Path planning based on improved Deep Deterministic Policy Gradient algorithm

Yandong Liu ,Wenzhi Zhang* ,Fumin Chen ,Jianliang Li
Inner Mongolia University of Technology, Inner Mongolia University
Hohhot, China
*Email:Robotzwz@163.com

***Abstract*—Traditional DDPG algorithm experience replay is limited by the fixed replay buffer capacity, which cannot meet the demand for multi-feature data with the improvement of the learning ability of the algorithm. Aiming at the above problems, a variable capacity experience replay sampling method based on learning curve theory is proposed. By adding the learning curve to the DDPG algorithm, the algorithm realizes real-time adjustment of the replay buffer capacity according to its own learning curve, which improves the effectiveness of the sample data on the algorithm training. The simulation environment of path planning is built by using Python and Pyglet library. The simulation results show that the improved algorithm achieves better learning results.**

Keywords—Path planning; DDPG; experience replay; learning curve

I. INTRODUCTION

Path planning is a key problem that robots need to solve for navigation. The core idea is to explore an optimal or sub-optimal barrier-free path according to certain evaluation criteria in an obstacle environment to realize mobile robot path planning [1~2].

The main path planning methods of mobile robots mainly include artificial potential field method, fuzzy logic algorithm, genetic algorithm, neural network algorithm and reinforcement learning algorithm [3~4]. The traditional path planning algorithm usually assumes that the environment information is completely configured, but this method cannot meet the needs of quickly adapting to the unknown environment in practical applications. Intelligent algorithms such as neural networks and reinforcement learning are used alone for path planning, both of them have a certain degree of limitation. Deep Reinforcement Learning (DRL) method, which combines Reinforcement Learning (RL) and Deep Learning (DL), successfully solves the above problems. The agent can learn the characteristics of the original high-dimensional input data by using DL in an unknown complex environment. After that, the algorithm uses RL to make corresponding decisions based on acquired features. It provides a new perspective for solving path planning problems [5].

The successful application of deep reinforcement learning begins with the DQN (Deep Q-Learning) algorithm [6]. Because of the DQN algorithm is an algorithm oriented to discrete motion space control, it cannot be applied to the control of continuous action space. To solve this problem, Lillicrap [7] proposed the DDPG (Deep Deterministic Policy Gradient) algorithm based on the Actor-Critic algorithm framework and the DPG (Deterministic Policy Gradient) algorithm. Both the DQN algorithm and the DDPG algorithm use the sampling method of the experience replay, and randomly sample the training in the previous state transition experience. However, when the traditional DQN algorithm and DDPG algorithm train the neural network, the fixed replay buffer capacity and completely random sampling method reduce the sample utilization rate. For the shortcomings of random sampling in the experience replay, Ke Fengkai [8] proposed an optimized sampling algorithm based on TD_error and its changes, it proved the improved algorithm has improved significantly compared with the traditional algorithm. In order to improve the efficiency of the algorithm and reduce the complexity, Chen Xiliang [9] improved the traditional algorithm by using the optimal caching mechanism. This method ensures that deep reinforcement learning can obtain better samples at a higher probability for network training, and ultimately improves training efficiency of the algorithm. The traditional experience replay sampling method reduces the learning rate of the algorithm due to the fixed capacity of the replay buffer. Zong X [10] divides the learning cycle of the entire DDPG algorithm into two cycles, and increases the size of the sample in the second cycle, in this way algorithm learning rate has been greatly improved. The above optimization of the experience replay mechanism is mainly focussed on increasing the proportion of high quality samples, ignoring the impact of the replay buffer size on the algorithm. Although Zong X et al. have set up two different sizes of sample, this method still has certain limitations. The sample of two capacities cannot meet the learning rate of real-time change.

For better solve the problem of fixed learning pool capacity limit algorithm learning rate, this study proposes a DDPG algorithm based on learning curve theory. The learning curve is used to express the learning ability of the DDPG algorithm, and the replay buffer capacity is changed in real time according to the algorithm learning ability, which greatly improves the learning rate of the algorithm. By applying the algorithm to the mobile robot path planning simulation experiment, the learning curve theory can better represent the algorithm learning rate in the DDPG algorithm. Replay buffer capacity based on the algorithm learning rate change greatly increases the reward value of the agent in the path planning process. And successfully solves the problem that the replay buffer capacity suppresses the algorithm learning ability.

II. PATH PLANNING ENVIRONMENT MODEL ESTABLISHMENT

The necessary condition for implementing the path planning algorithm is to establish a robot path planning environment

978-1-5386-6243-4/19/$31.00 ©2019 IEEE

model. An environment model with high matching degree to the real environment is more effective for the training of the algorithm. At present, grid method, viewable method and topological map method are common environmental modeling methods [11]. Among them, grid method is widely used in path planning environment modeling because of its simple representation and easy implementation. However, the fixed grid structure limits the robot's exploration space to the environment, and it cannot to understand the environment map better. To solve the problem of low matching degree between the traditional modeling method and the real environment. This study builds a mobile robot working environment model based on Python and Pyglet as shown in Fig 1.

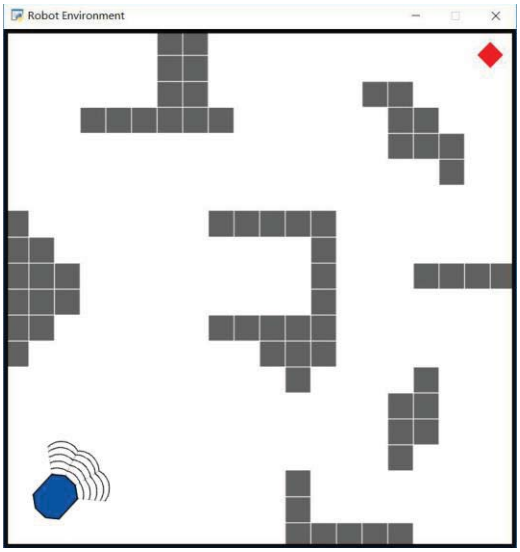

Fig. 1. Path planning simulation environment.

In this environment model of this paper, the real environment information is restored to the maximum extent, so that the robot can better understand the environment map. In the environmental model, red indicates the robot target position, black rectangle represents the obstacle, blue represents the robot body, and black curve represents the sensor detection range at the left front, front, and right front position of the robot. In the practical application, the environmental model has the following assumptions: the target position is known, the obstacle position is unknown, and there is no change in the target position and the obstacle position during the robot movement; the robot has the initial speed, and the speed does not change during the movement.

$d_{\min}$ and $d_{\max}$ are defined as the dangerous distance and the maximum detection distance between the robot and the obstacle. d is the actual distance between the robot and the obstacle. The process of robot obstacle avoidance judgment is as follows:

1) $d < d_{\min}$, the robot collides with the obstacle and initializes the robot position;

2) $d_{\min} < d < d_{\max}$, the obstacle is within the scope of the robot detection and adjusts its direction according to the obstacle position:

   a) The obstacle is in front, the robot turns left or turns right;

   b) The obstacle is located in the left front or right front, and the robot goes straight;

   c) The obstacle is located in the left front and front, the robot turns right;

   d) The obstacle is located in the right front and front, and the robot turns left;

3) $d > d_{\max}$, obstacle is not in the detection range, ignored.

III. DDPG Learning Algorithm

DDPG is the algorithm of DPG algorithm based on the Actor-Critic algorithm framework, which uses the experience replay sample collection method. It solved the problem that the original algorithm cannot deal with high-dimensional behavior space.

The Actor-Critic algorithm framework is shown in Fig 2. The Actor has the same structure as the Critic neural network and is a two-layer fully-linked neural network. Actor-Critic combines both policy-based and value-based methods. Actor uses policy gradient to learn strategies and select actions in the given environment. Critic uses policy evaluation to approximate the value function and generate signals to evaluate Actor's actions. The Actor network input environmental status output corresponding action, the Critic network input environmental status and the corresponding action output corresponding Q value.

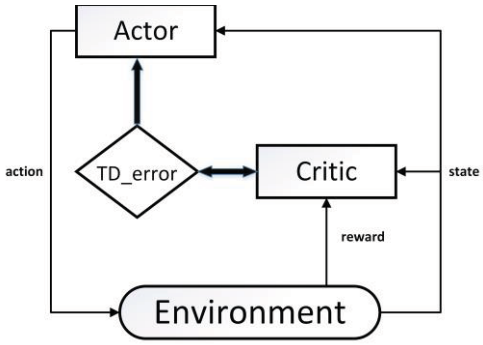

Fig. 2. Actor-Critic algorithm framework.

In DDPG algorithm, Actor and Critic are both represented by DNN (deep neural network). Actor Network and Critic Network perform function approximation for deterministic policy μ and value-action function Q respectively. The parameters are θ^{μ} and θ^{Q} respectively. When the algorithm is updated iteratively, the sample data of the replay buffer is accumulated until the minimum batch is reached. Then use the sample data to update the Critic network and update the parameter θ^{Q} through the loss function. After obtaining the gradient of the relative action of the objective function, update θ^{μ} with Adam optimizer.

When learning continuous action space, because of the strong correlation between adjacent states, continuous learning can easily make the neural network fall into the local optimal solution. The DDPG algorithm uses the sample data processing method of the experience replay to solve this problem. The experience replay process is as follows: Before the agent uses the algorithm to learn, the information of the interaction between the agent and the environment in each episode is stored in the replay buffer. It is stored in the form of data $[s_t, a_t, r_{t+1}, s_{t+1}]$. If the algorithm needs to use data, then randomly extract a set of data $[s, a, r, s']$ from the replay buffer for training. The method of experience replay effectively improves the data utilization efficiency. The random sampling method of data extraction disrupts the correlation between data and ensures data independence. It also improves the convergence speed of the algorithm.

IV. Improved DDPG Learning Algorithm

The DDPG algorithm uses the experience replay method to disrupt the data correlation and improve the convergence speed of the algorithm. But in practical application, with the improvement of learning ability, the replay buffer with fixed capacity cannot meet the requirement of data diversity, which limits the improvement of the algorithm learning ability. In view of the above problems, this study proposes a DDPG-vcep (Variable capacity experience pool) algorithm that uses the learning curve theory to adjust the capacity of the experience pool in real time.

A. Learning Curve Theory

The learning curve theory was first proposed by Wright in 1936 and it is a dynamic evaluation technique [12]. The learning curve is used to describe the process by which an individual continuously improves his or her ability to learn through the accumulation of experience. Among the many learning curves, the Wright learning curve (WLC) is the most widely used [13~14], and the learning curve equation is as follows:

$$y(x) = kx^{\alpha} \tag{1}$$

$$\alpha = \frac{\lg s}{\lg 2} \tag{2}$$

Where x is the number of explorations, k is the first learning effect, $y(x)$ is the time used for the x th time, α is the learning coefficient, and s is the working hour decreasing rate.

According to the WLC curve formula, the learning efficiency equation is constructed as follows:

$$\eta(x) = \frac{1}{k} x^{-\beta} \tag{3}$$

$$\beta = \frac{\lg \gamma}{\lg 2} \tag{4}$$

Where $\eta(x)$ is the x th learning efficiency, β is the learning coefficient, $\gamma \in (0,1)$ is the reward discount rate, and k is the default value of 1, indicating that the adjustment starts from the initial capacity of the replay buffer.

B. Improved DDPG Learning Algorithm

In this study, we combined the algorithm efficiency equations (3) and (4) to construct the replay buffer capacity change function. Ensure that as the number of training steps i increases, the experience pool capacity N can be changed accordingly. The improved algorithm is shown in the following table.

TABLE I. Improved DDPG Learning Algorithm [7]

DDPG-vcep algorithm

Randomly initialize critic network $Q(s,a|\theta^Q)$ and actor $\mu(s|\theta^\mu)$ with weights θ^Q and θ^μ.

Initialize target network Q' and μ' with weights $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$

Initialize replay buffer R

for episode=1, M **do**

Initialize a random process $\mathcal{N}$ for action exploration

Receive initial observation state s_1

for t=1, T **do**

Select action $a_t = \mu(s_t|\theta^\mu) + \mathcal{N}_t$ according to the current policy and exploration noise

Execute action a_t and observe reward r_t and observe new state s_{t+1}

Store transition (s_t, a_t, r_t, s_{t+1}) in R

Replay buffer capacity changes as follows:

$$\mathbf{C} \leftarrow \mathbf{C}\frac{\mathbf{1}}{\boldsymbol{k}}(\boldsymbol{i})^{-\frac{\log \gamma}{\log 2}}$$

Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R

Set $y_i = r_i + \gamma Q'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}$

Update critic by minimizing the loss:

$$\mathrm{L} = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta^Q))^2$$

Update the actor policy using the sampled policy gradient:

$$\nabla_{\theta^\mu} \approx \frac{1}{N}\sum_i \nabla_a Q(s,a|\theta^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i}$$

Update the target networks:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$$

end for

end for

V. Simulation Verification and Analysis

For verifying the effectiveness of the improved DDPG algorithm, experiments were carried out in the simulation environment of path planning. The traditional DDPG algorithm DDPG-nature, Zong X improved algorithm DDPG-other and this paper improved algorithm DDPG-vecp were performed 100 times, 500 times, 1000 times, 1500 times, 2000 times of algorithm training experiments. Setting the replay buffer initial

capacity is 2000, the reward discount factor γ =0.9. This computer is configured as Intel Core i5-7500 CPU, clocked at 3.4 GHz, running memory 8G; operating system is Windows 10, Python 3.6.0 version, Pyglet 1.3.1 version.

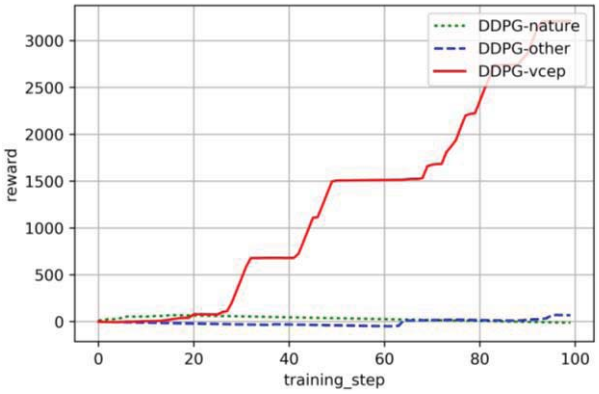

Fig. 3. Training 100 times.

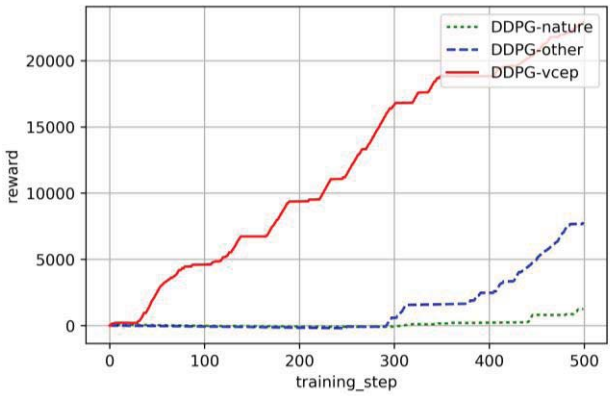

Fig. 4. Training 500 times.

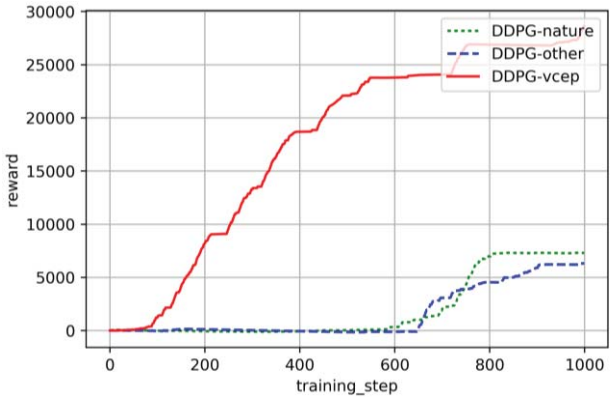

Fig. 5. Training 1000 times.

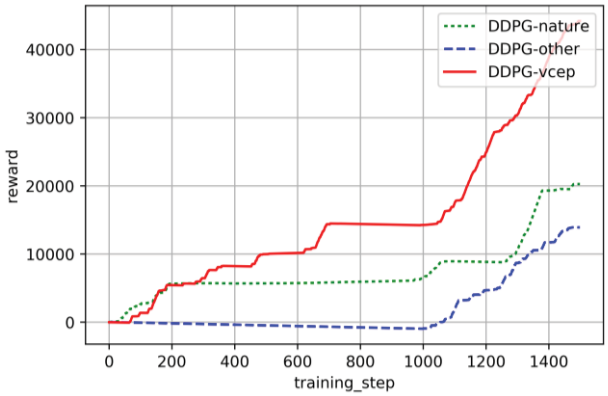

Fig. 6. Training 1500 times.

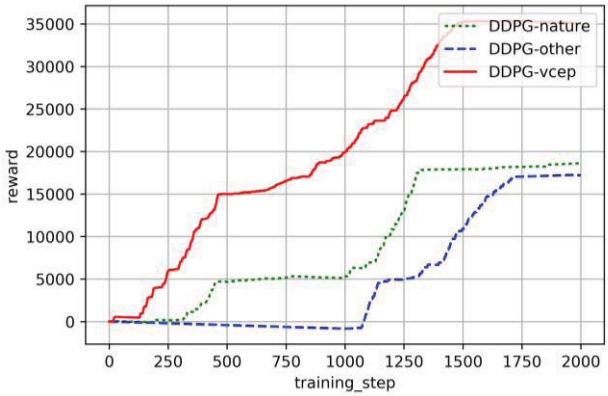

Fig. 7. Training 2000 times.

From Fig 3 to Fig 7, show the reward comparison of three algorithms at different training times. Combining the results of the algorithm for different training times, it is easy to see:

For the DDPG-other algorithm, the algorithm performs better when the number of training times is as small as 100 times and 500 times. As the capacity of the sample increases in the second half of the algorithm, the sample utilization rate increases, and the award value of the algorithm gradually exceeds the DDPG-nature algorithm. However, as the number of trainings increases, the different capacity of the sample still cannot meet the needs of the algorithm for sample data diversity.When the number of training steps is gradually increased to 1000 times, 1500 times, and 2000 times, the DDPG-other algorithm slowly does not have the advantage. For the DDPG-vcep algorithm, it always obtains a higher cumulative reward in the comparison experiments with different training times. It is shown that the dynamic adjustment of the replay buffer capacity guarantees the diversity of sample data in the training process of the algorithm, and proves the effectiveness of the improved algorithm in this study.

VI. CONCLUSIONS

In order to solve the problem that traditional path planning modeling has low matching degree with real environment, this study based on Python and Pyglet built a simulation

environment of robot path planning. Ensure that the algorithm is better adapted when it is migrated from the simulation environment to the real environment.

Fixed replay buffer capacity of the traditional DDPG algorithm limits the diversity of sample data. It is unable to meet the demand for multi-feature sample data with the improvement of algorithm learning ability, and inhibits the learning ability of the algorithm. To improve the learning efficiency of the algorithm, this paper constructs the empirical pool capacity change function based on the learning curve theory. The algorithm can dynamically adjust the capacity of the replay buffer with the improvement of its own learning ability. The comparison with other algorithms under different training times proves that the improved algorithm has obtained higher reward value and better learning ability.

REFERENCES

[1] Wang Dianjun. Indoor mobile-robot path planning based on an improved *A algorithm [J]. Journal of Tsinghua University(Science and Technology), 2012(8):1085-1089.

[2] Li Lei, Ye Tao, Tan Min, et al. Present state and future development of mobile robot technology research [J]. Robot, 2002, 24(5): 475-480.

[3] Bakdi A, Hentout A, Boutami H, et al. Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control[J]. Robotics & Autonomous Systems, 2016, 89(1):95–109.

[4] Zhu Daqi, Yan Mingzhong. Survey on technology of mobile robot path planning [J]. Control and Decision, 2010, 25(7): 961-967.

[5] Liu Quan, Zhai Jianwei, Zhang Zongchang, et al. A Survey on Deep Reinforcement Learning [J]. Chinese Journal of Computers, 2018(1): 1-27.

[6] Mnih V, Kavukcuoglu K, Silver D, et al. Playing Atari with Deep Reinforcement Learning[J]. Computer Science, 2013.

[7] Lillicrap T P, Hunt J J, Pritzel A, et al.Continuous control with deep reinforcementlearning[J]. Computer Science, 2015, 8(6):A187.

[8] Ke Fengzhen, Zhou Weizhen, Zhao Daxing. An Optimized Deep Deterministic Policy Gradient Algorithm for Rapid Localization of Robot Arm [J]. Computer Engineering and Applications.

[9] Chen Xiliang, Cao Lei, Li Chenxi, et al. Deep reinforcement learning via good choice resampling experience replay memory [J]. Control and Decision, 2018(4).

[10] Zong X, Xu G, Yu G, et al. Obstacle Avoidance for Self-Driving Vehicle with Reinforcement Learning[J]. SAE International Journal of Passenger Cars - Electronic and Electrical Systems, 2017, 11(1).

[11] Che H, Wu Z, Kang R, et al. Global path planning for explosion-proof robot based on improved ant colony optimization[C]// Intelligent Robot Systems. IEEE, 2016:36-40.

[12] Wright T.Factors affecting the costs of airplanes[J] .J of Aeronautical Sci, 1936, 3(4):122～128

[13] Xiao Qianqiao, Li Yibing, Zuo Shaoxiong, et al. Dual-resource configuration of manufacturing system based on lean production considering learning curve [J]. Computer Integrated Manufacturing Systems, 2016, 22(12): 2800-2808.

[14] Yin Xiang, Chen Wenying. Cost of carbon capture and storage and renewable energy generation based on the learning curve method [J]. Journal of Tsinghua University(Science and Technology), 2012(2): 243-248.