Deep Reinforcement Learning for Spatio-Temporal Wildlife Management

Anmol Singh

NorwAI
Norwegian Research Center for AI Innovation

NTNU

Abstract

In the past 50 years, global wildlife populations have plummeted resulting in a biodiversity crisis. This thesis tested the performance of various deep reinforcement learning (DRL) algorithms on the task of spatio-temporal wildlife management, for the purpose of maintaining biodiversity. The results obtained demonstrate the potential of DRL for wildlife management.

Introduction

DRL is a subfield of machine learning that combines deep neural networks with reinforcement learning, thus enabling an RL agent to solve complex problems in intricate environments. While DRL has been successfully applied in games like Chess and Atari 2600, there have been limited efforts to apply it to wildlife management. To address this, a tri-trophic spatio-temporal wildlife management simulation was created and the DRL algorithms DQN, A2C, and PPO were tested on it. The focus of this thesis was to find the best action set for the RL agent, and the DRL algorithm with the best performance. The performance of the algorithms was based on the sizes of the species populations.

Methodology

Spatio-temporal Wildlife Environment

An RL environment was created from scratch to simulate a realistic tri-trophic ecosystem. This spatio-temporal wildlife environment allowed the RL agent to explore and learn good wildlife management policies. The environment provided the agent with information about the three species' population sizes and their geographic distribution. Random initialization of population sizes could make the environment unbalanced as time passed. The agent's aim was to keep a stable, diverse and rich ecosystem. To achieve its aim, the agent could remove or add a species' population in any segment of the environment at each time step. The performance of the RL agent was measured with a reward function, which was based on a combination of biodiversity metrics and action costs.

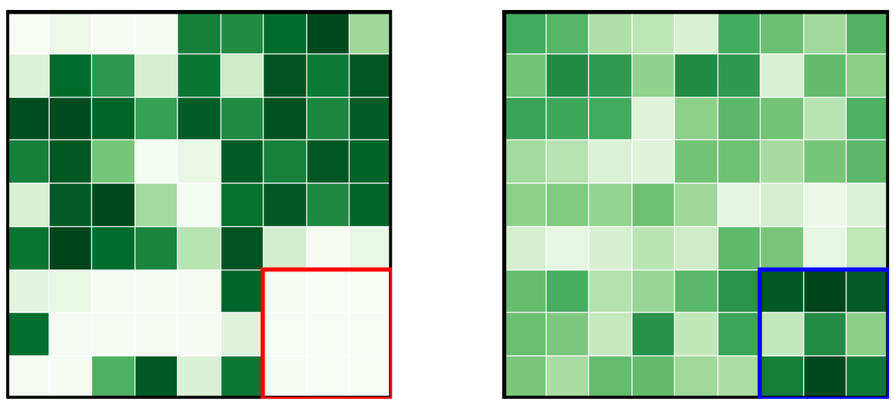

Figure 1: Examples of removing and adding a species' population in a segment of an environment. Darker cells represent higher population density.

The RL environment was implemented using the *Gymnasium* framework (which is built on *OpenAI gym*).

DRL Algorithms

Three popular DRL algorithms were tested to train the RL agent in the wildlife environment: DQN, A2C and PPO. DQN is a value-based method that attempts to learn good state-value approximations which are used to decide what actions to take. A2C on the other hand is a policy gradient approach, meaning that it directly updates its policy based on policy gradients. PPO is similar to A2C, but uses a clipped "surrogate" objective function to ensure that there are no significant updates to its policy. This ensures more stability in its performance.

As both A2C and PPO are on-policy algorithms, they could run multiple *workers* in parallel. This allowed for both more diverse data and faster training.

The *Stable Baseline3* implementations of the DRL algorithms were used. These were built on *PyTorch* and are compatible with Gymnasium environments.

Experimental Setup

Research question 1 aimed to find the best action set for the agent. The action set could vary with the number of animals added (*action multiplier*), and the segment size (*action unit size*). The experiments were carried out on a 9×9 environment, where the action multiplier could be 5x, 10x or 15x. Higher action multiplier meant more animals added, and higher cost. The action unit size could be 2×2, 3×3 or 4×4. Each DRL algorithms was trained for 200 000 environment steps at each run. To get robust results, each DRL algorithm had 20 training runs. The mean and 95% confidence interval was calculated for each DRL algorithm based on these runs.

The best action set was found in a systematic way where the DRL algorithms were first trained with varying action unit sizes, but a fixed action multiplier. Based on this data one could determine the best action unit size with high certainty. Using that action unit size, experiments were ran with varying action multipliers. The action multiplier and action unit size that enabled the DRL algorithms to perform the best, with regards to average episode reward, would be the best action set.

Research question 2 focused on finding the DRL algorithm with the best performance. This was based on the data collected to answer the first research question. Performance was based on average episode reward, but also biodiversity metrics and training speed and stability.

Results

The results showed that while all DRL algorithms performed best with action unit size 2×2, they performed best with different action multipliers. A deeper look at the trained DRL algorithm runs revealed that the different algorithms employed different policies, which could explain why they have different best action sets. However, all algorithms performed at least average with an action multiplier of 10x.

When it came to algorithm performance A2C achieved the highest reward, but was relatively unstable. DQN also obtained high rewards, but had slow training speed. PPO, while not improving quickly, achieved nearly similar rewards over the same time while being remarkably stable. It also scored highest on the biodiversity metrics, thus performing best overall.

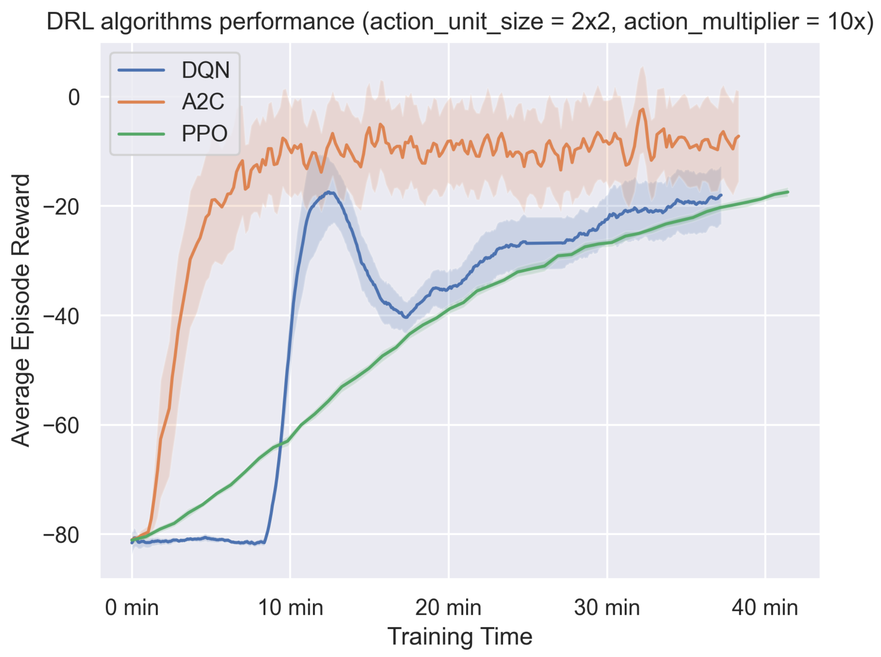

Figure 2: Performance of DQN, A2C and PPO over equal training time

Conclusion

In this thesis, a spatio-temporal wildlife management simulation was created, which acted as an RL environment for the DRL algorithms DQN, A2C and PPO. Their performance was evaluated and a decent action set was found. Algorithm performances with regards to rewards, biodiversity metrics, training times and stability was also discussed. PPO scored high on most of these metrics. Overall, the results in this thesis display the largely unexplored potential of DRL in wildlife management.