2.2 Pseudocode

Algorithm 1 PPO in 9 steps

Require:

A differentiable policy parameterization $\pi(a|s, \theta)$

A differentiable state-value function parameterization $\hat{v}(s, w)$

Initialize the policy parameters θ, the state-value weights w, learning rates α_θ and α_w

for Loop (for each episode): **do**

Generate an episode $S_0, A_0, R_1, \cdots, S_{T-1}, A_{T-1}, R_T$ with policy $\pi(\cdot|\cdot, \theta)$

Calculate value estimates $V(S_t) = \hat{v}(S_t, w)$ over timesteps $t = 0, 1, ..., T-1$

#PPO 7

Calculate lambda returns over timesteps $t = 0, 1, ..., T-1$ based on corresponding value estimates using $G_t^\lambda = R_{t+1} + \gamma[(1-\lambda)V(S_{t+1}) + \lambda G_{t+1}^\lambda]$

#PPO 3

Collect a batch of episodes

if batch size == B **then**

$\pi_{old}(.|., \theta) \leftarrow \pi(.|., \theta)$

Calculate $\pi_{old}(A_t|S_t, \theta)$

#PPO 2

Calculate the advantage function $H_t = G_t^\lambda - V(S_t)$ from the computed lambda returns and value estimates across the collected batch.

#PPO 8

Normalize advantage across the batch as $H_t = \dfrac{(H_t - H_{mean})}{H_{stddev}}$

#PPO 5

for Loop (for each epoch): **do**

Shuffle batch

#PPO 4

for Loop (for each minibatch in batch) **do**

Calculate $\pi(A_t|S_t, \theta)$

Reuse previously computed $\pi_{old}(A_t|S_t, \theta)$ for corresponding minibatch

#PPO 6

Calculate $\rho = \dfrac{\pi(A_t|S_t, \theta)}{\pi_{old}(A_t|S_t, \theta)}$

$loss_{value} = E[(G_t^\lambda - V(S_t))^2]$ over minibatch

#PPO 1, #PPO 9

$\rho^{Clip} = \text{Clip}(\rho\ ,\ 1-\epsilon\ ,\ 1+\epsilon)$ using $\epsilon = 0.2$

$loss_{policy} = -E[min(\rho.H_t\ ;\ \rho^{Clip}.H_t)] + loss_{value}$ over minibatch

Backpropagate on policy network using Adam over minibatch

Backpropagate on value network using Adam over minibatch

end for

end for

Make batch empty to collect new episodes

end if

end for