Random Sampling Weights Allocation Update for Deep Reinforcement Learning

Mengzhang Cai[1(✉)], Wengang Zhou[1,2], Qing Li[1], and Houqiang Li[1,2]

[1] CAS Key Laboratory of GIPAS, EEIS Department, University of Science and Technology of China (USTC), Hefei, China
{caimz,liqingya}@mail.ustc.edu.cn
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center, Hefei, China
{zhwg,lihq}@ustc.edu.cn

Abstract. Deep Reinforcement Learning (RL) has achieved great success in many tasks, and the key challenge of Reinforcement Learning now is the inefficient exploration and the unstable training problems brought by high-dimensional input state. Recently, some ensemble works have utilized multiple critics to provide a more specific Q-value and explore more by increasing the diversity of critics. However, these works can not ensure both robust training with effective exploration and thus get limited performance on high-dimensional continuous control tasks. To address this challenge, in this work, we propose Random Sampling Weights Allocation (RSWA), a new critic ensemble framework. Our method introduces the random sampling weights mechanism to increase training robustness and re-allocate the weights according to the Temporal-Difference in every training step to encourage efficient exploration. Our method is compatible with various actor-critic algorithms and can effectively improve the performance of them. We conduct experiments that couple RSWA with various current actor-critic RL algorithms on different OpenAI Gym and DM-Control tasks to verify the effectiveness of this method.

Keywords: Reinforcement learning · Ensemble · Temporal-difference

1 Introduction

With the integration of deep learning, we have witnessed the great success of reinforcement learning (RL) in many complex tasks [7,14] recent years. In RL, an agent is learned to maximize the cumulative rewards through a series of interactions with a dynamic environment. Deep Q-Network algorithm [11] is the first to combine non-linear function approximation with the Q-learning algorithm and introduce experience replay buffer to increase the training stability. In continuous

This work was supported in part by the National Natural Science Foundation of China under Contract 61836011 and U20A20183, and in part by the Youth Innovation Promotion Association CAS under Grant 2018497.

T. Mantoro et al. (Eds.): ICONIP 2021, CCIS 1516, pp. 676–684, 2021.
https://doi.org/10.1007/978-3-030-92307-5_79

control tasks, actor-critic algorithms [4,5,9,10], which combine policy iteration and value iteration, have achieved promising performance. Although these algorithms have been successful applied in many problems, they are still poor in some tasks for two main reasons: one is unstable training leads to serious shock, and the other is inefficient exploration. Recently, ensemble works like Averaged-DQN [2], Bootstrapped DQN [12], Random ensemble mixture (REM) [1] have been proved to be beneficial and efficient to solve these problems for RL. Averaged-DQN utilize averaging previously learned Q-value estimates to reduce the target approximation error variance that leads to stable training and improved performance. In offline RL task, REM uses a random convex combination of multiple Q-value estimation, which improves the stability during RL training. These works indicate that using ensemble techniques can effectively improve performance and stability of deep RL. However, these ensemble works can not ensure robust training with effective exploration and thus get limited improvement in RL algorithms on high-dimensional continuous control tasks.

To solve this problem, in this work, we propose **R**andom **S**ampling **W**eights **A**llocation Update (**RSWA**), a new multi-critic ensemble framework that updates critic Q-value by re-allocating random sampling weights to multiple critics according to Temporal-Difference (TD) [16] calculated on each head in every time step. RSWA introduces a new policy improvement strategy that use randomly parameterized critic to update the actor to ensure robust training, analogous to dropout. Besides, by giving larger weights to critics with bigger TD errors, the critics that can not estimate current state and action well will be more weighted. It means the sensitivity of each critic to different samples are diverse and the diversity of critics will encourage more effective exploration further, like Bootstrapped DQN [12]. Since RSWA allocates random weights to multiple critics, it can both improve the robustness of the updates and the efficiency of exploration.

To verify the effectiveness of our method, we couple RSWA with three state-of-the-art actor-critic algorithms: Twin Delayed Deep Deterministic policy gradient algorithm (TD3) [5], Soft Actor-Critic (SAC) [6], and Proximal Policy Optimization algorithm (PPO) [13]. We conduct experiments of these algorithms on multiple OpenAI Gym tasks (Mujoco) [3] and DeepMind Infrastructure for Physics-Based Simulation tasks (DM-Control) [15]. The experimental results show that our RSWA not only improves the performance of these actor-critic algorithms but also surpasses previous ensemble works.

2 Related Work

2.1 Actor-Critic Reinforcement Learning

Actor-critic method combines both policy gradient and Temporal-Difference learning. It consists of two models: Critic and Actor. Critic updates the value function parameters θ. Value function can be action-value $Q_\theta(s, a)$ or state-value $V_\theta(s)$ depending on the algorithm. And actor updates the policy parameters ϕ for $\pi_\phi(a \mid s)$ in the direction suggested by the critic.

2.2 Ensemble in RL

Averaged-DQN [2] is a simple extension to the DQN algorithm, based on averaging previously learned Q-value estimates, which leads to a more stable training procedure and better performance by reducing approximation error variance in the target values. Bootstrapped DQN [12] use multiple Q-functions updated with different sets of training samples to encourage deep exploration. AUMC [8] initialize critics with random parameters independently to increase the diversity of critics. In offline RL tasks, REM [1] obtains a robust Q-learning by enforcing optimal Bellman consistency on a random convex combination of multiple Q-value estimates. These ensemble works all improve the stability or performance of RL algorithm. However, they do not notice the balance between exploration and robust training. In RSWA, we update critic Q-value by matching different random weights to multiple Q-value estimates according to TD-error calculated in each head that both can ensure efficient exploration and the robustness of training.

3 Method

In this section, we introduce our proposed method: Random Sampling Weights Allocation (RSWA), a critic ensemble framework that achieves both effective exploration and robust training in RL. In RSWA, we use multiple parameterized Q-functions to estimate the Q-value, similar to Average-DQN [2]. Different from previous ensemble works, we randomly generate Q-value weights for multiple critics that the convex combination of multiple Q-value estimates leads to more robust training and stable updates. Besides, RSWA introduces TD-error, to allocate random generated weights to multiple critics. And the diversity of multiple critics brought by TD allocation will encourage deep exploration.

3.1 Random Sampling Weights Allocation

In actor-critic RL algorithm, we get the action tuple (s, a, r, s') at each time step. We then compute the absolute value of TD-error for all critics:

$$\boldsymbol{\delta} = \left\{|\delta_k|\right\}_{k=1}^{K} = \left\{\left|r + \gamma Q_{\bar{\theta}}^{k}\left(s', \tilde{a}'\right) - Q_{\theta}^{k}(s, a)\right|\right\}_{k=1}^{K}, \tilde{a}' \sim \pi_{\bar{\phi}}\left(s'\right), \tag{1}$$

where i represents the index of the critic, $\bar{\phi}$ and $\bar{\theta}$ are the delayed parameters of the actor and the critic, respectively. Since SAC add the entropy bonus to value function, when coupling RSWA with SAC, Eq. (1) is defined as follows:

$$|\delta_k| = \left|r + \gamma\left(Q_{\bar{\theta}}^{k}\left(s', \tilde{a}'\right) + \alpha\mathcal{H}\left(\pi_{\phi}\left(\cdot \mid s'\right)\right)\right) - Q_{\theta}^{k}(s, a)\right|, \quad \tilde{a}' \sim \pi_{\phi}\left(s'\right). \tag{2}$$

We draw K dimensional weights for the critics from the $Uniform(0, 1)$ and normalize them to get a valid categorical distribution at each time step as follows:

$$\bar{\boldsymbol{w}} = \{\bar{w}_k = w'_k / \sum w'_i\}_{k=1}^{K}, \boldsymbol{w'} = \{w'_k \sim \mathrm{U}(0, 1)\}_{i=1}^{K}. \tag{3}$$

Then we resort $\boldsymbol{w}$ according to current TD-errors $\boldsymbol{\delta}$ in non-increasing order and get the re-allocated weights $\boldsymbol{w}$. And $w_1, \cdots, w_K \in (0,1)$ indicate the sensitivity of each citric to current samples. We make the citric be more sensitive to the unfamiliar tuples, which improve the diversity of critics that encourage more effective exploration.

3.2 Policy Evaluation

In RSWA, we use K critics $Q_1, Q_2, ..., Q_K$ to estimate Q-values and minimize the loss of critics:

$$\begin{gathered} y_k \leftarrow r + \gamma Q_{\bar{\theta}}^k \left(s', \pi_{\bar{\phi}}\left(s'\right)\right), \\ J_Q(\theta) = \sum_{k=1}^K J_{Q^k}(\theta) = \sum_{k=1}^K w_k \left(y_k - Q_\theta^k(s,a)\right)^2. \end{gathered} \tag{4}$$

The target Q-value of critic k is y_k, and RSWA tries to minimize the loss remixed by the weights w_k. Since SAC and TD3 both use clipped double Q-learning mechanism, when coupling RSWA with them, the target Q-value is formulated as:

$$y = r + \gamma \min_{i=1,2} \sum_{k=1}^K w_k Q_{\bar{\theta}_i}^k \left(s', \pi_{\bar{\phi}}\left(s'\right)\right). \tag{5}$$

In SAC, the target Q-value function includes entropy regularization item:

$$y = r + \gamma \min_{i=1,2} \left(\sum_{k=1}^K Q_{\bar{\theta}_i}^k \left(s', \tilde{a}'\right) + \alpha \mathcal{H}\left(\pi_\phi\left(\cdot \mid s'\right)\right) \right), \tag{6}$$

where actions $\tilde{a}$ are stochastically sampled from the current policy π_ϕ.

PPO is an on-policy algorithm and use advantage function to measure the relative advantage of action a in state s. When coupling RSWA with PPO, we change the computation of value function as:

$$V_{\bar{\theta}}(s) = \sum_{k=1}^K w_k V_{\bar{\theta}}^k(s). \tag{7}$$

3.3 Policy Improvement

RSWA mixes multiple critics to evaluate the policy and increase the diversity by allocating random sampling weights to different critics according to TD-error. In this way, the algorithm can both maintain the robust training and more efficient exploration. The gradient of the deterministic policy (TD3) is:

$$\nabla_\phi J_\pi(\phi) = \left.\nabla_a Q_\theta(s,a)\right|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s). \tag{8}$$

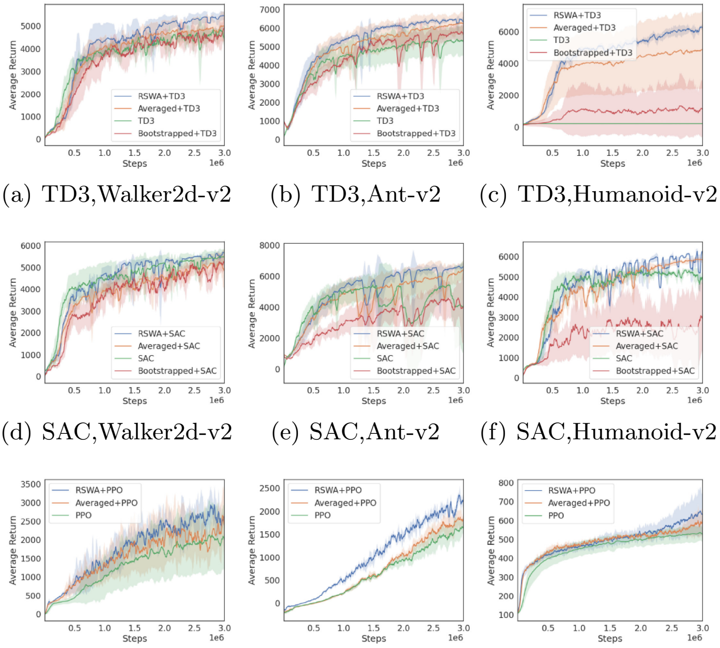
and the gradient of the stochastic policy (SAC) is:

$$\nabla_\phi J_\pi(\phi) = \nabla_\phi \left[\left.\left(Q_\theta(s,a) - \alpha \log \pi_\phi(a \mid s)\right)\right|_{a \sim \pi_\phi(s)} \right], \tag{9}$$

where the critic Q_θ is random sampling weights mixed critics at each training step. The the gradient of Proximal Policy Optimization (PPO) in RSWA is formulated as:

$$\nabla_\phi J_\pi(\phi) = \nabla_\phi \left[\left(\frac{\pi_\phi(a \mid s)}{\pi_{\bar{\phi}}(a \mid s)} A_{\pi_{\bar{\phi}}}(s, a) \right) |_{a \sim \pi_{\bar{\phi}}} \right], \tag{10}$$

where the advantage function $A_{\pi_{\bar{\phi}}}$ is based on the current mixed value function $V_{\bar{\theta}}$.



(a) TD3,Walker2d-v2 (b) TD3,Ant-v2 (c) TD3,Humanoid-v2

(d) SAC,Walker2d-v2 (e) SAC,Ant-v2 (f) SAC,Humanoid-v2

(g) PPO,Walker2d-v2 (h) PPO,Ant-v2 (i) PPO,Humanoid-v2

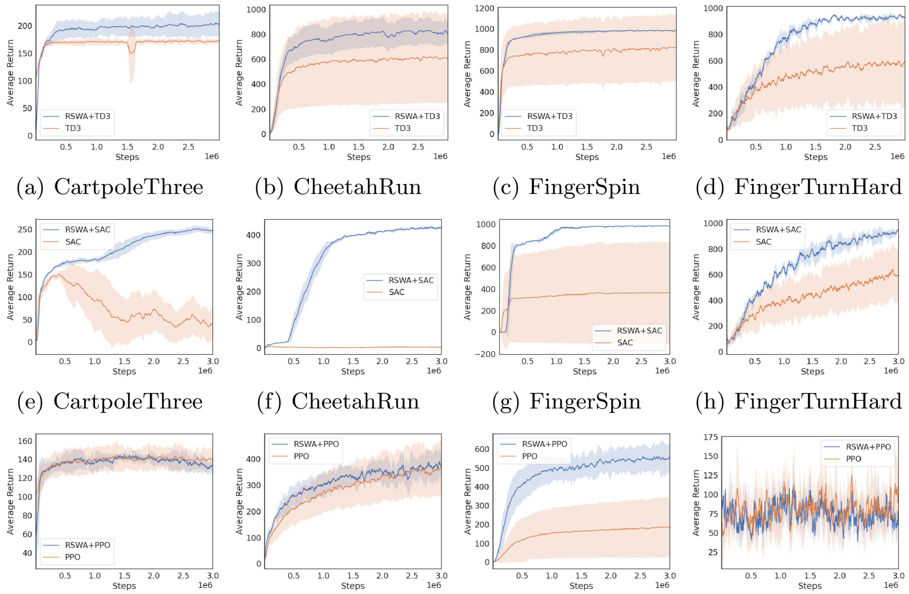
Fig. 1. Learning curves of RSWA coupled with TD3 (first line), SAC (second line), and PPO (third line) compared to Bootstrapped(PPO is not compatible with Bootstrapped), Averaged ensemble method and original algorithms separately.

4 Experiments

In this section, we couple our method RSWA with state-of-the-art actor-critic RL algorithms, like TD3 [5], SAC [6], and PPO [13], to verify the effectiveness of RSWA. We also compare our method with previous ensemble works, such as the Bootstrapped [12] and Averaged ensemble method [2]. Moreover, we further make ablation study to verify the effectiveness of random weights allocation.

4.1 Benchmarks

We separately couple RSWA, Bootstrapped method and Averaged Ensemble method with three state-of-the-art RL algorithms and evaluate these methods on 3 MuJoCo continuous tasks (Walker2d-v2, Ant-v2, and Humanoid-v2) in OpenAI Gym [3]. Besides, we further implement our method on 12 DM-Control tasks [15] to verify the improvement to original RL algorithms. The performance of algorithms on each environment is demonstrated by plotting the mean cumulative reward. For results plots, the solid lines represent the mean cumulative rewards and the shaded regions represent the standard deviation of the average evaluation over 4 different random seeds.



(a) CartpoleThree (b) CheetahRun (c) FingerSpin (d) FingerTurnHard

(e) CartpoleThree (f) CheetahRun (g) FingerSpin (h) FingerTurnHard

(i) CartpoleThree (j) CheetahRun (k) FingerSpin (l) FingerTurnHard

Fig. 2. Learning curves of TD3, SAC, and PPO with and without RSWA on four DM-Control (CartpoleThree, CheetahRun, FingerSpin and FingerTurnHard) tasks over 3 million time steps. Other tasks in DM-control are in appendix.

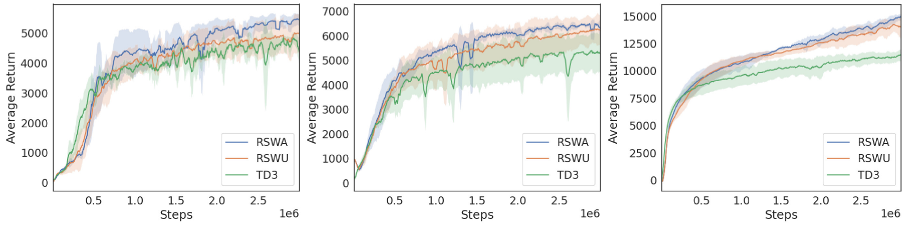
4.2 Implementation Details

For existing RL algorithms (TD3, SAC, and PPO), value-network and policy-network are implemented with MLP consist of two hidden layers and learning rate is $3e^{-4}$. The batch size is 256, the replay buffer size is 1×10^6, and the discount factor is 0.99 during training process. For Random Sampling Weights Allocation, we sample the weights from the $Uniform$ $(0, 1)$ and the number of critics is set to 100 ($K = 100$). While in Bootstrapped Method and averaged ensemble method the number of critics is set to 10 ($K = 10$). For Bootstrapped Method, we sample K-dimensional binary masks from Bernoulli distribution with fixed parameter that denotes the probability of allocating the samples to the critics for Bootstrapped Method. And in Bernoulli distribution, we set $p = 0.5$. For fair comparison, other hyper-parameters in Bootstrapped and Averaged method is the same with our RSWA.

4.3 Results

The results of total average return during training for RSWA, Bootstrapped and Averaged ensemble method on PPO, TD3 and SAC in Mujoco environments are shown in Fig. 1. For all Mujoco tasks, our method RSWA consistently improves the performance of three algorithms. And the improvement in PPO and TD3 is larger than in SAC. It should be noting that the performance of Bootstrapped method in SAC is also worse. In SAC, the introducing of entropy regularization playing the same role as diverse multiple critics limits the performance of method that encourage to explore more. However, our method in Mujoco tasks still perform better than original SAC algorithm, which verify performance improvement in RSWA not only from the effective exploration but also from the robustness brought by random sampling weights in each time step. Besides, Fig. 2 shows the performance of RSWA on 12 different DM-Control tasks. RSWA improve the performance of three actor-critic RL algorithms on almost all tasks. These results demonstrate that RSWA can improve current state-of-the-art actor-critic RL algorithms and work better than other existing ensemble methods. It verifies that our method can achieve both exploration and robust update efficient by reasonably allocating random sampling weights.

4.4 Ablation Study



(a) Walker2d-v2 (b) Ant-v2 (c) HalfCheetah-v2

Fig. 3. Performance comparison of RSWA, RSWU and TD3 on Walker2d-v2, HalfCheetah-v2 and Ant-v2 to verify effectiveness of random sampling weights and TD allocation.

We conduct ablation study to further examine which particular component of RSWA is essential for the performance. We set TD3 as the baseline algorithm to conduct the following experiments.

As discussed in the previous section, the improvement of performance in RSWA is mainly contributed by robust training and effective exploration, which is separately brought by random weighs sampling and the allocation according to TD-error. We therefore run the experiments that remove the TD allocation from RSWA called Random sampling weights Update (RSWU). The learning curve compared to RSWA and baseline TD3 are shown in Fig. 3. On Ant-v2 and HalfCheetah-v2, RSWU is better than baseline TD3 while it is worse than RSWA. The results verify our ideas that the random weighs sampling and the TD allocation are complementary to each other and can both improve the performance of actor-critic RL algorithms.

5 Conclusion

In this work, we propose the RSWA framework, a multi-critic ensemble mechanism that is compatible with various current state-of-the-art actor-critic RL algorithms. RSWA achieves robust training with effective exploration through the allocation of random sampling weights according to TD-errors. The experiments on the OpenAI Gym and DM-Control benchmarks demonstrate that our method can significantly improve the performance of RL algorithms, such as TD3, SAC, and PPO. And we further conduct ablation study to verify that the random weighs sampling and TD allocation are Complementary. Moreover, our method is time efficient and compatible to various actor-critic RL algorithms.

References

1. Agarwal, R., Schuurmans, D., Norouzi, M.: An optimistic perspective on offline reinforcement learning. In: IMCL, pp. 104–114. PMLR (2020)
2. Anschel, O., Baram, N., Shimkin, N.: Averaged-DQN: Variance reduction and stabilization for deep reinforcement learning. In: ICML, pp. 176–185. PMLR (2017)
3. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: Openai gym. arXiv preprint arXiv:1606.01540 (2016)
4. Ciosek, K., Vuong, Q., Loftin, R., Hofmann, K.: Better exploration with optimistic actor-critic. arXiv preprint arXiv:1910.12807 (2019)
5. Fujimoto, S., Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods. In: ICML, pp. 1587–1596. PMLR (2018)
6. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: ICML, pp. 1861–1870. PMLR (2018)
7. Li, J., Koyamada, S., et al.: Suphx: Mastering mahjong with deep reinforcement learning. arXiv preprint arXiv:2003.13590 (2020)
8. Li, Q., Zhou, W., Zhou, Y., Li, H.: Attentive update of multi-critic for deep reinforcement learning. In: ICME, pp. 1–6. IEEE (2021)
9. Lillicrap, T.P., Hunt, J.J., et al.: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015)
10. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. In: ICML, pp. 1928–1937. PMLR (2016)

11. Mnih, V., Kavukcuoglu, K., et al.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
12. Osband, I., Blundell, C., Pritzel, A., Van Roy, B.: Deep exploration via bootstrapped DQN. arXiv preprint arXiv:1602.04621 (2016)
13. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
14. Silver, D., Huang, A., et al.: Mastering the game of go with deep neural networks and tree search. Nature **529**(7587), 484–489 (2016)
15. Tassa, Y., Tunyasuvunakool, S., et al.: dm_control: software and tasks for continuous control. arXiv preprint arXiv:2006.12983 (2020)
16. Tesauro, G., et al.: Temporal difference learning and TD-Gammon. Commun. ACM **38**(3), 58–68 (1995)