Université de Strasbourg

Faculté des sciences économiques et de gestion
Université de Strasbourg

Master 2 Analyse et Politique Économique – DS2E
Semestre 1

UE Machine Learning – Reinforcement Learning

Solving Cliff Walking with Reinforcement Learning

Sous la responsabilité de M. Bertrand Koebel, Professeur de Sciences Économiques à la Faculté des Sciences Économiques et de Gestion
Université de Strasbourg

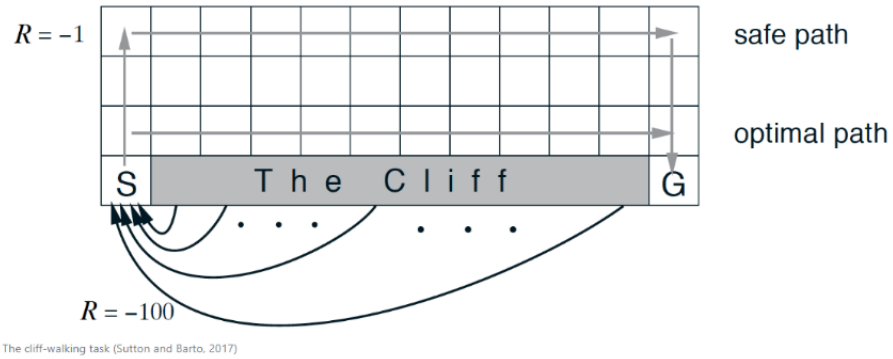
Zachary JANATI, Chafi KHERMOUCHE

1. Description de l'environnement

La "Marche sur les falaises" représente un scénario classique défini par Sutton et Barto en 2018, dans lequel un agent aspire à naviguer depuis la tuile en bas à gauche jusqu'à la tuile en bas à droite. L'objectif de l'agent est de minimiser le nombre de pas tout en évitant la zone périlleuse, symbolisée par la falaise. Un épisode de ce scénario se conclut soit lorsque l'agent marche hors des limites sur la falaise, engendrant une grande récompense négative, soit lorsqu'il atteint la tuile cible, générant une récompense positive.

Les tuiles dans cet environnement peuvent être catégorisées comme suit :

- Tuile sûre : Dans cette configuration, l'agent peut avancer en toute sécurité sans risque de conséquences néfastes.
- Tuile dangereuse (la falaise) : Cette zone constitue un danger potentiel, entraînant une situation où l'agent serait bloqué de manière permanente, mettant ainsi fin au jeu.



The cliff-walking task (Sutton and Barto, 2017)

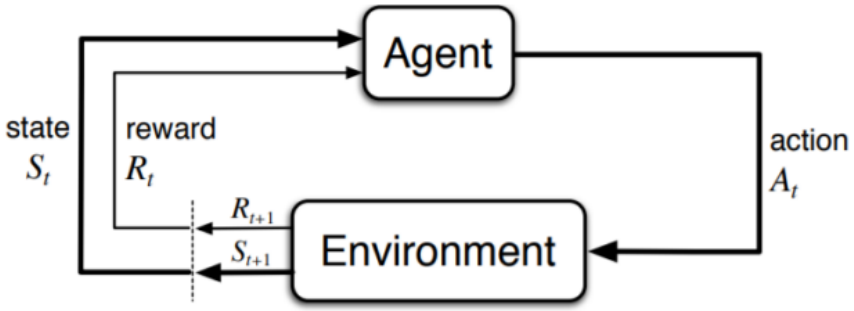
L’environnement est défini par une grille de 4x12 carreaux, où chaque carreau est identifié par une lettre spécifique

L'agent a 4 actions possibles :

- ➔ Aller à GAUCHE
- ➔ Aller en BAS
- ➔ Aller à DROITE
- ➔ Aller en HAUT

2. Interaction entre l’environnement et l’agent (actions)

L'apprentissage par renforcement est un concept central en machine learning, où l'apprentissage se produit à travers un processus itératif de tentatives et d'erreurs. Une manière de visualiser ce concept est à travers un diagramme qui présente deux acteurs principaux : l'agent (comme un marcheur) et l'environnement (comparable à une montagne avec des falaise).



L'agent observe son environnement et utilise ces observations, ainsi que des récompenses le cas échéant, pour prendre des décisions. L'objectif premier de l'agent est de maximiser la récompense totale attendue. Dans ce contexte, imaginons la marche comme un jeu, où l'objectif de l'agent est de passer de l'état de départ (D) à l'état de victoire (V) sans tomber dans les états de piège (T). Chaque état représente une position possible sur la surface, et pour chacun de ces états, l'agent peut choisir parmi quatre actions : aller à gauche, en bas, à droite ou en haut.

L'apprentissage dans ce contexte implique de déterminer quelle action choisir dans chaque état. Pour ce faire, il est utile d'attribuer une valeur de qualité (Q-value) à chaque action pour chaque état. Au total, il y a 48 états (16 x 4) et 4 actions possibles, ce qui signifie que l'agent doit calculer 48 x 4 = 192 valeurs de Q. Pour organiser ces informations, une table peut être utilisée, où les lignes correspondent aux états et les colonnes aux actions. Chaque cellule de cette table contient la valeur Q(s, a), qui représente la qualité de l'action a dans l'état s. Une valeur de Q élevée indique que l'action est plus prometteuse, tandis qu'une valeur de Q basse indique qu'elle est moins recommandée.

Lorsque l'agent se trouve dans un état et doit prendre une décision, il peut consulter cette table pour déterminer quelle action a la valeur de Q la plus élevée. Cependant, il est important de noter que choisir systématiquement l'action avec la valeur de Q la plus élevée n'est pas nécessairement la stratégie optimale. Parfois, l'exploration d'autres actions, même celles avec des valeurs de Q plus basses, peut être bénéfique, car cela peut conduire à de meilleures découvertes à long terme. L'apprentissage par renforcement vise à équilibrer l'exploitation (choisir la meilleure action connue) et l'exploration (essayer de nouvelles actions) pour atteindre l'objectif de maximisation des récompenses cumulées attendues.

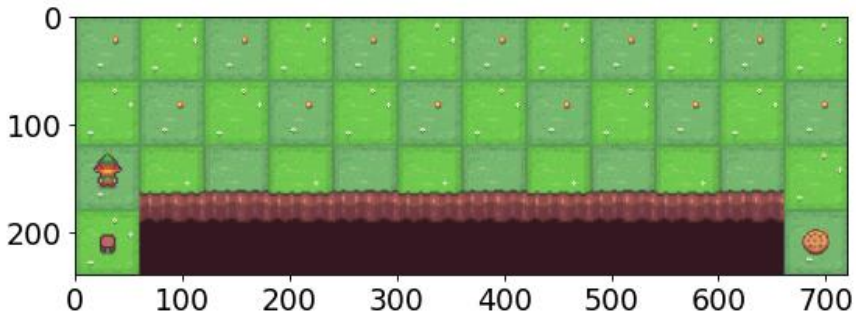
Le tableau ci-dessous est complémentaire et permet de bien visualiser le chemin que l'agent cherchera à prendre, avec deux « chemins » : un chemin risqué et un chemin optimal.

Le chemin optimal est celui qui maximise les récompenses de l'agent tout en minimisant les pénalités, le conduisant de manière sûre et efficace de la position de départ à la position de but sans tomber de la falaise. En revanche, un chemin risqué pourrait comporter des déplacements qui passent près de la falaise ou qui nécessitent des actions plus hasardeuses, augmentant le risque de tomber et de recevoir des pénalités importantes, ce qui se traduit par une série de valeurs d'état plus négatives et une stratégie globalement moins efficace.

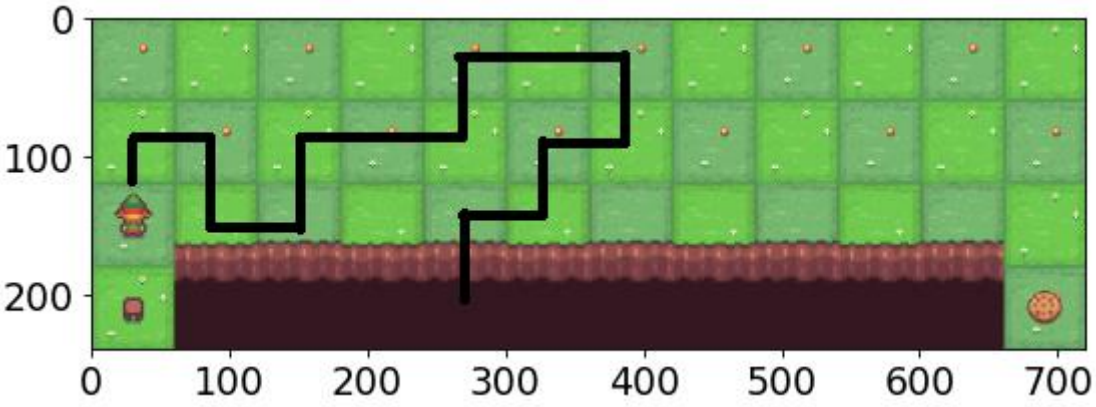
3. Le modèle non déterministe

Lorsque nous envisageons de traduire les actions de chaque état de la table ci-dessus en termes de probabilités, nous constatons que, pour chaque action, il y a une probabilité équivalente de 1/3 (environ 0.33) de se retrouver dans le carreau souhaité. Cela s'explique par la présence d'une possibilité de glisser et de se retrouver dans un endroit non souhaité, représentant une probabilité de 2/3 (environ 0.67). Dans de tels cas, on qualifie le modèle de non déterministe, indiquant une incertitude quant à l'issue de l'action entreprise, car il existe une chance non négligeable de se retrouver dans une situation non prévue (probabilité p=2/3). Ainsi,
Par exemple, l'agent commence sur l'état « Départ » au début, et si l'agent choisit l'action « en haut » puis « à droite », nous aurons les probabilités suivantes dans le modèle non déterministe :

- ➔ P(En bas) = ¼ = P(Perte de la partie)
- ➔ 1-P(En bas) = P(Se déplacer ailleurs) = ¾



De manière logique, il est observé qu'une des solutions pour atteindre l'état de "victoire" (V) consiste à effectuer une monté, suivis de dix déplacements vers la droite, et enfin d'une descente (politique 1). Cependant, en raison de la nature non déterministe du modèle, cela entraînera une probabilité de succès, notée p(V=1|politique 1), équivalente à (1/3)^2 x (1/4)^10 (0.4% ce qui est très faible) , si l'on suit les actions spécifiées précédemment par la politique 1.



Exemple de situation possible dans le cadre du modèle non déterministe

Comme nous pouvons le remarquer ci-dessus dans le modèle non déterministe, l'agent tombe dans la falaise malgré qu'on lui ait indiqué de suivre la procédure par la politique 1 puisque pour une action imposée à l'agent, il a une probabilité plus grande (3/4) de se retrouver dans un carreau non souhaité.

4. Le modèle déterministe

Dans un modèle déterministe, les actions prescrites par une politique sont exécutées avec une probabilité de 1, ce qui signifie qu'elles sont certaines. Cette certitude découle du fait que chaque action entreprise par l'agent mène de manière invariable à un état spécifique, conformément aux règles définies par le modèle. Dans le contexte de la politique 1, l'agent suit les directives avec une certitude totale, se déplaçant de manière précise vers les emplacements spécifiés jusqu'à ce qu'il atteigne l'état de victoire (s=V).

En d'autres termes, dans un modèle déterministe, l'agent a une connaissance parfaite des conséquences de ses actions, et les résultats sont prédéterminés à partir de l'état initial. Cette prévisibilité renforce la certitude associée à l'exécution des actions prescrites par la politique en question, ce qui contraste avec les modèles non déterministes où l'issue des actions peut varier en raison de l'incertitude introduite par des éléments aléatoires.

- Q-LEARNING

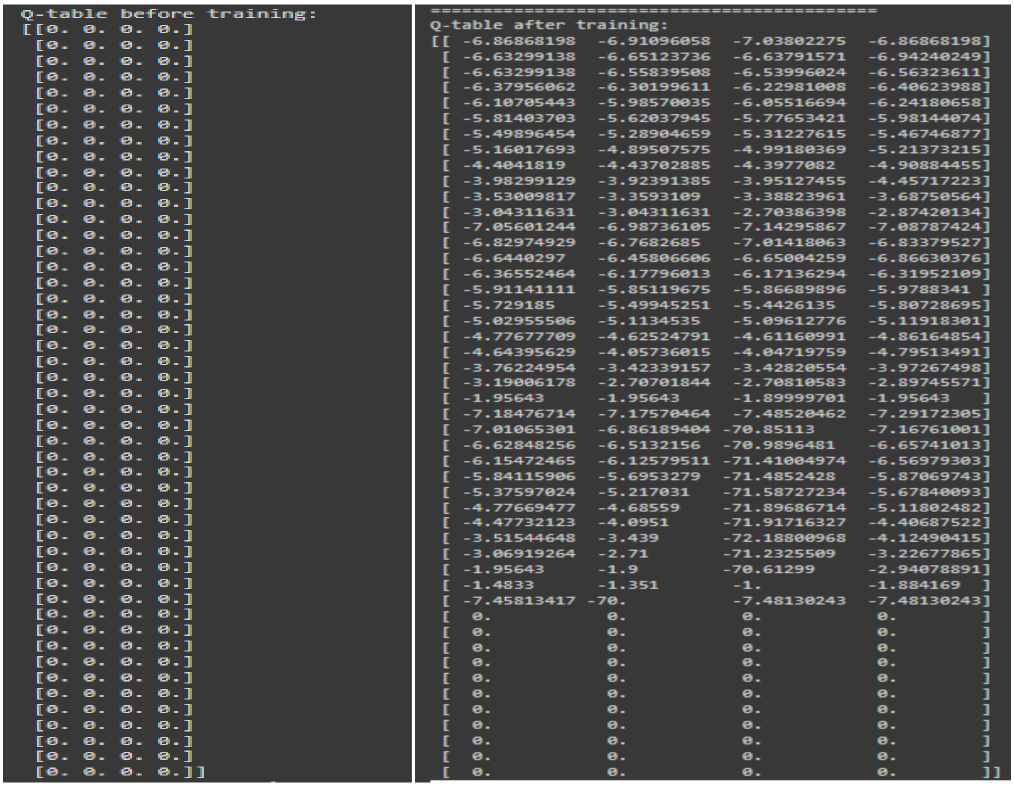
Supposons que notre agent a réussi à atteindre l'état de victoire (s=V). La question se pose de savoir comment répercuter cette information jusqu'à l'état initial pour améliorer le processus d'apprentissage. C'est précisément ce que le Q-learning propose comme solution à ce défi.

Pour mettre en œuvre cette idée, nous devons mettre à jour les valeurs de Q pour chaque paire état-action (représentée dans chaque cellule du tableau Q). Cette mise à jour se fait en tenant compte de deux éléments essentiels : la récompense obtenue pour atteindre l'état suivant et la valeur la plus élevée attendue dans cet état suivant.
Plus précisément, lorsqu'un agent atteint l'état V, il reçoit une récompense qui indique que sa performance a été couronnée de succès. Cette récompense est utilisée pour mettre à jour la valeur Q de l'état initial D vers l'état de victoire V. Cette mise à jour est effectuée en considérant la valeur la plus élevée possible (max Q) dans l'état V.

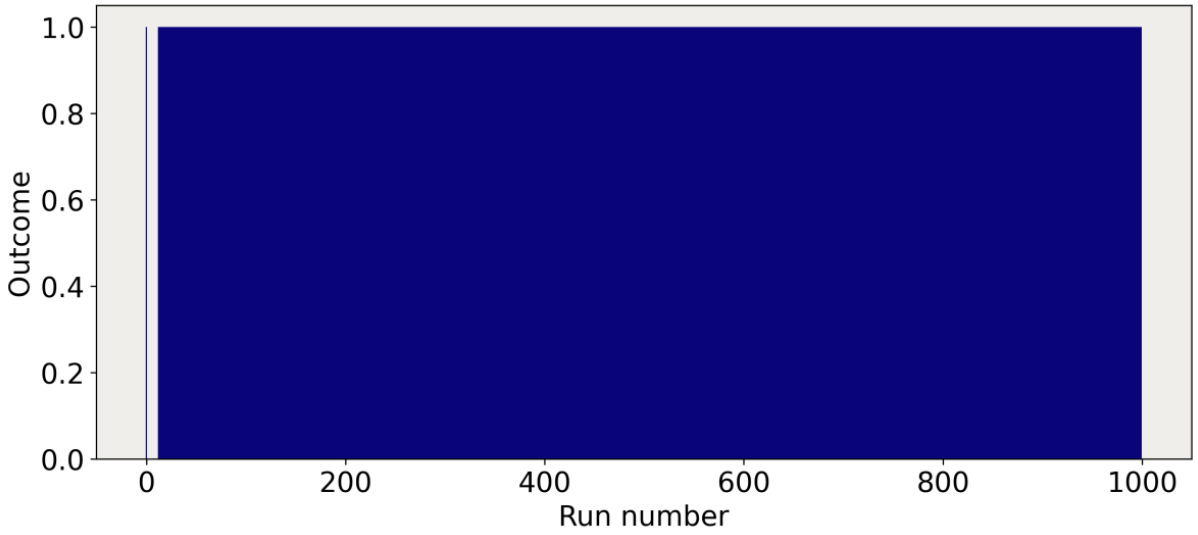
Ainsi, en utilisant le Q-learning, nous mettons à jour les informations stockées dans la table Q pour refléter les meilleures décisions possibles dans chaque état. Cela permet à l'agent d'apprendre progressivement à optimiser ses actions pour maximiser les récompenses à long terme en se basant sur les récompenses instantanées obtenues et les estimations de la valeur future. C'est un mécanisme clé de l'apprentissage par renforcement qui permet à l'agent de s'améliorer au fil du temps en utilisant les informations acquises au cours de ses expériences passées.

5. Apprentissage (Q-learning) et résultats

a) Modèle déterministe


```
Q-table before training:
[[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]
```

```
==========================================
Q-table after training:
[[ -6.86868198  -6.91096058  -7.03802275  -6.86868198]
 [ -6.63299138  -6.65123736  -6.63791571  -6.94240249]
 [ -6.63299138  -6.55839508  -6.53996024  -6.56323611]
 [ -6.37956062  -6.30199611  -6.22981008  -6.40623988]
 [ -6.10705443  -5.98570035  -6.05516694  -6.24180658]
 [ -5.81403703  -5.62037945  -5.77653421  -5.98144074]
 [ -5.49896454  -5.28904659  -5.31227615  -5.46746877]
 [ -5.16017693  -4.89507575  -4.99180369  -5.21373215]
 [ -4.4041819   -4.43702885  -4.3977082   -4.90884455]
 [ -3.98299129  -3.92391385  -3.95127455  -4.45717223]
 [ -3.53009817  -3.3593109   -3.38823961  -3.68750564]
 [ -3.04311631  -3.04311631  -2.70386398  -2.87420134]
 [ -7.05601244  -6.98736105  -7.14295867  -7.08787424]
 [ -6.82974929  -6.7682685   -7.01418063  -6.83379527]
 [ -6.6440297   -6.45806606  -6.65004259  -6.86630376]
 [ -6.36552464  -6.17796013  -6.17136294  -6.31952109]
 [ -5.91141111  -5.85119675  -5.86689896  -5.9788341 ]
 [ -5.729185    -5.49945251  -5.4426135   -5.80728695]
 [ -5.02955506  -5.1134535   -5.09612776  -5.11918301]
 [ -4.77677709  -4.62524791  -4.61160991  -4.86164854]
 [ -4.64395629  -4.05736015  -4.04719759  -4.79513491]
 [ -3.76224954  -3.42339157  -3.42820554  -3.97267498]
 [ -3.19006178  -2.70701844  -2.70810583  -2.89745571]
 [ -1.95643     -1.95643     -1.89999701  -1.95643   ]
 [ -7.18476714  -7.17570464  -7.48520462  -7.29172305]
 [ -7.01065301  -6.86189404 -70.85113     -7.16761001]
 [ -6.62848256  -6.5132156  -70.9896481   -6.65741013]
 [ -6.15472465  -6.12579511 -71.41004974  -6.56979303]
 [ -5.84115906  -5.6953279  -71.4852428   -5.87069743]
 [ -5.37597024  -5.217031   -71.58727234  -5.67840093]
 [ -4.77669477  -4.68559    -71.89686714  -5.11802482]
 [ -4.47732123  -4.0951     -71.91716327  -4.40687522]
 [ -3.51544648  -3.439      -72.18800968  -4.12490415]
 [ -3.06919264  -2.71       -71.2325509   -3.22677865]
 [ -1.95643     -1.9        -70.61299     -2.94078891]
 [ -1.4833      -1.351       -1.          -1.884169  ]
 [ -7.45813417 -70.          -7.48130243  -7.48130243]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]
 [  0.           0.           0.           0.        ]]
```



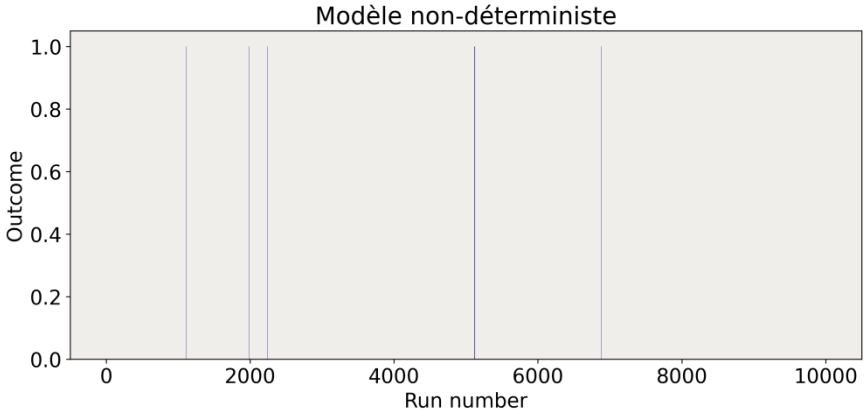
Nous pouvons observer que l'agent a accompli un apprentissage fructueux très rapidement. Chaque barre bleue dans la représentation graphique symbolise une réussite, illustrant ainsi que l'agent a rencontré des difficultés initiales pour atteindre la victoire au tout début du processus d'entraînement. Toutefois, après avoir expérimenté la réussite de manière répétée, à partir de certaines itérations spécifiques (notamment à partir de … répétitions), l'agent a démontré une amélioration significative, manifestant une capacité à remporter régulièrement et à chaque fois des victoires.

L'analyse de la table-Q entraînée s'avère tout aussi fascinante. Ces valeurs reflètent la séquence unique d'actions que l'agent a assimilée pour atteindre son objectif. Cette représentation offre un aperçu détaillé des connaissances acquises par l'agent au cours de son processus d'apprentissage, décrivant les choix d'actions spécifiques qui ont conduit aux succès observés dans la tâche.

b) Modèle non déterministe

Dans le cas du modèle non déterministe, nous pouvons voir qu'il est quasi impossible de former correctement l'agent puisqu'on met en place une probabilité de réaliser l'action voulu (malgré l'augmentation à 10 000 répétitions du jeu). En effet, même si l'agent réussi à atteindre l'objectif, la répétition de la même politique ne permettra pas forcément de réussir à nouveau puisqu'il y'aura toujours une probabilité de 3/4 de glisser (d'où la répétition non continue des victoires suivie des pertes à chaque fois). On peut également voir que les valeurs de notre table-Q sont plus faible et qu'il y a plus de valeurs nulles.

```
Q-table after training:
[[  -7.93810871   -7.70901471   -7.71157742   -7.93811076]
 [  -7.7090156    -7.45446288   -7.45729738   -7.93811101]
 [  -7.45445823   -7.17162794   -7.17476446   -7.7090105 ]
 [  -7.17162816   -6.85736691   -6.86093387   -7.45445971]
 [  -6.85735509   -6.50819845   -6.51200077   -7.17162055]
 [  -6.50816111   -6.12022398   -6.12485703   -6.85722118]
 [  -6.11999206   -5.69020488   -5.69437035   -6.50806101]
 [  -5.6813638    -5.21367358   -5.21579003   -6.11773194]
 [  -5.21315864   -4.68458901   -4.68524089   -5.68963047]
 [  -4.68477523   -4.09479819   -4.09493069   -5.21297343]
 [  -4.0860976    -3.43896878   -3.43873522   -4.6850542 ]
 [  -3.43894332   -3.43897503   -2.70998139   -4.0943887 ]
 [  -7.93811278   -7.45731544   -7.45737258   -7.71157046]
 [  -7.70901432   -7.17480359   -7.17485076   -7.71154997]
 [  -7.45445855   -6.86093978   -6.86095578   -7.45731222]
 [  -7.17162447   -6.512192     -6.51228731   -7.17478036]
 [  -6.85736862   -6.12480426   -6.12468507   -6.86087122]
 [  -6.50817804   -5.69437462   -5.69483915   -6.51189723]
 [  -6.11486685   -5.21662498   -5.2161577    -6.1241969 ]
 [  -5.68612557   -4.68523038   -4.68528254   -5.69276518]
 [  -5.21055214   -4.09488495   -4.09483005   -5.21345964]
 [  -4.68342171   -3.43882652   -3.4388275    -4.68491533]
 [  -4.09442725   -2.7099734    -2.70985904   -4.09042818]
 [  -3.43896815   -2.7099817    -1.89998225   -3.4384958 ]
 [  -7.71158291   -7.17487063   -7.71162815   -7.45736514]
 [  -7.45729622   -6.8610013  -106.711576     -7.45736774]
 [  -7.17472513   -6.51224109 -106.7115705    -7.17486777]
 [  -6.86086911   -6.1247999  -106.71155833   -6.86092133]
 [  -6.51182438   -5.69460554 -106.71153034   -6.51223472]
 [  -6.12377806   -5.2165998  -106.71132913   -6.12483054]
 [  -5.68558894   -4.68520886 -106.71146727   -5.6936711 ]
 [  -5.21321939   -4.09486171 -106.71087355   -5.21475758]
 [  -4.68486778   -3.43879418 -106.70856014   -4.68503589]
 [  -4.09481989   -2.70980862 -106.70974137   -4.09198178]
 [  -3.43805328   -1.89987487 -106.68589932   -3.43732972]
 [  -2.70985422   -1.89901634   -0.99999237   -2.70947492]
 [  -7.45736661 -106.71162396   -7.7116292    -7.71162541]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]
 [   0.            0.            0.            0.        ]]
```



6. Conclusion et ouverture

Le Q-learning est une méthode d'apprentissage par renforcement qui est efficace dans les cas simples, mais qui montre ses limites dans les cas plus complexes.

Dans le cas simple, l'environnement est déterministe, ce qui signifie que l'agent sait avec certitude ce qui se passera si il prend une action donnée. Dans ce cas, le Q-learning peut apprendre rapidement et efficacement la politique optimale.
Dans le cas complexe, l'environnement est non déterministe, ce qui signifie que l'agent ne sait pas toujours ce qui se passera si il prend une action donnée. Dans ce cas, le Q-learning peut avoir du mal à apprendre la politique optimale.

Par exemple, dans le cas de la marche, si l'action "aller à droite" a une probabilité de 2/3 de mener à la falaise, le Q-learning aura du mal à apprendre que cette action est mauvaise.

Les limites du Q-learning dans les cas complexes peuvent être expliquées par les facteurs suivants :

- La complexité de l'espace d'états et d'actions : Plus l'espace d'états et d'actions est complexe, plus il est difficile pour le Q-learning d'apprendre la politique optimale.
- La non-déterminisme de l'environnement : Le Q-learning est basé sur la notion de valeur de qualité, qui est une estimation de la probabilité qu'une action mène à un résultat donné. Dans un environnement non déterministe, la valeur de qualité d'une action n'est pas toujours fiable.
- Le temps d'apprentissage : Le Q-learning peut prendre beaucoup de temps pour converger vers la politique optimale, en particulier dans les cas complexes.

Le machine learning supervisé est une autre méthode d'apprentissage automatique qui peut être utilisée pour résoudre des problèmes complexes. Dans le machine learning supervisé, l'agent est formé sur un ensemble de données de données d'entraînement. Ces données d'entraînement contiennent des exemples d'états et d'actions, ainsi que les résultats souhaités.

Dans le cas de la marche, l'agent pourrait être formé sur un ensemble de données de données d'entraînement qui contiennent des exemples de positions du marcheur, ainsi que les actions qui ont mené à la victoire.

Le machine learning supervisé est généralement plus efficace que le Q-learning dans les cas complexes. Cependant, il nécessite un ensemble de données d'entraînement de bonne qualité, ce qui peut être difficile à obtenir.