SafeRL

Roopak Srinivasan
(rs2386)

Yogesh Eshwar Patil
(yp445)

1 Introduction

Constrained Reinforcement Learning (RL) is a subfield of RL that focuses on learning policies that satisfy constraints while maximizing the reward. In many real-world applications, it is important to ensure that the agent does not violate certain constraints. One such important constraint is safety, where the agent must avoid unsafe actions that could lead to catastrophic failures.

In this project, we look at a Lunar Lander agent from Gymnasium environment [1] that must land safely on the moon without crashing. We train two policies, base and safe models, using Proximal Policy Optimization (PPO) [2]. The base model is trained without any constraints, while the safe model is trained with constraints that enforce the agent to land within a certain parabolic curve trajectory. We evaluate the performance of both models in a new unknown world environment with high wind speeds and compare the performance of the base and safe models against the expert model of the unknown world environment. We analyze the trajectories of the Lunar Lander to understand how the policies differ in terms of safety.

1.1 Lunar Lander Environment

The Lunar Lander environment is a two-dimensional spacecraft that must land on the moon's surface within a certain distance of the landing pad. The agent must control the Lunar Lander's engines to adjust its position and velocity to land safely. The agent receives a reward based on its landing performance, with higher rewards for safer landings.

The environment

Action Space	Discrete(4)
Observation Shape	(8,)
Observation High	[1.5 1.5 5. 5. 3.14 5. 1. 1.]
Observation Low	[-1.5 -1.5 -5. -5. -3.14 -5. -0. -0.]
Import	`gym.make("LunarLander-v2")`

Figure 1: Lunar Lander Environment

2 Problem

The problem we are trying to solve is to train a policy that can land the Lunar Lander safely on the moon without crashing. The Lunar Lander is a two-dimensional spacecraft that must land on the moon's surface within a certain distance of the landing pad. The goal is to maximize the reward while satisfying the constraints.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

2.1 The Unknown World Environment

The unknown world environment is a new environment

3 Approach

3.1 Defining the Environments

We first define three environments: base, safe, and unknown world. The base environment is the standard Lunar Lander environment where the agent must land the Lunar Lander without any constraints. The safe environment is the Lunar Lander environment with constraints that enforce the agent to land within a certain parabolic curve trajectory. The unknown world environment is a new environment with high wind speeds that the agent has not seen during training.

3.2 Training the Policies

Algorithm 1 Step Function for Lunar Lander Environment

procedure STEP($action$)
    $observation, reward, done, \_, info \leftarrow env.step(action)$
    $x, y \leftarrow observation[0], observation[1]$
    $path.append((x, y))$
    $y_{desired} \leftarrow a \cdot x^2 + b \cdot x + c$
    if $y < y_{desired}$ **then**
        $deviation \leftarrow y_{desired} - y$
        $penalty \leftarrow -\sqrt{deviation} \cdot altitude\_scaling\_factor$
        $shaped\_reward \leftarrow \max(max\_penalty, penalty)$
        $reward \leftarrow reward + shaped\_reward$
        if $action \in \{1, 2, 3\}$ **then**
            $reward \leftarrow reward + reduced\_engine\_penalty + 0.3$
        end if
    end if
    if $done$ **then**
        $info['safety'] \leftarrow score\_landing\_path(x, y)$
    end if
    if $done$ and $debug$ **then**
        $SafePara.plot\_landing\_path(path, curving, x, y, info['safety'])$
    end if
    return $observation, reward, done, \_, info$
end procedure

4 Results

References

[1] Lunar Lander - Gym Documentation.

[2] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.