An Improved Method of Q-Learning Algorithm

Bakun Zhu*, Weigang Zhu¹, Wei Li²

¹ Department of Electronic and Optical Engineering, Space Engineering University, City, Beijing 101416, China
² Graduate School, Space Engineering University, Beijing 101416, China

*Corresponding author’s e-mail: 2474519020@qq.com

Abstract. Q-learning algorithm is a basic algorithm in reinforcement Learning, which has a wide range of applications in many discrete control and planning problems. However, the convergence speed of Q-learning algorithm is not very fast, which brings great difficulties to the solution of practical problems. This paper proposes a method to suppress the cowardly behaviour of Q-learning in the training process, which can greatly improve the convergence speed of the algorithm and is of great significance to the research of Q-learning algorithm.

1. Introduction
Reinforcement learning, a hot area of machine learning research, is different from supervised learning methods such as deep learning because it does not require a large number of samples and labels. Due to such characteristics, reinforcement learning is often used to solve control or planning decision problems, such as games[1], spacecraft control[2], financial transactions[3], energy management[4]. However, reinforcement learning also has drawbacks, that is, reinforcement learning agents need a lot of interaction with the environment to learn the final strategy, which is unacceptable for many problems with high real-time requirements. Therefore, it is of great significance to improve the existing reinforcement learning algorithm and improve the convergence speed.

It is found that Q-learning algorithm is prone to timid negative behaviour in the training process, and the existence of this negative behaviour will reduce the convergence efficiency. If this negative behaviour can be overcome, it is expected to improve the convergence speed of reinforcement Learning algorithm. The main work of this paper is to improve the hand convergence speed of the algorithm by suppressing the timid negative lines.

In the rest of the paper, the second section introduces the basic principles of reinforcement learning and Q-learning algorithm, and analyses the negative behaviours of reinforcement learning agents in the training process. In the third section, the methods of inhibiting reinforcement learning agents are proposed. In the fourth section, the simulation experiment is carried out to prove the effectiveness of the proposed method. The fifth section summarizes the whole paper and looks forward to the next step.

2. Theoretical Basis
The main content of this section includes the basic theory of reinforcement Learning, Q-learning algorithm, and the influence of income setting on the behaviour of agents, which are the basis for the algorithm improvement in the next step.
2.1. Reinforcement learning theory

Reinforcement learning is a machine learning method to study the optimal behaviour strategies of the agent through interaction with the environment. The interaction process between the agent and the environment is Markov Decision Process (MDP). The agent can make corresponding actions to adapt to the environment by observing the data sequence obtained by interaction with the environment. Its essence is to learn the optimal sequential decision. Figure 1 shows the process of "interactive" learning in reinforcement learning in the form of Markov decision process. In each "interaction", an action $A_t$ generated by intelligence acts on the environment, and the state of the environment is transferred from $S_t$ to $S_{t+1}$ under the action. The agent observes the change of environmental state and obtains a benefit $R_t$ from it, and then adjusts the decision model according to the benefit. After a large number of "interactions" with the environment, the agent gradually learns how to take actions to maximize the cumulative benefits, which can be expressed as:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots$$

where $\gamma \in [0,1]$ is the discount rate, which represents the present value of future earnings at the present time.

Strategy, the method by which an agent selects an action, is a mapping from state to action: $\pi : S \rightarrow A$. The goal of reinforcement learning is to learn an optimal strategy to maximize the cumulative benefits of the agent.

![Figure 1 The MDP of reinforcement learning](image)

2.2. Q-Learning algorithm

In the Q-Learning algorithm, the cumulative revenue expectation obtained by selecting actions according to the strategy under the state is called the state-action value, denoted as $Q$. Is the basis for the selection of actions. Generally, methods are used to select actions, such as the formula below, to select the actions that reach the maximum with the probability of, and to randomly select the actions with the probability of $\epsilon$, which is called the exploration factor, to balance the learning and exploration in Q-Learning algorithm.

The best strategy satisfies Behrman's optimal equation:

$$Q_\pi(s, a) = \sum_{r(s, a)} p(s', r | s, a) (r(s, a) + \gamma \cdot \max_a Q_\pi(s', a))$$

Where, $s'$, $a'$ denote the state and action at the next moment. Dynamic environment characteristics are represented by $p(s', r | s, a)$, and revenue is represented as a function of state $s$ and action $a$, namely $r(s, a)$. In Q-learning, the difference method is adopted to update the Q value, and the optimal strategy can be obtained by iterating continuously through the following formula until the value of $Q_\pi(s, a)$ converges.

$$Q(s, a) \leftarrow Q(s, a) + \alpha [r(s, a) + \gamma \cdot \max_a Q(s', a') - Q(s, a)]$$

Where $\alpha$ is the learning rate.
2.3. Effect of payoff setting on agent behavior

The setting of payoff directly reflects the goal of reinforcement learning, which directly determines the quality of reinforcement learning tasks. The reinforcement learning agent is very sensitive to the setting of payoff. The unreasonable setting of payoff will bring about a series of negative behaviours, which mainly include rashness, greed and timidity. The study of this negative behaviour is helpful for us to better study the setting of the return function.

Rashness, no punishment or punishment for an undesirable event is too small, so that the agent cannot learn to actively avoid the event, or still choose to accept the punishment in exchange for greater benefits after weighing the pros and cons.

Greedy, in the process of income function setting, tend to set some bonus to better guide the agent complete goal task, these additional rewards on the one hand, will be better in smart goals and tasks to complete, on the other hand may be left agent in pursuit of infinite loop bonus, lead to the failure of the target task.

Timidity is another negative behaviour just opposite to greed. Due to the excessively large penalty in the income setting, the agent will receive a large amount of negative feedback in the process of exploration, which makes the agent shrink down and dare not explore.

The above negative behaviours are the issues that need to be considered in the setting of the payoff function. The goal of setting the payoff function is to avoid negative behaviours while accelerating the convergence of the algorithm.

3. The suppression of timid behaviour

Timid behaviour is caused by the fact that the agent can gain more by staying in the original state, while leaving the current state will cause the agent to suffer losses. As a result, the agent does not dare to explore and prefers to stay in the same place. In order to restrain the agent's behaviour, it can be punished when the agent chooses to stay in the original state. The more times an agent continues to stay in the same state, the greater the penalty. \( r \) is used to represent the return function of reinforcement learning problem.

\[
T(s, s') = r(s, s') + T(s, s')
\]

(4)

\( T(s, s') \) is the added correction function to suppress timid behaviour, and the expression of \( T(s, s') \) is as follows:

\[
T(s, s') = (C(s') - 1) \omega_g \times (-1)
\]

(5)

\( C(s') \) represents the number of state continuations \( s' \), and \( \omega_g \) represents the income inhibition coefficient, which is used to control the degree of income acquisition of decision agents that suppress cognitive interference.

4. Simulation experiment and analysis

To verify the method proposed in this paper, it is assumed that the common environment state of MDP problem is 20, represented by \( \{s, s_2, \ldots, s_20\} \), and the number of actions in the action set is 10, which are \( \{a_1, a_2, \ldots, a_{10}\} \). The transition relationship between states is shown in the Figure 2:
The task of reinforcement learning intelligence is to transfer the environmental state from $s_1$ to $s_{10}$. The basic parameter Settings are shown in Table 1:

| Parameters | Value | Name          |
|------------|-------|---------------|
| $\alpha$  | 0.01  | Learning rate |
| $\gamma$  | 0.95  | Discount factor|
| $\epsilon$| 0.1   | Explore rate  |

Under the same environment and basic parameter Settings, Monte Carlo experiments were conducted for 20 times. The experimental results are shown in the figure below:

In Figure 3, the yellow curve drops significantly faster than the blue curve. Compared with the traditional Q-learning algorithm, the proposed method can greatly improve the convergence rate of the algorithm.

5. Conclusions
In this paper, based on the original Q-learning algorithm, a correction term of timid behaviour is added to the income function. Simulation results show that the proposed method is effective. This method is
simple, easy to implement, has good expansibility, and has important theoretical value and application value. This method can be applied to the interference decision problem of reinforcement learning to improve the convergence speed of the algorithm. In this paper, Q-learning algorithm is improved from the perspective of revenue. Of course, algorithms can also be improved in combination with other perspectives, which is worth further study in the future.

References
[1] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, et al. Mastering the game of go without human knowledge. 2017;550(7676):354-9.
[2] Yoo J, Jang D, Kim HJ, Johansson KHJICSL. Hybrid reinforcement learning control for a micro quadrotor flight. 2020;5(2):505-10.
[3] Park H, Sim MK, Choi DGJESwa . An intelligent financial portfolio trading strategy using deep Q-learning. 2020;158:113573. Thompson, J.N. (1984) Insect Diversity and the Trophic Structure of Communities. In: Ecological Entomology. New York. pp. 165-178.
[4] Wu J, Wei Z, Li W, Wang Y, Li Y, Sauer DUJIToII. Battery thermal-and health-constrained energy management for hybrid electric bus based on soft actor-critic DRL algorithm. 2020;17(6):3751-61.
[5] Watkins CJ, Dayan PJMI. Q-learning. 1992;8(3-4):279-92.