Robust Deep Reinforcement Learning with Adversarial Attacks

Extended Abstract

Anay Pattanaik
University of Illinois at Urbana-Champaign
anayp2@illinois.edu

Zhenyi Tang*
University of Illinois at Urbana-Champaign
ztang11@illinois.edu

Shuijing Liu*
University of Illinois at Urbana-Champaign
sliu105@illinois.edu

Gautham Bommannan
University of Illinois at Urbana-Champaign
bommnnn2@illinois.edu

Girish Chowdhary
University of Illinois at Urbana-Champaign
girishc@illinois.edu

ABSTRACT

This paper proposes adversarial attacks for Reinforcement Learning (RL). These attacks are then leveraged during training to improve the robustness of RL within robust control framework. We show that this adversarial training of DRL algorithms like Deep Double Q learning and Deep Deterministic Policy Gradients leads to significant increase in robustness to parameter variations for RL benchmarks such as Mountain Car and Hopper environment. Full paper is available at (https://arxiv.org/abs/1712.03632) [7].

KEYWORDS

Adversarial Machine Learning; Deep Learning; Reinforcement Learning

ACM Reference Format:
Anay Pattanaik, Zhenyi Tang*, Shuijing Liu*, Gautham Bommannan, and Girish Chowdhary. 2018. Robust Deep Reinforcement Learning with Adversarial Attacks. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

Advances in deep neural networks (DNN) has a tremendous impact in addressing the curse of dimensionality in RL and offers state of the art results in several RL tasks ([4], [8], [5], [6], [9]). However, it has been shown in [2] that DNN can be fooled easily into predicting wrong label by perturbing the input with adversarial attacks. It opens up interesting frontier regarding robustness of machine learning algorithms in general. Robust and high performance policies is critical to enable successful adoption of deep reinforcement learning (DRL) for autonomy problems. More specifically, robustness to real world parameter variations, such as changes in the environmental parameters of the dynamical system are critical.

We address these challenges in an adversarial training framework. We first engineer "optimal" attack on the DRL agent and then leverage these attacks during training that leads to significant improvement in robustness and improves policy performance in challenging continuous domains. Our approach is loosely inspired from the idea of robust control, in which the best case policy is sought over the set containing the worst possible parameters of the system. We translate this into a problem of best performing policy trained in presence of adversary. The key difference, however, is that while robust control approaches tend to be conservative, our approach leverages the inherent optimization mechanisms in DRL to enable learning of policies that have even higher performance over a range of parameter and dynamical uncertainties like friction, mass etc.

The paper is organized as follows. We provide introduction in Section 1. Adversarial attacks and their use for improving robustness have been described in Section 2, and results have been presented in Section 3. Finally, concluding remarks and future directions have been discussed in Section 4.

2 METHOD

2.1 Adversarial Attack

Definition 2.1. An adversarial attack is any possible perturbation that leads the agent into increased probability of taking "worst" possible action in that state. Here, the "worst" possible action for a trained RL agent is the action which corresponds to least Q value.

2.1.1 Naive adversarial attack. First, we propose a naive method of generating adversarial attack. The adversarial attack is essentially a search across nearby observation which will cause the agent to take wrong action. For generating adversarial attack on the DRL policies, we sample a noise with finite (small) support. The particular noise that causes least estimate of the value function is selected as adversarial noise. This noise is added to the current observation.

For naive attack on DDPG, the critic network can be used to ascertain value functions when required and actor network determines the behavior policy to pick action. Thus, the objective function used by adversary in this case is the $Q^*_{critic}(s, a)$, that is, the value function determined by the trained critic network.

2.1.2 Gradient based adversarial attack . In this subsection, we show that a proposed cost function different from the one used in traditional FSGM ([3])) is more effective in finding worst possible action in the context of reinforcement learning with discrete actions.

*Equal Contribution.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

Theorem 2.2. *Let the optimal policy be given by conditional probability mass function (pmf) $\pi^*(a|s)$, the action which has maximum pmf be given as a^* and the worst possible action be given by a_w. Then the objective function whose minimization leads to optimal adversarial attack on RL agent is given by $J(s, \pi^*) = -\sum_{i=1}^{n} p_i log \pi_i^*$ where $\pi_i^* = \pi^*(a_i|s)$, $p_i = P(a_i)$, the adversarial probability distribution P is given by*

$$P(a_i) = \begin{cases} 1, & if\ a_w = 1 \\ 0, & otherwise \end{cases} \quad (1)$$

This is the cross entropy loss between the adversarial probability distribution and optimal policy generated by the RL agent

Q values can be converted into pmf by using softmax function. We can shown that the objective function that should be used for engineering attack on RL algorithm should be given by Theorem 2.2 as it is consistent with Def. 2.1 [7]. FSGM algorithm can be used to minimize this objective function. We must point out that this objective function is different from ones in literature [3]. The objective functions mentioned in [3] will result in $\min_s \pi^*(a^*|s)$ (a^* is the best possible action for given state s). This leads to decrease in the probability of taking best possible action. This won't necessarily lead to increase in probability of taking worst possible action.

The objective function that adversary need to minimize being given by the optimal value function of critic ($Q^*(s, a)$). Here the gradient is given by $\nabla_s Q^*(s, a) = \frac{\partial Q^*}{\partial s} + \frac{\partial Q^*}{\partial U^*} \frac{\partial U^*}{\partial s}$. Here, U^* represents the optimal policy given by actor.

2.1.3 SGD based attack. We also used Stochastic Gradient Descent approach where we followed the gradient descent for same number of sampling time and selected the state that we end up in as adversarial state.

2.2 Robust Reinforcement Learning by harnessing adversarial attacks

2.2.1 Adversarial Training. Adversary fools the agent into believing that it's in a "fooled" state different from actual state such that the optimal action in "fooled" state leads to worst action in actual current state. In other words, the adversary fools the agent into sampling worst trajectories directly. We have used gradient based attack for adversarial training as it performed best amongst all attacks (results presented in Section 3).

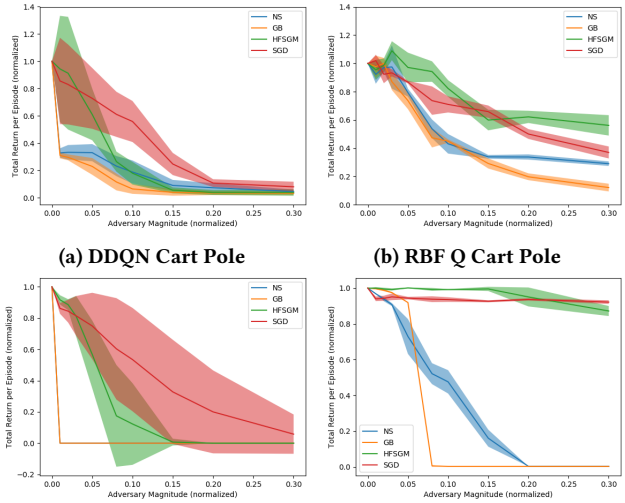
3 RESULTS

We discuss results for proposed adversarial attack and adversarially trained robust policy. All the experiments have been performed within OpenAi gym environment ([1]) with MuJoCo ([10]).

We show that the proposed attack(s) outperform attacks in [3] as shown in Fig. 1. We also present results (Fig. 2) that show significant improvement in robustness because of proposed adversarial training algorithm.

4 CONCLUSION

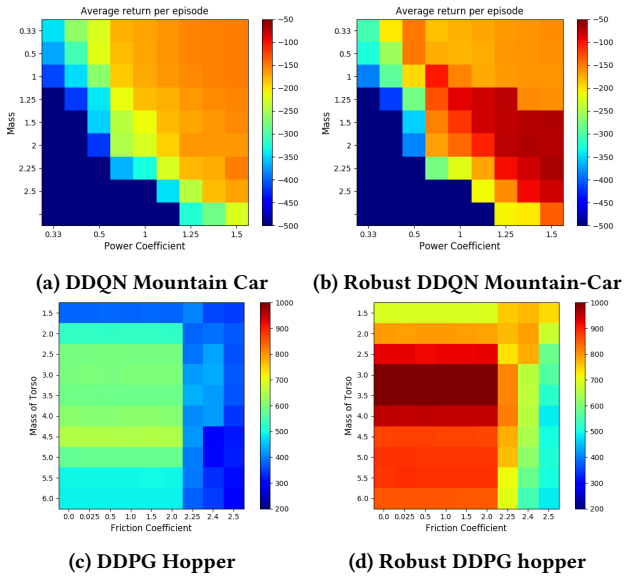
In this paper, we have proposed adversarial attack for reinforcement learning algorithms. We leveraged these attacks to train RL agent that led to robust performance across parameter variations for DDPG and DDQN. Future direction involves providing theoretical relationship between these attacks and robustness of the algorithms (to parameter variation).



(a) DDQN Cart Pole (b) RBF Q Cart Pole

(c) DDQN Mountain Car (d) RBF Q mountain car

Figure 1: Comparison of different attacks. It can be observed that Gradient Based (GB) attack performs better than Naive Sampling (NS) which in turn outperform Stochastic Gradient Descent (SGD) as well as HFSGM ([3]). RBF Q learning is relatively more resilient to adversarial attack than DDQN.



(a) DDQN Mountain Car (b) Robust DDQN Mountain-Car

(c) DDPG Hopper (d) Robust DDPG hopper

Figure 2: Comparison of "vanilla" RL with robust RL. Note the improvement across parameters because of robust training.

5 ACKNOWLEDGEMENT

This work was supported by AFOSR#FA9550-15-1-0146.

REFERENCES

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI gym. *arXiv preprint arXiv:1606.01540* (2016).

[2] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[3] Sandy Huang, Nicolas Papernot, Ian Goodfellow, Yan Duan, and Pieter Abbeel. 2017. Adversarial Attacks on Neural Network Policies. *arXiv preprint arXiv:1702.02284* (2017).

[4] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. 2016. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research* 17, 39 (2016), 1–40.

[5] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[6] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.

[7] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. 2017. Robust Deep Reinforcement Learning with Adversarial Attacks. *arXiv preprint arXiv:1712.03632* (2017).

[8] John Schulman, Sergey Levine, Pieter Abbeel, Michael I Jordan, and Philipp Moritz. 2015. Trust Region Policy Optimization.. In *ICML*. 1889–1897.

[9] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529, 7587 (2016), 484–489.

[10] Emanuel Todorov, Tom Erez, and Yuval Tassa. 2012. MuJoCo: A physics engine for model-based control. In *IROS*.