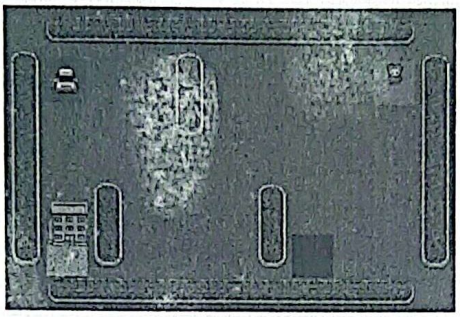
Université Mohamed Premier Oujda
École Nationale de l'Intelligence Artificielle et du Digital Berkane
Année universitaire : 2024 / 2025

Filière : IA
Prof : Mohamed Khalifa BOUTAHIR

APPRENTISSAGE PROFOND POUR LES JEUX – TP 4

Objectif du TP :

L'objectif de ce TP est de familiariser les étudiants avec l'implémentation de l'algorithme **Proximal Policy Optimization (PPO)**. À travers ce TP, les étudiants apprendront à construire une **table de politiques**, à **mettre à jour la valeur des états** et à **entraîner un agent** à résoudre le problème de transport de passagers dans l'environnement **Taxi-v3**.



Exercice 1 : Initialisation de l'environnement et des structures de données

- Initialiser l'environnement **Taxi-v3** et afficher **le nombre d'états et d'actions**.
- Créer une **table de politique** où chaque état a une probabilité égale pour chaque action.
- Créer une **table de valeurs** initialisée à zéro.
- Ajouter un affichage des premières lignes de policy_table et value_table.

```
import gymnasium as gym
import numpy as np

# Initialisation de l'environnement
env = gym.make("Taxi-v3")

# Nombre d'états et d'actions
state_size = env.observation_space.n
action_size = env.action_space.n

.....
```

Exercice 2 : Exploration et collecte d'épisodes

- Faire exécuter un **agent aléatoire** dans l'environnement **pendant 20 épisodes**.
- Afficher les **actions exécutées et les récompenses obtenues**.

```
state, _ = env.reset()
for t in range(20):
  .....
```

Exercice 3 : Mise à jour de la politique avec PPO

L'algorithme PPO optimise la politique $\pi\theta$ en maximisant la fonction suivante avec un terme de clipping pour éviter des mises à jour trop brutales :

$$L(\theta) = \mathbb{E}\left[\min\left(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t\right)\right]$$

Étapes clés de la mise à jour PPO

1. **Calcul des récompenses cumulées** Rt (discounted rewards).
2. **Calcul de l'avantage** At=Rt - V(st).
3. **Mise à jour de la politique avec clipping** : Ajuster $\pi\theta$ en respectant les contraintes de PPO.
4. **Mise à jour de la fonction de valeur** V(s).

- Calculer **les récompenses cumulées** (discounted rewards).
- Mettre à jour la **fonction de valeur** pour chaque état visité.
- Mettre à jour **la politique avec PPO** (en respectant le clip).
- Ajouter une mise à jour de value_table[state] avec une learning rate.

```
gamma = 0.99
lr_policy = 0.1
clip_epsilon = 0.2

episode_states = [state1, state2, ...]  # Liste des états
episode_actions = [a1, a2, ...]  # Liste des actions
episode_rewards = [r1, r2, ...]  # Liste des récompenses
.....
```

Exercice 4 : Évaluation de l'agent après entraînement

- Tester l'agent entraîné **pendant 20 épisodes**.
- Comparer **les performances** avant et après entraînement.

```
num_eval_episodes = 20
total_rewards = []

for ep in range(num_eval_episodes):
  state, _ = env.reset()
  total_reward = 0

  .....
```