

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.

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

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. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$

Plutôt que de mettre à jour après chaque action, on attend la fin d'un certain nombre de trajectoires avant toute optimisation.

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. Pour la cible $V(s')$ et pour la baseline dans la fonction d'avantage, il est conseillé d'utiliser un réseau annexe copiant les paramètres du réseau principal toutes les k itérations (par exemple 10000).

On pourra considérer une version Rollout Monte-Carlo (où V_t est comparé à R_t), une version TD(0) (où V_t est comparé à $r_t + \gamma V_{t+1}$ comme dans l'algorithme précédent) et une version TD(λ). Idem pour le calcul de l'avantage utilisé par l'acteur.

Bonus: Avantage Compatible

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