Comparaison d’algorithmes d’apprentissage par renforcement dans le cadre du taxi driver.

Introduction

L’apprentissage par renforcement émerge en 1957 lorsque Bellman introduit le processus de décision markovien. L’objectif est alors de mettre au point un contrôleur qui minimise au cours du temps une mesure donnée du comportement d’un système dynamique. Un processus markovien est un modèle stochastique où un agent prend des décisions et où les résultats de ses actions sont aléatoires.
Un agent dans un tel modèle suivra une trajectoire correspondant à l’enchaînement des états du modèle et des actions réalisées par l’agent. La politique de l’agent permettra donc de réaliser les actions maximisant les gains de l’environnement, et c’est sur cette politique que vient jouer l’algorithme d’apprentissage par renforcement.
L’objectif de cette étude consiste en la détermination du meilleur algorithme actuel dans la résolution du problème du taxi driver de l’environnement gym (google).

Matériels et méthodes

Notre expérience de comparaison s’est réalisée sous python à l’aide de la librairie gym, dans l’environnement taxi-v3. Nous avons intégré et étudié les algorithmes suivant, le Q-learning, le SARSA et le Deep-Q-Learning. L’expérience contrôle est réalisée par une politique de parcours systématique de l’espace de l’environnement.

Les mesures effectuées afin de comparer les résultats sont le nombre de pas réalisés par l’algorithme avant d’obtenir la récompense, la valeur de la récompense et l’évolution de la capacité exploratoire du modèle (epsilon).

Résultats

Le modèle Q-Learning

L’entraînement du modèle Q-Learning a été réalisé à l’aide des paramètres suivants :

- *Number of episodes*: `25 000,`,
- *Learning Rate*: `0.01`,
- *Gamma*: `0.99`,
- *Starting Epsilon*: `1`,
- *Ending Epsilon*: `0.001`,
- *Epsilon Decay Per Episode*: `0.01`

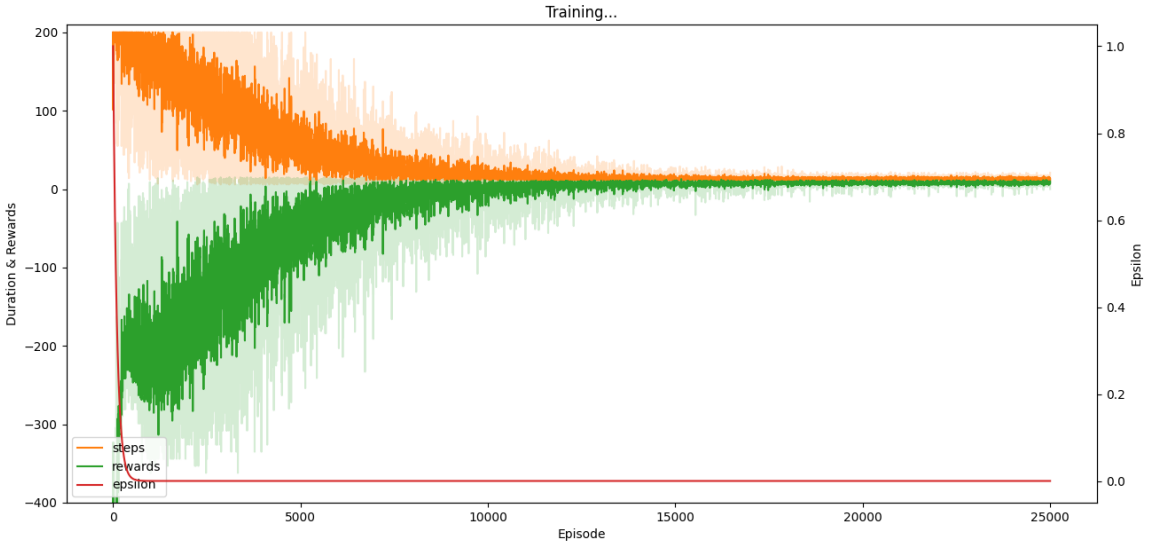

Figure 1: Mesure epsilon, nombre de pas par époque et valeur de la récompense lors de l'entraînement du modèle Q-Learning

Nous observons sur le résultat de ce modèle que que le nombre de pas nécessaires à la réalisation d'une l'épisode a des valeurs comprises entre 50 et 200 à l'épisode 0, puis atteint des valeurs comprises entre 0 et 25 à l'épisode 10 000. La récompense est comprise entre -400 et -50 à l'épisode 0 puis est comprise entre 0 et -50 à l'épisode 10 000. Enfin la valeur epsilon est à 1 à l'épisode 0 et atteint 0 à l'épisode 2500.

Le modèle Deep Q Learning 1

L'entraînement du modèle Deep Q Learning 1 a été réalisé à l'aide des paramètres suivants :

- *Episodes*:`10 000`,
- *Batch Size*:`32`,
- *Gamma*:`0.99`,
- *Starting Epsilon*:`1`,
- *Ending Epsilon*:`0.1`,
- *Decay Factor Epsilon*:`400`,
- *Number of episodes between model update*: `20`,
- *Max number of steps per episodes*: `100`,
- *Warmup Episodes*: `10`,
- *Starting Learning Rate*: `0.001`,
- *Ending Learning Rate*: `0.0001`,
- *Decay Factor Learning Rate*: `5 000`,
- *Memory Size*: `50 000`,

Figure 2: Mesure epsilon, nombre de pas par époque et valeur de la récompense lors de l'entraînement du modèle Deep-Q-Learning 1

Nous observons sur le résultat de ce modèle que que le nombre de pas nécessaires à la réalisation d'une l'épisode a des valeurs comprises entre 50 et 200 à l'épisode 0, puis atteint des valeurs comprises entre 0 et 25 à l'épisode 60 000. La récompense est comprise entre -400 et -50 à l'épisode 0 puis est comprise entre 0 et -50 à l'épisode 60 000 jusqu'à la fin. Enfin la valeur epsilon est à 1 à l'épisode 0 et atteint 0,05 à l'épisode 1700. Nous observons régulièrement des pics à -150 après l'épisode 8000.

Le modèle Deep Q Learning 2

L'entraînement du modèle Deep Q Learning 2 a été réalisé à l'aide des paramètres suivants :

- *Episodes*:`10 000`,
- *Batch Size*:`128`,
- *Gamma*:`0.99`,
- *Starting Epsilon*:`1`,
- *Ending Epsilon*:`0.1`,
- *Decay Factor Epsilon*:`400`,
- *Number of episodes between model update*: `20`,
- *Max number of steps per episodes*: `100`,
- *Warmup Episodes*: `10`,
- *Starting Learning Rate*: `0.001`,
- *Ending Learning Rate*: `0.0001`,

- *Decay Factor Learning Rate*: `5 000`,
- *Memory Size*: `50 000`,

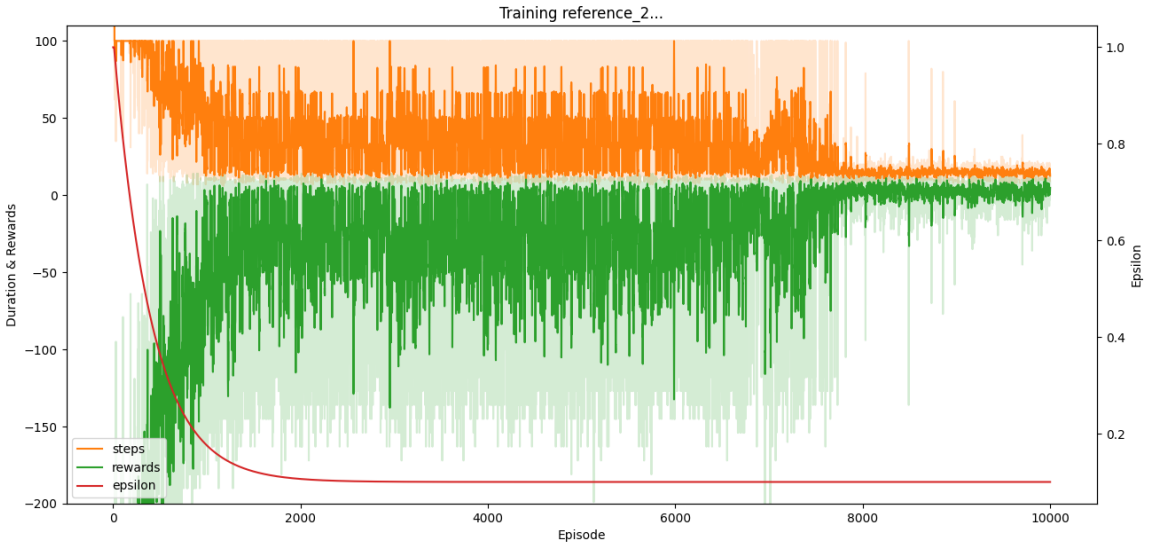

Figure 3: Mesure epsilon, nombre de pas par époque et valeur de la récompense lors de l'entraînement du modèle Deep-Q-Learning 2

Nous observons sur le résultat de ce modèle que que le nombre de pas nécessaires à la réalisation d'une l'épisode a des valeurs comprises entre 50 et 200 à l'épisode 0, puis atteint des valeurs comprises entre 0 et 100 à l'épisode 1000 puis 0 et 25 à 8000. La récompense est comprise entre -200 et -100 à l'épisode 0 puis est comprise entre -150 et 0 à l'épisode 1000 puis entre -25 et 0 à l'épisode 8000. Enfin la valeur epsilon est à 1 à l'épisode 0 et atteint 0,05 à l'épisode 1700. Nous observons quelques pics à -100 après l'épisode 8000.

Le modèle SARSA

L'entraînement du modèle SARSA a été réalisé à l'aide des paramètres suivants :

- *Episodes*: `10 000`,
- *Alpha*: `0.85`,
- *Gamma*: `0.99`,
- *Starting Epsilon*: `1`,
- *Ending Epsilon*: `0.001`,
- *Epsilon Decay per Episode*: `0.01`

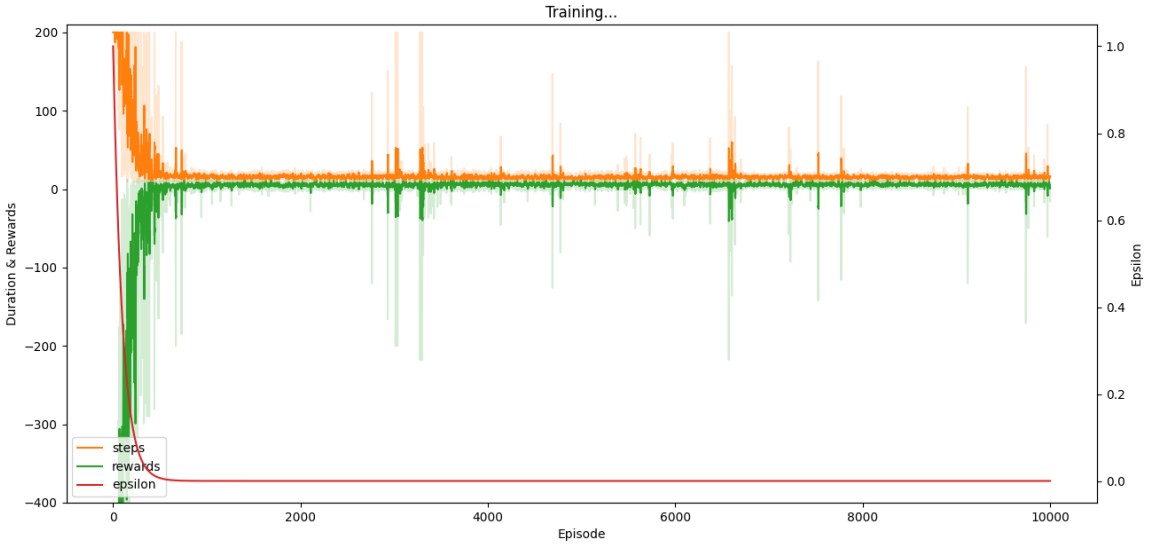

Figure 4: Mesure epsilon, nombre de pas par époque et valeur de la récompense lors de l'entraînement du modèle SARSA

Nous observons sur le résultat de ce modèle que que le nombre de pas nécessaires à la réalisation d'une l'épisode a des valeurs comprises entre 50 et 200 à l'épisode 0, puis atteint des valeurs comprises entre 0 et 25 à l'épisode 500. La récompense est comprise entre -400 et -50 à l'épisode 0 puis est comprise entre 0 et -50 à l'épisode 500. Enfin la valeur epsilon est à 1 à l'épisode 0 et atteint 0 à l'épisode 500.

Des valeurs de récompense faibles sont observées, pics entre -200 et -100 sont situé entre les épidosdes 3000 et 4000, 2 pics de même valeur entre les épisodes 4000 et 5000, 4 pics entre 6000 et 8000 puis 3 entre 9000 et 1000.

Discussion

Nous pouvons remarquer que le modèle SARSA est le modèle qui converge le plus rapidement vers une solution optimale, cependant nous observons des échecs régulièrement. La solution proposée par SARSA ne semble pas constante.

L'algorithme Q-Learning converge moins rapidement vers une solution qui semble cependant constante, aucune erreurs n'apparaît après la convergence du modèle vers la solution.

L'algorithme de Deep-Q-Learning 1 converge lentement vers la solution pour atteindre un comportement de résolution qui varie de manière importante, avec des erreurs importante.

L'algorithme de Deep-Q-Learning 2 converge soudainement à l'épisode 8000, il semble proposer une bonne solution qui observe moins de survenue d'erreurs que le modèle DQL 1.

En somme, le modèle SARSA semble être le modèle le plus rapide à converger, en 500 épisodes, avec les meilleurs résultats (une récompense supérieur à -25). Cependant l'hétérogénéité de ses résolutions pose problème dans une situation où ce modèle serait utilisé dans un environnement où ces écarts posent problème.

Le modèle Q-Learning semble être une bonne alternative au modèle SARSA étant donné qu'il semble plus constant dans l'ensemble de ses résolutions une fois entraîné. Cependant il prendra plus de temps à entraîner (10000 épisodes) et pour des résolutions offrant récompenses généralement similaires au SARSA.

Les modèles de Deep-Q-Learning ne semblent pas particulièrement intéressant dans cet environnement, ils mettent plus de temps à s'entraîner, et ne semblent pas proposer de résolutions constantes. Sans compter qu'entraîner un modèle de deep learning est plus contraignant en termes de ressources computationnelles que les autres algorithmes de machine learning.

Conclusion

Nous ne saurions donc que recommander le SARSA ou le Q-Learning. Cependant le Q-Learning semble sans doute plus intéressant de part la constance de ses résolutions. Dans un cadre où l'erreur serait coûtante dans cette situation, le Q-Learning semble donc être l'algorithme de choix, dans cet environnement.