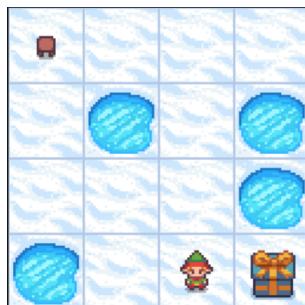


IHDCM037 – Machine Learning

Travaux Pratiques

Séance 6 : Reinforcement Learning

Cette séance est consacrée au Reinforcement Learning, et plus particulièrement au Q-learning. Nous allons étudier cette méthodologie pour la navigation dans un monde en 2D, l'environnement Frozen Lake de la librairie gymnasium. De la documentation est disponible sur https://gymnasium.farama.org/environments/toy_text/frozen_lake/.



Première partie

Initialisez votre environnement et votre Q-table :

1. Chargez l'environnement Frozen Lake avec la commande `gymnasium.make('FrozenLake-v1')`. Utilisez les paramètres `is_slippery=False` et `render_mode='ansi'`.
2. Initialisez un Q-table de dimension *nombre d'états possibles x nombre d'actions possibles*. Le tableau est au départ rempli de 0.
3. Définissez les paramètres alpha, gamma, et le nombre d'épisodes d'entraînement ($0 < \alpha < 1$, $0 < \gamma < 1$).

Deuxième partie

Créez une boucle d'entraînement pour votre environnement, implémentant l'équation du Q-learning :

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right]$$

Pour chaque épisode :

1. Remettez l'environnement au point de départ avec `env.reset()`

2. Tant que l'épisode n'est pas fini, choisissez la prochaine action optimale à prendre :
 - a. Si toutes les actions ont une valeur égale pour Q, choisissez une action au hasard
 - b. Sinon, choisissez l'action avec la plus haute valeur de Q
3. Effectuez votre action sur l'environnement avec `env.step(action)`, et updatez votre Q-table

Troisième partie

Observez les résultats de plusieurs manières différentes :

1. Exécutez un épisode avec votre stratégie apprise, et observez cet épisode directement sur l'environnement en format texte avec `env.render()`
2. Mesurez le succès de votre stratégie sur une série d'itérations
3. Visualisez la stratégie optimale pour chaque état avec la fonction `plot_Q` fournie sur WebCampus

Variez les paramètres alpha et gamma et expliquez vos observations. Appliquez cette méthodologie sur le même environnement avec le paramètre `is_slippery=True` qui rend le déplacement stochastique (1 chance sur 3 d'aller dans la direction souhaitée, et 1 chance sur 3 d'aller dans chacune des directions perpendiculaires à celle souhaitée).