The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

COMP3071 DESIGNING INTELLIGENT AGENTS

COURSEWORK TOPIC APPROVAL

STUDENT NAME: MALVI SURESH BID
STUDENT ID: 20187945

Table of Contents

Project Name

DinoRL: A Reinforcement Learning Agent for Chrome Dino

Problem Statement

The problem is to design an intelligent agent that can play the Chrome Dino game efficiently and score high. The Chrome Dino game is a simple yet challenging game where the player controls a dinosaur that runs through a desert landscape, jumping over obstacles and avoiding cacti. While seemingly straightforward, the game requires quick reflexes and precise timing, making it an ideal testbed for developing intelligent agents that can learn to perform complex tasks through trial and error.

The main challenge in creating such an agent is designing an environment that accurately captures the dynamics of the game, including the dinosaur's movements, the obstacles and their positions, and the reward structure. Furthermore, the agent must be able to learn a policy that maximizes its long-term reward, despite the stochasticity and partial observability of the game. In addition, the agent must be able to handle variations in the game's difficulty level, such as changes in obstacle speed or frequency, to ensure robust performance.

To address these challenges, I propose to use reinforcement learning to develop an intelligent agent that can learn to play the Chrome Dino game. By using a trial-and-error approach to learn from experience, the agent can adapt to the complex and dynamic environment of the game, generalize its knowledge to new situations, and discover innovative strategies for maximizing its reward (Exploration and Exploitation).

Through experimental evaluation, the aim is to demonstrate that this approach can achieve human-level performance (highest human score is 35,464) on this challenging task. In addition, see the highest score it can achieve, efficiently.

Using RL agents to play games can be a useful and effective approach for developing intelligent systems that can operate in complex and dynamic environments. Furthermore, the development of RL agents for games may have broader implications for developing autonomous systems in other domains, like robotics, finance, and healthcare by providing transferable algorithms and techniques, simulation and testing environments, exploring human-AI interaction, and addressing data efficiency issues.

Existing Solutions

There are several approaches to playing the Chrome Dino game without using reinforcement learning, including:

1. Rule-based systems: These systems use predefined rules to make decisions and act in the game. These rules can be hand-coded or generated automatically using techniques like decision trees. For example, the agent can jump when an obstacle is detected, or it can adjust the height and distance of the jump based on the size and position of the obstacle. Although, simple and fast to implement, these systems are limited in their ability to handle complex situations and they require a lot of manual tuning to perform well. They are also unable to learn from experience.

2. Genetic Algorithms: These algorithms can be used to evolve a population of Chrome Dino players by applying natural selection to different sets of inputs and output actions. Over time, the best-performing (fittest) players can be identified and used to create a new generation of players with slightly modified input/output mappings. This solution has the advantage of being able to explore a wide range of possible solutions (it can handle complex situations and find innovative solution, and adapt to changing conditions), but it may require a large number of generations to find a good solution, it is computationally expensive, and it may not find the optimal solution.

3. Neural networks: Neural networks can be used to learn a policy for playing the Chrome Dino game by training on a dataset of game screens and corresponding actions. This approach requires a large amount of training data and can be computationally expensive but can often result in more accurate and robust game-playing strategies than other approaches.

4. Hybrid approaches: Finally, hybrid approaches can be used to combine different techniques, such as a rule-based/ hand-coded algorithm with a neural network, to create a more effective Chrome Dino player.

While these solutions can be effective in playing the Chrome Dino Game, they may not be as effective as Reinforcement Learning in certain scenarios.

Differentiation

The points below highlight why Reinforcement Learning is a powerful tool for playing complex games like the Chrome Dino game[1]:

1. Adaptability: Reinforcement learning allows the agent to learn from experience and adapt to new situations, without requiring explicit programming or rules. This makes it possible to create agents that can handle complex and dynamic environments, such as the Chrome Dino game, where the optimal actions may change depending on the state of the game.

2. Generalization: Reinforcement learning agents can generalize their knowledge to new situations, even if they have not been explicitly trained on those situations. This makes it possible to create agents that can perform well on a wide range of environments, without requiring extensive training on each environment.

3. Flexibility: Reinforcement learning allows the agent to explore a wide range of possible solutions and strategies, without being limited by the biases or assumptions of the programmer. This makes it possible to create agents that can discover new and innovative solutions to the problem.

4. Efficiency: Reinforcement learning agents can learn from experience using online learning algorithms, which update the agent's policy in real-time as it interacts with the environment. This makes it possible to create agents that can learn quickly and efficiently, without requiring large amounts of data or computational resources.

Research Questions

The key research questions to solve the problem are:

1. Can a deep reinforcement learning agent learn to play the Chrome Dino game efficiently?
2. How can the agent be designed to recognize the game state and make decisions?
3. How to evaluate the performance of the agent?
4. What is the optimal architecture for the agent to maximize performance?

Experiments

The following experiments will be conducted to answer the research questions:

1. Collect observations of game states and actions.
2. Train a reinforcement learning agent using the collected observations.
3. Evaluate the performance of the agent by measuring its score and efficiency.
4. Experiment with different architectures and hyperparameters to optimize performance.

Metrics

The following metrics will be used to evaluate the success of the proposed solution:

1. Score: The maximum score achieved by the agent over a fixed number of game iterations.
2. Efficiency: The number of actions taken by the agent to achieve a specific score.
3. Adaptability: The ability of the agent to generalize to different scenarios and environments.

References

1. *Introduction to reinforcement learning with David Silver* (2015) *DeepMind*. Available at: https://www.deepmind.com/learning-resources/introduction-to-reinforcement-learning-with-david-silver.