AEQUAM Capital

—

Projet Reinforcement Learning

Réunion du 3 juillet 2019

Avancement 1/2 : Cadre général configuré

- Scripts:
  - « **trading_environment.py** » permet de définir un **environnement de trading** : actif à échanger, variables et transformées considérées, fenêtre temporelle, philosophie du signal recherché, … permet aussi, pour une **stratégie** implémentée, d'en suivre le **comportement au cours du temps** en termes de positions, performances, etc
  - « **rl_policy.py** » permet de définir une **« politique » et son apprentissage** : comment une action est décidée (en fonction de quoi), comment l'apprentissage se déroule, etc
  - « **launch_train.py** » permet de lancer un **apprentissage complet**, à partir d'un environnement de trading et d'une politique (cf supra), et d'en **rapporter les résultats et la progression** visuellement (plus ou moins…) avec un **fichier pdf**

Avancement 2/2 : Premiers entraînements lancés

- Premiers rapports : **peu de signe d'apprentissage**, **différence** selon philosophie de signal

- https://drive.google.com/file/d/15RiNboxt-kayCix0u14gwWYGExX-mYnl/view?usp=sharing
- https://drive.google.com/file/d/1h1tacQfa4f0ErQ9y8JrbJwOOMdWQoLbU/view?usp=sharing

- Buy-and-hold n'est pas l'allocation optimale recherchée, de la même manière qu'un signal errant de façon chaotique entre 0 et 1

- Evolution en fonction des choix déjà effectués…

Choix faits / implicites dans le design et modifications envisagées

- *Reward* : gain brut par rapport au portefeuille de départ (indifférence entre gain absolu et rendement) : à modifier (sharpe, sharpe modifié, etc.)
- *Réseau de neurones* pour décision à prendre : à adapter ou à tester autre chose (en termes d'algorithmes et/ou de spécification)
- *Durée de l'épisode* : un historique complet (**à réduire drastiquement !**)
- *Vitesse d'apprentissage* : faible (mais apprend déjà vite quand apprentissage il y a)
- (*Taux de discount temporel* : méthode y semble peu sensible)

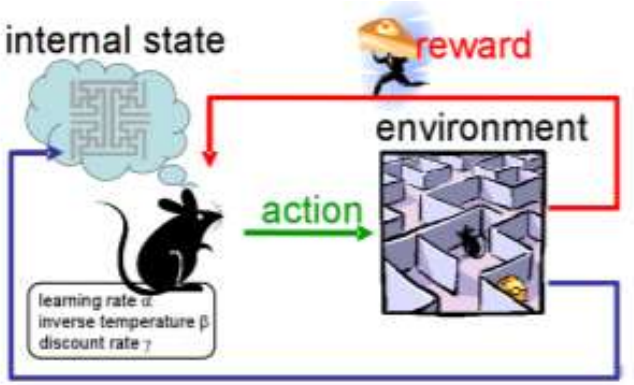
Pistes de recherche / d'avancement

- Tout ce qui a été fait est hautement modulable → de nombreux tests pendant la période d'absence (variables, transformées, nombre d'épisodes, rapidité d'apprentissage, méthode d'apprentissage, …)
- Bien comprendre les premiers résultats pour faire les bons choix ensuite
- Mettre tous les codes propres sur Github pour review pendant la période d'absence

- (Wikis à créer et alimenter sur ce qui a été fait, les choix pris)
- (Faire un rapport plus lisible)

Annexes

Formalisation théorique



- **Agent**/environnement
- **Etat** (***state***) : ce que perçoit l'agent
- **Action** : interaction de l'agent avec l'environnement
- **Récompense** (***reward***) : quantité perçue après chaque action
- **Politique** (***policy***) : une fonction de sélection de l'action selon l'état

Objectif : trouver une politique qui permet de maximiser l'ensemble des récompenses reçues

Source : http://dac.lip6.fr/master/wp-content/uploads/2019/01/ARF-2018-cours8.pdf (slide 3)