Homework 4

4771 Reinforcement Learning (Spring 2026)

Submission deadline: 11:59pm, April 12

The latex template can be found here. The starter code is here. This starter code merges the implementations of DQN and PPO. We will focus on the PPO part, while the DQN part serves as a solution for HW3.

1 PPO for CartPole

In this homework, we implement PPO with the Monte Carlo estimator to play CartPole. The algorithm is shown in Algorithm 1. It has a different policy update rule compared to the version discussed in the class. Define $\varrho_\theta(s,a) = \frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}$ and $\hat{A}(s,a) = \hat{G}(s,a) - V_\phi(s)$. **Instead of** updating the policy with

$$\theta \leftarrow \theta + \alpha \nabla_\theta \left(\varrho_\theta(s,a)\hat{A}(s,a) - \beta \mathrm{KL}(\pi_\theta(\cdot|s), \pi_{\theta_k}(\cdot|s)) \right) \tag{1}$$

as in Page 37 of this slide (omitting batching), we update it with

$$\theta \leftarrow \theta + \alpha \nabla_\theta \left(\min\{ \varrho_\theta(s,a)\hat{A}(s,a), \varrho_\theta^{\text{clip}}(s,a)\hat{A}(s,a) \} \right) \tag{2}$$

where $\varrho_\theta^{\text{clip}}(s,a)$ is a projection of $\varrho_\theta(s,a)$ to the range $[1-\epsilon, 1+\epsilon]$ for some ϵ usually chosen as 0.1 or 0.2. See Algorithm 1 for more detailed pseudocode.

The update rules (1) and (2) are two variants of PPO (see Equation (7) and (8) of the PPO paper), and are usually called PPO-Penalty and PPO-Clip respectively (spinning up webpage). The KL penalty and the clipping operation both introduce a more stable policy updates. The former imposes a soft penalty, while the latter simply makes the gradient being zero if the policy change $\varrho_\theta(s,a)$ is outside a certain range.

In the CartPole environment, we find that PPO-Clip is significantly more stable, so this will be the version you implement in this homework. Besides the policy update, please verify yourself the equivalence between Algorithm 1 and the algorithm on in Page 37 of this slide.

(a) (40%) Implement PPO-Clip to play CartPole, and paste the output plot here. Your score for this part will be:

$$\min\left\{ \frac{\text{maximum running average within 2000 episodes}}{475}, 1 \right\} \times 100\%.$$

Recommended choices of the hyperparameters are provided in the starter code.

(b) (5%) List the hyperparameters you use in (a):

Hyperparameter	Value
`N` (N in Algorithm 1)	
`M` (M in Algorithm 1)	
`MINIBATCH_SIZE` (B in Algorithm 1)	
`LR` (α in Algorithm 1)	
`EPS` (ϵ in Algorithm 1)	

If you use any trick beyond what is described in Algorithm 1, please briefly describe it here. Also, list any additional hyperparameters you introduced in the table above.

Algorithm 1 PPO-Clip

Parameters: N, M, B, α (learning rate), λ (GAE parameter), β (KL penalty coefficient), ϵ (clipping parameter).
Initialize the policy network π_θ with weights θ
Initialize the value network V_ϕ with weights ϕ.

Sample $s_{\text{tmp}} \sim \rho$, where ρ is the initial state distribution.
for $k = 1, \dots$ **do**
  $s_1 \leftarrow s_{\text{tmp}}$
  for $i = 1, 2, \dots, N$ **do**
    Select action $a_i \sim \pi_\theta(\cdot|s_i)$.
    Observe reward r_i and the next state s'_i.
    If s'_i is a terminal state, sample $s_{i+1} \sim \rho$; otherwise, set $s_{i+1} = s'_i$.
  $s_{\text{tmp}} \leftarrow s_{N+1}$
  Compute $\hat{G}(s_i, a_i)$ for all $i = 1, \dots, N$ by calling Algorithm 2 with inputs $\{s_i, a_i, r_i, s'_i\}_{i=1}^N$ and V_ϕ.
  Define $\hat{\pi}(a_i|s_i) = \pi_\theta(a_i|s_i)$ for all $i = 1, \dots, N$.
  Define $\hat{A}(s_i, a_i) = \hat{G}(s_i, a_i) - V_\phi(s_i)$ for all $i = 1, \dots, N$.
  Define $\hat{V}(s_i) = r_i + \gamma V_\phi(s'_i)\mathbb{I}\{s'_i \text{ is not a terminal state}\}$ for all $i = 1, \dots, N$.
  $\hat{G}(s_i, a_i), \hat{\pi}(a_i|s_i), \hat{A}(s_i, a_i), \hat{V}(s_i)$ should all be viewed as constant below—call `tensor.detach()` for them.

  for $j = 1, \dots, M$ **do**
    Randomly sample a minibatch $\mathcal{B} \subset \{(s_i, a_i, r_i, s'_i)\}_{i=1}^N$ with size $|\mathcal{B}| = B$.
    Update the policy network:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \frac{1}{B} \sum_{(s,a,r,s') \in \mathcal{B}} \min\left\{ \varrho_\theta(s,a)\hat{A}(s,a), \varrho_\theta^{\text{clip}}(s,a)\hat{A}(s,a) \right\},$$

    where $\varrho_\theta(s,a) \triangleq \frac{\pi_\theta(a|s)}{\hat{\pi}(a|s)}$ and $\varrho_\theta^{\text{clip}}(s,a) \triangleq \max\{1-\epsilon, \min\{1+\epsilon, \varrho_\theta(s,a)\}\}$.
    Update the value network:

$$\phi \leftarrow \phi - \alpha \nabla_\phi \frac{1}{B} \sum_{(s,a,r,s') \in \mathcal{B}} \left(V_\phi(s) - \hat{V}(s) \right)^2.$$

Algorithm 2 Monte Carlo Estimator

Input: $\{(s_i, a_i, r_i, s'_i)\}_{i=1}^N$ and V_ϕ.
$\hat{G}(s_N, a_N) \leftarrow r_N + \gamma V_\phi(s'_N)\mathbb{I}\{s'_N \text{ is not a terminal state}\}$
for $i = N-1, \dots, 1$ **do**
  $\hat{G}(s_i, a_i) \leftarrow r_i + \gamma \hat{G}(s_{i+1}, a_{i+1})\mathbb{I}\{s'_i \text{ is not a terminal state}\}$
return $\{\hat{G}(s_i, a_i)\}_{i=1}^N$.

2 Survey

(5%) Leave any feedback for the course.