Stealthy and Efficient Adversarial Attacks against Deep Reinforcement Learning

Prakash K. Naikade
prna00001@stud.uni-saarland.de

1 Summary

This paper proposes two novel techniques to attack a Deep Reinforcement Learning (DRL) agent efficiently and stealthily. First, $the\ Critical\ Point\ Attack$, in which the adversary builds a domain-specific model to predict the states of the next few steps as well as the damage consequences of all possible attack strategies, then the adversary adopts a Damage Awareness Metric with the criteria of the agent's end goal to assess each attack strategy, which guarantees that the adversary can select the optimal solution with a very small number of attack steps. Second, $the\ antagonist\ attack$, in this the adversary trains a domain-agnostic antagonist model to identify the best attack strategy from one specific state automatically. To train this model, the agent's end goal and the reward function as the optimization object is used. To achieve the maximum damage, the model will decide when and how to add perturbations. The efficiency of the proposed attacks is demonstrated by experiments and their results.

1.1 Overview of attack techniques

Proposed problem statement modifies the Markov Decision Process (MDP) to decrease reward R by adding perturbation δ_t into the agent's observation s_t and forcing the agent into performing the wrong action a_t. Specifically, the adversary tries to minimize the expected reward $R = \sum_{t=0}^{T-1} \mathbb{E}_{a_t \sim u(s_t + k_t \delta_t)}[\gamma^t r(s_t, a_t)]$, where the attack strategy k_t denotes whether at time step t the perturbation should be applied ($k_t = 1$) or not ($k_t = 0$).
They simplify this problem by dividing it into two sub-problems, (1) $when\ to\ attack$ where they identify the optimal attack strategy $\{k_0, ..., k_{T-1}\}$ by using proposed critical point attack and antagonist attack, and (2) $how\ to\ attack$ where they compute corresponding perturbations $\{\delta_0, ..., \delta_{t-1}\}$ by using existing adversarial example techniques to compute and add perturbations in the selected critical moments from the previous step.

1.1.1 Critical point attack

To carry out this attack paper proposed prediction model which exposes the subsequent states and agent's actions. At step t, the adversary considers all possible N consecutive perturbations from this step to step $t + N - 1$, predicts the environment state and evaluates the attack damage at step $t + M$ where $M \geq N$ is a predefined parameter. The adversary picks the N perturbations that can cause the most severe damage at step $t + M$ and adds them to the states of the following N steps. This is done in two steps, (1) the adversary samples the observations of the target environment and trains a prediction model, (2) then they introduce divergence function and Danger Awareness Metric DAM to assesses the potential damage of all possible strategies at specific step. If this DAM is larger than a threshold Δ then adversary will conduct the attacks in the next N steps following this strategy and if not adversary will repeat this assessment process from the next step $t + 1$.

Preprint. Under review.

1.1.2 Antagonist attack

To carry out this attack they introduced antagonist (adversarial agent) which maintains a policy $u^{adv} : s_i \mapsto (p_i, a'_i)$, mapping the current state to the attack strategy. At each time step t, the antagonist observes state s_t, and produces the strategy (p_t, a'_t). If $p_t > 0.5$, then step t is chosen as the critical point and the adversary adds the perturbation to mislead the agent to trigger the action a'_t. Otherwise, the agent follows the original action a_t.

1.1.3 Finding and adding perturbation

The proposed problem statement to find perturbation δ_t at time step t is, $argmax_i F'(s_t + \delta_t)_i = a'_t$, where F' is policy network which outputs the action probability, t is critical moment, a'_t is adversarial target action and computed δ_t by using C&W attack technique to generate the adversarial perturbation.

2 Strengths

The paper is clean and easy to follow. This paper proposed a more optimal technique as it assesses and compares all the possible attack scenarios; based on the fact that critical point assault requires only one step to attack TORCS and two steps to attack Atari Pong and Breakout, whereas strategically-timed attack (Lin et al.2017) requires 25% of the total steps to attack the same applications. The proposed attacks are generic and effective for various reinforcement learning tasks and algorithm.

3 Weaknesses

The experimental evaluation in terms of performance comparison with Lin et al.2017 is not clear. Experiments doesn't have complex and satisfactory number of environments to prove the generic nature of attack techniques to various agents trained by algorithms like A3C, Deep Deterministic Policy Gradient method(DDPG), and Proximal Policy Optimization (PPO).

4 Possible Improvements

The antagonist policy could be trained with many safety goals to increase the robustness of DRL algorithms.

5 Possible Extensions

Developing defenses against adversarial attacks of this nature could be a critical extension of this research such as augmenting training data with adversarial examples or training a sub-network to detect adversarial input at test time and deal with it cautiously.