Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Master in Data Science

Master Thesis

Design and Prototyping of an Ecosystem Simulator for Intelligent Agents

Author: Jorge Lizcano Gómez-Calcerrada

Madrid, «February, 2024»

This Master Thesis has been deposited in ETSI Informáticos de la Universidad Politécnica de Madrid.

Master Thesis
Master in Data Science

Title: Design and Prototyping of an Ecosystem Simulator for Intelligent Agents
«February, 2024»

Author: Jorge Lizcano Gómez-Calcerrada
Supervisor: Juan A. Fdez del Pozo
Computational Intelligence Group
ETSI Informáticos
Universidad Politécnica de Madrid

Summary

<Here is the summary of the TFM. Maximum extension 2 pages.»

Abstract

«Abstract of the Master Project. Maximum length: 2 pages.»

Contents

Chapter 1

Introduction

This work has been inspired both by the recent advancements in the field of reinforcement learning in which there is a bottleneck for the researchers due to the lack in the variety of training environments for their algorithms.

1.1 Motivation

The field of artificial intelligence has been increasing it´s popularity in the last years. Since the publication of the paper "Attention is all you need" [3], the transformer arquitecture that it introduced has brought the creation of gigantic models cappable of processsing huge amounts of data. This event has revolutionized the industry generating an increasing variety of tools around what has been stablished as generative artifficial intelligence, for it´s capacity for creating text, images, and even videos or 3D models.

However all these improvements are based in the classic core idea of suppervised learning, with the only difference been the arquitectures that allow much more examples in the training process. If we continue with this trend, all the research advancements will be reserved for the big companies who have economic resources that grant them access to a high volume of infrastructure for creating huge models. With this scenario in mind, it is interesting that we take a step back and consider exploring other algorithms and learning methods that do not only deppend on raw data and computation power.

From the many types of algorithms in the field or artificial intelligence, lets dive into the area of artiffcial agents. Their strong theoretical foundations allow a wide range of applicability. They are usually consired only when talking about robotics or games, however they are used at industry for many applications that requiere applying intelligence to automation systems. Even they are starting to be included as part of generative ai solutions in order to have a mayor control of the model´s outputs and actions.

When working with agents, it is implied that they need an environment to interact with, receiving feedback, and performing actions that modify their surroundings. The research about intelligent agents its limited by the available environments that can be used to test them. Training an agent on real life scenarios is expensive and danger-

ous, while many of the digital simulators are centered in robotics or are commercial software, and there are few which focus on agent´s behaviour.

Reinforcement Learning tries to model behaviour for intelligent agents with no more data than the input that the proper agents receive from the environment. This family of algorithms are heavily based in game theory and the concept of reward. As stated in the publication "Reward is Enough" [1], the idea of receiving possitive feedback creates a strong guidance for learning complex tasks.

Supporting the idea that using the reward is enough for an agent to learn about complex task is the work presented by openai [2]. Here they show how the agents can have emergent behaviours using tools just by playing a game of hiding and chasing. It is a real possibility that in the near future we will see more works related with these emergent beahviours from agents. It has the potential of being the next big step for artifficial intelligence research, because it could allow to model solutions for really complex questions while giving some explainability by how the agent reach that solution from apparently simple rewards.

1.2 Objectives

The main porpouse behind this work is to provide a reboust, functional al complex environment for testing behavioral algorithms on intelligent agents. It has been initially though to be aproached as a reinforcement learning problem. That is why the framework is going to be developed by implementing the API standard for multi-agent reinforcement learning from PettingZoo [4].

This environment will have as goal emulating a natural ecosystem in which the intelligent agents take the rol of animals trying to survive by searching and consuming both plants and other animals. In the state of the art, some environments of this nature are mentioned, however, in general they tend to be in a way very simple depictions of a specific scenerio. For this simulator, it is intended to represent complex situations having as much similarity as possible with the real world, as this work is trying to approach the hyphothesis stated in the motivation section about the possibility of modeling really complex behaviours with simple stimuli. For this purpouse, the software is planned dynamic conditions in the environment. The climate conditions will have multiple states such as rain, wind, or sunny, that will affect the agents, which will also have variables representing their level of exhaustion, hunger, health, etc...

Under this scenario, a software will be developed as a simulator in which the visualization will be held by a 2D rendering framework called PyGame [5]. The simulator has to be compatible with the state of the art frameworks for reinforcement learning training. Apart from the visualization in 2D graphics of the environment, the simulator also will be able to log all the generated data in real time and save it for further analysis. The logging data will be composed by the state of the system, including both the individual state data for the agents, their internal data and observations, and the external state of the environment, climatic conditions and elements present.

The simulator will generate random scenarios, with the possibility of saving them in case an experiment needs to be repeated. As we are planning a random approach, we have to make some restrictions to prevent the creation of useless environments, for example the case an agent is trapped between terrain elements. With this in

mind, it could also be interesting to implement that the agents have in some degree the possibily of altering the environment elements, for example by breaking a block representing a tree.

For the scope of this work, it is spected to develop a prototype of the described simulator, and in the future add the functionalities that has not been covered, or new ones that could arise from specific experiments. This tool would be released as opensource and it is meant to be used by reinforcement learning researcher to test multiple algorithms in the agents.

It is important that the environment is compatible with the standard reinforcement learning frameworks so apart from developing the simulator, it will have to be integrated with the several libraries and tested to check its functionality. It is out of the scope of this project to find an optimal policy for the agents, however several algorithms will be tested both for assuring the integration is correct and to collect some insights about the problem.

1.3   Problem Description

The presented simulator, has the goal of creating a natural ecosystem in which the agents are the different animals present in it. This environment is designed for the agents to replicate animal behaviour in the task of survival. We are facing here a multi-agent reinforcement learning problem, because the different agents will be in a competition for resources. Some animals will be searching for plants, and other animals will be hunting for other animals in order to survive.

Here we could think of many possible behaviour of agents. The simulation, is not going to even be near of having the same number of variables that a real ecosystem will have. With the simulation we want to approach reality as much as possible just for checking how far the agents can learn to survive using reinforcement learning as cognitive engine.

For this purpose, in the simulation many aspects of the ecosystem will be simplified, because we are just focusing on behaviour. For example, in a real ecosystem, any animal would have a incredible complex system just for its vision, processing light signals, detecting color, depth, and shapes. However in the simulation we are not replicating that system but instead, using a simple way of vision just having multiple lines of vision from the animal that represent its field of view and inform the agent of the distance to an object. This simplification is used in many systems like videogames and other environments. It is neccessary because when running a complex simulation or a game, you dont have enough computing power to be dealing in real time and simultaneously with graphics,and separate computation systems for each one of the sensors of an agent [6]

To add a layer of complexity to the agents of the proposed environment, they are not only to have as state variables their external observations, but also their internal state such as the hunger, that motivates them to search for food; health and exhaustion, that can be recovered when resting, moving them to hide from other animals; and strenght, that could bring to the table cooperative behaviours for example if many weak agents collaborate to fight against a stronger agent.

For the creation of the simulator, there are some aspects that must be adressed.

One of them is the world´s physics, the agents have to be able to move in a realistic way, collide with the different terrain elements, etc... The simulation will be a 2D view from above, so the world would work as a bidimensional grid. In the future, it could be added a height dimension, but it is not going to be taken into account for the prototype that is going to be delivered in this work. Another is the agents interactions and observations in the environment. The interactions will be related to modify their internal state such as eating food for hunger, or sleeping, while the observations will be the information that they get from the environment.

As stated in this section, we are dealing with a simulation with multiple agents each one having many possible states and also exists competition between them. Finding a policy for the agents that allow them to exhibit multiple and different behaviours is out of the scope of this work. The goal of this work is providing a good foundation for the proposed simulation, with enough modularity to facilitate future implementation of features, and taking into account reinforcement learning standards for applying state of the art algorithms.

1.4 State of the art

1.4.1 Environments

1.4.2 Reinforcement Learning

1.4.3 Multi-Agents

Chapter 2

Master's Project Content

2.1 Section 1 chapter 2

2.1.1 Subsection 1 section 1 chapter 2

2.1.1.1 Subsection 1

2.1.1.2 Subsection 2

2.1.2 Subsection 2 section 1 chapter 2

2.2 Section 2 chapter 2

2.3 Section 3 chapter 2

Chapter 3

Results and conclusions

Summary of results obtained in the TFM. And personal conclusions of the student about the work done.

Bibliography

[1] Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2021). Reward is enough. Artificial Intelligence, 299, 103535. •

[2] Baker, B., Kanitscheider, I., Markov, T., Wu, Y., Powell, G., McGrew, B., & Mordatch, I. (2019). Emergent Tool Use From Multi-Agent Autocurricula. 8th International Conference on Learning Representations, ICLR 2020. •

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems, 2017-December, 5999–6009. •

[4] PettingZoo Documentation. •

[5] Pygame. •

[6] I. Millington, "AI for Games, Third Edition," AI for Games, Mar. 2019, doi: 10.1201/9781351053303. •

Annex

This is an optional chapter, and will be written according to the instructions of the Tutor.