Developing Flexible Reinforcement Learning Environments to Understand Their Effect on the Trained Agent

Tomos Ody (ODY14527411)
Department of Computer Science, University of Lincoln

Extension Authorisation Code: 2cb87a6b

I. Introduction

In this project the aim is to develop a framework through which to design environments used to train reinforcement learning (RL) agents in a fundamental and simplistic manner. This will be achieved by studying the academic literature pertaining to RL to understand its strengths and weakness. Also to understand existing applications to assist the development of the proposed framework. These ideas will then be developed and tested to research the correct framework methodology.

This literature review will explore and describe the academic bases of this project. Investigating how best to achieve the aims of the project using proper academically based methodologies. Tools to achieve the project aim will also be investigated to better understand the requirements of this scope. Frameworks through which to design RL solutions will be studied to better inform the aim of the project as it is developed. Common applications of RL will be investigated to examine the scope of RL problems and see when and how RL can be applied and to what effect.

The review will identify the areas of research to be conducted to achieve the aims and objectives of the project: Outlining strong academic bases for the methodologies and techniques to be used in RL tasks; A comprehensive methodology outline for designing RL frameworks must be developed from relative academic resources; How conducted research and development should contribute to this framework design should be outlined.

II. Academic basis

The standard RL model consists of an agent interacting with an environment via perception and action. At each step of the training process the agent receives some state information. Based on this information the agent will then take an action based in it's observations of the environment. Depending on the action taken a scalar reinforcement signal is sent to the agent as a reflection of success. The agent aims to maximise the long-run sum of the reinforcement signal. This process leads to the agent choosing the rewarded actions leading to the development of the desired behaviour in the agent. As outlined in [1], a survey of RL field from 1996. This basic outline is corroborated by [2], a survey of multiagent RL from 2008 and [3], a survey of RL in robotics from 2013. This basic conception of RL models is consistent across all literature and appears in papers up to the current day.

This learning method does not fit neatly of the two main machine learning paradigms of supervised and unsupervised learning. Instead of being directly trained by correct examples or inferring patterns from the data, a RL agent learns from it's own interactions with the environment quantified by a scalar reward described in the 2018 book [4], cited in many RL works. This agent driven learning style leads to a trade-off between exploration and exploitation [1]–[4]. Due to the need for the agent to explore the environment as well as perform actions to maximise the positive reward (exploitation). The trade-off stems from the incompatibility of these two behaviours which are both required for an effective agent. Since the agent must explore the environment to understand the positive and negative options possible yet avoid negative actions to succeed [1]. For most RL problems Markov Decision Processes (MDP) are applied as the methodology for modelling decisions to calculate action rewards accounting for the exploration-exploitation trade-off [3]. MDPs work by considering rewards temporally, meaning that a series of actions can be attributed to the delayed reward function [1]–[3]. This is in contrast to more simple greedy strategies which reward the agent directly for each individual action. However these methods can fall victim to unlucky reward sampling early in the training process [1]. A useful framing problem for understanding the balancing of exploration and exploitation is the K-Armed Bandit problem. This is used in most survey papers [1]–[3] to explore the performance of RL methods in the scope of exploration vs exploitation. The more recent papers [2], [3] analysing these methods to balance exploration and exploitation tend towards the more dynamic methods like MDP as opposed to the more simplistic greedy strategies mentioned in [1]. The other simplistic methods suggested by [1] are not mentioned in the more recent surveys of RL [2], [3], only in [4] are these methods mentioned as introductory methods to RL. The literature seems to be in agreement on many of the details of RL methods, however many methods mentioned in the older [1] seem to be less widely used in the surveys written in recent years [2], [3].

Most of the methods discussed in RL are very old, for example MDPs were originally proposed by [5] in 1958 which is cited in each of the RL surveys [1]–[3] as well as the seminal [4]. The vast majority of improvements made to RL methods are minor improvements to MDPs.

III. Scope of reinforcement learning

Reinforcement learning is a very adaptable machine learning approach due to the wide range of approaches and methodologies that can be used to form a solution. However [1] concludes that RL performs well on small problems but poorly on larger problems due to the efficiency of generalising the problems. This point is reaffirmed by [6], adding that an efficient generalisation of a complex problem requires expert domain knowledge. This suggests that whilst versatile RL is only generally suited to simple problems.

One of the primary areas of RL research is robotics since the domain of controlling a robot fits neatly into a RL problem as discussed in[3], whereas other machine learning disciplines are very difficult to utilise for robotic control systems. This view of RL in robotics is shared by [7]. Many other hurdles are introduced by using reinforcement learning in a real context such as observation inaccuracies and the expense of hardware [3], however this does not pertain to this project. Robotic RL agents instead present a strong base of real agent-environment interactions [3].

Up till this point single-agent systems have been discussed, however Multi-Agent Reinforcement Learning (MARL) is another large body of work with multiple agents training in the same environment. This domain is similar to single-agent RL but multiple agents open up a new set of challenges and advantages. As described in [2] the main challenge faced is an exacerbation of the exploration-exploitation trade-off due to inter-agent interactions adding a new level of complexity to the issue. Multiple agents can also be turned into an advantage in MARL by allowing agents to teach their experience to other agents. Additionally more advanced reward functions can encourage collaborative actions between agents. One clear advantage over other RL domains is that MARL can benefit from parallel computing increasing the computational efficiency.

Most RL environments could be described as games, unsurprisingly the most active domain of RL is playing video games. Video games present a host of environments easily adaptable to RL applications, however RLs low complexity tolerance still takes effect. This leads to either very simplistic games being used to train complete agents to perform well such as in [8] or more high level applications like [9] where a RL agent controls the strategies of an artificial intelligence agent designed to play the game. These applications both use MDPs with Q-Leaning [10] to consider the higher level temporal planning required for playing video games.

A criticism is raised in [11], a critical survey of MARL using MDPs, as to the definition of many RL solutions in that they are often performed from an ill defined domain basis in how learning best occurs. It outlines a four stage method of properly defining a RL problem, describing how too many RL solutions work off previous bases without exploring their applicability to the problem being addressed. This demonstrated by the contrast between [9] where little academically supported domain knowledge is presented versus [12] where the domain knowledge is comprehensively explored before designing the RL solution. This suggests that certain areas of RL problems are ill guided. This should remain pertinent through this research project and be addressed where applicable.

IV. Reinforcement learning techniques

Whilst not encompassing all of RL, MDPs represent the vast majority of modern machine learning solutions. This is due to the simplicity with which a complex problem can be represented, taking into account time and previous actions as outlined in [13]. As a result of this focus there is a wealth of different MDP strategies in the literature to draw from. An MDP is a framework for modelling actions with controllable yet undetermined outcomes to be optimised [5]. Utilising an MDP a complex problem can be abstracted down to a comparably computationally simple problem, this however requires much domain specific knowledge to be properly exercised [11].

MDPs are a mathematical framework through which to quantify and rationalise decisions made in a Stochastic Ggame (SG) [5]. SGs are step by step games which quantify an environment probabilistically [14], this hugely simplifies reward and action computation in RL problems [2].

In both [8], [9] Q-Learning is used, it is an approach to MDPs which allows a RL agent to explore an environment to experience the various reward states associated with each action. The Q-Learning task will use these immediate rewards to plan temporally how to maximise the reward overall as described in [10].

Much of the research pertaining to RL problems seems much too condensed, with single authors appearing in multiple papers such as Barto in [13], [15] as well as the book [4]. Kaelbling also appears in both [1], [7]. Other concentrations like this appear all across the RL literature which makes me sceptical of the academic basis of many of these methodologies. I feel this particular area requires additional and more broad research.

V. Reinforcement learning validation methods

Validation methods for RL solutions vary wildly since each application has a differing goal state. Unlike supervised and unsupervised learning success is entirely subject to the task at hand [1]–[3]. A broader search of specific RL applications will be required to effectively devise a methodology for effective validation of RL implementations. Although it is highlighted in [11] that domain specific knowledge should be used to devise a testing methodology.

References

[1] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, 1996. DOI: 10.1613/jair.301.

[2] L. Busoniu, R. Babuska, and B. De Schutter, “A comprehensive survey of multiagent reinforcement learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008. DOI: 10.1109/tsmcc.2007.913919.

[3] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: A survey,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013. DOI: 10.1177/0278364913495721.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, Second edition. MIT Press, 2018.

[5] R. Bellman, “Dynamic programming and stochastic control processes,” *Information and Control*, vol. 1, no. 3, pp. 228–239, 1958. DOI: 10.1016/s0019-9958(58)80003-0.

[6] C. Wirth, R. Akrour, G. Neumann, and J. Fürnkranz, “A survey of preference-based reinforcement learning methods,” *Journal of Machine Learning Research*, vol. 18, 136:1–136:46, 2017.

[7] W. Smart and L. Pack Kaelbling, “Effective reinforcement learning for mobile robots,” *Proceedings 2002 IEEE International Conference on Robotics and Automation*, 2002. DOI: 10.1109/robot.2002.1014237.

[8] M. G. Bellemare, J. Veness, and M. H. Bowling, “Investigating contingency awareness using atari 2600 games,” in *AAAI*, 2012.

[9] C. Amato and G. Shani, “High-level reinforcement learning in strategy games,” in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1, Toronto, Canada, 2010, pp. 75–82, ISBN: 978-0-9826571-1-9.

[10] C. J. C. H. Watkins and P. Dayan, “Q-learning,” *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992, ISSN: 1573-0565. DOI: 10.1007/BF00992698.

[11] Y. Shoham, R. Powers, and T. Grenager, “Multi-agent reinforcement learning: A critical survey,” Jun. 2003.

[12] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, “Autonomous inverted helicopter flight via reinforcement learning,” in *Experimental Robotics IX*, M. H. Ang and O. Khatib, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 363–372, ISBN: 978-3-540-33014-1.

[13] A. G. Barto and S. Mahadevan, “Recent advances in hierarchical reinforcement learning,” *Discrete Event Dynamic Systems*, vol. 13, no. 4, pp. 341–379, 2003. DOI: 10.1023/A:1025696116075.

[14] L. S. Shapley, “Stochastic games,” *Proceedings of the National Academy of Sciences*, vol. 39, no. 10, pp. 1095–1100, 1953. DOI: 10.1073/pnas.39.10.1095.

[15] A. G. Barto and S. Mahadevan, *Discrete Event Dynamic Systems*, vol. 13, no. 4, pp. 341–379, 2003. DOI: 10.1023/a:1025696116075.

DEVELOPING FLEXIBLE REINFORCEMENT LEARNING ENVIRONMENTS TO UNDERSTAND THEIR EFFECT ON THE TRAINED AGENT

Tomos Ody (ODY14527411)

Department of Computer Science University of Lincoln

Abstract

Reinforcement learning represents a versatile framework through which to frame a problem with success and failure states defined by the creator. The agent will use whatever tools at it's disposal to achieve these success states and avoid the failure states. This means that with relatively few agent directions very complex behaviours can arise. The aim of this project is to understand how the environments these agents are trained in effects their performance, better understanding the complex behaviours that can arise from it. Examining multiple implementations of reinforcement learning (RL) through this lens to propose a method through which to better design these environments.

Introduction

This project aims to explore the common problem domains of RL implementations and explore environmental characteristics of their training process. Then to design an implementation of these that are adaptable to explore how changes to this environment effect the agents performance. The agents themselves should be simple so to allow their implementation in environments to be flexible; avoiding the complexity of an agent causing it to integrate poorly with changes in the training environment. Note that this does not mean poor agent performance but weak agent-environment interactions. For this reason the implementations must be observable whilst training, to spot these undesirable interactions, resulting in faster iterations on the environment.

These scalable environments and their simplistic agents will then be put through an investigation phase where the reaction of the agent against changes to the training environment will be recorded and analysed. The best performing agents will then be tested to understand their performance characteristics. These performance results will then be used to understand how the environment shaped the performance of the agent. The agents performance will then be compared to similar results from the literature to understand the effect of environmental changes on the learning process.

These results will then be collated to attempt to understand the affects of the environments on agent performance. Presenting a methodology through which to design an environment for simplistic RL tasks.

A project similar to this would be OpenAI's Gym [1] a framework through which to standardise RL environments. Allowing for meaningful testing of multiple algorithms and agent designs using the same environment. This provides a wealth of examples of simplistic RL tasks with a wide test bed to explore in a meaningful fashion.

Aims

In this project the aim is to explore the common applications of simplistic RL problems, to understand how training environments effect agent performance. Common RL applications will be analysed for similarity and a generalised implementation will be created and tested against the results of similar implementations from the academic literature. This testing will assess the designed environment-agent system's performance.

The systems that perform well, will then be altered by manipulating the environment to observe the change in performance and behaviour of the agent. From these investigations environmental effects on an agent a range of hypotheses will be outlined and tested using these developed systems with respect to the real world results from the literature.

From these observations and hypotheses, a framework for designing RL environment-agent systems will be proposed. This framework should allow for framing of simplistic RL tasks and provide an intuitive methodology for designing environmental and agent features effectively. These systems should be extensible and versatile so as to be more generally applicable.

Using this designed framework, suitable adaptable problem domains will be developed into RL solutions. These solutions will then be tested using similar methodologies to the testing of existing implementations to analyse the effectiveness of the proposed framework.

Finally the proposed framework will be revised from the information gathered over each testing phase. This revised framework will evaluated with analysis of the specific strengths and weaknesses for respective problem domains.

Objectives

- Perform a wide search of the RL academic literature to collate a strong bed of RL problem domains. Examining the methodologies used and collecting similar applications into groupings.
- Develop simplistic, scalable and versatile solutions to the problem domains analysed from the literature. These systems will then be tested against related results from the literature to verify the system works as intended.
- Observe the affect of changes to the environment on the agent performance and behaviour. Develop a testing methodology to best explore these effects in a manner which is inter-comparable with the other developed systems.
- Design a framework through which to design these scalable solutions from problem domain definition to testing. Developing systems using this framework to explore the effectiveness on different problem domains.

Research Methods

In this project a wide range of RL problems will me researched to understand the scope of RL problems as a whole. By finding commonalities between various RL problem domains they can be grouped, additionally particular areas of interest can be highlighted.

Using this research base a range of simple RL environments will be developed with particular care taken to investigate the domain knowledge of each problem to more effectively design each implementation as outlined in [2]. Thought should also be given to validation methods along similar lines.

These RL problems will be developed as Markov Decision Processes [3] using Python so as to keep the basis highly flexible to allow for testing of how an environment can bound agent performance in certain problem domains. Different methodologies for managing Markov Decision Processes will be investigated and implemented such as Q-Learning [4]. The developed RL problems should be tested in a similar fashion to the original basis of the implementation to validate their functionality as a RL model. These will then be tested with varying environment conditions to observe how environmental changes effect agent performance.

Using the domain based validation methods researched the various environmental states should be tested to assess their performance and identify how the environment effects the agent performance. This data should be coallated and then assessed as a whole to identify environmental effects upon the agents.

Using the data gathered from the RL implementations developed and research made into other RL applications and their environmental interactions a research plan should be created for developing the framework through which to design an effective and adaptable reinforcement learning environment.

From the conducted research with basis in the results a framework for designing RL environments will be proposed. A number of test cases for the framework will then be investigated to understand the applicability of the framework and to improve upon it.

This framework will then be presented alongside academic backing, a thorough report of the development process and results from the testing of the framework to demonstrate it's potential application.

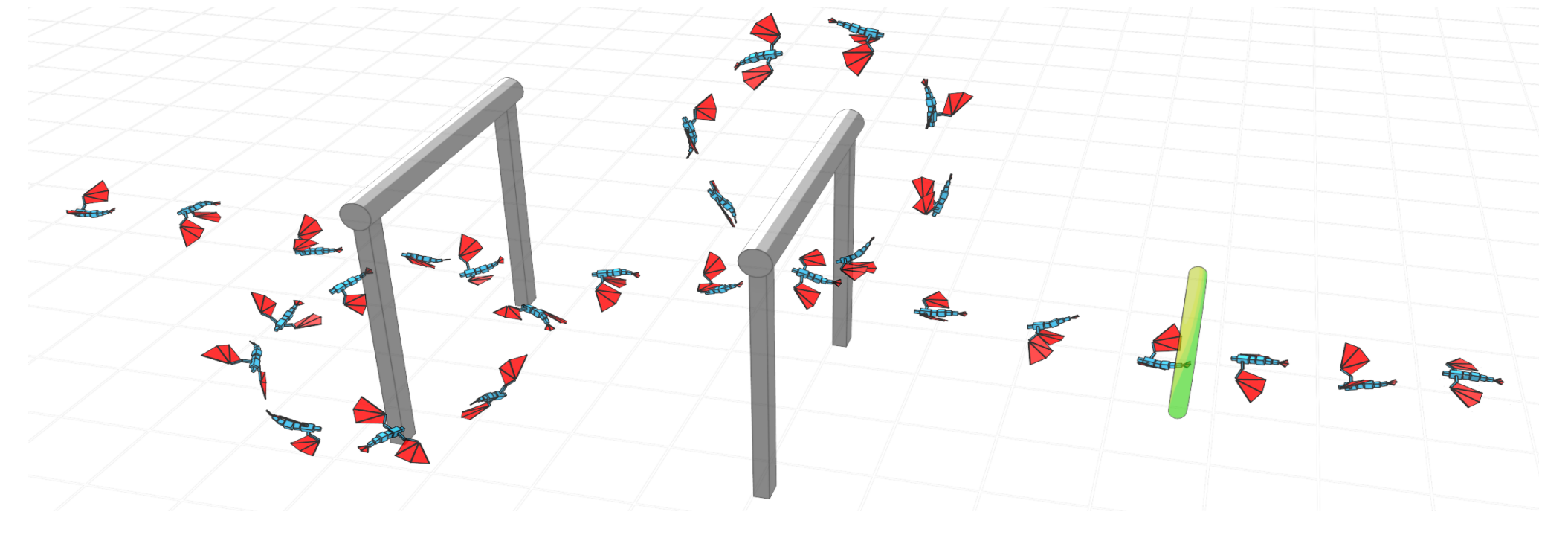

Figure 1:A RL agent trained to perform aerobatic manoeuvres whilst preserving momentum, an example of environmental constraints effecting the trained agent [5]

Research Plan

The project is divided into three main phases: research and development, collating the literature to understand the various applications of RL and developing and assessing developed solutions; hypothesising and testing, exploring the developed solutions to understand the environmental effects upon the agent; framework design, propose a framework through which simple problem domains can be effectively turned into RL problems with versatile solutions and test the frameworks suitability.

Each individual phase of this project must have progress before next can be begun, this however does not mean that the work must be done in one drive. Due to the rather modular nature of the development and testing cycles the first and second phases can be handled fairly independently from one another. This means that the first and second phase will be repeated multiple times before the third phase begins. However this process should be skewed to allow for a more considered workflow with more research being done in the initial stages before developing a more rigorous testing methodology to complete phase two. More research will be required for the third phase although the research from the first and second should strongly contribute to this. The development of the framework should be completed in a single drive to ensure that it is strongly considered. The validation part of phase three once again splits into a modular workflow as each test base developed using the framework requires it's own research, development and testing. Once these results have been collated the framework will be revised and analysed to demonstrate its strengths and weaknesses.

Finally once the research is completed the notes taken throughout will be produced into a document to explain the research, findings and implementations to present the framework in a meaningful manner with a strong academic basis.

References

[1] OpenAI. (2016), Openai gym, [Online]. Available: https://gym.openai.com/ (visited on 04/15/2019).

[2] Y. Shoham, R. Powers, and T. Grenager, "Multi-agent reinforcement learning: A critical survey,", Jun. 2003.

[3] R. Bellman, "Dynamic programming and stochastic control processes," *Information and Control*, vol. 1, no. 3, pp. 228–239, 1958. DOI: 10.1016/s0019-9958(58)80003-0.

[4] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992, ISSN: 1573-0565. DOI: 10.1007/BF00992698.

[5] J. Won, J. Park, and J. Lee, "Aerobatics control of flying creatures via self-regulated learning," *ACM Trans. Graph.*, vol. 37, no. 6, 181:1–181:10, 2018, ISSN: 0730-0301. DOI: 10.1145/3272127.3275023. [Online]. Available: http://doi.acm.org/10.1145/3272127.3275023.