Apprentissage par renforcement appliqué

Considérations pratiques [revision 2.0]

Brahim Chaib-draa

Brahim.Chaib-Draa@ift.ulaval.ca

Université Laval

2020-08-04

GLO-7050 Apprentissage machine en pratique

Module sur le RL appliqué

Synthèse et diapositives par

Luc Coupal

Planification d'un projet de RL

Planification d'un projet de RL

Planification et Agilité en développement logicielle

« . . . mais planifier compromet l'Agilité en développement logiciel ! »

« . . . mais planifier compromet l'Agilité en développement logiciel ! »

Mise au point : La planification **excessive sous l'assomption du statue quo** compromet l'Agilité.

Il n'y a aucune quantité d'agilité qui peut nous venir en aide quand on réalise trop tard qu'on s'est dirigé dans un précipice et qu'on ne sait pas voler.

Exemple de précipice : développer une solution RL pour un bras robot industriel basé sur un algorithme de type Monte-Carlo ON-policy sans tenir compte du fait que ce bras robot est fragile et coute très cher à réparer.

Il y a moyen d'exécuter une planification de projet et de rester Agile en procédant comme suit :

1. Répondre aux questions **quoi** et **pourquoi**
2. Répondre à la question **comment** en limitant l'analyse aux éléments qui peuvent avoir une incidence critique sur la progression du développement et sur la capacité à atteindre les objectifs finaux du projet

« . . . mais planifier compromet l'Agilité en développement logiciel ! »

Mise au point : La planification **excessive sous l'assomption du statue quo** compromet l'Agilité.

Il n'y a aucune quantité d'agilité qui peut nous venir en aide quand on réalise trop tard qu'on s'est dirigé dans un précipice et qu'on ne sait pas voler.

Exemple de précipice : développer une solution RL pour un bras robot industriel basé sur un algorithme de type Monte-Carlo ON-policy sans tenir compte du fait que ce bras robot est fragile et coute très cher à réparer.

Il y a moyen d'exécuter une planification de projet et de rester Agile en procédant comme suit :

1. Répondre aux questions **quoi** et **pourquoi**
2. Répondre à la question **comment** en limitant l'analyse aux éléments qui peuvent avoir une incidence critique sur la progression du développement et sur la capacité à atteindre les objectifs finaux du projet

« ... mais planifier compromet l'Agilité en développement logiciel ! »

Mise au point : La planification **excessive sous l'assomption du statue quo** compromet l'Agilité.

Il n'y a aucune quantité d'agilité qui peut nous venir en aide quand on réalise trop tard qu'on s'est dirigé dans un précipice et qu'on ne sait pas voler.

Exemple de précipice : développer une solution RL pour un bras robot industriel basé sur un algorithme de type Monte-Carlo ON-policy sans tenir compte du fait que ce bras robot est fragile et coute très cher à réparer.

Il y a moyen d'exécuter une planification de projet et de rester Agile en procédant comme suit :

1. Répondre aux questions **quoi** et **pourquoi**
2. Répondre à la question **comment** en limitant l'analyse aux éléments qui peuvent avoir une incidence critique sur la progression du développement et sur la capacité à atteindre les objectifs finaux du projet

Planification d'un projet de RL

Poser les bonnes questions

« Quoi ? Pourquoi ? Comment ? 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 observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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 continu ou discret ?
      - l'espace d'action/observation est en haute dimension ?
      - l'environnement est complètement observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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 continu ou discret ?
      - l'espace d'action/observation est en haute dimension ?
      - l'environnement est complètement observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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 continu ou discret ?
      - l'espace d'action/observation est en haute dimension ?
      - l'environnement est complètement observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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 continu ou discret ?
      - l'espace d'action/observation est en haute dimension ?
      - l'environnement est complètement observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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 continu ou discret ?
      - l'espace d'action/observation est en haute dimension ?
      - l'environnement est complètement observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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 continu ou discret ?
      - l'espace d'action/observation est en haute dimension ?
      - l'environnement est complètement observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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 continu ou discret ?
      - l'espace d'action/observation est en haute dimension ?
      - l'environnement est complètement observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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 continu ou discret ?
      - l'espace d'action/observation est en haute dimension ?
      - l'environnement est complètement observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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 continu ou discret ?
      - l'espace d'action/observation est en haute dimension ?
      - l'environnement est complètement observé ou partiellement observé ?
      - l'environnement est épisodique ou continuel ?
        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
   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 ⇐ méthodes par programmation dynamique
        quasi optimal ou mieux ⇐ méthodes RL
        robuste en situation d'adversité ⇐ 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. Quelles sont les ressources computationnelles disponibles ?
   b. Quelles sont les ressources de stockage disponibles ?
   c. Combien de temps on dispose pour produire l'agent ?
   d. Si l'environnement est réel, est-ce qu'il y a des risques physiques ?
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. **Quelles 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. Quelles sont les ressources de stockage disponibles ?
   c. Combien de temps on dispose pour produire l'agent ?
   d. Si l'environnement est réel, est-ce qu'il y a des risques physiques ?
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. Quelles sont les ressources computationnelles disponibles ?
   b. **Quelles 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 : Appache Arrow, Redis

      Solution clé en main pour le RL : RAY RLlib

   c. Combien de temps on dispose pour produire l'agent ?
   d. Si l'environnement est réel, est-ce qu'il y a des risques physiques ?
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. Quelles sont les ressources computationnelles disponibles ?
   b. Quelles sont les ressources de stockage disponibles ?
   c. **Combien de temps on dispose pour produire l'agent ?**

      Remarque : Considéré lors de votre planification que le temps à allouer individuellement à chaque étape du développement d'un algorithme de RL (design, implémentation, débogage, entraînement de l'agent, évaluation des performances de l'agent) peut 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

   d. Si l'environnement est réel, est-ce qu'il y a des risques physiques ?
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. Quelles sont les ressources computationnelles disponibles ?
   b. Quelles sont les ressources de stockage disponibles ?
   c. Combien de temps on dispose pour produire l'agent ?
   d. **Si l'environnement est réel, est-ce qu'il y a des risques physiques ?**

      Exemple : un quadrirotor qui pourait décider de foncé sur tout ce qui bouge par ce que

      $$\widehat{V}_\theta^\pi(\langle \text{speed : 100km, safety distance to humain : 0m} \rangle)$$

      est l'état le plus payant.
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 **efficientes 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'**efficience de l'échantillonnage** des méthodes basées sur les valeurs et la **stabilité à l'entraînement** des méthodes par recherche de politique (stabilité $\longrightarrow$ 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 **efficients 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.

Bonne pratique en développement RL

1 Planification d'un projet de RL

2 Bonne pratique en développement RL

- Les difficultés liées au développement en RL
- L'art du débogage en RL
- L'importance des détails d'implémentation

3 Pour aller plus loin

Bonne pratique en développement 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 logicielle classique liés ...
   - au design de l'algorithme ;
   - à l'implémentation ;
2. Problèmes propres au domaine de l'apprentissage machine en lien avec ...
   - le modèle ;
   - les données ;
3. Problématiques additionnelles propres au RL en lien avec ...
   - la stochasticitée du système ;
   - l'aspect temporel ;
   - l'**absence de *feedback* immédiat** lorsque l'algorithme ne fonctionne pas comme il devrait ... ou pire l'**absence totale de feedback** ;

1. Extrait de Why is machine learning 'hard'? par S. Zayd Enam [2]. Une réflexion sur les difficultés liées au développement de projet d'apprentissage machine.
   Note : Pour appliquer cette réflexion au contexte du RL, ajouter les dimensions stochasticitée et temporalité ainsi que les problématiques liées au signal de récompense.

Bonne pratique en développement RL

L'art du débogage en RL

« . . . ***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 totale 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 de petit ajustement au hasard et me croiser les doigts !

– Peut-être que l'algorithme a 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 !

« . . . ***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 totale 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 de 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 totale 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 de 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 totale 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 de 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. ?
- . . .

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

« . . . ***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 totale de *feedback*** ;

Recommandation :

1. Implémentez des outils afin de faire parler votre code
2. Procédez méthodiquement et de façon délibérée

Recommandation :

1. **Implémentez des outils afin de faire parler votre code** :
   - Collectez des métriques qui suivent l'évolution de l'entraînement
   - Implémentez les tests unitaires appropriés
   - Utilisez des assertions dans votre code
   - Observez périodiquement l'agent agir dans l'environnement
2. Procédez méthodiquement et de façon délibérée

Recommandation :

1. Implémentez des outils afin de faire parler votre code :
   - **Collectez des métriques qui suivent l'évolution de l'entraînement** :
     - Est-ce que l'agent apprend quelque chose ? $\longleftarrow G(\tau)$ (mean/min/max/stdev)
     - Est-ce que l'agent survie de plus en plus longtemps ? $\longleftarrow lenght(\tau)$
     - Est-ce que la politique apprend quelque chose ? $\longleftarrow D_{KL}\left(\pi_{old\theta}(\cdot|\mathbf{s}) \middle\| \pi_{\theta}(\cdot|\mathbf{s})\right)$
     - ...
   - Implémentez les tests unitaires appropriés
   - Utilisez des assertions dans votre code
   - Observez périodiquement l'agent agir dans l'environnement
2. Procédez méthodiquement et de façon délibérée

Recommandation :

1. Implémentez des outils afin de faire parler votre code :
   - Collectez des métriques qui suivent l'évolution de l'entraînement
   - **Implémentez 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 Developpement*.
   - Utilisez des assertions dans votre code
   - Observez périodiquement l'agent agir dans l'environnement
2. Procédez méthodiquement et de façon délibérée

Recommandation :

1. Implémentez des outils afin de faire parler votre code :
   - Collectez des métriques qui suivent l'évolution de l'entraînement
   - Implémentez les tests unitaires appropriés
   - **Utilisez des assertions dans votre code** :
     - **Validez les entrées/sorties** attendues des composantes ;
     - Écrire des assertions **informatives** avec des messages d'erreur explicite ;
     - ⚠ Pour les **composantes exécutées massivement**, ajoutez une fonctionnalité pour **désactiver les assertions à l'entraînement** afin de ne pas ralentir l'exécution. C'est particulièrement important en Python puisque c'est un langage interprété ;
   - Observez périodiquement l'agent agir dans l'environnement
2. Procédez méthodiquement et de façon délibérée

Recommandation :

1. Implémentez des outils afin de faire parler votre code :
   - Collectez des métriques qui suivent l'évolution de l'entraînement
   - Implémentez les tests unitaires appropriés
   - Utilisez des assertions dans votre code
   - **Observez périodiquement l'agent agir dans l'environnement**

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

     ⚠ Le rendu d'épisode force l'algorithme à tourner avec $t = \textit{wall clock time}$
2. Procédez méthodiquement et de façon délibérée

Recommandation :

1. Implémentez des outils afin de faire parler votre code
2. **Procédez méthodiquement et de façon délibérée**
   - Vérifiez que les données sont acheminées au bon endroit (porter attention aux détails)
   - Cherchez une « preuve de vie » sur un problème jouet pour valider votre implémentation
   - Utilisez plusieurs « random seed » et faite une moyenne de vos résultats

Recommandation :

1. Implémentez des outils afin de faire parler votre code
2. Procédez méthodiquement et de façon délibérée
   - **Vérifiez que les données sont acheminées au bon endroit** (porter attention aux détails)
     Exemple :

$$(\mathbf{s}', \mathbf{a}, r, \mathbf{s}') \not\equiv (\mathbf{s}, \mathbf{a}, r, \mathbf{s}')$$

```
# RL feedback loop: NON working example
obs = env.reset()
done = False
while not(done):
    action = myRlAgent.act(obs)
    next_obs, reward, done, info = env.step(action)
    obs = next_obs
    myRlAgent.getFeedback(obs, action, reward, next_obs)
```

```
# RL feedback loop: working example
obs = env.reset()
done = False
while not(done):
    action = myRlAgent.act(obs)
    next_obs, reward, done, info = env.step(action)
    myRlAgent.getFeedback(obs, action, reward, next_obs)
    obs = next_obs
```

L'implémentation de gauche cause la politique π à apprendre un « mapping »

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

au lieu de

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

L'agent va donner l'impression que tout fonctionne bien, mais il n'apprendra pas.

   - Cherchez une « preuve de vie » sur un problème jouet pour valider votre implémentation
   - Utilisez plusieurs « random seed » et faite une moyenne de vos résultats

Recommandation :

1. Implémentez des outils afin de faire parler votre code
2. Procédez méthodiquement et de façon délibérée
   - Vérifiez que les données sont acheminées au bon endroit (porter attention aux détails)
   - **Cherchez une « preuve de vie » sur un problème jouet pour valider votre implémentation**

     c.-à-d. un environnement facile et rapide à résoudre (ex. : OpenAI Gym CartPole)

     Ensuite passée à des environnements difficiles, long à résoudre, en haute dimension, ...

     Remarque : Apprenez à bien connaitre quelques problèmes jouets afin de développer un œil critique vis-à-vis le comportement attendu de l'agent.
   - Utilisez plusieurs « random seed » et faite une moyenne de vos résultats

Recommandation :

1. Implémentez des outils afin de faire parler votre code
2. Procédez méthodiquement et de façon délibérée
   - Vérifiez que les données sont acheminées au bon endroit (porter attention aux détails)
   - Cherchez une « preuve de vie » sur un problème jouet pour valider votre implémentation
   - **Utilisez plusieurs « random seed »** et faite une moyenne de vos résultats

     Remarque : Plusieurs composantes sont stochastiques en RL ex. : l'initialisation de la politique, le modèle de l'environnement, politique exploratoire, ...
     Votre implémentation fonctionne peut-être seulement avec ce « random seed » spécifique sur cet environnement spécifique.

Bonne pratique en développement RL

L'importance des détails d'implémentation

« Est-ce que les détails d'implémentation sont importants en RL ? »

« Est-ce que les détails d'implémentation sont importants en RL ? »

Réponse courte : Oui ! Mais ça dépend du contexte, de l'environnement, de l'algorithme ...

Explication courte :

- **Il n'y a pas de solution « one size fits all » en RL** ;
- Certains détails peuvent ne pas avoir d'incidence dans un certain contexte, mais être critiques dans un autre ;
- Certains algorithmes sont plus fragiles dans certains environnements et peuvent donc nécessiter une attention particulière pour fonctionner ;
- Certaine contraintes au niveau des ressources physiques peuvent être déterminante au succès d'un projet ;

⋮

Recommandation

- Assurez-vous bien comprendre les subtilités de l'algorithme avant de commencer son implémentation ;
- Prenez le temps de planifier votre projet (ref. Plannification d'un projet de RL) ;

Note : Voir *Do implementation details matter in Deep Reinforcement Learning ?* (2019) par Luc Coupal [4] pour une réflexion détaillée sur le sujet. Blog post

« Est-ce que les détails d'implémentation sont importants en RL ? »

Réponse courte : Oui ! Mais ça dépend du contexte, de l'environnement, de l'algorithme ...

Explication courte :

- **Il n'y a pas de solution « one size fits all » en RL** ;
- Certains détails peuvent ne pas avoir d'incidence dans un certain contexte, mais être critiques dans un autre ;
- Certains algorithmes sont plus fragiles dans certains environnements et peuvent donc nécessiter une attention particulière pour fonctionner ;
- Certaine contraintes au niveau des ressources physiques peuvent être déterminante au succès d'un projet ;

⋮

Recommandation

- Assurez-vous bien comprendre les subtilités de l'algorithme avant de commencer son implémentation ;
- Prenez le temps de planifier votre projet (ref. Plannification d'un projet de RL) ;

Note : Voir *Do implementation details matter in Deep Reinforcement Learning ?* (2019) par Luc Coupal [4] pour une réflexion détaillée sur le sujet. ▸ Blog post

Ressources supplémentaires

Débogage et conduite d'expérience

- *The Nuts and Bolts of Deep RL Research* [5] présenté à l'université Berkeley dans le cadre du Deep RL Bootcamp (2017) ▸ Présentation ▸ Diapositives ▸ Deep RL Bootcamp
  par John Schulman
- *Advice for short term machine learning research projects* (2018) ▸ Blog post
  par Tim Rocktäschel, Jakob Foerster et Greg Farquhar
- *Lessons Learned Reproducing a Deep Reinforcement Learning Paper* (2018) ▸ Blog post
  par Amid Fish
- *Spinning Up as a Deep RL Researcher* (2018) [3] ▸ OpenAI Spinning Up
  par Joshua Achiam

Débogage en TensorFlow

- *A Practical Guide for Debugging TensorFlow Codes* (2017) ▸ Diapositives
  par Jongwook Choi
- *Testing Tensorflow code* (2017) ▸ Blog post
  par Guillaume Genthial

Pour aller plus loin

1 Planification d'un projet de RL

2 Bonne pratique en développement RL

3 Pour aller plus loin

- Complément théorique
- Ressources éducationnelles et frameworks de recherches
- Références

Pour aller plus loin

Complément théorique

Complément théorique

- *Reproducibility, Reusability, and Robustness in Deep Reinforcement Learning* ▸ Présentation
  par Joelle Pineau, ICLR 2018
- *TD or not TD : analyzing the role of temporal differencing in deep reinforcement learning* [6] ▸ Publication
  par Artemij Amiranashvili, Alexey Dosovitskiy, Vladlen Koltun et Thomas Brox
- *Deep Reinforcement Learning that Matters* [7] ▸ Publication Accompanying code
  par Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup et David Meger
- *An Opinionated Guide to ML Research* ▸ Blog post
  par John Schulman's

Pour aller plus loin

Ressources éducationnelles et frameworks de recherches

Ressources éducationnelles et frameworks de recherches

- *OpenAI | Spinning Up* [3] ▸ OpenAI | Spinning Up
  par Joshua Achiam, OpenAI
- *Reinforcement Learning Coach* ▸ Reinforcement Learning Coach
  par Intel AI Lab
- *OpenAI | Baselines* [8] ▸ OpenAI | Baselines
  par Dhariwal et al., OpenAI
- *Stable Baselines – RL Baselines Made Easy* (2018) [9] ▸ Stable Baselines
  par Hill et al.
- *TensorFlow | Agents* ▸ TensorFlow | Agents ▸ TensorFlow
  par Google
- *Ray | RLlib : Scalable Reinforcement Learning* [10] ▸ Ray | RLlib ▸ Ray
  par The Ray Team
- *Google | Dopamine* [11] ▸ Google | Dopamine
  par Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar et Marc G. Bellemare

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).
4. Coupal, L. Do implementation details matter in Deep Reinforcement Learning? *redleader962.github.io/blog*. https://redleader962.github.io/blog/2019/do-implementation-details-matter-in-deep-reinforcement-learning/ (2019).
5. Schulman, J. NIPS 2016 Tutorial: The Nuts and Bolts of Deep RL Research. http://rll.berkeley.edu/deeprlcourse/docs/nuts-and-bolts.pdf (2016).
6. Amiranashvili, A., Dosovitskiy, A., Koltun, V. & Brox, T. TD or not TD: Analyzing the Role of Temporal Differencing in Deep Reinforcement Learning. 1-14. arXiv : 1806.01175. http://arxiv.org/abs/1806.01175 (2018).

Références II

7. Henderson, P. *et al. Deep reinforcement learning that matters*. in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018* (2018), 3207-3214. ISBN : 9781577358008. arXiv : 1709.06560.

8. Dhariwal, P. *et al. OpenAI Baselines*. https://github.com/openai/baselines. 2017.

9. Hill, A. *et al. Stable Baselines*. https://github.com/hill-a/stable-baselines. 2018.

10. Liang, E. *et al. RLlib: Abstractions for distributed reinforcement learning*. in *35th International Conference on Machine Learning, ICML 2018* (2018). ISBN : 9781510867963. arXiv : 1712.09381.

11. Castro, P. S., Moitra, S., Gelada, C., Kumar, S. & Bellemare, M. G. Dopamine: A Research Framework for Deep Reinforcement Learning. http://arxiv.org/abs/1812.06110 (2018).