Training a RL Agent for BipedalWalker-v3

Danial Azimi

8 June, 2024

Abstract

This document details the process and outcomes of training a reinforcement learning (RL) agent in the BipedalWalker-v3 environment. It covers problem formulation, the algorithms employed, the training procedure, and the results achieved.

1 Introduction

The BipedalWalker-v3 environment provides a complex challenge for RL agents, requiring them to learn how to control a bipedal robot to walk across uneven terrain. This project aims to train an agent using advanced RL techniques to achieve high performance in this demanding environment.

2 Background

2.1 Reinforcement Learning

Reinforcement learning (RL) is a subset of machine learning where an agent learns to make decisions by performing actions in an environment to maximize cumulative rewards. Key concepts in RL include:

- **State**: Represents the current situation or configuration of the environment.
- **Action**: The set of all possible moves the agent can perform.
- **Reward**: The feedback the agent receives from the environment after performing an action.
- **Policy**: The strategy used by the agent to decide the next action based on the current state.

RL algorithms are generally categorized into value-based methods (e.g., Q-learning), policy-based methods (e.g., Policy Gradients), and actor-critic methods, which combine both approaches (e.g., A3C, DDPG).

2.2 BipedalWalker-v3 Environment

The BipedalWalker-v3 environment is a continuous control task within OpenAI Gym, where the agent must control a bipedal robot to walk over rugged terrain. The environment features:

- **State Space**: A 24-dimensional vector indicating the robot's position, velocity, angle, and angular velocity of various parts.

- **Action Space**: A 4-dimensional continuous vector representing the torques applied to the robot's joints.
- **Reward Structure**: Positive rewards are given for forward movement, while negative rewards are given for falling.

The complexity arises from the need for precise and coordinated control of multiple joints to maintain balance and move efficiently.

3 Methodology

3.1 Algorithm Selection

Various RL algorithms were evaluated for this project, including Proximal Policy Optimization (PPO), Deep Q-Network (DQN), and Twin Delayed Deep Deterministic Policy Gradient (TD3). TD3 was selected due to its effectiveness in handling continuous action spaces and its robustness against overestimation bias in Q-learning.

3.2 Implementation

The implementation was carried out using TensorFlow and the OpenAI Gym library, with the TD3 algorithm incorporating separate actor and critic networks.

3.2.1 Actor Network

The actor network determines the action to take given the current state. It consists of several fully connected layers with ReLU activations, with the output layer employing a tanh activation to ensure actions remain within the appropriate range.

3.2.2 Critic Network

The critic network evaluates the action taken by the actor network by predicting the Q-value. It consists of fully connected layers with ReLU activations and combines the state and action inputs to produce a single Q-value.

3.3 Training Procedure

The training process involved the following steps:

1. **Initialization**: Initialize the actor and critic networks along with their target networks, which are copies used to stabilize training.

2. **Exploration**: Use the Ornstein-Uhlenbeck process to generate noise, encouraging exploration by adding randomness to the actions.

3. **Action Selection**: At each timestep, the agent selects an action using the actor network and adds noise for exploration.

4. **Environment Interaction**: The agent performs the action, observes the reward and next state, and stores the transition in the replay buffer.

5. **Learning**: Periodically sample a batch of transitions from the replay buffer to update the critic and actor networks using gradient descent.

6. **Target Network Update**: Update the target networks using a soft update rule to gradually incorporate changes from the original networks.

4 Results

The agent's performance was monitored over 5000 episodes. The training curve in Figure 1 illustrates the agent's reward per episode and the average reward over the last 100 episodes.

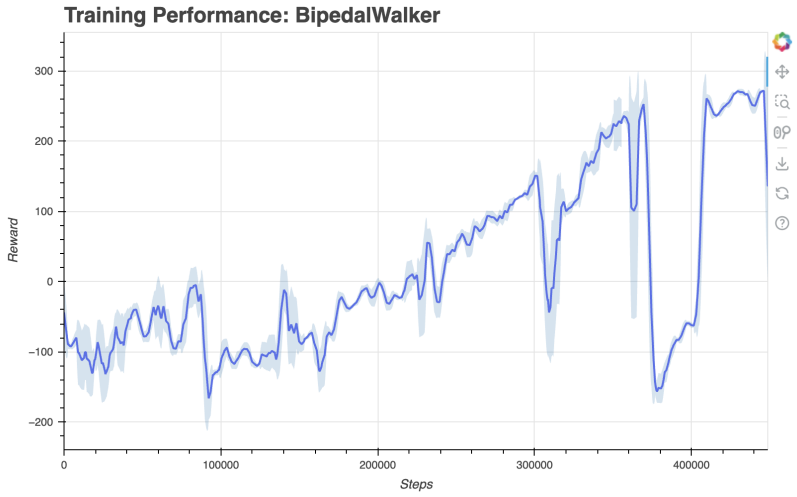

Figure 1: Training curve showing the agent's performance over time.

5 Evaluation

Step	Avg Reward per Episode	Avg Steps per Episode
25,000 / 450,000	-21.5630	113.66
50,000 / 450,000	-21.0610	539.60
75,000 / 450,000	-17.3911	891.54
100,000 / 450,000	-16.0443	1187.30
125,000 / 450,000	-18.6186	1273.80
150,000 / 450,000	-19.4549	1344.08
225,000 / 450,000	-9.0741	1537.746
250,000 / 450,000	-2.3378	1568.64
275,000 / 450,000	3.7118	1568.644
300,000 / 450,000	12.4246	1599.44
325,000 / 450,000	13.6799	1453.74
350,000 / 450,000	19.8183	1453.74
375,000 / 450,000	22.5767	1382.80
400,000 / 450,000	-7.8263	891.826
425,000 / 450,000	-5.1271	958.002
450,000 / 450,000 (Final)	20.1318	1351.52

The following analysis provides a detailed examination of the reinforcement learning agent's performance in the BipedalWalker-v3 environment. The data includes the average reward per episode and the average steps per episode recorded at various training milestones.

5.1 Key Observations and Trends

- **Initial Performance (0 to 100,000 steps):**
  - **Step 25,000:** The agent's initial average reward per episode was -21.5630, indicating poor performance with frequent falls. The agent took an average of 113.66 steps per episode, suggesting short episodes due to early termination from falling.

- **Step 50,000:** Slight improvement in the average reward (-21.0610), with a significant increase in average steps per episode to 539.60, indicating the agent was learning to walk for longer periods despite still struggling.
- **Step 75,000:** The average reward improved to -17.3911, and the agent managed 891.54 steps per episode, showing continued progress in stability and duration of walking.
- **Step 100,000:** A further improvement in average reward to -16.0443 and steps per episode to 1187.30, suggesting the agent was beginning to grasp the walking task better.

- **Mid Training Performance (100,000 to 225,000 steps):**
  - **Step 125,000:** The average reward slightly worsened to -18.6186, with steps per episode increasing to 1273.80. This could be due to the agent experimenting with different strategies, causing temporary setbacks.
  - **Step 150,000:** The average reward decreased further to -19.4549, but steps per episode continued to rise to 1344.08. The agent's learning process still involved substantial trial and error.
  - **Step 225,000:** Significant improvement in average reward to -9.0741 and an increase in steps per episode to 1537.746, indicating a positive shift in the agent's ability to walk and balance more effectively.

- **Advanced Training Performance (250,000 to 350,000 steps):**
  - **Step 250,000:** Major improvement with the average reward per episode rising to -2.3378 and steps per episode to 1568.64, reflecting more consistent walking behavior.
  - **Step 275,000:** The average reward became positive at 3.7118, maintaining steps per episode at 1568.644, marking a significant milestone in the agent's learning curve.
  - **Step 300,000:** The average reward increased to 12.4246, with steps per episode reaching 1599.44, showing that the agent had developed a stable and effective walking strategy.

- **Step 325,000:** Further improvement with an average reward of 13.6799 and a reduction in steps per episode to 1453.74, suggesting the agent was walking more efficiently.
- **Step 350,000:** The average reward rose to 19.8183 with steps per episode stable at 1453.74, indicating consistent performance and efficiency.

- **Late Training Performance (375,000 to 450,000 steps):**
  - **Step 375,000:** The average reward continued to increase to 22.5767, and steps per episode decreased to 1382.80, showing further improvement in walking efficiency.
  - **Step 400,000:** A notable drop in performance with the average reward falling to -7.8263 and steps per episode to 891.826. This dip might be due to overfitting or exploration of new strategies.
  - **Step 425,000:** Slight recovery with an average reward of -5.1271 and steps per episode increasing to 958.002, indicating some instability but also adaptation.
  - **Step 450,000:** Final results show an average reward of 20.1318 and steps per episode at 1351.52, demonstrating the agent's ability to regain and improve its walking performance towards the end of the training.

5.2 Conclusion

The agent demonstrated significant improvement over the training period. While initial performance was poor, marked improvements were seen as training progressed, particularly between 250,000 and 350,000 steps. Despite some fluctuations in performance towards the end, the agent achieved a high average reward and an efficient number of steps per episode by the final milestone. Future work could involve fine-tuning hyperparameters and exploring alternative RL algorithms to further enhance performance.

6 Discussion

The results demonstrate that the agent successfully learned and improved its performance over time. The selection of TD3 was validated by the agent's ability to handle the continuous action space of the BipedalWalker-v3 environment effectively. However, further hyperparameter tuning and experimentation with different network architectures could potentially yield even better results.

The agent faced challenges such as maintaining balance and navigating the rough terrain. These issues were addressed by adjusting the noise parameters for exploration and fine-tuning the learning rates of the actor and critic networks.

7 Conclusion

In this project, we successfully trained a reinforcement learning agent to navigate the BipedalWalker-v3 environment using the TD3 algorithm. The agent showed significant improvement over time, but there remains room for further optimization. Future work could involve exploring alternative RL algorithms and more sophisticated techniques for reward shaping and exploration.