IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Game Solving Through Deep Learning Agents

Durgaram Borker[1*], Teslin Jacob[2]

[1,2]Dept of Computer Science, Goa College of Engineering, Farmagudi, Ponda, Goa, India

DOI: https://doi.org/10.26438/ijcse/v7i5.852855 | Available online at: www.ijcseonline.org

Accepted: 12/May/2019, Published: 31/May/2019

Abstract: Learning to make machine learning agents work on visual inputs has been a hurdle that researchers have faced since the early days of machine learning. There are currently techniques like deep learning and reinforcement learning which can be used in multi-feature environments. These techniques have stood their ground for a long time and have proven to be efficient. Therefore combining these fundamental concepts in order to realize a bigger goal is the best way to get best out of both. The Deep Learning Agent is to be designed using traditional machine learning methods like deep learning, reinforcement learning and deep Q-learning. Hence the agent is able to make the highest rewarding decision the will maximize the agent skill and make the learning process worth-while and efficient.

Keywords: *Reinforcement Learning, Deep Learning, Game Bot, Agent, Action, Environment, Reward*.

I. INTRODUCTION

Solving games is considered interesting as well as the basis of problem solving structure because of the hostile, competitive challenges and agents involving the environment. Thus, developing a game AI opens the path of implementing the problem-solving ideas in real life situations [6]. Beside that deep reinforcement learning has achieved several high-profile successes in difficult decision-making problems [4]. But these algorithms typically require a huge amount of data before they reach reasonable performance [6]. In case of hard-coded bots we do not need these data for making the right move for right situations. Most cases the coded bots use greedy search algorithms for finding the best move possible for a particular state [6]. As an example, in snake's game a bot will try to avoid the walls and find the shortest path for the reward (an apple or a worm). So how can we make a bot understand the same thing by not telling them about the environment, like what the wall is or worm is, just giving them the feedback of what they are doing, which is a way for solving the self-learning feature.

This paper highlights the need of a general artificial intelligence model, which is trained in multiple game environments, which will prove that a single model can be successfully applied to different environments, can be trained on those environments and improve its performance.

II. OBJECTIVES.

1. To design game bots with their game environment.

2. To define values of thresholds, rewards and penalties in the game, that the agent will use to facilitate the learning process.

3. To employ a learning approach that improves the skill of the bot over time.

4. To supply the bot with valuable feedback that it will learn in order to successfully make precise inferences and improve its game-play.

III. PROBLEM DEFINITION.

Machine learning implementations constitute models which are designed using neural networks, and perfected over the time with continuous training with supervised or unsupervised learning. But these models are highly specialized in their respective domains [8] [9], and have nothing to offer when they are taken out of their native operational environment, and integrated into a foreign environment. This happens because of their high degree of specialty in terms of their data inputs. That is, they can only accept the input data in a specialized format which they were trained in, and they are useless in a foreign environment since they cannot make sense of that data. This has given rise to accumulation of machine learning agents in a single (specialized) domain. So we need a model which can accept multiple forms of data given to it, make sense of it, and learn effectively to improve its performance. Hence, the deep learning agent that will be designed will be able to take data from multiple environments we supply it with, and learn to improve its performance in those environments. Hence it will be a step towards achieving general artificial intelligence compared to specialized artificial intelligence that we see today.

IV-RELATED WORK

[1]- This survey talks about importance of games and their contained challenges which have helped them dominate the entertainment market today [1], and how they have found their way into military and industrial applications [1]. They state that games have helped the military accomplish and realize real-world training, in order to improve the skills of soldiers by augmenting games into real-world field exercises [1]. They focus on the impact the games can have on a soldier's skills and rather improve it by adding in the most realistic non-human units that can mimic the human behavior and accuracy [1]. Games have helped military realize realistic battlefield environments that could've otherwise cost so much money, human labor as well as a lot of time to build. A test environment to practice hostage rescue and bomb diffuse scenario which for example, consists of 10 buildings, one bridge a river, and about 7-10 civilians. To achieve this in real world, the cost estimate is about four hundred thousand dollars. But because of game simulations, the cost reduces by about 95% leading to just over 3 thousand dollars for the simulation and game software. Academia remains the domain that is most actively pursuing innovative AI research [1] [6]; however, while considerable research is taking place it is not necessarily focused on applicability to games. Increased efforts have been made in the last decade to increase the interaction between academic AI researchers and industry AI developers [1]. This rapprochement promises to bring accelerated improvements in the integration of advanced AI in games. The principal conclusion is that there is no clear leader in game AI technology either in industry, government, or academia [1]. Commercial games, which might be expected to be at the vanguard of AI development, are not [1]. The reality is that the video game industry, naturally, holds the gamer experience as the highest priority, and all game decisions are made in that context [1]. Consequently, commercial game development has focused on the environment of the game: graphics and character models. The impact is that AI research by the video game industry is meager and only a small portion of the computational resources in a game are allotted to AI [1] [5].

[2]- This paper introduces two novel algorithms for learning with the help of humans, according to the rewards that they will provide [2]. The specialty of these algorithms is that instead of viewing feedback as a reward with a numerical value, they view it as a discrete signal provided by the human trainer. For example, indicating lack of improvement by giving lack of feedback. A significant body of work exists on the problem of learning from human trainers, and specifically on the problem of learning from trainer-provided feedback [2]. Here, they focus on learning from demonstration, which uses human-provided examples of a target behavior, while training on feedback always works on maximizing the numerical reward [2] [4] [6]. While there have been exciting developments in both areas, they argue that none of the models are apt for training the agent efficiently. First, providing demonstration is always better and apt for effective learning. Second, the positive or negative discrete feedback that agent gets from the human trainer is not treated like a numerical reward [2] [6]. Initially we argued that most existing work on learning from feedback, which treats trainer feedback as a numerical reward, is not always sufficient for describing the ways in which human trainers provide feedback. They presented empirical data indicating that humans deliver discrete feedback and follow different training strategies when teaching.

They have developed two Bayesian learning algorithms, SABL [2] and I-SABL [2] that can leverage knowledge about those strategies. SABL [2] encodes assumptions about trainer strategies as the probabilities of explicit feedback given the correctness of actions, and I-SABL infers those probabilities online [2]. The user studies and simulation experiments demonstrate that the SABL and I-SABL algorithms learn in substantially fewer episodes, and with less feedback, than algorithms modeled after existing numerical-reward-maximizing algorithms [2]. Further, they have shown this approach can be effective both in contextual bandit and sequential decision making domains. They also indicated that Future work would expand the space of feedback strategies considered by SABL and I-SABL to allow temporal delays and variable feedback distributions, and to incorporate knowledge from trainer demonstrations [2]. Future work would also consider how these algorithms can be applied to sequential tasks without enumerating possible reward functions [2], allowing them to be used in more complex domains.

[3] This paper focuses on Construction of game agents through simulated evolution [3]. Simulated evolution is concept that trains agents based on the process of natural selection. This is also possible with the help of Evolutionary computation by Eiben and Smith 2003 at al [3]. EC is a machine-learning technique that can be applied to sequential decision-making problems with large and partially observable state spaces, like video games [3]. EC calculates solution(s) to a problem in the following manner: Initially, a random collection of candidate solutions, called the population [3], is generated and evaluated in a task within some environment. Because of randomness in how the population was generated, there will be variation in the performance of diff erent candidate solutions [3]. At this point a new population is generated from the old population using a mixture of selection, recombination, and mutation Selection is based on Darwin's concept of natural selection by Darwin et al. [3] by which fitter individuals enjoy higher reproductive success. Recombination [3] gives new solutions by combining the fitter solutions from older population. Mutation [3] operations are applied with low probability and generally result in small changes to a candidate solution. The new population of candidate solutions is labeled the next generation of the evolutionary process [3]. The new population now also undergoes evaluation and is subject to selection, recombination, and mutation, which lead to yet

another generation, and so on. This paper concludes by stating that Evolutionary computation is a powerful machine learning technique that has been used to discover skilled and interesting agent behavior in many domains.

[4] This paper talks about using Reinforcement learning in order to train web spiders to navigate the hyperlinks more efficiently. This paper argues that the training of web spiders gives best results when they are trained through reinforcement learning [4] [6], a type of machine learning paradigm which involves decision making and observation of future rewards. One advantage of reinforcement learning is that, it provides a way to estimate the outcome of an action with respect to the future. They have presented an algorithm which will map hyperlinks [4] to the words searched by the user using a naive Bayes text classifier [4]. Spiders are the agents which are trained to explore the web pages on the internet in order to index the links and display them on the search engine [4]. when a query pertaining to that topic is typed on the search engine. Spiders are becoming key elements in the creation of Web-based knowledge bases [4]. The key feature that proves that topic specific spidering is best solved with reinforcement learning is that there is the involvement of an time delayed reward [4] [6] for which reinforcement learning is known to produce the best results. The results provide strong evidence that reinforcement learning is an excellent framework within which to perform Web spidering. Experimental results on two data sets show a three-fold improvement in spidering efficiency over traditional breadth-first search [4].

[5] This paper talks about conducting a Turing test between a human player and a computer game bot and seeing whether it is possible to distinguish correctly between the two. The primary goal of the test is to check whether the bot passes the Turing test and is successfully able to deceive the judges into thinking that he (the bot) is a human player [5]. Here, they have considered a popular game of the time called Unreal tournament [5], a first person shooter game in which a player has to navigate across a virtual world, and kill enemies with the weapon he is provided with. They took 3 human players and 3 bots for the test, and arranged a panel of judges who would decide whether a particular player was a human or a computer bot. the game was held and results were calculated, and they came to an conclusion. Computer bots cannot play like humans-not yet [2], and are easily distinguishable because bots always displayed a radically different behavior than the human players [5] [7]. However, this can be improved with behavior modifications in the programming of the bots so that they can display higher imitations with humans in their game play.

[6] This paper talks about using deep reinforcement learning to carry out learning in parameterized space with continuous values. This paper extends the Deep Deterministic Policy Gradients (DDPG) algorithm by Lillicrap et al, [6] [9] into a parameterized action space [6]. They have modified the original algorithm and added new features, like bounding action space gradients [6]. Deep, model-free RL in discrete action spaces can be performed using the Deep Q-Learning method introduced which employs a single deep network to estimate a maximally valued output for a specific input [4] [6]. Several variants of DQN have been explored. These networks are highly accurate in continuous state spaces but suffer in continuous action spaces, as they output Q-values [4] [6]. Half Field Offence [6] bounds the values to a range of [-180-180], but training the network would frequently force the values to go out of bounds. This was due to the critic that always encouraged the actor network to increase the values even if they went out of bounds. They explore three approaches for limiting parameters in their intended ranges: Zeroing Gradients [6] [9], Squashing Gradients [6], and Inverting Gradients [6]. This paper has shown that deep reinforcement learning can be used in continuous action spaces like games. The work represents a step towards fully learning complex agents [4] [5] [6] [8]. More generally they have demonstrated the capability of deep reinforcement learning in parameterized action space.

[7] This paper talks about considering imitation as a mechanism to make computer bots behave like humans [5] [7]. Imitation is a powerful and pervasive primitive underlying example of intelligent behavior in nature. Can we use it to make agents behave in the way like humans? [5] [7] This question is studied in the context of a Turing-like test where computer game bots compete by attempting to fool human judges into thinking they are just another human player [5]. The major differing factor faced by the agent, is navigating like humans do. Building robots that act human requires solutions to many challenging problems, ranging from engineering to vision and natural language understanding. Imitation is a powerful and pervasive primitive in animals and humans [7], with recently discovered neuro-physiological correlates [7]. Children observing adult behavior are able to mimic and reuse it rationally even before they can talk [3] [7]. As robotics platforms continue to develop, it is becoming increasingly possible to use similar techniques in human-robot interaction The Human Trace Controller [7] presented in the paper that, we can use the human traces, in order to improve the navigation policies [5] [7] that the agent adapts in its training However, the technique is much more generally applicable and can be extended to further improve the navigation system, to generalize to previously unseen environments, and to support higher-level decision making such as opponent modeling [7]. One further area where human trace data can be useful is to improve other components of the navigation subsystem. Some areas of game levels may be missing nav-points [5] [7] or edges in places where they would be quite useful to bots. The Human Trace Controller component takes a step towards building human-like behavior [5] [7] in a complex virtual environment by directly replaying segments of recorded human behavior. The resulting behavior appears

smooth and human-like on observation, while also allowing the bot to navigate the environment with a minimal number of failures

V. CONCLUSION

The project aims at developing a model to solve 2D and 3D games with Deep reinforcement learning approach. It will introduce a new deep learning model to play games like cartpole, mountain car, breakout etc, using OpenAI gym, as well as Tensorflow libraries and Keras API. It will showcase all the epochs and the average improvement in the scores in a comprehensible format such as a graph to better understand the efficiency and convergence supported by deep learning.

REFERENCES

[1] M.A.J.Bourassa and L.Massey, "*Artificial Intelligence In Games A Survey Of The State Of The Art"*, Defence Research &Development Organization Canada.(DRDC). Technical Memorandum, pp.3-40. DRDC Ottawa TM 2012-084 August, 2012.

[2] Robert Loftin, Michel.L.Littman, Jeff Huang, "*A Strategy Aware Technique for Learning Behaviours from Discrete human feedback*", Berckely Brown Association for the Advancement of Artificial Intelligence, pp 1-5. June, 2014

[3] Jacob Schrum, Risto Miikkulakainen. "*Constructing Game Agents through Simulated Evolution."*, In Encyclopedia of Computer Graphics and Games, pp.1-10 March, 2016 Springer.

[4] Jason Rennie, Andrew McCallum. "*Using Reinforcement Learning To Spider The Web Efficiently"*, ICML proceedings of sixteenth International Conference on Machine Learning, pp.334-345. June 27-30, 2009.

[5] Philip Hingston Senior Member IEEE. "*A Turing Test For Computer Game Bots"*, IEEE Transactions On Computational Intelligence in AI in Games, pp.1-18. September, 2009.

[6] Matthew, Peter Stone. "*Deep Reinforcement Learning In Parameterized Action Space"*, ICLR International Conference on Learning Representations, pp.143-155. Feb 2016.

[7] Igor. V. Karpov, Jacob Schkrum, Risto Miikkulakainen. In Philip.F.Hingston,*"Belivable Bot Navigation via Playback of Human Traces"*, Believable Bots, pp.151-170. 2012. Springer.

[8] N.S.Lele, "*Image Classification Using Convolutional Neural Network*". International Journal of Scientific Research in Computer Science and Engineering Vol.6, Issue.3, pp.22-26, June, 2018.

[9] A. Deepa, E. Chandra Blessie. "*Input Analysis for Accreditation Prediction in Higher Education Sector by Using Gradient Boosting Algorithm*". International Journal of Scientific Research in Network Security and Communication. Vol-6, Issue-3, June, 2018.

Authors Profile

Mr. Durgaram Borker pursed Bachelor of Engineering in Information Technology from Goa College of Engineering and is currently persuing Master of Engineering in Computer Science from Goa College of Engineering. His main research work focuses on Artificial Intelligence, Machine learning, and Game Solving strategies.

Prof. Teslin Jacob pursed Bachelor of Engineering in Computer science from PCCE Goa and Master of Technology in Software Engineering from Manipal Institute of Technology. He has published over 4 research papers in various journals including Springer. He is Currently teaching at Goa College of engineering since 2013, and has over 3 years of industrial experience in reputed companies like TCS and Intel technologies. His main research work focuses on Machine learning, Data mining, sensor networks, Cyber Security, and algorithms.He has 6 years of teaching experience and 3 years of research experience.