Reinforcement Learning – Street Fighter II

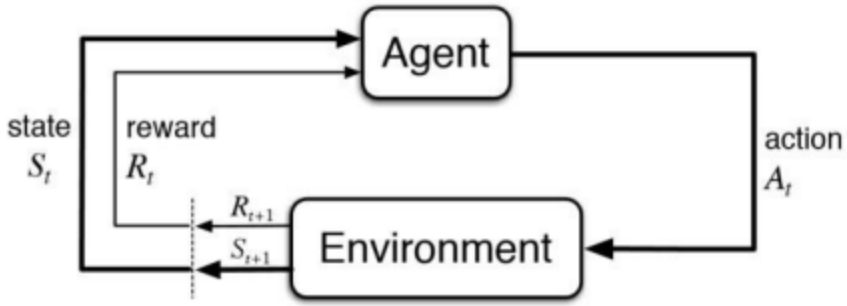
Nicolas LE ROUX
Antoine LUU
Julia FOUCHIER

Plan de la présentation

- Contexte et application
- Théorie et principe de fonctionnement
- Présentation des données traitées
- Simulation et problèmes rencontrés
- Conclusion et pistes d’amélioration

Contexte et application

/ **Définition : L'apprentissage par renforcement** est une méthode d'auto-apprentissage pour les systèmes autonomes, où ces derniers apprennent à prendre des actions en fonction de l'expérience acquise, dans le but d'optimiser une récompense cumulative sur une période de temps.



/ **Nombreux algorithmes :**
- / Q-learning
- / Monte Carlo
- / PPO

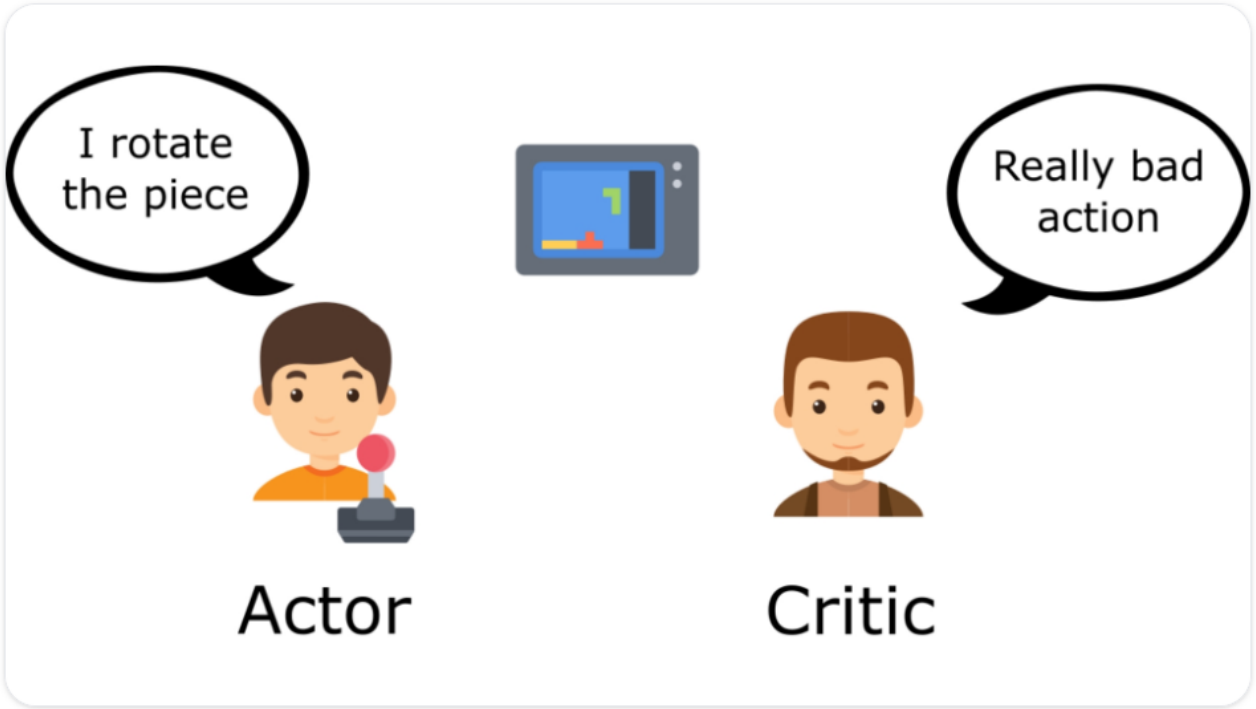
/ **Nombreuses applications :**
- / Véhicules autonomes ;
- / Trading ;
- / NLP ;
- / Gaming ;
- / Robotique
- / …

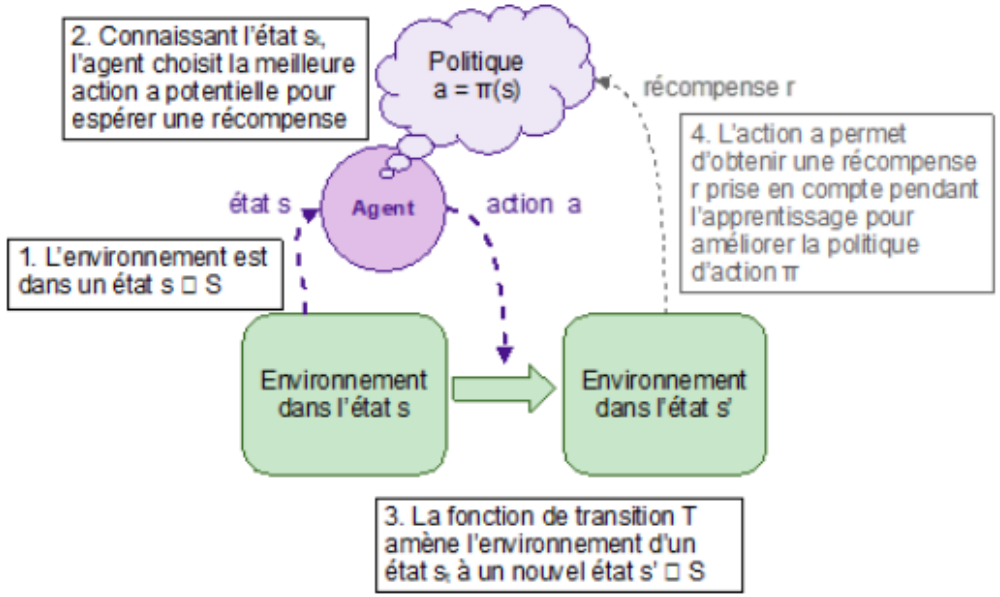
Contexte et application

- 2015 : Place importante pour les jeux Atari (*Mnih et al*).
- 2018 : AlphaGo Zero (*Hessel et al*) : l'agent apprend en étant son propre professeur.
- **Objectif** : Entraîner une IA grâce à l'apprentissage par renforcement pour jouer au jeu Street Fighter II.

Théorie et principe de fonctionnement

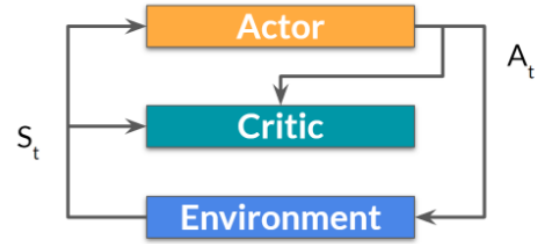
/ Première notion : Actor – Critic Process



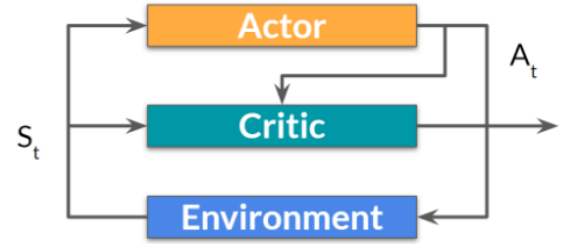


Théorie et principe de fonctionnement

/ Première notion : Actor – Critic Process

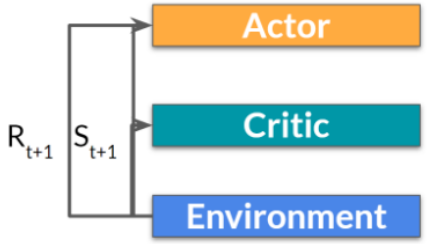


Step 1



Step 2

$$\hat{q}_w(S_t, A_t)$$



Step 3

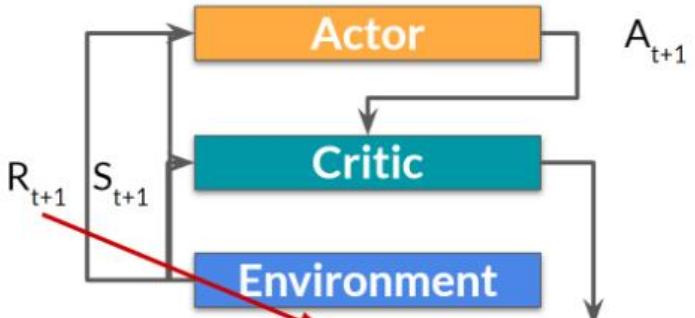
Step 4

$$\Delta\theta = \alpha\nabla_\theta(log\pi_\theta(s,a))\hat{q}_w(s,a)$$

Change in policy parameters (weights)

Action value estimate

Step 5



$$\Delta w = \beta\,(R(s,a) + \gamma\hat{q}_w(s_{t+1},a_{t+1}) - \hat{q}_w(s_t,a_t))\nabla_w\hat{q}_w(s_t,a_t)$$

Policy and value have different learning rates

TD error

Gradient of our value function

Théorie et principe de fonctionnement

- **Proximal Policy Optimization (PPO)**
  - Méthode d'apprentissage profond basée sur le Deep-Q learning (DQL)
  - Détermination d'une politique optimale en interagissant avec l'environnement
  - Stabilité de sa politique

Théorie et principe de fonctionnement

/ Proximal Policy Optimization (PPO)

$$L^{PG}(\theta) = E_t[\log \pi_\theta(a_t|s_t) * A_t]$$

log probability of taking that action at that state

Advantage if A>0, this action is better than the other action possible at that state

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

As we can see, $r_t(\theta)$ denotes the probability ratio between the current and old policy:

- If $r_t(\theta) > 1$, the **action a_t at state s_t is more likely in the current policy than the old policy.**
- If $r_t(\theta)$ is between 0 and 1, the **action is less likely for the current policy than for the old one.**

So this probability ratio is an **easy way to estimate the divergence between old and current policy.**

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}_t\left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)\right]$$

c1 and c2 are coefficients.

Add an entropy bonus to ensure sufficient exploration

Squared-error value loss: $(V_\theta(s_t) - V_t^{targ})^2$

Explications des données manipulées I

Gymnasium

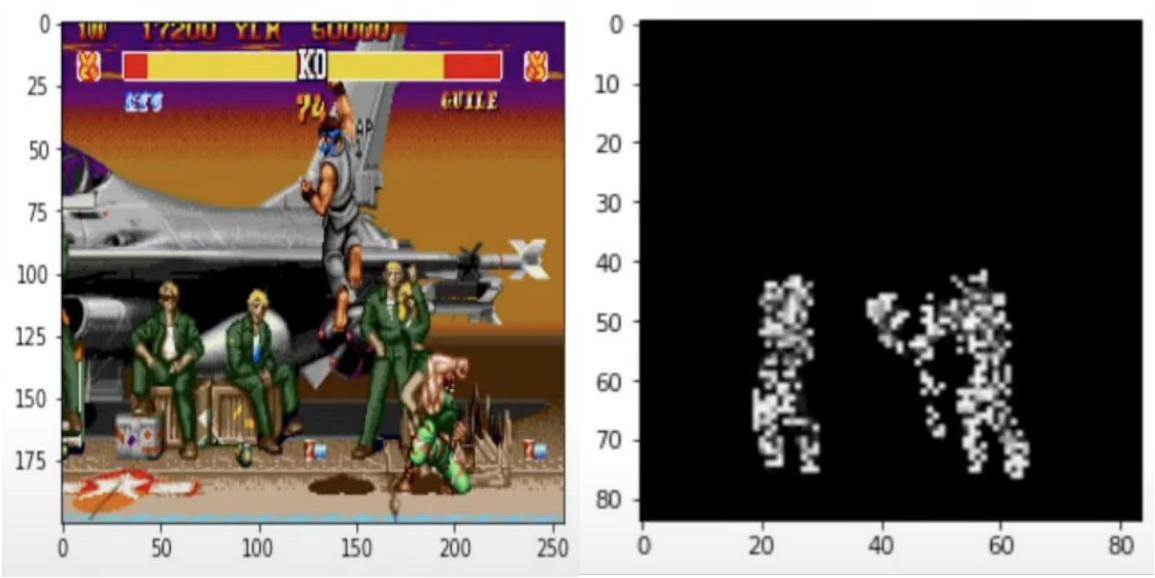
- **Gymnasium :** Projet d'OpenAI, offre des API pour des projets de RL sur différentes applications, dont des jeux Atari.
- **Env :** classe de Gym (module de Gymnasium) pour simuler les jeux, sépare l'interface des données.
  - **Variables d'instance :**
    - observation_space : première valeur de obs (cf. Step())
    - action_space : vecteur qui représente toutes les actions possibles (12 coordonnées pour StreetFighter II)
  - **Méthodes d'instance :**
    - **step() :** Avance la simulation d'un pas, et renvoie:
      - obs : observation de l'état de la simulation, sous forme de matrice (200,256,3) de base.
      - reward : le résultat de la reward function suite au step.
      - done : booléen pour terminer le jeu
      - info : dictionnaire avec des informations en anglais sur l'état du jeu (score, parties gagnées, points de vie, etc.)
    - **reset() :** Remettre la simulation à 0
    - **render() :** Génère le visuel du jeu
    - **close() :** ferme l'instance de Env, nécessaire car on ne peut pas en avoir plusieurs en même temps.

Présentation des données traitées

- Emulateur : Utilisation de GymRetro et de roms
- Démonstration

Explications des données manipulées II

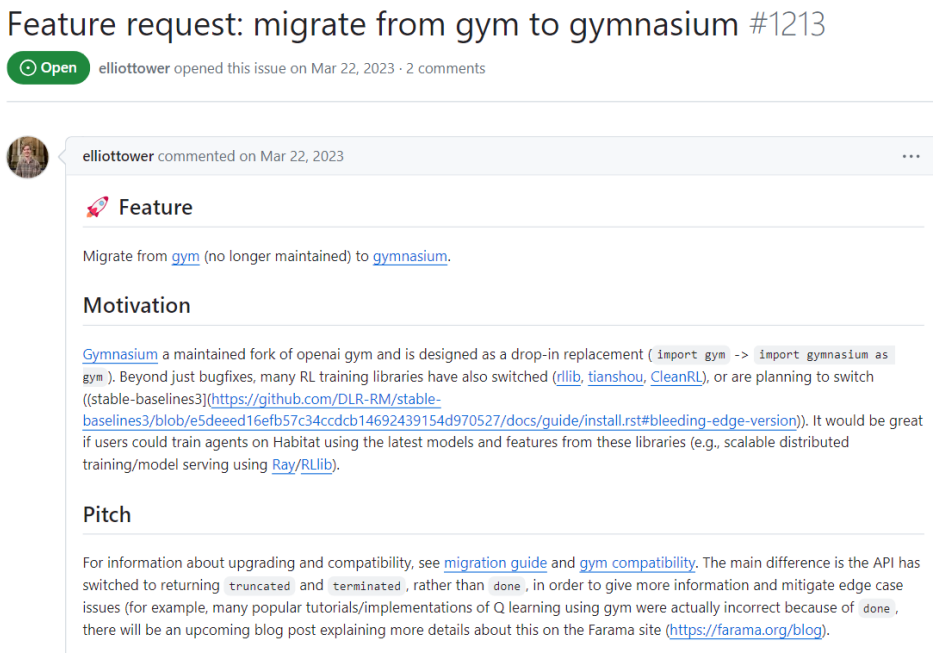
- **StreetFighter :** nouvelle classe qui hérite de Env
  - **Nouvelles méthodes :**
    - **preprocess() :** pour passer en niveaux de gris, réduire la taille de (200, 256, 3) à (84, 84, 1)
  - **Modification des méthodes :**
    - Changement du constructeur incorporer preprocess
    - Changements de step : changement de reward (passage de matchs gagnés au score), l'obs devient le frame_delta



Simulation et problèmes rencontrés

/ Problème de versions

- / La resource suivie ne présente pas ses requirements.
- / Les dernières versions de Gymnasium et Stable Baselines 3 ne sont pas compatibles



Feature request: migrate from gym to gymnasium #1213

Open elliottower opened this issue on Mar 22, 2023 · 2 comments

elliottower commented on Mar 22, 2023

Feature

Migrate from gym (no longer maintained) to gymnasium.

Motivation

Gymnasium a maintained fork of openai gym and is designed as a drop-in replacement (`import gym` -> `import gymnasium as gym`). Beyond just bugfixes, many RL training libraries have also switched (rllib, tianshou, CleanRL), or are planning to switch ((stable-baselines3](https://github.com/DLR-RM/stable-baselines3/blob/e5deeed16efb57c34ccdcb14692439154d970527/docs/guide/install.rst#bleeding-edge-version)). It would be great if users could train agents on Habitat using the latest models and features from these libraries (e.g., scalable distributed training/model serving using Ray/RLlib).

Pitch

For information about upgrading and compatibility, see migration guide and gym compatibility. The main difference is the API has switched to returning `truncated` and `terminated`, rather than `done`, in order to give more information and mitigate edge case issues (for example, many popular tutorials/implementations of Q learning using gym were actually incorrect because of `done`, there will be an upcoming blog post explaining more details about this on the Farama site (https://farama.org/blog).

Conclusion et pistes d'amélioration

Pistes d'améliorations

- Partir sur un modèle plus simple de RL, sans utiliser d'autre librairie, par exemple un simple Q-Learning. Limité en temps

- Récupérer un tutoriel avec des versions à jour

- Références:
  Build a Street Fighter AI Model with Python, Nicholas Renotte
  https://www.youtube.com/watch?v=rzbFhu6So5U&t=9s
- Hugging face
  https://huggingface.co/blog/deep-rl-ppo
- Gym retro
  https://openai.com/research/gym-retro