Tactics of Adversarial Attack on Deep Reinforcement Learning Agents

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

1 Summary

The paper present the "when to attack" extension of a misclassification attack (Carlini and Wagner, 2016) to make the agent choose its worst action, i.e., lowest Q-score or lower probability for policy π, instead of best, as a "strategically-timed attack" and it also explains an extension of the single time step (s, a, r, s') version to a sequence version, through the use of a forward model (Oh et al., 2015) as an "enchanting attack".

This paper suggests uniform attack is more susceptible to detection as adversary perturbs the observation at every time instance, and therefore authors proposes two novel attacks, strategically-timed attack and enchanting attack.

1.1 Strategically-timed attack

Strategically-timed attack aims at attacking a deep RL agent at critical moments in an episode to minimize the agent's reward. Authors proposed optimization problem to minimize the expected accumulated reward by strategically attacking less than all the time steps for this attack is as follows,

$$\begin{aligned} \min_{b_1,b_2,...,b_L,\delta_1,\delta_2,...,\delta_L} \quad & R_1(\bar{s}_1,...,\bar{s}_L) \\ & \bar{s}_t = s_t + b_t\delta_t && \text{for all } t = 1,...,L \\ & b_t \in \{0,1\}, && \text{for all } t = 1,...,L \\ & \sum_t b_t \leq \Gamma \end{aligned} \tag{1}$$

Where, $\{s_1,...,s_L\}$ are sequence of observations or states in an episode, $\{\delta_1,...,\delta_L\}$ are sequence of perturbations, $\mathcal{R}_1$ is the expected return at the first time step, $b_1,...,b_L$ denote when to attack.

In policy gradient-based approaches like the A3C algorithm, an agent strongly desires an action with a high probability; this means that performing that action is crucial, as otherwise the accumulated reward would be lowered. The authors devised the function c to answer the "when to attack" (to compute $\{b_1,...,b_L\}$) problem based on this insight, is as follows,

$$c(s_t) = \max_{a_t} \pi(s_t,a_t) - \min_{a_t} \pi(s_t,a_t). \tag{2}$$

When the relative action preference function c has a value greater than a threshold parameter, adversary strikes the deep RL agent at time step t.

Authors defined the function c for value-based approaches like DQN, based on the same rationale,

$$c(s_t) = \max_{a_t} \frac{e^{\frac{Q(s_t,a_t)}{T}}}{\sum_{a_k} e^{\frac{Q(s_t,a_k)}{T}}} - \min_{a_t} \frac{e^{\frac{Q(s_t,a_t)}{T}}}{\sum_{a_k} e^{\frac{Q(s_t,a_k)}{T}}}. \tag{3}$$

Using the softmax function (Huanget al., 2017) with the temperature constant T, the authors transform the estimated Q-values of actions into a probability distribution over actions.

Preprint. Under review.

The authors employed the attack strategy presented in Carlini and Wagner, 2016 as a solution to the "how to attack" (to compute $\{\delta_1, ..., \delta_L\}$) problem using the output of a trained deep RL agent as a cue to design effective adversarial examples for decreasing accumulated rewards. This method solves following optimization problem to generate adversarial example,

$$\min_{\delta} \mathcal{D}_I(x, x+\delta)$$
$$\text{subject to} \quad f(x) \neq f(x+\delta), \tag{4}$$

Where x is an image, f is a DNN, $\mathcal{D}_I$ is an image similarity metric which finds minimal perturbation, δ, so that it can lead to misclassification of an image by DNN.

According to the authors, their the strategically-timed attack can achieve same effect as the uniform attack by attacking just 25% of the time steps in an episode.

1.2 Enchanting attack

Proposed enchanting attack lures the deep RL agent from current state s_t at time step t to a specified target state s_g after H steps. The authors assume that they have complete control over the agent and that it can do whatever action they want at any time. The authors then used the method described in Carlini and Wagner, 2016 to create adversarial examples in order to persuade an agent to take planned action sequence. For future state prediction authors proposed prediction model M introduced in Oh et al., 2015, which used a generative model to predict a video frame in the future as follows,

$$s^M_{t+H} = M(s_t, A_{t:t+H}) \,, \tag{5}$$

where $A_{t:t+H} = \{a_t, ..., a_{t+H}\}$ is the given sequence of H future actions beginning at step t, s_t is the current state, and s^M_{t+H} is the predicted future state.

The authors also computed a sequence of actions to steer the RL agent toward the target state using the sampling-based cross-entropy method (Rubinstein and Kroese, 2013). The success of the attack is determined by distance between s_g and s^M_{t+H}, which is given by $D(s_g, s^M_{t+H})$.

According to experiments performed by authors, when $H < 40$, the success rate was more than 70%.

2 Strengths

In comparison to Huang et al., 2017, this is an innovative use of a "classic test time" adversarial attacks. The overall experimental design is sound, and it covers all potential scenarios. The research topic is interesting, and the contributions are substantial.

3 Weaknesses

The strategically-timed attack technique only considered the attack effect on one time step, ignoring the impact on the following states and actions, i.e., not considering the final end goal. In an enchanting attack, accurate prediction and enforcement of future states and actions is difficult, and thus this approach suffers from a low attack success rate. The experimental evaluation in general and in-terms of performance comparison with Huang et al., 2017 is weak and vague. The paper's flaw is that the problem isn't well-motivated, and the authors jumped right into the content without explaining the publications they utilized to develop these adversarial tactics.

4 Possible Improvements

This paper requires a more thorough overview of the relevant work on which it is based, as well as a more fluid narrative and structure.

5 Possible Extensions

Developing defenses against these kind of adversarial attacks could be crucial possible extension of this work. Authors could develop a more sophisticated strategically-timed attack strategy. They can also concentrate on enhancing the generative model's video prediction accuracy in order to increase the success rate of enchantment attacks in more complex environments.