Projet Recherche

Combiner apprentissage par renforcement profond et méthodes évolutionnaires : l'algorithme CEM-ERL

Ben Kabongo, **Théo Charlot** et **Wassim Ouni**

Mars 2022

1 Introduction

L'apprentissage par renforcement consiste, pour un agent, à apprendre par le biais d'interactions avec un environnement les actions à réaliser dans chaque état, afin de maximiser la somme des récompenses au cours du temps. Dans **l'apprentissage par renforcement profond**, les agents sont des réseaux de neurones artificiels.

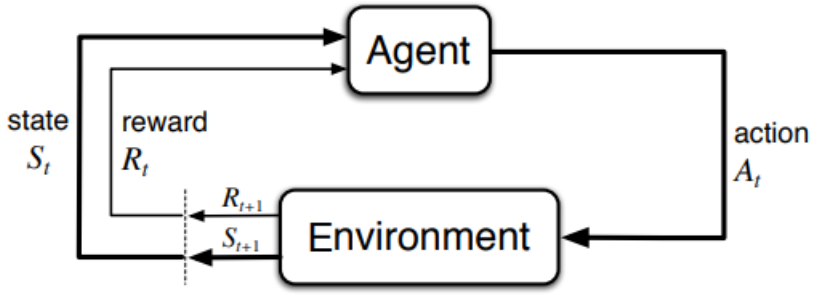

Figure 1: Apprentissage par renforcement

Les méthodes évolutionnaires s'inspirent de la théorie de l'évolution. Elles consistent en l'évolution d'une population dont les individus sont différentes solutions à un problème donné, suivie d'une sélection des meilleurs.

Notre projet consiste en la conception d'un algorithme qui combine l'apprentissage par renforcement profond et les méthodes évolutionnaires. Cet algorithme s'appelle **CEM-ERL**. Il est une combinaison des algorithmes **CEM-RL** et **ERL**.

2 Evolution et apprentissage

Quelques définitions :

- **Politique** ou stratégie : est un comportement décisionnel de l'agent, une fonction qui à tout état s associe l'action à exécuter. L'apprentissage par renforcement a pour but la recherche des politiques optimales.
- **Critique** : donne, pour chaque couple (état s, action a), une indication sur l'espérance des récompenses futures de l'agent s'il choisit l'action a depuis l'état s.
- **Acteur** : donne, pour chaque état s, l'action a avec la plus grande espérance des récompenses futures depuis l'état s.

- **Descente de gradient** : est un algorithme d'optimisation permettant de trouver le minimum d'une fonction en convergeant progressivement vers ce dernier. Ce procédé est utilisé dans les différents algorithmes pour mettre à jour les poids (les paramètres) des agents.
- **Expérience** : définie par un tuple (état courant s, action a, état suivant s', récompense r), qui est une transition de l'agent de l'état s à l'état s', après avoir effectué l'action a, qui lui apporté la récompense r.
- **Replay buffer** : tampon, mémoire dans laquelle sont stockées les expériences.
- **Episode** : succesion d'expériences avec un environnement jusqu'à un critère d'arrêt.
- **Score** : le score d'un individu correspond à la somme cumulée des récompenses obtenues à la suite d'interactions avec l'environnement au cours d'un épisode
- **Evaluation** : calcul du score des individus de la population.
- **Elites** : une proportion donnée d'individus d'une population aux scores les plus élevés après évaluation.

Les algorithmes **CEM-RL**, **ERL** et **CEM-ERL** sont tous une combinaison de l'apprentissage par renforcement profond et des méthodes évolutionnaires.

Ils commencent par initialiser une population d'individus et des agents d'apprentissage par renforcement profond (critiques et acteurs). A la suite d'épisodes et d'évaluations, une élite est sélectionnée afin de constituer la prochaine génération d'individus de la population.

A chaque épisode, les expériences des individus sont enregistrées dans un replay buffer. Les agents d'apprentissage par renforcenement utilisent les informations contenues dans ce dernier afin de se mettre à jour. Et à une certaine fréquence, les poids du réseau de neurones des critiques et des acteurs sont copiés ou injectés dans une partie ou la totalité de la population, selon l'algorithme.

3 Algorithmes

3.1 Cross Entropy Method

La **Cross Entropy Method (CEM)** est une méthode d'optimisation qui n'utilise pas de gradient. Elle part d'une distribution suivant une loi normale. Un nombre de vecteurs aléatoires (paramètres, trajectoires) représentant la population sont générés suivant cette distribution. Les vecteurs sont évalués et l'élite est sélectionnée. Afin de prévenir la convergence prématurée vers un optimum local, du bruit gaussien est rajouté autour de la moyenne de l'élite, donnant ainsi la distribution de la prochaine génération. La matrice de covariance donne des informations sur les différentes variables aléatoires des vecteurs.

3.2 CEM-RL

CEM-RL, quant à lui, combine **CEM** et un algorithme d'apprentissage par renforcement profond (TD3 ou DDPG).

L'algorithme commence par initialiser aléatoirement les individus de la population CEM ainsi qu'un critique pour l'apprentissage par renforcement.

Selon la matrice de covariance courante et à chaque itération, une population d'individus est échantillonnée par CEM, avec du bruit gaussien autour de la moyenne.

Le critique est mis à jour à l'aide des données générées par tous les individus. Puis il est utilisé pour mettre à jour la moitié des individus de la population en leur appliquant la direction de son nouveau gradient pour un nombre d'étapes fixe. Ensuite, tous les individus de la population sont évalués.

Les nouveaux paramètres de CEM sont calculés sur base de la moitié d'individus la plus performante de la population obtenue.

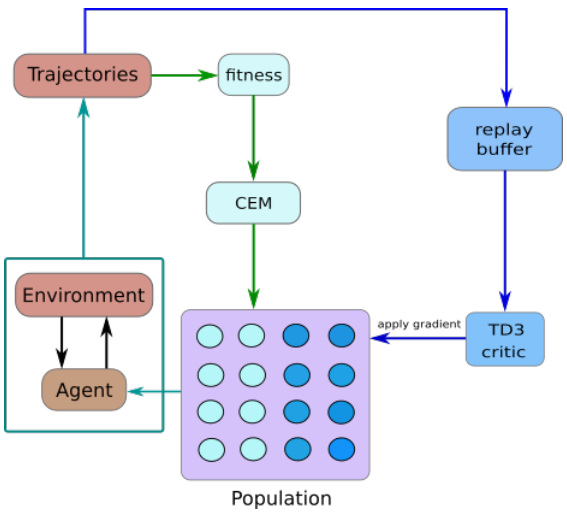

Figure 2: CEM-RL : Combining evolutionary and gradient-based methods for policy search

3.3 ERL : Evolutionary Reinforcement Learning

ERL initialise une population d'individus, par initialisation Glorot. Il s'agit d'un procédé d'initialisation de réseaux de neurones permettant d'obtenir des meilleures performances.

L'algorithme initialise également un critique et un acteur d'apprentissage par renforcement.

Au fil des épisodes, on sélectionne les élites. Ces derniers sont préservés, et le reste des individus sélectionnés subissent mutations et croisements. Les individus obtenus et les élites constituent les individus de la prochaine génération. Egalement, le critique et l'acteur se mettent à jour.

Périodiquement, les poids de l'acteur sont copiés dans un individu de la population en évolution, afin de tirer parti des informations apprises par la méthode de descente de gradient et de stabiliser l'apprentissage. Si la politique apprise est bonne, elle sera sélectionnée et étendue dans les générations suivantes. Sinon, elle ne sera pas sélectionnée. Cela garantit que les informations obtenues de l'acteur soient constructives.

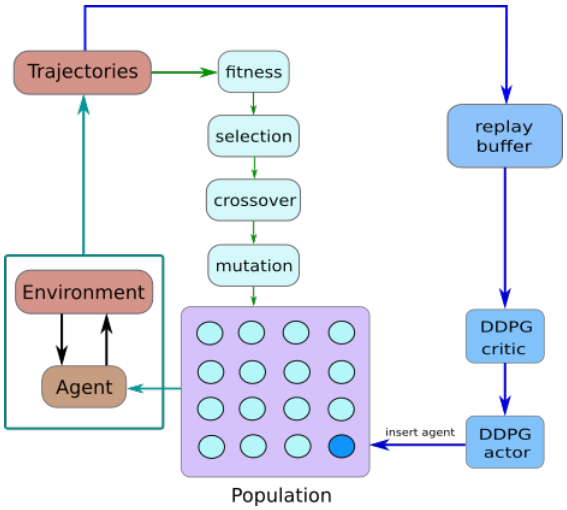

Figure 3: ERL

3.4 CEM-ERL : Combinaison de CEM-RL et de ERL

CEM-ERL est l'algorithme que nous devons implémenter pour notre projet, pour ensuite comparer ses performances à celles de CEM-RL et ERL, dont il fusionne les principes fondamentaux.

L'algorithme initialise une population d'individus CEM, un acteur et un critique pour l'apprentissage par renforcement.

Selon la matrice de covariance et à chaque iteration, CEM échantillonne une population. Périodiquement, l'acteur est rajouté aux individus de la population CEM. A la suite d'épisodes et d'évaluations, les élites sont sélectionnées. Ils déterminent les nouveaux paramètres de la distribution de la prochaine génération.

Une première intuition est qu'en début d'apprentissage, l'acteur rajouté dans la population aura très peu de chances de faire partie des élites ; mais qu'au fur et à mesure de l'apprentissage, ses chances d'en faire partie seront plus grandes.

4 Evolution du projet

Nous avons commencé ce projet par la compréhension de l'apprentissage par renforcement profond et des méthodes évolutionnaires. Une fois les bases maîtrisées, nous nous sommes intéressés à des implémentations des différents algorithmes en **Python**. Nous avons choisi d'utiliser la bibliothèque **Salina**, qui est dédiée à l'apprentissage par renforcement.

Un premier travail à réaliser pour ce projet a été d'implémenter l'algorithme **CEM-RL** en utilisant Salina. Dans la suite, nous allons implémenter **CEM-ERL**.

Après l'implémentation de CEM-ERL, nous procéderons à l'analyse de l'algorithme. Nous ferons ensuite des visualisations expérimentales ainsi que des comparaisons avec les algorithmes CEM-RL et ERL.