Practical AI Project: Application of Reinforcement Learning Algorithms

João Lopes - 37133 Rúben Pinto - 40115

June 5, 2023

Abstract

This report presents the application of Reinforcement Learning (RL) algorithms using Stable-Baselines3 (SB3) in predominantly discrete environments. The project focuses on training agents in three different environments: Acrobot (with continuous observation space), an Atari game (specifically, DemonAttack), and a customized environment.

Using SB3 algorithms and Gymnasium, the agents were trained to learn optimal policies in these discrete environments. The main concepts of RL and the selected SB3 algorithms are introduced and justified. The report provides a detailed explanation of the experimental setup, including the tools, libraries, and configurations used.

This project demonstrates the successful application of RL algorithms in discrete environments using SB3. The findings highlight its potential in various real-world applications.

Contents

Introduction

Reinforcement Learning (RL) is an area of Machine Learning (ML) that focuses on training intelligent agents to make sequential decisions in order to maximize cumulative rewards. RL has gained significant attention in recent years due to its potential for solving complex problems and its applications in various domains, such as game theory, control theory, operations research, information theory, simulation-based optimization, multi-agent systems, swarm intelligence, and statistics.

The goal of this project is to explore the application of RL algorithms in training agents in predominantly discrete environments using Stable-Baselines3 (SB3) [3]. Discrete environments are characterized by a finite number of possible actions and observations, making them suitable for discrete RL algorithms. SB3 provides a robust and flexible framework for implementing RL algorithms and offers a wide range of algorithms for training agents in diverse environments.

We will consider three different environments to evaluate the performance of RL agents trained using SB3 algorithms, specifically Proximal Policy Optimization (PPO), Advantage Actor-Critic (A2C) and Deep Q-Network (DQN).

The environments are:

- Acrobot;
- an Atari game, specifically Demon Attack;
- and, finally, a set of customized environments in HighwayEnv [2].

By exploring these diverse environments and using PPO and A2C algorithms, we aim to analyze the effectiveness and adaptability of SB3 algorithms in training agents to learn optimal policies in discrete settings. The results obtained will provide insights into the performance and behavior of RL agents in different scenarios and contribute to the understanding of RL algorithms in practical applications.

Our project can be found on Github at `https://github.com/40115/ProjectAI`

Background and Concepts

RL is a framework within ML that involves an agent interacting with an environment to learn a sequence of actions that maximize cumulative rewards. In RL, the agent learns by trial and error, aiming to make optimal decisions based on feedback from the environment.

At the core of RL are several fundamental concepts. The agent is the learner or decision-maker, responsible for taking actions in the environment. The environment represents the external system or problem space with which the agent interacts. The state refers to the current condition or representation of the environment, which the agent perceives as input. The action represents the decision made by the agent to transition from one state to another. The reward is a scalar signal that provides feedback to the agent, indicating the desirability of its actions.

RL algorithms play a crucial role in training agents to learn optimal policies. These algorithms leverage different approaches, such as value-based methods, policy-based methods, and actor-critic methods, to explore and exploit the environment in order to maximize rewards. Value-based methods estimate the value of different actions or states, while policy-based methods directly optimize the agent's policy. Actor-critic methods combine both approaches, using a value function to guide policy updates.

In this project, we focus on training agents in predominantly discrete environments. The discrete action space refers to a finite set of possible actions available to the agent, while the discrete observation space represents the possible states or perceptions of the environment that the agent can observe.

Experimental Setup

In this chapter, we provide a detailed description of the experimental setup used in our project to train RL agents in predominantly discrete environments using SB3 algorithms. We discuss the RL framework, algorithms, environments, observation and action spaces, hyperparameters, training procedure, hardware and software setup, evaluation metrics, and data collection and analysis.

RL Framework and Algorithms

We utilized the SB3 library as our RL framework. SB3 is a powerful and flexible library that provides a wide range of RL algorithms and supports various environments. It offers a user-friendly interface, allowing for easy implementation and experimentation with different RL algorithms.

The two RL algorithms employed in our experiments are PPO and A2C. PPO is a policy optimization algorithm that balances exploration and exploitation through the use of surrogate objective functions and policy updates. A2C is an actor-critic algorithm that combines value-based and policy-based approaches, using a value function to guide policy updates. DQN, on the other hand, is a value-based algorithm that uses a deep neural network to approximate the Q-values and makes decisions based on the maximum predicted Q-value.

PPO, A2C, and DQN are well-suited for training RL agents in discrete environments, as they can handle discrete action spaces effectively and provide good exploration-exploitation trade-offs.

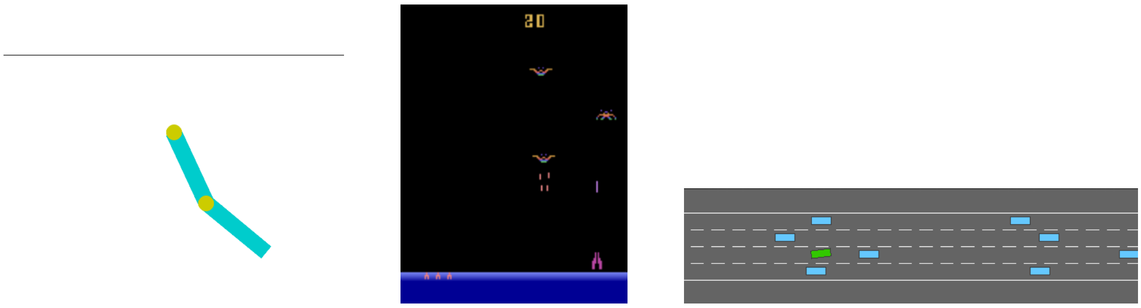
Environments

As stated previously, we considered three different environments for training our RL agents: Acrobot, DemonAttack (an Atari game), and Highway (a customized environment). While the first two are two environments on Gymnasium [1], the Highway environment is a third-party collection of environments for autonomous driving and tactical decision-making tasks. All of these environments have discrete action spaces and continuous observation spaces.

The Acrobot environment consists of two links connected linearly to form a chain, with one end of the chain fixed. The joint between the two links is actuated. The goal is to apply torques on the actuated joint to swing the free end of the linear chain above a given height while starting from the initial state of hanging downwards. It is stochastic in terms of its initial state.

The Demon Attack environment is an Atari game is simulated via the Arcade Learning Environment (ALE) through the Stella emulator. Points are accumulated by destroying demons. You begin with 3 reserve bunkers, and can increase its number (up to 6) by avoiding enemy attacks. Each attack wave you survive without any hits, grants you a new bunker. Every time an enemy hits you, a bunker is destroyed. When the last bunker falls, the next enemy hit will destroy you and the game ends. This environment presents a more complex and visually rich environment for the RL agents to navigate.

The HighwayEnv environments are customized environments simulating different traffic scenarios. The RL agents are required to learn to navigate safely and efficiently through the traffic, making it a challenging RL task.



(a) Acrobot (b) Demon Attack (c) Highway

Figure 1: Environments

Results and Analysis

The results obtained from training RL algorithms in different environments are presented in Table 1, which shows the mean reward values for each algorithm in the Acrobot, Demon Attack, and Highway environments.

Table 1: Mean Reward Values

Environment	**PPO**	**A2C**	**DQN**
Acrobot	-82.50 ± 10.46	-96.00 ± 56.19	
Demon Attack	0.00 ± 0.00	65.00 ± 9.22	
Highway	28.29 ± 1.04	27.67 ± 0.84	4.85 ± 1.64

From the results, it can be observed that in the Acrobot environment, both PPO and A2C algorithms achieved negative mean rewards, indicating suboptimal performance. However, the A2C algorithm had a higher variance in its rewards compared to PPO. In the Demon Attack environment, A2C achieved a positive mean reward of 65.00 ± 9.22, while PPO had a mean reward of 0.00 ± 0.00, suggesting limited learning in this environment. In the Highway environment, both PPO and A2C algorithms achieved similar mean rewards, with PPO having a slightly higher mean reward and lower variance compared to A2C.

It is important to note that these results are based on a limited number of experiments and do not provide conclusive information. There are several limitations on this research. There was a lack of exploration of different hyperparameters and we did not gather information from multiple tests with each algorithm. And other metrics and benchmarks should've been taken into account besides the mean reward, like episode length and success rate. The experiments were primarily focused on getting the training and predictions to run successfully, rather than thoroughly comparing their performance.

Conclusion

We explored the application of RL algorithms in discrete environments using SB3. Our objective was to gain initial insights into the performance of a few algorithms that supported discrete action spaces - PPO, A2C, and DQN - in three distinct environments: Acrobot, an Atari game (Demon Attack), and a customized environment (Highway).

Our experiments provided valuable preliminary findings regarding the performance of the RL algorithms. However, it is important to note that the study had certain limitations. Due to time constraints, we were unable to extensively experiment with different hyperparameters or conduct multiple tests for each algorithm. Our main focus was on getting the training and prediction processes up and running.

From the limited experiments conducted, we observed some trends in the performance of the RL algorithms. PPO and A2C showed comparable results in the Acrobot environment, with PPO achieving a mean reward of -82.50 ± 10.46 and A2C achieving -96.00 ± 56.19. In the Highway environment, the DQN algorithm achieved a mean reward of 4.85 ± 1.64. However, direct comparisons between the algorithms should be made cautiously, considering the limitations of our study.

One of the key limitations of our project was the lack of extensive hyperparameter tuning. Different hyperparameter configurations could potentially lead to improved performance for each algorithm. Additionally, conducting multiple tests and averaging the results would provide more reliable and robust conclusions.

Despite these limitations, our project serves as a preliminary exploration of RL algorithms in discrete environments. It highlights the potential of PPO, A2C, and DQN algorithms in solving complex problems and improving agent performance.

In conclusion, our project provides initial insights into the performance of PPO, A2C, and DQN algorithms in discrete environments. While the limitations of the study hindered a thorough comparison, the results suggest promising avenues for future research. With additional time and resources, it would be worthwhile to further investigate the performance of these algorithms, explore hyperparameter optimization, and conduct more extensive evaluations.

Bibliography

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[2] Edouard Leurent. An environment for autonomous driving decision-making. `https://github.com/eleurent/highway-env`, 2018.

[3] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.