With the increasing advancement of deep learning models and the significant results from integrating these models with reinforcement learning applications, the fourth exercise will focus on examining these algorithms and models. Solving reinforcement learning problems using neural networks has a long history, and with the development of computational hardware over the past two decades, the speed of deep model development for reinforcement learning problems has increased significantly. Using neural networks allows us to solve the problem using an end-to-end model. Considering this, acquiring the skills to work with deep learning models and solve reinforcement learning problems using these models is an essential skill in the field of reinforcement learning. This exercise serves as an introduction to these problems, and naturally, gaining more skills in this area requires further study and practice.

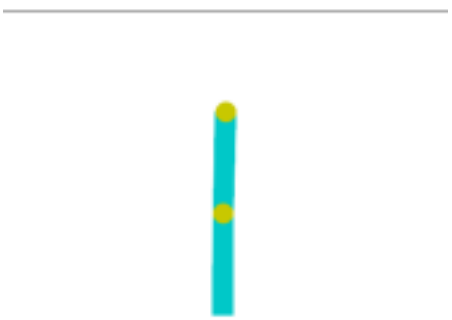
Analytical questions

1. Explain one advantage and one disadvantage of Deep RL algorithms.
2. Mention one advantage of the Deep Q-Learning algorithm over the Q-Learning algorithm.
3. Name one reason for using an experience buffer.

The environment used in the implementation question.

In this exercise, you will work with the Gymnasium library. Gymnasium is an API previously known as Gym, which serves as a standard for reinforcement learning and provides a diverse set of different environments. A simple explanation of how to use its environments is provided at this link. In this exercise, you will become

familiar with and work with the Acrobot environment. This system consists of two links that are connected linearly to form a chain, with one end of the chain fixed. The connection between the two links is actuated. The objective is to apply torques to the actuated joint to swing the free end of the linear chain above a certain height, starting from an initial state where it hangs down. See the illustration bellow:



Two blue links are connected by two green joints. The joint between the two links is actuated. The goal is to rotate the free end of the outer link to reach the target height (the black horizontal line above the system) by applying torque to the actuator.

To use this environment, considering the large volume of computations, it is recommended to use Google Colab. You can install and import the required libraries using the following code snippet.

```
!pip install gymnasium[classic-control]
import gymnasium as gym
env = gym.make('Acrobot-v1')
observation, info = env.reset()
for _ in range(1000):
```

```
    action = env.action_space.sample()

    observation, reward, terminated, truncated, info = env.step(action)

    if terminated or truncated:

        observation, info = env.reset()

env.close()
```

To gain a better understanding of how this environment works, you can refer to the GitHub repository of the library itself.

Part 1 – Familiarization with the problem environment

Based on the provided explanations and link, run the environment in Acrobot-v1mode.

1. Describe the actions, states, and rewards that the agent receives from the environment. For example, explain which attributes of the environment are represented by the agent's state.
2. Research the continuity and discreteness of the states and actions in this environment.

Part 2 – Implementation of the Deep RL algorithm based on...

The goal in this section is to implement the Deep Q-Learning algorithm [1].

1. Implement the DQN algorithm without using the `baselines_stable` library. For implementation, it is preferable to use the PyTorch library. Using

the TensorFlow library is also not prohibited. Your agent should learn the Acrobot task from the Gymnasium environment.

2. After the agent learns, include a graph of the rewards obtained during learning by the agent over several runs in your report. The graph should include a 95% confidence interval.
3. Provide a video (render) from several test episodes of the agent after learning. For rendering, it is recommended to save the model weights to avoid issues with Google Colab and perform this locally on your computer.
4. Present the parameters you used in your report in a table.

Part 3 – Improving Value-based Deep RL Methods

As a grading option, you can choose one of the following two tasks and implement it:

- Implement the DQN algorithm with Prioritized Experience Replay. You can read about this algorithm in the paper [2]. Also, explain how adding this feature improves the DQN algorithm compared to the version without this feature.

- Implement the Deep Q-Learning algorithm with Dueling Networks. You can find explanations about this algorithm in the paper [3]. Additionally, explain what advantages this algorithm has over the classic DQN algorithm.

In both cases, you should solve the problem from section two and compare the results with those from section two. For comparison, you can use appropriate plots. Is the difference significant? You can use statistical tests to support your claim.

References

1. V. Mnih et al., “Playing Atari with deep reinforcement learning,” arXiv [cs.LG], 2013
2. T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” arXiv [cs.LG], 2015.
3. Wang, Ziyu, et al. "Dueling network architectures for deep reinforcement learning." International conference on machine learning. PMLR, 2016.