1 Markov Decision Process (MDP)

D'après Wikipédia: Un processus de décision markovien est un processus de contrôle stochastique discret. À chaque étape, le processus est dans un certain état **s** et l'agent choisit une action **a**. La probabilité que le processus arrive à l'état **s'** est déterminée par l'action choisie. Plus précisément, elle est décrite par la fonction de transition d'états **T(s,a,s')**. Donc, l'état **s'** dépend de l'état actuel **s** et de l'action **a** sélectionnée par le décideur. Cependant, pour un état **s** et une action **a**, le prochain état est indépendant des actions et états précédents. On dit alors que le processus satisfait la propriété de Markov.

Quand le processus passe de l'état **s** à l'état **s'** avec l'action **a**, l'agent gagne une récompense **r**.

2 Q-Learning

Le Q-Learning, basé sur le processus de décision markovien, est une technique d'apprentissage automatique utilisée en intelligence artificielle, plus particulièrement en apprentissage par renforcement.

2.1 Q-Function

La Q-Function est la base du Q-Learning. Pour un etat **s** et une action **a**, elle donne la recompense esperée **Q(s,a)**.

2.2 Policy

La Politique π a l'etat **s** est la facon de choisir l'action **a** qui maximise la recompense esperée **Q(s,a)**. C'est a dire, si on est a l'etat **s** et qu'on choisi l'action **a** selon la politique π, alors, on aura une recompense optimale.

$$\pi(s) = argmax_a(Q(s,a))$$

2.3 Rewards

La recompense esperée R_t a l'instant t pour un etat s_t et une action a_t est la somme des recompenses futures.

$$R_t = Q(s_t,a_t)$$

$$R_t = r_t + r_{t+1} + r_{t+2} + r_{t+3} + ... + r_n$$

Mais, sachant que le processus est stochastique, alors plus on va dans le future, moins les recompenses sont evidentes, ainsi, on introduit un **facteur de discontinuite** γ.

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + ... + \gamma^{n-t} r_n$$

Ainsi,

$$\begin{aligned} R_t &= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + ... + \gamma^{n-t} r_n \\ R_t &= r_t + \gamma(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... + \gamma^{n-(t+1)} r_n) \\ R_t &= r_t + \gamma R_{t+1} \end{aligned} \tag{1}$$

2.4 Optimal Reward

La recompense à l'instant t et à l'etat s_t, en choidissant l'action a_t suivant la polique π est donc optimale et a la valeur $Q^\pi(s_t,a_t)$.

$$Q^\pi(s_t,a_t) = r_t + \gamma max_{a_{t+1}} Q(s_{t+1},a_{t+1})$$

On peut aussi ecrire:

$$Q(s_t,a_t) = r_t + \gamma max_{a_{t+1}} Q(s_{t+1},a_{t+1}), \quad a_t = \pi(s_t)$$

Note,

$$\begin{aligned} Q^\pi(s_t,a_t) &= max R_t \\ &= max(r_t + \gamma R_{t+1}) \\ &= r_t + \gamma max(R_{t+1}) \\ Q^\pi(s_t,a_t) &= r_t + \gamma max_{a_{t+1}} Q(s_{t+1},a_{t+1}) \end{aligned} \tag{2}$$

2.5 Loss

$$Q(s,a) = max_{a'} R(s,a')$$
$$Q(s_t,a_t) = r_t + \gamma max_{a_{t+1}} Q(s_{t+1},a_{t+1})$$

En effet,

$$\begin{aligned} R_t &= r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + ... + \gamma^{n-t} r_n \\ &= r_t + \gamma(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + ... + \gamma^{n-(t+1)} r_n) \\ &= r_t + \gamma R_{t+1} \end{aligned}$$

$$\begin{aligned} maxR_t &= max(r_t + \gamma R_{t+1}) \\ &= r_t + \gamma max(R_{t+1}) \end{aligned} \tag{3}$$

$$Q(s_t,a_t) = r_t + \gamma max_{a_{t+1}} Q(s_{t+1},a_{t+1})$$

A l'instant **t**, le processus etant a l'etat s_t, en choisissant l'action **a**, le processus arrive a l'etat s_{t+1} et gagne une recompense R_t.

$$R_t = r_t + r_{t+1}$$

Il utilise le modèle MDP a n-etapes (fini) telle pour un état **s** et un action **a**, **Q(s,a)** représente l'optimum des récompenses futurs.

$$loss = (\underbrace{r + \gamma \max_{a'} Q'(s',a')}_{target} - \underbrace{Q(s,a)}_{prediction})^2$$