Deep Q-Network pour le Problème CartPole

Théorie, Modélisation et Fondements Mathématiques

EL HAMRAOUI Ismail
Master Ingénierie des Systèmes Complexes
École d'Ingénieurs du Littoral Côte d'Opale

November 21, 2025

Sommaire

Le Problème CartPole

Définition du problème

- Environnement classique en apprentissage par renforcement
- Objectif : Équilibrer un mât sur un chariot mobile
- Contrôle : Appliquer des forces gauche/droite
- Échec : Si le mât dépasse un angle critique ou le chariot sort des limites

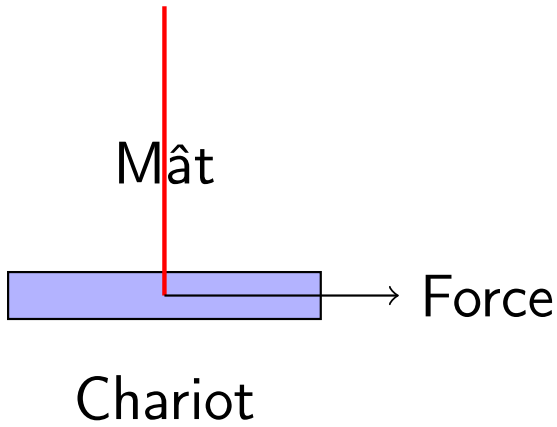


Figure: Représentation schématique de CartPole

Modélisation Dynamique

Variables et paramètres du modèle

x : Position du chariot

$\dot{x}$: Vitesse du chariot

θ : Angle du mât

$\dot{\theta}$: Vitesse angulaire

m_c : Masse du chariot

m_p : Masse du pendule

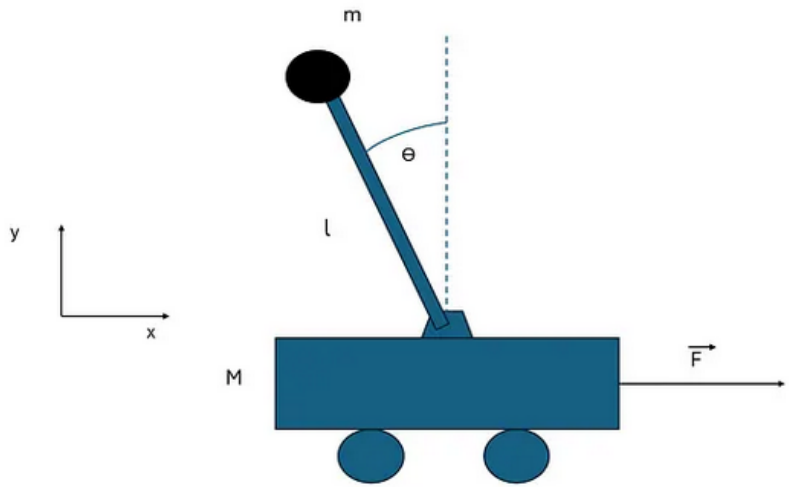
l : Longueur du mât

F : Force appliquée

Équations du mouvement

$$\ddot{x} = \frac{F + m_p l \dot{\theta}^2 \sin\theta}{m_c + m_p}$$

$$\ddot{\theta} = \frac{g \sin\theta - \cos\theta \cdot \ddot{x}}{l}$$



Formalisation MDP

Processus de Décision Markovien (MDP)

- **Espace d'états** : $\mathcal{S} = \mathbb{R}^4$ (position, vitesse, angle, vitesse angulaire)
- **Espace d'actions** : $\mathcal{A} = \{\text{gauche}, \text{droite}\}$
- **Fonction de transition** : $P(s'|s, a)$ définie par la physique
- **Fonction de récompense** : $r(s, a) = +1$ tant que le mât reste debout

Fonction de valeur

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s, a_0 = a\right]$$

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s\right]$$

Architecture du Réseau de Neurones

Structure du DQN

- **Entrée** : 4 neurones (état du système)
- **Couches cachées** : 2 couches de 64 neurones avec ReLU
- **Sortie** : 2 neurones (valeurs Q pour chaque action)

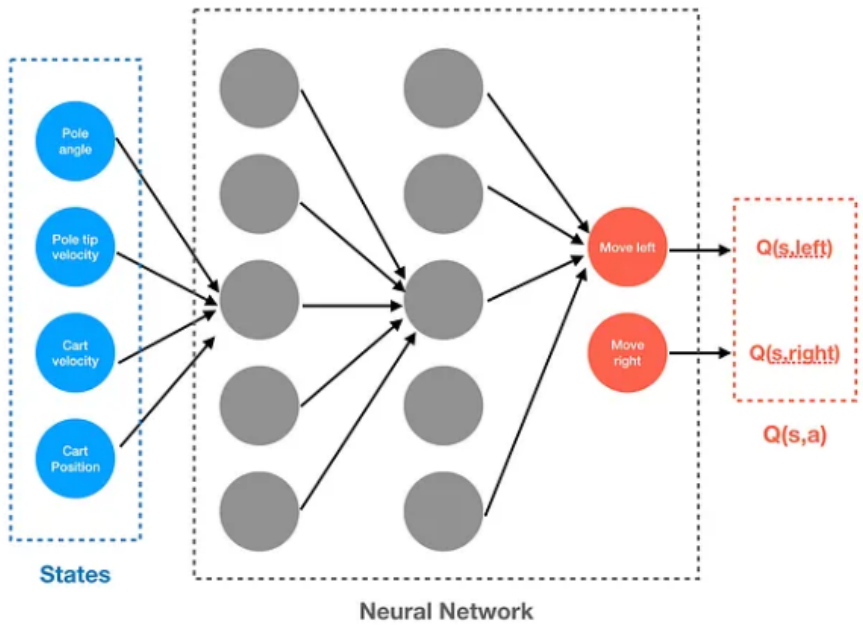


Figure: Architecture du réseau DQN

Fonction de Perte et Optimisation

Fonction de perte de Bellman

$$L(\theta) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(r + \gamma \max_{a'} Q_{\theta^-}(s', a') - Q_\theta(s, a)\right)^2\right]$$

Composantes clés

- θ : Paramètres du réseau principal (policy network)
- θ^- : Paramètres du réseau cible (target network)
- $\gamma = 0.99$: Facteur d'actualisation
- $\mathcal{D}$: Mémoire de replay

Experience Replay

Principe

- Stockage des transitions $(s, a, r, s', done)$
- Taille de la mémoire : 10,000 transitions
- Échantillonnage aléatoire par lots de 64

Avantages

- Réduction de la corrélation entre échantillons
- Réutilisation des expériences passées
- Amélioration de la stabilité de l'apprentissage

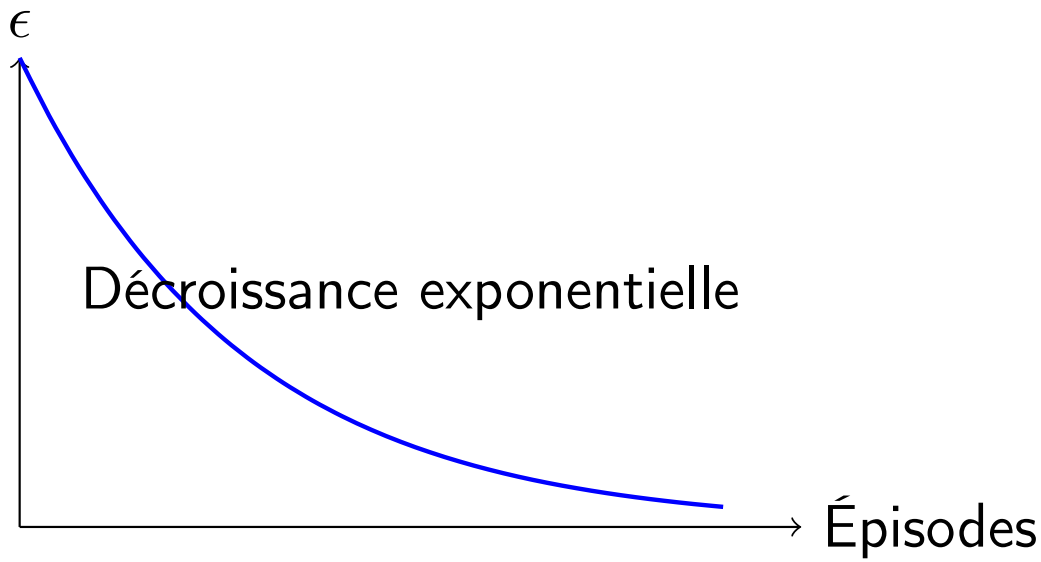
Target Network et ϵ-greedy

Target Network

- Copie périodique des poids
- Mise à jour toutes les 10 épisodes
- Stabilise les cibles d'apprentissage

Stratégie ϵ-greedy

- ϵ initial : 1.0
- Décroissance : $\epsilon \leftarrow \epsilon \times 0.995$
- ϵ final : 0.01
- Équilibre exploration/exploitation



Boucle d'Apprentissage

Algorithme d'entraînement

1. Initialisation des réseaux et de la mémoire
2. Pour chaque épisode :
    1. Réinitialisation de l'environnement
    2. Pour chaque pas de temps :
        1. Sélection d'action avec ϵ-greedy
        2. Exécution et observation (s, a, r, s')
        3. Stockage dans la mémoire
        4. Entraînement sur lot aléatoire
    3. Mise à jour de ϵ et du réseau cible

Hyperparamètres

Paramètre	Valeur	Description
γ	0.99	Facteur d'actualisation
α	0.001	Taux d'apprentissage
Batch Size	64	Taille des lots
Memory Size	10,000	Capacité mémoire
ϵ_{start}	1.0	Exploration initiale
ϵ_{end}	0.01	Exploration finale
ϵ_{decay}	0.995	Taux de décroissance
Target Update	10	Fréquence mise à jour

Table: Hyperparamètres du DQN

Métriques de Performance

Indicateurs de succès

- Récompense cumulée par épisode
- Nombre de pas avant échec
- Évolution de ϵ
- Performance du réseau cible

Critères de convergence

- Récompense maximale atteinte (200 pas)
- Stabilité des performances
- Faible exploration ($\epsilon \approx 0.01$)

Conclusion

Points clés

- DQN combine Q-learning et réseaux de neurones profonds
- Experience replay et target network stabilisent l'apprentissage
- ϵ-greedy gère exploration/exploitation
- Application réussie au problème CartPole

Extensions possibles

- Double DQN
- Dueling DQN
- Priorized Experience Replay
- Distributional DQN

Composants du Système

Éléments clés de l'implémentation

- **Environnement CartPole** : Simulation physique discrétisée
- **Policy Network** : Réseau principal pour la sélection d'actions
- **Target Network** : Réseau stabilisateur pour les cibles
- **Replay Memory** : Mémoire d'expériences passées
- **Optimiseur Adam** : Descente de gradient stochastique

Équations de mise à jour

$$\theta \leftarrow \theta - \alpha \nabla_\theta L(\theta)$$

$$\theta^- \leftarrow \theta \quad \text{(périodiquement)}$$