

REINFORCEMENT LEARNING & ADVANCED DEEP

M2 DAC


TME 5. Policy Gradients

Ce TME a pour objectif d'expérimenter les approches de renforcement Policy Gradients vues en cours.

1 Online A2C

Implémenter l'algorithme online actor-critic donné dans la figure ci-dessous et l'appliquer aux 3 problèmes du TP précédent (CartPole, LunarLander et GridWorld)

online actor-critic algorithm:


- 
1. take action $\mathbf{a} \sim \pi_\theta(\mathbf{a}|\mathbf{s})$, get $(\mathbf{s}, \mathbf{a}, \mathbf{s}', r)$
 2. update \hat{V}_ϕ^π using target $r + \gamma\hat{V}_\phi^\pi(\mathbf{s}')$
 3. evaluate $\hat{A}^\pi(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \gamma\hat{V}_\phi^\pi(\mathbf{s}') - \hat{V}_\phi^\pi(\mathbf{s})$
 4. $\nabla_\theta J(\theta) \approx \nabla_\theta \log \pi_\theta(\mathbf{a}|\mathbf{s}) \hat{A}^\pi(\mathbf{s}, \mathbf{a})$
 5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

À l'étape 2, la fonction V est mise à jour par un coût de Huber pour faire tendre la différence temporelle TD(0) vers 0 comme au TP précédent.

2 Batch A2C

Implémenter l'algorithme batch actor-critic donné dans la figure ci-dessous et comparer les performances avec l'algorithme précédent (au moins sur LunarLander).

batch actor-critic algorithm:

- 
1. sample $\{\mathbf{s}_i, \mathbf{a}_i\}$ from $\pi_\theta(\mathbf{a}|\mathbf{s})$ (run it on the robot)
 2. fit $\hat{V}_\phi^\pi(\mathbf{s})$ to sampled reward sums
 3. evaluate $\hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i) = r(\mathbf{s}_i, \mathbf{a}_i) + \gamma\hat{V}_\phi^\pi(\mathbf{s}'_i) - \hat{V}_\phi^\pi(\mathbf{s}_i)$
 4. $\nabla_\theta J(\theta) \approx \sum_i \nabla_\theta \log \pi_\theta(\mathbf{a}_i|\mathbf{s}_i) \hat{A}^\pi(\mathbf{s}_i, \mathbf{a}_i)$
 5. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Dans cette version, plutôt que de mettre à jour après chaque action, on attend la fin d'une trajectoire avant toute optimisation. On pourra considérer une version Roll-out Monte-Carlo (où V_t est comparé à R_t) et une version TD(0) (où V_t est comparé à $r_t + \gamma V_{t+1}$ comme dans l'algorithme précédent). Si le temps le permet, on pourra également considérer une version avec traces d'éligibilité pour la définition de la fonction d'avantage, comme dans GEA vu en cours

3 Bonus: Avantage Compatible

Développer une version d'A2C (online ou batch) qui considère une fonction d'avantage compatible (i.e., $\hat{A}^\pi = f_w$, avec $\nabla_w f_w = \nabla_\theta \log \pi_\theta$), comme discuté en cours.