Playing the Chrome Dino Game with Reinforcement Learning: A Deep Q-Learning Approach with Custom Environment Design

Malvi Bid | 20187945 | hcymb2@nottingham.edu.my

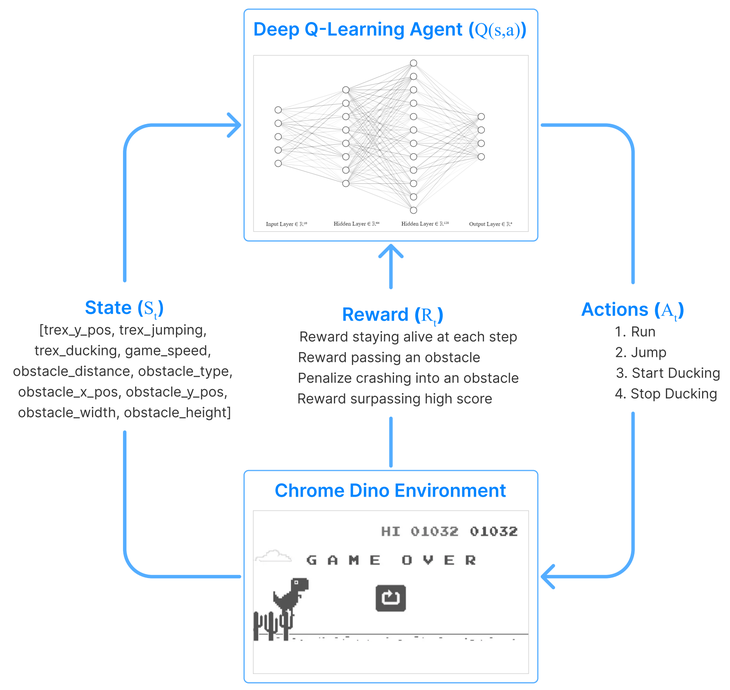
Introduction

The Chrome Dino game is a simple runner game requiring players to control a dinosaur through a desert landscape, skillfully avoiding obstacles like cacti and pterodactyls. The player's objective is to reach a high score by surviving for as long as possible. This game demands quick reflexes and precise timing which serves as an ideal platform for training intelligent agents to learn complex skills through trial and error. To tackle this challenge, a custom environment was created to accurately replicate the game's dynamics, while employing Reinforcement Learning with Deep Q-Learning to develop a highly proficient agent capable of adapting to varying difficulty levels and achieving robust performance.

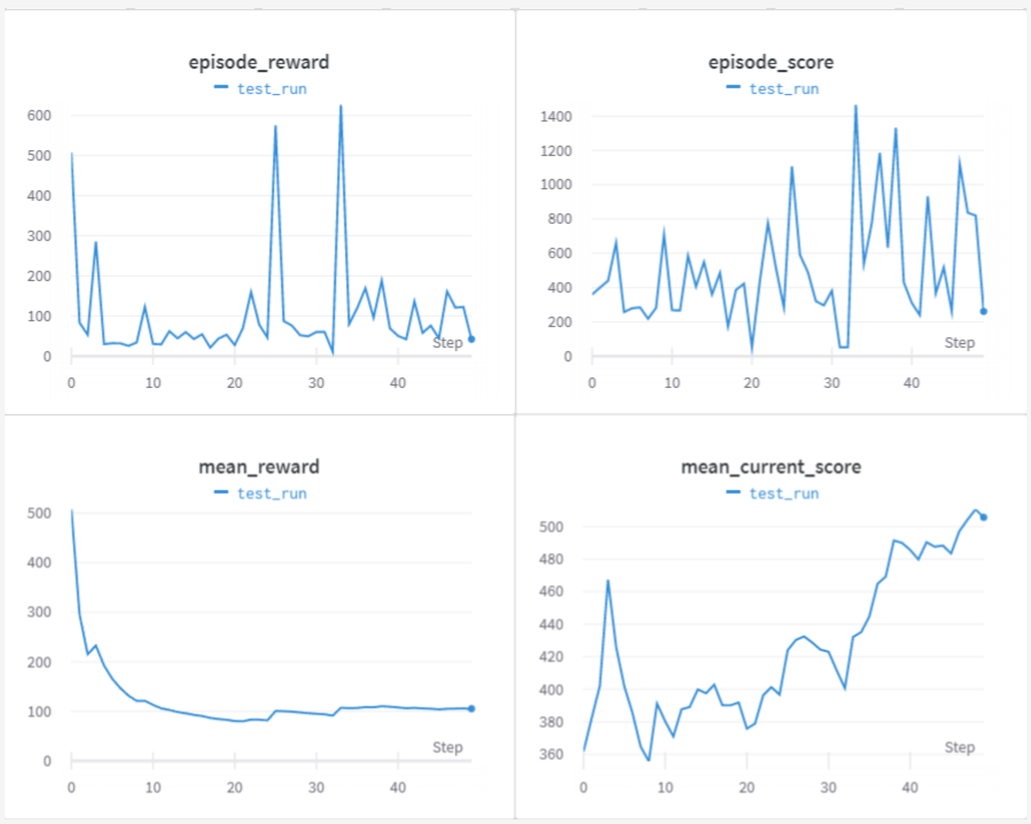
Methodology

As illustrated in the figure on the right, the problem involves an agent interacting with the game environment by executing a sequence of actions and learning from the observations and rewards received. The game is stochastic and partially observable.

The custom environment, `DinoEnvironment`, is a subclass of the Env class from the Gymnasium library. I used the Selenium Test Automation library to access the Chrome Dino webpage, get the game state using the game's "`Runner.instance_`" javascript object and to perform the actions predicted by the agent.



The agent, `DinoAgent`, iteratively updates its Q-function, $Q(s,a)$ using the Bellman equation, by randomly sampling the experiences it collected in the replay buffer, and then selects an action using the epsilon-greedy exploration-exploitation strategy. Its goal is to find an optimal policy, $Q^*(s,a)$ that maximizes the cumulative reward.



Experiments and Results

In the experiment phase, different reward strategies were tested to optimize the agent's learning in the Chrome Dino game. After training iterations, the best model was obtained with effective jumping and running abilities, but occasional unnecessary actions. Attempts were made to refine the reward rules to encourage correct actions, but strict rules proved challenging for the model to learn effectively. Additional information about the T-rex was also tested but did not yield significant improvements. To enhance the learning process, an experimental technique called "exploration breaks" was introduced, balancing exploration and exploitation.

The best performing agent achieved proficiency in jumping over obstacles and adapted well to the increasing game speed. However, it had difficulties with proper ducking, hindering its progress in more complex scenarios. Despite these challenges, the agent showed promising adaptability and ongoing learning. The highest observed score during testing was 1,618, and further evaluation over 50 episodes gave a high score of 1467 and showed an upward trend in the average score, indicating ongoing improvement. The stochastic nature of the game, as evidenced by the noticeable fluctuations in the episode reward and episode score graphs, presented challenges and showcased the dynamic and complex environment the agent had to master.

A video showcasing the best trained Dino agent playing the game is available at: https://youtu.be/QFf0_4FCh0w.

Discussion

The main challenges encountered during training arise from the stochastic nature of the game and the random sampling approach of experience replay. The game's randomness makes it difficult for the agent to learn an optimal policy, while uniform sampling from the memory buffer gives equal importance to both good and bad actions. Consequently, the agent may struggle to learn from more relevant experiences or distinguish between high-quality and low-quality actions. Furthermore, the epsilon-greedy strategy limits the agent's exploration. Due to the random actions taken by the agent while exploring, the game often ends prematurely, preventing the agent from reaching more difficult stages of the game hampering its ability to handle complex scenarios. Inconsistent performance between episodes makes it challenging to monitor training and determine the best model to save.

Optimizations such as Prioritized Experience Replay and fixed exploration phases can improve the learning process. Prioritized Experience Replay assigns importance weights to each experience in the memory buffer, allowing the agent to sample more meaningful experiences more frequently during learning. On the other hand, a fixed exploration phase refers to a predetermined period during which the agent focuses on exploring the environment and gathering experiences before transitioning to the exploitation phase, where it starts learning and refining its policy.

Conclusion

In conclusion, the best-performing model demonstrated considerable success in playing the Chrome Dino Game despite some limitations in handling specific obstacles. This project has laid a foundation for future research in optimizing DQN agents for more complex tasks and environments. Further work could explore techniques that encourage agents to learn from meaningful experiences, such as Prioritized Experience Replay and other advanced learning strategies.