Reinforcement Learning in Videogames

David Fuentes Insa

9 de març de 2025

Resum– Resum del projecte, màxim 10 línies. ....

Paraules clau– Paraules clau del treball, màxim 2 línies . ...

Abstract– Versió en anglès del resum . ...

Keywords– Versió en anglès de les paraules clau. ....

1 INTRODUCTION

REINFORCEMENT learning is one of the most promising approaches of artificial intelligence, especially in the videogames sector. Meanwhile other AI approaches use supervised training with labeled data, RL agents learn by interacting with the enviroment throught trial and error, and improve based on rewards or penalties for taking actions.

• E-mail de contacte: david.fuentesinsa@gmail.com
• Menció realitzada: Computació
• Treball tutoritzat per: Jordi Casas Roma
• Curs 2024/25

The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them. Inthe most interesting and challenging cases, actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics —trial-and-error search and delayed reward— are the two most important distinguishing features of reinforcement learning. [1]

In 2013, the company DeepMind released a paper regarding how their RL models could surpass a human expert playing Atari 2600 classic games. The model beat the human expert in 3 of 6 games tested. [3] In 2015. they released a more extended paper tested on 49 games. The model got to the level of a human professional. [4]. The most impressive archievement is AlphaGo, the model who beat the

European Go champion Fan Hui by 5 games to 0. [5].

Videogames provide an ideal enviroment to test the models because of the clear objectives and controlled actions. However, reinforcement learning applications extend not only in videogames, also in real life problems such as robotics, autonomous vehicles, healthcare, resource management, and much more.

2 OBJECTIVES

The main objective of this project is fully understanding Single-agent Reinforcement Learning and what can be acomplished with it using existing libraries but also how it works from zero. The next five objectives define the goal by the end of the project.

1. Learn RL Fundamentals: Study core RL concepts such as Markov Decision Processes (MDPs), value functions, and policy optimization. Implement simple algorithms like Monte Carlo or Dynamic Programming to build a solid base.
2. Learn RL Advanced Methods: Such as Temporal Difference learning (Q-learning and SARSA) and deep RL approaches like DQNs.
3. Implementing my RL model from zero: Without using libraries.
4. Experiment with self-play training in a videogame: Using a ping pong enviroment, train the agents from both sides of the game.
5. Analyze and Compare every model: With empiric data.

3 METHODOLOGY AND PLANNING

The methodology chosen is Kanban, an agile approach that has a clear visualization of the workflow, the evolution is continuous and is very flexible. Kanban methodology typically defines three columns that represent the state of a task. In the “To Do” column there are the tasks that are not being worked on at the moment. In the “Progress’ column, the tasks that are assigned and are currently being working on. And the final column “Done'”, with tasks that are fully finished.

The tool used for the implementation of Kanban is GitHub Projects [2], a free tool provided by GitHub.

The planning is divided in five phases of three or four weeks each.

- Phase 1: Reinforcement learning fundamentals and the State of Art
  - Task 1: Research and document core RL concepts (agents, environments, states, actions, rewards)
  - Task 2: Study non-tabular methods (Dynamic Programming, Monte Carlo, Q-learning, SARSA)
  - Task 3: Study tabular methods (DQN)
  - Task 4: Document the State of Art
- Phase 2: Enviroment setup and algorithms
  - Task 1: Study gymnasium library
  - Task 2: Implement non-tabular methods
  - Task 3: Implement tabular methods
  - Task 4: Document the implementations
- Phase 3: Implement RL from zero
  - Task 1: Select the algorithm to implement
  - Task 2: Program from zero the algorithm
  - Task 3: Test the algorithm
  - Task 4: Document the algorithm
- Phase 4: Test the algorithms in enviroments and compare
  - Task 1: Select and implement the enviroments for testing
  - Task 2: Test the algorithms in the enviroments
  - Task 3: Train a self-play model in ping-pong enviroment
  - Task 4: Compare empirically all the algorithms
- Phase 5: Final inform and conclusions
  - Task 1: Document the conclusions
  - Task 2: Final inform
  - Task 3: Project presentation

REFERÈNCIES

[1] Sutton & Barto (2018). Reinforcement learning: an introduction.

[2] About Projects - GitHub Docs https://docs.github.com/es/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects

[3] Mnih, Volodymyr and Kavukcuoglu, Koray and Silver, David and Graves, Alex and Antonoglou, Ioannis and Wierstra, Daan and Riedmiller, Martin. Playing Atari with Deep Reinforcement Learning. (2013). https://www.cs.toronto.edu/ vmnih/docs/dqn.pdf

[4] Mnih, V., Kavukcuoglu, K., Silver, D. et al. Human-level control through deep reinforcement learning. Nature 518, 529–533 (2015). https://doi.org/10.1038/nature14236

[5] Silver, D., Huang, A., Maddison, C. et al. Mastering the game of Go with deep neural networks and tree search. Nature 529, 484–489 (2016). https://doi.org/10.1038/nature16961