Advances in Intelligent Systems Research, volume 147

International Conference on Network, Communication, Computer Engineering (NCCE 2018)

Local Trajectory Planning of Mobile Robot with Deep Reinforcement Learning Based on Q Value

Yunxiong Wu [a)]

School of computers, Guangdong University of Technology, Guangzhou 510003, China.

[a)] Corresponding author: 2441596205@qq.com

Abstract. The deep reinforcement learning algorithm based on visual perception and intelligent decision combines the perception ability of convolutional neural network with the decision control ability of reinforcement learning via end-to-end learning style and realizes the process from raw visual input to decision action output. It has been extensively applied to high-dimensional visual input and decision control tasks since it was put forward. In this paper, the deep reinforcement learning algorithm based on Q value was proposed to realize local trajectory planning of mobile robot in a dynamic environment. Compared with the vulnerability of artificial design expert system, this algorithm possesses stronger robustness. By realizing the transformation from experience-driven man-made features into data-driven representation learning, this algorithm has greatly improved the real-time obstacle avoidance performance of robots.

Key words: Mobile Robot, Intelligent Decision-making, Visual Control, Local path planning.

INTRODUCTION

As the application areas of mobile robot [6]. become increasingly extensive, higher and higher requirements are raised for the intellectualization of mobile robot. The mobile robot should be able to move along the path from a given starting point to the target point in a complicated dynamic environment and conduct real-time local trajectory planning to avoid obstacles when encountering dynamic obstacles. The mobile robot is required to bypass all dynamic obstacles safely with the minimum deviation from trajectory. But if the robot acquires local environmental information by relying on sensors with limited perception when there is no prior information in the dynamic environment, inaccuracy will be inevitably caused to the environmental map model established. Local trajectory planning depending on uncertain environmental model will certainly result in uncertainty of transmissibility.

In order to solve the problem of control for the robot's intelligent decisions in a complicated dynamic environment, we proposed the deep reinforcement learning [1-2], method based on Q value. It extracts features of the robot's current local environmental information with deep convolutional neural network as the decision basis of reinforcement learning and realizes the mapping of state perception information into motion activity through end-to-end learning style. Hence, the problem of trap area and local minimum existing in traditional local trajectory planning algorithms can be solved.

RELEVANT WORK

Convolutional Neural Network

Originating from artificial neural network [3], convolutional neural network is often composed of multilayer nonlinear arithmetic units by setting output of lower layer as input of higher layer. It will automatically study complicated characteristic mapping from the mass training data, so as to discover distributed characteristics [5], of data. Compared with traditional pattern recognition models, it can extract relevant characteristics and eliminate

irrelevant information from the image through handcrafted feature extractor. A more interesting pattern of deep learning is to complete the task of image characteristic extraction by relying on the built-in self-learning function of feature extractor.

Reinforcement Learning

Reinforcement learning [4]is a branch of machine learning. Compared with other classic machine learning algorithms, it is an algorithm of studying and solving sequence multistep decision-making problems in interaction and aims to search the optimum strategy π that can make the intelligent agent acquire the maximum cumulative return, which will be its biggest characteristic.

As for the intelligent agent of reinforcement learning, the sum of rewards from time t to time T is defined as:

$$\mathrm{R}_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} \tag{1}$$

The discount attenuation coefficient $\gamma \in [0,1]$ is introduced, to avoid falling into infinite loop, and to measure the value proportion of future return at the current moment.

The state action value function $\mathrm{Q}^{\pi}(s,a)$ is defined as the target maximum cumulative return of intelligent agent of reinforcement learning, and the long-term expected return of strategy π after taking action a under states is quantified. Till the plot of strategy π ends, it is defined as:

$$Q^{\pi}(s,a) = E\left[R_t \mid s_t = s, a_t = a, \pi\right] \tag{2}$$

For all state action pairs, if the expected return of one strategy π^* is greater than or equal to the expected return of all other strategies, then strategy π^* will be called optimum strategy [7]. It is defined as:

$$Q^*(s,a) = \max_{\pi} E\left[R_t \mid s_t = s, a_t = a, \pi\right] \tag{3}$$

Formula (3) is called optimum state action value function, and this optimum state action value function follows Bellman's optimal equation, i.e.:

$$Q^*(s,a) = E_{s'\sim S}\left[\mathrm{r} + \gamma \max_{a'} Q(s',a') \mid s, a\right] \tag{4}$$

In traditional RL, Q value function is often solved through the iteration of Bellman equation:

$$Q_{i+1}(s,a) = E_{s'\sim S}\left[r + \gamma \max_{a'} Q_i(s',a') \mid s, a\right] \tag{5}$$

DEEP REINFORCEMENT LEARNING ALGORITHM BASED ON Q VALUE

In this paper, the Q learning method [9] based on time difference was adopted. The intelligent agent of reinforcement learning based on value function focuses on value function only, rather than try to understand how the model works. It is a method to evaluate the Q value of each action first, and then to solve the optimum strategy $\pi(a \mid s)$ according to Q value. The strategy function can be obtained indirectly from value function. The interaction process between intelligent agent of reinforcement learning and environmental kinetic model forms a closed loop between the robot's situation awareness and decision control to adjust the weight of convolutional neural network model. The process observed by the intelligent agent of reinforcement learning is a partially observable Markov decision process, i.e.:

$$P(S_{t+1}, R_{t+1} \mid S_0, A_0, R_1, \ldots S_t, A_t) = P(S_{t+1}, R_{t+1} \mid S_t, A_t) \tag{6}$$

In Q-learning method based on time difference, action strategy (strategy producing data) and strategy to be evaluated are not the same strategy. The action strategy produces data through strategy ε - greedy . The current behavior value function is updated via time difference target.

State Action Value Function Representation

As for the conditionality problem of Q-learning, it updates and iterates Q value based on chart method according to the past state action space, thus its applicable state and action space are very small. Q-learning cannot process a state that never appears. In other words, Q-learning lacks generalization ability. In order to endow Q-learning with predictive ability, we fitted Q value with function via regression method: $Q(s,a;\theta) \approx Q^*(s,a)$. θ represents the model parameter, and the model is either linear or nonlinear. In this paper, convolutional neural network was used to fit Q value, and the network weight was updated with small batch data showing stochastic gradient descent directly from raw input. Generally speaking, it can gain generalization ability better than manual design features.

$$L(\theta) = E[(r + \gamma \max_{a'} Q(s', a': \theta) - Q(s, a: \theta))^2] \tag{7}$$

$$\frac{\partial L(\theta)}{\theta} = E[(r + \gamma \max_{a'} Q(s', a') - Q(s, a)) \frac{\partial Q(s, a, \theta)}{\partial \theta}] \tag{8}$$

Preprocessing and Convolutional Neural Network Model System Structure

The input of deep convolutional neural network is the 80x80 original image perception region centering on robot, and this state means environmental perception scope of laser radar and camera sensor. Besides, the environment on which the robot makes a decision is also based on this scope, and it is not necessary to make a decision in the total perception scope. Three layers of CNN and two layers of FNN are adopted in network architecture design. The network inputs state S only and the output is Q value corresponding to various offline actions a. The network structure is presented in Fig. 1.

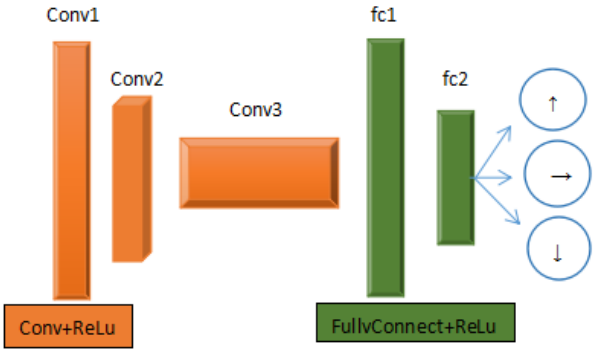

FIG. 1. Convolutional network structure

EXPERIMENTAL RESULTS AND CONCLUSION

Training Visualization of Convolutional Neural Network Model

The network input is a local state image centering on the robot provided by the simulation environment, and self-interactive learning is conducted according to reward and punishment function set for dynamic obstacle avoidance. The reward function ranges from 10 to -6. Such rewarding mode has restricted the scale of derivatives and made training easier. Meanwhile, it can make the intelligent agent of reinforcement learning tend to high-return reward of different scales. During network training of each time, 32 data were used, and the stochastic gradient descent algorithm was applied. The behavioral strategy in the training process was random – greedy search algorithm. Network training was conducted for 6 million times, and experience replay pool [8] of 200 thousand frames was used. We conducted model evaluation through calculating loss function and timely return of the model at regular intervals in the network model training process. Generally speaking, Timely returns often contain noise. as a tiny change of strategy weight will lead to the variation of distribution of various policy access states. Fig. 2 presents the variation trend of loss function value, and Fig. 3 shows the change of timely return in the robot training process. According to the figure, after neural network training reaches the 60th stage, the network tends to converge and become relatively stable. The network divergence problem was not encountered in the experiment.

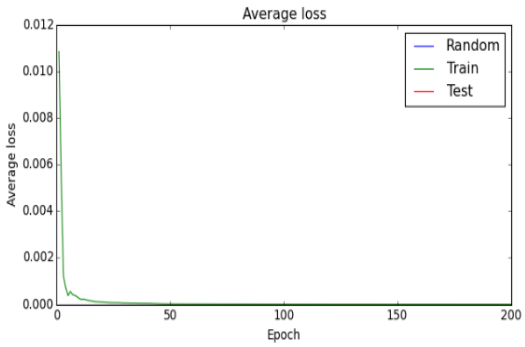

FIG.2. Network training loss function

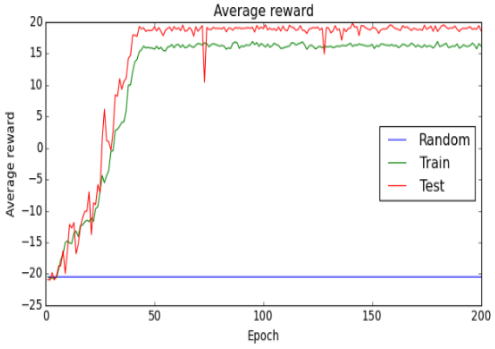

FIG.3. The average state action value

Experimental Results and Conclusion

The fragment screenshots of Fig. 4 show experimental results about the local trajectory and dynamic obstacle avoidance of mobile robot. The green mobile robot ball moves from the initial position at the left bottom to the target point at the top right corner along the trajectory, and the blue obstacle ball moves from the top right corner to the left bottom along the trajectory. As shown in the figure, when the mobile robot moves toward the target point along the trajectory and encounters the obstacle, the deep reinforcement learning algorithm will choose the optimum action according to the requirement of minimum deviation from trajectory and dynamic obstacle avoidance. The robot will continue to move toward the target point after avoiding the obstacle.

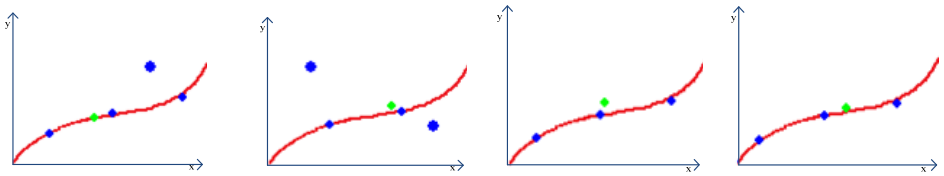

FIG.4. Fragment screenshots of mobile robot trajectory tracking and dynamic obstacle avoidance

The experiment shows that the deep reinforcement learning algorithm based on Q value is characterized by simple implementation structure, small calculation amount and good instantaneity in local trajectory planning and possesses very strong robustness for the state space of mobile robot. The path planned can not only conduct dynamic obstacle

avoidance effectively in a short time, but also strengthen environmental adaptation of the system. The reinforcement learning method has realized the transformation from single task solving into solving of a group of tasks, but the application scope is still in low-dimensional and discrete action space.

REFERENCES

1. Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning//Proceedings of Workshops at the 26th Neural Information Processing Systems 2013. Lake Tahoe, USA, 2013:201-220
2. Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540):529-533
3. Yu Kai, Jia Lei, Chen Yu-Qiang, Xu Wei. Deep learning: yesterday, today, and tomorrow. Journal of Computer Research and Development, 2013, 50(9): 1799-1804 (in Chinese)
4. Watkins C J C H. Learning from delayed rewards. Robotics & Autonomous Systems, 1989, 15(4): 233-235
5. Sun Zhi-Jun, Xue Lei, Xu Yang-Ming, Wang Zheng. Overview of deep learning. Application Research of Computers, 2012, 29(8):2806-2810 (in Chinese)
6. Kober J, Peters J. Reinforcement learning in robotics: a survey. International Journal of Robotics Research, 2013, 32(11): 1238-1274
7. Sutton R S, Barto A G. Reinforcement learning: an introduction. Cambridge: MIT press, 1998
8. Lin L J. Reinforcement learning for robots using neural networks. USA: Defense Technical Information Center, DTIC Technical Report: ADA261434, 1993
9. Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. Computer Science, 2016, 8(6): A187.