Journal of Systems Engineering and Electronics
Vol. 33, No. 2, April 2022, pp.381 – 392

Choice of discount rate in reinforcement learning with long-delay rewards

LIN Xiangyang*, XING Qinghua, and LIU Fuxian

Department of Air Defense and Anti-Missile, Air Force Engineering University, Xi'an 710051, China

Abstract: In the world, most of the successes are results of long-term efforts. The reward of success is extremely high, but before that, a long-term investment process is required. People who are "myopic" only value short-term rewards and are unwilling to make early-stage investments, so they hardly get the ultimate success and the corresponding high rewards. Similarly, for a reinforcement learning (RL) model with long-delay rewards, the discount rate determines the strength of agent's "farsightedness". In order to enable the trained agent to make a chain of correct choices and succeed finally, the feasible region of the discount rate is obtained through mathematical derivation in this paper firstly. It satisfies the "farsightedness" requirement of agent. Afterwards, in order to avoid the complicated problem of solving implicit equations in the process of choosing feasible solutions, a simple method is explored and verified by theoreti cal demonstration and mathematical experiments. Then, a series of RL experiments are designed and implemented to verify the validity of theory. Finally, the model is extended from the finite process to the infinite process. The validity of the extended model is verified by theories and experiments. The whole research not only reveals the significance of the discount rate, but also provides a theoretical basis as well as a practical method for the choice of discount rate in future researches.

Keywords: reinforcement learning (RL), discount rate, long-delay reward, Q-learning, treasure-detecting model, feasible solution.

DOI: 10.23919/JSEE.2022.000040

1. Introduction

The essence of reinforcement learning (RL) is actually an optimization of strategies and algorithms. The research history of the RL basic mathematical model can be traced back to the proposal of Bellman equation in the 1950s [1]. The development of animal learning psychology in the 1980s inspired a renaissance in RL research. The rapid update of computer hardware in recent years has brought about a new wave of artificial intelligence. With the powerful function fitting capabilities of deep neural networks, RL research has entered a new stage, attracting attentions of countless researchers.

In practical application, the famous production is the AlphaGo [2] designed by DeepMind company, whose outstanding performance in the game of Go makes RL one of the most popular topics in this new era. Today, the research of RL is not limited in the original game field [3–6], but has expanded to various aspects of life and production. For example, intelligent life involves the control of smart cars, unmanned aerial vehicles, and various robots [7–10], the network planning of smart city [11], as well as the indoor temperature control and heating energy optimization of buildings [12]. In industrial production, it is used to optimize process routes in chemical manufacturing [13]. In medical treatment, it is used for the motion control of surgical robots [14]. Even in the finance field, it can be used to optimize financial portfolio trading strategies intelligently [15]. Besides, RL is also applied to solve classic optimization problems, such as the traditional multiple traveling salesman problem (MTSP) [16].

In the RL model, the value of the discount rate can decide learning results. An inappropriate value of the discount rate may cause that agent would only focus on the short-term low rewards, rather the highest reward in the longer future. The research on the discount rate began withAinslie'spigeonexperimenttostudythetheoryofdelayed rewards [17]. Later, many researchers conducted various experiments from perspectives of biology and medicine [18–21]. In terms of theoretical research, Papale et al. studied the interactions between deliberation and delay-discounting [22]; Yamaguchi et al. researched the "discounted problem" on the learning preferences of animals [23]; Knox et al. studied the problem of temporal discounting from the perspective of human reward [24].

In the past researches on RL, most of them just simply

Manuscript received December 17, 2020.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (71771216; 71701209; 72001214).

pointed out the general definition of the discount rate, but there are few specific analysis on the effect of the discount rate on RL. As a result, 0.9 or a higher random value is usually chosen by default in most of experiments [25–28], but a complete set of theory and an effective choice method are not put forward. On the basis of previous studies, firstly the influence of the discount rate on the RL model is studied from the perspective of mathematical theory in this paper, and the feasible region of the discount rate is deduced, which enables an agent to obtain the highest reward in a long-delay rewards model. Then through theoretical derivation, a simple solution method for the feasible solution of the discount rate which ensures the "farsightedness" ability of agent under normal conditions is explored. Finally, a series of experiments are used to verify the validity of theories and the usefulness of the method.

2. Importance of discount rate

RL is a process in which an agent continuously interacts with environment to learn how to map current state to an action to maximize the gains. It includes two objects: agent and environment. The basic model is shown in Fig. 1.

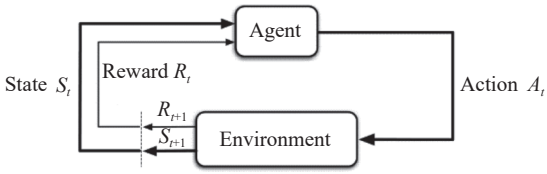


Fig. 1 Basic model of RL

RL is a semi-supervised learning method, that is, during the learning process, the agent is not told which action should be taken, but it just needs to try constantly different actions by itself and evaluate the value of different states and actions according to different rewards. The rewards used for value evaluating generally refer to not only the immediate reward of the current action, but also the delayed rewards of the next state and all subsequent states.

Therefore, these two characteristics, trial-and-error search and delayed rewards, are the two most important distinguishing features of RL [29]. Among various RL models, there is a special one. That is the long-delay rewards model: the agent can choose between low rewards in near term and high reward in the future. If it chooses the former, it will get only low rewards from beginning to end. Otherwise, if it chooses the ultimate high reward, it must endure long period of negative rewards firstly. This model corresponds to the long-term investment process in real life. The most typical examples are the education process and the scientific research process. Only after a long period of investment can the ultimate reward be extremely high. The goal of this RL model is to make the expected gain G_t maximal, which is a function of the reward sequence $\{R_{t+k+1}\}$. Generally G_t is defined as

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{1}$$

where discount rate γ is a parameter $(0 < \gamma < 1)$, determining the effect of delayed rewards on the whole expected gain. When γ approaches 0, the agent is "myopic". When the value of γ is set as the limit $\gamma = 0$, the agent only cares about the maximization of the current immediate reward, and the goal of learning is to choose an appropriate action to maximize the value of R_{t+1}. With γ increasing, the agent gradually becomes more and more "foresighted" and pays the more attentions to future rewards.

There are two kinds of concepts of the value in RL, namely the state value and the action value.

The state value is described by state value function $V_\pi(s)$, which denotes the value of state s of the agent when following policy π.

$$V_\pi(s) = \mathrm{E}_\pi[G_t|S_t = s] = \mathrm{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots|S_t = s] \tag{2}$$

where $\mathrm{E}_\pi[\cdot]$ denotes the expected value of a random variable given that the agent follows policy π, and t is any time step. S_t denotes the state representation at time t.

The action value is described by the state action value function $Q_\pi(s,a)$, which denotes the value of action a on current state s when following police π.

$$Q_\pi(s,a) = \mathrm{E}_\pi[G_t|S_t = s, A_t = a] = \mathrm{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots|S_t = s, A_t = a] \tag{3}$$

where A_t denotes the action representation at time t.

The value of γ affects discounted future rewards, which further affects the value function. Therefore, for some long-delay rewards models, choice of γ is extremely important. For example, in a realistic scenario, only after years of study can a child have stronger abilities than others, and then get a higher reward in work after graduation. In this process, the immediate reward is negative because of the boring learning, but the final reward is extremely high; on the contrary, if the child chooses to play or rest, the immediate reward may be a little higher, but the final reward will be very low. Similarly, when γ is high enough, the agent will consider future reward more and choose the "boring learning"; on the contrary, the agent will not be "farsighted" enough and choose to "play" or "rest" instead.

The above scenario can be converted into a commonly used model of RL: the treasure-detecting model.

As shown in Fig. 2, this model belongs to the Markov decision process (MDP) in discrete state spaces. The

initial position of the agent is the state (0,0) in red in the upper left corner. The agent can only choose right and down actions, and only move one grid per step. It is defined that when the agent performs a rightward action, it denotes investment to find the treasure. In this case, immediate reward of action is a small negative value $R_{\min}$. It is defined that when the agent performs a downward action, it denotes a rest. In this case, immediate reward of action is a zero value R_{zero}. Only when the agent chooses to keep investing in finding treasure, that is, perform rightward actions all the way and follow the green path to the end state (5,0) in yellow in the upper right corner, the treasure could be obtained. Otherwise, the agent will reach the end states (0,5),(1,4),$\cdots$,(4,1) in black , and in this case no treasure could be obtained. It is defined that the immediate reward of treasure is a great positive value $R_{\max}$, and the immediate reward is a zero value R_{zero} in other end states.

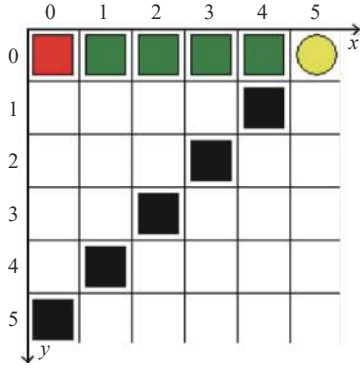


Fig. 2 Treasure-detecting model with long-delay rewards

In this model, in order to get the treasure reward, the agent must firstly endure multiple consecutive negative rewards $R_{\min}$ before. If γ is too small, the discounted value of treasure rewards will become too small to offset the sum of the discounted value of the previous negative rewards. Finally the action value of looking for treasure will be less than the value of rest. The external manifestation is that the agent is not "farsighted" enough: no matter how many episodes of training, it cannot reach the treasure state, but will eventually prefer to the state (5,0). That is, the downward action which denotes rest will be chosen from the beginning to the end. Obviously the choice of γ has a vital influence on results of the agent's training. Therefore, the specific feasible region of γ would be discussed in detail in the next chapter.

3. Feasible region of discount rate

In the long-delay rewards model shown in Fig. 2, in order to obtain the treasure reward, the discount rate has different feasible regions for different model parameters, which include the maximum steps number in each episode, the immediate reward of treasure, etc.

The model corresponding to Fig. 2 is further abstracted: the state space is expanded from 6×6 to $(n+2)\times(n+2)$. The initial position of the agent is still at (0,0). The treasure position is set to (n+1,0). The agent can only choose right and down actions, and only move one grid per step. The action of rightward movement is defined as $A_t=0$; the other is defined as $A_t=1$. The steps number of the agent in each episode is determined as $T=(n+1)$, that is, when the agent performs $T=(n+1)$ actions and moves to a certain point among $(0,n+1)$, $(1,n)$, $(2,n-1)$, $\cdots$,$(n+1,0)$, the current episode ends. During the movement, if the next state is not the end state, the immediate reward of action of rightward movement will be $R_{t,A=0}$, and the reward of action of downward movement will be $R_{t,A=1}$. If the next state is the end state, then if the end state is the treasure state $(n+1,0)$, the immediate reward will be $R_{t=T}=R_{TM}$, otherwise if the end state is not the treasure state including $(0,n+1)$, $(1,n)$, $(2,n-1)$, $\cdots$, $(n,1)$, the immediate reward will be $R_T=R_{T0}$.

The value of the reward are defined as

$$\begin{cases} R_{t,A=0}=R_{\min}/|R_{\min}|=-1 \\ R_{t,A=1}=R_{\text{zero}}/|R_{\min}|=0 \\ R_{TM}=R_{\max}/|R_{\min}|=R_{\max} \\ R_{T0}=R_{t,A=1}=0 \end{cases}. \tag{4}$$

According to (1), in this model, the gain function of the agent in an episode of training is

$$G_{t=0}=R_1+\gamma R_2+\gamma^2R_3+\cdots+\gamma^nR_{n+1}= \sum_{k=0}^{n}\gamma^kR_{k+1},\ 0<\gamma<1;n\in\mathbf{N}^+;n\gg1. \tag{5}$$

If the agent can get treasure, all its actions must be rightward movement. Thus R_k and A_k should meet the following conditions:

$$\begin{cases} R_k=-1,\ 1\leqslant k\leqslant n \\ R_k=R_{\max},\ k=n+1 \\ A_{k-1}=0,\ 1\leqslant k\leqslant n+1 \end{cases}. \tag{6}$$

Substituting (6) into (5), we get

$$G_{t=0}(A_{k-1}=0,1\leqslant k\leqslant n+1)= -(1+\gamma+\gamma^2+\cdots+\gamma^{n-1})+\gamma^nR_{\max}=\gamma^nR_{\max}-\frac{1-\gamma^n}{1-\gamma}. \tag{7}$$

When the agent chooses other paths, compared to (6), R_k and A_k would meet the following conditions:

$$\begin{cases} R_k=-1,\ A_{k-1}=0;\ 1\leqslant k\leqslant n \\ R_k=0,\ A_{k-1}=1;\ 1\leqslant k\leqslant n \\ R_k=0,\ k=n+1 \\ \sum\limits_k A_{k-1}\neq0,\ 1\leqslant k\leqslant n+1 \end{cases}. \tag{8}$$

Substituting (8) into (5), we get

$$G_{t=0}\left(\sum_k A_{k-1}\neq 0,\ 1\leqslant k\leqslant n+1\right)=n+1-\sum_k A_{k-1}. \quad (9)$$

Obviously as for (9), when $G_{t=0}$ is the maximum, the agent should choose downward actions all, that is,

$$A_{k-1}=1,\ \ 1\leqslant k\leqslant n+1. \quad (10)$$

Substituting (10) and (9) into (5), at this condition we get the maximum gain

$$G_{t=0}(A_{k-1}=1,1\leqslant k\leqslant n+1)=0\times(1+\gamma+\gamma^2+\cdots+\gamma^n)=0. \quad (11)$$

Our goal is to find the feasible γ, which guarantees that the agent can learn a suitable action value function $Q_\pi(s,a)$ after enough episodes of training. This function is referred to simply as the Q-Table. When the agent executes a purely greedy policy based on the data in the Q-Table, it can choose rightward action all the way and finally get treasure reward $R_{\max}$. In order to achieve this goal, the following conditions need to be met:

$$R_{\max}\gg n, \quad (12)$$

$$G_{t=0}(A_{k-1}=0,1\leqslant k\leqslant n+1)>G_{t=0}(A_{k-1}=1,1\leqslant k\leqslant n+1). \quad (13)$$

Substituting (7) and (11) into (13), we get

$$\gamma^n R_{\max}-\frac{1-\gamma^n}{1-\gamma}>0 \quad (14)$$

$\Rightarrow$

$$Q(S=[0,0],A=0)=\mathrm{E}_\pi[R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots|S_t=s]=-1(1+0.6+0.6^2+0.6^3)+20\times0.6^4=0.416$$

Define

$$F(\gamma)=\frac{\gamma^n(1-\gamma)}{(1-\gamma^n)}-\frac{1}{R_{\max}}. \quad (15)$$

Take the derivative of $F(\gamma)$ and we get

$$F'(\gamma)=\frac{\gamma^{n-1}\left[\gamma^{n+1}-(1+n)\gamma+n\right]}{(1-\gamma^n)^2}. \quad (16)$$

Define

$$J(\gamma)=\gamma^{n+1}-(1+n)\gamma+n. \quad (17)$$

Taking the derivative of $J(\gamma)$, we get

$$J'(\gamma)=(1+n)\gamma^n-(1+n). \quad (18)$$

With $0<\gamma<1$, we get

$$0<\gamma^n<1, \quad (19)$$

$$J'(\gamma)<0. \quad (20)$$

Thus $J(\gamma)$ is decreasing. Because $\lim\limits_{\gamma\to1}J(\gamma)=0$, we get

$$J(\gamma)>0, \quad (21)$$

$$F'(\gamma)>0. \quad (22)$$

Thus $F(\gamma)$ is increasing. Then we can get

$$\lim_{\gamma\to0}F(\gamma)=-\frac{1}{R_{\max}}<0, \quad (23)$$

$$\lim_{\gamma\to1}F(\gamma)=\frac{1}{n}-\frac{1}{R_{\max}}>0. \quad (24)$$

Because $F(\gamma)$ is monotonically increasing ($0<\gamma<1$), according to the "Zero Theorem", there must be a solution $\gamma_0\,(0<\gamma_0<1)$, which meets the requirement of the equation:

$$F(\gamma_0)=0. \quad (25)$$

Thus the feasible region of γ meeting (13) is

$$\gamma_0<\gamma<1. \quad (26)$$

In (26), γ_0 is the solution of (25).

4. A feasible solution of discount rate

In Section 3, the feasible region of discount rate γ is derived, which enables the agent so "farsighted" to obtain treasure. The key point is the solution of γ_0. Since (25) is implicit, it is difficult to solve it. This chapter will introduce a simpler method of choosing a feasible solution.

According to $0<\gamma<1, n\in\mathbf{N}^*, n\gg1$, we get

$$\gamma^n\to0^+. \quad (27)$$

Define

$$f(\gamma)=\gamma^n(1-\gamma)-\frac{1}{R_{\max}}. \quad (28)$$

Therefore,

$$f(\gamma)<F(\gamma). \quad (29)$$

Take the derivative of $f(\gamma)$ and we get

$$f'(\gamma)=n\gamma^{n-1}-(n-1)\gamma^n. \quad (30)$$

Therefore,

$$\begin{cases} f'(\gamma)>0,\ \ 0<\gamma<\gamma'_0\\ f'(\gamma)=0,\ \ \gamma=\gamma'_0\\ f'(\gamma)<0,\ \ \gamma'_0<\gamma<1 \end{cases}. \quad (31)$$

In (31), $\gamma'_0=n/(n+1)$. Therefore, $f(\gamma)$ increases in range $(0,\gamma'_0)$, decreasing in range $(\gamma'_0,1)$, and gets the maximum value $f(\gamma'_0)$ at the point of $\gamma=\gamma'_0$.

According to (29), when $f(\gamma'_0)\geqslant0$, there must be $F(\gamma'_0)>0$.

Substituting $\gamma'_0=n/(n+1)$ into (28), we get

$$f(\gamma'_0,n)=\left(\frac{n}{n+1}\right)^n\left(1-\frac{n}{n+1}\right)-\frac{1}{R_{\max}}. \quad (32)$$

Making $f(\gamma'_0,n)\geqslant0$, we get

$$R_{\max} \geqslant \left(1+\frac{1}{n}\right)^n (n+1). \tag{33}$$

because

$$\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n = \mathrm{e}^{-}, \tag{34}$$

$$R_{\max} \geqslant (n+1)\mathrm{e}. \tag{35}$$

As explained above, when $R_{\max}$ and n satisfy (35), there must be $F(\gamma_0') > f(\gamma_0') > 0$, where e is the natural constant, and $\gamma_0' = n/(n+1)$. Therefore, when $R_{\max}$ and n in the model are determined, the first step is to compare whether their value satisfies the constraint relationship of (35). If yes, it is the simplest to choose $\gamma = n/(n+1)$ directly. It can ensure that after innumerable episodes of training, the agent can learn a suitable Q-Table, and choose a path of "bitter before sweet" to get treasure reward. If no, the easy method above is not applicable. In this case the only method to get a feasible solution of γ is to solve implicit (25). Any value in the feasible region can be chosen.

Then setting $R_{\max}$ as well as n multiple different values, we get the corresponding change trends of $F(\gamma)$ and $f(\gamma)$ with respect to γ shown in Fig. 3.

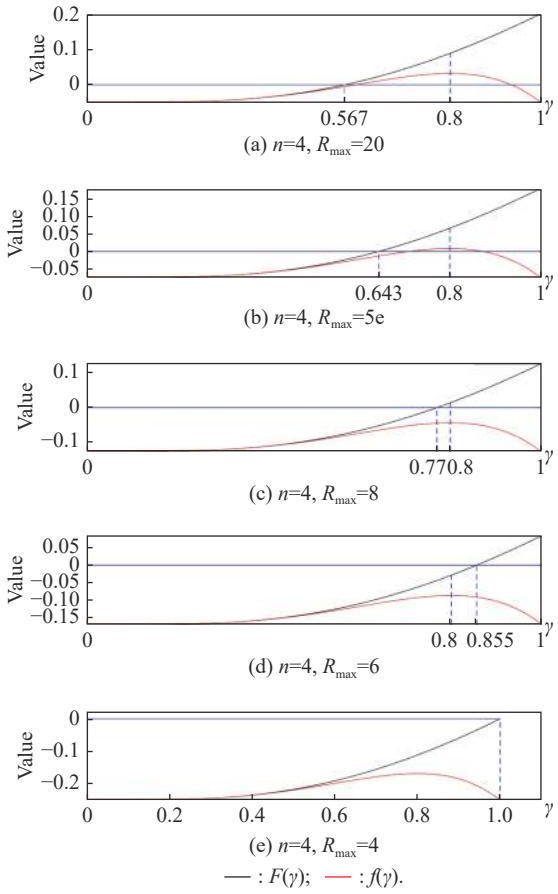


Fig. 3 $F(\gamma)$ and $f(\gamma)$ with respect to γ under different $R_{\max}$ and n

As shown in Fig. 3(a), when $R_{\max} = 20 \geqslant 5\mathrm{e} = (n+1)\mathrm{e}$, $F(\gamma)$ and $f(\gamma)$ both have a positive value at the point of $\gamma_0' = n/(n+1) = 0.8$. In Fig. 3(b), when $R_{\max} = 5\mathrm{e} = (n+1)\mathrm{e}$, $F(\gamma)$ and $f(\gamma)$ both have a positive value at the point of $\gamma_0' = 0.8$, too. What is different is that $f(\gamma)$ is slightly greater than zero, while $F(\gamma)$ is extremely greater.

Compare Fig. 3(c) and Fig. 3(d), which correspond to the conditions $R_{\max} = 8 < 5\mathrm{e}$ and $R_{\max} = 6 < 5\mathrm{e}$ respectively. They do not satisfy the constraint relationship of 35, so both of their $f(\gamma)$ are negative values at the point of $\gamma_0' = 0.8$. The difference is $F(\gamma)$. The former is positive but the latter is negative. That means when $R_{\max}$ and n do not satisfy the constraint relationship of (35), $n/(n+1)$ is not necessarily in the feasible region of γ and cannot be used directly. However, both of them satisfy $R_{\max} > n$, so they both have a solution as for the implicit (24). Every value in feasible region $(\gamma_0, 1)$ can be a feasible solution, as the interval $(0.77, 1)$ in Fig. 3(c) and the interval $(0.855, 1)$ in Fig. 3(d).

Fig. 3(e) is an extreme situation: $R_{\max} = n$. As is shown, $F(\gamma)$ is negative in $(0, 1)$. Only when γ takes the extreme value $\gamma = 1$, there is $F(\gamma) = 0$. It means that when the immediate reward of treasure is too small to offset the sum of the immediate negative rewards, the agent cannot choose the treasure. That is also consistent with the common sense.

5. Experiments and discussions

In previous sections, the choice of γ is researched for different $R_{\max}$ as well as n in the long-delay rewards model. Then the correctness of the theory is verified by a series of experiments in this section.

5.1 Experiments design

Take the model in Fig. 2 as an example. For this model, there is $n = 4$, and state space is a 6×6 array. The value of $R_{\max}$ is set as 20, 5e, 8, and 6 respectively. Train the agent with the off-policy temporal-difference learning method, that is, the Q-learning method. The specific algorithm of Q-learning is shown in Algorithm 1.

Algorithm 1: Q-learning (off-policy temporal-difference (TD) control)

Algorithm parameters: step size $\alpha \in (0, 1]$ small $\varepsilon > 0$
Initialize $Q(s, a)$, for all $s \in S^+$, $a \in A(s)$, arbitrarily except that $Q(\text{terminal}, \cdot) = 0$
Loop for each episode:
  Initialize S
  Loop for each step of episode:
    Choose A from S using policy derived from Q (e.g., ε – greedy)
    Take action A, observe R, S'
    $Q(S, A) \leftarrow Q(S, A) + \alpha[R + \gamma \max_a Q(S', a) -$

$Q(S, A)]$

$S \leftarrow S'$

until S is terminal

For every value of $R_{\max}$, the value of γ is set as 0, 0.1 , 0.2, $\cdots$, 1. For every specific combination of $R_{\max}$ and γ, the agent needs to be trained for 20 000 episodes respectively. The action policy during training is the $\varepsilon-$greedy policy. In order to enhance the agent's exploration ability, the value of ε is set as 0.5. In order to ensure stability of the training process, the value of learning step length α is set as 0.01. After the training process, test agent using the pure greedy policy in the same model, and we can get Q-Table and the optimal path as for every specific combination of $R_{\max}$ and γ.

5.2 Results and discussions

After training and tests, we get the optimal path using the pure greedy policy as for every specific combination of $R_{\max}$ and γ. The optimal paths are recorded in Table 1. As shown in Table 1, at a premise that $R_{\max}$ is fixed as 20, the optimal path of the agent changes from continuous downward to continuous rightward when γ increases to 0.7. However, the above theory points out that the agent should change the path when γ is greater than γ_0. That is, according to zero point of $F(\gamma)$ in Fig. 3(a), the solution of equation 25 is 0.576, therefore, the optimal path of the agent should change when γ increases to 0.6 rather than 0.7 in this model. Similarly, in other groups corresponding to different values of $R_{\max}$, there are also parts that are inconsistent with theory. Such as $R_{\max}$ = 5e, according to Fig. 3(b), theoretically the optimal path of the agent should change when γ increases to 0.7, but the actual result is 0.8. Under the conditions of $R_{\max}$ = 8 and $R_{\max}$ = 6, all the optimal path of the agent are continuous downward, which is obviously inconsistent with theory.

Table 1 Optimal paths of agent for different combinations of $R_{\max}$ and γ

γ	$R_{\max}$			
	20	5e	8	6
0.0	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.1	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.2	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.3	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.4	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.5	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.6	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.7	{0,0,0,0,0}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.8	{0,0,0,0,0}	{0,0,0,0,0}	{1,1,1,1,1}	{1,1,1,1,1}
0.9	{0,0,0,0,0}	{0,0,0,0,0}	{1,1,1,1,1}	{1,1,1,1,1}
1.0	{0,0,0,0,0}	{0,0,0,0,0}	{1,1,1,1,1}	{1,1,1,1,1}

In response to discrepancy between theory and experiment under the condition of $R_{\max}$ = 20 and γ = 0.6, the corresponding Q-Table is picked up and shown in Table 2.

Table 2 Q-Table under the condition of $R_{\max}$= 20 and γ= 0.6

State	Action	
	0	1
[0,0]	−1	0
[0,1]	−1	0
[0,2]	−1	0
[0,3]	−1	0
[0,4]	−1	0
[1,0]	−0.742 49	0
[1,1]	−1	0
[1,2]	−1	0
[1,3]	−1	0
[2,0]	1.108 516	0
[2,1]	−0.999 27	0
[2,2]	−0.999 97	0
[3,0]	5.794 161	0
[3,1]	−0.899 89	0
[4,0]	16.190 77	0

According to Table 2, there is $Q(S = [4,0], A = 0) =$ 16.19. In fact, when the state converts to [4,0], the agent gets treasure if it performs action $A = 0$. Thus the theoretical action value should be equal to the immediate reward of treasure $R_{\max} = 20$.

It is inferred that the reason for the error may be that the learning step α is too short, or the number of training episodes is not enough. As a result that training is incomplete, and the error between the data in Q-Table and the theoretical value is too large. Next, increase the number of training episodes to 100 000 and perform a set of supplementary experiments again.

5.3 Supplementary experiments and result discussions

Use the same training model in Subsection 5.1; increase the number of training episodes from 20 000 to 100 000; and carry out the experiment again.

After the supplementary experiments, we get the optimal path using the pure greedy policy as for every specific combination of $R_{\max}$ and γ again. The new optimal paths are recorded in Table 3. By comparing Fig. 3 and Table 3, it can be seen that experiment results at this time are fully consistent with the theory. For $R_{\max}$ = 20, the optimal path of the agent changes when γ increases to 0.6. This corresponds to the point of γ_0 = 0.567 in Fig. 3(a),

which is the solution of (25). Similarly, for $R_{\max}$ = 5e, the optimal path changes when γ increases to 0.7. This corresponds to the point $\gamma_0 = 0.643$ in Fig. 3(b); for $R_{\max} = 8$, the optimal path changes when γ increases to 0.8. This corresponds to the equation's solution $\gamma_0 = 0.77$ in Fig. 3(c); for $R_{\max} = 6$, the optimal path changes when γ increases to 0.9. This corresponds to the equation's solution $\gamma_0 = 0.855$ in Fig. 3(d).

Table 3 Optimal path for different combinations of $R_{\max}$ and γ in supplementary experiments

γ	$R_{\max}$			
	20	5e	8	6
0.0	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.1	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.2	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.3	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.4	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.5	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.6	{0,0,0,0,0}	{1,1,1,1,1}	{1,1,1,1,1}	{1,1,1,1,1}
0.7	{0,0,0,0,0}	{0,0,0,0,0}	{1,1,1,1,1}	{1,1,1,1,1}
0.8	{0,0,0,0,0}	{0,0,0,0,0}	{0,0,0,0,0}	{1,1,1,1,1}
0.9	{0,0,0,0,0}	{0,0,0,0,0}	{0,0,0,0,0}	{0,0,0,0,0}
1.0	{0,0,0,0,0}	{0,0,0,0,0}	{0,0,0,0,0}	{0,0,0,0,0}

We also get the Q-Table under the condition of $R_{\max} = 20$ and $\gamma = 0.6$ in supplementary experiments, which is shown as Table 4.

Table 4 Q-Table under the condition of $R_{\max}$= 20 and γ= 0.6 in supplementary experiments

State	Action	
	0	1
[0,0]	0.416	0
[0,1]	−1	0
[0,2]	−1	0
[0,3]	−1	0
[0,4]	−1	0
[1,0]	2.36	0
[1,1]	−1	0
[1,2]	−1	0
[1,3]	−1	0
[2,0]	5.6	0
[2,1]	−1	0
[2,2]	−1	0
[3,0]	11	0
[3,1]	−1	0
[4,0]	20	0

It can be the seen that the experimental value in Table 4 has converged to the theoretical value.

For example, in the case of $S = [0,0]$, $A = 0$, the theoretical value can be calculated according to (3) as follows:

$$Q(\mathrm{S} = [0,0], A = 0) = \mathrm{E}_{\pi}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + |S_t = s] = -1(1 + 0.6 + 0.6^2 + 0.6^3) + 20 \times 0.6^4 = 0.416.$$

It is consistent with the experimental value in Table 4. Similarly, it can be verified that the other experimental values in Table 4 are also consistent with their theoretical values. It proves that all of experimental values have converged to theoretical values with a certain accuracy at this time.

The supplementary experiment verifies the rightness and feasibility of theory in Section 3 and Section 4. That is, for an RL model with long-delay rewards, when $R_{\max}$ and n are determined, the value of γ will have a qualitative impact on training results. As for arbitrary combination of $R_{\max}$ and γ, we just need to solve implicit equation (25) and get the solution γ_0, then choose the arbitrary value of γ from interval $(\gamma_0, 1)$. The agent can correctly find treasure after enough training. Specially, when the relation between $R_{\max}$ and n satisfies constraint (35), we can choose $n/(n+1)$ as the value of γ directly. That method ensures convergence of results and eliminates the complicated process of solution to implicit equations. Through comparison of experiments in the following section, it is shown that in practice, the completeness of training will also affect the training effect. In order to get real training results, and ensure the error small enough between the experimental value and the theoretical value, it is not only necessary to control relationship between γ, $R_{\max}$ and n, but also have enough training episodes to ensure the completeness of training.

6. Model expansion and comparative experiments

It is pointed out that only when the discount rate γ is greater than a certain threshold, the agent have stronger foresight ability and succeed in obtaining the treasure.

However, the choice of the discount rate is not the bigger the better. The detailed proof is demonstrated in this section.

6.1 From finite to infinite

In the model shown in Fig. 2, the move steps of the agent in each episode are limited in a finite number T due to the existence of the end state. This process is called the finite MDP. In reality, more problems are persistent tasks that run for an infinite long time but have no end. That

process is called the infinite MDP. In an infinite MDP model, $T \to \infty$. That puts forward a new constraint on the choice of discount rate γ in (1).

According to the finite MDP treasure-detecting model in Fig. 2, the state transition graph of the infinite MDP model with long-delay rewards is shown in Fig. 4.

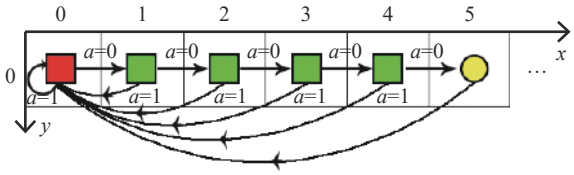


Fig. 4 State transition graph of the infinite MDP model with long-delay rewards

There are six state spaces in the model, namely (0,0), (1,0), (2,0),$\cdots$,(5,0). The initial state is the leftmost state (0,0) in red, and there is no ending state. The action spaces of the agent are still right and down. The right action denotes the effort, in which the corresponding immediate reward is $R_{\max}$ if the next state is treasure state (5,0) in yellow, or $R_{\max}$ if the next state is the other in green. The down action denotes the rest, in which the corresponding immediate reward is R_{zero}.

The state transfer rule is that the agent moves one grid per step. No matter what state the agent is in, as long as it chooses to move down, it will immediately return to the leftmost state (0,0). It can only reach the treasure state (5,0) when it chooses to move right for five consecutive times. This state is the instantaneous transition state, that is, when agent reaches the state (5,0), it will immediately return to the state (0,0) and go on. The whole process starts from the state (0,0) and has no ending.

The model in Fig. 4 is further abstracted: the state spaces are expanded from six to $(n+2)$. The initial state is (0,0). The treasure state is (n+1,0). The action spaces are still right and down. The right action is defined as $A_t = 0$ and the down action is defined as $A_t = 1$. The agent moves one grid per step and its steps number is $T = \infty$. During the moves, if the next action is going down, the next state returns to (0,0), and the immediate reward is $R_{t,A=1}$. Otherwise, when the next action is right, if the next state is not treasure state $(n+1,0)$, the immediate reward is $R_{t,A=0}$; if yes , the immediate reward is R_{TM} and the state will return to (0,0) immediately.

The value of the reward is defined as

$$\begin{cases} R_{t,A=0} = R_{\min}/|R_{\min}| = -1 \\ R_{t,A=1} = R_{\text{zero}}/|R_{\min}| = 0 \\ R_{TM} = R_{\max}/|R_{\min}| = R_{\max} \end{cases}. \tag{36}$$

According to (1), in this model, the gain function in the process of training is

$$G_t = R_{t+1} + \gamma G_{t+1}. \tag{37}$$

If the agent chooses the right actions consecutively to obtain the treasures from $t = 0$, then

$$\begin{cases} R_k = -1, \ k \notin \{n+1, 2(n+1), 3(n+1), \cdots\} \\ R_k = R_{\max}, \ k \in \{n+1, 2(n+1), 3(n+1), \cdots\} \\ A_{k-1} = 0, \ k \geqslant 1 \end{cases} \tag{38}$$

Substituting (38) into (39), we get

$$\begin{cases} G_{t=0} = -1 + \gamma G_{t=1} \\ G_{t=1} = -1 + \gamma G_{t=2} \\ \quad \vdots \\ G_{t=n} = R_{\max} + \gamma G_{t=n+1} \\ G_{t=n+1} = G_{t=0} \end{cases}. \tag{39}$$

Then, we get

$$\begin{aligned} &G_{t=0}(A_{k-1} = 0, k \geqslant 1) = \\ &-(1 + \gamma + \gamma^2 + \cdots + \gamma^{n-1}) + \gamma^n R_{\max} + \gamma^{n+1} G_0 = \\ &\left(\gamma^n R_{\max} - \frac{1-\gamma^n}{1-\gamma}\right) \cdot \left(\frac{1}{1-\gamma^{n+1}}\right). \end{aligned} \tag{40}$$

If the agent chooses other paths, namely it moves to right continuously from $t = 0$ until $t = T(0 \leqslant T \leqslant n)$ turns downward, there are conditions

$$\begin{cases} A_{k-1} = 0, \ 1 \leqslant k \leqslant T \\ R_k = -1, \ 0 \leqslant k \leqslant T \\ A_k = 1, \ k = T \\ R_{k+1} = 0, \ k = T \end{cases}. \tag{41}$$

The gain function during $0 \leqslant t \leqslant T$ is

$$\begin{aligned} &G_{t=0}(A_{k-1}=0, \ A_T = 1, \ 1 \leqslant k \leqslant T) = \\ &-(1 + \gamma + \gamma^2 + \cdots + \gamma^{T-1}) + \gamma^{T+1} G_0 = \\ &-\left(\frac{1-\gamma^T}{1-\gamma}\right) \cdot \left(\frac{1}{1-\gamma^{T+1}}\right). \end{aligned} \tag{42}$$

As $0 < \gamma < 1$, the inequalities $1/(1-\gamma^{T+1}) > 0$, $1/(1-\gamma) > 0$ hold.

And only when the parameter $T = 0$, can G_0 get the maximum value

$$G_{t=0}(A_{k-1} = 1, k \geqslant 1) = 0. \tag{43}$$

Similarly to the finite process in Fig. 2, in an infinite process, if the agent can continuously obtain treasure

reward, (12) and the following equation should also be satisfied:

$$G_{t=0}(A_{k-1}=0,\ k\geqslant 1) > G_{t=0}(A_{k-1}=1,\ k\geqslant 1). \tag{44}$$

From (44), (13) can be deduced. And in the end, the same conclusions can be obtained between the finite process and the infinite process. It means that the theories as well as the method, which is conducted and verified above and is appropriate for the finite process, are also appropriate for the infinite process.

6.2 Choice of discount rate in infinite process

In a finite MDP, the convergence speed of the action value function is generally accelerated with the increase of the discount rate, because the length of the learning process is determined in each episode.

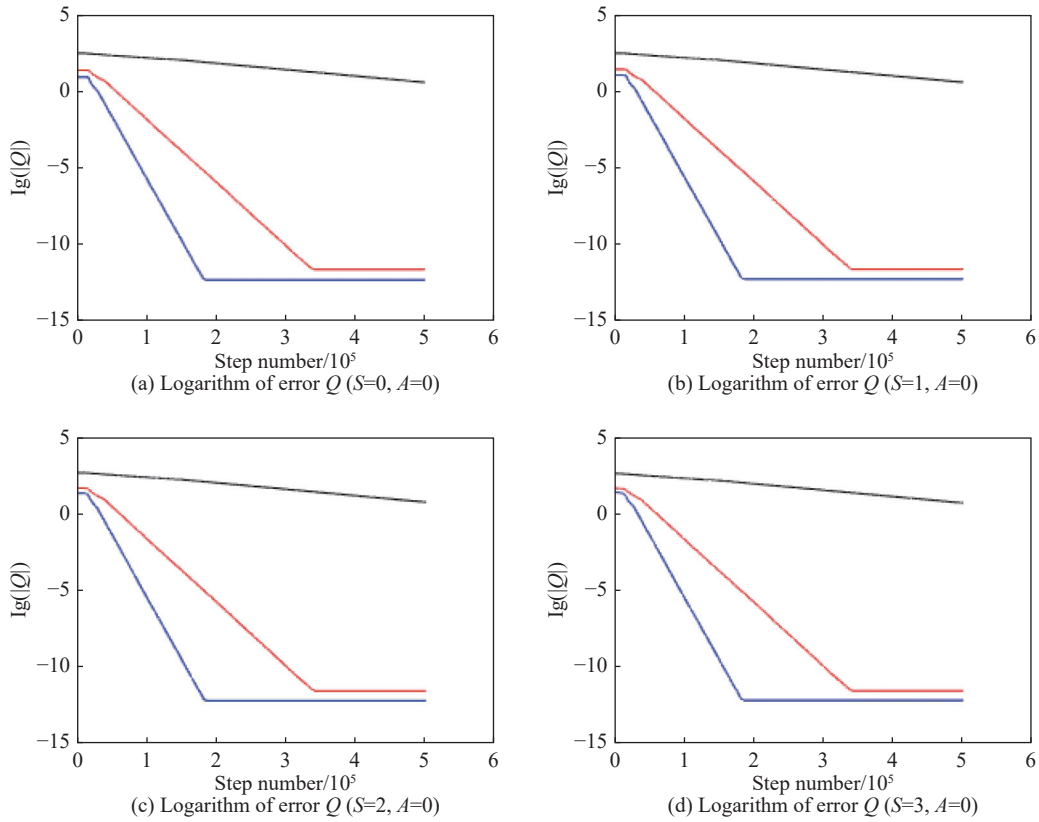
In an infinite process, because the length of the learning is $T\to\infty$, the gain function approaches infinity. It is easy to make each state influence each other greatly in the learning process at the same time. These make the convergence speed of the action value function slowdown, and even make the value function converge to a wrong value.

For the above theory, the experiments are designed comparing with the ones in Section 5.

Choose the model in Fig. 4 as an example. For this model, $n=4$, and there are six state spaces, namely (0,0), (1,0), (2,0) ,$\cdots$,(5,0). Fix the parameter $R_{\max}=20$. The value of discount rate γ is the only variable. According to the conclusion in Section 4, $\gamma_1=n/(n+1)=0.8$ is set as the experimental group. In general researchers' experiments, the discount rate γ is usually chosen as $\gamma_2=0.9$ by default [25,26], so it is set as a control group. In order to enhance the credibility of the theory and the persuasiveness of the comparative experiment, the extreme control group $\gamma_3=0.99$ is specially added. Substitute the three different values of γ into (39) respectively, and the true values of Q-Tables in different γ conditions are calculated.

Then, the agent is trained with Q-learning in different γ conditions. The length of training T is set as 500 000. The action policy during training is the $\varepsilon-$greedy policy. The value of ε is set as 0.5. The value of learning step length α is set as 0.01.

After the training process, the errors between the true values and experimental values of Q-Tables in different γ conditions are obtained. Their variation trends with the training process are shown in Fig. 5.



(a) Logarithm of error Q (S=0, A=0)

(b) Logarithm of error Q (S=1, A=0)

(c) Logarithm of error Q (S=2, A=0)

(d) Logarithm of error Q (S=3, A=0)

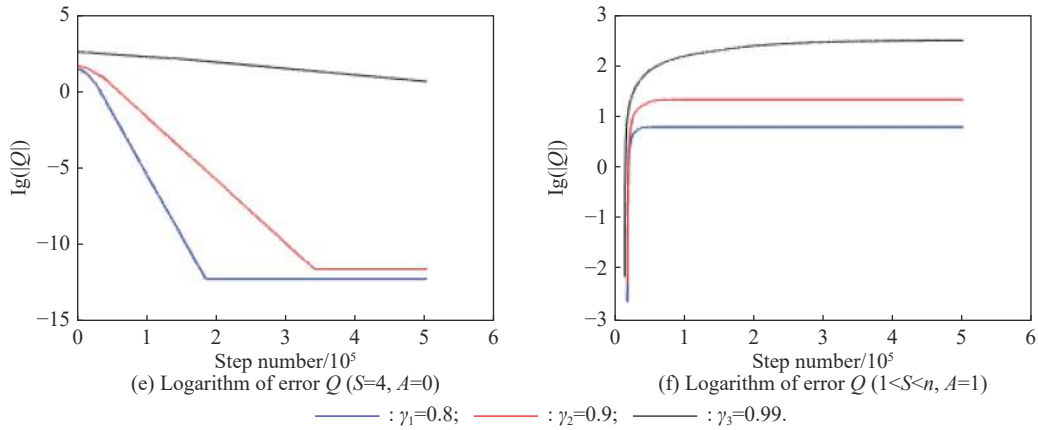


Fig. 5 Variation trends of the error between true and experimental action values with the training process in different γ conditions

Comparing Fig. 5(a)–Fig. 5(e), it can be concluded that when the agent chooses action $A = 0$, the variations of the errors between the true action values and experimental action values are roughly in three stages. Firstly, there is a short stage of large error. After about the 10th to the 20th thousand steps, the variation enters a stage of exponential convergence. Finally error converges into a small and stable value in the last stage.

In each subfigure, it can be observed that the blue curve corresponds with $\gamma_1 = 0.8$, compared with the red curve corresponding with $\gamma_2 = 0.9$, has three common characteristics:

(i) The blue curve enters the error convergence stage earlier than the latter;

(ii) The blue curve has a faster convergence speed than the latter;

(iii) The blue curve converges into a smaller error than the latter.

It can be observed that the black curve corresponding with $\gamma_3 = 0.99$ always maintains large errors and cannot converge. In fact, it is found in subsequent experiments that the error function does not converge until the 3 000 000th step, and the minimum error is many orders of magnitude higher than the other two.

In Fig. 5(f), the variation trends of the errors between the true and experimental action values in different γ conditions when the agent chooses action $A = 1$ is shown. It can be observed that three groups of errors all increase exponentially at first and finally enter the stable stage. The difference is that the value of blue curve corresponded with $\gamma_1 = 0.8$ is much smaller than the value of the red curve corresponding with $\gamma_2 = 0.9$ in the stable stage. The black curve corresponding with $\gamma_3 = 0.99$ has the fastest increasing speed and the highest error value compared with the others.

According to the above experimental results, it can be concluded that for the infinite long-delay rewards RL model, a large value of γ will not only lead to too slow convergence rate of the value function, which will slow down the learning speed. In addition, the error between the final convergence value and the true value is too large, which makes it difficult for the agent to learn effectively, or even makes it learn wrongly. Therefore, in order to improve the learning efficiency and reduce the learning bias, it is most appropriate to choose the smaller value of γ as far as possible within the feasible region, rather than to choose the larger value of γ blindly. The choosing method in Section 3 and Section 4 can ensure the optimal value of γ.

7. Conclusions

Starting from practical problems, the discount rate in a long-delay rewards RL model is researched. First, through mathematical analysis of delay steps number, final reward and discount rate in the model, the feasible region of the discount rate is derived, which enables the agent to be "farsighted" enough to get the final reward under specific parameter conditions. Besides, a simple method of solving a feasible solution of the discount rate under general conditions is explored by theoretical derivation and proved by mathematical experiments. Then, the validation and practicability of the theory are verified by a series of training experiments of RL, including the supplementary experiments. Finally, the long-delay RL model is extended from the finite process to the infinite process. Through theoretical derivation, it is proved that the method of choosing discount rate researched above has the same applicability in these two kinds of process. By designing a series of contrast test, the advantages of the method of the choosing discount rate are shown. The discount rate chosen makes the gain function converge more accurate and more rapid, and improves the effect of RL

Through the in-depth research of the relationship between the discount rate and the other parameters in model, the significance of the discount rate is revealed. That not only can be applied to many RL models with

long-delay rewards, but has some guiding meanings to some problems in reality. The derivation of the feasible region of the discount rate provides a theoretical reference for the choosing of the discount rate in RL with long-delay rewards. The new method of feasible resolution makes the choosing more easily. The experiments, especially the supplementary experiments in this paper, provide a solid foundation for the theory, prove the practicability of the method, and have certain reference significance to the practical experiment implementation in the future.

References

[1] BELLMAN R. A problem in the sequential design of experiments. The Indian Journal of Statistics, 1956, 16(34): 221–229.

[2] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of Go with deep neural networks and tree search. Nature, 2016, 529(7587): 484–489.

[3] LIN C J, JHANG J Y, LEE C L, et al. Using a reinforcement Q-learning-based deep neural network for playing video games. Electronics, 2019, 8(10): 1128.

[4] TAMASSIA M, ZAMBETTA F, RAFFE W L, et al. Learning options from demonstrations: a pac-man case study. IEEE Trans. on Computational Intelligence and AI in Games, 2018, 10(1): 91–96.

[5] WYDMUCH M, KEMPKA M, JASKOWSKI W. ViZDoom competitions: playing doom from pixels. IEEE Trans. on Computational Intelligence and AI in Games, 2019, 11(3): 248–259.

[6] JADERBERG M, CZARNECKI W M, DUNNING I, et al. Human-level performance in 3D multiplayer games with population-based reinforcement learning. Science , 2019, 364(6443): 859–865.

[7] LIANG L, CHEN Y C, LIAO L C, et al. A novel impedance control method of rubber unstacking robot dealing with unpredictable and time-variable adhesion force. Robotics and Computer-Integrated Manufacturing, 2021, 67: 102038.

[8] GAO J L, YE W J, GUO J, et al. Deep reinforcement learning for indoor mobile robot path planning. Sensors , 2020, 20(19): 5493.

[9] XIE J Y, PENG X D, WANG H J, et al. UAV autonomous tracking and landing based on deep reinforcement learning strategy. Sensors, 2020, 20(19): 5630.

[10] XU X, ZUO L, LI X, et al. A reinforcement learning approach to autonomous decision making of intelligent vehicles on highways. IEEE Trans. on Systems, Man and Cybernetics Systems, 2018, 50(10): 3884–3897.

[11] HE Y, YU F R, ZHAO N, et al. Software-defined networks with mobile edge computing and caching for smart cities: a big data deep reinforcement learning approach. IEEE Communications Magazine, 2017, 55(12): 31–37.

[12] BRANDI S, PISCITELLI M S, MARTELLACCI M, et al. Deep reinforcement learning to optimise indoor temperature control and heating energy consumption in buildings. Energy and Buildings, 2020, 224(1): 110225.

[13] KHAN A, LAPKIN A, Searching for optimal process routes: a reinforcement learning approach. Computers and Chemical Engineering, 2020, 141(4): 107027.

[14] MA R, VANSTRUM E B, LEE R, et al. Machine learning in the optimization of robotics in the operative field. Current Opinion in Urology, 2020, 30(6): 808–816.

[15] PARK H, SIM M K, CHOI D G. An intelligent financial portfolio trading strategy using deep Q-learning. Expert Systems with Applications, 2020, 158(15): 113573

[16] HU Y, YAO Y, LEE W S. A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs. Knowledge-Based Systems, 2020, 204(27): 106244.

[17] AINSLIE G W. Impulse control in pigeons. Journal of the Experimental Analysis of Behavior, 1974, 21(3): 485–489.

[18] TAKAHASHI T. Loss of self-control in intertemporal choice may be attributable to logarithmic time-perception. Medical Hypotheses, 2005, 65(4): 691–693.

[19] NAKAHARA H, KAVERI S. Internal-time temporal difference model for neural value-based decision making. Neural Computation, 2010, 22(12): 3062–3106.

[20] JARMOLOWICZ D P, HUDNALL J L, HALE L, et al. Delay discounting as impaired valuation: delayed rewards in an animal obesity model. Journal of the Experimental Analysis of Behavior, 2017, 108(2): 171–183.

[21] FOSCUE E P, WOOD K N, SCHRAMM-SAPYTA N L. Characterization of a semi-rapid method for assessing delay discounting in rodents. Pharmacology Biochemistry and Behavior, 2012, 101(2): 187–192

[22] PAPALE A E, STOTT J J, POWELL N J, et al. Interactions between deliberation and delay-discounting in rats. Cognitive, Affective, & Behavioral Neuroscience, 2012, 12(3): 513–526.

[23] YAMAGUCHI Y, SAKAI Y, Reinforcement learning for discounted values often loses the goal in the application to animal learning. Neural Networks, 2012, 35(1): 88–91

[24] KNOX W B, STONE P. Framing reinforcement learning from human reward: reward positivity, temporal discounting, episodicity, and performance. Artificial Intelligence, 2015, 225(1): 24–50

[25] WANG J P, WANG G, MAO X B, et al. Motion control method of two-link manipulator based on deep reinforcement learning. Journal of Computer Applications, 2021, 41(6): 1799–1804. (in Chinese)

[26] WEI H B, HE S C. Multi-objective optimal control strategy for plug-in diesel electric hybrid vehicles based on deep reinforcement learning. Journal of Chongqing Jiaotong University (Natural Science), 2021, 40(1): 44–52. (in Chinese)

[27] LI C, HUANG Y Y, ZHANG Y L, et al. Multi-agent decision-making method based on Actor-Critic framework and its application in wargame. Systems Engineering and Electronics, 2020, 43(3): 755–762. (in Chinese)

[28] ZHANG Q H, AO B Q, ZHANG Q X. Reinforcement learning guidance law of Q-learning. Journal of Systems Engineering and Electronics, 2019, 42(2): 414–419. (in Chinese)

[29] SUTTON R S, BARTO A G. Reinforcement learning: an introduction. 2nd ed. Cadge: MIT Press, 2018.

Biographies

LIN Xiangyang was born in 1994. He received his B.S. and M.S. degrees from Air Force Engineering University, Xi'an, in 2017 and 2019, respectively, where he is currently a Ph.D. student. His research interests include reinforcement learning and intelligent decision.
E-mail: 95014052@qq.com

XING Qinghua was born in 1966. She received her B.S. degree from Shanxi University, Shanxi, China, in 1989, and M.S. and Ph.D. degrees from Air Force Engineering University, Xi'an, in 1992 and 2003, respectively, where she is currently a professor. Her research interests include system simulation modeling, combat decision analysis computer vision, and military system decision.
E-mail: qh_xing@126.com

LIU Fuxian was born in 1962. He received his B.S. degree from Lanzhou University, Lanzhou, China, in 1994, and M.S. and Ph.D. degrees from Air Force Engineering University, Xi'an, in 1998 and 2001, respectively, where he is currently a professor. His research interests include deep learning and military system decision.
E-mail: liuxqh@126.com