Class 10

Deep Reinforcement Learning

Christophe Eloy

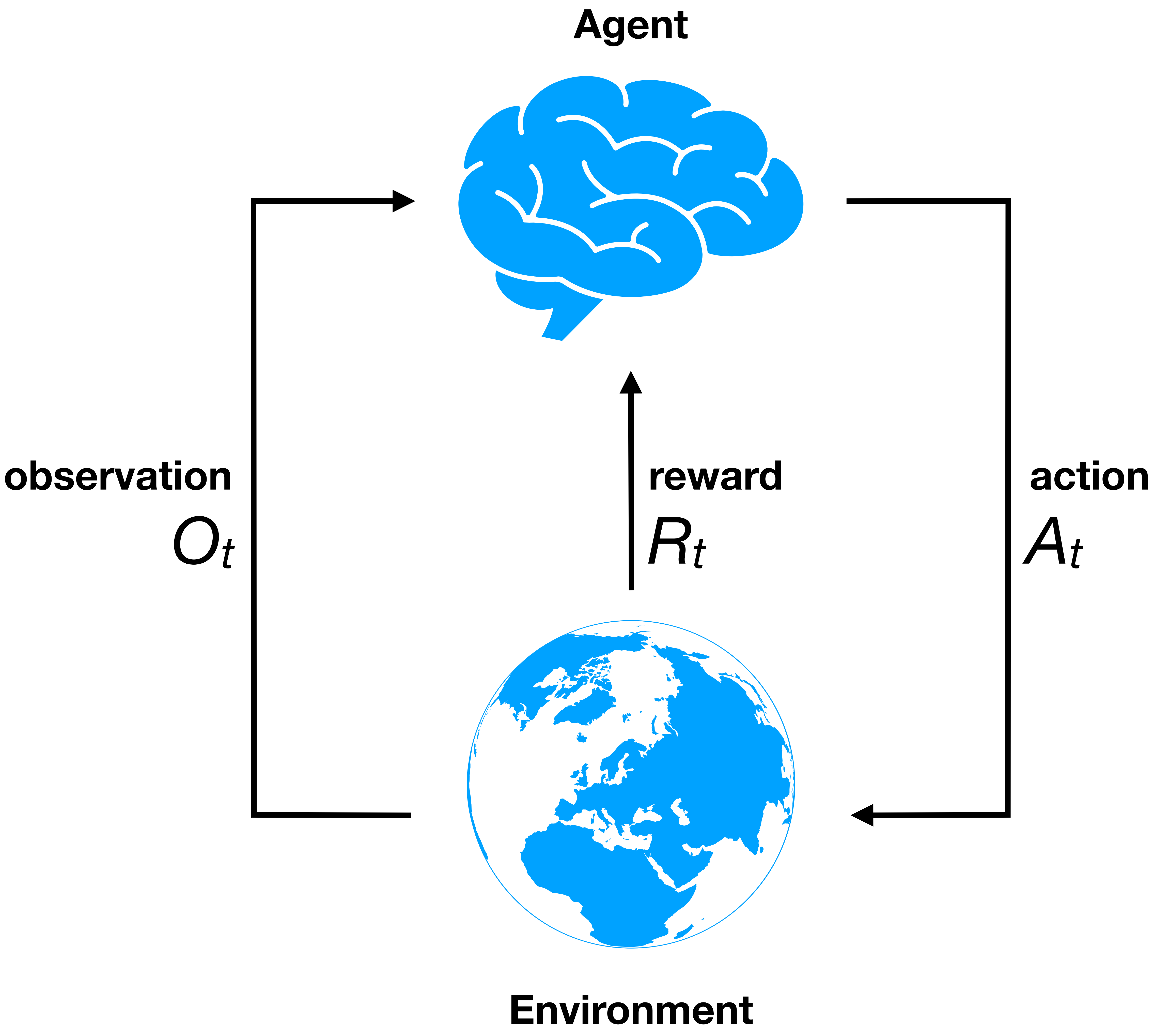
Agent interacts with environment

At each step t

- The agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t

- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}

Time t increments at each step



Components of a RL agent

- **Policy** is the function that pick agent's action as a function of its state

$$a = \pi(s) \qquad \text{(deterministic)}$$

$$\pi(a \mid s) = \mathbb{P}\left[A_t = a \mid S_t = s\right] \qquad \text{(stochastic)}$$

- **Value function** is a prediction of future (discounted) rewards

$$v_\pi(s) = \mathbb{E}_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s\right]$$

- A **model** predicts what the environment will do next

$$\mathcal{P}^a_{ss'} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right] \qquad \text{(predicts next state)}$$

$$\mathcal{R}^a_s = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right] \qquad \text{(predicts next reward)}$$

Policy and value functions

- Stochastic policy

$$\pi(a \mid s) = \mathbb{P}\left[A_t = a \mid S_t = s\right]$$

- Return: $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$

- State-value function

$$v_\pi(s) = \mathbb{E}_\pi\left[G_t \mid S_t = s\right]$$

- Action-value function (Q-function)

$$q_\pi(s, a) = \mathbb{E}_\pi\left[G_t \mid S_t = s, A_t = a\right]$$

Model-free prediction

- Dynamic programming
- Iterative procedure to approach state-value function, given a known policy
- Monte-Carlo algorithm (high variance, no bias)

$$V(S_t) \leftarrow V(S_t) + \alpha \left(G_t - V(S_t) \right)$$

- Temporal difference algorithm (lower variance, some bias)

$$V(S_t) \leftarrow V(S_t) + \alpha \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right)$$

Theorem: MC and TD algorithm converges towards $v_\pi(s)$

Model-free control

- Iterative procedure to approach Q-function, given sequences S, A, R, S', A'

- SARSA algorithm (on-policy)

$$Q(S,A) \leftarrow Q(S,A) + \alpha \left(R + \gamma Q(S',A') - Q(S,A) \right)$$

- Q-learning algorithm (off-policy)

$$Q(S,A) \leftarrow Q(S,A) + \alpha \left(R + \gamma \max_{a'} Q(S',a') - Q(S,A) \right)$$

Theorem: SARSA and Q-learning converge towards q*

DP, MC and TD

- Dynamic programming, Monte Carlo, temporal difference

- State-value function

$$V(S) \leftarrow \mathbb{E}\left[R + \gamma V(S') \,|\, S\right] \qquad \text{DP (iterative policy evalution)}$$

$$V(S) \leftarrow V(S) + \alpha\left(G - V(S)\right) \qquad \text{MC}$$

$$V(S) \leftarrow V(S) + \alpha\left(R + \gamma V(S') - V(S)\right) \qquad \text{TD}$$

- Q-function

$$Q(S,A) \leftarrow \mathbb{E}\left[R + \gamma Q(S',A') \,|\, S,A\right] \qquad \text{DP}$$

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left(G - Q(S,A)\right) \qquad \text{MC}$$

$$Q(S,A) \leftarrow Q(S,A) + \alpha\left(R + \gamma Q(S',A') - Q(S,A)\right) \qquad \text{TD (SARSA algorithm)}$$

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

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

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

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

- 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

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