IFERP
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 11, Issue 12, December 2024

Reinforcement Learning in Game Development: A Comprehensive Survey

[1] Rahul Bhat, [2] Radhika Malhotra, [3] Shivam Shekhar, [4] Atharva Mondhe, [5] Abhishek Budhiraja, [6] Shivani Sharma

[1] [2] [3] [4] [5] Computer Science and Engineering Chandigarh University Mohali, India
Corresponding Author Email: [1] bhattrb001@gmail.com, [2] malhotraradhika087@gmail.com, [3] shekharshivam2002@gmail.com, [4] atharvamondhe2003@gmail.com, [5] abhishekbudhiraja5@gmail.com, [6] shivani.e16453@cumail.in

Abstract— **Reinforcement learning (RL) has emerged as a transformative and game-changing method which offers powerful techniques for creating adaptive and intelligent behaviors to enhance gameplay experience. In this research we focus on brief history of Reinforcement learning and its applications in game development. In games we look deeper on its role in crafting responsive agents, designing dynamic non-player character strategies (NPC) which includes model-based, model-free. Additionally, the challenges and difficulties of integration in day-to-day games has been discussed. Also, we discuss on what future holds for this in games, how this would improve and enhance game experience. By merging RL with techniques from NLP and computer vision, developers are creating the agents capable of interacting with complex in-game scenarios, thereby expanding the potential experiences within virtual environment. In conclusion, we consider future developments and directions for research in this field, including improved algorithms, scalability, and enhancing ethical considerations as RL continues progresses within gaming development.**

Index Terms*— *Reinforcement Learning, Natural Language Processing, Artificial Intelligence.

I. INTRODUCTION

Reinforcement Learning (RL) has been constantly growing in past few years with new algorithms and techniques coming out every week, becoming a go-to solution for any simple or complex thing that requires interacting and learning from environment. Unlike the tradition supervised learning methods which depend on labeled data, RL enables agents to make decisions and optimize the techniques through a process of trial and error, allowing them to adapt to changing and different scenarios easily. This ability to learn on its own sets it apart from other AI algorithms and makes it well suited for thing such as decision making and other complex tasks. Games are perfect for this. In games decisions are unpredictable and can have many outcomes. So, to test out which thing works and which does not would require an intelligent being and that is where Enforcement learning comes to "play", quite literally

It is better to have a brief knowledge of how RL works in the back of mind. REWARD, ACTION, AGENT, and ENVIRONMENT.

Reward is feedback system that ensures that the path taken that has yielded a good result is given leverage over others so that it is included in future generations too. Agent is something that interacts with environment.

Games provide an excellent platform for testing and improving RL strategies because of their abundance, billions of hours of gameplay, well established rules, objectives, and the need for strategic decision-making. RL is currently in its next generation with thus already being tested on games ranging from board games to chess to complex games with more inputs. The success of prominent projects like, OpenAI's Dota 2, DeepMind's AlphaGo, showcases the RL's potential in complex game situations, emphasizing its transformative and revolutionary impact on the gaming industry.

Here we have gone through the history of RL with major papers and milestones discussed in the next section. The technique is applied using python on a basic snake game. The objective of this classic game is to eat food while avoiding boundaries and the body itself. The objective is to grow as long as possible without dying. Normally, we can get the location of the food and make snake reach there but the path also matters which is given by RL. RL runs on it and gives out the best path possible.

II. II.RELATED RESEARCH WORK

The deep learning algorithms were created in 2006 and since then these have been used by various researchers and industries in recent years.

Name of paper	Year	Authors	Features
Reinforcement Learning in Game Industry—Review, Prospects and Challenges [1]	2023	Konstantinos Souchleris, George K. Sidiropoulos, George A. Papakostas	Deep learning algorithms, Markov Decision problem
The Use of Reinforcement Learning in	2020	Ao Chen, Taresh Dewan,	Deep Q Network (DQN) and

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

Gaming: The Breakout Game Case Study [2]		Manva Trivedi, Danning Jiang	Double Deep Q Network (DDQN) algorithms
A Review Paper on Implementing Reinforcement Learning Technique in Optimising Games Performance [3]	2019	Mohd Azmin Samsuden; Norizan Mat Diah; Nurazzah Abdul Rahman	Q-learning and State-Action-Reward-State-Action (SARSA) method
Cooperative Game Theoretic Approach Using Fuzzy QLearning For Detecting and Preventing Intrusions In Wireless Sensor Network [4]	2024	S. Shahaboddin, P. Ahmed, B. A. Nor, M. K. Laiha and A. Ajith	Game theory principles applied to Intrusion Detection and Prevention System (IDPS)
Deep reinforcement learning with experience replay based on SARSA [5]	2015	Dongbin Zhao; Haitao Wang; Kun Shao; Yuanheng Zhu	SARSA, Q learning

[6] In 1989, Q learning was proposed by Watkins. Q learning is popular for RL-based agents. However, it is not useful for the complicated problems. The methods containing Q learning was brought forward in many papers.

[7] Koray et al. were successful in training a deep reinforcement learning agent from various visual inputs having many pixels. Previous RL methods have had trouble in various areas which includes feature extraction, whereas deep RL has been successful in handling various complex tasks because it learns from the data at different levels of the features.

[8] Istvan Szita et al. were able to implement Reinforcement Learning algorithms and the additional ideas of placing domain knowledge for scaling up. All the potentials, challenges and limitations are listed in the paper. More in-depth reviews of different games and solution approaches are provided.

[9] Marc Lanctot et al. served a paper in which an overview of the terminologies, code base, algorithms and core-concepts across the field of Reinforcement Learning and computational game theory. They told uses of OpenSpiel. It is collection of algorithms and environments for research in RL and planning in games.

[10] Lukasz Kaiser et al. explored the video prediction models enables agents to solve Atari games with lesser interactions than model-free methods. They described SimPLe (Simulated Policy Learning), a model-based Reinforcement Learning algorithm and did comparison of various models.

[11] Yaodong Yang et al. presented the paper which was about MARL (Multi-agent Reinforcement Learning) which included basic knowledge of MARL, basic solutions, and the limitations. They provided a assessment of current state of art MARL algorithms from a game theory perspective.

III. METHODOLOGY

A. Implementation is done as follows:

Equation for new generation

NewQval = W(s,a) + α[R(s,a) + γmaxW'(s',a') – W(s,a)]

Where, NewQval is updated-value, W(s,a) is the value that has been initialized , α is the learning rate, R(s,a) is function that deals with reward , γ is discount rate, and W'(s',a') is maximum of the the function

Setting up the environment:

Select a game where algorithm must be applied. Here classic snake game is selected. Make sure the code of the game is there because the code must be tweaked for future generations. Here how the game looks:

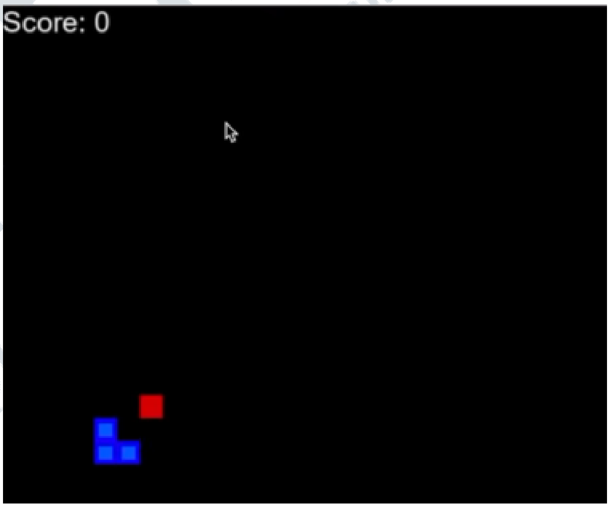

Figure 1. Game's Look

The blue is the snake and red is the food. For every food that snake eats it grows by some length The goal is to grow as larger as one can without dying.

One can die by touching the boundary or touching the snake body itself.

To implement this, an agent.py model, the RL model and the main game is taken. The objective of the agent is to interact with the main game and give commands and run future generations.

The game which has these basic movements ts taken.

[0,0,1]

[0,1,0]

[1,0,0]

These movements define left right and down. First, the is

IFERP
ISSN (Online) 2394-2320
International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 11, Issue 12, December 2024

give a normal direction and then let the game run. The high score, length and other variables is taken care of.

After the end is reached, new generation is trained on the results from previous and agent learns to play the game slowly

B. Leaning by trial and error:

As already discussed, this is a repetitive process and after running for generations agent has a better hold of game and can eventually play it for long time all by learning that by itself.

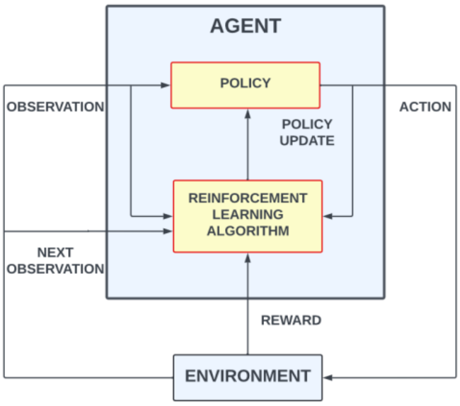

Figure 2. Agent and Environment

C. Algorithms of Reinforcement Learning:

Q-Learning:

This is an algorithmic technique in which a value function is used to get the value of a specific action that has been performed in some state. After this, the agent looks for the best result by optimizing the Q value which was initialized earlier

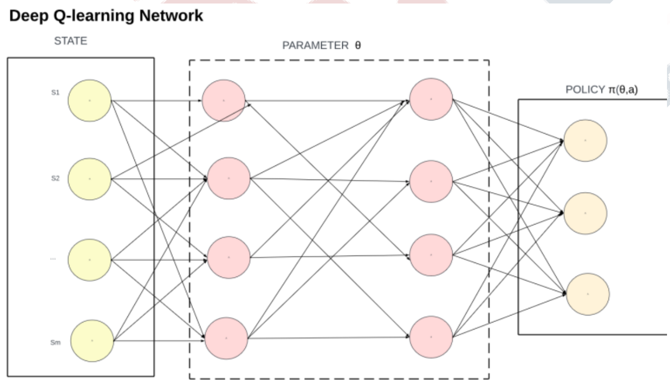

Figure 3. Above implemented learning network(Q)

This learning approach is used instead of traditional ML approach.

The two main components in RL are agents and its environment. The agent makes a strategy to take decisions. This strategy is called a policy. The agent can predict the best action. A Q-table is a matrix that has a relation with the state of agent with all the actions it can take. The values in the table are taking as a measure of cumulative award. Deep Q-Learning expands the potential of Q-Learning by transforming the table into Deep Neural Network. The Bellman equation is used to update the Q-values:

Initially Q Value is generated randomly and basic on this the game is played and then new q value is generated. This keeps on happening.

For the game following system is there:

Reward=+10, for eating food

0, for nothing happening

-10, for death

Using these new Q values are calculated

For the model that we trained we reached the following efficiency for the following constraints

MAX MEM=10mb

BATCH SIZE=1000

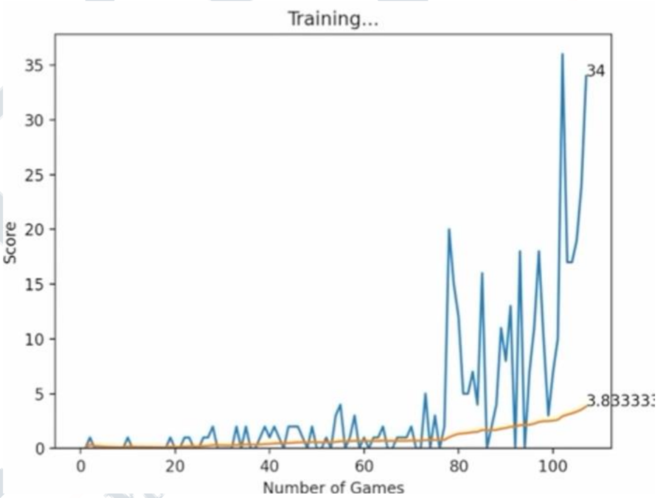

Figure 4. Graph for score and no. of games

The deep neural network tries to maximize the reward. For this it optimizes the action(output) for a state(input). The loss function is used to find out that how good the prediction is in comparison to the truth. The job to minimizes the loss is done by the neural network.

IV. FUTURE SCOPE

AI Personalization: RL can help in creating more depth characters with more emotions in them. It can also produce personalized gaming environment by giving NPCs the level of expertise depending upon the skill of the player playing. Also, it can produce more tailored and intelligent games as it can mimic actions and thus by producing agents or NPCs that are somewhat intelligent

Optimization of game development and testing: As now these systems can play games, this can save a lot of money in testing. Also, it can be set to some random setting to discover random glitches in games which can be helpful in customer retention as well as optimizing the game further

Merging with other AI algorithms: Combining Reinforcement Learning with various approaches and

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)

Vol 11, Issue 12, December 2024

methodologies like CV (Computer Vision) and NLP (Natural Language Processing) can enhance the experience of the user. This merging can help in enhancing and elevating the experience even more

Quality assurance and game testing using AI: The reinforcement Learning agents can help in simplifying game testing and quality assurance as they are trained to fix bugs or design flaws. Future researches can help in developing RL agents that can find potential issues and offer solutions to the bugs.

Improving multiple players and social dynamics: For multiplayer games this is a boon because not all games get the same user base which further hiders other people from playing. This a toxic cycle and this can be broken by using these bots which can play equally good as other humans

V. CONCLUSION

Reinforcement Leaning is recognized as a powerful technique for training agents in gaming environment. This research focuses on RE algorithms with focus on implementing one of the new techniques called the Q. Through a structured literature review, experimental analysis, and methodology, we have deep dived into the potential, challenges, and future opportunities of applying the algorithms of reinforcement learning in game development. Our research showed about the capabilities of Reinforcement Learning algorithms like Deep Q-Network in the game environment. The research highlighted that the reinforcement leaning agents which can improve their performance by the interaction with the environment. One of the challenges for this model include the scalability factor. As the game's complexity rises, the action and state spaces can grow. The scalability issue would hamper the ability of reinforcement learning algorithm to efficient learn from entire space. In addition to it, reward designing is also a challenge. Designing a reward structure is complex. To avoid unintended consequences, rewards must be designed in a particular manner. Ineffective design might decrease the efficiency of learning process and would fail to meet broader objectives. Moreover, in complex environments, training the Reinforcement Learning requires more time and computational resources. There is sometimes high resource requirement which can restrict the feasibility of sophisticated models. The paper gives the knowledge about how reinforcement learning techniques can be used for development of games.

REFERENCES

[1] Konstantinos Souchleris, George K. Sidiropoulos and George A. Papakostas, "Reinforcement Learning in Game Industry—Review, Prospects and Challenges" 2019

[2] Ao Chen, Taresh Dewan, Manva Trivedi and Danning Jiang, "The Use of Reinforcement Learning in Gaming: The Breakout Game Case Study", 2020

[3] Mohd Azmin Samsuden; Norizan Mat Diah; Nurazzah Abdul Rahman, "A Review Paper on Implementing Reinforcement Learning Technique in Optimising Games Performance", 2019

[4] Shahaboddin Shamshirband, Ahmed Patel, Nor Badrul Anuar, Miss Laiha Mat Kiah, Ajith Abraham, "Cooperative Game Theoretic Approach Using Fuzzy QLearning For Detecting and Preventing Intrusions in Wireless Sensor Network", 2014

[5] Dongbin Zhao, Haitao Wang, Kun Shao and Yuanheng Zhu, "Deep reinforcement learning with experience replay based on SARSA", 2017

[6] Christopher J.C.H. Watkins & Peter Dayan, "Machine Learning", 1992

[7] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, "Human-level control through deep reinforcement learning", 2015

[8] Istvan Szita, "Reinforcement Learning for Games", 2012

[9] Marc Lanchot, Edward Lockhart, Jean-Bapstiste Lespiau, OpenSpiel: "A Framework for Reinforce ment Learning in Games", 2019

[10] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, "Model-Based Reinforcement Learning for Atari", 2017

[11] Yaodong Yang, Jun Wang, "An Overview of Multi-Agent Reinforcement Learning from Game Theoretical Perspective", 2020