Inaccuracy of State-Action Value Function For Non-Optimal Actions in Adversarially Trained Deep Neural Policies

Ezgi Korkmaz
KTH Royal Institute of Technology
Stockholm, Sweden
ezgikorkmazk@gmail.com

Abstract

The introduction of deep neural networks as function approximator for the state-action value function has led to the creation of a new research area for self-learning systems that explore policies from high dimensional input. While the success of deep neural policies has resulted in the deployment of these policies in diversified application domains, there are significant concerns regarding their robustness towards specifically crafted malicious perturbations introduced to their inputs. Several studies have focused on making deep neural policies resistant to such perturbations via training with the existence of these perturbations (i.e. adversarial training). In this paper we focus on conducting an investigation on the state-action value function learned by state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies. We perform several experiments in the OpenAI Baselines and we show that the state-action value functions learned by vanilla trained deep neural policies have better estimates for the non-optimal actions than the state-of-the-art adversarially trained deep neural policies. We believe our study lays out intriguing properties of adversarial training and could be critical step towards obtaining robust and reliable policies.

1. Introduction

Advancements in deep neural networks proliferated leading to expansion in the domains where deep neural networks are deployed including image classification [14], natural language processing [23], speech recognition [8] and learning systems via exploration. In particular, deep reinforcement learning became an emerging field with the introduction of deep neural networks as function approximators [18]. Hence, deep neural policies have been deployed in many different domains from pharmaceuticals to self driving cars [4, 10, 11, 19].

As the advancements in deep neural networks continued some research focused on their vulnerabilities towards a certain type of specifically crafted perturbations computed via the cost function used to train the neural network [24, 7, 16, 15, 5]. While some research focused on producing optimal ℓ_p-norm bounded perturbations to cause the most possible damage to the deep neural network models, an extensive amount of work focused on making the networks robust to such perturbations [16, 3, 22].

The vulnerability to such specifically crafted perturbations was inherited by deep neural policies as well [9, 13, 20, 12]. Thus, robustness to such perturbations in deep reinforcement learning became a concern for the machine learning community, and several studies proposed various methods to increase robustness [21, 6, 26]. For these reasons, in this paper we focus on adversarially trained deep neural policies and the state-action value function learned by these training methods in the presence of an adversary.

In this paper we aim to seek answers for the following questions: (i) How accurate is the state-action value function on estimating the values for non-optimal actions?, and (ii) Does adversarial training effect the estimates of the state-action value function for the non-optimal actions? To be able to answer these questions we focus on adversarial training and robustness in deep neural policies and make the following contributions:

- We conduct an investigation on the Q-values and what they represent for the adversarially trained deep reinforcement learning agents and vanilla trained deep reinforcement learning agents.
- We perform several experiments in environments with large state spaces from the OpenAI Atari baselines.
- We find that, for vanilla trained agents, the state-action value function $Q(s, a)$ has a more accurate representation of the actions which are not decided as optimal by the deep neural policy than for adversarially trained deep reinforcement learning agents.

2. Background

2.1. Preliminaries

In deep reinforcement learning the goal is to learn a policy for taking actions in a Markov Decision Process (MDP) that maximize expected cumulative reward. An MDP is represented by a tuple $\mathcal{M} = (S, A, P, r)$ where S is a set of continuous states, A is a discrete set of actions, P is a transition probability distribution on $S \times A \times S$, and $r : S \times A \rightarrow$ is a reward function. The goal in reinforcement learning is to learn a policy $\pi : S \rightarrow \mathcal{P}(A)$ which maps states to probability distributions on actions in order to maximize the expected cumulative reward $R = \mathbb{E} \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t)$ where $a_t \sim \pi(s_t)$. In Q-learning the policy is determined by a learned state-action value function $Q(s, a)$. In particular, the policy is given by choosing the action $a^*(s) = \arg\max_a Q(s, a)$ in state s.

2.2. Adversarial Methods

Szegedy et al. [24] observed that imperceptible perturbations could change the decision of a deep neural network and proposed a box constrained optimization method to produce such perturbations. Goodfellow et al. [7] suggested a faster method to produce such perturbations based on the linearization of the cost function used in training the network. In particular,

$$x_{\text{adv}} = x + \epsilon \cdot \frac{\nabla_x J(x,y)}{||\nabla_x J(x,y)||_p}, \tag{1}$$

in which $J(x, y)$ represents the cost function used to train the deep neural network, x represents the input, and y represents the output labels. Kurakin et al. [15] proposed the iterative version of the fast gradient sign method proposed by Goodfellow et al. [7] inside an ϵ-ball.

$$x^0_{\text{adv}} = x, \tag{2}$$

$$x^{N+1}_{\text{adv}} = \text{clip}_\epsilon(x^N_{\text{adv}} + \alpha \text{sign}(\nabla_x J(x^N_{\text{adv}}, y))) \tag{3}$$

While several other methods have been proposed (e.g. [12]) using a momentum-based extension of the iterative fast gradient sign method,

$$v_{t+1} = \mu \cdot v_t + \frac{\nabla_{s_{\text{adv}}} J(s^t_{\text{adv}} + \mu \cdot v_t, a)}{\|\nabla_{s_{\text{adv}}} J(s^t_{\text{adv}} + \mu \cdot v_t, a)\|_1} \tag{4}$$

$$s^{t+1}_{\text{adv}} = s^t_{\text{adv}} + \alpha \cdot \frac{v_{t+1}}{\|v_{t+1}\|_2} \tag{5}$$

adversarial training has mostly been conducted with perturbations computed by projected gradient descent (PGD) proposed by Madry et al. [16] (i.e. Equation 3). Majority of adversarial training methods are based on training with adversarial examples produced by PGD method.

2.3. Adversaries and Training in Deep Neural Policies

The robustness of deep neural policies is a great concern. Thus, imperceptible perturbations were initially applied to the observations of deep reinforcement learning agents concurrently by Kos et al. [13] and Huang et al. [9], with both works utilizing the fast gradient sign method proposed by [7]. While several works have focused on the optimization part of producing adversarial perturbations [12, 20], a line of research has focused on gaining resistance to such perturbations. Madlekar et al. [17] propose adding perturbations to states in training time in robotics application of deep reinforcement learning. Pinto et al. [21] propose a joint training strategy in the existence of an adversary whose aim is to minimize the expected cumulative rewards of the agent based on zero-sum Markov game modelling. Glaeve et al. [6] considers an adversary limited to take natural actions in the environment instead of introducing ℓ_p-norm bounded perturbations. Authors in this paper model the relationship between the adversary and the agent as zero-sum Markov game and solve it via self playing. Quite recently, Huan et al. [26] modeled this interaction between the adversary and the agent as a modified MDP called a state-adversarial MDP, and claimed that their proposed algorithm State Adversarial Deep Q-Network learns theoretically certified robust policies against natural noise and adversarial perturbations.

3. Experimental Details

Our experiments are conducted in the OpenAI [2] Atari baselines designed by Bellemare et al. [1]. The vanilla trained deep neural policy is trained via Double Deep Q-Network (DDQN) [25] and the state-of-the-art adversarially trained deep neural policy is trained via State-Adversarial Double Deep Q-Network (SA-DDQN) [26]. Our results are averaged over 10 episodes. In all of our figures we have also included the standard error of the mean. In detail, we measure the performance drop of an agent as,

$$\mathcal{P} = \frac{\text{Score}_{\text{clean}} - \text{Score}_{\text{actmod}}}{\text{Score}_{\text{clean}} - \text{Score}^{\text{fixed}}_{\text{min}}}. \tag{6}$$

where $\text{Score}_{\text{clean}}$ represent the clean run of the game where no perturbations introduced to the agent's observations, $\text{Score}^{\text{fixed}}_{\text{min}}$ represents the minimum score available for a given game, and $\text{Score}_{\text{actmod}}$ represents the run of the game where the actions of the agent are modified for a fraction of the state observations.

4. An Analysis on the Inaccuracy of State-Action Values for Non-optimal Actions

In this paper we examine the state-action value function of the state-of-the-art adversarially trained deep neu-

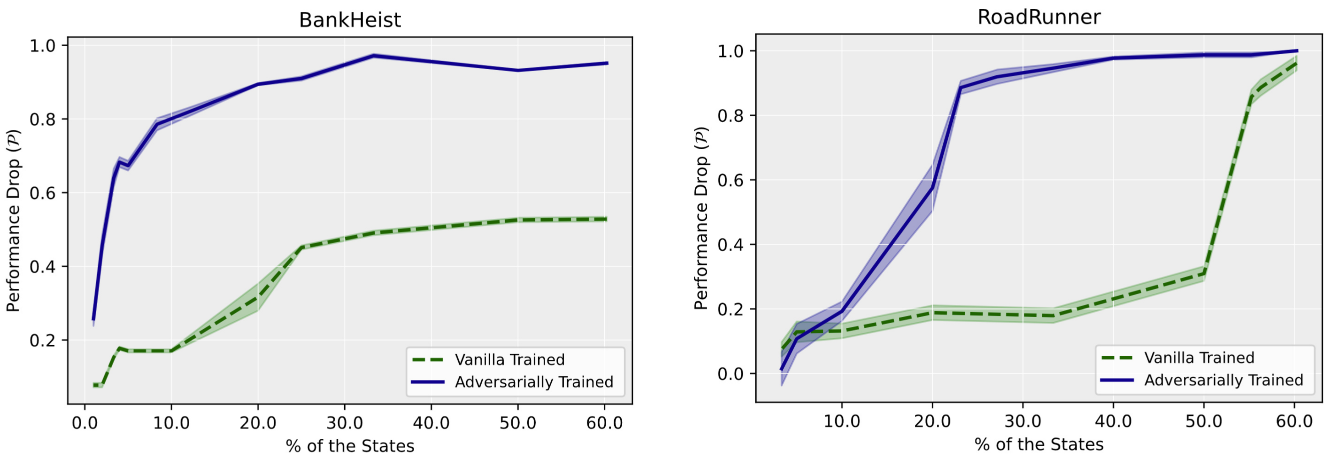

Figure 1: Performance drop with respect to action modification percentage for the state-of-the-art adversarially trained deep neural policies [26] and vanilla trained deep neural policies.

ral policies and vanilla trained deep neural policies. To achieve this goal we systematically modify the action taken by the agent and analyze the effects on the performance of the trained deep neural policy. In particular, we make the agent choose non optimal actions for n state observations in a given episode e consisting of m total state observations. Formally, let $a_{i^{\text{th}}}$ be the i$^{\text{th}}$ best action decided by the deep neural policy in a given state s (i.e. $Q(s, a)$ is sorted in decreasing order, and $a_{i^{\text{th}}}$ is the action corresponding to i$^{\text{th}}$ largest Q-value). We record the scores obtained by the action-modified deep neural policies at the end of the episode and compute the impact of the action modification on the performance of the deep neural policy. Our aim is to provide an analysis on how accurate the state-action value function is in representing values for the non-optimal actions.

In Figure 1 we show the performance drop as a function of the fraction of states in which the action modification is applied for state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies. In particular, the action modification is set for the second best action $a_{2^{\text{nd}}}$ decided by the state-action value function $Q(s, a)$. As we increase the fraction of states in which the action modification set to $a_{2^{\text{nd}}}$ is applied, we observe a performance drop for both of the deep neural policies. However, we observe that vanilla trained deep neural policies experience a lower performance drop with this modification. Especially in BankHeist we observe that the performance drop does not exceeds 0.55 for even when the action modification is applied for a large fraction of the visited states for the vanilla trained deep neural policies. This gap in the performance drop between the adversarially trained and vanilla trained deep neural policies indicates that the state-action value function learnt by vanilla trained deep neural policies has a better estimate for the non-optimal actions. We hypothesize that the adversarial training places higher emphasis on ensuring that the highest ranked action (i.e. the action that maximizes the state-action value function in a given state) does not change under small ℓ_p-norm bounded perturbations, rather than accurately computing the state-action value function. Since historically Q-learning suffered from overestimation of Q-values, a method which places higher emphasis on the highest ranked action risks converging to a state-action value function with overestimated Q-values.

5. Conclusion

In this paper we focused on the state-action value function for the state-of-the-art adversarially trained deep neural policies and vanilla trained deep neural policies. We tested trained deep neural policies with systematic action modification in various fractions of the visited states and recorded the performance drop of the trained policies. In particular, we made the deep neural policy choose $a_{i^{\text{th}}}$, the i^{th} best action decided by the deep neural policy, for various fractions of the observed states. We observe that adversarially trained deep neural policies experience a larger performance drop for the same action modification. Thus, our observation indicates that the state-action value function learnt by vanilla trained deep neural policies have a better estimate for the non-optimal actions. We believe our investigation lays out intrinsic properties of adversarial training and can be conducive to building robust and optimal deep neural policies.

References

[1] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research.*, page 253–279, 2013. 2

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv:1606.01540*, 2016. 2

[3] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, and Percy Liang. Unlabeled data improves adversarial robustness. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11190–11201, 2019. 1

[4] Liu Daochang and Tingting. Jiang. Deep reinforcement learning for surgical gesture segmentation and classification. *In International conference on medical image computing and computer-assisted intervention.*, pages 247–255.Springer, Cham, 2018. 1

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. *In Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 1

[6] Adam Gleave, Michael Dennis, Cody Wild, Kant Neel, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. *International Conference on Learning Representations ICLR*, 2020. 1, 2

[7] Ian Goodfellow, Jonathan Shelens, and Christian Szegedy. Explaning and harnessing adversarial examples. *International Conference on Learning Representations*, 2015. 1, 2

[8] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Diamos Greg, Erich Else, Ryan Prenger, Sanjeev Satheesh, Sengupta Shubho, Ada Coates, and Andrew Ng. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014. 1

[9] Sandy Huang, Nicholas Papernot, Yan Goodfellow, Ian an Duan, and Pieter Abbeel. Adversarial attacks on neural network policies. *Workshop Track of the 5th International Conference on Learning Representations*, 2017. 1, 2

[10] Tseng Huan-Hsin, Sunan Cui, Yi Luo, Jen-Tzung Chien, Randall K. Ten Haken, and Issam El. Naqa. Deep reinforcement learning for automated radiation adaptation in lung cancer. *Medical physics 44*, pages 6690–6705, 2017. 1

[11] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey. Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018. 1

[12] Ezgi Korkmaz. Nesterov momentum adversarial perturbations in the deep reinforcement learning domain. *International Conference on Machine Learning, ICML 2020, Inductive Biases, Invariances and Generalization in Reinforcement Learning Workshop.*, 2020. 1, 2

[13] Jernej Kos and Dawn Song. Delving into adversarial attacks on deep policies. *International Conference on Learning Representations*, 2017. 1, 2

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 2012. 1

[15] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1, 2

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 1, 2

[17] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Li Fei-Fei, and Silvio Savarese. Adversarially robust policy learning: Active construction of physically-plausible perturbations. *In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3932–3939, 2017. 2

[18] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, arc G Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. 1

[19] Laura Noonan. Jpmorgan develops robot to execute trades. *Financial Times*, page 1928–1937, July 2017. 1

[20] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, and Bommannan Gautham. Robust deep reinforcement learning with adversarial attacks. *In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2040–2042, 2018. 1, 2

[21] Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. *International Conference on Learning Representations ICLR*, 2017. 1, 2

[22] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7909–7919. PMLR, 2020. 1

[23] Ilya Sutskever, Oriol Vinyals, and Quoc V. . Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 2014. 1

[24] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dimutru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *In Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 1, 2

[25] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando. De Freitas. Dueling network architectures for deep reinforcement learning. *Internation Conference on Machine Learning ICML.*, page 1995–2003, 2016. 2

[26] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane S. Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state

observations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 1, 2, 3