LETTER

Reward-Based Exploration: Adaptive Control for Deep Reinforcement Learning

Zhi-xiong XU[†a)], **Lei CAO**[†], **Xi-liang CHEN**[†], ***Nonmembers*, *and*** **Chen-xi LI**[†], ***Student Member***

SUMMARY Aiming at the contradiction between exploration and exploitation in deep reinforcement learning, this paper proposes "reward-based exploration strategy combined with Softmax action selection" (RBE-Softmax) as a dynamic exploration strategy to guide the agent to learn. The superiority of the proposed method is that the characteristic of agent's learning process is utilized to adapt exploration parameters online, and the agent is able to select potential optimal action more effectively. The proposed method is evaluated in discrete and continuous control tasks on OpenAI Gym, and the empirical evaluation results show that RBE-Softmax method leads to statistically-significant improvement in the performance of deep reinforcement learning algorithms.

key words: *deep reinforcement learning, reward, exploration, exploitation*

1. Introduction

Reinforcement learning (RL) studies that how an agent maximize rewards in a previously unknown environment through trial and error. An agent can hardly find an optimal policy unless it has sufficiently explored the environment. As a result, how to balance the ratio between exploration and exploitation is one of core challenges in RL, which has great effect on the agent's learning process. On the one hand, too much exploration prevents the agent from maximizing short-term reward because selected exploration actions may yield negative reward from the environment. On the other hand, exploiting uncertain environment knowledge prevents from maximizing long-term reward since selected actions may remain suboptimal. This problem is well known as the dilemma of exploration and exploitation [1].

Most of the recent state-of-the-art RL algorithms have been using simple exploration strategies such as ε-*greedy* method [2]. Compared with uniform sampling [3] and independent identically distributed/correlated Gaussian noise [4] method, whose sample complexity grows exponentially with state space size in tasks, ε-*greedy* method is more efficient and requires little memory space for environment. However, the disadvantage exists in this method is obvious, ε-*greedy* exploration strategy highly depends on the careful setting of meta-parameters, which are usually tuned by hand instead of taking advantage of the agent's learning process. In fact, it's unclear for agent when to adapt the parameters ε in a given learning task. As a result, how to tune ε for a better performance becomes a time-consuming work especially in the face of relatively complex problems. Value-Difference Based Exploration combined with Softmax action selection (VDBE-Softmax) [5] is one of methods to solve this problem, which utilizes the temporal-difference error as a measure of the agent's uncertainty about the environment to adapt exploration parameters, but it rely on the statistics of different states of the ε value and can't operate on the tasks with large-scale continuous state space. Count-based exploration [6] method uses a hash table to record the state visited, thus adjusting the direction of exploration, but the division of the state of the problem requires high. Variational Information Maximizing Exploration (VIME) [7] encourage agent to explore by acquiring information about environment dynamically, but only for simple environment. Those exploration strategies can only be specific to their own problem domain, we haven't seen a simple and fast method that can work across different domains.

In this paper, we present a reward-based exploration method called RBE-Softmax exploration strategy, which combines reward-difference method and Softmax exploration strategy. Our exploration strategy not only take advantage of agent's learning process to adapt exploration parameters dynamically, but also can solve the problems with discrete and continuous state or action space.

We validated our proposed method on OpenAI Gym with the discrete and continuous control tasks. By combining different deep reinforcement learning algorithms, we present empirical evidence that RBE-Softmax exploration strategy is able to guide the process of learning and improve the performance of basic algorithms.

2. Reward-Based Exploration Combined with Softmax Action Selection Method

Currently, one of the most common deep reinforcement learning algorithms is the DQN (Deep Q Networks) algorithm [3], which introduce two necessary methods to ensure the stability and efficiency. One is experience replay, observed transitions are stored in memory bank and sampled uniformly to update the network. The other one is the target network, which aims at improving the stability of the DQN algorithm.

However, while DQN solves problems with high-dimensional observation spaces, it can only handle discrete and low-dimensional action spaces. Lillicrap [8] proposes an actor-critic algorithm called Deep DPG (DDPG) based

Manuscript received January 17, 2018.
Manuscript revised May 3, 2018.
Manuscript publicized June 1, 2018.

[†]The authors are with Institute of Command Information System, Army Engineering University, Nanjing, 210007, China.
a) E-mail: xu.nj@foxmail.com

DOI: 10.1587/transinf.2018EDL8011

on the deterministic policy gradient, which can operate over continuous action spaces and robustly solves more than 20 simulated physics tasks.

2.1 Basic Exploration/Exploitation Strategies

Two widely used methods for balancing the ratio between exploration and exploitation are *ε-greedy* and Softmax. For *ε-greedy* exploration strategy, the agent selects a random action with a fixed probability ε, $0 \leq \varepsilon \leq 1$, the exploration strategy in detail is

$$\pi(s) = \begin{cases} \textit{random action from } A(s) & \textit{if } \xi < \varepsilon \\ \arg\max_{a \in A(s)} Q(s,a) & \textit{otherwise} \end{cases} \tag{1}$$

Where $\pi(s)$ is an agent's policy, $Q(s,a)$ is the value function, ε is a uniform random number drawn at each time step.

For Softmax method, it chooses action according to action-selection probabilities, which depends on the rank of the value function estimates using a Boltzmann distribution:

$$\pi(a|s) = \Pr\{a_t = a | s_t = s\} = \frac{e^{\frac{Q(s,a)}{\tau}}}{\sum_{a \in A} e^{\frac{Q(s,a)}{\tau}}} \tag{2}$$

Where τ is a positive parameter called Boltzmann temperature. All actions will be equiprobable if temperature τ is high, whereas low temperature τ will cause greedy action selections.

In fact, both exploration strategies have advantages and disadvantages. *ε-greedy* method is efficient in practice but when it explores it choose equally among all actions, which might bring the worst action. Softmax method varies the action probability according to estimated value, but Boltzmann temperature is sensitive and difficult to adjust. Both policies have been reported as methods for describing the action-selection process in the deep reinforcement learning algorithms.

2.2 Exploration Based on Reward

Previously VDBE and VDBE-softmax methods have been applied in solving bandit problems by adapting exploration parameters dynamically. However, one of drawbacks of those exploration strategies is that they have to record exploration parameters for each state, it's inefficient when encountered with large-scale continuous state or action space. As a result, we propose a versatile method called RBE-Softmax, which utilizes the difference between mid-term reward and long-term reward to adapt exploration parameter ε dynamically.

In order to control the exploration policy on the basis of the learning process, the core idea of RBE-Softmax method is to extend the *ε-greedy* and Softmax strategies by introducing a reward-dependent exploration probability, $\varepsilon(r)$, instead of hand-tuning a global parameter. As a result, we introduce the mid-term reward $r_{MT}(t)$ at time t and the long-term reward $r_{LT}(t)$ at time t are defined as follows:

$$r_{MT}(t) = \frac{r_{MT}(t-1)}{\tau_{MT}} + r(t) \tag{3}$$

$$r_{LT}(t) = \frac{r_{LT}(t-1)}{\tau_{LT}} + r_{MT}(t) \tag{4}$$

$$\Delta r = |r_{LT}(t) - r_{MT}(t)| \tag{5}$$

where Δr means the difference between mid-term reward and long-term reward, $r(t)$ refers to an instant reward at time t, τ_{MT} and τ_{LT} represent the time constants for $r_{MT}(t)$ and $r_{LT}(t)$ respectively. If an agent tends to take the desired actions than before, then the mid-term reward will be larger than the long-term reward. If not so, the long-term reward has a larger value than the mid-term reward.

Such a dynamic behavior is obtained by computing a reward-dependent exploration probability, $\varepsilon(r)$, according to the difference in a Boltzmann distribution of the value before and after learning:

$$k(t,\tau) = \left| \frac{e^{\frac{r_{MT}(t)}{\tau}}}{e^{\frac{r_{MT}(t)}{\tau}} + e^{\frac{r_{LT}(t)}{\tau}}} - \frac{e^{\frac{r_{LT}(t)}{\tau}}}{e^{\frac{r_{MT}(t)}{\tau}} + e^{\frac{r_{LT}(t)}{\tau}}} \right| \tag{6}$$

$$\varepsilon_{t+1}(\Delta r) = \theta \cdot k(t,\tau) + (1-\theta) \cdot \varepsilon_t(\Delta r) \tag{7}$$

Where τ is a positive parameter called Boltzmann temperature and $\theta \in [0,1)$ is a weighted parameter, which decides the effect of reward difference on the selection of action.

Finally, we combine the *ε-greedy* method with Softmax method as RBE-Softmax exploration strategy, we redefine the RBE-Softmax method as follows:

$$\pi(s) = \begin{cases} \text{Softmax action } a \text{ according to (2)} & \text{if } \xi < \varepsilon \\ \arg\max_{a \in A(s)} Q(s,a) & \text{otherwise} \end{cases} \tag{8}$$

Where ε comes from Eq. (7).

The framework of reward-based deep reinforcement learning is shown in Fig. 1, which combines reward-based module with deep reinforcement learning.

The details are described as below:

1) Deep reinforcement learning agent selects a reasonable action according to exploration strategy and current state, environment receives the action and give the agent an instant reward, meanwhile, the state transitions to a new state.
2) The reward-based module utilizes the instant reward

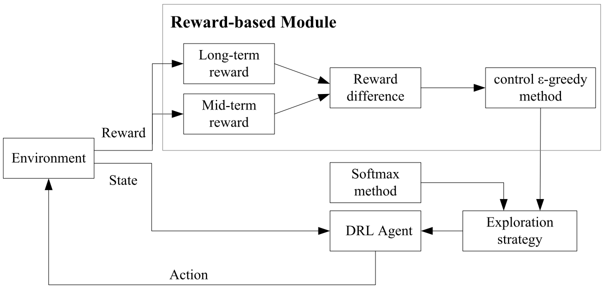

Fig. 1 Framework of reward-based deep reinforcement learning.

from the environment to generate the long-term reward and the mid-term reward, and controls the exploration parameter ε in ε-*greedy* strategy by calculating the value of reward difference.

3) Combining Softmax method and ε-*greedy* strategy as a hybrid exploration strategy, the agent choose an optimal action according to hybrid exploration strategy and current state.

3. Experiments

We test our RBE-Softmax exploration strategy on OpenAI Gym with high-dimensional state space, discrete and continuous action space [9].

3.1 Acrobot-v1 and LunarLanderContinuous-v2

We choose Acrobot-v1 and LunarLanderContinuous-v2 tasks in OpenAI Gym to validate the proposed method. We choose DQN and DDPG as the basic deep reinforcement learning algorithm, use ε-*greedy* method and Softmax method for each algorithms. The parameters of DQN and DDPG are same as the original paper, in the ε-*greedy* method, we set $\varepsilon = 0.1$, in the Softmax method, set $\tau = 5$, and in the RBE-Softmax method, set $\tau_{MT} = \tau_{LT} = 2$, $\tau = 5$. Besides, We call the improved DQN and DDPG algorithms as RB-DQN, RB-DDPG algorithms.

In the discrete Acrobot-v1 task, an under-actuated, two-link robot has to swing itself into an upright position [26]. It consists of two joints of which the first one has a fixed position and only the second one can exert torque. The observation of Acrobot-v1 task consists of two joint angle, θ_1 and θ_2, and their velocities, v_1 and v_2, the action is the torque applied at the second joint. The reward is $r(s, a) := -1 - \cos(\theta_1) - \cos(\theta_1 + \theta_2)$.

'LunarLanderContinuous-v2' is a video game to control a lander to land on the surface of moon safely which is based on a 2D physics engine called 'Box2d'. More details can be found in homepage of OpenAI Gym.

We independently carried out each experiment 20 times respectively. For each running time, the learned policy will be tested 50 episodes respectively by every 100 training episodes to calculate the average scores.

Figures 3 and 4 shows the average score of basic algorithms and improved algorithms on Acrobot-v1 and LunarLanderContinuous-v2.

(a) Acrobot-v1 (b) LunarLanderContinuous-v2

Fig. 2 Example screenshots of environments.

3.2 Comparison and Discussion

Tables 1 and 2 records the average scores and standard deviations after the convergence of the six algorithms on the Acrobot-v1 and LunarLanderContinuous-v2 tasks, and quantitatively analyzes the experimental results. Compared with the DQN based on ε-*greedy* and DQN based on Softmax, the RB-DQN improves the scores by 13.6% and 15.9%. In terms of stability, the standard deviation is reduced by 27.5% and 47.3%. Compared with the DDPG based on ε-*greedy* and DDPG based on Softmax, the RB-DDPG improves the scores by 14.7% and 27.7 %, the stan-

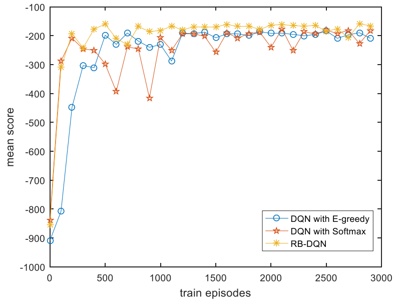

Fig. 3 The average score of DQN with ε-*greedy*, DQN with Softmax and RB-DQN in Acrobot-v1.

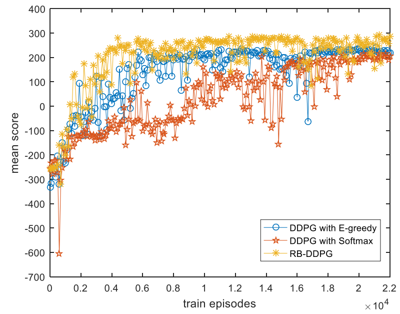

Fig. 4 The average score of DDPG with ε-*greedy*, DDPG with Softmax and RB-DDPG in LunarLanderContinuous-v2.

Table 1 The average score and standard deviation of ε-*greedy*, Softmax and RBE-Softmax method in Acrobot-v1.

Task (AVG,STD)	Random	DQN with ε-*greedy*	DQN with Softmax	RB-DQN
Acrobot-v1	-855.2	(-195.2, 18.2)	(-199.8, 26.4)	(-171.8, 13.9)

Table 2 The average score and standard deviation of ε-*greedy*, Softmax and RBE-Softmax method in LunarLanderContinuous-v2.

Task (AVG,STD)	Random	DDPG with ε-*greedy*	DDPG with Softmax	RB-DDPG
LunarLanderContinuous -v2	-304.9	(210.8, 28.6)	(188.9, 39.3)	(241.2, 19.3)

Table 3 The hypothesis test of Acrobot-v1.

(H, SIGNIFICANCE)	Alternative hypotheses	
	SDQN-ε-greedy<SRB-DQN	SDQN-Softmax<SRB-DQN
Acrobot-v1	(1, 1.1204e-04)	(1, 0.0053)

Table 4 The hypothesis test of LunarLanderContinuous-v2.

(H, SIGNIFICANCE)	Alternative hypotheses	
	SDDPG-ε-greed<SRB-DDPG	SDDPG-Softmax<SRB-DDPG
LunarLanderContinuous-v2	(1, 2.5048e-04)	(1, 4.8328e-08)

dard deviation is reduced by 32.5% and 50.8%. In contrast to basic algorithms, the RB-DQN and RB-DDPG significantly increases experimental scores, besides, the stability of original algorithm has been improved greatly.

Furthermore, several kinds of experiments of deep reinforcement learning algorithm were tested for significance. The significance level was set as $\alpha = 0.05$. As shown in Tables 3 and 4, the test of RB-DQN and RB-DDPG algorithms all rejects original hypothesis and accepts alternative hypothesis. It further shows that RBE-Softmax method indeed has improved experimental performance of basic deep reinforcement learning algorithms.

During the learning process, in very first episode the agents' knowledge is little, the reward received by the agent is unstable, and the reward difference is high in the initial learning stage. The agent needs to increase exploration efforts to accelerate the learning speed. As the agents' knowledge about environment increasing, the reward difference slowly decreases, the agent should increase exploitation. The proposed RBE-Softmax method not only utilizes the characteristic of the agent's learning process to adapt exploration parameters dynamically, but also combines the Softmax strategy to choose exploration action better. On the one hand, the proposed RBE-Softmax method adjusts the exploration parameters more flexibly according to the reward given by the environment, and helps agent to balance the exploration and exploitation better. On the other hand, it combines the Softmax method to choose potential optimal action and improve learning efficiency.

4. Conclusions

This paper proposes a novel exploration strategy called RBE-Softmax method. In contrast to the basic exploration strategy, the proposed method not only takes advantage of the reward difference produced during the learning process to guide the balance between exploration and exploitation, but also combines Softmax method to avoid blind exploration of choice of action. Using the discrete and continuous action control tasks in OpenAI Gym, we have shown that RBE-Softmax method indeed has improvements in the learned policies while combining with basic deep reinforcement learning algorithms.

Acknowledgments

This work was supported by the National Natural Science Fund Projects (61203192), and was also supported by the Natural Science Fund Project in Jiangsu province (BK2011124).

References

[1] R.S. Sutton and A.G. Barto, Reinforcement Learning: An Introduction, MIT Press, 1998.

[2] F. Zhang, J. Leitner, M. Milford, et al., "Towards vision-based deep reinforcement learning for robotic motion control," Computer Science, vol.56, no.2, pp.12–32, 2015.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," Nature, vol.518, no.7540, p.529, 2015.

[4] J. Schulman, S. Levine, P. Moritz, et al., "Trust region policy optimization," ICML, 2015.

[5] M. Tokic and G. Palm, "Value-difference based exploration: Adaptive control between epsilon-greedy and softmax," Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol.7006, pp.335–346, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[6] H. Tang, R. Houthooft, D. Foote, et al., "Exploration: A study of count-based exploration for deep reinforcement learning," vol.64, no.3, pp.65–98, 2016.

[7] R. Houthooft, X. Chen, Y. Duan, et al., "VIME: Variational information maximizing exploration," Neural Information Processing Systems, 2016.

[8] T.P. Lillicrap, J.J. Hunt, A. Pritzel, et al., "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.

[9] G. Brockman, V. Cheung, and L. Pettersson, et al., "OpenAI gym," arXiv preprint arXiv:1606.01540, 2016.