AUTONOMOUS CAR RACING WITH DEEP REINFORCEMENT LEARNING

TEAM INFORMATION: Sonal Prasad - 017475432, Anoushka Sawant - 015154074, Sravani Neelapala - 016801486

DATA: Gymnasium

We are considering two environments (first inbuilt and second external)::

The 2D car racing environment is part of the Box2D environments. It's a top-down racing environment, with a random track in every episode. In continuous, there are three actions, and in discrete there are five actions. The observation has a top-down 96x96 RGB image of the car and race track.

https://gymnasium.farama.org/environments/box2d/car_racing

The 3D car racing environment uses Pybullet to simulate real-world physics. It includes observations such as pose, velocity, and rgb-camera, which are obtained by sensors. Similar to the 2D environment, it also has three actions in the continuous space. Vehicle and track configuration are customizable and multi-agent racing is possible with PettingZoo API.

https://github.com/axelbr/racecar_gym/

PROJECT IDEA:

Simple autonomous driving is a useful application of reinforcement learning. In our project, we will focus on autonomously navigating a race track. Potential actions include accelerating, breaking, and turning left and right. Rewards are granted based on proximity to the finish line. Our approach is to use Deep Q learning to predict Q values of different actions. If time permits, we will also explore Proximity Policy Optimization (PPO) or other algorithms we have encountered in research papers, such as DQN (Deep Q-Network), DDQN(Double Deep Q-Network), and LSTM with SAC (Long Short-Term Memory with Soft Actor-Critic).

BRIEF DESCRIPTION:

i) Set up environments: First, we will get the CarRacing-v0 (2D) and racecar_gym (3D) environments set up and these will be our training and testing environments for our agents. Racecar-gym also requires installing external dependencies, so we will only decide to use it after testing the instructions in the github.

ii) Implement baseline: The first thing we will do is implement a DQN agent discretized in actions. This will give us a baseline to compare how much better we are performing using more advanced techniques.

iii) Advanced method: Next, we will develop a DDQN, PPO, SAC with LSTM agent based on our literature review to provide continuous controls such as steering and braking to enable smoother and more realistic driving.

iv) Reward shaping: We will develop reward functions that encourage positive driving habits for example, staying on the road, constant speed, and economical lap running.

v) Evaluation: We will compare the performance of the baseline agent with the other algorithms across different tracks and will come up with benchmark results. We will measure results using metrics like average reward, maximum speed and driving speed.

vi) Report & present: Finally, we will present our findings, note the enhancements, and prepare both a written report and a class presentation.

REFERENCES:

A list of 2 - 3 related references that each of you will read

https://github.com/axelbr/racecar_gym/
https://gymnasium.farama.org/environments/box2d/car_racing/
https://ieeexplore-ieee-org.libaccess.sjlibrary.org/stamp/stamp.jsp?tp=&arnumber=8946332
https://ieeexplore-ieee-org.libaccess.sjlibrary.org/document/9660123

TENTATIVE ROLES OF EACH TEAM MEMBER:

i) All teammates will contribute toward setup, implementation, and evaluation. We may experiment with 2-3 different algorithms individually as mentioned in the Project Idea section above. All teammates will contribute toward the report writing.

ii) We will use a Scrum Board to assign and keep track of tasks.

HPC Access:
Please grant us access if possible :)