

INF8225 TP3 H18

Date limite : ?

Alexandre Piché

1 Gradient de la Politique

- a) Derivez les gradients ($\nabla_{\theta_\mu} \log \pi(a|s, \theta)$, $\nabla_{\theta_\sigma} \log \pi(a|s, \theta)$) pour une politique gaussienne (Notez que $\nabla \log f(x) = \frac{\nabla f(x)}{f(x)}$).

$$\begin{aligned}\pi(a|s, \theta) &= \frac{1}{\sigma(s, \theta)\sqrt{2\pi}} \exp\left(-\frac{(a - \mu(s, \theta))^2}{2\sigma(s, \theta)^2}\right) \\ \mu(s, \theta) &= \theta_\mu^T x_\mu(s) \\ \sigma(s, \theta) &= \exp(\theta_\sigma^T x_\sigma(s))\end{aligned}$$

- b) Quel est le gradient de la politique avec les retours suivants:
- (i) le retour Monte Carlo $G_t = \sum_{i=0}^T \gamma^i r_{t+i}$.
 - (ii) le retour Monte Carlo G_t et la value function est $V(s_t)$ et quelle est utilisée comme variable de contrôle.
 - (iii) le retour est estimé avec $r_t + V(s_{t+1})$.
 - (iv) la trajectoire et le retour G_t viennent de la politique $\phi(a|s)$.
- c) Commentez brièvement la variance des gradients b).

2 Apprentissage avec Double Réseaux Q

Dans cette section, vous allez implémenter deux modifications de "Deep Q Network" (DQN) (Mnih et al. 2013). Implémentez "double deep Q networks (DDQN)" (Van Hasselt, Guez, and Silver 2016) et "double dueling DQN" (Wang et al. 2015) à partir du code disponible ici: <https://github.com/AlexPiche/INF8225/blob/master/tp3/DDQN.ipynb>

- a) Implémentez la fonction "act" pour qu'elle sélectionne la meilleure action avec probabilité $1 - \epsilon$ et une action au hasard avec probabilité ϵ
- b) Implémentez la fonction "backward" pour calculer les gradients du réseau Q. Utilisez "Polyak averaging" pour mettre à jour le réseau "target Q".¹
- c) Modifiez le réseau de neurones pour estimer l'avantage ($A(s, a)$) et la valeur de la situation $V(s)$.

References

- [1] Volodymyr Mnih et al. "Playing atari with deep reinforcement learning". In: *arXiv preprint arXiv:1312.5602* (2013).
- [2] Hado Van Hasselt, Arthur Guez, and David Silver. "Deep Reinforcement Learning with Double Q-Learning." In: *AAAI*. Vol. 16. 2016, pp. 2094–2100.
- [3] Ziyu Wang et al. "Dueling network architectures for deep reinforcement learning". In: *arXiv preprint arXiv:1511.06581* (2015).

¹Polyak averaging: $\phi' \leftarrow \tau\phi' + (1 - \tau)\phi$.