Adversarial Policy Training against Deep Reinforcement Learning

Chen Gong

12 May 2021

1 Summary

Currently, Deep Reinforcement Learning (DRL) has been widely applied in various fields, and shows superior performance on video games, such as Atari 2600 games [4], Go [3], StarCraft II [5] and so on. Besides, combining DRL with adversarial attacks, an adversary could force a well-trained agent to behave abnormally by perturbing the input of agent' s policy network or training an adversarial agent to exploit the weakness of the victim [7].

In this paper, authors extends a reinforcement learning algorithm (Proximal Policy Optimization, PPO) to guide the training of the adversarial agent in the two-agent competitive game setting. The main contributions are concluded as follows,

1. **Strong Practicality:** In many recent research works, they assume an attack has the privilege to manipulate the environment freely, and explicit knowledge to the opponent agent policy [8]. However, these attacks are not practical in the real world. Although the new attack against RL is designed without manipulation of environment[5], this newly method usually shows a relatively low success rate of failing the opponent agent. This is because the attack is a simple application of the PPO, and PPO does not train an agent for exploiting the weakness of the opponent agent.

   The attack mechanism is the first work that can effectively exploit the weakness of victim agents without manipulation of the environment, explicit knowledge of the opponent policy network and state-transition model. So this attack mechanism is practical, and trains an adversarial agent in an effective and efficient fashion.

2. **Explainable AI Technique:** To facilitate the search of the adversarial policy network, authors adjust the weight of the action deviation based on by how much the victim agent pays attention to the action of the action of the adversarial. The weight is estimated by an explainable AI technique.

3. **SOTA Performance:** Authors conduct experiments using MuJoCo and roboschool Pong to evaluation their attack. Compared with the SOTA technique [2], the adversarial agent exhibits a much stronger capability in exploiting the weakness of victim agents (an average of 60% vs. 50% winning rate for MuJoCo game and 100% vs. 90% for the Pong game). The attack method in this paper can conduct an adversarial agent with a 50% winning rate in fewer sampling (11 million vs. 20 million iterations for MuJoCo game, and 1.0 million vs. 1.3 million iterations for Pong game). Besides, adversarial attack shows less variation in the training process.

This paper extends a SOTA reinforcement learning algorithm to guide the training of the adversarial agent in the two-agent competitive game setting. However, there are also some disadvantages: The attack method can only be applied to the two-agent competitive game setting. The analysis of the algorithm designed is presented in Appendix.

2 Potential directions to explore

1. **Distributional Reinforcement Learning for Exploring Weakness of Opponent:** The goal of conventional reinforcement learning is to maximize the Q values, which is the expectation of the accumulative returns, denoted by Z. However, the distribution of Z is critical to the robustness of agents, indicating under what situations (states) agents perform well or poorly. The states with low distributional values can be vulnerabilities of agents. Distributional reinforcement learning [1] aims to learn value distribution rather than the expected values. There are several advantages: (1). it applies more decision information (i.e., for some risk-sensitive tasks, we may choose actions that have a lower variance or a better worst-case scenario than actions that have a higher mean); and (2). it alleviates the problem of reward sparsity, and sparse rewards are more likely to be retained during propagation.

   On the one hand, distributional reinforcement learning is a way to train more robust agents. On the other hand, it can also be used as adversarial attacks to conventional RL agents by exploring the opponent's worst-case scenarios.

2. **Meta Reinforcement Learning for Stronger Transferability:** The paper also mentions some future work to explore the transferability: whether an adversarial policy network trained against one opponent agent could also be used to defeat the other agents. We can take one step further: adopting meta reinforcement learning (Meta RL) to enhance the transferability of adversarial agents. Meta-RL applies meta-learning ideas to reinforcement learning tasks. A good meta-learning model is expected to generalize to new tasks or new environments that have never been encountered during training. The meta RL includes two sections, meta-training and meta-adaptation. Meta-learning process learns knowledge from the past Markov decision process. Meta adaptation is that how to quickly change the network to adapt to a new task [6]. In our problem, meta RL algorithms can make an adversarial agent attack different opponent agents in the same environment, or the adversarial agent only requires simple retraining to be able to defeat an opponent agent that it never encounters.

3. **Adversarial Attack for Multi-Agent Reinforcement Learning:** We can extend the attack mechanism to multi-agent environments, where multiple participants collaborate or compete with each other. When multiple agents interact with the environment at the same time, the system becomes a multi-agent system. Each agent still follows the goal of reinforcement learning, that is, to maximize the cumulative return. The transition of the state of the environment is related to the joint actions of all agents so that the influence of the joint actions should be considered. Can we extend the multi-agent RL algorithms to study the attack methods in multi-agent environments?

References

[1] Marc G. Bellemare, Will Dabney, and Rémi Munos. A Distributional Perspective on Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 449–458, Sydney, NSW, Australia, 2017. JMLR.org.

[2] Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. *arXiv preprint arXiv:1905.10615*, 2019.

[3] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[4] David Silver, Aja Huang, Chris J. Maddison, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484, 2016. Publisher: Nature Publishing Group.

[5] Oriol Vinyals, Igor Babuschkin, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575:350–354, 11 2019.

[6] Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

[7] Xian Wu, Wenbo Guo, Hua Wei, and Xinyu Xing. Adversarial policy training against deep reinforcement learning. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[8] Chaowei Xiao, Xinlei Pan, Warren He, Jian Peng, Mingjie Sun, Jinfeng Yi, Mingyan Liu, Bo Li, and Dawn Song. Characterizing attacks on deep reinforcement learning. *arXiv preprint arXiv:1907.09470*, 2019.

Appendix

In this section, I introduce the details in algorithm. This method extend the PPO loss function L_{PPO} a new loss term

$$L_{\text{ad}} = \text{maximize}_{\theta} \left(\left\| \hat{a}_v^{(t+1)} - a_v^{(t+1)} \right\|_1 - \left\| \hat{o}_v^{(t+1)} - o_v^{(t+1)} \right\|_1 \right), \tag{1}$$

where θ is parameters in adversarial agent π_α. $\hat{o}_v^{(t+1)}$ and $\hat{a}_v^{(t+1)}$ are the different observation and action taken by the opponent agent if, at the time step t, the adversarial agent takes an action different from the ones indicated by the trajectory rollouts. The key idea of this term can be concluded as, when maximizing the deviation of an opponent action, we need to ensure the minimal change of the environment observation.

Neither the opponent policy network π_v nor state-transition model $\mathcal{P}_v^{ss'}$, we cannot get the observation of the opponent agent $\hat{o}_v^{(t+1)}$ and the action of the opponent agent $\hat{a}_v^{(t+1)} = \pi_v(\hat{o}_v^{(t+1)})$. Therefore, author build two individual deep nature networks to approximate the opponent policy network and its state-transition model.

The state-transition model is defined as $H(o_v^{(t)}, a_v^{(t)}, a_\alpha^{(t)}; \theta_h)$, where $(o_v^{(t)}, a_v^{(t)}, a_\alpha^{(t)})$ is input and $o_v^{(t+1)}$ is output. So the $\hat{a}_v^{(t+1)}$ is the output of $H(o_v^{(t)}, a_v^{(t)}, \hat{a}_\alpha^{(t)}; \theta_h)$.

The policy opponent policy model is defined as $F(o_v^{(t)}; \theta_f)$, where $(o_v^{(t)})$ is input and $a_v^{(t+1)}$ is output. So the $\hat{a}_v^{(t+1)}$ is the output of $F(H(o_v^{(t)}, a_v^{(t)}, \hat{a}_\alpha^{(t)}; \theta_h); \theta_f)$.

The L_{ad} can be rewritten as

$$L_{ad} = \text{maximize}_\theta \left(\left\| F\left(H\left(o_v^{(t)}, a_v^{(t)}, \hat{a}_\alpha^{(t)}\right)\right) - a_v^{(t+1)} \right\|_1 - \left\| H\left(o_v^{(t)}, a_v^{(t)}, \hat{a}_\alpha^{(t)}\right) - o_v^{(t+1)} \right\|_1 \right) \tag{2}$$

Obviously, the total loss function is defined as $L_{\text{PPO}} + \lambda L_{\text{ad}}$. The hyperparameter λ indicate the importance of newly added term L_{ad}, which is calculated based on the weight that the opponent agent pays attention to the adversarial. The λ in step time t is defined as

$$I^{(t)} = \left\| F\left(o_v^{(t)}\right) - F\left(o_v^{(t)} \odot \left(\tilde{g}^{(t)} \odot M\right)\right) \right\|_\infty, \quad \lambda^{(t)} = \frac{1}{1 + I^{(t)}}, \tag{3}$$

where $\tilde{g}^{(t)} = \sum_{j=1:q} g_{ij}^{(t)}$, and $g^{(t)ij} = \nabla_{(o_i^{(t)})} F(o_v^{(t)})_j$. In $o_v^{(t)}$, if the corresponding observation dimensions indicate the actions of adversarial agent, we assign 1 to the corresponding element in M, and the rest is assign to 0.