Actor-Critic reinforcement learning based on prior knowledge

Zhenyu Yang[*]

Qilu University of Technology Jinan, China

Received 1 March 2014, www.cmnt.lv

Abstract

In order to improve the incremental learning algorithm Actor-Critic learning efficiency, from a policy learning, introduce experience sample data into incremental Actor-Critic algorithm, make effective use of the useful information contained in the sample data of experience in the learning process. Given the recursive least-squares temporal difference, RLSTD (λ) algorithm and incremental least-squares temporal difference, iLSTD (λ) algorithms are able to make good use sample data collected in the past, respectively RLSTD and iLSTD algorithm is applied to policy evaluation Critic's. Then, Critic learned value function based on RLSTD or iLSTD algorithm, Actor gradient update strategy based on conventional parameters, so the improvement of Critic effectiveness assessment will help Actor to improve strategy-learning performance. Finally, simulation studies on two control problems with continuous state space, analyse the impact of different parameters on the performance of the learning algorithm and verify its effectiveness.

Keywords: actor-critic, RLSTD, on-policy, ILSTD

1 Introduction

Reinforcement learning as an effective method for solving a class of Markov decision model problem, has wide application in optimization and control. As a kind of reinforcement learning in the field of research, policy gradient reinforcement learning is a way to search directly on the strategy parameter space, this method overcomes the reinforcement learning algorithm based on the value of the function can not guarantee the convergence of shortcomings. However, due to the variance in the gradient estimation process is too large, resulting in policy gradient reinforcement learning method converges too slowly, thus impeding the policy gradient reinforcement learning method is widely used. By combining the value function method, come up with an Actor-Critic (AC) reinforcement learning method, AC approach combines the advantages of fast learning based RL value function method and strategies gradient RL is easy to converge. You can reduce the gradient estimation variance is an important strategy gradient reinforcement learning method has wide application in solving large-scale and high-dimensional Markov Decision learning control problems.

Actor-Critic learning structure shown in Figure 1. Actor-Critic learning consists of policy evaluation and policy improvement, which, Critic (evaluator) present a prediction problem, according to the time difference learning to estimate the value of the function, Actor (actuators) presents a control problem, update the policy parameters dynamically according to learn the value of the function. But the policy iteration method is different, Actor-Critic gradient method to update the policy parameters in the direction of increasing the expected returns, but policy iteration is by maximizing the value of the function on the action space to update the policy.

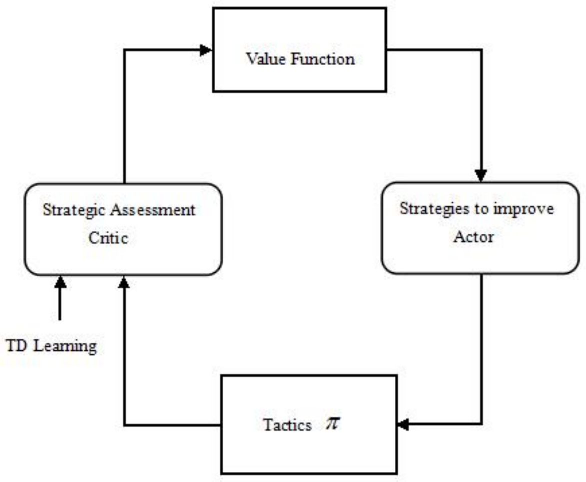


FIGURE 1 Actor-Critic architecture

Actor-Critic is an online on-policy learning algorithm, at each time step agent collects samples in accordance with current policy, based on the sample, Critic use the time difference algorithm to learn value function, Actor update strategy based on the estimated value function. When the policy updates, agent on the basis of new strategy to re-collect samples for learning, and discard the old sample, in order to obtain a satisfactory strategy, agent requires constant sampling more data, so that useful information is not contained in the old sample is fully utilized, not only waste a lot of sample data, and the impact of the algorithm's learning speed and low learning efficiency. One kind of AC algorithm can effectively improve the learning efficiency is experience playback, which can take full advantage of the useful information contained in the

[*] ***Corresponding author*** e-mail: yang_zhenyu@163.com

sample data experience. Therefore, this paper will examine how to introduce experience in the sample data in the AC learning process to improve the learning performance of the algorithm.

2 Efficient data use in incremental Actor-Critic algorithms

A major part of the Actor-Critic learning algorithm is policy evaluation, based on sample data online estimated strategy corresponding value function. TD learning to update the estimates based on the estimated value already exists, is a Bootstrapping method. AC in incremental learning algorithm, at each time step Critic learning value function according to the TD, Actor according to a biased conventional gradient to update policy parameters, in order to facilitate comparison with the algorithms behind this algorithm is denoted as AC-TD.

In the policy evaluation, TD learning at each time step only based on the current sample to estimate value function, a small amount of calculation, but it cannot effectively use the last sample data, you need a lot of time step to learn an accurate estimate. In order to improve data availability, while overcoming the difficulties of learning step design. Bradtke put forward a linear value function approximation based on LSTD learning algorithm, the algorithm directly to Markov decision process value function approximation of mean square error as the performance index [4]. In order to further improve the generalization performance of LSTD algorithm, Boyan et. combined eligibility trace λ extended for LSTD and proposed LSTD (λ) algorithm [5].When using the function approximator solving a value function parameters, although LSTD (or LSTD (λ)) algorithm can make more effective use of sample data considerably more than the last TD algorithm to get the correct estimate, however, when asked questions has great feature space and the time required for online estimation, LSTD algorithm at each time step requires a lot of computing, resulting in decreased learning efficiency. Therefore, LSTD online learning algorithm for solving problems with a large number of feature vectors is impractical. In order to reduce the computational burden LSTD (λ) algorithm, Xu et proposed a Recursive LSTD (λ) - RLSTD (λ) algorithm [9]. RLSTD (λ) algorithm is more suitable for online learning, and more effective use sample data than TD algorithm. In addition, Geramifard et put forward an Incremental LSTD, iLSTD learning algorithm [6] and then combined eligibility trace to extend iLSTD and proposed iLSTD (λ) algorithm [7]. iLSTD (λ) algorithm played a role of a compromise between TD algorithm in a low utilization rate of the data and a large computation complexity of LSTD algorithm, it is more effective to use sample data than TD to obtain a good approximation, and compared with LSTD algorithm, reducing the computational complexity. In summary, in order to overcome the problem of low utilization of data caused by TD Critic assessment in incremental algorithm in AC, as well as improving the efficiency of learning algorithms, in this paper, starting at learning strategies, the introduction of experience in policy evaluation sample incremental algorithm in AC. The two variants LSTD (λ) method (i.e. RLSTD (λ) and iLSTD (λ)) were applied to the Critic proposes two new incremental AC algorithm, namely AC-RLSTD and AC-iLSTD. Critic use RLSTD (λ) or iLSTD (λ) algorithm to estimate the value function, Actor update strategy based on a conventional gradient obtained by the TD error. Critic use two variants LSTD (λ) method in policy evaluation to improve the performance Critic learning assessments, and thus can more efficiently update Actor strategy parameters based on the assessment results.

3 Based on TD incremental Actor-Critic learning

Policy gradient reinforcement learning goal is to learn an optimal or suboptimal strategy, namely to estimate expected returns relative to the gradient of the policy parameters, and then use the gradient parameter adjustment the policy parameters. According to the policy gradient theorem, the expected return on the policy parameters 'Vanilla' gradient can be expressed as [14]:

$$\begin{aligned}&\frac{\partial J(\theta)}{\partial\theta}=\sum_s d^\pi(s)\sum_a\frac{\partial\pi(a|s,\theta)}{\partial\theta}Q^\pi(s,a)=\\&\sum_s d^\pi(s)\sum_a\nabla_\theta\pi(a|s,\theta)(Q^\pi(s,a)-b^\pi(s))\end{aligned}, \qquad (1)$$

where $b^\pi(s)$ represents a baseline, other, Bhatnagar et al [15,9] and Sutton et al [16] proved $Q^\pi(s,a)-b^\pi(s)$ that you can use a compatible function approximation $f_{\bar{w}}(s,a)=\bar{w}^T\psi_{sa}$ to express, where, $\bar{w}$ is a parameter vector, when $b^\pi(s)=V^\pi(s)$,The device is compatible function approximation of minimum mean square error [15,9]. Therefore, the Equation (1) can be written as:

$$\begin{aligned}&\nabla_\theta J(\theta)=\frac{\partial J(\theta)}{\partial\theta}=\\&\sum_s d^\pi(s)\sum_a\frac{\partial\pi(a|s,\theta)}{\partial\theta}(\bar{w}^{*\mathrm{T}}\psi_{sa})=\\&\sum_s d^\pi(s)\sum_a\pi(a|s,\theta)\psi_{sa}A^\pi(s,a)\\&\psi_{sa}=\nabla_\theta\log\pi(a\,|\,s,\theta),\end{aligned} \qquad (2)$$

where, $A^\pi(s,a)=Q^\pi(s,a)-V^\pi(s)$ is an advantage function.

From the above Equation (2) can be seen, $\nabla_\theta J(\theta)$ calculation depends on the advantages of function $A^\pi(s,a)$, however, in [14] pointed out. Advantage of $A^\pi(s,a)$ function not just in the approximation $f_{\bar{w}}^{\ \pi}(s,a)$ is obtained based on TD learning. To this end, Bhatnagar et al [15, 9] and Morimura [11] by TD error status value function to construct the advantages of function.

According to Bellman equation, TD error is defined as: $\delta_t = r_{t+1} + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$. Text [15, 9, 11] proved that δ_t consistent estimate of the advantages of function $A^{\pi}(s,a)$.

However, calculate δ_t need to a known state value function, now use a linear function approximation to estimate the value function, the estimated state value function is $\hat{V}^{\pi}(s) = \varphi(s)^{\mathrm{T}}\omega$. TD error can be estimated according to $\hat{\delta}_t = r_{t+1} + \gamma\varphi(s_{t+1})^{\mathrm{T}}\omega_t - \varphi(s_t)^{\mathrm{T}}\omega_t$, where ω_t represents the parameter vector at t times update. In incremental AC algorithm, Critic on the basis of TD error estimated to update value function parameters, Actor biased according to a conventional gradient update strategy parameters:

$$\omega_{t+1} = \omega_t + \alpha_t\hat{\delta}_t\varphi(s_t)\,, \tag{3}$$

$$\theta_{t+1} = \theta_t + \beta_t\hat{\delta}_t\psi_{s_t a_t}\,, \tag{4}$$

where, α_t and β_t are the Critic and Actor update step, meet the following conditions:

$$\begin{aligned} &\sum_t \alpha_t = \sum_t \beta_t = \infty \\ &\sum_t \alpha_t^2, \sum_t \beta_t^2 < \infty\ . \\ &\beta_t = \mathrm{o}(\alpha_t) \end{aligned} \tag{5}$$

From the Equation (3) can be seen, update the parameter ω is only determined by the data that current observed in the sample, after the current update is completed, the sample is discarded. Consider the case of non-sparse feature representation, Critic at each time step only needs to calculate the amount of $\mathrm{O}(k)$, but, Critic wasted a lot of samples, resulting in the learning process needs to continue to collect more data in order to obtain a satisfactory sample gradient estimates.

4 Based RLSTD (λ) or ILSTD (λ) incremental Actor-Critic learning

TD playback experience is a way to improve the effectiveness of learning data [5]. For example, at each time step, LSTD (λ) method put all observations TD update set to zero to solve the value function parameters. However, the data validity LSTD (λ) method at the expense of a large amount of computation. In order to balance the relationship between data validation and calculation effectiveness of the Critic assessment, respectively RLSTD (λ) and iLSTD (λ) is introduced into Critic and proposes two new incremental AC algorithm.AC algorithm proposed in, Critic according RLSTD (λ) or iLSTD (λ) algorithm to learn the value function, Actor estimate $\hat{\nabla}_{\theta}J(\theta)$ parameter update strategy based on a regular gradient.

4.1 BASED RLSTD (λ) TO ASSESS THE CRITIC

Let $u_t(\omega)$ be the sum of the TD updated within t times, according to TD and the definition of the error function of the state value, are:

$$\begin{aligned} u_t(\omega) &= \sum_{n=1}^{t} u_n(\omega) = \sum_{n=1}^{t} z_n(r_{n+1} + \gamma\varphi(s_{n+1})^{\mathrm{T}}\omega - \varphi(s_n)^{\mathrm{T}}\omega) \\ &= \sum_{n=1}^{t} z_n r_{n+1} - \sum_{n=1}^{t} z_n(\varphi(s_n) - \gamma\varphi(s_{n+1}))^{\mathrm{T}}\omega = b_t - \tilde{A}_t\omega \end{aligned} \tag{6}$$

where, $z_n = \gamma\lambda z_{n-1} + \varphi(s_n)$ is a qualified track, $\lambda \in [0,\ 1]$ is a parameter set in advance. Online learning, LSTD observed data into each vector and matrix b, $\tilde{A}$, then order sum DT update to calculation parameters, and order the Equation (6) is equal to 0, to obtain a new parameter: $\omega_{t+1} = \tilde{A}_t^{-1}b_t$. A and b when the update is completed, observation of the sample data is ignored, and at this time there is no loss of information.

When b and $\tilde{A}$ of the update is completed, observation of the sample data is ignored, and at this time there is no loss of information. In LSTD algorithm, each time step in the computational complexity of the inverse matrix is A, large computational burden. To this end, the policy evaluation incremental AC algorithms in this section use RLSTD (λ) algorithm to solve LSTD (λ) the computational complexity of the problem.

Order $F_t = A_t^{-1}$, $F_0 = \rho I$, $G_{t+1} = F_{t+1}z_t$, wherein A is a positive number, is a unit matrix. Find the matrix inverse theorem [12] shows that the value of the function parameter update rules is as follows:

$$G_{t+1} = F_t z_t / (1 + (\varphi(s_t)^{\mathrm{T}} - \gamma\varphi(s_{t+1})^{\mathrm{T}})F_t z_t)\,, \tag{7}$$

$$\omega_{t+1} = \omega_t + G_{t+1}(r_t - (\varphi(s_t)^{\mathrm{T}} - \gamma\varphi(s_{t+1})^{\mathrm{T}})\omega_t)\,, \tag{8}$$

$$F_{t+1} = F_t - \frac{F_t z_t(\varphi(s_t)^{\mathrm{T}} - \gamma\varphi(s_{t+1})^{\mathrm{T}})F_t}{1 + (\varphi(s_t)^{\mathrm{T}} - \gamma\varphi(s_{t+1})^{\mathrm{T}})F_t z_t}\,. \tag{9}$$

Calculation of each time step is $\mathrm{O}(k^2)$. There is an additional parameter in Critic evaluation, the initial value ρ of the initial covariance matrix F_0. As Xu et al [9] above, the initial constant ρ plays an important role in convergence RLSTD (λ) of the algorithm. Performance RLSTD (λ) has a larger initial constant of ρ is similar to the LSTD algorithm. In some cases, RLSTD having a smaller value of ρ RLSTD algorithm is faster than having a larger value of ρ algorithm convergence speed, can refer to this phenomenon [13] Theoretical analysis. In [13] Moustakides proposed a recursive least squares method, noting that at moderate or high SNR should use a relatively small initial matrix, and in the low SNR choose a relatively large initial matrix for optimal performance. Based on the above analysis, similar to RLSTD (λ) algorithm, this paper will analyse the initial constant value ρ in the following simulation impact AC algorithm performance based RLSTD (λ) incremental.

4.2 ILSDR BASED (λ) TO ASSESS THE CRITIC

Unlike LSTD algorithms, iLSTD (λ) algorithm to solve the value function parameters in incremental form.TD update error until all observations data reduced to zero. In iLSTD (λ) algorithm, the update with the status transition and value function parameters occurred, $u_t(\omega_t)$ calculation of the incremental update. Further, based on the observed state transition and returns a new, $\tilde{A}_t$ and b_t are incrementally updated:

$$b_t = b_{t-1} + r_{t+1} z_t = b_{t-1} + \Delta b_t \,, \tag{10}$$

$$\tilde{A}_t = \tilde{A}_{t-1} + z_t(\varphi(s_t) - \gamma\varphi(s_{t+1}))^{\mathrm{T}} = \tilde{A}_{t-1} + \Delta\tilde{A}_t \,, \tag{11}$$

where, Δ represents the amount of change in adjacent time variable. Given $\tilde{A}_t$ and b_t, there is:

$$u_t(\omega_t) = u_{t-1}(\omega_t) + \Delta b_t - (\Delta\tilde{A}_t)\omega_t \,. \tag{12}$$

LSTD and TD algorithm updates the parameter vector ω is composed of all the elements, and iLSTD (λ) algorithm updates only ω of all the constituent elements of a small portion. For example, consider updating the first *A* elements:

$$\omega_{t+1} = \omega_t + \alpha_t u_t(i) e_i \,, \tag{13}$$

where, $u_t(i)$ is the first element of u_t, i, e_i is a column vector, e_i mere element in row i is 1, the remaining behaviour 0.Then incremental form *A* is:

$$u_t(\omega_{t+1}) = u_t(\omega_t) - \alpha_t u_t(i)\tilde{A}_t e_i \,. \tag{14}$$

According to Equation (13) and (14),each time you select an element update, repeated many times, you can complete the updated parameter vector all elements of ω .Thus, an element of choice here will inevitably encounter problems, choose which element to update it? Geramifard and put forward two commonly used feature selection mechanism in [7]: random selection mechanism and greedy selection mechanism. Random selection mechanism based iLSTD (λ) (iLSTD-random) algorithm can converge to a TD (λ) the same result, and the selection mechanism based iLSTD greedy (λ) (iLSTD-greedy) algorithm does not satisfy the convergence because all full conditions, there is no guarantee of convergence. However, Geramifard etc. found in [7] in the performance iLSTD-greedy algorithm is slightly better than iLSTD-random algorithm. In the simulation study the following learning control problems, we also found that although iLSTD-greedy algorithm converges in theory, the lack of guarantee, however, learning performance selection mechanism based on greedy algorithm is superior to AC-iLSTD based on random selection mechanism AC-iLSTD algorithm.

The main idea of greedy selection mechanism is to select the elements with the largest sum TD update, i.e. $i = \arg\max(|u_t(i)|)$. Suppose that $\tilde{k}$ represents the number of elements need to be updated at each time step, the iLSTD (λ) is calculated for each time step in the algorithm complexity of $\mathrm{O}(k^2 + \tilde{k}k)$. For online learning, when the number represented by the characteristic dimensions are very large, the calculation efficiency iLSTD (λ) is higher than LSTD algorithm, and can be used more effectively than in the past sample data TD algorithm.

4.3 ALGORITHM STEPS

Based on the above analysis, are given based on RLSTD (λ) or iLSTD (λ) AC incremental learning algorithm as follows:

Step 1. Given a random parameters strategy $\pi(a\,|\,s,\theta)$, feature vector valued functions $\varphi(s)$, there are $\psi_{sa} = \nabla_\theta \log\pi(s,a)$.

Step 2. Initialization vector policy parameters $\theta = \theta_0$, valued functions parameter vector $\omega = \omega_0$, two learning step $\alpha = \alpha_0$ and $\beta = \beta_0$, the discount factor γ , eligibility trace parameter λ , the initial variance matrix parameters ρ ,convergence error ε . Select an initial state $s_0 \in S$, order $z_{-1} = 0$, $\tilde{A}_{-1} = 0$, $u_{-1} = 0$.

Step 3. Select the action according to *A*, observe the next state *B*, and get immediate return r_{t+1} .

Step 4. If you are an absorbing state, then s_{t+1} is set to an initial state, so that eligibility trace z_t is a zero vector. Otherwise, update qualifications trace z_t .

Step 5. Critic use RLSTD (λ) algorithm according to the Equation (7) - (9) to update the value of the function parameters ω_t , or the use of iLSTD (λ) algorithm according to the Equations (10) - (14) update value of the function parameter ω_t .

Step 6. Update TD error δ_t value function according to learn, Actor according to Equation (4) to update the policy parameters θ_t .

Step 7. If you terminate the algorithm, otherwise go to Step 3.

5 Simulation study

In this paper inverted pendulum on two proposed algorithms to assess the incremental AC, the proposed AC-

RLSTD (λ) and AC-iLSTD (λ) two algorithms with AC-TD algorithm were compared. For AC-iLSTD algorithm, characteristics were tested using a random selection mechanism AC-iLSTD (AC-iLSTD-random) feature selection algorithm and greedy mechanism AC-iLSTD (AC-iLSTD-greedy) performance of the algorithm, in both feature selection mechanism, the number of elements selected for each update is $\tilde{k}=1$.

In order to compare different incremental AC algorithm performance, random selection of parametric Gibbs distribution strategy

$$\pi(a|s,\theta)=\frac{e^{\theta^{\mathrm{T}}\varphi_{sa}}}{\sum_{a'\in A}e^{\theta^{\mathrm{T}}\varphi_{sa'}}},$$

where φ_{sa} is a $k\times|A|$ dimensional state - action feature vector. The initial policy parameters θ_0 and ω_0 the initial state value function parameter is set to 0. Critic and Actor step length, respectively [15]:

$$\beta_t=\frac{\beta_0\cdot\beta_c}{\beta_c+t},\ \alpha_t=\frac{\alpha_0\cdot\alpha_c}{\alpha_c+t^{2/3}}.$$

Which, according to the empirical method, $\alpha_0=0.1$, $\beta_0=0.01$, $\alpha_c=1$, $\beta_c=1{,}000{,}000$, and step length rule satisfies (1-5), $\gamma=0.95$, $\varepsilon=10^{-6}$.

5.1 INVERTED PENDULUM

Inverted pendulum control is a nonlinear, complex system instability control problems in automatic control, artificial intelligence and machine learning methods are usually used to test the performance of different learning. Reinforcement learning environment as a method for solving a class of unknown model complex problems, often used to solve the inverted pendulum control problem, in order to test the performance of reinforcement learning.

This article uses an inverted pendulum [8] described the model, shown in Figure 2. Inverted pendulum problem is solved by applying a force to the car to balance an unknown quality and length of the pendulum, the pendulum system in a small car. Continuous state space, φ vertical angle and angular velocity of the pendulum composition $\dot{\varphi}$: $s=(\varphi,\ \dot{\varphi})^{\mathrm{T}}$. Action space is discrete from the left edge -50 N, and the right power +50 N 0 N composed of three actions. Simulation, the inverted pendulum system is described by the following equation:

$$\ddot{\varphi}=\frac{g\sin\varphi(m_{\mathrm{c}}+m_{\mathrm{p}})-(a+m_{\mathrm{p}}l_{\mathrm{p}}\dot{\varphi}^2\sin\varphi)\cos\gamma}{\frac{4}{3}l_{\mathrm{p}}(m_{\mathrm{c}}+m_{\mathrm{p}})-m_{\mathrm{p}}l_{\mathrm{p}}\cos^2\varphi},\qquad(15)$$

wherein, $g=9.8\ \mathrm{m/s^2}$ is the gravity constant, $m_{\mathrm{p}}=2.0\ \mathrm{kg}$ is the mass of the pendulum, $m_{\mathrm{c}}=8.0\ \mathrm{kg}$ is the mass of the trolley, $l_{\mathrm{p}}=0.5\ \mathrm{m}$ is the length of the pendulum. Simulation process, using fourth-order Runge - Kutta method to simulate dynamic systems, simulation time step is set 0.1s. When $|\varphi|\le\pi/2$ smart body was immediately return 0.The period of time from the initial state to a failed state is defined as between a scene, each scene is set to the initial state equilibrium $(0,\ 0)^{\mathrm{T}}g$ .When $|\varphi|>\pi/2$ curtain or greater than the length of training 1000, they think the system encountered a failed state, the scene stops, agent get in return is -1.

Simulation, the state of the feature vector consists of a constant and 9 radial basis function consists of:

$$\varphi(s)=(1,\ e^{-\frac{\|s-c_1\|^2}{2\sigma^2}},\ \cdots,\ e^{-\frac{\|s-c_9\|^2}{2\sigma^2}})^{\mathrm{T}},$$

wherein, $\{c_i\}_{i=1}^{9}$ is located in the state space of nine points a 3 × 3 two-dimensional lattice on $\left\{-\frac{\pi}{4},\ 0,\ +\frac{\pi}{4}\right\}\times\{-1,\ 0,\ +1\}$, $\sigma^2=1$. Based learning strategies, according to the expectations of rewards and balanced steps to assess the different incremental AC algorithms, and analyse the impact eligibility trace algorithm parameters and AC-RLSTD initial variance matrix parameters ρ for its performance of the algorithm. In every training to learn the strategy behind an assessment that is based on learning strategies, so that the inverted pendulum from the initial state of the test run was repeated 10 times. When first run time step over 1000, it has been able to successfully control considered inverted pendulum balance. From the same initial policy and state value function parameters, every one AC algorithm test was repeated 20 times.

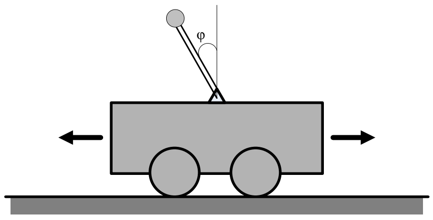
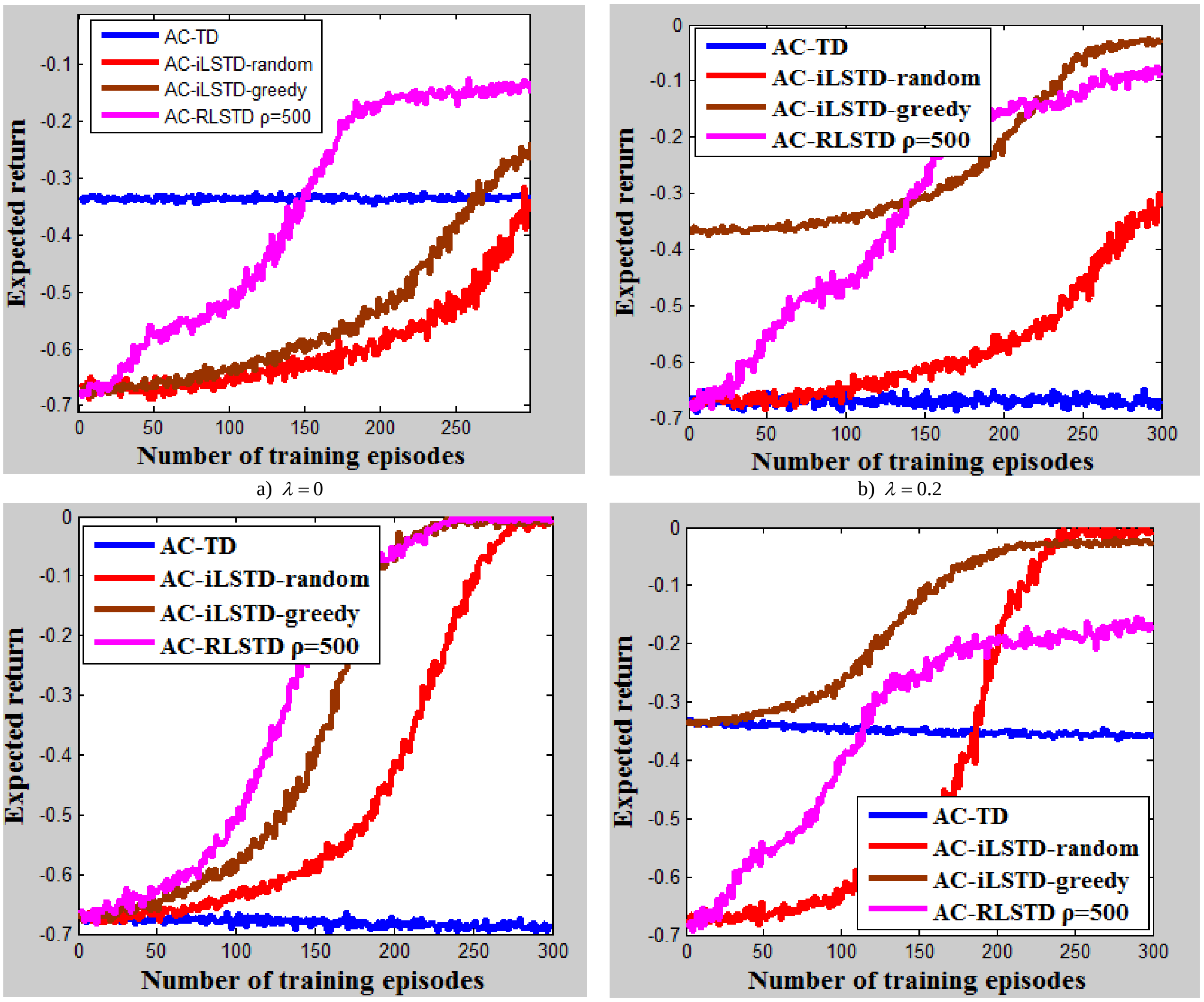


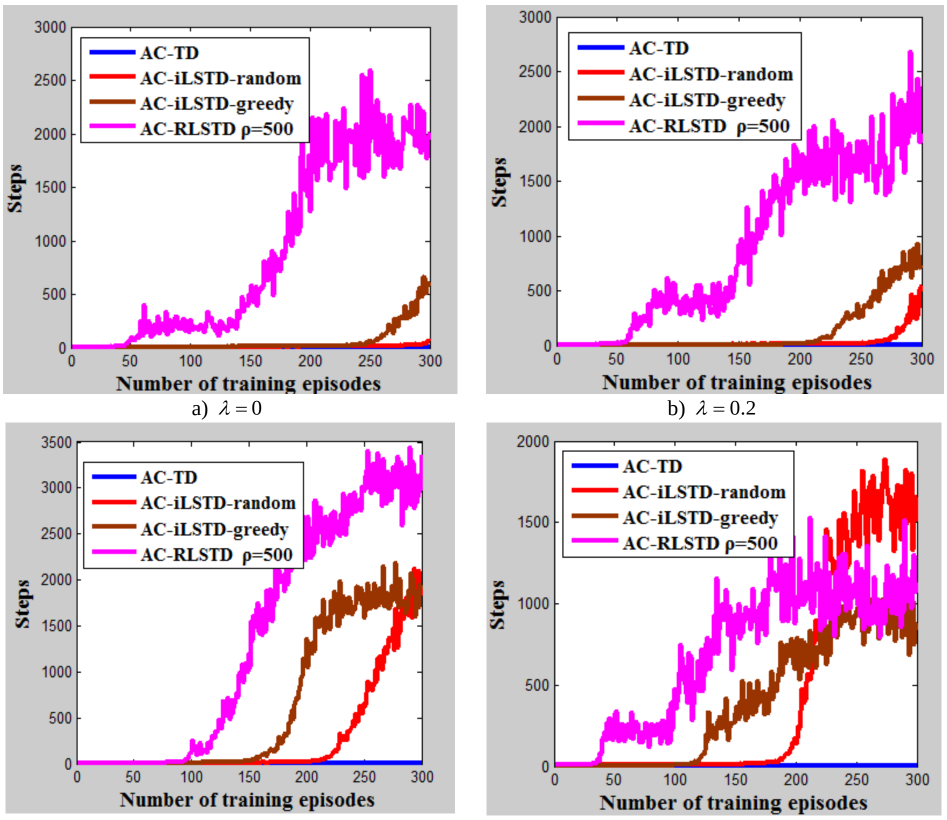
FIGURE 2 Inverted pendulum model

Figure 3 and Figure 4 shows the comparative performance of different AC algorithms under four different values, the horizontal axis represents the number of training curtain ordinate were expected returns and balance the number of steps the average results of 20 runs, where the initial variance matrix parameters AC-RLSTD algorithm ρ to 500. As can be seen from Figures 3 and 4, with the increase of the two AC performance of the algorithm proposed in this section A is significantly improved, the number of steps and the balance obtained by the AC-RLSTD than the AC algorithm step-iLSTD balancing algorithm to obtain a large number. Also found that, although the lack of convergence iLSTD-greedy algorithm guarantee, however, when $\lambda \neq 1$, AC-iLSTD-greedy algorithm outperforms AC-iLSTD-random algorithm. For $\lambda = 1$ situation, AC-RLSTD and AC-iLSTD-greedy algorithm performance declined slightly, AC-iLSTD-random algorithm outperforms both above. In the AC-TD any algorithm λ is unstable and, when the worst performance $\lambda = 0.2$ and $\lambda = 0.8$. Therefore, can be drawn from the above analysis, introduce the RLSTD (λ) and iLSTD (λ) algorithm into Critic assessment, AC-RLSTD and AC-iLSTD algorithm can get better than AC-TD algorithm performance. Although the AC-RLSTD and AC-iLSTD each time step algorithm requires more computation than the AC-TD algorithm, but their data is AC-TD high utilization ratio, this is because the evaluation RLSTD (λ) and the Critic iLSTD (λ) than TD (λ) can make more effective use of empirical data, thus requiring fewer screen or data to learn a better strategy to successfully control the inverted pendulum balance.



a) $\lambda = 0$ b) $\lambda = 0.2$ c) $\lambda = 0.8$ d) $\lambda = 1.0$

FIGURE 3 Expected returns obtained by different AC algorithms with different λ values

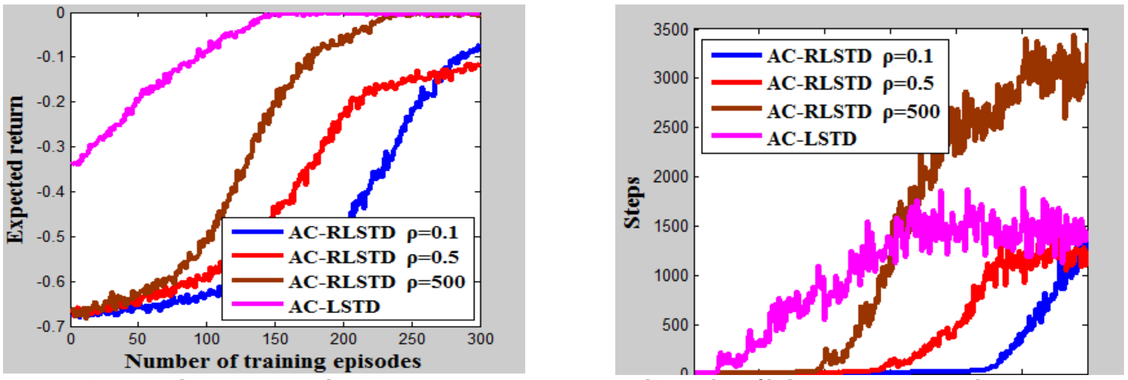


a) $\lambda = 0$ b) $\lambda = 0.2$

c) $\lambda = 0.8$ d) $\lambda = 1.0$

FIGURE 4 Number of balancing steps obtained by different AC algorithms with different λ values

As mentioned in the previous section, the initial value of the initial variance matrix will affect the convergence performance RLSTD (λ) algorithm will RLSTD (λ) algorithm is applied to the AC, AC-RLSTD algorithm above problems also exist. For this reason, the impact of the following test parameters ρ study on the performance of AC-RLSTD. Due to the constant RLSTD larger initial learning performance (λ) algorithm and LSTD (λ) is similar, therefore, has a larger initial constant learning performance AC-RLSTD algorithm perhaps is similar to AC (AC-LSTD) algorithm based LSTD (λ) assessment. To test this possibility, the following also compare the performance AC-RLSTD between the algorithm and the AC-LSTD. Figure 5 shows the performance with different A and B values in the AC-RLSTD algorithm, wherein Figure 5a) and Figure 5b), respectively, by the expected return AC-RLSTD balancing algorithm to obtain the number of steps and the average results of 20 runs. As can be seen, the performance of the algorithm when $\rho = 500$ the AC-RLSTD better than at the time $\rho = 0.1$ and $\rho = 0.5$ .Meanwhile, the simulation results also verified the AC-RLSTD algorithm performance with a larger initial constant A is similar to AC-LSTD. Therefore, when the use of AC-RLSTD algorithm inverted pendulum problem, choose a larger parameter B can obtain a satisfactory performance.



a) Expected returns averaged over 20 runs b) Number of balancing steps averaged over 20 runs

FIGURE 5 Performance comparison of AC-RLSTD algorithm with different initializing constants

6 Conclusions

In summary, compared with the AC-TD method, when $0<\lambda<1$, AC-iLSTD and AC-RLSTD algorithm may use the screen or less to obtain a good sample data strategy and their learning better performance. Although iLSTD-greedy algorithm does not guarantee convergence, however, in both learning control problems found AC-iLSTD-greedy algorithm than the AC-iLSTD-random algorithm has better performance. In addition, the simulation also found that the initial constants ρ has an impact on AC-RLSTD algorithm performance. In the inverted pendulum learning problems, AC-RLSTD use a relatively large value of ρ can get a better performance, while in the car climbing the learning control problems, AC-RLSTD in a relatively small value of ρ can get a better performance.AC-RLSTD algorithm showed two different characteristics according to the recursive least squares method different SNR [13] to explain the situation.

References

[1] Sutton R S, Barto A G 1998 Reinforcement learning: an introduction Cambridge MIT Press

[2] Hirashima Y 2010 A reinforcement learning system for transfer scheduling of freight cars in a train *Proceedings of the International Multi-Conference of Engineering and Computer Scientists* 112-7

[3] Shen Z P, Guo C, Zhang N 2010 A general fuzzified CMAC based reinforcement learning control for ship steering using recursiveleast-squares algorithm *Neurocomputing* **73**(4-6) 700-6

[4] Bradtke SJ, Barto AG 1996 Linear least-squares algorithms for temporal difference learning *Machine Learning* **22**(1-3) 33-57

[5] Boyan J A 2002 Technical update: least-squares temporal difference learning *MachineLearning* **49** 233-46

[6] Geramifard A, Bowling M, Sutton R S 2006 Incremental least-squares temporal difference learning *Proceedings of the Twenty-First National Conference on Artificial Intelligence Boston Massachusetts The AAAI Press* 356-61

[7] Geramifard A, Bowling M, Zinkevich M 2006 iLSTD: eligibility traces and convergence analysis *Proceedings of advances in neural information processing systems Vancouver Canada The MIT Press* 826-33

[8] Lagoudakis MG, Parr R 2003 Least-squares policy iteration *Journal of Machine Learning Research* **4** 1107-49

[9] Bhatnagar S, Sutton R S, Ghavamzadeh M 2009 Natural-gradient actor-critic algorithms *Automatica* **45**(11) 2471-82

[10] Xu X, He H, Hu D 2002 Efficient reinforcementlearning using recursive least-squares methods *Journal of Artificial Intelligence Research* **16** 259-92

[11] Morimura T, Uchibe E, Doya K. 2005 Utilizing natural gradient in temporal difference reinforcement learning with eligibility traces *Proceedings of the 2nd International Symposium on Information Geometry and its Applications* 256-63

[12] Ljung L, Soderstron T 1983 Theory and practice of recursive identification *Cambridge MIT Press*

[13] Moustakides G V 1997 *IEEE Transactions on Signal Processing* **45**(10) 2468-76

[14] Peters J, Schaal S 2008 Natural actor-critic *Neurocomputing* **71** 1180-90

[15] Bhatnagar S, Sutton R S, Ghavamzadeh M 2007 Incremental natural actor-critic algorithms *Proceedings of Advances in Neural Information Processing Systems Vancouver Canada The MIT Press* 105-12

[16] Sutton R S, McAllester D, Singh S 2000 Policy gradient methods for reinforcement learning with function approximation *Proceedings of Advances in Neural Information Processing Systems* 1057-63

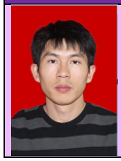
[17] Williams R J 1992 Simple statistical gradient-followingalgorithms for connectionist reinforcement learning *Machine Learning* **8**:229-56

[18] Baxter J, Bartlett P L 2001 Infinite-horizon policy-gradient estimation *Journal of Artificial Intelligence Research* **15**:319-50

[19] Greensmith E, Bartlett PL, Baxter J 2004 Variance reduction techniques forgradient estimation in reinforcement learning *Journal of Machine Learning Reseach* **5** 1471-530

[20] Weaver L, Tao N 2001 The optimal reward baseline forgradient-based reinforcement learning *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence Washington* 538-45

Author



Yang Zhenyu, born on September 1, 1980, Jinan, China

Current position, grades: lecturer.
University studies: computer software.
Scientific interest: machine learning, visualization in scientific computing and artificial intelligence.