1 Rapport de Projet : Implémentation de DQN pour Atari Breakout Mathieu Latournerie

1.0.1 Introduction

Ce rapport présente une implémentation d'un réseau de neurones appelé Deep Q-Network (DQN) pour jouer au jeu Breakout d'Atari. Ce projet s'inspire des travaux pionniers de DeepMind et inclut deux améliorations notables : le Prioritized Experience Replay et le Dueling DQN.

1.0.2 Mise en Œuvre du DQN

Le modèle DQN est un réseau de neurones convolutionnel qui traite les données de pixels provenant de l'environnement du jeu. Le réseau apprend à associer les états du jeu à des valeurs d'action, guidant l'agent vers les actions qui maximisent les récompenses futures.

1.0.2.1 Architecture du Réseau

- **Couche d'Entrée** : Prend en entrée l'état du jeu.
- **Couches Cachées** : Composées de plusieurs couches convolutionnelles.
  - La première couche convolutionnelle comporte 32 filtres, avec une taille de noyau de 8 et un pas de 4.
  - La deuxième couche a 64 filtres, avec une taille de noyau de 4 et un pas de 2.
  - La troisième couche possède également 64 filtres, avec une taille de noyau de 3 et un pas de 1.
- **Couche de Sortie** : Produit des valeurs d'action pour chaque action possible dans le jeu.

1.0.3 Choix des Hyperparamètres

Le choix des hyperparamètres est crucial pour la performance du DQN. Voici les détails des hyperparamètres clés utilisés dans ce projet :

- **Taux d'Apprentissage (epsilon)** : Varie de 1 à 0.1 sur un million de frames, contrôlant la balance entre exploration et exploitation.
- **Facteur de Discount (gamma)** : Fixé à 0.99, ce paramètre détermine l'importance accordée aux récompenses futures.
- **Taille du Buffer de Replay (N)** : Limité à 45 000 en raison des contraintes de mémoire RAM.
- **Taille du Batch** : Fixée à 32, comme indiqué dans l'article de référence.
- **Fréquence de Mise à Jour** : Définit la fréquence à laquelle le réseau cible est mis à jour, un paramètre crucial pour la stabilité de l'apprentissage.

Pour le Prioritized Experience Replay : - **Alpha** : Fixé à 0.6, contrôlant le degré de priorisation dans le replay buffer. - **Beta** : Commence à 0.4 et

augmente linéairement jusqu'à 1 sur 500 000 frames, ajustant l'importance de l'échantillonnage priorisé au fil du temps.

1.0.4 Améliorations

1.0.4.1 Prioritized Experience Replay

- **Description** : Donne la priorité aux expériences importantes basées sur la différence entre les récompenses prédites et réelles, conduisant à un apprentissage plus efficace.
- **Implémentation** : Modification du buffer de replay pour stocker et échantillonner les expériences en fonction de leur priorité.

1.0.4.2 Dueling DQN

- **Description** : Sépare l'estimation de la valeur de l'état et l'avantage de chaque action, permettant au réseau d'apprendre quels états sont précieux sans devoir apprendre l'effet de chaque action.
- **Implémentation** : Modification de l'architecture du réseau pour inclure deux flux – un pour la valeur de l'état et un autre pour l'avantage de l'action.

1.0.5 Analyse des Résultats

1.0.5.1 Approche Expérimentale Les expériences ont été menées pour évaluer les performances de différentes configurations du DQN sur le jeu Atari Breakout, en utilisant le nombre moyen de points obtenus pendant 100 tentatives comme principale métrique de performance.

1.0.5.2 Résultats des Versions sans Priorité

- **Nombre de Frames d'Entraînement** : 4 millions, puis augmenté à 6 millions.
- **Performance** : Un score moyen de 32 a été atteint avec 4 millions de frames d'entraînement. Aucune amélioration significative n'a été observée avec 2 millions de frames supplémentaires.

1.0.5.3 Résultats avec Prioritized Experience Replay

- **Performance** : Le score moyen est tombé à 13, ce qui suggère que l'introduction du Prioritized Experience Replay sans un ajustement adéquat des hyperparamètres peut ne pas être bénéfique.

1.0.5.4 Résultats avec Dueling DQN

- **Apprentissage Accéléré** : Le Dueling DQN a montré une amélioration significative, atteignant un score moyen de 27, indiquant un apprentissage plus rapide et plus efficace.

1.0.5.5 Problèmes et Ajustements

- **Terminal on Life Loss** : Initialement, le "terminal on life loss" n'était pas utilisé, ce qui a empêché le réseau d'apprendre efficacement. Son intégration ultérieure a amélioré les performances d'apprentissage. Cependant cela peut nuire au test car le réseau ne sait pas comment se comporter après la première mort, et le jeu est donc bien plus dur.
- **Ajustement d'Epsilon** : Il y avait un problème dans la gestion de la décroissance d'epsilon, car elle était mise à jour par épisode plutôt que par frame. Cela a entraîné une décroissance trop rapide, affectant l'apprentissage.
- **Implémentation Prioritized Replay** : Le Prioritized Experience Replay a été implémenté en premier. Cependant, en raison de la difficulté à ajuster correctement les hyperparamètres, cette fonctionnalité n'a pas donné les résultats escomptés.

1.0.5.6 Exploration des résultats Dans le dossier `test` se trouve tous les résultats du benchmark sur chaque emodèle que j'ai entraîné pour ce projet, ainsi que des vidéos des agents jouant une partie.

Les reward à l'apprentissage sont aussi disponible dans le dossier `results`, et montre la tendance de chaque réseau, ceci m'a été très utile lors de la phase de développement des différents réseaux.

1.0.5.7 Conclusion Ces expériences soulignent l'importance d'un réglage minutieux des hyperparamètres et de la prise en compte des spécificités de l'environnement d'apprentissage, comme l'importance du "terminal on life loss" et la gestion de la décroissance d'epsilon. Le Dueling DQN a démontré une amélioration significative dans l'apprentissage, suggérant son potentiel pour des performances accrues dans des scénarios similaires.

Les détails et visualisations de ces résultats peuvent être consultés dans le fichier `data_analysis.ipynb`.

1.0.6 Conclusion

J'ai trouvé ce projet très intéressant, et cela m'a permis de bien comprendre l'architecture du DQN ansi que des différentes améliorations que j'ai implémenté. J'ai aussi implémenter le transformer car je voulais en tester les performances, malheureusement le temps d'entraînement était trop long pour que je puisse avoir des résultats sur ma machine.