A review of deep reinforcement learning for game AI development

Atte Viitanen
atte.viitanen@aalto.fi

Tutor: Anton Debner

Abstract

Deep reinforcement learning (DRL) has seen massive growth in the AI field for the last few years. Like many other methods in the machine learning space, DRL also has significant ties to video games and simulations due to their use in training neural networks. This paper discusses the current uses and future prospects of the use of deep reinforcement learning in game AI development and compares its advantages to more traditional methods. This is achieved by conducting a literature review on recent papers regarding DRL and game AI. The review aims to provide analysis on both the current and future states of DRL usage in game AI development from both a technological and commercial point of view.

KEYWORDS: game development, AI, deep reinforcement learning, neural networks

1 Introduction

Through significant advances since its proposal, deep reinforcement learning (DRL) has seen a steady increase in use cases across a variety of fields. These fields include robotics and autonomous vehicles, natural language processing, computer vision, finance, healthcare and many more [12]. Another use case is in the field of video games and video game development, where this paper focuses.

Video games in general have been a a great testbed [16] and proving ground for DRL in general. DRL-based AI has seen many known achievements such as beating world champions in both traditional turn-based games such as Go in the case of AlphaGo in 2016 [20] and even teamwork-based real-time esports games such as Dota 2 in the case of OpenAI Five in 2019 [5]. In video game development, DRL has been proposed especially as a solution to traditionally more algorithmic problems such as procedural content generation [21] and game AI [24]. In addition, recently it's been proposed for use in the design process itself for problems such as adaptive gameplay [10] and automated game testing [4] as well.

This paper conducts a literature review of recent papers on the use of DRL in game development, with a focus on game AI. The aim of this review is to provide insight into what uses cases DRL has in game AI development, how it performs in said cases compared to more traditional approaches from both a quality and an ease of implementation standpoint as well as what the future potential of DRL in game AI development looks like.

This paper is organized into 3 main sections as follows. First is a background section that introduces the main concepts of DRL and it's related technologies. Second is a literature review of recent papers discussing DRL use in game AI. Lastly is a conclusion section where the some findings from the review are discussed and conclusions are drawn.

2 Background

This section briefly introduces the main concepts and fundamentals of machine learning directly related to DRL as well as provides a general description DRL itself. The section is divided into three sections. The first two sections concern the main components of deep reinforcement learning: reinforcement learning and deep learning. The last section then describes deep reinforcement learning and how it relates to the previous concepts.

2.1 Reinforcement learning

Reinforcement learning (RL) is an area of machine learning focused on learning what to do—how to map situations to actions—to maximize a scalar reward or reinforcement signal [22]. The learner is not told which actions to take, and must independently discover a mapping from situations to actions which yields the highest reward by trial and error. The reward is not necessarily immediate, and can be delayed for example in a situation where an action may have an effect on both an immediate reward and the following situation or state, and through that all the following rewards as well. These complex delayed rewards and the aforementioned trial-and-error search are typically considered the two most important distinguishing features of RL [23].

Reinforcement learning was first coined by Marvin Minsky in 1961, but was largely forgotten until the early 1980s when it became an active and well-known area of machine learning research [23]. Today reinforcement learning is used and studied in a multitude of disciplines such as game theory, control theory, economics and information theory.

2.2 Deep learning

Deep learning (DL), also known as deep structured learning, is a broad family of methods in machine learning that enables computers to learn from experience and understand hieararchies and concepts in raw natural data with minimal supervision [13]. DL solves the limitations of more conventional machine learning techniques where constructing pattern recognition or other machine learning systems required considerable expertise and care. These limitations were mainly due to having to design feature extractors to tranform raw input data into suitable internal representations or feature vectors for the learning system to use and recognize patterns in. [15].

In a simplified sense, DL approaches date processing by attempting to build complicated hierarchies and concepts present in input data out of smaller and simpler ones. These smaller hierarchies are contained in a graph many layers deep. [13] These graphs, known as deep artificial neural networks, imitate the brain in how it processes natural data.

The first general DL solution was published by Alexey Ivakhnenko and Lapa in 1967 [14]. Research quickly picked up speed especially in the 2000s as the advantages of deep neural networks became more clear as the increased processing power of computers enabled use of more complex networks. Today deep learning is used widely throughout the marchine learning space, most typically in applications where natural data is processed raw.

2.3 Deep reinforcement learning

Deep reinforcement learning (DRL) is a form of machine learning that combines the concept of reinforcement learning with the methods used in deep learning. This integration has a long history, but with recent advancements in computation processing power, software packages and big data, deep reinforcement learning has been growing increasingly popular [16]. The data processing capabilities of neural networks help mitigate the reliance on specific domain expertise when building reinforcement learning systems as they allow for automatic feature learning straight from raw input data.

Despite its popularity, DRL has had its fair share of problems to overcome. Deep learning applications would typically require large amounts of hand-labelled training data, while RL algorithms had to be able to learn from a scalar signal that could frequently be sparse, noise and delayed. Typical deep learning algorithms would also assume data samples to be independent, while in reinforcement learning there would often be sequences of highly correlated states. Lastly, the changing of data distribution as an RL algorithm learned new behaviours also proved problematic for DL methods which would often assume a fixed distribution. [17]

3 Deep reinforcement learning in game development

This section goes over the use of deep reinforcement learning in the field of game development. The section is divided into a general section and a literature review that focuses on the more specific subject area of DRL game AI. First the general section briefly discusses the use of deep reinforcement learning in the field of game development. The literature review then aims to crasp the current state and use cases of deep reinforcement learning in game AI with examples cases. These example cases are then compared to traditional non-ML game AI solutions and their associated problems.

In recent years, the realization that video games are perfect testbeds for modern artificial intelligence methods has spread widely in the AI research community [24]. A variety of games have since been effectively conquered, starting with classic turn-based ones and later advancing to more complex ones and even games requiring realtime input. Following the research interest, a multitude of software frameworks, environments and tookits such as the General Video Game AI (GVGAI) framework [18], the Arcade Learning Environment (ALE) [3] and the OpenAI Gym [7] have been developed. In addition to establishing benchmarks for AI comparison and development, these software platforms have even stirred up a lively competitive video game AI scene.

The allure of Video games for DRL and machine learning or AI development in general comes from the isolated and controllable playground it provides. A video game offers perfect or near-perfect repeatability and control over the game state. AI can be integrated directly into a game so that metrics on the game state may be measured and collected directly without for example monitoring the visual output of the game, simplifying the setup greatly. Even the speed of the simulation may be altered to scale with available processing power to speed up or slow down when training networks.

With all this focus on video games in the machine learning research space, its's no surprise that game development has found various uses for machine learning as well, including DRL. One major use case is in developing video game AI, to which the AI research discussed previously applies directly. While this paper will focus game AI, it should be noted that it's by no means the only use case for DRL in game development.

Due to the substantial amount of DRL and other machine learning research using games as a test bed, game AI development is a more or less an obvious use case for DRL. Traditionally game AIs intended to provide players an opponent or challenge have been done algorithmically using solutions such as finite-state machines, various pathfinding algorithms, triggers, sets of predefined, scripted actions and such. While these implementations can work when done well, they typically simply imitate some expected behaviour patterns of a real player and do not truly emulate the actual thought process a real player might go through in a given situation. As such, their behavior can seem simplistic, near-sighted or break down in atypical scenarios. In more complex games, the in-game AI is still generally considered easily distinguishable from a real player.

3.1 Game AI

Improving RTS Game AI by Supervised Policy Learning, Tactical Search, and Deep Reinforcement Learning

Barriga et al. [2] proposed a new method of using deep neural networks to select between action choices in Real-Time Strategy (RTS) game AI systems and conducted a case study investigating its use with DRL in the modern RTS game Total War: Warhammer by Creative Assembly. The case study's AI produced notably higher win rates than known state-of-the-art machine learning RTS AI solutions. Against the game's built-in traditional AI opponent, Barriga et al.'s [2] solution reached over 90% winrates in more simple macthes and 82±3% and 77±3% winrates in more complicates scenarios with various mixed unit types with 3v3 and 6v6 matchups respectively.

Barriga et al.'s [2] setup was of course not flawless, though: For scenarios with more than 3 opponents per side, learning was unstable and as such the results were poor. Barriga et al [2] presumed it might have been due to the game environment changing too rapidly with so many players. When using a hierarchical RL approach, Barriga et al [2] also failed to obtain obtain good results when training both high-level and low-level policies or actions jointly, failing to reach over 40% winrates against the default AI.

The network training methodology in Barriga et al.'s [2] study highlighted similarities to real humans learning a game: the best learning results were achieved by slowly introducing more complicated scenarios and difficulty as the training progressed. The paper [2] noted that the AI quickly learned techniques and strategies often used by human players such as pinning, flanking and using each unit's unique strengths to their advantage. This kind of learning inherent to the ML apprioch is something that traditional game AI methods completely lack. To achieve similar results with traditional means, the developer would have to observe human players and then manually configure state machines, scripts, or some other action templates to reproduce those behaviors.

Barriga et al. [2] also stated that the ability for a human player to learn from such an AI's increasingly complex strategies was a way to create more satisfying game experiences. In other words, a DRL-based AI could learn to perform intuitive strategies not doable with traditional AI methods and essentially teach those strategies to the human player, drastically improving the experience. Barriga et al. [2] concluded that the accelerated research efforts in the ML game AI space together with the availability of frameworks to take advantage of readily available game replay data would likely revolutionalize both game design and playing in the near future.

Developing Combat Behavior through Reinforcement Learning in Wargames and Simulations

Boron and Darken [6] explored the use of DRL in RTS games for achieving optimal offensive behavior in small tactical engagements. However, Boron and Darken [6] took a more sophisticated approach to performance validation. Instead of focusing on the end results of the engagements, performance was evaluated based on two real-world principles of war — mass and economy of force. In simpler terms, mass is the the concentration of forces in the decisive place at the decisive time, while economy of force means allocating the minimum combat power to secondary efforts to achieve a superiority in mass. Three RL algorithms were tested; Vanilla Policy Gradient (VPG), Proximal Policy Optimization (PPO), and Trust Region Policy Optimization (TRPO). Using said algorithms, four different combat models were examined in thee different scenarios in a custom time-stepped, turn-based simulation.

While Boron and Darken's [6] results are hard to quantify in relation to traditional AI as no such AI was used, they found that the AI agents successfully employed combat strategies to maximize the two measured principles of combat. By varying the learning parameters the AI agents would switch between favoring mass or economy of force in their behaviour. The benefits of DRL were highlighted here again, as in traditional AI programming it would be very difficult to impossible to have AI intuitively maximize such real-world principles of war, especially considering that the balance of a game can change with updates and so on. It also worth noting that, depending on the implementation, with traditional game AI methods, the easy switch between favoring either mass or economy of force seen here could likely require significant amounts of work in manually re-weighing and fine-tuning various predetermined actions and patterns.

Boron and Darken [6] noted that for engagements bigger and more complex, evaluation of the AI's behavior and assessing whether a global performance optimum has been reached becomes increasingly difficult. There is also a distinction between the concepts of optimal and tactical behaviour to consider. Due to these factors, Boron and Darken [6] resorted to simpler, smaller engagements but stated that future work would aim to validate performance in larger and more complex sccenarios. It was also noted that other kinds of missions such as defence or raids should be considered as well.

Beating the World's Best at Super Smash Bros. Melee with Deep Reinforcement Learning

Firoiu et al. [11] demonstrated a DRL-based AI capable of beating professionals in the popular retro fighting game Super Smash Bros. Melee (SSBM) by HAL Laboratory. Contrary to what's usually seen in most other fighting game AI research, SSBM is a multi-player game, and as such features more complex mechanics and a constantly evolving metagame. For training, Firoiu et al.'s [11] AI was pitted agaist both SSBM's built-in traditional AI at the highest difficulty setting as well as itself.

Firoiu et al. [11] found their AI beating the built-in version relatively quickly with various methods. Depending on the methods used, the DRL-based AI could either beat its traditional counterpart fair and square or even employ a relatively unorthodox but impressive multi-step tactic to trick the enemy into killing itself by falling off the stage. In the case of professinal players, the AI performed favorably against anyone who was willing to face it, with a total of 10 top-100 ranked players in the world having a try with multiple attempts each.

Firoiu et al.'s [11] solution was not without faults, though. The network used was only trained to play as and against specific characters at once. Firoiu et al. [11] also did not have success in training the AI with experiences from multiple game character's points of view. Such attempts resulted in the networks failing to adapt to different character strategies and ended with inferior strategies solely utilizing a set amount of basic actions that were common to the trained characters.

Firoiu et al. [11] also noted that the AI exhibited some strange weaknesses, one example being a loophole a player found where the network would behave very oddly when faced with an enemy crouching at the edge of the stage, resulting in the AI character falling of the stage. Firoiu et al. [11] hypothesized that these anomalies were likely due to a lack of diversity in training, where such odd situations were unlikely to happen.

4 Conclusions

In general, it seems that DRL is not yet, at least widely, in use commercially as an AI solution that could provide real players an opponent or challenge in a game. While current research seems to be advancing at a quick pace, it's lacking in references to any commercial or real-world use cases. While ML methods have been used for game AI in the past, an example being integrating the popular game engine Unity with a popular ML framework [8], DRL has not seen nearly as much real-world use.

Apart from massive scale AI's such as OpenAI Five [5], most research, including what was reviewed in this paper, were typically limited in some scope. As an example, Boron and Darken [6] had to resort to very simplistic engagement scenarios due to training becoming unstable and Firoiu et al. [11] only managed to train their network to play as and against specific characters at once. However, in both cases the conclusions were optimistic in regards to resolving these issues and broadening the scope.

The lack of commercial and other real-world use-cases could be due to current research only focusing on improving the AI's general performance for now. As such the goal is often to simply come up with solutions that play the best. This is not that useful for commercial game applications though, as games typically use AI to provide a challenge and thus need balancing and to adapt to the human player's skill level. Andrade et al. [1] describe game balancing as a key feature of subbesful games. The problem of balancing is described to consist in changing parameters, scenarios and behaviours in order to avoid the player either getting frustrated at the the game being too hard or bored because it's too easy. This is something that current DRL game AI in research lacks completely.

Another clear issue with commercial DRL use in games is performance. Excluding various cloud gaming services, games run on consumer devices with limited power and processing power budgets and capabilities. The performance impact of running neural networks associated with DRL are by no means light. In research, these networks are typically run on server hardware in a highly parallelized fashion with much less worry over performance constraints. As an example, Firoiu et al. [11] used over 50 or more servers in parallel when training their SSBM AI, and in an even more extreme case the famous OpenAI Five [5] AI utilized a peak of 1536 graphics processing units in parallel during training.

While there might be optimizations to be made for commercial applications, there is also the issue that neural network processing is typically accelerated with graphics processing units. However, in a consumer device, the graphics processor tends to already be under heavy stress in a game workload as it's used for graphics rendering.

The above issues are mostly temprorary though, and overall the future for DRL use in game AI looks very promising. The issue of taking into account game balancing is simply pending further research, and game AI balancing has already familiar topic in traditional RL context, for example in the studies of Andrade et al. [1] and more recently Pfau et al. [19] The issues regarding performance is also simply pending computing power advancements. In addition, graphics processor manufacturers have already started including AI-dedicated harware in consumer devices as well with the example of Nvidia's tensor cores [9]. Such developments could get rid of the need to cut back on graphics processing to be able to do AI computation.

Currently, the main gripe for game studios in regards to seriously adopting DRL for a game project would likely be the general growing pains of DRL. As everything related to state-of-the-art DRL AI is so new, it's also hard to place expectations on the performance and viability of the solution without first investing serious effort into it. There is always the danger that later on when training a network it's realized that the AI is simply too unstable like in Barriga et al.'s [2] situation in scenarios with more than 3 opponents per side. Similarly, the AI might turn out to have some quirk that makes it unsuitable for built-in AI use, such as the odd edge crouching issue with Firoiu et al.'s [11] SSBM AI. As DRL and it's related technologies are evolving at such a fast pace, there's also the uncertainty of an implementation becoming obsolete if some major breakthrough or improvement is figured out later on. All this uncertainty does not play in DRL's favor when game studios consider the commercial viability of any prospective technologies when starting out a project.

In regards to the impact DRL could have for game AI, Barriga et al. [2] claimed that the ability to construct DL-based high-performance game AI systems to act against real players would revolutionize the video game industry. This makes sense, as in an ideal case developing an DRL game AI could be analogous to training a human to play a game and then asking them to perform as the enemy for other players. In terms of training, the training methodology for DRL networks can already resemble training a real human as seen in Barriga et al.'s [2] training methodology. In regards to the AI's behaviour, the resulting AI's strategies also tend to already resemble human gameplay in both their ability and intuitivity as seen in all of the reviewed papers.

Such a simplistic way of building a game AI is a far cry from the very manual and labor intensive processes that traditional game AI development can entail, especially for more sophisticated ones. It also avoids the pitfalls that traditional AI can have where the AI needs to be carefully adjusted if game mechanics change, get added or are removed. In such a situation, an DRL AI would either be able to adapt without changing anything or at the worst case would need to be re-trained, which is a relatively repeatable and even automatable process assuming the game mechanics changes were not too groundbreaking.

With the above in mind, it's no wonder DRL and other methods in the ML space are so often seen as the future for AI in general. Considering the ideal game AI discussed, it certainly seems to apply to game AI as well. As seen in Barriga et al.'s [2] study, DRL based AI clearly have the capability to reach and exceed the level of traditional AI. It's simply a matter of adapting it to the role of an adjustable and suitably challenging opponent for real players. What that entails is waiting for these DRL methods and associated frameworks to mature even further and for consumer hardware to catch up to their requirements.

References

[1] Gustavo Andrade, Geber Ramalho, Hugo Santana, and Vincent Corruble. Extending reinforcement learning to provide dynamic game balancing. In *Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games, 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 7–12, 2005.

[2] Nicolas A Barriga, Marius Stanescu, Felipe Besoain, and Michael Buro. Improving rts game ai by supervised policy learning, tactical search, and deep reinforcement learning. *IEEE Computational Intelligence Magazine*, 14(3):8–18, 2019.

[3] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, Jun 2013.

[4] Tollmar Gisslen Quirin Hashme Shariq Hesse Chris Bergdahl, Gordillo et al. Augmenting automated game testing with deep reinforcement learning. 2020.

[5] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[6] Jonathan Boron and Chris Darken. Developing combat behavior through reinforcement learning in wargames and simulations. In *2020 IEEE Conference on Games (CoG)*, pages 728–731. IEEE, 2020.

[7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[8] Ciro Continisio and Alessia Nigretti. *UnityTechnologies/MachineLearningRoguelike: A small Roguelike game that uses Machine Learning to power its entities. Originally used in talks by Ciro Alessia.*, (accessed November 29, 2020). https://github.com/UnityTechnologies/MachineLearningRoguelike.

[9] Nvidia Corporation. *Tensor Cores | NVIDIA Developer*, (accessed November 29, 2020). https://developer.nvidia.com/tensor-cores.

[10] Aline Dobrovsky, Uwe M Borghoff, and Marko Hofmann. Improving adaptive gameplay in serious games through interactive deep reinforcement learning. In *Cognitive infocommunications, theory and applications*, pages 411–432. Springer, 2019.

[11] Vlad Firoiu, William F. Whitney, and Joshua B. Tenenbaum. Beating the world's best at super smash bros. with deep reinforcement learning. *CoRR*, abs/1702.06230, 2017.

[12] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G Bellemare, and Joelle Pineau. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.

[13] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[14] Alexey Grigorevich Ivakhnenko and Valentin Grigorevich Lapa. Cybernetics and forecasting techniques. 1967.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[16] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[17] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[18] Diego Perez-Liebana, Spyridon Samothrakis, Julian Togelius, Simon M Lucas, and Tom Schaul. General video game ai: Competition, challenges, and opportunities. 2016.

[19] Johannes Pfau, Antonios Liapis, Georg Volkmar, Georgios N Yannakakis, and Rainer Malaka. Dungeons & replicants: automated game balancing via deep player behavior modeling. In *2020 IEEE Conference on Games (CoG)*, pages 431–438. IEEE, 2020.

[20] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[21] Adam Summerville, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K Hoover, Aaron Isaksen, Andy Nealen, and Julian Togelius. Procedural content generation via machine learning (pcgml). *IEEE Transactions on Games*, 10(3):257–270, 2018.

[22] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[23] Richard S Sutton, Andrew G Barto, et al. *Introduction to reinforcement learning*, volume 135. MIT press Cambridge, 1998.

[24] Ruben Rodriguez Torrado, Philip Bontrager, Julian Togelius, Jialin Liu, and Diego Perez-Liebana. Deep reinforcement learning for general video game ai. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2018.