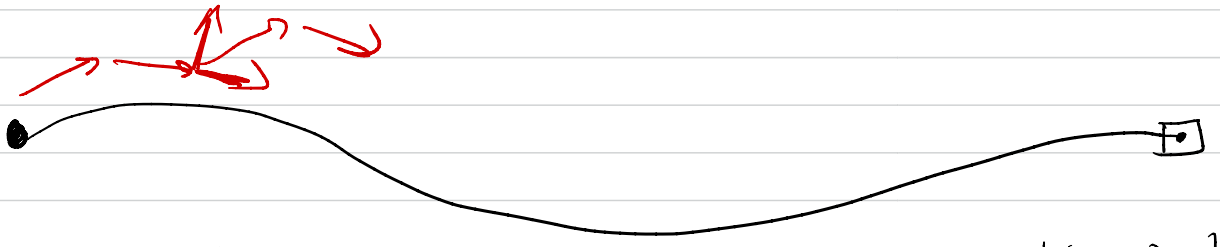
TP12 – Imitation learning

→ run LunarLander

- behavioral cloning
- GAIL = PPO (-clip) + reward ⟹ may ~~log~~ discriminateur
  - expert (vs) modèle
  - (=1) (=0)

→ utiliser .pkl

torch.nn.functional.one_hot



Behavioral learning : apprentissage supervisé sur $\{(s_1, a_1), \dots, (s_n, a_n)\}$

But : $\max_\theta \sum_{(a_i, s_i) \in \text{Expert}} \log \pi_\theta(a_i | s_i)$

$$\mathbb{E}_{s,a \sim \text{traj}(s_0, a_0)} \left[\log P(\text{expert} | s, a) \,|\, s_0, a_0 \right]$$

→ Entraînement du discriminateur : étiqueter artificiellement les (s,a) générées / réelles.

$$\begin{bmatrix} (s_1, a_1) \\ \vdots \\ (s_n, a_n) \end{bmatrix} \xrightarrow{\log D} \begin{bmatrix} r_1 \\ \vdots \\ r_n \end{bmatrix} \longrightarrow \begin{bmatrix} \sum_{i \geq 1} r_i / n \\ \vdots \\ \sum_{i \geq n-2} r_i / 3 \\ \sum_{i \geq n-1} / 2 \\ r_n / 1 \end{bmatrix}$$

$t_1, \dots, t_n \sim \pi_\theta$ (trajectoires selon π_θ, politique actuelle)

où $t_i = \left((s_1^{(i)}, a_1^{(i)}), \dots, (s_{l_i}^{(i)}, a_{l_i}^{(i)}) \right)$ où l_i longueur de la traj. i.

(1) Entraîner le discriminateur $d \in \mathbb{R}$ (logit) :

$$\mathbb{E}_{(s,a) \sim \text{Expert}} \left[\log(d(s,a)) \right] + \mathbb{E}_{(s,a) \sim T} \left[\log(1 - d(s,a)) \right]$$

$$= \text{BCEWithlogitsloss}\big(\underset{\in \text{Exp}}{(s,a)}, 1 \big) + \text{BCEWithlogitsloss}\big(\underset{\substack{\in T \\ \text{(policy actuelle)}}}{(s,a)}, 0 \big)$$

(2) Rewards :

$$T_i \to \begin{matrix} s_1^{(i)}, a_1^{(i)} \\ \vdots \\ s_{l_i}^{(i)}, a_{l_i}^{(i)} \end{matrix} \xrightarrow{\log \text{sigmoid}(d)} \begin{matrix} r_1^{(i)} \\ \vdots \\ r_{l_i}^{(i)} \end{matrix} \xrightarrow{\mathbb{E}_{s,a}[r \mid s_0, a_0]} \begin{bmatrix} \frac{\sum_{j=1}^{l_i} r_j^{(i)}}{l_i} \\ \vdots \\ \frac{\sum_{j=l_i-1}^{l_i} r_j^{(i)}}{2} \\ \sum_{j=l_i}^{l_i} r_j^{(i)} = r_{l_i}^{(i)} \end{bmatrix} := Q$$

(3) PPO avec la fonction Q.