Introduction à PPO et aux Réseaux de Neurones

1 PPO et les réseaux de neurones : L'architecture

PPO utilise deux réseaux de neurones principaux :

1. **Le réseau de la politique (π_θ)** :
   - C'est lui qui décide **quelle action prendre** dans chaque situation.
   - Il donne une probabilité pour chaque action possible (exemple : "80 % de chances d'aller à gauche, 20 % d'aller à droite").
2. **Le réseau de la valeur (V_ϕ)** :
   - Il prédit **la qualité d'une situation**.
   - En langage technique, il estime **combien de récompenses l'agent peut espérer gagner** à partir d'un état donné.

Ces deux réseaux sont entraînés grâce au **deep learning**, en ajustant leurs paramètres (leurs poids) pour mieux remplir leur rôle.

2 Pourquoi PPO utilise des réseaux de neurones ?

Dans des environnements simples, on pourrait utiliser une "table" pour stocker toutes les actions possibles et leur qualité (comme dans le Q-learning classique). Mais pour des problèmes complexes, comme jouer à des jeux vidéo ou contrôler des robots, ce n'est plus possible :

- **Problème 1 : Trop de combinaisons possibles** Par exemple, dans un jeu Atari, une seule image d'écran peut contenir des **millions de pixels**. Chaque combinaison de pixels représente un état différent.
- **Problème 2 : Besoin de généraliser** Si le robot rencontre un état qu'il n'a jamais vu, il doit pouvoir **faire des déductions** grâce à ce qu'il a appris.

Les **réseaux de neurones** résolvent ces deux problèmes, car ils peuvent :

- **Comprendre des motifs complexes**, comme reconnaître des objets dans une image ou détecter un mouvement dans un jeu.
- **Généraliser** à partir de ce qu'ils ont vu pour prendre des décisions dans des situations nouvelles.

3 Comment PPO utilise le deep learning ?

Étape 1 : Prendre une décision avec le réseau de politique

Le réseau de politique (π_θ) utilise les **caractéristiques de l'état** pour prédire **quelle action prendre**.

Exemple dans CartPole :

- **Entrée du réseau :**
  - Position du chariot.
  - Angle du bâton.
  - Vitesse du chariot.
  - Vitesse angulaire du bâton.
- **Sortie du réseau :**
  - Probabilités pour chaque action :
    - Aller à gauche : 60 %.
    - Aller à droite : 40 %.

Ce processus ressemble à un réseau de classification en deep learning, mais ici, on prédit des probabilités d'actions au lieu de catégories fixes.

Étape 2 : Évaluer la qualité d'un état avec le réseau de valeur

Le réseau de valeur (V_ϕ) prend l'état actuel comme entrée et prédit une **valeur numérique unique** :

"Combien de récompenses futures puis-je espérer depuis cet état ?"

Exemple : Si le bâton est presque en équilibre, le réseau de valeur pourrait dire : *"Cet état vaut 50 points, car je peux probablement garder le bâton en équilibre encore longtemps."*

Étape 3 : Entraîner les réseaux avec les données collectées

Quand l'agent joue dans l'environnement, il collecte des données sur ce qui fonctionne ou non :

- Les états qu'il a rencontrés.
- Les actions qu'il a prises.
- Les récompenses qu'il a reçues.

Ces données sont utilisées pour **entraîner les deux réseaux de PPO** grâce à l'algorithme de rétropropagation (backpropagation) utilisé en deep learning.

4 Le rôle clé de l'entraînement supervisé en PPO

PPO entraîne ses réseaux comme suit :

1. **Pour le réseau de politique (π_θ) :**
   - Il maximise la probabilité des actions qui ont donné de bonnes récompenses (grâce à la fonction objectif avec clipping).
   - Si une action était bénéfique, le réseau apprend à **augmenter sa probabilité**.
   - Si une action était mauvaise, le réseau apprend à **réduire sa probabilité**.
2. **Pour le réseau de valeur (V_ϕ) :**
   - Il minimise l'erreur entre la valeur prédite ($V_\phi(s_t)$) et la valeur cible (V_{target}).
   - Cela permet au réseau de mieux évaluer chaque situation.

Techniquement : Ces ajustements se font grâce à des **optimisateurs** comme Adam, qui modifient les poids du réseau pour réduire l'erreur ou maximiser l'objectif.

5 Pourquoi PPO est stable grâce au deep learning ?

Le deep learning est souvent instable quand on change les poids d'un réseau trop rapidement. PPO utilise plusieurs techniques pour stabiliser cet entraînement :

1. **Clipping :** PPO limite les mises à jour trop brutales dans le réseau de politique, ce qui évite que l'agent devienne instable.
2. **Replay des données :** PPO utilise les expériences collectées pour entraîner les réseaux plusieurs fois, augmentant ainsi leur efficacité.
3. **Deux réseaux distincts :** Le réseau de politique et le réseau de valeur sont entraînés séparément, ce qui améliore la stabilité.

6 Une analogie pour mieux comprendre

Imagine que tu es un **robot chauffeur** et que PPO est ton coach :

- **Le réseau de politique (π_θ)** : C'est ton **intuition** : *"Que dois-je faire dans cette situation?"* Le coach t'aide à affiner cette intuition pour que tu prennes de meilleures décisions.
- **Le réseau de valeur (V_ϕ)** : C'est ton **sens du jugement** : *"Est-ce que je suis dans une bonne situation ou pas?"* Le coach améliore ce sens pour que tu puisses mieux planifier tes actions.

7 Pourquoi PPO et deep learning sont inséparables ?

PPO ne pourrait pas fonctionner sans les réseaux de neurones, car :

- Les réseaux permettent de **représenter des politiques complexes** même dans des environnements compliqués (comme des jeux ou des robots).
- Le deep learning permet à PPO de **généraliser** : l'agent apprend des motifs et peut les appliquer à des situations nouvelles.
- Les réseaux de neurones rendent PPO **scalable**, c'est-à-dire qu'il peut fonctionner dans des problèmes avec des millions de possibilités.

Résumé final : Comment PPO utilise le deep learning

1. PPO utilise **deux réseaux de neurones** :
   - Un pour prendre des décisions (réseau de politique).
   - Un pour juger la qualité des états (réseau de valeur).
2. Ces réseaux sont entraînés avec des techniques de **deep learning**, comme :
   - La rétropropagation (backpropagation).
   - Des optimisateurs comme Adam.
3. PPO stabilise l'entraînement grâce à des innovations comme le **clipping** et les pénalités d'entropie.
4. **Le résultat :** PPO permet à un agent d'apprendre à prendre des décisions complexes dans des environnements difficiles, en combinant l'apprentissage par renforcement et le deep learning.