

REINFORCEMENT LEARNING & ADVANCED DEEP

M2 DAC

TME 6. Advanced Policy Gradients

Ce TME a pour objectif d'expérimenter l'approche PPO.

1 PPO Adaptative KL

Implémenter l'algorithme PPO avec coût KL adaptatif donné dans la figure ci-dessous et l'appliquer aux 3 problèmes des TP précédents (CartPole, LunarLander et GridWorld)

Algorithme 1 : Algorithme PPO - version KL adaptatif

Input : Paramètres initiaux θ_0 et ϕ , KL cible δ , pas d'apprentissage α , nombre d'étapes d'optimisation K

```
1  $\beta_0 \leftarrow 1$ 
2 for  $k = 0, 1, 2, \dots$  do
3   Collecte d'un ensemble de trajectoires  $\mathcal{D}_k$  selon politique  $\pi_{\theta_k}$ 
4   Calcul des avantages  $\hat{A}^{\pi_{\theta_k}}$  pour toutes les transitions de  $\mathcal{D}_k$  selon  $TD(\lambda)$ 
5    $\theta \leftarrow \theta_k$ 
6   for  $s$  de 1 à  $K$  do
7      $\theta \leftarrow \theta + \alpha (\nabla_{\theta} \mathcal{L}_{\theta_k}(\theta) - \beta_k \nabla_{\theta} \bar{D}_{KL}(\theta_k | \theta))$ 
8     avec  $\mathcal{L}_{\theta_k}(\theta) = \frac{1}{|\mathcal{D}_k|} \sum_{(s,a) \in \mathcal{D}_k} \frac{\pi_{\theta}(a|s)}{\pi_{\theta_k}(a|s)} \hat{A}^{\pi_{\theta_k}}(s, a)$ 
9     et  $\bar{D}_{KL}(\theta_k | \theta) = \frac{1}{|\mathcal{D}_k|} \sum_{s \in \mathcal{D}_k} D_{KL}(\pi_{\theta_k}(\cdot | s) | \pi_{\theta}(\cdot | s))$ .
10  end
11   $\theta_{k+1} \leftarrow \theta$ 
12  if  $\bar{D}_{KL}(\theta_k | \theta_{k+1}) \geq 1.5\delta$  then
13     $\beta_{k+1} \leftarrow 2\beta_k$ 
14  end
15  if  $\bar{D}_{KL}(\theta_k | \theta_{k+1}) \leq \delta/1.5$  then
16     $\beta_{k+1} \leftarrow 0.5\beta_k$ 
17  end
18  Mise à jour de  $V_{\phi}$  selon  $TD(\lambda)$  sur  $\mathcal{D}_k$ 
19 end
```

Notons les K pas de gradients à chaque optimisation sur les trajectoires récoltées (plutôt que 1 avec les PG classiques).

Au fait d'après vous, que vaut le gradient de la KL au premier passage ? À quoi correspond l'algo PPO si on prend $K = 1$?

Note: on pourra comparer deux versions de la KL, $\bar{D}_{KL}(\theta_k|\theta)$ comme dans le papier original et en mode reverse $\bar{D}_{KL}(\theta|\theta_k)$

2 PPO with Clipped Objective

Implémenter l'algorithme PPO avec objectif "clippé" donné ci-dessous.

Algorithme 2 : Algorithme PPO - version clippée

Input : Paramètres initiaux θ_0 et ϕ , KL cible δ , pas d'apprentissage α , nombre d'étapes d'optimisation K

```

1  $\beta_0 \leftarrow 1$ 
2 for  $k = 0, 1, 2, \dots$  do
3   Collecte d'un ensemble de trajectoires  $\mathcal{D}_k$  selon politique  $\pi_{\theta_k}$ 
4   Calcul des avantages  $\hat{A}^{\pi_{\theta_k}}$  pour toutes les transitions de  $\mathcal{D}_k$  selon  $TD(\lambda)$ 
5    $\theta \leftarrow \theta_k$ 
6   for  $s$  de 1 à  $K$  do
7      $\theta \leftarrow \theta + \alpha \nabla_{\theta} \mathcal{L}_{\theta_k}^{CLIP}(\theta)$ 
8     avec  $\mathcal{L}_{\theta_k}^{CLIP}(\theta) =$ 
9        $\frac{1}{|\mathcal{D}_k|} \sum_{(s,a) \in \mathcal{D}_k} \min\left(\frac{\pi_{\theta}(a|s)}{\pi_{\theta_k}(a|s)} \hat{A}^{\pi_{\theta_k}}(s, a); \text{clip}\left(\frac{\pi_{\theta}(a|s)}{\pi_{\theta_k}(a|s)}, 1 - \epsilon, 1 + \epsilon\right) \hat{A}^{\pi_{\theta_k}}(s, a)\right)$ 
10    $\theta_{k+1} \leftarrow \theta$ 
11   Mise à jour de  $V_{\phi}$  selon  $TD(\lambda)$  sur  $\mathcal{D}_k$ 
12 end
```

3 Comparaison

Comparer les performances des deux versions de PPO avec A2C et une version de PPO sans KL ni clipped objective (au moins sur Cartpole et LunarLander). Pour cela, une fois des "bons" hyper-paramètres trouvés, faire tourner les algos un certain nombre de fois (e.g., 10) et tracer les points moyens de performance (e.g., toutes les 100 trajectoires collectées).

4 Bonus: Entropie

Vous pourrez considérer une version avec coût d'entropie évitant aux politiques de converger trop rapidement vers des solutions sous-optimales.