Rapport sur le Mini-Projet sur l'apprentissage par renforcement

Membre du groupe:

- Denilson NZOGNENG
- Johanu GANDONOU

2.1 Compréhension du DQN

L'algorithme Deep Q-Network (DQN) est une technique d'apprentissage par renforcement qui associe l'algorithme classique de Q-learning à des réseaux neuronaux profonds. Cette approche permet de traiter des problèmes complexes dans des environnements comportant un grand nombre d'états et d'actions, où il devient impossible d'utiliser des tables classiques pour stocker les valeurs $Q(s,a)$. Le Q-learning est un algorithme classique d'apprentissage par renforcement (RL) dont l'objectif de base est d'apprendre une fonction d'action-valeur Q(s,a) qui associe à chaque état s et action a a une valeur représentant la récompense future attendue. L'objectif est de déterminer la politique optimale qui maximise la récompense cumulée sur le long terme.

références:

- https://medium.com/@samina.amin/deep-q-learning-dqn-71c109586bae
- https://www.geeksforgeeks.org/machine-learning/q-learning-in-python

2.2 Décrivez le contexte de votre domaine d'application du DQN et justifier le pourquoi du DQN

Contexte : Trading Algorithmique sur le NASDAQ-100

Le **NASDAQ-100** est un indice boursier composé des 100 plus grandes entreprises non-financières cotées au NASDAQ, incluant des géants technologiques comme Apple, Microsoft, Amazon, Tesla et Google. Le trading sur cet indice représente un environnement dynamique, complexe et hautement volatil où les décisions d'achat, de vente ou de conservation d'actifs doivent être prises en temps réel en fonction de multiples facteurs de marché.

Notre application consiste à développer un **agent de trading autonome** capable de :
- Analyser l'évolution des prix et des indicateurs techniques du NASDAQ-100
- Prendre des décisions optimales (Acheter, Vendre, Conserver) pour maximiser le profit
- S'adapter aux conditions de marché changeantes (tendances haussières, baissières, volatilité)
- Gérer le risque en évitant les pertes importantes

Justification du choix du DQN

Le **Deep Q-Network (DQN)** est particulièrement adapté au trading algorithmique pour plusieurs raisons fondamentales :

- **Prise de décision séquentielle dans un environnement incertain**

Le trading est un problème de décision séquentielle où chaque action (achat, vente) influence l'état futur du portefeuille et les opportunités futures. Le DQN excelle dans ce type d'environnement car il apprend à maximiser les récompenses cumulées à long terme (profit total) plutôt que de se concentrer uniquement sur des gains immédiats.

- **Espace d'états complexe et continu**

L'état du marché est représenté par de nombreuses variables continues. Les réseaux de neurones profonds du DQN peuvent traiter cet espace d'états multi-dimensionnel et extraire des patterns complexes que les méthodes traditionnelles (Q-Learning tabulaire) ne peuvent pas gérer.

- **Apprentissage sans modèle explicite du marché**

Les marchés financiers sont non-stationnaires et imprévisibles. Le DQN n'a pas besoin d'un modèle mathématique explicite du marché (qui serait imprécis et difficile à construire). Il apprend directement de l'interaction avec l'environnement, ce qui le rend robuste aux changements de dynamique du marché.

- **Gestion de la mémoire de répétition (Experience Replay)**

Le DQN utilise une mémoire de répétition pour stocker les expériences passées (état, action, récompense, état suivant) et les réutilise aléatoirement pour l'entraînement. Cette technique :
- Brise les corrélations temporelles entre les échantillons successifs (crucial en trading où les données sont séquentielles)
- Permet un apprentissage plus stable et efficace
- Évite le sur-apprentissage sur des patterns récents

- **Stabilisation avec le réseau cible (Target Network)**

Le trading présente une forte variance dans les récompenses (profits/pertes). Le DQN utilise un réseau cible fixe temporairement pour calculer les valeurs Q futures, ce qui stabilise l'apprentissage et évite les oscillations dans la politique de trading.

- **Avantages par rapport à l'apprentissage supervisé classique**

Contrairement aux approches supervisées qui prédisent simplement les prix futurs :
- Le DQN optimise directement la stratégie de trading (politique d'action)
- Il intègre naturellement le compromis exploration/exploitation
- Il apprend à gérer le risque à travers le système de récompenses
- Il considère l'impact des actions passées sur les opportunités futures

- **Adaptabilité aux différents régimes de marché**

Le marché passe par différentes phases (bull market, bear market, consolidation). Le DQN peut apprendre des stratégies différentes pour chaque régime sans programmation explicite de règles, ce qui le rend plus flexible que les stratégies de trading algorithmique traditionnelles basées sur des règles fixes.

2.3 Architecture de l'agent-apprenant DQN pour le trading du NASDAQ-100**

- **Espace d'États (State Space)**

L'état capture l'information du marché et du portefeuille sur une **fenêtre glissante de 10 pas de temps** :

Features par timestep (16 au total) :
- Prix OHLCV (5) : Open, High, Low, Close, Volume normalisés
- Indicateurs techniques (11) : SMA_10, SMA_30, EMA_12, RSI_14, MACD, MACD_Signal, MACD_Hist, BB_Upper, BB_Lower, ATR_14, Return
- **Dimension totale** : 10 timesteps × 16 features = 160 dimensions

- **Espace d'Actions (Action Space)**

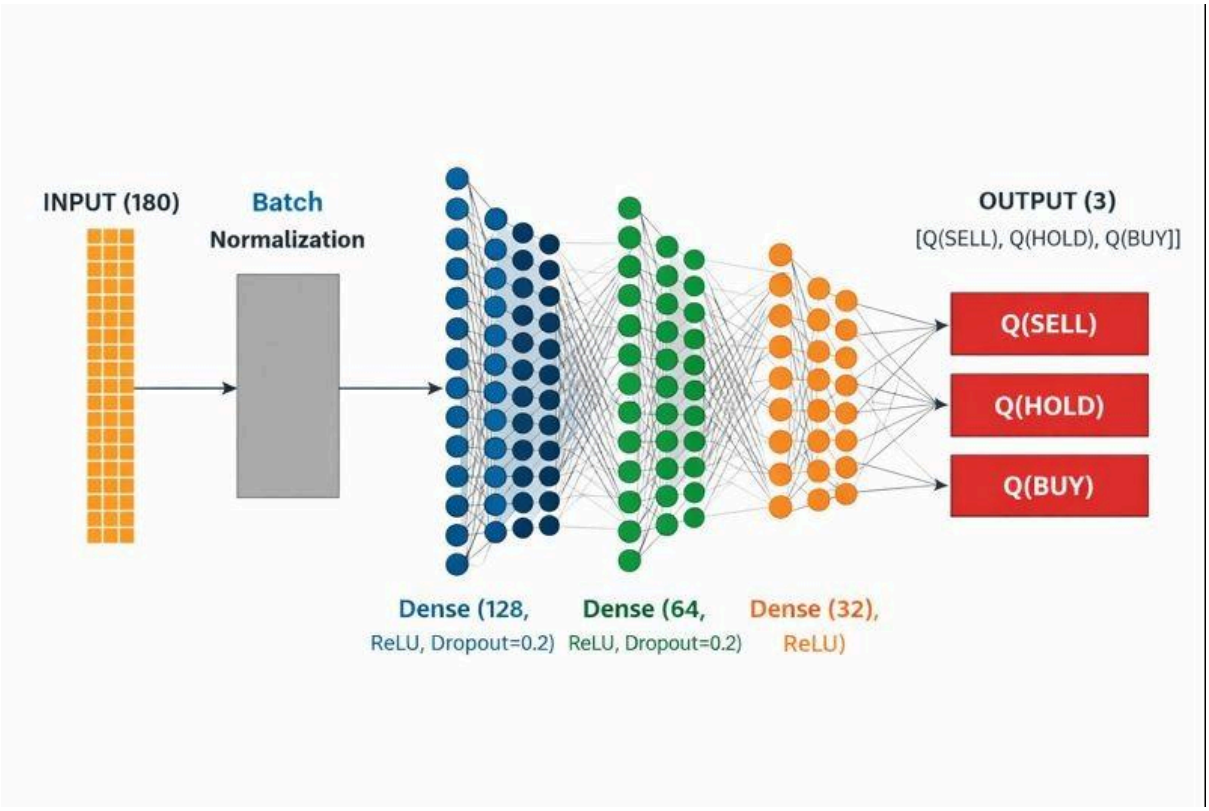
3 actions discrètes :
- **HOLD (0)** : Conserver la position
- **BUY (1)**: Acheter avec 100% du cash disponible
- **SELL (2)** : Vendre 100% des actions détenues

- **Fonction de Récompense**

Lorsque l'agent effectue une transaction, la différence de prix entre la dernière transaction et l'actuelle est utilisée pour calculer la récompense. Si l'agent vend à un prix plus élevé que son prix d'achat (position longue), il reçoit une récompense positive. Cependant, si l'agent vend à un prix inférieur à son prix d'achat (position longue), la récompense sera négative, ce qui représente une perte.

La récompense dans ton système de trading guide l'agent à prendre des décisions qui maximisent son profit en fermant des positions, que ce soit pour des positions longues ou courtes.

- **Architecture du Réseau de Neurones (Q-Network)**



Structure du réseau:

Input(180) → Dense(128, ReLU, Dropout=0.2)
→ Dense(64, ReLU, Dropout=0.2)

→ Dense(32, ReLU)
→ Output(3)

Politique ε-greedy*:
- Probabilité ε : action aléatoire (exploration)
- Probabilité 1-ε : action optimale (exploitation)
- ε décroît de 1.0 à 0.01

- **Mémoire de Répétition (Replay Buffer)**

- **Capacité**: 50,000 transitions
- **Format**: tuples (state, action, reward, next_state, done)
- **Batch size**: 64 échantillons aléatoires pour l'entraînement

- **Hyperparamètres**

Paramètre	Valeur
Discount factor (γ)	0.99
Learning rate	0.005
Batch size	64
Replay buffer size	10,000
ε initial	1.0
ε final	0.05
ε decay	0.995
Target update frequency	30 épisodes

Cette architecture permet à l'agent d'apprendre une stratégie de trading optimale en maximisant les profits à long terme.