Implementation and Critic of the OpenAI paper on Proximal Policy Optimization Algorithms

Alvaro Caudéran
Implementation: https://github.com/Alvaro2112/AtariPPO

Abstract

In this paper I will talk about the Reproduction / Implementation of the 2017 paper on Proximal Policy Optimization (PPO) published by OpenAI. I will briefly present the original paper main ideas, discuss my results and compare them with the original paper results. Finally I'm going to make a brief critic on how readable/reproducible I found the paper to be.

1 Introduction

The following paper was written for the EEML Application, and is a summary of the PPO method.

2 PPO Overview

PPO or Proximal Policy Optimization is an on-policy Reinforcement Learning algorithm that aims to be scalable, data-efficient, and robust on both continuous and discrete action spaces, unlike previous methods that lacked one of these aspects (ie. Q-learning, Vanilla policy Gradient methods, TRPO).

It proposes a new objective that uses clipped probability ratios, which forms a worst-case estimate of the performance of the policy. And to optimize policies, it alternates between sampling data from the policy and performing several epochs of optimization on the sampled data.

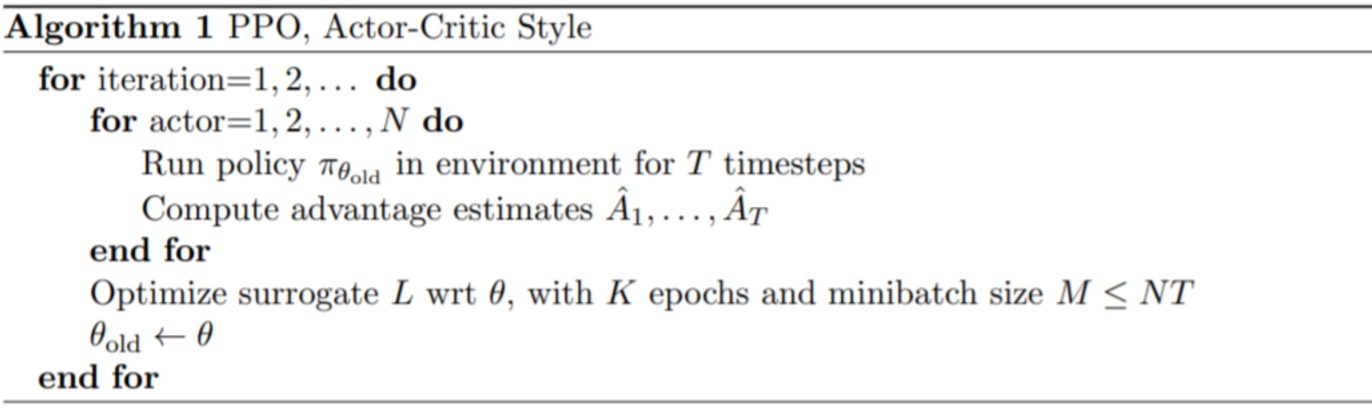
2.1 Objective function

The main objective that was proposed is the following:

$$L^{CLIP}(\theta) = \hat{E}_t[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)]$$

The reasoning behind this objective is as such, first the clipping function makes sure that the update that is being done isn't too large (it checks that the ratio between the old and new policy is in a certain range). Then the Advantage term (actual value minus expected value) is introduced to provide a direction and a magnitude in which to update. Finally, the min function is there so that the objective only ignores the changes in probability ratio when it would make the objective improve, and it includes it when it makes the objective worse

2.2 Algorithm used:

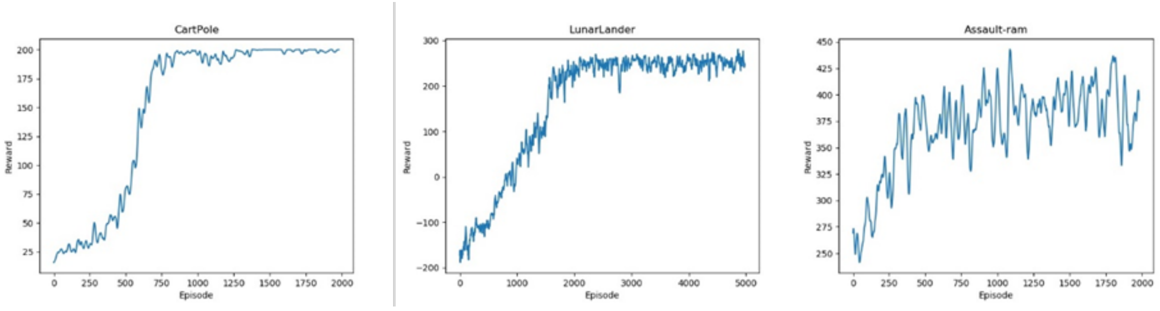


Algorithm 1 PPO, Actor-Critic Style

```
for iteration=1, 2, ... do
    for actor=1, 2, ..., $N$ do
        Run policy $\pi_{\theta_{\text{old}}}$ in environment for $T$ timesteps
        Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
    end for
    Optimize surrogate $L$ wrt $\theta$, with $K$ epochs and minibatch size $M \leq NT$
    $\theta_{\text{old}} \leftarrow \theta$
end for
```

3 Experiments

I have tried running the algorithm on 3 Atari environments from the OpenAI Gym, due to the computational power limitations I have on my computer, I only trained the Algorithm on environments that did not require an excessive number of episodes to be solved:



Training Rewards in Atari environments.

These results confirm what the team at OpenAI had originally found, that PPO outperforms most other algorithms (At the time of release, 2017) in most Atari games and is at the same time very robust. Of course, my data is very limited as I did not test it on many environments, but on the ones that I did I had very good and consistent results compared to other algorithms I had implemented (ie. DDQN, Q-Learning), and all of this without changing Hyper Parameters (except for maximum steps per episode).

4 Final thoughts and Critics

It comes without saying that the implemented paper was extremely interesting and taught me a lot of new practices. But, although I know that this paper is intended for more experienced researchers (ie. PhD's), I did find some complications along the process of implementing the paper. For example, although most part of the algorithm is explained in detail, I found that some essential aspects of it were not explained in detail. But I would assume that these details are well known among the more educated audience and that this paper is not tailored to a more novice audience.

And finally, I did encounter some problems using some of their Hyper Parameters as they did not help my Agent, so I ended up tuning the Hyper Parameters that suited my implementation best.

Overall, this paper still helped me a lot to understand many of the concepts and I did not find it too difficult to understand most of the topics explained. It was definitely a fascinating paper, as you would expect from the OpenAI team.

References

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, OpenAI ”Proximal Policy Optimization Algorithms”