IFERP
ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

Delineation of Agents for Games using Deep Reinforcement Learning

[1] Pranav Batra, [2] Sambhav Jain, [3] Asst. Prof. Yamini Ratawal
[1][2][3] Department of Computer Science, Dr. Akhilesh Das Gupta Institute of Technology and Management

Abstract: Artificial Intelligence (AI) agents should have a precise representation of their surroundings and should be able to retain information based on their past experiences and from sensory inputs. It is done by developing smart self-learning agents using various Deep Reinforcement Learning algorithms such as Q-learning (DQN) and SARSA learning (DSN). These algorithms have helped game developers in removing the monotonous nature of games helping in the augmentation of the gaming industry. Intelligent agents can also be used in game development as these agents inhibit human-like behaviour which can help with the game balancing and assessment, almost identical to the performance of game developers. First Person Shooter (FPS) games have seen a significant gain in popularity due to increased difficulty and lack of predictability. This paper is an attempt in doing the same by reviewing various existing works in this field and some of the popular techniques of developing intelligent Agents.

Keywords— Artificial Intelligence (AI), Reinforcement Learning (RL), Q-learning (DQN), SARSA Learning (DSN).

1. INTRODUCTION

Designing and improving smart self-learning agents using Deep Reinforcement Learning [1-4] is a demanding task with the key intricacy of vast action spaces and inadequate exploration. Due to the recent developments in Deep Reinforcement Learning, autonomous game agents are now often outperforming human beings in decision making. From an Artificial Intelligence perspective, designing adaptive games that augments the player's immersion has been challenging since it is not very clear as of now how to design and execute adaptive games efficiently and which game aspects can be altered to accomplish this target. In this paper, we are evaluating some methods to apply Reinforcement Learning to design and improve smart self-learning agents in a computer game.

Artificial Intelligence [5-8] plays a remarkable role in computer games by benefitting Non-player characters (NPC's). These NPC's have access to most of the concealed particulars of the game's domain. Due to this, these agents often become invulnerable. To counter this problem, various game developers have to alter the data accessible by these agents and to crumble their performance by incorporating factors such as distance, field of view, and time to make the game less monotonous. However, this approach often influences the gaming experience and generates NPC's, which are more likely to have non-human behaviour.

There are various drawbacks to the non-human behaviour, the most significant one being the dissatisfaction of the users resulted due to the monotonous nature of the non-player character. It makes the game less challenging as the player can easily predict the next move of the non-player character. Also, humanised behaviour will have better adaptability and augmented intelligence, and therefore it becomes essential to have intelligent non-player characters in the game.

In today's scenario, AI plays a very crucial role in most of the games; while, games are also presenting complex environments for the AI, challenging its robustness and interaction with humans. Current games require stimulation of human behaviour, which can be processed and interpreted by using Deep Reinforcement Learning. The derived behaviour of humans is served to these agents for providing a better gaming experience. Fortnite [9], developed by Epic Games, is not just an internationally renowned game; it also provides a virtual space to players for socialising. Similarly, another popular console PlayStation VR released by Sony [10] enables us to experience virtual reality in the solace of our homes. There exist many more examples of such revolutionising changes that occurred in the gaming industry, which has resulted due to the recent advancements in AI. Fig. 1. depicts some of the advantages of the DRL in games [11], which has recently assisted in enhancing the gaming experience. With these advancements we can currently extrapolate how AI is going to affect the gaming industry in the upcoming decades.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

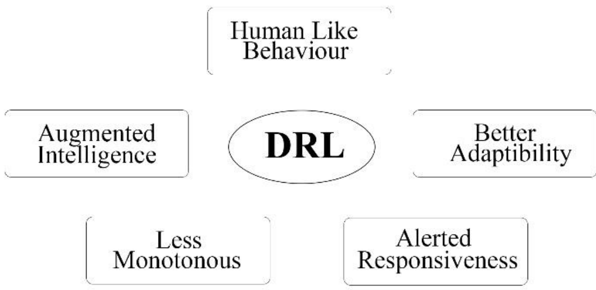

Fig. 1. Advantages of DRL in Games

This paper aims to present Deep Reinforcement Learning agents as a way of rendering non-player characters which are smart, self-learning, adaptable and can also evolve. The fundamental idea of this work is to discuss the shortcomings and the future aspects of this technology for a congruous development of the gaming industry, enhancing its user experience.

2. LITERATURE REVIEW

Deep Reinforcement Learning is of great importance in the field of AI. It allows representing mathematical methods for agents to act flawlessly in complex environments. A great deal of work has been done for developing smart self-learning agents for games. However, DQN and DSN are just a couple of available methods which allows us to embed idiosyncrasies in these agents. Therefore, it is necessary to perform a comprehensive study of the existing works to re-evaluate the issues and the room for advancements.

Enhancing Game Development

Yunqi Zhao [12] et al. emphasizes the creation of smart self-learning agents to corroborate the process of game development. These agents inhibiting human-like behaviour can help with the game balancing and assessment, almost identical to the performance of game developers. They bestowed four case studies, each demonstrating a parity between skill and style. Also, they have evaluated multi-faceted theories in this paper with each having their pragmatic implication. In the first two case studies, they have created playtesting agents for mobile games that showed how the observation space could substantially influence the efficacy of the solutions trained with RL. The other two case studies are engrossed towards designing game playing agents that shows how the existing RL models need to be tweaked to perform well on the various environments, even including Atari. All these case studies helped in recognizing the challenges game designers have to encounter in order to accomplish their desired outcome. Many such intelligent agents are developed for improving games [13] which plays a crucial role in enhancing the gaming experience.

Playing Atari

Volodymyr Mnih [14] et al. aims to provide a detailed model of a convolutional neural network that rests upon the methods of Q learning. The method is applied to a range of Atari 2600 games where the input is given in the form of raw pixels and the output received is a value function giving an idea of the future rewards. All the Atari 2600 games are in The Arcade Learning Environment (ALE) [15]. A set of tasks was given to the agents introducing a challenging level for human players. These agents partially observed the tasks as it was impossible to understand the current situation by just analyzing the current screen. For six out of seven games, in which the method is applied, the network has outstripped all the earlier RL algorithms and even outperformed an expert human player on the three of them. Finally, the author was successful in providing a new Deep Learning Model for Reinforcement Learning for agents in the Atari 2600 games without altering the architecture and hyper-parameters.

Playing FPS Games

Guillaume Lample [16] et al. have presented the primary architecture to address the 3D environments in first-person shooter games. Generally, agents made from typical Deep Reinforcement Learning methods use visual input, i.e. raw pixels for training. However, in this paper, the author reinforced these methods to utilise game feature information; for instance, the position of enemies and the placement of items. These adjustments have not only shown to improve the performance of the agent but also to enhance the training speed. In the complex scenarios, such as deathmatches, the standard DRQN model was not able to handle the tasks as efficiently as the improved model was able to. The ViZDoom platform [17] was used to perform all the experiments and analyse the methods on the deathmatch scenarios. Ultimately, the author was able to show that the new model outperformed not only built-in AI agents of the game but also standard humans in deathmatches.

Reinforcement Learning

Michelle McPartland [18] et al. shows how the implementation of Reinforcement Learning can benefit first-person shooter games. They have applied the Sarsa RL algorithm to an FPS [19-20] game built for the intended purpose. A custom-built game was used instead of a commercial game because in commercial games turning rendering on and off can be a bit difficult plus the processor cycle is also needed to keep to a minimum. In the first half of their analysis, the author demonstrated how the RL algorithm was able to grasp an approach for navigation control. It was able to acquire a satisfactory strategy for trail designing. However, it was not up to par with the quality of standard pathfinder algorithm. In the second part, they have created more generic bot AI by using RL controllers showing the strength of RL algorithms and how they can be used in FPS games and help bots with a variety of tasks like combat, navigation and item collection.

A Survey of DRL

Kun Shao [21] et al. points out the application of DRL through various game research platforms in Atari games, first-person perspective games [22] (ViZDoom [17], TORCS, Minecraft, DeepMind Lab) and real-time strategy games [23] (StarCraft [24], MOBA [25] and Dota2). The very detailed yet alleviated explanation elucidated in this paper about the classes of deep neural networks, i.e. Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN). Following it, the author thoroughly explains the challenges faced in the games while implementing DRL. The significant challenge that is faced is when there are multiple agents, and multiple-agent learning needs to be performed, such as in StarCraft [24].

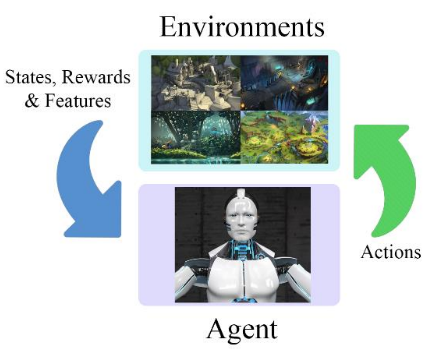

Fig. 2. Correlation between Environments and Agent

This paper surveys through most of the types of games in which DRL can be embedded. Finally, it concludes that while Game AI with Deep Reinforcement Learning is challenging, it provides a promising direction.

Double-Q Learning

Hado van Hasselt [26] et al. has given an advancement over the popular Q-learning algorithm. It is stated that the Q-learning algorithm overestimates the action values as well as deteriorate the performance in some specific situations. The author constructed a new algorithm, called Double DQN, which not only provides more accurate values of the estimates but also augmented the scores in many games. Experiments were performed on six Atari games using DQN and Double DQN, and both of them were trained under the same conditions. It was seen that DQN was more extreme about overestimations and was highly unstable on some of the games.

On the contrary, Double DQN showed much more stability for those games suggesting that the actual reason for the instabilities was Q-Learning's overoptimism. At last, it was concluded that Double DQN was more robust and had a cut above the standard DQN. Double DQN attained better policies and obtained spectacular results on the Atari 2600 domain.

DRE-Bot

Frank G. Glavin [27] et al. provides us with a framework to enhance the working of Non-player characters in the FPS game Unreal Tournament 2004 [28]. DRE-Bot, i.e. danger, replenish and explore, is constructed on these three terms and makes the use of Sarsa algorithm [19-20]. The bot is built upon the foundation of learning through trial and error method. This foundation helps the bot to inherit the qualities of human-like behaviour. As many different game types exist within FPS games, in this paper, the author is trying to develop a bot that is only competent in the deathmatch game type. The bot should be able to make real-time decisions such as, path finding and combat, which is always a complicated task. A tangential hint in the importance of graphics for the

ISSN (Online) 2394-6849

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 7, Issue 9, September 2020

TABLE I.
OVERVIEW OF LITERATURE

Source	Topic of review	Key facets	Room for Advancements
[12]	Enhancing Game Development	Artificial Intelligence, Playtesting, Multi-agent Learning, Imitation Learning	Some policies can be implemented that could match to variety of teammates without ample tuning
[14]	Playing Atari	Deep learning, Q-Learning, Atari games, Sarsa Algorithm	Can have a network to perceive blueprint that builds up over prolonged time scale
[16]	Playing FPS Games	Recurrent Q networks, FPS Games, Game feature Augmentation, KD Ratio	Methods can be integrated with two separate estimators
[18]	Reinforcement Learning	Computer Games, Reinforcement Learning, FPS Games, MAS	Bots can be made more convoluted by augmenting the state and action spaces
[21]	A Survey of DRL	Arcade Games, Complex 3D, Simple 2D, Challenging Environments	New frameworks are required for complex environments
[26]	Double-Q Learning	Double DQN, Overoptimisms of Q-Learning, Neural Networks, Atari 2600	Policies and architecture can be improved
[27]	DRE-Bot	First Person Shooter, Non Player character, Bot Architecture	Can be further evaluated in more demanding environment, having sizeable maps and added opponents
[29]	3D Video Game	3D, Delivery Duel, OpenAI, Gym, ML Agents, Game environment representation	More experiments can be carried out on various game environments

games are also provided in the paper. Finally, the main objective of the paper is to develop a bot that can learn from its strategy and simultaneously adapt over time. This bot will act as a more challenging and natural opponent for all kinds of skilled human players.

3D Video Game

Samuel Arzt [29] et al. has applied two methods, namely DQN and A2C, to develop an agent for a 3D video game called Delivery Duel. The author describes the working of agents and their specifications. The main idea of his work is on the impact of changes on the learning performance. The representations of the environments are Complex 3D and Simple 2D.

Fig. 3. Render modes- Complex 3D (Left) and Simple 2D (Right) [29]

In the game, Delivery Duel, the player needs to control a delivery van and agent's goal is to deliver pizza to the predefined location. The foundation of the game is built upon physics-based motion control which increases the difficulty level of the game but also enhances the fun. Finally, the author was able to implement these methods to train the agent in these environments. The continuation of the reward signal very much improved the learning performance. The results concluded from this encourages further experiments on this environment

3. OBSERVATIONS

After reviewing the aforementioned work, it would be appropriate to say that several issues need to be addressed, as well as several techniques can be further investigated. The development of smart self-learning agents in games using DRL has undoubtedly given a boost to the gaming industry by removing the monotonous nature of games; however, the methods carried out to achieve this can be improved for a much better result. Below are some of the key facets which can be further improved.

Atari Agents

Agents developed for Atari games using DRL gives the game a cutting edge over the other Atari games. As the previous work is undoubtedly state-of-the-art, some tweaks will make it better than before. Modifications should be done to the Atari agents to ensure that the experience replay buffer is preserved during training runs. Also, while some methods can clinch performance superior to an expert human player, some are way far from human-like performance. These methods can be reevaluated with more convoluted networks.

Network

The network used by the agent is essential. A network such as DQN is used to estimate the value of the Q function. The approach of using Double DQN over DQN appeared to be better for some games. Furthermore, the networks employed by the agents should be refined through network optimization techniques, such as dropout. Dropout is used to decrease the overfitting within a network with many layers and neurons.

Performance

It is evident that AI has significantly improved the performance of the agents in various kinds of games. The agents have gained much more qualities resembling human-like behaviour and give a better combat experience. The training provided has become much more rigorous and detailed. Nevertheless, continuous evaluation and early stopping mechanism can be implemented, stopping training when agents reach the desired performance.

Complexity

Future work should increase the complexity of the agent behaviour by increasing the state and action spaces. Interpolated tables should be used for smoother state transitions, and the agents should then be tested in commercial FPS games.

Architecture

Many architectures have been provided for various kinds of agents. These architectures not only help develop the agent but also gives it the desired qualities. Each kind of architecture has its pros and cons. Although, the hierarchical architecture can be further investigated and shall be made more suitable for learning skills. The higher level of macro actions can also be designed.

Algorithms

Although there are numerous algorithms used in Deep Reinforcement Learning, in this paper, we have discussed Sarsa and Q-Learning algorithms. These algorithms are very detailed and thorough on their own. The algorithms were used to train the reinforcement learning agents. The learning outcome should be saved as an SGF file to generate opening libraries.

Neural Networks

Objective function-based learning should be studied via the incorporation of a neural network model. It is expected that the guidance provided by the neural network will further improve the learning outcome.

Other intelligent methods, such as Temporal Difference (TD) Algorithm [30] can also be used to improve the Reinforcement Learning process of the agents. The incorporation of the TD Algorithm will provide an agent with the ability to learn and will significantly improve the sensing-feedback mechanism. TD Learning can also be used to optimize the parameters of the neural network.

4. Conclusion

Our paper is aimed to investigate the performance of AI agents created using both the DRL techniques, Q-learning (DQN) and SARSA learning (DSN). The differences between the on-off policy methods can be emphasized by using these agents for standard games which also intimates us how the on-policy Sarsa can give better and efficient solutions under intensely convoluted environments when juxtaposed with the off-policy Q-learning ones. The idea that the agents can learn from raw sensory inputs is also corroborated by the Atari 2600 agents, which also bolster the concepts that Sarsa is better than Q-Learning.

Also, we have noticed that the agents made using Deep Reinforcement Learning not only removed the monotonous nature but also enhanced the adaptability of these agents to the games, increasing its robustness. While the observed techniques and methods have their respective drawbacks, they are still considered extremely beneficial for the gaming industry. Ranging from Atari 2600 to first-person shooter games, the methods have performed exceptionally well in most types of games. Furthermore, it can be concluded that Double DQN surpasses standard DQN by providing more accurate values of the estimates, and it also increased the scores in many games. It can be extrapolated that the methods discussed above can be used for simple 2D as well as complex 3D games. The various models discussed have outperformed the built-in AI agents of the game as well as fledging human players enhancing the gaming experience. The incapability of creating an Atari agent which exceeds human performance is one of the drawbacks of this evaluation. However, we can successfully conclude that AI Agents can learn from their past experiences and raw sensory inputs, enabling them to have a precise representation of their surroundings.

REFERENCES

1. L. P. Kaelbling, M. L. Littman, and A. W. Moore, “Reinforcement learning: A survey,” Journal of artificial intelligence research 4, pp.237-285, May 1996.
2. V. Zambaldi, D. Raposo, A. Santoro, V. Bapst, Y. Li, I. Babuschkin, K. Tuyls et al. "Deep reinforcement learning with relational inductive biases," International Conference on Learning Representations, 2018.
3. K. Clary, E. Tosch, J. Foley, and D. Jensen. "Let's Play Again: Variability of Deep Reinforcement Learning Agents in Atari Environments," arXiv preprint arXiv:1904.06312, 2019.
4. D. Silva, F. Leno, P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "Uncertainty-Aware Action Advising for Deep Reinforcement Learning Agents," AAAI, pp. 5792-5799, 2020.
5. L. Gruenwoldt, S. Danton, and M. Katchabaw, "Creating reactive non player character artificial intelligence in modern video games," Proceedings of the 2005 GameOn North America Conference, pp. 10-17, 2005.
6. S. S. Esfahlani, J. Butt, and H. Shirvani, "Fusion of artificial intelligence in neuro-rehabilitation video games," IEEE Access 7, pp. 102617-102627, 2019.
7. W. Westera, R. Prada, S. Mascarenhas, P. A.

Santos, J. Dias, M. Guimarães, K. Georgiadis et al., "Artificial intelligence moving serious gaming: Presenting reusable game AI components," Education and Information Technologies 25, no. 1, pp. 351-380, 2020.

8. R. Thawonmas, J. Togelius, and G. N. Yannakakis, "Artificial general intelligence in games: Where play meets design and user experience," 2019.
9. B. J. Pohl, A. Harris, M. Balog, M. Clausen, G. Moran, and R. Brucks, "Fortnite: supercharging CG animation pipelines with game engine technology," Proceedings of the ACM SIGGRAPH Digital Production Symposium, pp. 1-4, 2017.
10. Entertainment, Sony Interactive. "Playstation vr," Website, (2017), Retrieved April 10 2020.
11. A. Dobrovsky, U. M. Borghoff, and M. Hofmann, "Improving adaptive gameplay in serious games through interactive deep reinforcement learning," Cognitive infocommunications, theory and applications, Springer, Cham, pp. 411-432, 2019.
12. Y. Zhao, I. Borovikov, F. D. M. Silva, A. Beirami, J. Rupert, C. Somers, J. Harder et al., "Winning Isn't Everything: Enhancing Game Development with Intelligent Agents," IEEE Transactions on Games, 2020.
13. J. Suarez, Y. Du, P. Isola, and I. Mordatch, "Neural MMO: A massively multiagent game environment for training and evaluating intelligent agents," arXiv preprint arXiv:1903.00784, 2019.
14. V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," arXiv preprint arXiv:1312.5602, 2013.
15. A. Taïga, W. Fedus, M. C. Machado, A. Courville, and M. G. Bellemare, "Benchmarking bonus-based exploration methods on the arcade learning environment," arXiv preprint arXiv:1908.02388, 2019.
16. G. Lample, and D. S. Chaplot, "Playing FPS games with deep reinforcement learning," Thirty-First AAAI Conference on Artificial Intelligence, February 2017.
17. A. Khan, N. Naeem, Z. A. M. Z. Asghar, A. U. Din, and Atif Khan, "Playing first-person shooter games with machine learning techniques and methods using the VizDoom Game-AI research platform." Entertainment Computing 34, 100357, 2020.
18. M. McPartland, and M. Gallagher, "Reinforcement learning in first person shooter games," IEEE Transactions on Computational Intelligence and AI in Games, vol. 3(1), pp. 43-56, 2010.
19. B. B. Vivien, "Features and Performance of Sarsa Reinforcement Learning Algorithm with Eligibility Traces and Local Environment Analysis for Bots in First Person Shooter Games." PhD diss., Waseda University, 2020.
20. M. H. Olyaei, H. Jalali, A. Olyaei, and A. Noori, "Implement Deep SARSA in Grid World with Changing Obstacles and Testing Against New Environment," Fundamental Research in Electrical Engineering, Springer, pp. 267-279, Singapore, 2019.
21. K. Shao, Z. Tang, Y. Zhu, N. Li, and D. Zhao, "A survey of deep reinforcement learning in video games," arXiv preprint arXiv:1912.10944, 2019.
22. D. Choi, T. Konik, N. Nejati, C. Park, and P. Langley, "A believable agent for first-person perspective games." 3rd Artificial Intelligence and Interactive Digital Entertainment International Conference, 2007.
23. Y. Tian, Q. Gong, W. Shang, Y. Wu, and C. L. Zitnick, "Elf: An extensive, lightweight and flexible research platform for real-time strategy games," Advances in Neural Information Processing Systems, pp. 2659-2669, 2017.
24. O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani et al, "Starcraft ii: A new challenge for reinforcement learning," arXiv preprint arXiv:1708.04782, 2017.
25. M. Mora-Cantallops, and MÁ. Sicilia, "MOBA games: A literature review," Entertainment computing, vol. 26, pp. 128-138, 2018.
26. H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," Thirtieth AAAI conference on artificial intelligence, March, 2016.
27. F. G. Glavin, & M. G. Madden, "DRE-Bot: A

hierarchical First-Person Shooter bot using multiple Sarsa (λ) reinforcement learners," 2012 17th International Conference on Computer Games (CGAMES), pp. 148-152, IEEE, July 2012
28. Games, Epic. Unreal tournament 2004. Atari, 2004.
29. S. Arzt, G. Mitterlechner, M. Tatzgern, and T. Stütz, "Deep Reinforcement Learning of an Agent in a Modern 3D Video Game."
30. W. Konen, "Reinforcement learning for board games: The temporal difference algorithm," Research Center CIOP (Computational Intelligence, Optimization and Data Mining), TH Köln–Cologne University of Applied Sciences, Tech. Rep, 2015.