Duality in Deep Reinforcement Learning--Theory

Jie Bai[a], Jianfei Li[b], Zihao Luo[c], Yaobing Wang[d], Li Liu[e]

Peking University, Center of Excellence for Intelligent Robotics, Beijing Institute of Spacecraft System Engineering, Beijing, China

[a]baijierobot@163.com, [b]lijianfei_hit@foxmail.com, [c]438940765@qq.com, [d]iamwyb@163.com, [e]m55852@126.com

Keywords: Deep Reinforcement Learning, Duality, Prioritized Sampling, Prioritized Learning

Abstract: More and deeper reinforcement learning algorithms have been proposed and demonstrated on a series of decision-making domains. However, little research has been hammered at algorithm extraction. With duality in deep reinforcement learning substantially summarized, we propose a conceptually simple framework for deep reinforcement learning based on duality. Then, we propose the dual method of prioritized sampling: prioritized learning. Finally, we give the formula and analysis for the duality with priority. The algorithm implementation and experiment will be put on Part II-Implementation.

1. Introduction

Deep Reinforcement Learning (DRL) has been demonstrated on a series of challenge domains, from games [1-2] to robotic control [3]. There are basically two types: model-free algorithms and model-based algorithms, except the problems of sparse rewards, such as HER [4] and the problems hard to define rewards HRL [5]. Model-free algorithms are focused on how to achieve data efficiency, such as asynchronous methods, priority methods, while model-based algorithms commit themselves to data generation from expert demonstrations.

Synoptically, support for model-free algorithms depend on two aspects: Actor-Critic algorithms and off-policy methods. Actor-Critic algorithms are available to distributed computing, where enable the models to decouple actors from learners. While off-policy methods can make the utmost of replay memory. However, replay memory in DQN have been based on the uniform distribution, which is not substantial for data efficiency. Prioritized DQN [6] assigns a greater sampling weight for the state of high learning efficiency. Despite this, this is not sufficient.

Machine learning is the problem of interdisciplinary integration, including information theory, optimization algorithm, sampling theory, etc. [7], which is one of the inspirations for this work. Here we introduce straightly the conclusion, and more details can be seen in Section III. What we refer is Duality. It has been demonstrated that the learners can use temporal-difference (TD) error to give the actors sampling priority [8]. According to the duality between actors and learners in Actor-Critic algorithms, why not use the actors to give the learners learning priority?

This work summarizes previous deep reinforcement learning algorithms and obtains some general conclusions. Our contribution is to present the Duality in deep reinforcement learning. Based on this, we propose an approach for deep reinforcement learning throughout combining priority, to improve the efficiency of data.

2. Background

In this section, we formulate reinforcement learning problems as a standard Markov Decision Process (MDP) [9], and introduce the necessary algorithmic foundations which are aimed to guide our work.

2.1 Reinforcement Learning

We consider a reinforcement learning process where an agent interacts with an environment in a

discrete timesteps. At state s_t in timestep t, the agent selects action a_t according to the stochastic distribution $\pi(a_t|s_t)$ or a deterministic mapping $a_t = \pi(s_t)$, transitions to new state s_{t+1} according to the dynamics $P(s_{t+1}|s_t, a_t)$, and receives a reward $R(s_t, a_t)$. Here, the return from state s_t can be defined as the sum of γ-discounted future reward $G_t = \sum_{i=t}^{T} \gamma^{(i-t)} R(s_t, a_t)$, where γ is the discounting factor, and $T = \infty$ in an infinite-horizon or $T < \infty$ in a finite-horizon. The goal of reinforcement learning is to learn a policy which the expected state return $J = \mathbb{E}_{\pi}[G_{t=0}]$ from an initial distribution $p(s_0)$ is maximized over the agent's trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots)$.

2.2 Actor-Critic Framework

The approach to reinforcement learning problems can be divided into two alternative methods. The first one called value function approaches (Critic-only), is an estimate of the expected future reward based on the policy π, where the value-action function Q^{π} is defined as:

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[\sum_{t=0}^{+\infty} \gamma^t R(s_t, a_t)] \quad (1)$$

The policy is implicitly derived from Q^{π} as $\pi(s) = \text{argmax}_{a \in \mathcal{A}} Q^{\pi}(s, a)$. The other available method is *policy search* (Actor-only). In the policy search methods, policies are represented by a variety of approaches and can be directly optimized to maximize the cumulative reward, given:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta(\tau)}[r(\tau)] = \int_{\tau \sim \pi_\theta(\tau)} \pi_\theta(\tau) r(\tau) \, d\tau \quad (2)$$

where $r(\tau)$ represents the total reward of the trajectory. Actor-Critic framework (Peters et al., 2008) consists of value function approaches and policy search methods, where the Critic estimates the value function based on the temporal difference (TD) learning, while the Actor updates the policy parameters according to the learned value function.

2.3 Priority

The Priority is used for experience replay. Experience replay has been demonstrated on improving data efficiency in reinforcement learning, in that experience replay can disrupt the correlation of samples and allow the agent to learn from off-policy methods. Prioritized experience replay extends classic experience replay according to importance sampling techniques. The experience of the agent is not of equal importance to the learning. The idea of prioritized experience replay is to break up uniform sampling and give bigger sampling weight to the state with high learning efficiency, to increase the sampling probability of valuable samples. Ape-X [8] also extends priority based on parallel computation. It does not wait for the learner to update priorities, which can occur to a myopic focus on the most recent data. In contrast, it evaluates multi-actors' local copies of the policy, and computes suitable priorities for new transitions online.

3. Related Works

3.1 Duality Summary

Table 1. Duality in deep reinforcement learning

Duality	The Former	The Latter
Actor vs. Critic	(1) Bias estimate (2) Low variance	(1) Unbiased estimate (2) Large variance
On-policy vs. Off-policy	(1) Data inefficiency (2) Easy to convergence	(1) Data efficiency (2) Easy to divergence
Synchronous vs. Asynchronous	No unified argument	
Exploration vs. Exploitation	(1) Policies with wide coverage (2) Hard to convergence	(1) Trap into the sub-optimal solution (2) Easy to convergence

In the early development of Linear Programming, the most important discovery is the dual problem. Every linear programming problem is accompanied by another linear programming

problem, called the dual problem [10]. The idea is not limited to optimization problems, here we just refer Duality. In this section, research on duality in deep reinforcement learning has been substantially investigated and summarized in Table 1.

3.2 Duality in Actor vs. Critic

One of the core problems in machine learning is trade-off bias and variance, which consists of duality in Actor-Critic architecture. For policy-based models (Actor-only), the estimation of gradient produces a large variance, which cause a slow learning speed. In contrast, for value function-based models (Critic-only), TD error learning method reduces the variance, but it is difficult to apply for the continuous problem due to the large amount of calculation. Actor-Critic algorithms combine the fast learning of value function-based models and easy to convergence of policy-based models [11]. The Critic in Actor-Critic architecture estimates the value function according to the TD error learning, and the Actor in Actor-Critic architecture dynamically updates the policy parameters according to the learned value function. However, this method results in a bigger bias. In order to good trade-off bias and variance, multi-step learning [12] is imported on TD error learning.

3.3 Duality in On-policy vs. Off-policy

Under the Actor-Critic architecture, two policies need to be followed: the policy of data generation is called behavior policy, and the policy of learning from data is called target policy. If the two policies are the same policy, they are called on-policy, otherwise off-policy. Despite slow convergence of off-policy methods, their applications are more powerful and general due that all behaviors can be covered and their experiences can be available to self-generated or external data. Its dual method is on-policy, where its straightforward policy and easy to converge show its superiority. However, it easily traps itself into local optimum. As for data efficiency, we demonstrated that, mature off-policy algorithms, such as Deep Deterministic Policy Gradient (DDPG) [11] and Normalized Advantage Function (NAF) [13] can achieve a well-used data efficiency. In contrast, some general-purpose on-policy DRL algorithms, such as Trust Region Policy Optimization (TRPO) [14] and Asynchronous Advantage Actor-Critic (A3C) [12], require new samples to be collected for each learning step on the policy, which occurs to the data inefficiency.

3.4 Duality in Synchronous vs. Asynchronous

The idea of distributed system has recently been induced into deep reinforcement learning, which benefits from the parallelism architecture DistBelief [15] and the learning architecture (Actor-Critic). Parallel sampling from the interaction with the environment and parallel training are important features of asynchronous algorithms. Because on-policy methods take more time for data generation. After samples collected, each thread completes the training independently and gets the parameters update, then completes the global parameter synchronization asynchronously. However, this does not explain the advantages of the asynchronous approach. The Advantage Actor-Critic (A2C) algorithm, mentioned in OpenAI's official website, centralizes decision-making and training tasks into one place (GPU), while other processes (multi-core CPUs) are only responsible for interaction with the environment. The experiment shows that A2C is superior to A3C in implementation and final performance. At present, there is no unified demonstration on the duality of asynchronization and synchronization.

3.5 Duality in Exploration vs. Exploitation

Trade-off exploration and exploitation becomes a dual problem. Exploration-only leads to the insufficient attempt of the optimal policy and the inaccurate value function, which ultimately makes the agent deprive the chance to select the optimal policy. In contrast, exploitation-only enables the agent to trap itself into the sub-optimal solution. Research on exploration and exploitation has always been decoupled. Some of the art-of-the-state exploration algorithms include Noisy networks [16], Parameter space noise [17], which represent the model as neural networks and add the noise to parameter space, while utilize parameter perturbations for more efficient exploration. On the contrary, we can think about the exploitation, such as Deep Q-learning from Demonstrations (DQfD) [18],

which leverages even very small amounts of demonstration data to massively accelerate learning.

4. Our Method

4.1 Terminology

In the previous section, duality in deep reinforcement learning has been summarized. In this section, we import the duality in robot control to define the necessary terminology for dual system. Then we formulate the Duality in deep reinforcement learning (DDRL) according to Entropy.

In the robot force control field, Hogan has extended a system description method called Bond Graph [19]. The bond graph shows the transfer of energy between parts of the system. It defines the physical quantity such as force and voltage as the effort, while speed and current as the flow. If the effort multiples with the flow, it produces the instantaneous power, given by

$$\text{power} = \text{effort} \times \text{flow} \tag{3}$$

Also, some systems input the effort (such as force) and output the flow (such as speed) called admittance systems, while others input the flow (such as speed) and output the effort (such as force), called impedance systems. Some details can be referred to Hogan's research.

We demonstrate that Bond graph defines the Duality in the robot's interaction with the environment, where the robot can be also regarded as the agent. The effort produces the energy while the flow consumes the energy. The system with the effort input and the flow output is called the admittance system, and vice (duality) called the impedance system.

Our work is to extend the idea into the DRL field. Bond graph can describe the transfer of energy flow in mechanical or electrical systems, while we describe the transfer of data flow in DRL systems. Here we just consider the Actor-Critic architecture for the Duality.

We define the part for data generation as the actor (effort) and for the data consumption (learning) as the learner (flow). The actor generates data and supplies sample experiences to the learner, which can be referred as the acting (also called policy evaluation, forward system). While the learner consumes data and provides the actor with update parameters, which be considered as the learning (policy improvement, feedback system). To unify mechanical (electrical) systems with DRL systems, we define the process that the energy (data) from the effort to the flow calls on-flow, vice called off-flow. Therefore, we call the acting as on-flow and the learning as off-flow. The dual terminology can be referred in Table 2.

Table 2. Duality in mechanical (electrical) systems vs. DRL systems

System	Mechanical	DRL
Drive	Energy flow	Data flow
Input vs. Output	Effort vs. Flow	Actor vs. Learner
On-flow vs. Off-flow	Admittance vs. Impedance	Acting vs. Learning

4.2 Duality with Entropy

In the robot force control field, the effort multiples with the flow and produces the instantaneous power. Therefore, it can reveal the relationship of energy storage and conversion. However, energy cannot be obtained by multiplying the effort times the flow in data flow systems. Here we induce Entropy and introduce the duality with entropy in DRL systems.

In fact, we point the Kullback–Leibler (KL) divergence, which describes the difference between two probability distributions p and q. The formula of KL divergence is given:

$$\mathrm{KL}(p||q) = \mathbb{E}_{x\sim p}\left[\log\frac{p(x)}{q(x)}\right] \tag{4}$$

KL divergence is a good measure of the distance between two probability distributions. The closer the two distributions are, the smaller the KL divergence is. It can be proven that the KL divergence is non-negative.

In TRPO, the goal is to find the new policy to keep the return function monotonous nondecreasing. A natural idea is to decompose the expected return based on the new policy into the expected return based on the old policy and the other term. In this way, as long as the other term is non-negative, the expected return of the new policy is guaranteed to remain monotonous nondecreasing. The formula is proposed by Sham, given by

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0, a_0, \ldots \sim \tilde{\pi}}[\sum_{t=0}^{+\infty} \gamma^t A_\pi(s_t, a_t)] \tag{5}$$

where $\eta(\pi)$ and $\eta(\tilde{\pi})$ means the expected return by the old policy π and the new policy $\tilde{\pi}$, respectively. The second part on the right side of the equation can be proved to be non-negative, which ensures the monotonous nondecreasing expected return of the new policy [20]. Then, the importance sampling is adopted to improve the return function near the old policy. Therefore, the constraint of the learning step length can be given by

$$\mathrm{KL}_{\max}(\pi||\tilde{\pi}) \leq \delta \tag{6}$$

Another example can also describe duality based on KL in Guide Policy Search (GPS) [21]. By optimizing trajectories in tandem with the policy, GPS methods combine the flexibility of trajectory optimization with the generality of policy search. In other word, GPS means the acting generates data by interacting with the control phase, and the learning learns from the data generated from the control phase by the monitor phase. We can give the formula of loss function:

$$\min_{\theta,q} \mathrm{KL}(q(\tau)||\,\rho(\tau)) \tag{7}$$

where we denoted q(τ) as the policy on distributions over good trajectories, and ρ(τ) as the expected sum return by the policy $\pi_\theta(\tau)$. When alternating policy and trajectory optimization, q(τ) and $\pi_\theta(\tau)$ are gradually brought into agreement.

No matter what duality in on-policy and off-policy (TRPO) or duality in on-flow and off-flow (GPS), KL divergence is suitable for measuring the difference between two probability distributions. We denote that probability distributions p and q as the probability distribution of data generation (Target distribution) and the probability of data learning (Approximate distribution). Then, we define Momentum projection (M-projection) as

$$q = \arg\min_q \mathrm{KL}(p||q) \tag{8-a}$$

where it forces that the approximate distribution q has a high probability at the state where the target distribution p has a high probability. While, we also define Information projection (I-projection) as

$$q = \arg\min_q \mathrm{KL}(q||p) \tag{8-b}$$

where it forces that the approximate distribution q is zero at the state where the target distribution p is zeros. M-projection attempts to cover all the trajectories, while I-projection attempts to restrain the bad trajectories. The two consist of Duality in DRL.

4.3 Duality Based on Priority

In this part, we start from a brief introduction of the Prioritized DQN. In off-policy methods, we can take advantage of experience replay. We demonstrate that the experience of the agent is not of equal importance to the learning. The idea of prioritized experience replay is to break up uniform sampling and give bigger sampling weight to the state with high learning efficiency, to increase the sampling probability of valuable samples. In other word, the worse performance the agent interacts with the environment (the bigger TD error), the higher weight the learning should distribute while the higher sampling probability the acting should distribute, hence there it contributes to a better learning efficiency, and vice versa.

We assume that the TD error at the sample i is δ_i, so the sampling probability at this sample i is $P(i) = p_i^\alpha / \sum_k p_k^\alpha$, where p_i^α depends on δ_i. However, when we sample by the probability

distribution of the prioritized replay, the estimate of the value function is a biased estimate. In order to rectify the deviation, we can add an importance sampling weight $\omega_i = \left(1/N \cdot 1/P(i)\right)^{\beta}$ to each sample before learning, which makes it unbiased. The two steps consist of duality, where priority increases the probability of valuable samples (efficiency) and importance sampling weight is guaranteed to the convergence of the model (stability).

It has been introduced above that the learners can use TD error to give the actors sampling priority. According to the duality in Actor-Critic algorithms, why not use the actors to give the learners learning priority? Here we propose the prioritized learning as the duality of the prioritized sampling, where prioritized sampling takes advantage of the learning to optimize the sampling process of the acting (on-flow) while prioritized learning makes use of the acting to optimize the learning process of the learning (off-flow).

The goal of the acting is to provide valuable samples (high TD error) for the learners, enabling the learners to update parameters quickly. While the goal of the learning is to provide parameter updates for the actors, enabling the actors to generate the higher reward (or return). So, the reward should be the key of prioritized sampling. To simplify the formula, we directly adopt the method of prioritized sampling to realize prioritized learning (or other methods). Here we only need to replace the TD error in prioritized sampling with the reward in prioritized learning. Data flow for duality based on priority has been shown in Figure 1.

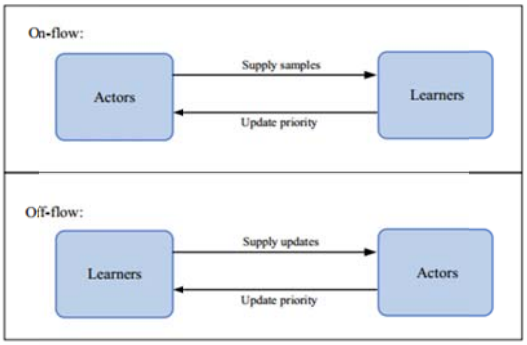

Fig. 1. Data flow in Multi-Actors-Multi-Learners (MAML) systems based on priority

4.4 Formula and Analysis

In this section, we formulize the duality with priority. According to the KL divergence in DDRL, we propose the loss function for DDRL based on priority. Finally, we give a simple convergence analysis.

In Actor-Critic algorithms, we usually consider using the TD-error to estimate the return on the trajectory, where Equation 9-a (Single-step Learning) or Equation 9-b (Multi-steps Learning) is used to replace the return r(τ) on the trajectory of Equation 2.

$$\delta_t = r(s_t, a_t) + \gamma v(s_{t+1}) - v(s_t) \qquad (9\text{-a})$$

$$\delta_t = \sum_{i=0}^{n-1} \gamma^i r_{t+i} + \gamma^n v(s_{t+n}) - v(s_t) \qquad (9\text{-b})$$

We denote the prioritized sampling function as f (We're not going to derive the complicated function here, because we know that prioritized sampling function and absolute TD error are positively related), so the probability distribution of prioritized sampling can be indicated as $p = f_1(|\delta|)$. Similarly, we give the probability distribution of priority learning as $q = f_2(|\Delta|)$. We need to explain that δ denotes absolute TD error, Δ denotes the return error, given by

$$\Delta_t = \overline{r_t} - r(s_t, a_t) \qquad (10)$$

Where $\overline{r_t} = \sum_i r(s_i, a_i)\ /t$ denotes the average cumulative return. According to duality in DRL, function f_1 and f_2 are set to the same f. Here, we refer prioritized learning as a sampling process for the gradient. Now, we can use the KL divergence formula to measure the distance between the two sampling policies.

In the optimal policy, we can find that TD-error converges to zero. Again, the return error

converges to zeros. At this point, we can think $p = q = 0$ (No exist priority). In other word, when the agent interacts by the optimal policy, $\mathrm{KL}(p||q) = 0$. Therefore, we can consider $\mathrm{KL}(p||q)$ as loss function or a regular term in loss function. In addition, we can induce the regular term $-\mathrm{H}(p)$ into loss function for sufficient exploration, to avoid plunging into local optimum.

5. Summary

In this work, we first induce Duality and analyze some dual problems in deep reinforcement learning (DRL). Then we introduce the application of KL divergence in DRL and refer it as Duality in deep reinforcement learning (DDRL) with entropy. According to DDRL, we propose the dual problem of prioritized sampling: prioritized learning. Finally, we give the formula and simple analysis of priority based DDRL. In addition, as for the implementation of DDRL, especially the implementation of DDRL based on dual priority, we will put it into Part II-Implementation.

References

[1] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning. In *NIPS Deep Learning Workshop*, 2013.

[2] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[3] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms. *arXiv preprint arXiv*:1707.06347, 2017.

[4] Andrychowicz M, Wolski F, Ray A, et al. Hindsight Experience Replay. *arXiv preprint arXiv*:1707.01495, 2017.

[5] Kulkarni T D, Narasimhan K R, Saeedi A, et al. Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation. *arXiv preprint arXiv*:1604.06057, 2016.

[6] Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay. *arXiv preprint arXiv*:1511.05952, 2015.

[7] Nasrabadi N M. Pattern recognition and machine learning. *Journal of electronic imaging*, 2007, 16(4): 049901.

[8] Horgan D, Quan J, Budden D, et al. Distributed prioritized experience replay. *arXiv preprint arXiv*:1803.00933, 2018.

[9] Sutton R S, Barto A G. Reinforcement learning: An introduction. *MIT press*, 1998.

[10] Hillier F S. Introduction to operations research. *Tata McGraw-Hill Education*, 2012.

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

[12] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. Nature, 2015.

[13] Gu S, Lillicrap T, Sutskever I, et al. Continuous deep q-learning with model-based acceleration. In *International Conference on Machine Learning*, 2016.

[14] Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization. Trust region policy optimization. In *International Conference on Machine Learning*, 2015.

[15] Dean J, Corrado G, Monga R, et al. Large scale distributed deep networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, 2012.

[16] Fortunato M, Azar M G, Piot B, et al. Noisy networks for exploration. *arXiv preprint arXiv:*1706.10295, 2017.

[17] Plappert M, Houthooft R, Dhariwal P, et al. Parameter space noise for exploration. *arXiv preprint arXiv:*1706.01905, 2017.

[18] Hester T, Vecerik M, Pietquin O, et al. Deep Q-learning from Demonstrations. *arXiv preprint arXiv:*1704.03732, 2017.

[19] Hogan N. Impedance control-An approach to manipulation. I-Theory. II-Implementation. III-Applications. *ASME Transactions Journal of Dynamic Systems and Measurement Control B*, 1985, 107: 1-24.

[20] Kakade, Sham. A natural policy gradient. *In Advances in Neural Information Processing Systems*, 2002, pp. 1057–1063.

[21] Levine S, Koltun V. Guided policy search. In Proceeding of International Conference on Machine Learning. 2013: 1-9.