

Apprentissage par renforcement appliqué

Considérations pratiques [revision 1.0]

Brahim Chaib-draa

Brahim.Chaib-Draa@ift.ulaval.ca



Université Laval

2020-07-31

1 Plannification d'un projet de RL

2 L'art du débogage en RL

3 La naissance d'un agent RL en pratique

4 Pour aller plus loin

Plannification d'un projet de RL

1 **Plannification d'un projet de RL**

2 **L'art du débogage en RL**

3 **La naissance d'un agent RL en pratique**

4 **Pour aller plus loin**

« Quoi, comment et par où commencer ? »

- 1. Est-ce un problème solvable par apprentissage par renforcement ?**
- 2. Définir le problème**
- 3. Analyser les contraintes de développements**
- 4. Choisir le type d'algorithme approprié**

1. Est-ce un problème solvable par apprentissage par renforcement ?

- Est-ce un problème de décision séquentiel ?

- Est-ce que l'environnement renvoie un signal utilisable comme récompense ?

2. Définir le problème

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

- ✓ Est-ce un problème de décision séquentiel ?
- Est-ce que l'environnement renvoie un signal utilisable comme récompense ?

2. Définir le problème

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

- ✓ Est-ce un problème de décision séquentiel ?
- ✓ Est-ce que l'environnement renvoie un signal utilisable comme récompense ?

2. Définir le problème

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?
- 2. Définir le problème**
 - a. Analyser l'environnement d'apprentissage
 - b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?
3. Analyser les contraintes de développements
4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continu ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?

Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?

- l'échantillonnage est ...
 - rapides ou lent à produire ?
 - risqué à produire ou non ?
 - couteux à produire ou non ?
- est-ce qu'on a accès à un modèle de l'environnement ?
 - Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continue ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?

Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?

- l'échantillonnage est ...
 - rapides ou lent à produire ?
 - risqué à produire ou non ?
 - coûteux à produire ou non ?
- est-ce qu'on a accès à un modèle de l'environnement ?
 - Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continue ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?

Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?

- l'échantillonnage est ...
 - rapides ou lent à produire ?
 - risqué à produire ou non ?
 - coûteux à produire ou non ?
- est-ce qu'on a accès à un modèle de l'environnement ?
 - Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continu ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?

Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?

- l'échantillonnage est ...
 - rapides ou lent à produire ?
 - risqué à produire ou non ?
 - coûteux à produire ou non ?
- est-ce qu'on a accès à un modèle de l'environnement ?
 - Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continu ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?

Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?

- l'échantillonnage est ...
 - rapides ou lent à produire ?
 - risqué à produire ou non ?
 - couteux à produire ou non ?
- est-ce qu'on a accès à un modèle de l'environnement ?
 - Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continue ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?
 - Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?
- l'échantillonnage est ...
 - rapides ou lent à produire ?
 - risqué à produire ou non ?
 - coûteux à produire ou non ?
- est-ce qu'on a accès à un modèle de l'environnement ?
 - Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continu ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?

Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?

- l'échantillonnage est ...
 - rapides ou lent à produire ?
 - risqué à produire ou non ?
 - couteux à produire ou non ?
- est-ce qu'on a accès à un modèle de l'environnement ?
 - Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continue ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?

Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?

– l'échantillonnage est ...

rapides ou lent à produire ?

risqué à produire ou non ?

coûteux à produire ou non ?

- est-ce qu'on a accès à un modèle de l'environnement ?

Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continue ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?

Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?

- l'échantillonnage est ...
 - rapides ou lent à produire ?
 - risqué à produire ou non ?
 - coûteux à produire ou non ?
- est-ce qu'on a accès à un modèle de l'environnement ?
 - Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

- l'environnement est réel ou en simulateur ?
- l'espace d'action/observation est continue ou discret ?
- l'espace d'action/observation est en haute dimension ?
- l'environnement est complètement observable ou partiellement observable (pas au cours) ?
- l'environnement est épisodique ou continu ?

Si oui, est-ce que les trajectoires sont **garanties** de toujours terminer ?

- l'échantillonnage est ...
 - rapides ou lent à produire ?
 - risqué à produire ou non ?
 - coûteux à produire ou non ?
- est-ce qu'on a accès à un modèle de l'environnement ?
 - Si oui, est-ce que ce modèle est fiable ?

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

– Définition de la tâche d'apprentissage

ex. : Pac-Man autonome, prédire le bon moment pour vendre un lot d'action, ...

– Déterminer de quelle manière l'agent devra exécuter la tâche apprise

optimale garantie \Leftarrow méthodes par programmation dynamique

quasi optimal ou mieux \Leftarrow méthodes RL

robuste en situation d'adversité \Leftarrow méthodes RL par entropie maximale (pas au cours)

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

- Définition de la tâche d'apprentissage

 - ex. : Pac-Man autonome, prédire le bon moment pour vendre un lot d'action, ...

- Déterminer de quelle manière l'agent devra exécuter la tâche apprise

 - optimale garantie \Leftarrow méthodes par programmation dynamique

 - quasi optimal ou mieux \Leftarrow méthodes RL

 - robuste en situation d'adversité \Leftarrow méthodes RL par entropie maximale (pas au cours)

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

a. Analyser l'environnement d'apprentissage

b. Quels objectifs concrets cherche-t-on à atteindre avec notre agent ?

– Définition de la tâche d'apprentissage

ex. : Pac-Man autonome, prédire le bon moment pour vendre un lot d'action, ...

– Déterminer de quelle manière l'agent devra exécuter la tâche apprise

optimale garantie \Leftarrow méthodes par programmation dynamique

quasi optimal ou mieux \Leftarrow méthodes RL

robuste en situation d'adversité \Leftarrow méthodes RL par entropie maximale (pas au cours)

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?
2. Définir le problème
- 3. Analyser les contraintes de développements**
 - a. Quel sont les ressources computationnelles disponibles ?
 - b. Quel sont les ressources de stockage disponibles ?
 - c. Combien de temps on dispose pour produire l'agent ?
4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?
2. Définir le problème
3. Analyser les contraintes de développements

- a. Quel sont les ressources computationnelles disponibles ?

Remarque : En DRL, **l'échantillonnage est souvent un *bottleneck* plus important que l'étape d'optimisation** du réseau de neurones. Cependant, certains algorithmes peuvent être implémentés en suivant une architecture parallèle de façon à utiliser plusieurs **Worker** exécutant l'échantillonnage (1 par coeur disponible) et 1 **Learner** (sur son propre coeur) responsable d'optimiser le réseau neurone ex : Asynchronous Advantage Actor-Critic (A3C).

Pour cette raison, **le nombre de coeurs d'un processeur a généralement plus de valeur** que l'accès a un GPU.

Voir *Asynchronous Methods for Deep Reinforcement Learning*, 2016 par Mnih et al. [1]

- b. Quel sont les ressources de stockage disponibles ?
- c. Combien de temps on dispose pour produire l'agent ?

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

3. Analyser les contraintes de développements

a. Quel sont les ressources computationnelles disponibles ?

b. Quel sont les ressources de stockage disponibles ?

Remarque : Si vous avez suffisamment de mémoire vive disponible, considéré utiliser un *framework* exploitant le paradigme de shared memory (même si vous ne prévoyez pas implémenter votre algorithme en parallèle).

Ce type de *framework* permet le stockage des échantillons de trajectoires et le/les réseaux de neurones in-memory ce qui accélère considérablement le passage d'informations entre les **workers** et le **learner**.

In-memory framework : Apache Arrow, Redis

Solution clé en main pour le RL : RAY RLlib

c. Combien de temps on dispose pour produire l'agent ?

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

3. Analyser les contraintes de développements

a. Quel sont les ressources computationnelles disponibles ?

b. Quel sont les ressources de stockage disponibles ?

c. **Combien de temps on dispose pour produire l'agent ?**

Remarque : Considérer lors de votre planification que le temps a allouée individuellement à chaque étape du développement d'un algorithme de RL (design, implémentation, débogage, entraînement de l'agent, évaluation des performances de l'agent) peuvent varier fortement d'un projet à l'autre en fonction de la complexité de l'algorithme, de l'environnement, des ressources disponibles et **de votre expérience personnelle à implémenter spécifiquement des algorithmes de RL.**

Lecture recommandée :

Lessons Learned Reproducing a Deep Reinforcement Learning Paper par Amid Fish

4. Choisir le type d'algorithme approprié

1. Est-ce un problème solvable par apprentissage par renforcement ?
2. Définir le problème
3. Analyser les contraintes de développements
- 4. Choisir le type d'algorithme approprié**
 - a. Programmation dynamique, apprentissage sans modèle ou apprentissage basé sur un modèle ?
 - b. Méthodes tabulaires ou approximatives
 - c. Basée sur les valeurs ou par recherche de politique ?
 - d. EN-ligne ou HORS-ligne ?
 - e. *ON-policy* ou *OFF-policy* ?

1. Est-ce un problème solvable par apprentissage par renforcement ?
2. Définir le problème
3. Analyser les contraintes de développements
4. Choisir le type d'algorithme approprié
 - a. **Programmation dynamique, apprentissage sans modèle ou apprentissage basé sur un modèle ?**

Remarque : Les méthodes par programmation dynamique **requièrent un modèle parfait** de l'environnement.

Les méthodes sans modèle sont **agnostiques au modèle** de l'environnement.

Les méthodes basées sur un modèle sont généralement plus **efficaces sur le plan échantillonnage**. Le modèle peut être appris ou fourni en fonction de la méthode.

- b. Méthodes tabulaires ou approximatives
- c. Basée sur les valeurs ou par recherche de politique ?
- d. EN-ligne ou HORS-ligne ?
- e. *ON-policy* ou *OFF-policy* ?

1. Est-ce un problème solvable par apprentissage par renforcement ?
2. Définir le problème
3. Analyser les contraintes de développements
4. Choisir le type d'algorithme approprié

a. Programmation dynamique, apprentissage sans modèle ou apprentissage basé sur un modèle ?

b. Méthodes tabulaires ou approximatives

Remarque : Les méthodes tabulaires fonctionnent bien sur les **espaces en basse dimension** (ex. l'espace d'action discret : gauche, droit, monter, descendre) et sont plus simple à implémenter que les méthodes approximatives.

Les méthodes approximatives sont les méthodes appropriées pour les **espaces en haute dimension** comme les espaces d'action continue

- c. Basée sur les valeurs ou par recherche de politique ?
- d. EN-ligne ou HORS-ligne ?
- e. *ON-policy* ou *OFF-policy* ?

1. Est-ce un problème solvable par apprentissage par renforcement ?
2. Définir le problème
3. Analyser les contraintes de développements
4. Choisir le type d'algorithme approprié
 - a. Programmation dynamique, apprentissage sans modèle ou apprentissage basé sur un modèle ?
 - b. Méthodes tabulaires ou approximatives
 - c. **Basée sur les valeurs ou par recherche de politique ?**

Remarque : C'est un compromis entre l'**efficacité de l'échantillonnage** des méthodes basées sur les valeurs et la **stabilité à l'entraînement** des méthodes par recherche de politique (stabilité → meilleure convergence).
 - d. EN-ligne ou HORS-ligne ?
 - e. *ON-policy* ou *OFF-policy* ?

1. Est-ce un problème solvable par apprentissage par renforcement ?

2. Définir le problème

3. Analyser les contraintes de développements

4. Choisir le type d'algorithme approprié

a. Programmation dynamique, apprentissage sans modèle ou apprentissage basé sur un modèle ?

b. Méthodes tabulaires ou approximatives

c. Basée sur les valeurs ou par recherche de politique ?

d. **EN-ligne ou HORS-ligne ?**

Remarque : Les algorithmes HORS-ligne peuvent être **utilisés seulement sur des environnements épisodiques** et qui sont **garantie de terminer à toutes les trajectoires**.

Les algorithmes EN-ligne peuvent être un **bon choix lorsque la capacité à optimiser l'agent est plus rapide** que la capacité à produire des échantillons.

e. *ON-policy* ou *OFF-policy* ?

1. Est-ce un problème solvable par apprentissage par renforcement ?
2. Définir le problème
3. Analyser les contraintes de développements
4. Choisir le type d'algorithme approprié

- a. Programmation dynamique, apprentissage sans modèle ou apprentissage basé sur un modèle ?
- b. Méthodes tabulaires ou approximatives
- c. Basée sur les valeurs ou par recherche de politique ?
- d. EN-ligne ou HORS-ligne ?
- e. **ON-policy** ou **OFF-policy** ?

Remarque : Les algorithmes *OFF-policy* sont plus **efficacés sur le plan échantillonnage**. Ils permettent l'utilisation d'échantillons collectés en suivant une politique différente, collectés antérieurement ou même d'ensemble de données de trajectoires.

L'art du débogage en RL

1 Plannification d'un projet de RL

2 L'art du débogage en RL

- Les difficultés liées au développement en RL
- Recommandation

3 La naissance d'un agent RL en pratique

4 Pour aller plus loin

L'art du débogage en RL

Les difficultés liées au développement en RL

*« What is unique about machine learning is that **it is exponentially harder to figure out what is wrong** when things don't work as expected »*¹ – S. Zayd Enam

Origines des difficultés affectant le développement d'un projet en RL :

- 1 Problèmes d'ingénierie logiciel classique liés ...
 - au design de l'algorithme ;
 - à l'implémentation ;
- 2 Problèmes propre au domaine de l'apprentissage machine en lien avec ...
 - le model ;
 - les données ;
- 3 Problématiques additionnel propre au RL en lien avec ...
 - la stochasticité du système ;
 - l'aspect temporel ;
 - l'**absence de feedback immédiat** lorsque l'algorithme ne fonctionne pas comme il devrait ... ou pire l'**absence total de feedback** ;

1. Extrait de Why is machine learning 'hard'? par S. Zayd Enam [2]. Une réflexion sur les difficultés lié au développement de projet d'apprentissage machine.

Note : Pour appliquer cette réflexion au contexte du RL, simplement ajouter les dimensions stochasticité, temporalité et les problématiques lié au signal de récompense.

« ... *broken RL code almost always fails silently*, ... »

– Josh Achiam, OpenAI Spinning Up [3]

⚠ **Problème :** **Absence de *feedback* immédiat** lorsque l'algorithme ne fonctionne pas comme il devrait
... ou pire **absence total de *feedback*** ;

Contexte : Le code compile, l'agent semble réagir et donne l'impression qu'il fonctionne mais en réalité il n'apprend pas assez pour atteindre l'objectif ou il n'apprend pas du tout.

– Est-ce que c'est un problème d'hyperparamètre ?

– Je pourrais faire des petit ajustement au hasard et me croiser les doigts !

– Peut-être que l'algorithme à besoin de plus de temps avant de pouvoir exhiber signe de vie ?

– Peut-être que c'est un problème d'implémentation ?

– Je devrais réviser mon code ... tout mon code, caractères par caractère !

– Je devrais réviser mon code ... tout mon code, caractères par caractère !

« ... *broken RL code almost always fails silently*, ... »

– Josh Achiam, OpenAI Spinning Up [3]

▲ **Problème :** **Absence de *feedback* immédiat** lorsque l'algorithme ne fonctionne pas comme il devrait ... ou pire **absence total de *feedback*** ;

Contexte : Le code compile, l'agent semble réagir et donne l'impression qu'il fonctionne mais en réalité il n'apprend pas assez pour atteindre l'objectif ou il n'apprend pas du tout.

- Est-ce que c'est un problème d'hyperparamètre ?
- Je pourrais faire des petit ajustement au hasard et me croiser les doigts !
- Peut-être que l'algorithme à besoin de plus de temps avant de pouvoir exhiber signe de vie ?
- Est-ce que c'est un problème d'implémentation ?
- Je devrais relire mon code ... tout mon code, caractère par caractère !
- Peut-être que je devrais sérieusement remettre en question ma carrière en A.I. ?

« ... *broken RL code almost always fails silently*, ... »

– Josh Achiam, OpenAI Spinning Up [3]

⚠ **Problème :** **Absence de *feedback* immédiat** lorsque l'algorithme ne fonctionne pas comme il devrait ... ou pire **absence total de *feedback*** ;

Contexte : Le code compile, l'agent semble réagir et donne l'impression qu'il fonctionne mais en réalité il n'apprend pas assez pour atteindre l'objectif ou il n'apprend pas du tout.

- Est-ce que c'est un problème d'hyperparamètre ?
- Je pourrais faire des petit ajustement au hasard et me croiser les doigts !
- Peut-être que l'algorithme à besoin de plus de temps avant de pouvoir exhiber signe de vie ?
- Est-ce que c'est un problème d'implémentation ?
- Je devrais relire mon code ... tout mon code, caractère par caractère !
- Peut-être que je devrais sérieusement remettre en question ma carrière en A.I. ?
- ...

« ... *broken RL code almost always fails silently*, ... »

– Josh Achiam, OpenAI Spinning Up [3]

⚠ **Problème :** **Absence de *feedback* immédiat** lorsque l'algorithme ne fonctionne pas comme il devrait ... ou pire **absence total de *feedback*** ;

Contexte : Le code compile, l'agent semble réagir et donne l'impression qu'il fonctionne mais en réalité il n'apprend pas assez pour atteindre l'objectif ou il n'apprend pas du tout.

- Est-ce que c'est un problème d'hyperparamètre ?
- Je pourrais faire des petit ajustement au hasard et me croiser les doigts !
- Peut-être que l'algorithme à besoin de plus de temps avant de pouvoir exhiber signe de vie ?
- Est-ce que c'est un problème d'implémentation ?
- Je devrais relire mon code ... tout mon code, caractère par caractère !
- Peut-être que je devrais sérieusement remettre en question ma carrière en A.I. ?
- ...

Qu'elle est le problème ? Qu'est-ce qu'on cherche ? Qu'est-ce qu'on change ?

L'art du débogage en RL

Recommandation

« ... *broken RL code almost always fails silently*, ... »

– Josh Achiam, OpenAI Spinning Up [3]

⚠ Problème : Absence de *feedback* immédiat lorsque l'algorithme ne fonctionne pas comme il devrait
... ou pire **absence total de *feedback*** ;

Recommandation :

1. Implémenter des outils afin de faire parler votre code
2. Procéder méthodiquement et de façon délibéré

Recommandation :

1. Implémenter des outils afin de faire parler votre code :

- ▶ Collecter des métriques qui suivent l'évolution de l'entraînement
- ▶ Implémenter les tests unitaires appropriés
- ▶ Utiliser des assertions dans votre code
- ▶ Observer l'agent agir dans l'environnement périodiquement

2. Procéder méthodiquement et de façon délibéré

Recommandation :

1. Implémenter des outils afin de faire parler votre code :

▶ **Collecter des métriques qui suivent l'évolution de l'entraînement :**

- Est-ce que l'agent apprend quelque chose? $\leftarrow G(\tau)$
- Est-ce que l'agent survit de plus en plus longtemps? $\leftarrow \text{length}(\tau)$
- Est-ce que la fonction de perte change? $\leftarrow \text{loss}(\pi)$
- ...

▶ Implémenter les tests unitaires appropriés

▶ Utiliser des assertions dans votre code

▶ Observer l'agent agir dans l'environnement périodiquement

2. Procéder méthodiquement et de façon délibéré

Recommandation :

1. Implémenter des outils afin de faire parler votre code :

- ▶ Collecter des métriques qui suivent l'évolution de l'entraînement
- ▶ **Implémenter les tests unitaires appropriés**

Remarque : C'est considérablement **plus facile de déboguer quand on a confiance en notre implémentation** des composantes de base et ça permet de **circonscrire nos recherches**.

Prenez soin de tester le comportement attendu, les cas limites ainsi que l'interaction entre les composantes.

Recommandation : Adoptez la méthodologie Test-Driven Development.

- ▶ Utiliser des assertions dans votre code
- ▶ Observer l'agent agir dans l'environnement périodiquement

2. Procéder méthodiquement et de façon délibéré

Recommandation :

1. Implémenter des outils afin de faire parler votre code :

- ▶ Collecter des métriques qui suivent l'évolution de l'entraînement
- ▶ Implémenter les tests unitaires appropriés
- ▶ **Utiliser des assertions dans votre code :**
 - **Valider les entrées/sorties** attendu des composantes ;
 - Écrire des assertions **informatives** avec des messages d'erreur explicite ;
 - **▲** Pour les **composantes exécutées massivement**, ajouter une fonctionnalité pour **désactiver ces assertions à l'entraînement** afin de ne pas ralentir l'exécution.
C'est particulièrement important en Python puisque c'est un langage interprété ;
- ▶ Observer l'agent agir dans l'environnement périodiquement

2. Procéder méthodiquement et de façon délibéré

Recommandation :

1. Implémenter des outils afin de faire parler votre code :

- ▶ Collecter des métriques qui suivent l'évolution de l'entraînement
- ▶ Implémenter les tests unitaires appropriés
- ▶ Utiliser des assertions dans votre code
- ▶ **Observer l'agent agir dans l'environnement périodiquement**

Remarque : Son comportement peut fournir de précieuses indications sur l'état de l'entraînement.

⚠ Le rendu d'épisode force l'algorithme à tourner avec $t = \text{wall clock time}$

2. Procéder méthodiquement et de façon délibéré

Recommandation :

1. Implémenter des outils afin de faire parler votre code
2. **Procéder méthodiquement et de façon délibéré**
 - ▶ Vérifier que les bons éléments sont insérés dans les bonnes composantes
 - ▶ Assurez vous de bien comprendre le fonctionnement de l'algorithme, les subtilités sont parfois critique au bon fonctionnement
 - ▶ Valider votre implémentation sur un environnement facile à résoudre et que vous connaissez bien, ensuite passée à des environnements difficiles

Recommandation :

1. Implémenter des outils afin de faire parler votre code
2. Procéder méthodiquement et de façon délibéré
 - ▶ Vérifier que les bons éléments sont insérés dans les bonnes composantes
 - ▶ Assurez vous de bien comprendre le fonctionnement de l'algorithme, les subtilités sont parfois critique au bon fonctionnement
 - ▶ Valider votre implémentation sur un environnement facile à résoudre et que vous connaissez bien, ensuite passée à des environnements difficiles

Recommandation :

1. Implémenter des outils afin de faire parler votre code
2. Procéder méthodiquement et de façon délibéré

- ▶ Vérifier que les bons éléments sont insérés dans les bonnes composantes
- ▶ Assurés vous de bien comprendre le fonctionnement de l'algorithme, les subtilités sont parfois critique au bon fonctionnement

Exemple : « Like collecting the *right observation* s_t that triggered the action a_t , ... here t is a critical detail. I say the *right observation* because collecting accidentally the *observation* s_{t+1} will cause the algorithm to map a policy

$$(\text{action} \times \text{observe reaction}) \longrightarrow [0, 1]$$

instead of

$$(\text{observation} \times \text{action}) \longrightarrow [0, 1]$$

and the agent won't learn.

This mistake is very easy to make when implementing if we are not careful with detail. »

- ▶ Valider votre implémentation sur un environnement facile à résoudre et que vous connaissez bien, ensuite passée à des environnements difficiles

Recommandation :

1. Implémenter des outils afin de faire parler votre code
2. Procéder méthodiquement et de façon délibéré
 - ▶ Vérifier que les bons éléments sont insérés dans les bonnes composantes
 - ▶ Assurés vous de bien comprendre le fonctionnement de l'algorithme, les subtilités sont parfois critique au bon fonctionnement
 - ▶ Valider votre implémentation sur un environnement facile à résoudre et que vous connaissez bien, ensuite passée à des environnements difficiles

La naissance d'un agent RL en pratique

1 Plannification d'un projet de RL

2 L'art du débogage en RL

3 La naissance d'un agent RL en pratique

- Recommandation pour l'algorithme *Deep Q-network*
- Recommandation pour l'algorithme *Policy gradient*

4 Pour aller plus loin

La naissance d'un agent RL en pratique

Recommandation pour l'algorithme *Deep Q-network*

La naissance d'un agent RL en pratique

Recommandation pour l'algorithme *Policy gradient*

Pour aller plus loin

1 Plannification d'un projet de RL

2 L'art du débogage en RL

3 La naissance d'un agent RL en pratique

4 Pour aller plus loin

- Complément théorique
- Ressource théorique
- Framework et outil d'implémentation
- Références

Pour aller plus loin

Complément théorique

Pour aller plus loin

Ressource théorique

Pour aller plus loin

Framework et outil d'implémentation

Pour aller plus loin

Références

Références I

1. **MNIH, V. et al.** Asynchronous Methods for Deep Reinforcement Learning. **48**. arXiv : 1602.01783. <http://arxiv.org/abs/1602.01783> (2016).
2. **ENAM, S. Z.** *Why is machine learning 'hard'?*. 2016.
<http://ai.stanford.edu/~zayd/why-is-machine-learning-hard.html>.
3. **ACHIAM, J.** Spinning Up in Deep Reinforcement Learning.
<https://spinningup.openai.com/en/latest/index.html> (2018).