Chapitre 5 : Q-Learning

- Q-Learning = principal algorithme de RL
- permet d'apprendre la politique optimale d'une MDP sans connaître ni Φ, ni Γ.
- l'apprentissage est réalisé par interaction avec l'environnement.

* ——— *

Ce qu'on cherche à apprendre :
les paires (état, action) de la politique optimale.

Deux ingrédients :

- Equation de Bellman pour la fonction q^*
- Différence Temporelle

Eq. d'optimalité de Bellman

pour Q :

$$q^*(s,a) = \sum_{s' \in S} P(s'|s,a)\left(r(s'|s,a) + \gamma \max_{a' \in A} q^*(s',a')\right)$$

Comme pour l'algo. TD, on considère une trajectoire (indicé par t) et avec une estimation $\hat{q}$ de q^*, le terme $r_t + \max_{a'} \hat{q}(s_{t+1}, a') - \hat{q}(s_t, a_t)$ est une différence temporelle que l'on peut utiliser comme correction pour $\hat{q}(s_t, a_t)$.

L'algorithme peut alors être décrit par la formule :

$$q_{t+1}(s_t, a_t) = q_t(s_t, a_t) + \alpha_t(s_t, a_t)\left[r_{t+1} + \gamma \max_a q_t(s_{t+1}, a) - q_t(s_t, a_t)\right]$$

Algorithme (Q-Learning):

Input : S, A, γ

$\hat{q}(s,a) \leftarrow 0 \quad \forall s,a \in S \times A$

$n(s,a) \leftarrow 0 \quad \forall s,a \in S \times A$

Répéter

initialiser l'état initial s_0

$t \leftarrow 0$

tant que Episode non terminé **faire**

sélectionner l'action a_t et l'émettre

observer r_t et s_{t+1}

$$\hat{q}(s_t,a_t) \leftarrow \hat{q}(s_t,a_t) + \alpha(s_t,a_t,n(s_t,a_t)) \left[r_t + \gamma \max_{a'} \hat{q}(s_{t+1},a') - \hat{q}(s_t,a_t) \right]$$

$n(s_t,a_t) \leftarrow n(s_t,a_t)+1$

$t \leftarrow t+1$

fin tant que

jusqu'à critère d'arrêt vérifié

Un exemple : Retour au Grid World.

On reprend la classe Game.

→ mais nous fermet cette-fois-si de récupérer la récompense

SARSA : d'un autre algorithme d'apprentissage mais on-policy (l'action utilisée dans la mise à jour de q est celle qui est exécutée)