Autonomous Car Driving Based on Deep Reinforcement Learning

Zelin Zhang(✉)

Financial Big Data, Shandong University of Finance and Economics, Jinan 250014, Shangdong, China
zzl918955817zzl@163.com

Abstract. Autonomous driving car is an important direction for the future automobile development. In order to make its algorithm have better learning ability and decision-making ability, this paper proposes the M_TD3 algorithm by improving the TD3 algorithm. Improve the sampling method and redivide the experience pool into temporary, success and failure experience pools, with the data structure of the binary tree for each experience as a node. Through a large number of simulation experiments, the model of this algorithm is constructed and analyzed and verified with other algorithms. It is proved that the vehicle controlled by the M_TD3 algorithm has a higher running speed and has a guarantee of high safety and high comfort, besides the experiment verified the feasibility of this model.

Keywords: deep reinforcement learning · autonomous driving · twin delay depth certainty gradient strategy algorithm

1 Introduction

Cars provide great convenience for people's daily travel, but they also bring many potential dangers.Due to road safety, traffic conditions and other reasons, traffic accidents are common [1]. Conventional car driving has many advantages, but the number of deaths from car accidents every year is also a thrilling number.According to statistics, the average number of people dying in car accidents every day in China is about 200 [2]. Self-driving cars can provide people with a more comfortable experience, algorithmically controlled cars can well avoid these dangers, and deep reinforcement learning applied to autonomous driving can better solve various problems.

2 Related Theory

2.1 Reinforcement Learning Model

Reinforcement learning [3]. The essence is achieved through the Markov decision process, the pentple {S, A, P, R,} representation, S is the set of observed environmental states, at any time step t, the agent will first receive the state S of the observed current

F. Balli et al. (Eds.): ESFCT 2022, AEBMR 226, pp. 835–842, 2022.
https://doi.org/10.2991/978-94-6463-052-7_95

environment.In MDP, the environment is all observable, but in practical problems, the default environment is completely observable; A is a finite set of actions, the intelligent experience determines the next action according to the observed state and reward value; P is the state transfer matrix, expanding the finite-dimensional state transition matrix to the infinite-dimensional probability function; R receives an immediate reward function R from the current $state_t$; Is the reward discount factor. In addition, the intelligence follows the current state s of the strategy, that is, the mapping of a state s S and an action a A to the action probability distribution (a|s). The Markov decision process obtains the maximum expected reward by using the optimal policy to detrain the model to obtain the optimal agent. $\Pi(a|s) = p(A_t = a|S_t = s)$

2.2 The Actor-Critic Algorithm

The actor-critic algorithm (Actor-Critic) is a method based on both value and strategy. Actor network refers to the policy gradient algorithm can be used to make the continuous read action simple generation, it has the advantages that the traditional Q value learning does not have.The update process uses the time difference error term to represent the estimation error of two different state value functions. The mathematical expression is as follows:

$$\delta_t = r_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{1}$$

among:r_{t+1}For current reward; for discount factor; V (s_{t+1}) And V (s_t) Represents the reward values generated by the current and previous time steps, respectively.

3 Improved Autonomous Driving Car Model Based on the TD3 Algorthim

3.1 TD3 Algorithm

Deep deterministic gradient strategy algorithm (DDPG) for handling continuous action during simulation model construction of autonomous vehicles [4]) Has achieved good results, but it is easy to cause the phenomenon of overfitting, model in the process of update iteration for high Q value is extremely sensitive, produce convergence phenomenon, and twin delay depth deterministic gradient strategy algorithm (TD3) in the original DDPG algorithm on the basis, mainly use the following three key technologies:

(1) Truncated Double Q-Learning

The max operation in the DDPG algorithm leads to the Q overestimation problem, because the max operation selects the maximum Q value for each state, which makes the algorithm sensitive to overestimate the Q action, while the addition of noise makes the problem more obvious. The TD3 algorithm uses two Q-value networks and takes the smaller that value to calculate the Bellman equation to avoid the imprecision caused by the overestimation problem.approach

$$Q_{\theta_1'}(s', a') = Q_{\theta_1'}(s', \pi_{\varphi 1}(s')) \tag{2}$$

Fig. 1. Vehicle operation interface

$$Q_{\theta_2'}(s', a') = Q_{\theta_1'}(s', \pi_{\varphi 1}(s')) \tag{3}$$

The TD3 constructs the loss function by using the following formula:

$$\delta^{TD3} = r + \gamma(1 - d)\min_{i=1,2} Q_{w_i'}(s', a') - Q_w(s, a) \tag{4}$$

(2) Delay the Policy Update

The Actor network parameters are updated less frequently than the target network, and usually only after the Critic network update 2. This is because the target network requires multiple iterations to converge, and it provides the update target in the learning process of the algorithm, and the iteration under the error estimation will lead to more divergent policy update. If the error brought by the multiple iterations can be reduced, it can make its network have smaller variance.

(3) Smooth Operation of the Target Strategy

A problem in the deterministic strategy is that there may be overfitting to narrow peaks in the value space. By adding truncated normal distributed noise to each action, the calculation of Q value is smoother and avoids the phenomenon of overfitting.

3.2 AirSim Simulation Platform

AirSim simulation platform is an open source simulation simulator based on physical virtual engine Unreal developed by Microsoft. It has high performance physical simulation function but also has high rendering visual picture, which is suitable for simulation verification in computer vision fields such as deep learning.At the same time, the AirSim platform provides a variety of API interfaces that users can use to read data, control weather, and roads.

This paper uses the urban simulated road based on AirSim platform as the simulation experimental environment. Figure 1 Collect the depth view at the time step (Fig. 2), divide the view (Fig. 3) and the scene view (Fig. 4). In the model learning process, the completion of the road is a successful learning experience.

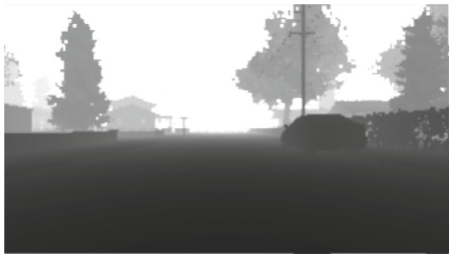

Fig. 2. An in-depth view

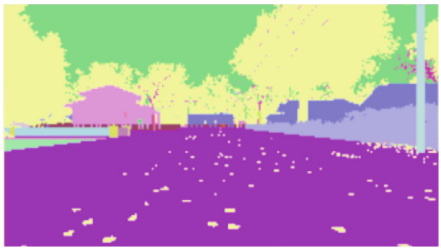

Fig. 3. Partition view

Fig. 4. Scene view

3.3 Model Ideas for Using the M_TD3 Algorithm

The TD3 algorithm uses the traditional empirical playback method, which uses the random sampling method to iteratively update the parameters of the neural network. However, such a sampling method will get uneven quality samples, making the training effect poor. Therefore, this model improves the experience pool into two parts: Ms and Mf to store successful and failed driving experiences, respectively.

Secondly, this model uses a modified method of priority experience playback, namely efficient priority experience extraction. First, its priority value will be set for the experiences in the experience pool. When the data structure of the binary tree is used, the leaf nodes are the priority values of each experience, the leaf nodes constantly stack up to form the binary tree, and the value of the root node is the sum of the priority values of all experiences.The values of the root nodes in the sampling season were divided by

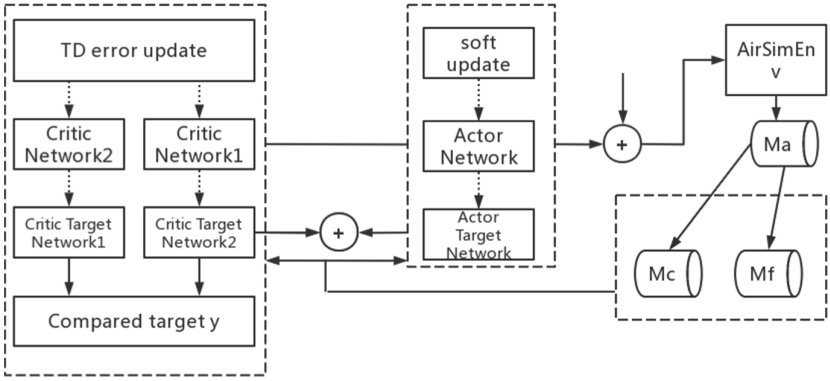

Fig. 5. Model structure of an autonomous vehicle based on the M_TD3 algorithm

batch_size and divided into batch_size intervals, and a node value was randomly drawn within each interval to find the values of the leaf nodes in a top-down manner.

Due to the characteristics of reinforcement learning, the learning experience in the experience pool affects the movement of the car at this time step. If the vehicle has successfully traveled for a turn at step t, the driving experience is default to the model.Excluding the experience pool Ms and Mf, the temporary experience pool Ma stores the recent x driving experience, each learning stores the experience to the temporary experience pool Ma and sets a priority value for the experience according to the corresponding Q value. After the temporary experience pool Ma is full, the first sequence experience is assigned to the experience pool Ms and Mf.Within the Ms and Mf experience pools, the binary tree is constructed according to the priority value of the learning experience.

Figure 5 is a schematic diagram of the learning process of this autonomous vehicle model. The main body of the model algorithm is the TD3 structure, using the following 6 deep neural networks: 1 Actor network, 1 Actor Target network, 2 Critic networks, and 2 Critic Target networks. At the time step t, the Actor network generates the current action feedback to the vehicle based on the collected environmental state and driving strategy. After the vehicle performs this action, the current image information and the vehicle state will be collected to calculate the current reward value according to the reward function and pass the corresponding parameters to the different Target networks for the next learning. Finally, the learning experience is stored in the temporary experience pool Ma and the corresponding priority value is calculated. After the temporary experience pool Ma is full, it is stored in the Ms and Mf experience pools according to the corresponding rules.

3.4 Reward Function

Reinforcement learning is the process that maximizes the reward value when interacting. In this paper, the reward function will be designed from the perspective of safety, and then it will be required to remain stable during driving to improve traffic efficiency and ride comfort.

3.4.1 Reward and Zero Clearance

The reward reset round ends with any of the following conditions during the vehicle:

(1) The car hits an obstacle
(2) The car speed is less than 2
(3) The minimum distance between the car and the road center is greater than 3.5
(4) With the maximum time of the set round, the running time exceeds the set time

3.4.2 Speed and Minimum Distance Coefficient of Vehicle and Road Center R1

The speed of the vehicle is the reward function of the vehicle. In order to ensure the operation efficiency of the vehicle, the agent gets the higher the reward at the faster, and sets the speed limit v in order to avoid traffic rule violation$_{max}$, If the speed exceeds the v_{max}, you will get a lot of punishment accordingly.

Order reward R1 used nonlinear associated with the minimum distance d of the car and pavement center with larger d value and smaller R1.

$$R1 = \alpha 1 \min\left\{\frac{v_{max} - v_e}{v_{max}}, 0\right\} - \alpha_2\left\{\frac{v_{max} - v_e}{v_{max}}, 0\right\} - \frac{d^2}{d_{max}^2} \tag{5}$$

3.4.3 Safety Factor R2

The safety factor uses the front spacing (Headway, WH) values HW and TTC values

$$R2 = -\alpha_3 \max\left\{0, \frac{3.5 - t_{ttc}}{3.5}\right\} - \alpha_4 f(H_w) \tag{6}$$

$$s^{'}(v, \Delta v, T) = s_d + \max\left\{0, v_e P + \frac{v_e \Delta v}{2\sqrt{a_1 b}}\right\} \tag{7}$$

among, s_d is the minimum distance in front; a1 is the maximum acceleration; b is the comfortable deceleration; P is the desired distance in front.

Below the car spacing [5]:

$$f(H_w) = \begin{cases} -\min\left\{k_1\left(\frac{H_w - s_{min}^{'}}{s_{min}^{'}}\right)^2, 1\right\} H_W < s_{min}^{'} \\ -\min\left\{k_1\left(\frac{H_w - s_{max}^{'}}{s_{max}^{'}}\right)^2, 1\right\} H_W > s_{max}^{'} \end{cases} \tag{8}$$

Among them, k1 and k2 are constant, because k2 is prone to collision hazard than expected hours, so $k1 < k2 < 0$ is selected.

3.4.4 Comfort Correlation Coefficient R3

When training the model, we should make the passengers more comfortable travel experience as far as possible, so there are:

$$R3 = \propto_5 \left(\max\{|a_e| - a_s, 0\} + \min\left\{\frac{a_e^{2\prime}}{a_s^{2\prime}}, 1\right\} + \max\{|\Delta\theta| - 0.1, 0\}\right) \tag{9}$$

In formula a_e is the acceleration; a_s is the maximum comfort and acceleration[6]. In summary, the reward function for the vehicle is:

$$R = R1 + R2 + R3 + C \tag{10}$$

The constant item C is set to make the car sustainable.

4 Simulation and Analysis

On the AirSim simulation platform, the implementation of the deep reinforcement learning decision control algorithm and the comparison algorithm is completed according to the Pytorch deep learning development framework. Through a large number of simulation tests and comparative tests, we analyze the operating energy efficiency of autonomous vehicles under the control of the proposed deep reinforcement learning algorithm, and verify the effectiveness and advantages of the proposed method.

4.1 Experimental Results

According to the records of successful car operation under different algorithms, the M_TD3 algorithm first appears successful driving behavior in 200 rounds, while the TD3 algorithm appears successful driving behavior in 900 rounds for the first time.

4.2 Safety Assessment

As can be seen in Fig. 12, compared with the traditional TD3 algorithm, the M_TD3 algorithm controlled autonomous vehicle TTC value is lower and there is a smaller probability of TTC value less than 1.5s (higher probability of risk when TTC is less than 1.5 s).

4.3 Driving Efficiency Assessment

According to the calculation of the front time distance (THW), the autonomous vehicle controlled by the M_TD3 algorithm has a more aggressive driving behavior, and the THW value is densely distributed between 1 and 2s, which complies with the setting of the safety factor R2 in the reward function.

4.4 Comfort Assessment

Jerk value reflects the ride experience comfort, the value and the comfort degree inverse relationship, from the calculation of the M_TD3 algorithm control autonomous vehicle Jerk value maximum absolute value is relatively small, and all the data are within the prescribed range of Jerk value, and the traditional TD3 algorithm generated driving data in nearly 8% of the data does not belong to reasonable Jerk value.

5 Conclusion

Through comparative analysis, the traditional TD3 algorithm takes longer time to learn but improved M_TD3 algorithm can be more efficient to explore and learn, and this model has a higher avoidance rate, ensuring the faster speed while its higher TTC value and THW value conforms to the safety factor, with reliable safety guarantee.We successfully verify the reasonable driving of this model on complex urban roads, and prove the feasibility of the improved TD3 algorithm for car driving.

References

1. Touran A, Brackstone MA, McDonald M.A collision model for safety evaluation of autonomous intelligent cruise control [J]. Accident Analysis & Prevention, 1999, 31(5): 567–578.
2. 2014 Statistics of National Road Traffic Accident Data [EB/OL]. [2017–03–08]. http://www.peichang.cn.
3. Sutton R.S., Barto A.G. Reinforcement learning: an introduction [J]. IEEE Transactions on Neural Net works, 1998, 9(5): 1045.
4. Lillicrap TP, Hunt JJ, Pritzel A, et al. Continuous Control with Deep Reinforcement Learning[J]. arXiv preprint arXiv: 1509. 02971, 2015.
5. Chen Hui, Wang Jiexin. A Decision-making Method for Lane Change of Automated Vehicles on Freeways Based on Driver's Dissatisfication [J]. China Journal of High way and Transport, 2019, 32(12): 1-9.
6. Kesting A, Treiber M, Helbing D, General Lane-changing Mode IMOBIL for Car-following Mod-els [J]. Transportation Research Record, 2007 (1999): 86-94.