Transfer in Deep Reinforcement Learning for General Video Game Playing

IGGI Proposal
Potential supervisor: Dr. Diego Pérez Liébana, Queen Mary University of London

Anaëlle Laurans
anaellelaurans@outlook.fr

Abstract General Video Game Playing (GVGP) is a sub-field of artificial intelligence which aims to design an agent which would achieve high-level play in any given game. Transfer learning is particularly suitable for this field since it is the learning a new task through the transfer of knowledge from a previous task. Today reinforcement learning combined with deep learning have shown great results to build high-level features which can be transferable. The proposed study will focus on creating an agent that can decompose its behavior into meaningful primitives that can be reused for another problem. On successful completion, the research will have an impact on the game level process, bringing an agent able to test level for easing the map creation.

1 Introduction

Transfer in reinforcement learning aims to transfer knowledge across different tasks. The experience gained in learning to perform one task can speed up the learning performance in a related, but different, task. It supposes that some common features are shared between these tasks. Video games share several entities: enemies, bonus items and scores (life, points, time). Moreover, humans solve games by breaking them up into small behaviors which are reused as attacking an enemy. Likewise, research has focused on hierarchical reinforcement learning where agents represent complicated behaviors as a short sequence of high-level actions. The option framework [9] is a popular formulation for considering the problem with a two-level hierarchy. The bottom level, called option, is a sub-policy which takes observations and outputs actions until the termination condition is met. The top level is a policy-over-options which allow the agent to picks an option and follows it until termination.

In recent works, the *meta-learning shared hierarchies* algorithm, presented in [2], aims to find essential sub-policies over a distribution of tasks. For each new task, sub-policies are shared for learning a new master policy. Another interesting work is the *FeUdal Networks* architecture introduced in [10]. The difference with [2] is the goal notion introduce by the top-level to the lower-level. Whereas choosing a sub-policy, the top-level module must produce a sub-goal to give to the lower-level model. Moreover, the top level module operates in a temporal dimension by the use of recurrent neural networks.

Although these papers deal with transfer learning, in the general video game set-

ting, we need to face different state spaces and actions sets. If state spaces differ significantly, we may not be able to transfer correctly. The framework described in [7] tackles this problem by using both knowledge and skill transfer to describe the commonalities between tasks.

Finally, the reward function may differ between tasks and changes the adaptability of the agent drastically. [1] brings the idea of successor features to decouple the dynamics of the environment from the reward, to aid transfer. In [11], this idea is used for the robot navigation which needs to adapt to new situations quickly. They also contribute to preserving the solutions of earlier tasks during the transfer learning. To do so, they compact the representation of the Q-value functions of all encountered tasks.

2 Research Proposal

The study I intent to carry out is to further explore the transfer learning to improves the adaptability of an agent in the field of General Video Game Playing (GVGP). More precisely, I wish to create agents that can learn to decompose their behavior into relevant primitives and then reuse them to acquire new behavior. To begin, I will compare the frameworks proposed in [10] and [2] to study their performance to transfer sub-policies on a new task.

Not all previous knowledge is useful, and the agent must be able to find which is relevant to a new problem. Thus, with the previous work, I will combine it with the work in [7]. I think we can improve GVGP by letting the agent finds commonalities between games to adapt its behavior and ease the transfer.

Until now, I have considered the transfer of knowledge games, but it will be interesting to consider how the knowledge is acquired and reuse into a game. In most of the video games, the level design is thought for the player learns physics and possible interaction of the game through each level. For instance, in Mario Bros, the coins placement are not trivial. By placing the coins in a semi-circle, it encourages the player to jump and because there is an enemy nearby, the player memorizes the effect when jumping on them. I think it will be interesting to study how from smalls behaviors learned from the first levels the agent reuse them to create more complex behaviors. Here the work in [11] and [7] can be a real benefit for this problem. In [11], they use a policy reuse method to improve the exploration from experience and [6] combines two forms of transfer that can be needed to transfer knowledge to the environment.

In addition, the thesis will consider multi-player games as well as single player. Video games studios tend to create open worlds and massively multi-player games. To work on this aspect, I intend to use the Malmo framework [5], based on the game Minecraft. In this context, the agent will learn to collaborate or compete. It will imply to find the rights sub-policies to be able to react accordingly to others and the environment. Moreover, we can expect that collaboration or competitive behaviors can be transferred between games.

On the other hand, it can be tedious to find all the hyper-parameters while training an reinforcement learning model since they are combined with neural networks architecture. I will use the Population-Based Training method proposed in [4] to find an optimal set up quickly. Moreover, I will be inspired by [3], which combine several extensions of the deep Q-Networks, to improve the learning loop (exploration, exploiting replay, multi-step learning).

Finally, if there is sufficient time, future work can lead to injecting the temporal notion in the output of the sub-policies. In [8], macro actions have proved to be efficient in GVGP. Although we can see sub-policies as macro actions, we can go further by changing the output of a sub-policy from an action to a sequence of actions. It can permit to perform larger planning by the agent.

3 Study Impact

The successful completion of this study would result in an agent able to create complex behaviors from experience to quickly adapt to a new situation.

Good gameplay says what to do at a level without revealing how to do it. The level designer will create a map with basic assets. Then, he/she must test several times his/her map and correct it until it seem good enough to provide a clear goal to the player and provide fun. We can suppose that the level designer can benefit from an agent that also tests the map. Additional results can confirm the mechanics employed in the game or expose unexpected behaviors. In extension of this use case, playtesting can also aid developers for finding bugs in the Quality Assurance (QA) process.

This research could be applied in game engines like Unity, developed by Unity Technologies, or Unreal Engine, developed by Epic Games, which are used by game designers and developers. Moreover, by collecting the transferable part of the agent, we can expect to build a better agent in the future. Also, as this research considered multi-player games, game developers framework as SpatialOS, by Improbable could benefit from this research to help the indie teams to play test their MMO games.

References

[1] Andre Barreto et al. "Successor Features for Transfer in Reinforcement Learning". In: *Advances in Neural Information Processing Systems 30*. Ed. by I Guyon et al. Curran Associates, Inc., 2017, pp. 4058–4068.

[2] Kevin Frans et al. "Meta Learning Shared Hierarchies". In: (Oct. 2017). arXiv: 1710.09767 [cs.LG].

[3] Matteo Hessel et al. "Rainbow: Combining Improvements in Deep Reinforcement Learning". In: (Oct. 2017). arXiv: 1710.02298 [cs.AI].

[4] Max Jaderberg et al. "Population Based Training of Neural Networks". In: (Nov. 2017). arXiv: 1711.09846 [cs.LG].

[5] Matthew Johnson et al. "The Malmo Platform for Artificial Intelligence Experimentation". In: *Proc. 25th International Joint Conference on Artificial Intelligence* (2016).

[6] G D Konidaris. "A framework for transfer in reinforcement learning". In: *Knowledge Transfer for Machine Learning* (2006).

[7] George Konidaris, Ilya Scheidwasser, and Andrew Barto. "Transfer in Reinforcement Learning via Shared Features". In: *J. Mach. Learn. Res.* 13.May (2012), pp. 1333–1371.

[8] D Perez-Liebana et al. "Introducing real world physics and macro-actions to general video game ai". In: *2017 IEEE Conference on Computational Intelligence and Games (CIG)*. Aug. 2017, pp. 248–255.

[9] Richard S Sutton, Doina Precup, and Satinder Singh. "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning". In: *Artif. Intell.* 112.1 (Aug. 1999), pp. 181–211.

[10] Alexander Sasha Vezhnevets et al. "FeUdal Networks for Hierarchical Reinforcement Learning". In: (Mar. 2017). arXiv: 1703.01161 [cs.AI].

[11] Jingwei Zhang et al. "Deep Reinforcement Learning with Successor Features for Navigation across Similar Environments". In: (Dec. 2016). arXiv: 1612.05533 [cs.RO].