Extended Python MARL framework - EPyMARL

A Thesis Submitted in Partial
of the Requirements for the Degree of
Master in Science

by

Girish,[*]

Supervised by Professor

Submitted April 15, 2024

[*] Student ID: Numeric ID 1, Email: `name1@nottingham.ac.uk`

Abstract

Multi-agent Reinforcement Learning (MARL) is emerging area within the Artificial Intelligence research which is dealing with the development of policies for the agents to learn the optimal behavior via the interactions with the surrounding environment and across agents. In contrast to single-agent scenarios, by dealing with the complexities engendered by the multiple agents, their interdependence must be handled, where the actions of each agent can manage the instantaneous state of the world and other agents' utility. The kind of evolving dimensions arise from the simple fact that the way the environment will appear to any particular single agent is not stable and changes as other agents figure out their own ways to survive.

Contents

Conventions and Notation

List of Figures

List of Tables

Acknowledgements

Preface

1. Introduction

In recent years, the Machine Learning, especially Deep Learning, has significantly progressed. This development has resulted in immense change in the perception of cooperation within multi-agent platforms. It gives rise to many dilemmas regarding these machines' ability to develop their very own communication protocols that can be used to coordinate their activities. Also, there is an emerging interest in the scope of deep learning in making agents more intelligent. Similarly, their endeavor to learn the language and the outcome of those attempts makes interesting observations. The dynamic ecosystem provides a basis for a richer discussion rife with possibilities and challenges of machine learning that helps create effective communication among the agents, an important avenue to consider in the wider debate on artificial intelligence.

Multiagent learning MAL is specially developed to see how the various agents in multi-agent models can act to ensure good coordination and how the model can explain the interaction of agents in more than one system [1]. It is becoming widespread in robotics, and taking into account its position in how things are distributed or managed, resources management and automated trading is an outstanding advantages. By adopting this approach, we open up routes that lead us to new understandings of how complex problems can be solved under the coordinated actions of the agents. The key problem in multi-agent learning (MAL) is that each agent is engaged in a game where one's own decision-making and volition depend on the adaptive behaviors of his/her peers. These indoor environments cause similar difficulties also, with the population constantly adapting to each other's style, making it even harder to maintain stable education goals. Basically, achieving this objective often becomes a game of a target that is always on the move. As a matter of fact, machine agents within MAL are faced with problems similar to those of human beings when it comes to the right use of information, as they have to predict the actions of other agents without full knowledge. Term of this is to include uncertainty in the process because the agents may act sensitive to the fact that they will need to deal with this complexity [2]. One of the key areas of investigation in such research is how different agents work together to achieve coordination. This interest in coordination starts from the knowledge that one agent's action can impact the outcome of the whole system depending on what the other agents do simultaneously. Hence, to ensure that the aims of all involved agents are validated; their activities should either be aligned or happening in a disciplined manner.

The concept of multi-agent learning for coordination has been studied by a lot of research efforts that utilize the idea of autonomous chemical systems as well as specific focus and assumptions. Such as, Techniques to multi-agent learning that have hierarchical nature were invented to utilize the interdependence of agents. This method enhance the speed of learning skills which work together by the agents in cooperative multi-agent systems. [3]. Here, interdependence, communication and cooperation among agents are explained through the division of the main task into a multitude of responsibilities whose achievement is ensured by particular agents. Besides this, multi-agents notion can be explored through the multiple learning strategies using connectivity graphs [4] and distributed value system [5], which should investigated the cooperation strategy for the local interactions among agents across networks. One of the main aspects of these strategies is MAL which implies the performance of MAL as a team with coordinated action. Coordinated multi-agent learning (MAL) strategies, usually based on the concrete assumption of independence, uses task partitioning or a predefined interaction system. Yet, there are some situations in which autonomous agents need to search for cooperative patterns without having prior information about their independence but only through interaction. Some MAL methods make it possible for agents to identify interdependencies for concerted work [6], but the assumption of the optimal policy of the agents or the complete visibility of the collective state and action significantly restricts their use in practical MAL solutions in multi-agent systems (MASs).

1.1. Litrature Review

Recent developments in deep multi-agent reinforcement learning (RL), however, have been a rapid field of research. Within this sector, some of the most complicated problem categories, namely, partially observable, cooperative, multi-agent learning, become manifest. The agents in this scenario must understand how to coherently take individualized actions without ever seeing the other group members. This is a powerful research focus as it applies to many viable systems and can be assessed easily compared to the general sum problems. Extensive research has been carried out with regard to the following subjects, such as value decomposition[7], communication, agent and parameter modeling, amongst others. On the other hand, the weakness of two methods is that they can't distinguish between different Nash equilibria, therefore most often these methods converge to a suboptimal solution [8].

The idea behind the Pareto-dominated equilibria when dual-agent reinforcement algorithms converge to similar state actions, for example, QMIX [7], is that exploration strategies like -greedy or soft policies are used by these strategies to explore state-action space. In particular, a policy gradient learner is responsible for choosing the next action depending on the modeling and learning outcomes. At the beginning of training, the stochastic models are likely to share the probability of choices, relying on the entropy item that drives the exploration. Coming back to the first parts of the training, where the algorithms that omit the synergies in actions may go for the individual actions that are somehow independent and do not really need to be coordinated and thus they can get higher expected returns. Conventionally, ALE (Arcade Learning Environment) and MuJoCo (Multi-Jointed Dynamics in Contact) platforms have permitted single-agent RL to be applied in more complex domains such as grid worlds. Nonetheless, it is still a challenge for cooperative multi-agent RL to provide common standard tasks as a backdrop for research, which would help to recognize the actual progress in this field, because of the widespread use of unique and laborious tasks. In our article, make a proposal to use the Multi-Agent Challenge (MAC) of SMAC as a benchmarking case to address this shortcoming [9].

Frameworks including Pymarl [9], Rllib [10] and Gym from OpenAI [11] which provide integrated environments and tools for MARL studies, are the clear examples. PyMARL, for the case in point, is an example of that a framework that presents an architecture that designs an environment for solving a multiple challenges of researching experiments with the new algorithms and further development of the new strategies. Currently, RLlib is done to introduce RL to reinforcement learning, focusing on multi-agents Reinforcement Learning frameworks, allowing large-scale simulations, and multi-agent environments for complex scenarios of coordination, competition, and mixed scenarios. AI Open Gym is a comprehensive tool kit for single-agent environments as it further extends it to multi-agent frameworks with the help of the uniform Application Programming Interface (API) for those who want to conduct the MARL tests. These python-based MARL frameworks, which are extremely important in the development of the field, offer researchers resourceful infrastructure to pursue new approaches to multi-agent learning and develop better algorithms, test if environments are the most appropriate for the purpose. The EPyMARL (Extented Python Educanth-Multi-Agent Reinforcement Learning) framework, an improved version of its predecessor that was known as

PyMARL, includes the choice of not sharing parameters, a feature that contrasts with the restriction put by the previous version which required parameter sharing only.

It increases the possibility of use by adding extra steps as implementations suggestion such as hard/soft updates, reward standardization, and so on, and the flexibility of algorithm configuration proceedings is also extended. EPyMARL provides consistent implementation of the algorithms among various algorithms basis on which its adoption as a framework is recommended. It paves the way for research and experiments with MARL that were unthinkable before, enabling the analysis of intricate agent dynamics in a level of detail that was impossible to obtain in the past.[12]. The algorithm we add in this model framework includes algorithms (IA2C [13], IPPO [14], MADDPG [15], MAA2C [16], and MAPPO [17]). We include implicit actor-critic approach (A2C) methods which also rely on the reinforced agent observability increasing as well as reducing compilation complexity for every agent exhibiting a superior performance on matters of optimality, robustness and efficiency in the simulations of both synthetic and real-world traffic scenarios. As compared to other joint learning methods, the independent PPO wherein each agents has local value function, performs equally well or even outperforms in cooperative multiagent reinforcement learning with respect to the “centralized training with decentralized execution” framework, which can be implemented on the SMAC benchmark suite and the capacity of catch up with joint learning are less affected by the hyper-parameters estimation.

One of the most interesting findings of this study is its robustness to some kinds of non-stationarity in the environment. MADDPG, which is an extension of DDPG to multi-agent environments, has a centralized critic that learns from all the agents observations and actions in order to apply decentralized policies without dealing with differentiable environment model or a particular communication structure. This approach supports the cooperative, the competitive and the mixed scenarios by training actors together and after that, they act independently based on their knowledge which makes them applicable to various scenarios. MAA2C, a multi-agent advantage actor-critic algorithm, is an extension of A2C (Advantage Actor-Critic) framework that is designed for multi-agent cases. It enables the training of individual as well as joint policies using the same information and common value function optimization that further enable the multiagents to function efficiently in complicated cases and make the right decision.

MARL as an efficient way to coordinate the online scheduling for production in the cellular-based manufacturing environments and the maintenance of high-availability of systems under those conditions of unpredictable machine breakdowns. Taking the centralized learning and decentralized path and the agent goal (made up for OpenAI’s Proximal Policy Optimization algorithm), we devise a ”Multi Agent Proximal Policy Optimization” (MAPPO) algorithm that aims to minimize resource usage and adapt dynamically to changing shop floor situations [**?**].

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

1.1.1. Subsubsection 2.1

2. Multi-Robot Warehouse (RWARE) environment

2.1. Environment Description

RWARE Warehouse is a highly changing and dynamic running simulation model that was developed in order to conduct research, study, as well as build intelligent multi-agent systems for reinforcement learning. The machine takes meticulous care of the routines copying a factory of age of autonomous robots who are assigned to move and transfer goods as needed. The next section of this article is specifically aimed at describing the RWARE environment in details. The latter feature focuses on the functionality, customization possibilities, and applications in the industry.

2.1.1. Simulation of Warehouse Operations

Basically, RWARE replicates a work area with concurrent activities for a number of the robots to work together at the same time. The robots concern here are designed to execute tasks that are similar to their real-world counterparts, and the main focus of the activity is the shelves that need to be retrieved and delivered laden with different stock. In the main workflow, a shelf will be picked up by a robot and will be moved to a particular work platform. Human workers or substitute agents have the capability of approaching the shelf to explore the materials inside. The robots interact with aisles and pick up all the items to return the shelve to their original spot or to a vacant space midst the warehouse.

Being similar to real ASRS all over the world, this simulator is based also on today's auto storage and retrieval systems that are found at distribution centers and logistics facilities. The machine work in such places is for purposes of increase in accuracy, efficiency and also reduction in human labor and in the end, it presents improvement in inventory management accuracy.

2.1.2. Configurability and Customization

One of the main characteristics of the RWARE environment is the possibility to give the simulation a high degree of configurability. In that, researchers can adapt the simulation according to the selected study aims. This plays the key role in the execution of a well planned ways to figure out controlled experiments in reinforcement learning and multi-agent systems. undefined

- **Size of the Warehouse:** The complexity of the environment can be adjusted by selecting the desired size of the warehouse. Larger warehouses present more challenging scenarios with longer travel distances and more complex navigation routes.
- **Number of Agents:** Users can specify the number of robotic agents operating within the environment. Managing a larger number of agents introduces complexities such as the need for coordination and collision avoidance, which are critical aspects of automation in dynamic settings.
- **Communication Capabilities:** The simulation can be configured to enable or disable communication between agents. This modification allows researchers to explore the impact of inter-agent communication on collective task performance and efficiency.

- **Reward Settings:** RWARE allows for the selection between cooperative and competitive reward structures. In cooperative settings, agents are evaluated based on collective performance, encouraging teamwork and shared objectives. In contrast, competitive settings reward agents based on their individual achievements, fostering competitiveness or independent action.

In the context of the Multi-agent Reinforcement Learning system (RWARE) in virtual warehousing environment, grasping the basis of action space design and the observation space understanding is the key point to the construction of efficient strategies. These principles form the basis for the tests that attempt to closely mimic real dynamics but also offer a relatively steady and the controlled environment which is necessary for experimentation.

2.1.3. Action Space

The action field in RWARE determines the options of actions that all robotic agents in the environment can perform. This space is discrete and includes four primary actions:This space is discrete and includes four primary actions:

- **Turn Left:** The robot undergoes rotation to its left with anticipation to experience tilting the opposite direction without making any movement.
- **Turn Right:** The robot turns to its right by rotating counterclockwise, just like humans. Its orientation does not change; instead, it simply orients itself in a different way.
- **Forward:** The simpled robot moves 1 unit of the grid on the line basis of its basic orientation. The maneuvering needs to be precautious, otherwise there can be a chance of colliding.
- **Load/Unload Shelf:** Built in to this action is the ability of a robot to pick up an empty shelf if it doesn't have any where or set down a shelf that it is carrying In order to do this action, the robot should be installed on the shelf down in the markedly specified locations for loading/unloading.

These movements together constitute primitive functions that are to be well executed by the robots, hence strategic thinking is required in order to optimize routes as the robots operate within the space of the warehouse. The structure of the actuator room has a specific focus on the accuracy of actions and the interaction between man and a complex environment which, in its turn, are the fundamental pillars of a precise routing and goal achievement in a warehouse full of objects.

2.1.4. Observation Space

The RWARE observation area defines what each robot can detect and consequently can use at any moment in the run. undefined

- **3x3 Grid:** Every robot is able to see a 3x3 region which is the area surrounding its center. The dimension of this area can vary from large enough to smaller in accordance with any experimental requirement. undefined
- **Self-information:** The robot has vision and webcam with which it can see its own location and orientation, as well as whether it is carrying a shelf.
- **Other Robots:** Grid positions and orientations of other robots are visible all over the grid. These data are fundamental things for avoiding conflicts and cooperation on movement.

- **Shelves:** The robots locate shelf positions and check whether these positions are part of the inventory area that indicates that items on this position need to be delivered/picked up.

A part of a space where the territory is not fully observable requires robots to make decisions as it is similar to the real world, where complete saturation with the information is often impossible. This aspect is testing the algorithms to acquire effective strategies that can deal with uncertainties and incompleteness of data.

2.1.5. Reward Structure

The RWARE system of rewards centralizes the operation of shelf retrieval and distribution—the most fundamental areas of the simulated environment. The rewards are issued as follows: The rewards are issued as follows:

- **Request and Delivery:** At any given time, particular hangers are requested, numbered with the indicated R. Shelves labeled ”work the floor,” on the other hand, are marked as being critical to continually supporting the overall advancement of the warehouse, for example, through fulfilling orders. When a reward is added to the shelf as a result of the system successfully placing it into a goal location or not, that reward is registered at that location. The standard reward being levied to a correct and successful delivery is 1 point for each shelf.
- **Continuous Requests:** Maintaining a dynamic environment and involving the user will be provided through the replenishment of a required box as soon as the shelf is successfully delivered into the queue of needs. Now the process is performed one a new rack is selected at random from the list of the products in stock and added to the queue of all orders. This mechanism ensures the poise of the system, creating a pre-set count of ongoing requests that keeps the agents continuously occupied.

Challenges in Reward Distribution

The reward structure in RWARE, while straightforward in its basic form, introduces several significant challenges related to the behavior optimization of the agents:The reward structure in RWARE, while straightforward n its basic form, introduces several significant challenges related to the behavior optimization of the agents:

- **Sparse Rewards:** One of the main challenges that are common in environments of the same kind is the scarcity reward signal for which. Because they are getting rewards only for the last trial when they follow the given shelf precisely, rather than for the series of events involved as finding shelves, avoid collisions, or plot course, agents have no immediate feedback. Such a stinginess forces the agents to spend much time on searching for correct and unsuccessful strategies while they perform a large number of the moves that do not promise rewards.
- **Finding Empty Locations:** The procedure is repeated until the destination of the shelf, delivering it, an agent needs an empty location to return the shelf it possibly carried before or to pick up the next requested item it must carry. This job involves extra navigating and making choices amid impossibility, since leave locations might not be immediately noticeable or reachable, and the environment may come across changes with other agents sitting shelves.

3. Data Analysis & Interpretation

3.1. Parameter selection

3.1.1. Parameter selection for QMIX

During the QMIX learning process to the multi-agent RWARE game, the parameters selection is very important for steering through the challenge of sparse rewards and the necessity for agents collaboration. We use an epsilon greedy action selection strategy for this. In the beginning the agents have to do a thorough exploration of the environment. This exploration is important because the rewards are not immediately available. The gradual decrease in epsilon after 50000 steps prevents the agents from premature convergence to a particular policy. Here, gamma at 0.99 is set to give more weight to future rewards so the agents will consider the long-term consequences of their actions which reflects reality where rewards are not frequent and agents have to do a series of correct actions to receive a reward. Double Q-learning as a learning process is employed to stabilize it by reducing the overestimation of action values mostly in the cases when agents have to learn from a small positive feedback. However, the learning rate is picked at only 0.0005 to allow for gradual learning. This will avoid abrupt policy changes, which could result from larger learning rates that can be destructive in a complex environment, where a small policy modification can yield significant result. Gradient norm clipping is done with a value of 10 in order to prevent the gradients from exploding during backpropagation and therefore to preserve the numerical stability of the learning process as a whole.

3.2. Observation QMIX

In RWARE, the QMIX algorithm has been brought into use to evaluate its performance in regards to complex goal achieving characterized by sparse rewards and cooperation of all agents. The warehouse navigation was the main goal of the algorithm. Hence, it has to make agents move, standing by their resource collecting and delivering functions adequately. Observational data suggest of absence of the listings of profit for training. The corresponding performance reports are constantly displaying the results with zero.

The information agrees with the benchmark paper that mentions the low level functionalities of value-based mutual multiple agent reinforcement learning like QMIX in the RWARE setting. This postulate might be rooted in the tricky surface of Reward Shopping without Accumulation, which undoubtedly is a complex field that an algorithm struggles to go through [18]. Also it can be referred from the paper that QMIX and all other value-based MARL algorithms we evaluated (IDQN, VDN) perform very poorly in the RWARE environment. They rarely receive any positive rewards (since it is very difficult to receive positive rewards in RWARE due to its sparse rewards).

In the logs, we could observe the initialization QMIX algorithm along with parameter setting for the action selector, epsilon anneal time, and gamma. As a part of the program, the model was initialized using training and then validated against a set of learned weights. Obtained data consists of several episodes of returns, which is validated by zero being present throughout the estimates. Moreover, these results indicate the need for changes in the exploration strategies and value function partition approach applied by the QMIX algorithm in the conditions of the RWARE environment which have the sparse rewards. And a scenario like this requires a convoluted cooperation, which is not quite covered by the separation model predicted by QMIX.

4. Discussion

Appendices

A Example 1

B Example 2

Bibliography

[1] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

[2] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[3] Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13(1-2):41–77, 2003.

[4] Jelle R Kok, Eter Jan Hoen, Bram Bakker, and Nikos Vlassis. Utile coordination: Learning interdependencies among cooperative agents. In *EEE Symp. on Computational Intelligence and Games, Colchester, Essex*, pages 29–36, 2005.

[5] Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

[6] Jelle R Kok and Nikos Vlassis. Sparse cooperative q-learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 61, 2004.

[7] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 21(1):7234–7284, 2020.

[8] Filippos Christianos, Lukas Schäfer, and Stefano Albrecht. Shared experience actor-critic for multi-agent reinforcement learning. *Advances in neural information processing systems*, 33:10707–10717, 2020.

[9] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

[10] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph E. Gonzalez, Michael I. Jordan, and Ion Stoica. RLlib: Abstractions for distributed reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2018.

[11] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[12] Filippos Christianos, Georgios Papoudakis, and Stefano V Albrecht. Pareto actor-critic for equilibrium selection in multi-agent reinforcement learning. *arXiv preprint arXiv:2209.14344*, 2022.

[13] Tianshu Chu, Jie Wang, Lara Codecà, and Zhaojian Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1086–1095, 2019.

[14] Christian Schroeder de Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

[15] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

[16] Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International conference on machine learning*, pages 2961–2970. PMLR, 2019.

[17] Oliver Lohse, Noah Pütz, and Korbinian Hörmann. Implementing an online scheduling approach for production with multi agent proximal policy optimization (mappo). In *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems: IFIP WG 5.7 International Conference, APMS 2021, Nantes, France, September 5–9, 2021, Proceedings, Part V*, pages 586–595. Springer, 2021.

[18] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. *arXiv preprint arXiv:2006.07869*, 2020.