Tsinghua University

Deep Reinforcement Learning

Spring 2024

Assignment 4

Subject: Proximal Policy Optimization
Author: Tonghe Zhang(张同和)
Email: zhang-th21@mails.tsinghua.edu.cn
Affiliation: Department of Electronic Engineering, Tsinghua University June 10, 2024

Contents

1 Proximal Policy Optimization Baseline implementation

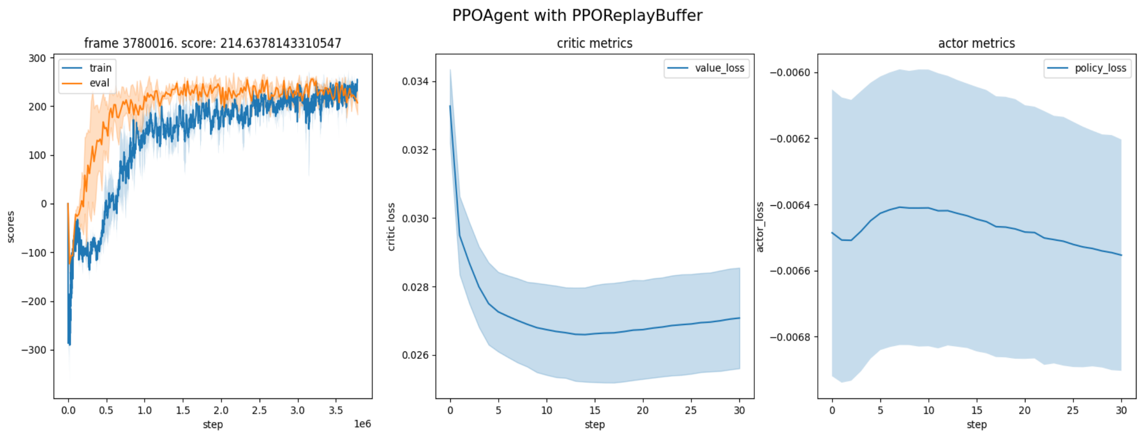

Fig 1: PPO training and evaluation result. Our agent obtains a score of 214.64 in the end, and the value loss and critic loss curves are smooth.

Few remarks during implementation of PPO:

- There are at least 37 variants of PPO. A very pedagogical introduction is provided in this website.
- The advantage function in PPO is different from those implemented in Q-lerning variants. It is a TD(λ)-style truncated accumulated reward, and it is not a function of (s, a) but a function of the states only. The V^{target} used to calculate value function loss is defined by the sum of current value function and current advantage.
- The clipped surrogate function should be maximized, so the clipped surrogate loss should take a minus sign. Also we should be careful to detach the old policy network and actor network when calculating the $L^{\text{clip}}(\theta)$, as it is only used to optimized the current policy.

2 Bonus: Proximal Policy Optimization with Clipped Value Loss

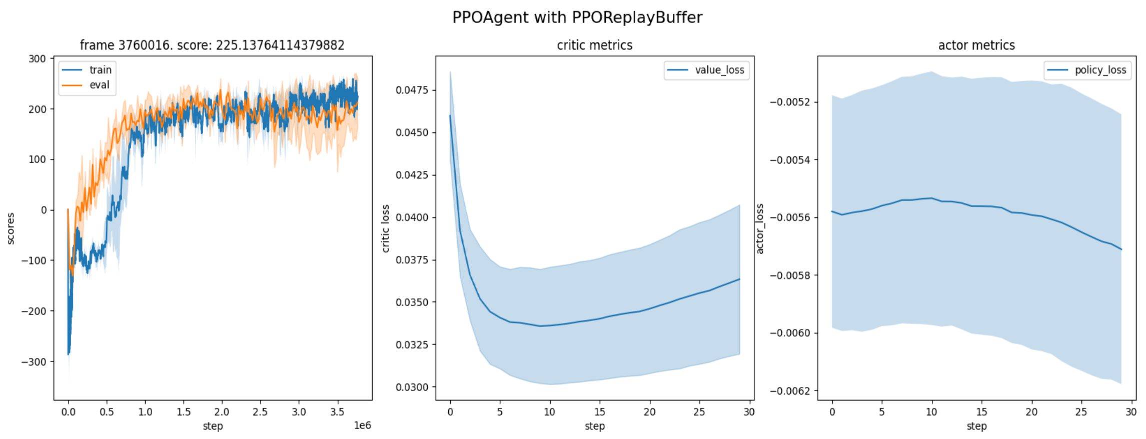

Fig 2: PPO training and evaluation result with clipped value loss. Our agent obtains a score of 225.13, surpassing that when using MSE critic loss.

The only two differences between this version and the baseline is that we changed the hyperparameter “value_clip_range” in “config//ppo.yaml” from null to 0.2, and we implementation clipped value loss in the following way:

```python
    if self.value_clip_range is None:
        value_loss = ((value-returns)**2).mean()
    else:
        clipped_value = torch.clamp(value, old_value-self.value_clip_range, old_value+self.value_clip_range)
        value_loss = (torch.max((value-returns)**2, (clipped_value-returns)**2)).mean()
```

We see a small increase in the returns after we clip the outputs of the value network during value_loss calculation. This can be atributed to the increase of learning stability using bounded value functions to calculate loss.