Reinforced learning for Realtime strategy games

Jagmohan Prajapat(1HK21AI021)
Shahid Hussain(1HK21AI033)
Mohammad Fatha P(1HK21AI037)
Syed Daniyal Ali(1HK21AI57)
Akhil Nair(1HK21AI002)

PROJECT DOMAIN

AI for Real-Time Strategy Games

ABSTRACT

Focus: Applying reinforcement learning (RL) to develop intelligent AI for real-time strategy (RTS) games.

Challenges:

- Representing game state
- Defining actions available to the agent
- Designing reward system (feedback for the agent)
- Balancing exploration (trying new things) and exploitation (using known good strategies)
- Learning effective policies (decision-making rules)
- Handling adversarial environments (opponent players)
- Integrating memory for past actions
- Making decisions in real-time

Current Approaches:

- Deep Q-learning
- Policy gradients
- Multi-agent reinforcement learning
- Transfer learning

Future Directions:

- Combining RL with other AI techniques (planning, search)
- Applying RL to real-world problems beyond games

Target Audience: Researchers and practitioners interested in RL for RTS games

SCOPE AND OBJECTIVE

- **1. Scope:**
- This research focuses on the application of reinforcement learning (RL) techniques to real-time strategy (RTS) games.
- The study encompasses both single-player and multiplayer RTS games.
- Various RL algorithms, including deep reinforcement learning (DRL), will be explored within the scope of this research.
- The research considers diverse aspects of the domain, such as state representation, action space, reward design, exploration vs. exploitation, adversarial learning, memory, and real-time constraints.

OBJECTIVES

- **To Analyze the Challenges:** Investigate the unique challenges posed by RTS games for RL algorithms, including the large state and action spaces, real-time decision-making, and adversarial nature.
- **To Develop Effective RL Strategies:** Design and implement RL strategies tailored to address the specific challenges of RTS games, focusing on state-of-the-art algorithms and techniques.
- **To Evaluate Performance:** Evaluate the performance of RL agents in various RTS game environments, comparing them against baseline strategies and human-level performance where applicable.
- **To Explore Transfer Learning:** Investigate the potential for transfer learning in RTS games, exploring how knowledge gained in one game environment can be applied to improve performance in others.

Existing System

If you're looking for information about existing systems or frameworks that apply reinforcement learning (RL) to real-time strategy (RTS) games, here are a few notable examples:

1. **OpenAI's "OpenAI Five" for Dota 2**:
    1. OpenAI developed "OpenAI Five," a team of five neural networks, to play the popular RTS game Dota 2.
    2. OpenAI Five demonstrated advanced coordination and strategic decision-making, competing against professional human players.
2. **AlphaStar for StarCraft II**:
    1. AlphaStar is an AI system developed by DeepMind for playing StarCraft II.
    2. It achieved Grandmaster level in the game, surpassing 99.8% of human players, and defeated top professional players.
3. **SC2LE (StarCraft II Learning Environment)**:
    1. Developed by DeepMind and Blizzard Entertainment, SC2LE provides a platform for training and evaluating AI agents in StarCraft II.
    2. It includes tools for interfacing with the game, collecting data, and training RL models.

Drawbacks of existing systems

1. **Complexity and Resource Requirements**:
    1. Many existing systems, such as AlphaStar and OpenAI Five, require significant computational resources and infrastructure for training and deployment. This can limit accessibility to researchers and developers with limited resources.
2. **Domain Specificity**:
    1. Some systems are tailored to specific games, such as Dota 2 or StarCraft II, making it challenging to generalize their approaches to other RTS games. This limits the applicability of the techniques to a broader range of game environments.
3. **Lack of Transparency**:
    1. The inner workings of some systems, particularly proprietary ones like AlphaStar, may not be fully transparent or accessible to the research community. This can hinder efforts to understand and reproduce their results.
4. **Limited Generalization**:
    1. While systems like AlphaStar and OpenAI Five have demonstrated impressive performance in specific contexts, their ability to generalize to unseen scenarios or different game versions may be limited. This raises questions about their robustness and adaptability in real-world applications.

Proposed System

Proposed System: RL-RTS Framework

- **1. Modular Architecture:**
- The system will feature a modular architecture designed to accommodate various RTS games and RL algorithms.
- Each module will be loosely coupled, allowing for easy integration and interchangeability of components.
- **2. Game Interface:**
- A flexible game interface will be developed to interact with different RTS games, enabling data exchange and control of game states and actions.
- This interface will provide hooks for observing game states, executing actions, and collecting rewards, facilitating seamless integration with different game engines and environments.
- **3. Algorithmic Flexibility:**
- The system will support a wide range of RL algorithms, including deep reinforcement learning (DRL), model-based methods, and multi-agent techniques.
- Researchers and developers will have the flexibility to experiment with different algorithms and architectures within the framework.
- **4. Scalability and Efficiency:**
- Emphasis will be placed on scalability and efficiency to enable training RL agents on large-scale game environments without excessive computational resources.
- Techniques such as distributed training, asynchronous methods, and parameter sharing will be explored to optimize training efficiency.

Advantages

1. **Modular Architecture**: The modular architecture enables flexibility and adaptability, allowing researchers and developers to customize and extend the framework to suit their specific needs and preferences.
2. **Compatibility with Multiple Games**: The framework is designed to interact with various RTS games, providing researchers with a versatile platform for experimentation and evaluation across different game environments.
3. **Algorithmic Flexibility**: By supporting a wide range of RL algorithms, the framework allows researchers to explore different approaches and techniques, facilitating innovation and comparison between methods.
4. **Scalability and Efficiency**: Emphasizing scalability and efficiency enables training RL agents on large-scale game environments without excessive computational resources, accelerating experimentation and iteration.

Timeline of the project

In the next two months (If god wills)