

# Gestion de portfolio par un RL *trading* BOT

*Etape 3 : Résultats finaux*

*Etape 4 : Sujets avancés*

**Projet - Planification en Intelligence Artificielle IFT 602 / IFT 708**

Présenté par :  
Tahar Adam AMAIRI  
Victor CHAU  
Omar CHIDA  
Céline ZHANG

Professeur : Froduald Kabanza  
Assistants : D'Jeff Nkashama & Jordan Félicien Masakuna



# **Plan de présentation**

- I. Introduction
- II. Etat de l'art
- III. Méthodologie
- IV. Démonstration (Résultats)
- V. Références



# I. Introduction

- Précédemment à l'étape 2 :
  - **Implémentation** : PPO (Proximal Policy Optimization)
  - **Résultats** : Mitigés dans l'ensemble, TDQN reste plus performant
  - **A venir** : Ajout d'un réseau LSTM
- Aujourd'hui :
  - Etape 3 :
    - **Implémentations** : TDQN + LSTM, PPO + LSTM
    - **Amélioration** : Restructuration du code
  - Etape 4 :
    - **Implémentations** : TDQN + GRU, PPO + GRU, Contexte, TDCQN



## II. Etat de l'art

- **Réseaux récurrents** : classe de réseaux de neurones adaptés pour les séries temporelles
- **Structure** : comparables à des réseaux de neurones classiques avec des contraintes d'égalité entre les poids du réseau
- **Problème** : disparition du gradient



## II. Etat de l'art

- **Solution** : Long short-term memory (LSTM)
- Proposée par **Hochreiter et al. (1997)**
- Permet de stocker des informations à long terme
- → idéal pour les données chronologiques.
- **Fonctionnement** :
  - Etape oubli
  - Etape d'entrée
  - Etape de sortie



## II. Etat de l'art

- **LSTM pour le trading :**
  - Données séquentielles (ex : les cours des actions)



## II. Etat de l'art

- **LSTM pour le trading :**
  - Données séquentielles (ex : les cours des actions)  
→ autres facteurs dépendant du temps



## II. Etat de l'art

- **LSTM pour le trading :**
  - Données séquentielles (ex : les cours des actions)  
→ autres facteurs dépendant du temps
  - Apprendre des politiques optimales dans des environnements où les transitions d'état ont une dépendance temporelle.

→ **Compréhension plus complète du marché**





## II. Etat de l'art

- **Solution** : Gated Recurrents Units (GRU)
- Proposée par **Kyunghyun Cho et al. (2014)**
- Permet aussi l'apprentissage des informations à long terme
- Moins complexe que LSTM
- **Fonctionnement** :
  - 2 portes au lieu de 3 :
    - une porte de réinitialisation
    - une porte de mise à jour.



## II. Etat de l'art

- Bloc Convolutif avec TDQN



## II. Etat de l'art

- **Bloc Convolutif avec TDQN**
  - Extraction de caractéristiques pertinentes pour prendre des décisions commerciales (Identification de tendances, de saisonnalités et d'anomalies)
  - Prédiction des mouvements du cours des actions
  - Incorporation du bloc convolutif pour informer le processus d'apprentissage Q



## II. Etat de l'art

- **Bloc Convolutif avec TDQN**
  - Extraction de caractéristiques pertinentes pour prendre des décisions commerciales (Identification de tendances, de saisonnalités et d'anomalies)
  - Prédiction des mouvements du cours des actions
  - Incorporation du bloc convolutif pour informer le processus d'apprentissage Q
- **Bloc Linéaire avec TDQN**
  - Réduction de la dimensionnalité des données d'entrée
  - Extraction de caractéristiques utiles pour l'apprentissage de la fonction Q
  - Stabilisation du processus d'apprentissage TDQN
  - Réduction du coût de calcul de la formation du réseau neuronal



## II. Etat de l'art

- **Bloc Attention**

- Ignorance des parties moins pertinentes
- Utilisation de matrices de requête, de clé et de valeur
- Amélioration de la précision et de l'efficacité des modèles
- Traitement sélectif de différentes parties des données financières, telles que des actions spécifiques ou des tendances de marché

# III. Méthodologie

- **Restructuration** du code existant : simplifie la modification d'architecture
  - Implémentation de classes pour chaque modèle d'apprentissage :
    - **DRLAgent** (classe mère)
    - **TDQN** (*Trading* DQN de **base**)
    - **TDRQN** (TDQN à réseau **récurrent**)
    - **TDCQN** (TDQN à réseau **convolutif**)
    - **PPO** (*Policy-Gradient* avec un réseau **Actor-Critic**)
  - Modification simplifiée des hyperparamètres des modèles
  - Préparation à l'ajout de contexte supplémentaire (informations du marché)

# III. Méthodologie

- **Restructuration** du code existant : simplifie la modification d'architecture
  - Implémentation de classes pour chaque modèle d'apprentissage :
    - **DRLAgent** (classe mère)
    - **TDQN** (*Trading* DQN de **base**)
    - **TDRQN** (TDQN à réseau **récurrent**)
    - **TDCQN** (TDQN à réseau **convolutif**)
    - **PPO** (*Policy-Gradient* avec un réseau **Actor-Critic**)
  - Modification simplifiée des hyperparamètres des modèles
  - Préparation à l'ajout de contexte supplémentaire (informations du marché)
- Intégration du **réseau LSTM** : permet l'**adaptation des poids** sur les données
  - **Mémoire** à court terme et à long terme : **poids ajustable**
  - Résiste à la **disparition du gradient** : couches **profondes**
  - Prédiction pour **t + n** : **fit** les **séries temporelles**
  - Application à l'apprentissage **TDQN** et **PPO**



## III. Méthodologie

- Intégration de **contexte supplémentaire** au *bot* : augmente la visibilité de l'agent
  - Ajout de nouvelles variables d'observation à l'entrée des modèles
  - Application du contexte sur les stratégies d'apprentissage





## III. Méthodologie

- Intégration de **contexte supplémentaire** au *bot* : augmente la visibilité de l'agent
  - Ajout de nouvelles variables d'observation à l'entrée des modèles
  - Application du contexte sur les stratégies d'apprentissage
- Intégration du réseau **GRU** (récurrent) : en comparaison avec **LSTM**
  - Une variante de LSTM avec **moins de paramètres**
  - **Réduction** du **temps** d'apprentissage et de la **complexité** mémoire
  - Application à l'apprentissage **TDQN** et **PPO**

# III. Méthodologie

- Intégration de **contexte supplémentaire** au *bot* : augmente la visibilité de l'agent
  - Ajout de nouvelles variables d'observation à l'entrée des modèles
  - Application du contexte sur les stratégies d'apprentissage
- Intégration du réseau **GRU** (récurrent) : en comparaison avec **LTSM**
  - Une variante de LSTM avec **moins de paramètres**
  - **Réduction** du **temps** d'apprentissage et de la **complexité** mémoire
  - Application à l'apprentissage **TDQN** et **PPO**
- Intégration des réseaux **customisés** : exploration de nouvelles horizons
  - Bloc **Convolutif** : réseau de neurones convolutif
  - Bloc **Linéaire** : t'ransformation linéaire aux données d'entrée
  - Bloc **Attention** : réseau de neurones basé sur l'attention



## IV. Démonstration

Allons voir le code !



## V. Références bibliographiques

- [1] Thibaut Théate, Damien Ernst. *An application of deep reinforcement learning to algorithmic trading* : <https://www.sciencedirect.com/science/article/pii/S0957417421000737>
- [2] Code source du *trading* l'algorithme de Thibaut Théate et Damien Ernst : <https://github.com/ThibautTheate/An-Application-of-Deep-Reinforcement-Learning-to-Algorithmic-Trading>
- [3] Hongyang Yang, Xiao-Yang Liu, Shan Zhong, Anwar Walid. Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy : [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=3690996](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3690996)
- [4] Wouter van Heeswijk. Proximal Policy Optimization (PPO) : <https://towardsdatascience.com/proximal-policy-optimization-ppo-explained-abed1952457b>