the different dense discrete obstacles environments, the success rate of using Double BP Q-learning algorithm model to plan different maps can reach 89%, while the success rate of Double Q-learning algorithm is 82%. Figure 14 shows partial simulation results of the trained Double BP Q-learning model in the dense discrete obstacles environments.

5.4. Simulation in the Special Obstacles Environment

In the special obstacles environment, there are some special scenarios that are easy to make the mobile robot enter the deadlock state. The model parameters approaching the target point are used as initialization to carry out parameter transfer. Some parameters are the same as those parameters approaching the target point, and other parameters are initialized as shown in Table 5.

Special obstacles environment mainly includes "U" type obstacles environment and "一" type obstacles environment. Mobile robot needs to avoid the special obstacles in advance or is able to escape after entering them. Through three sensors to detect obstacles information to set specific operation rules, as shown in Table 1, and combine the direction and distance information with the target point to make the mobile robot avoid entering deadlock area in advance. Reduce unnecessary study to speed up the convergence of the model. Figure 15 is the probability of reaching the target point after each round of parameter update successfully using the two algorithms.

Figure 16(a) and Figure 16(b) show the path comparison of special obstacles environment by the two algorithms, and the length is basically same. As is shown in Figure 15, the Double Q-learning algorithm converges faster in the early stage of training and has a higher probability of reaching the target point.

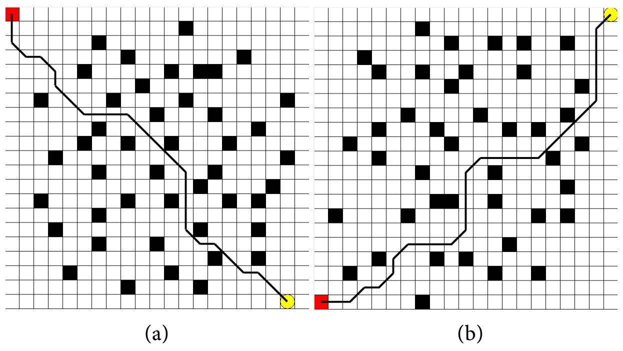

Figure 14. Simulation results of the dense discrete obstacles environment. (a) Scenario 1; (b) Scenario 2.

Table 5. Environmental parameters of special obstacles environment.

Parameter	Initialization values
ε_i	0.002
m_size	30,000
start_step	6000
episode	800

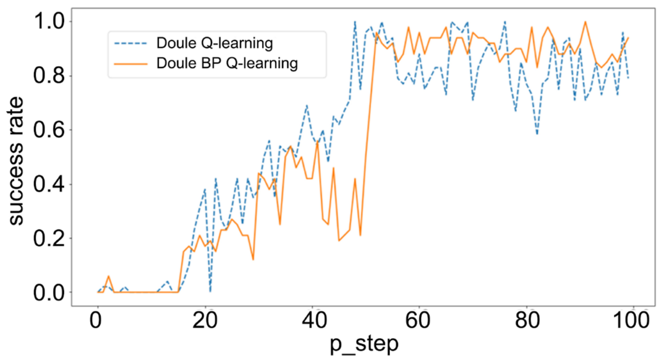

Figure 15. Probability comparison of successfully reaching the target point in each round of parameter update in the special obstacles environment.

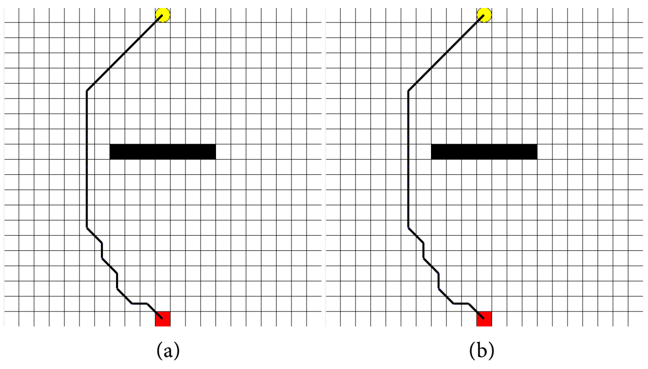

Figure 16. Paths comparison of the special obstacles environment. (a) Double Q-learning; (b) Double BP Q-learning.

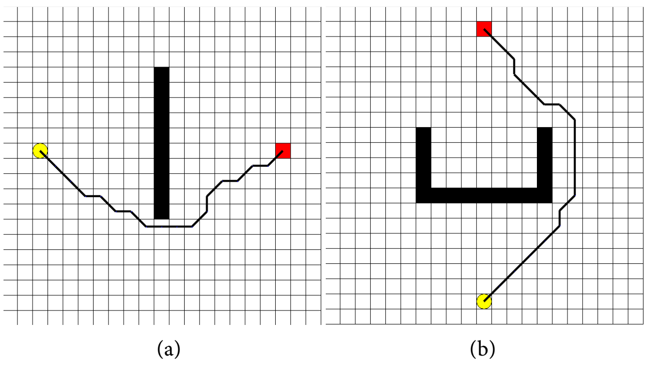

Figure 17. Simulation results of the special obstacles environment. (a) Scenario 1; (b) Scenario 2.

However, when the number of parameter update rounds reaches 50, the probability of reaching the target point is more than 90% for the model trained by the Double BP Q-learning algorithm. It tends to be stable and fluctuates less. It also has fewer redundant sections in the planned path, which indicates that the obstacles avoidance strategy learned can avoid certain deadlocks to reach the target point.

After several tests, the result is in line with the expected result of experiment. The simulation results of partial planned path under special obstacles environments using the trained Double BP Q-learning model are shown in Figure 17.

6. Conclusions

Double BP Q-learning algorithm based on the fusion of Double Q-learning algorithm and BP neural network is proposed. Compared with the traditional Double Q-learning algorithm, the designed algorithm can be applied to the operation environment with more complex obstacles. The convergence speed of the algorithm is faster, and the trained model has strong generalization ability and fewer path redundant sections. The success rate of the planned path in the dense discrete obstacles environment is higher, which can reach 89%. In the special obstacles environment, the robot can reach the target point avoiding the deadlocked area. The proposed Double BP Q-learning algorithm has the following characteristics:

1) Using BP neural network fitting value function instead of the Q value table can plan the path in a wider range of environment with more complex obstacles;

2) The mechanism of priority experience replay is added to speed up strategy learning;

3) Special operating rules are designed to help mobile robots escape from the deadlocked area under special obstacles environment;

4) Parameter transfer is used to initialize the model parameters in different obstacles environments for reducing unnecessary training, accelerating the convergence speed of the algorithm and increasing the generalization ability of the model.

Future research work is to combine deep reinforcement learning with neural network with memory function to process continuous high-dimensional state information and continuous action information, and improve path smoothness and reduce path redundancy.

Funding

This work is supported by the National Natural Science Foundation of China grant nos. 61473179, 61973184, 61573213.

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] Cheng, Y. and Wang, G.Y. (2018) Mobile Robot Navigation Based on Lidar. 2018 *IEEE Chinese Control and Decision Conference* (*CCDC*), Shenyang, 9-11 June 2018, 1243-1246. https://doi.org/10.1109/CCDC.2018.8407319

[2] Patle, B.K., Pandey, A., Parhi, D.R.K., *et al.* (2019) A Review: On Path Planning Strategies for Navigation of Mobile Robot. *Defence Technology*, **15**, 582-606. https://doi.org/10.1016/j.dt.2019.04.011

[3] Zafar, M.N. and Mohanta, J.C. (2018) Methodology for Path Planning and Optimization of Mobile Robots: A Review. *Procedia Computer Science*, **133**, 141-152. https://doi.org/10.1016/j.procs.2018.07.018

[4] Chen, Y.B., Luo, G.C., Mei, Y.S., Yu, J.Q. and Su, X.L. (2016) UAV Path Planning

Using Artificial Potential Field Method Updated by Optimal Control Theory. *International Journal of Systems Science*, **47**, 1407-1420. https://doi.org/10.1080/00207721.2014.929191

[5] Chen, J.W., Zhu, H., Zhang, L., *et al.* (2018) Research on Fuzzy Control of Path Tracking for Underwater Vehicle Based on Genetic Algorithm Optimization. *Ocean Engineering*, **156**, 217-223. https://doi.org/10.1016/j.oceaneng.2018.03.010

[6] Panov, A.I., Yakovlev, K.S. and Suvorov, R. (2018) Grid Path Planning with Deep Reinforcement Learning: Preliminary Results. *Procedia Computer Science*, **123**, 347-353. https://doi.org/10.1016/j.procs.2018.01.054

[7] Macek, K., Petrovi, C.I. and Peric, N. (2002) A Reinforcement Learning Approach to Obstacle Avoidance of Mobile Robots. 7*th IEEE International Workshop on Advanced Motion Control*, Maribor, 3-5 July 2002, 462-466.

[8] Vadakkepat, P., Tan, K.C., *et al.* (2000) Evolutionary Artificial Potential Fields and Their Application in Real Time Robot Path Planning. *Proceedings of the* 2000 *Congress on Evolutionary Computation*, Vol. 1, 256-263.

[9] Bounini, F., Gingras, D., Pollart, H., *et al.* (2017) Modified Artificial Potential Field Method for Online Path Planning Applications. 2017 *IEEE Intelligent Vehicles Symposium* (*IV*), Paris, 9-12 June 2017, 180-185. https://doi.org/10.1109/IVS.2017.7995717

[10] Guo, J., Li, C. and Guo, S. (2019) A Novel Step Optimal Path Planning Algorithm for the Spherical Mobile Robot Based on Fuzzy Control. *IEEE Access*, **8**, 1394-1405. https://doi.org/10.1109/ACCESS.2019.2962074

[11] Yen, C.T. and Cheng, M.F. (2018) A Study of Fuzzy Control with Ant Colony Algorithm Used in Mobile Robot for Shortest Path Planning and Obstacle Avoidance. *Microsystem Technologies*, **24**, 125-135. https://doi.org/10.1007/s00542-016-3192-9

[12] Zhang, Q., Li, M., Wang, X., *et al.* (2012) Reinforcement Learning in Robot Path Optimization. *JSW*, **7**, 657-662. https://doi.org/10.4304/jsw.7.3.657-662

[13] Lei, X., Zhang, Z. and Dong, P. (2018) Dynamic Path Planning of Unknown Environment Based on Deep Reinforcement Learning. *Journal of Robotics*, **2018**, Article ID: 5781591. https://doi.org/10.1155/2018/5781591

[14] Duguleana, M. and Mogan, G. (2016) Neural Networks Based Reinforcement Learning for Mobile Robots Obstacle Avoidance. *Expert Systems with Applications*, **62**, 104-115. https://doi.org/10.1016/j.eswa.2016.06.021

[15] Huang, B.Q., Cao, G.Y. and Guo, M. (2005) Reinforcement Learning Neural Network to the Problem of Autonomous Mobile Robot Obstacle Avoidance. 2005 *International Conference on Machine Learning and Cybernetics*, Vol. 1, 85-89.

[16] Saha, O., Dasgupta, P. and Woosley, B. (2019) Real-Time Robot Path Planning from Simple to Complex Obstacle Patterns via Transfer Learning of Options. *Autonomous Robots*, **43**, 2071-2093. https://doi.org/10.1007/s10514-019-09852-5

[17] Torrey, L. and Shavlik, J. (2010) Transfer Learning. In: *Handbook of Research on Machine Learning Applications and Trends*: *Algorithms*, *Methods*, *and Techniques*, IGI Global, Hershey, 242-264. https://doi.org/10.4018/978-1-60566-766-9.ch011

[18] Flennerhag, S., Moreno, P.G., Lawrence, N.D., *et al.* (2018) Transferring Knowledge across Learning Processes.

[19] Bianchi, R.A.C., Santos, P.E., Da Silva, I.J., *et al.* (2018) Heuristically Accelerated Reinforcement Learning by Means of Case-Based Reasoning and Transfer Learning. *Journal of Intelligent & Robotic Systems*, **91**, 301-312. https://doi.org/10.1007/s10846-017-0731-2

[20] Celiberto Jr., L.A., Matsuura, J.P., De Màntaras, R.L., *et al.* (2010) Using Transfer Learning to Speed-Up Reinforcement Learning: A Cased-Based Approach. 2010 *Latin American Robotics Symposium and Intelligent Robotics Meeting IEEE*, Sao Bernardo do Campo, 23-28 October 2010, 55-60.

[21] Celiberto, L., Matsuura, J.P., López de Mántaras, R., *et al.* (2011) Using Cases as Heuristics in Reinforcement Learning: A Transfer Learning Application. *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Barcelona, 16-22 July 2011, 1211-1217.

[22] Caltagirone, L., Bellone, M., Svensson, L., *et al.* (2017) LIDAR-Based Driving Path Generation Using Fully Convolutional Neural Networks. 2017 *IEEE 20th International Conference on Intelligent Transportation Systems* (*ITSC*), Yokohama, 16-19 October 2017, 1-6. https://doi.org/10.1109/ITSC.2017.8317618

[23] Tai, L. and Liu, M. (2016) A Robot Exploration Strategy Based on q-Learning Network. 2016 *IEEE International Conference on Real-Time Computing and Robotics*, Angkor Wat, 6-9 June 2016, 57-62. https://doi.org/10.1109/RCAR.2016.7784001

[24] Andor, D., Alberti, C., Weiss, D., *et al.* (2016) Globally Normalized Transition-Based Neural Networks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, 7-12 August 2016, 2442-2452. https://doi.org/10.18653/v1/P16-1231

[25] Van Seijen, H., Fatemi, M., Romoff, J., *et al.* (2017) Hybrid Reward Architecture for Reinforcement Learning. *Advances in Neural Information Processing Systems* 30: *Annual Conference on Neural Information Processing Systems* 2017, Long Beach, 4-9 December 2017, 5393-5403.

[26] Xu, N., Zhang, H., Liu, A.A., *et al.* (2019) Multi-Level Policy and Reward-Based Deep Reinforcement Learning Framework for Image Captioning. *IEEE Transactions on Multimedia*, **22**, 1372-1383. https://doi.org/10.1109/TMM.2019.2941820

[27] dos Santos Mignon, A. and da Rocha, R.L.A. (2017) An Adaptive Implementation of ε-Greedy in Reinforcement Learning. *Procedia Computer Science*, **109**, 1146-1151. https://doi.org/10.1016/j.procs.2017.05.431

[28] Xu, Z., Tang, J., Meng, J., *et al.* (2018) Experience-Driven Networking: A Deep Reinforcement Learning Based Approach. *IEEE INFOCOM* 2018—*IEEE Conference on Computer Communications*, Honolulu, 15-19 April 2018, 1871-1879. https://doi.org/10.1109/INFOCOM.2018.8485853

[29] Schaul, T., Quan, J., Antonoglou, I., *et al.* (2015) Prioritized Experience Replay.

[30] Han, D., Liu, Q. and Fan, W. (2018) A New Image Classification Method Using CNN Transfer Learning and Web Data Augmentation. *Expert Systems with Applications*, **95**, 43-56. https://doi.org/10.1016/j.eswa.2017.11.028