Cart-Pole with Sensor Noise

Supervising TA: **Xiaocan Li**

Preface

This document gives you a detailed description of the project proposal entitled **Cart-Pole with Sensor Noise**. The document takes you through the problem setting, how to develop the environment, what to investigate, and what to deliver.

1 Introduction

In this project, you will tackle the classic Cart-Pole challenge that you've seen in the assignments and course lectures. However, to simulate a more realistic and challenging scenario, you will incorporate sensor noise into the observations. This modification simulates real-world phenomena like sensor inaccuracies or failures, challenging the agent to learn in *non-ideal* conditions.

The goal of the Cart-Pole game is to balance a pole, hinged on a cart, by moving the cart left or right to prevent the pole from falling over. The standard observations include the position of the cart, the velocity of the cart, the angle of the pole, and the angular velocity of the pole.

Your task will be to develop and compare two types of RL agents:

1. one using a traditional non-deep learning algorithm, and
2. another using a deep RL algorithm.

This will allow you to evaluate how different approaches handle environmental uncertainties due to sensor noise. The purpose of the project is not only to test your understanding of reinforcement learning principles and algorithms but also to enhance your skills in dealing with real-world problems where data may be imperfect or incomplete. Most importantly, you'll learn how to do research in the process.

2 Project Milestones

There are four main milestones for this project. The initial ones are quite straightforward being rather of an assignment form than a project. The final milestone needs however more effort. In the sequel, each milestone has been illustrated briefly.

2.1 Environment Setup

Objective You need to set up the basic CartPole-v1 environment using Gymnasium.

Tasks This can be readily achieved by following steps:

1. Install necessary libraries such as gymnasium and numpy.
2. Initialize the CartPole-v1 environment and familiarize yourself with its dynamics and observation space.
3. Implement a random policy to test the environment, ensuring it runs correctly on your setup.

2.2 Introduce Sensor Noise via Gymnasium Observation Wrapper

Objective You need to modify the CartPole-v1 environment to introduce noise in its sensors, simulating inaccuracies in position, velocity, pole angle, and angular velocity.

Tasks This can be addressed by following steps:

1. Create a Gymnasium Observation Wrapper for the CartPole-v1 environment that adds Gaussian noise to each component of the state observation. The wrapper allows you to add extra operations to the existing environment while keeping the existing environment intact. Information about the Gymnasium Observation Wrapper can be found on the game page at Gymnasium documentation:

   `https://gymnasium.farama.org/api/wrappers/observation_wrappers`

   Requirement: The noise level should be configurable so that its impact can be studied systematically.

   Hints:

   a) The code can be as simple as 1 or 2 lines.

   b) How do you measure / tune the noise level?

2. Verify the behavior of the environment with noise by visualizing how the noisy observations differ from the true state.

2.3 Implement a Non-Deep RL Agent

Objective You need to implement a traditional RL algorithm, such as Q-Learning or SARSA, to solve the noisy CartPole-v1 environment.

Tasks This can be addressed by following steps:

1. Develop or adapt an existing non-deep RL algorithm to learn policies based on noisy observations.
2. Experiment with different parameters and settings to optimize performance.
3. Evaluate the agent's performance in terms of stability and ability to balance the pole despite observational noise.

Requirements You are **NOT** allowed to use off-the-shelf RL libraries, such as RLLib and Stable Baselines. You need to implement the RL algorithm on your own.

2.4 Implement a Deep RL Agent

Objective You need to implement a deep RL algorithm, such as DQN, to tackle the same task.

Tasks This can be addressed by following steps:

1. Set up a deep learning framework (e.g., TensorFlow or PyTorch) and implement DQN.
2. Design and train a neural network to handle noisy observations and learn effective control policies.
3. Compare the performance of the deep RL agent with the non-deep RL agent, analyzing strengths and weaknesses.

Requirements You are **NOT** allowed to use off-the-shelf RL libraries, such as RLLib and Stable Baselines. You need to implement the RL algorithm on your own.

3 Outcomes and Deliverables

There are several deliverables that you should submit by the end of this project. Some of these deliverables are semantic, i.e., you should answer some research questions using your final implementation. Some others are documentation, i.e., report, presentation, and source codes. These items are specified below:

3.1 Research Questions

Impact of Noise Investigate how the level of sensor noise impacts the learning process and the performance of both non-deep and deep RL agents.

Robustness of Algorithms Find out which RL algorithm demonstrates greater robustness to sensor noise and explain why.

Parameter Sensitivity Investigate how different parameters (like learning rate, discount factor, and exploration strategy) affect the agent's ability to learn in noisy environments.

Comparison of Deep vs. Non-Deep RL Explain in what ways the use of deep learning influences the agent's performance and stability in handling noisy, imperfect observations.

3.2 Submission Requirements

Like all projects, you need to deliver the following documentation for your project by the final deadline.

Source Code Submit all source code used for implementing the project.

Report A detailed report documenting your methods, experiments, results, and answers to the research questions. Include graphs and tables as necessary to support your conclusions. *If the results are not satisfactory, still provide analysis and your next steps based on your analysis. Don't worry, it happens a lot during research! That is also where the word RE-search comes from!*

Poster Prepare a poster summarizing your findings and challenges faced during the project.