DA6400 : Reinforcement Learning

Programming Assignment #1

Deadline: 30^{th} March, 2025 11:59 PM

1 Environments

In this programming task, you'll utilize the following Gymnasium environments for training and evaluating your policies. The links associated with the environments contain descriptions of each environment. Please use the exact version of the environment as specified:

- CartPole-v1: A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The pendulum is placed upright on the cart and the goal is to balance the pole by applying forces in the left and right direction on the cart.
- MountainCar-v0: The Mountain Car MDP is a deterministic MDP that consists of a car placed stochastically at the bottom of a sinusoidal valley, with the only possible actions being the accelerations that can be applied to the car in either direction. The goal of the MDP is to strategically accelerate the car to reach the goal state on top of the right hill. There are two versions of the mountain car domain in gymnasium: one with *discrete actions* and one with *continuous*. This version is the one with discrete actions.
- MiniGrid-Dynamic-Obstacles-5x5-v0: **[Bonus +2]** This environment is an empty room with moving obstacles. The goal of the agent is to reach the green goal square without colliding with any obstacle. A large penalty is subtracted if the agent collides with an obstacle and the episode finishes. This environment is useful to test Dynamic Obstacle Avoidance for mobile robots with Reinforcement Learning in Partial Observability.

2 Algorithms

You are tasked with training each of the following algorithms and assessing their comparative performance.

- **SARSA** $\rightarrow$ Use ϵ**-greedy exploration**
- **Q-Learning** $\rightarrow$ Use **Softmax exploration**

Reward shaping is allowed, but discouraged for trivial use cases if you can solve it without it.

3 Instructions

We expect a comprehensive report that involves the following details:

- Snippets of the important parts of the code.
- Result Plots — For each environment, there should be plots comparing SARSA and Q-Learning.
- Report top 3 best hyperparams.
  - With comparative justification (Hint: use wandb or mlflow or similar.)
- Inferences and conjectures from all your experiments and results.
- Github link to the code. Include your python `requirements.txt` and/or conda yaml file and steps to recreate your experiments.

You are required to compare each algorithm with it's own variant and not with the other algorithm. Please adhere strictly to the following instructions.

- (Recommended) Use $\gamma = 0.99$ for all experiments
- Tune the hyper-parameter to minimize the regret in all experiments.
- To account for stochasticity, use the average of 5 random seeds for each experiment/plot
- Plot the episodic return versus episodic number for every experiment
- The plots should consist the mean and variance across the 5 runs/seeds (Sample plot 3)

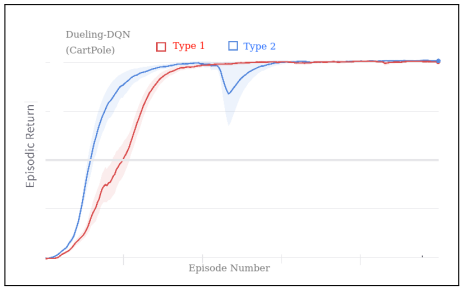

Figure 1: Sample plot of Dueling-DQN in CartPole. We expect 4 such plots minimum, one for each comparison

- Upload the corresponding code to a private repository in Github and attach the link in the report
- **Please strictly follow the academic code of conduct. Plagiarism will be penalized**

We expect one submission per group of 2 members.