Ananas : Apprentissage de conduite automatique par Deep Q Learning

Deep Learning - Projet

Clément, Grégoire, Nathan

January 31, 2025

Introduction

But du projet

Implémenter une voiture qui apprend à conduire en utilisant le Deep Reinforcement Learning.

Pour des raisons d'efficacité temporelle, reimplémentation de notre propre environement plutôt que d'en utiliser un déjà tout fait

Source principale : *Playing Atari with Deep Reinforcement Learning*

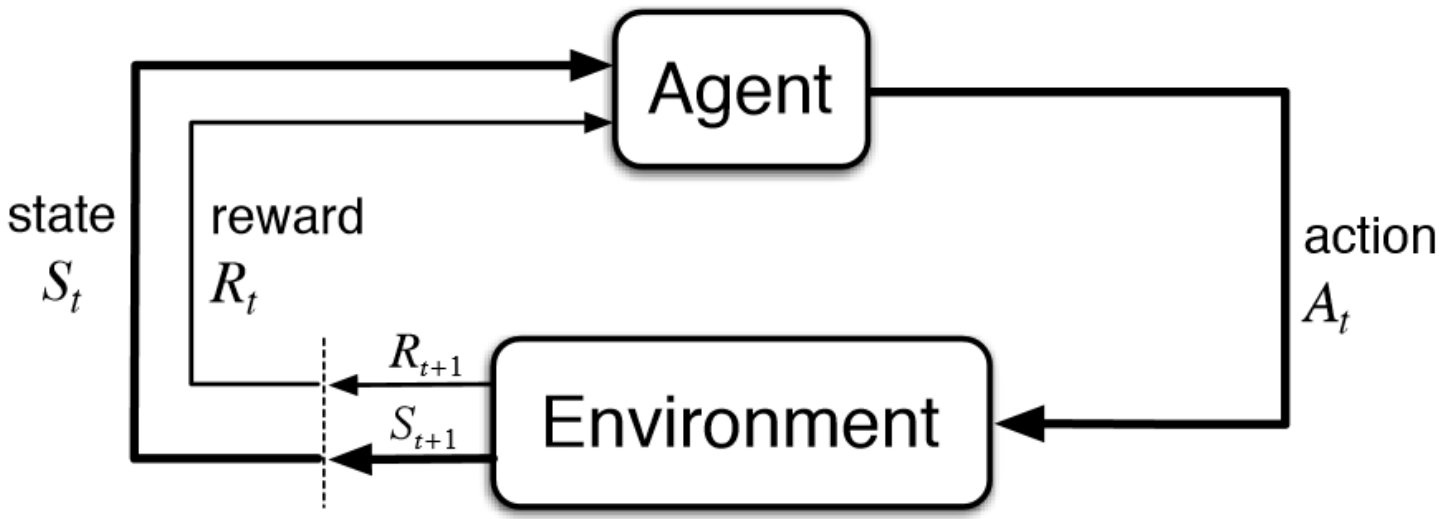
Reinforcement learning

A quoi ça sert

Reinforcement learning : Agent qui évolue dans un environement

Pour chaque action, récompenses ou pénalité.

Objectif : Maximiser les récompense



Formulation mathématiques

L'agent intéragit avec un environement stochastique $\mathcal{E}$ dans lequel il joue des partie composé d'un ensemble d'état, de score et d'action

A chaque étape, il sélectionne une action a_t parmis un ensemble $A = \{1, ..., K\}$ d'action légale

A chaque étape, l'agent a accès à un ensemble d'information $x_t \in \mathbb{R}^d$.

Objectif de l'agent

On définit le rendement futur attendu par

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$$

où T est l'instant auquel le jeu s'arrête.

But de l'agent : maximiser ce rendement.

Optimal action value function

On définit l'optimal action value function $Q*(s', a')$ qui donne le bon choix à faire en fonction de la situation actuelle, cest à dire

$$Q^*(s, a) = \mathbb{E}_\pi[R_t | s_t = s, a_t = a, \pi]$$

Si on dispose de cette fonction, alors il suffit de faire le meilleur choix à chaque fois.

Comment calculer cette fonction

Cette fonction suit l'équation de Bellman:

$$Q^*(s,a) = \mathbb{E}_{s' \sim \mathcal{E}}[r + \gamma \cdot \max_{a'} Q^*(s',a')|s,a]$$

Alors si on pose

$$Q_{i+1}(s,a) = \mathbb{E}[r + \gamma \max_{a'} Q_i(s',a')|\ s,a]$$

Q_i tend vers $Q*$ lorsque i tend vers $+\infty$ Problème : Trop dur à estimer, trop de situations différentes possibles

Estimation de la fonciton par un réseau

Solution : On approxime Q^* par un réseau de neurone $Q(s, a, \theta)$
On l'entraîne en minimisant les fonctions

$$L_i(\theta_i) = \mathbb{E}_{s,a\sim\rho(.)}[(y_i - Q(s, a; \theta_i))^2]$$

Deep Q Learning

Equations blabla comment ça marche

Environement de simulation

Pourquoi recoder notre propre enviornement

Inconvénient d'un émulateur : Temps d'accès potentiellement long, peu de flexibilité sur les informations qu'on peut passer à notre modèle donc moins d'experimentations possibles

En recodant notre simulation, on peut contourner tous ces problèmes

Notre simulation

La on met des images de différents exemples

Entraînement, abblation study et tout

Conclusion