

Cours DQN – Section 6.1

6.1 – Conclusion et ouverture vers les variantes de DQN

Le Deep Q-Network (DQN) constitue une avancée majeure dans le domaine de l'apprentissage par renforcement profond. Il a démontré qu'un agent pouvait apprendre à partir d'images brutes, uniquement guidé par des signaux de récompense, sans supervision humaine.

Ce qu'il faut retenir du DQN

- Il combine le Q-learning et les réseaux de neurones pour approximer une fonction de valeur $Q(s, a)$.
- Il utilise trois techniques clés pour stabiliser l'apprentissage :
 - **Replay Memory** : casse les corrélations temporelles.
 - **Target Network** : évite les oscillations instables.
 - **Échantillonnage aléatoire** : favorise une généralisation robuste.
- Il est capable de résoudre des environnements très complexes, comme les jeux Atari, à partir de simples images.

Mais DQN n'est pas parfait

Malgré ses succès, le DQN souffre encore de plusieurs limitations :

- Il peut surestimer les valeurs Q , ce qui biaise la politique d'action.
- Il est sensible au choix des hyperparamètres.
- L'échantillonnage uniforme ne distingue pas les expériences plus utiles que d'autres.

Vers des variantes plus robustes

Plusieurs améliorations ont été proposées pour pallier ces faiblesses :

- **Double DQN** :
 - Corrige la tendance à la surestimation des valeurs Q .
 - Utilise deux réseaux : un pour choisir l'action, l'autre pour évaluer sa valeur.
- **Dueling DQN** :
 - Décompose $Q(s, a)$ en deux parties :

$$Q(s, a) = V(s) + A(s, a)$$

- Sépare la valeur d'un état $V(s)$ de l'avantage d'une action $A(s, a)$.

- Améliore l'apprentissage dans des situations où certaines actions n'ont que peu d'impact.
- **Prioritized Experience Replay** :
 - Favorise les expériences avec un fort signal d'apprentissage (forte erreur TD).
 - Améliore la rapidité de convergence.
- **Rainbow DQN** :
 - Combine toutes les améliorations précédentes dans un seul algorithme.
 - Représente l'état de l'art pour l'apprentissage profond basé sur Q.

Conclusion finale

Le DQN n'est pas un aboutissement, mais un socle sur lequel repose toute une famille de méthodes plus puissantes. Comprendre son fonctionnement est essentiel pour aborder des approches modernes en intelligence artificielle décisionnelle.

Dans les prochains chapitres, nous étudierons :

- Comment implémenter un DQN complet avec Python et PyTorch.
- Les améliorations comme Double DQN, Dueling DQN et Prioritized Replay.
- Des études de cas concrètes pour mettre en œuvre ces algorithmes.