Deep Reinforcement Learning

Christophe Eloy

Outline

- Value function approximation
    - Implementation of a temporal difference algorithm with neural network
    - Batch methods
- Policy gradient
    - Objective functions
    - Score function
    - Policy gradient
    - Actor-critic algorithms

Value function approximation

- We use an approximation of the action (resp. state) value function

$$\hat{q}_\theta(s, a) \approx q_\pi(s, a)$$

- Minimization of the cost: $J(\theta) = \mathbb{E}_\pi \left[\frac{1}{2} \left(q_\pi(s, a) - \hat{q}_\theta(s, a) \right)^2 \right]$

- Stochastic gradient descent algorithm (SARSA)

$$\theta \leftarrow \theta + \alpha \left(r + \gamma \hat{q}_\theta(s', a') - \hat{q}_\theta(s, a) \right) \nabla_\theta \hat{q}_\theta(s, a)$$

Policy gradient

- Objective functions

$$J_1(\theta) = \mathbb{E}_{\pi_\theta}\left[v(s_1)\right]$$

$$J_{\mathrm{avV}}(\theta) = \mathbb{E}_{\pi_\theta}\left[v(s)\right]$$

$$J_{\mathrm{avR}}(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_a \pi(s,a) R_s^a\right]$$

- Score function: $\nabla_\theta \log \pi(s,a)$

- Policy gradient

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

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\left[\nabla_\theta \log\left(\pi(s,a)\right)\ Q_{\pi_\theta}(s,a)\right]$$

Advantage actor-critic

- The critic approximates the value function (SARSA)

$$\hat{\theta} \leftarrow \hat{\theta} + \alpha \left(r + \gamma \hat{v}_{\hat{\theta}}(s') - \hat{v}_{\hat{\theta}}(s) \right) \nabla_{\hat{\theta}} \hat{v}_{\hat{\theta}}(s)$$

- The actor updates in the direction suggested by the critic (policy-gradient)

$$\theta \leftarrow \theta + \alpha \nabla_{\theta} \log \left(\pi(s, a) \right) A_{\pi_\theta}(s, a)$$

- Where the advantage function is

$$A_{\pi_\theta}(s, a) = Q_{\pi_\theta}(s, a) - V_{\pi_\theta}(s) \approx r + \gamma \hat{v}_{\hat{\theta}}(s') - \hat{v}_{\hat{\theta}}(s)$$