TEST Engineering & Management

November-December 2019
ISSN: 0193-4120 Page No. 3004 - 3010

An Examination on Applying Machine Learning in Game Industry

*Jos Timanta Tarigan, Cindy Laurent Ginting, Elviawaty Muisa Zamzami
Faculty of Computer Science and Information Technology, Universitas Sumatera Utara
**jostarigan@usu.ac.id*

Article Info
Volume 81
Page Number: 3004 – 3010
Publication Issue:
November-December 2019

Article History
Article Received: *5 March 2019*
Revised: *18 May 2019*
Accepted: *24 September 2019*
Publication: *14 December 2019*

Abstract:
Machine learning is one of the biggest trends in the current computer industry. However, the implementation of machine learning in commercial games is rare and limited to a certain minor feature in the game. That being said, there are a few implementations of machine learning in game industry that actually works. In this paper, we present our observation result regarding the implementation of machine learning in computer games. The observation focus on the architecture and methods used in the system, and the performance and advantage of using machine learning as part of the system. We go through several commercially available games that implement machine learning and analyze the architecture and performance of the system. Additionally, we also observe a few research related to machine learning that uses game as its environment by collecting several publications related to this area.

Keywords: *Machine Learning, Reinforcement Learning, Neural Network, Game*

1. INTRODUCTION

Machine learning is a subset of artificial intelligence that aimed to create a computer with human-thinking capability. The implementation of machine learning allows computers to effectively perform a specific task relying on patterns and interference and without using explicit instructions. The term "Machine Learning" was coined for the first time in 1959 by A. L. Samuel *(Samuel, 1959)*. His research was focused on developing a computer able to learn the game of checkers.

The popularity of machine learning began in the early '90s. In that era, researchers in automatic text classification shifted from knowledge engineering approach where users have to manually set a certain kind of rules on how to classify a document to machine learning approach where the rules are generated by learning from a set of pre-classified documents. The advantage of having the rules generated automatically significantly reduces the expert labor cost since there is no need for expert intervention during the initial setup.

One of the biggest problems in machine learning is that it requires a learning phase. This phase, depending on the samples fed to the engine, may results differently between each sample. Hence, it is hard to predict the outcome of an event. This condition may be a problem to most games where developers are unable to tweak the difficulty of the games due to this randomness. Another issue in applying machine learning is the lack of knowledge regarding the outcome of the process. Most commercial games are intended to entertain people with a series of solvable

November-December 2019
ISSN: 0193-4120 Page No. 3004 - 3010

challenges. Making it hard or even unpredictable diminishes the purpose of the game.

In this paper, we are going to review several ideas that try to promote the use of machine learning in games. Our focus is to go through several studies regarding the application of machine learning in games, understanding the purpose of each study, and collect the conclusion. Later, we will go through several examples of how the industry implements machine learning in games. The paper will be formed as follows: in the second section, we will go through the basic understanding of Machine Learning and compare it with other AI methods. Later, we will observe through several examples of machine learning implementation in games either in research phase or implementation in the game industry. In the last section, we will describe the conclusion gathered during the test.

2. MACHINE LEARNING IN GAME INDUSTRY

2.1 Artificial Intelligence

Naturally, the part of games that may be affected by the implementation of machine learning is artificial intelligence.Developers use AI to give Non-playable Characters (NPC) interacts with users. Most current video games use a pre-defined script that can be described as a finite state machine or behavior tree. There are a few obstacles that make machine learning unsuitable for AI in games such as unpredictability and lack of control. Games are mostly intended for entertainment purpose and developers tune their games to make it enjoyable. Inability to predict the output of an event will make this tuning process harder and, in some cases, impossible.

Despite the problems previously described, there are a few commercial games that managed to implement machine learning methods as part of their AI component. Creatures is a game that considered to be one of the pioneers on using machine learning as part of its AI. In the game, the player is intended to interact with a newborn creature in a closed environment. Initially, the creature has the capability to see, hear, and touch. Furthermore, the player can teach the creature a simple verb-object language. The objective of the game is to keep the creature survive by giving them a set of instructions to help them explore their world, defending them against other species, and breeding them. The objective can be achieved by giving a set of instructions through keyboard and mouse input which will be treated as an input for the learning AI. The simplest form of input can be performed by giving a tickle for reward and slap for punishment. In the technical paper, Grand et al. described the architecture of Creatures neural-network-based creatures intelligence(Grand, Cliff, & Malhotra, 1997).A single creature (or agent) contains approximately 1,000 neurons grouped into 9 lobes and interconnected through roughly 5,000 synapses.The lobe defines the electrical, chemical, and morphological characteristics of a group of cells.

November-December 2019
ISSN: 0193-4120 Page No. 3004 - 3010

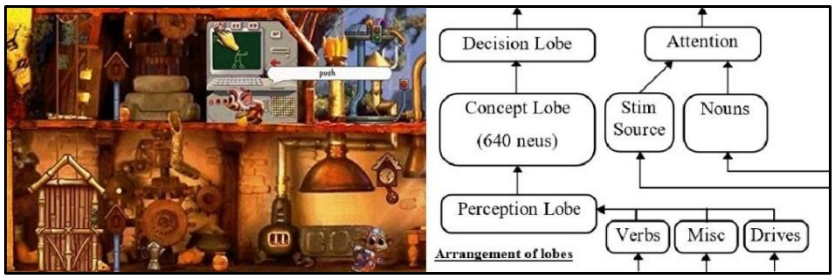

Figure 1. The Screenshot of Creature (left). Image Courtesy of RockPaperShotgun(Alec Meer, 2016). The Architecture the 9 Lobes (Right). Image Courtesy of Grand et al.(Grand et al., 1997)

Supreme Commander 2, a real-time strategy war game, uses multilayer perceptrons (MLP) with three layers of neurons to control the behavior of a platoon when encountering enemy units(Michael Robbins, 2013). The MLP is assigned to decide whether a unit should engage and attack a specific unit or retreat if it found itself unmatched and reevaluate the constantly changing situation. The MLP uses 17 variables as input which mostly contains statistical information regarding units strength in an area such as quantity, health, damage per second, and movement speed. These values are gathered from both friendly and enemy units. The output is mapped into various types of actions that fit the best to the condition. If, for example, the outputs we above a certain value which mean the friendly unit is outnumbered by the enemy, the enemy would by default retreat the battle area. In the training phase, the developer simply made the AI train itself by playing against itself. In the end, the developer is satisfied with the result of the system able to produce the expected actions in any situation.

Another interesting implementation of AI in commercial games can be found in Forza Motorsport series, developed by Microsoft-owned studio, Turn 10collaborating with Microsoft Research (Orland, 2013). The game's AI has a neural-network-based AI, capable of learning and mimicking user's behavior during racing. The learning process starts by registering user's control behavior (accelerate, decelerate/brake, and turn) on each track's features during gameplay sessions. One interesting feature of the Drivatar tech is it uses Microsoft's cloud technology to 'store' the neural network system for each registered user (McMullan, 2014). Hence, the system is capable to process a massive amount of data without affecting the user's system.

2.2 Graphics and Content Generation

Creating content is one of the most resource consuming task in game development. The process may involve many artists and may require massive work-hour. To tackle this issue, developers tend to upgrade their software to increase automation in content creation. In modern content creation application, procedural technique is commonly found to create content with a high degree of diversity. In Hello Game's No Man's Sky (*No Man's Sky*, 2016), the developers claimed users are able to navigate through 18 quintillion procedurally generated planets with different properties. Another interesting implementation of procedurally generated content can be seen in Diablo 3 (*Diablo 3*, 2012)which re-construct the level procedurally each time the player enters the game offering the player new experience on each game session.

November-December 2019
ISSN: 0193-4120 Page No. 3004 - 3010

While procedurally generated content is a promising solution to overcome the cost of content creation, it does come with a drawback; the lack of user's control during the creation process. Granted that the user can control the creation by defining certain rules or template, but it is hard to predict the outcome of the process. Hence, a post-creation evaluation process is sometimes necessary. Yet, evaluating the content generated can be time-consuming and may cost more than creating handcrafted content. An approach to tackle this problem is to implement a learning AI to automate the evaluation process. Search-based procedural content generation (SBPCG)(Togelius, Yannakakis, Stanley, & Browne, 2011) is an example of implementing content evaluation by using a learning algorithm. The method uses an evolutionary search algorithm to explore the space efficiently to find the content appealing to players. Similar to the genetic algorithm, the system relies on evolutionary algorithms where it generates content and grade it using a certain value. The content that fit a certain level of value will be passed and modified to create other content. This process is done repetitively until a content with expected grade is found. There are various other researches that use this method to generate levelin multiple types of game such as 2D platformer (Noor Shaker, Georgios N. Yannakakis, Julian Togelius, Miguel Nicolau, & Michael O'Neill, 2012), first-person shooter (Cardamone, Yannakakis, Togelius, & Lanzi, 2011), and real-time strategy (Togelius et al., 2010).

Despite the great number of researches on learning-based PCG, the implementation of this solution is rare in commercial games. Galactic Arms Race (Galactic Arms Race, 2010) is a game developed by Evolutionary Games that employ content-generating NeuroEvolution of Augmenting Topologies (cgNEAT) algorithm, that evolves game content based on player preferences as the game player. The system consists of NeuroEvolution of Augmenting Topologies (Saravanan & Fogel, 1995) and Compositional Pattern Producing Networks (CPPNs) (Stanley, 2007)to generate new weapons based on a player's preferences.

3. MACHINE LEARNING IN GAME-BASED RESEARCH

Machine learning application in computer games is currently an interesting trend. There are numerous early researches intended to implement machine learning, especially reinforcement learning, in classic computer games (Szita, 2012) such as Chess (Baxter, Tridgell, & Weaver, 2000) and Tetris (Boumaza, 2009). Other research put an effort to create a reinforcement learning in commercial modern games where the input parameter is more complex than classic computer games. One of the most popular game in AI research is Blizzard's Starcraft: Broodwar(*Starcraft: Broodwar*, 1998). Its popularity is due to the availability of BWAPI (Adam Heinermann, 2017). Using this API, Wender et al. observed the suitability of reinforcement learning to micro-managing combat units in Starcraft: Broodwar(Wender & Watson, 2012). The applied techniques are variations of Q-Learning (Christopher Watkins, 1989) and Sarsa (State-Action-Reward-State-Action) Algorithm (Rummery & Niranjan, 1994). The high-level view of the system can be seen in Figure 2 below.

November-December 2019
ISSN: 0193-4120 Page No. 3004 - 3010

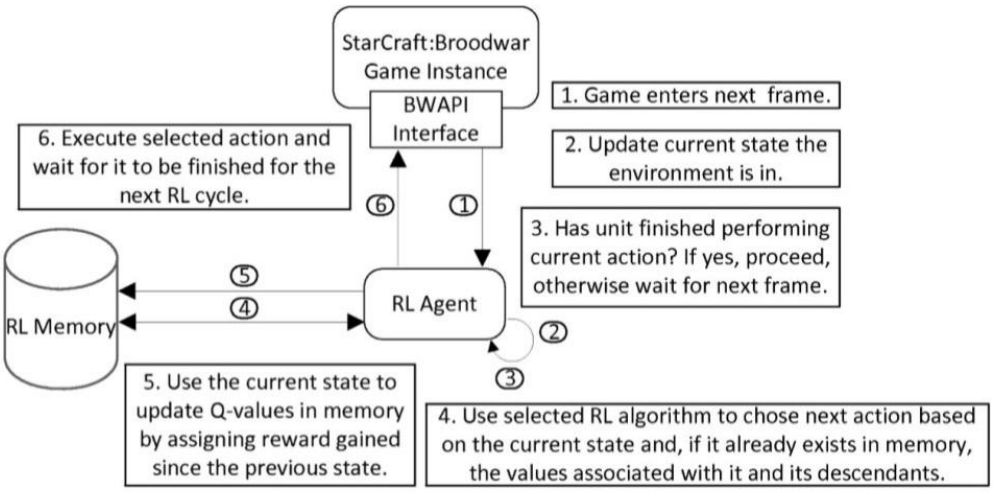

Figure 2. RL Integration with Starcraft: Broodwar. Image courtesy of Wender et al. (Wender & Watson, 2012)

Alaniz developed a deep reinforcement learning AI in Minecraft, an open-world 3D game where player is able to build various objects using blocks(Alaniz, 2018). The AI is given a task to place blocks in multiple colored tiles defined before the game is started. The constraint is to be able to achieve the purpose with 30 actions. The proposed system uses a model-based approach that combines learning a DNN-based transition model with Monte Carlo Tree search. Similarly, Wender et al. Q-learning to develop an AI that can optimally select city sites in Civilization IV, a turn-based strategy game launched in 2005 (Wender & Watson, 2008).

In 2016, a Google-owned AI Company DeepMind announced AlphaGo, a computer program that uses deep neural network as the core of its AI to play a classic board game Go. The system is capable of performing both supervised (using a human expert as its source) and reinforced learning by playing against itself only by knowing the rules game (Silver et al., 2017). Using the AlphaGo engine, the company released AlphaGo Zero, a software using a self-learned version of AlphaGo (no human intervention during the learning process). In March 2016, AlphaGo Zero won 4-1 Go Grandmaster, Lee Sedol (Borowiec, 2016). Figure 3 below describes the overall algorithm of AlphaGo Zero.

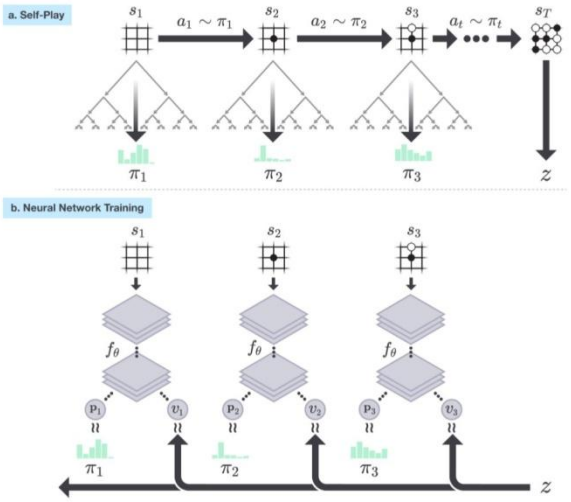

Figure 3. The Self-play reinforcement learning in AlphaGo Zero. Image courtesy of Silver et al. (Silver et al., 2017)

The training process of AlpaGoZero goes as follow: the program repeatedly trains against itself using every position $s_1, s_2, s_3.., s_t$ where s_t represents the terminal position. From each position s, it executes a Monte-Carlo Tree Search using the latest neural network f_θ to find probability π. Using the rules of the games, the terminal position s_t is scored to compute the game winner z. Using the raw position s_t, the AlphaGo Zero updated the neural network parameter θ by

November-December 2019
ISSN: 0193-4120 Page No. 3004 - 3010

passes s_t through many convolutional layers and outputs both a vector p_t that represent probably distribution over moves, and a scalar v_t that represent the probability of winning in position s_t.

4. CONCLUSION

In this paper, we perform an observation regarding the application of machine learning in game industry. Our finding shows that there are a limited amount of commercial games that implement machine learning as part of its system. We also noticed that most commercial games that use machine learning limited its function to a certain insignificant feature that does not affect the core gameplay.In our samples, most games use machine learning to increase the quality of AI. There are also some games that use machine learning to enhance the ability to create content based on the user's behavior.

Interestingly, there are numerous researches that use commercial games as their machine learning research. The complexity of current modern games gives a challenging topic for research. There are also various events that show the domination of machine learning based AI when faced against a professional human player. While the researches successfully implemented machine learning in games; none of them treat a game as an entertainment purpose; they merely use games as the environment of their research. Hence, these researches are not applicable to be implemented in the game industry.

5. REFERENCES

[1] Adam Heinermann. (2017). BWAPI (Version 4.2.0). Retrieved from https://bwapi.github.io/

[2] Alaniz, S. (2018). Deep Reinforcement Learning with Model Learning and Monte Carlo Tree Search in Minecraft. *ArXiv:1803.08456 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1803.08456

[3] Alec Meer. (2016, December 12). Have You Played… Creatures? | Rock Paper Shotgun. Retrieved April 15, 2019, from https://www.rockpapershotgun.com/2016/12/20/have-you-played-creatures/

[4] Baxter, J., Tridgell, A., & Weaver, L. (2000). Learning to Play Chess Using Temporal Differences. *Machine Learning*, *40*(3), 243–263. https://doi.org/10.1023/A:1007634325138

[5] Borowiec, S. (2016, March 15). AlphaGo seals 4-1 victory over Go grandmaster Lee Sedol. *The Guardian*. Retrieved from https://www.theguardian.com/technology/2016/mar/15/googles-alphago-seals-4-1-victory-over-grandmaster-lee-sedol

[6] Boumaza, A. (2009). On the evolution of artificial Tetris players. *2009 IEEE Symposium on Computational Intelligence and Games*, 387–393. https://doi.org/10.1109/CIG.2009.5286451

[7] Cardamone, L., Yannakakis, G. N., Togelius, J., & Lanzi, P. L. (2011). Evolving Interesting Maps for a First Person Shooter. In C. Di Chio, S. Cagnoni, C. Cotta, M. Ebner, A. Ekárt, A. I. Esparcia-Alcázar, … G. N. Yannakakis (Eds.), *Applications of Evolutionary Computation* (Vol. 6624, pp. 63–72). https://doi.org/10.1007/978-3-642-20525-5_7

[8] Christopher Watkins. (1989). *Learning From Delayed Rewards* (King's College). Retrieved from https://www.researchgate.net/publication/33784417_Learning_From_Delayed_Rewards

[9] *Diablo 3*. (2012). Retrieved from https://us.diablo3.com/en/

[10] *Galactic Arms Race*. (2010). Retrieved from http://galacticarmsrace.blogspot.com/

[11] Grand, S., Cliff, D., & Malhotra, A. (1997). Creatures: artificial life autonomous software agents for home entertainment. *Proceedings of the First International Conference on Autonomous Agents - AGENTS '97*, 22–29. https://doi.org/10.1145/267658.267663

[12] McMullan, T. (2014, October 14). Ghosts in the machine: how AI research is bringing

November-December 2019
ISSN: 0193-4120 Page No. 3004 - 3010

game characters to life. *The Guardian*. Retrieved from https://www.theguardian.com/technology/2014/oct/14/ai-research-artificial-intelligence-game-characters

[13] Michael Robbins. (2013). Using Neural Networks to Control Agent Threat Response. In *Game AI Pro: Collected Wisdom of Game AI Professionals* (p. 626). CRC Press.

[14] *No Man's Sky*. (2016). Retrieved from https://www.nomanssky.com/

[15] Noor Shaker, Georgios N. Yannakakis, Julian Togelius, Miguel Nicolau, & Michael O'Neill. (2012). Evolving personalized content for super mario bros using grammatical evolution. *AIIDE'12 Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 75–80. Stanford, California, USA: AAAI Press.

[16] Orland, K. (2013, October 25). How Forza 5 and the Xbox One use the cloud to drive machine-learning AI. Retrieved April 14, 2019, from Ars Technica website: https://arstechnica.com/gaming/2013/10/how-forza-5-and-the-xbox-one-use-the-cloud-to-drive-machine-learning-ai/

[17] Rummery, G. A., & Niranjan, M. (1994). *On-Line Q-Learning Using Connectionist Systems* (No. CUED/F-INFENG/TR166). Cambridge University Engineering Department.

[18] Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, *3*(3), 210–229. https://doi.org/10.1147/rd.33.0210

[19] Saravanan, N., & Fogel, D. B. (1995). Evolving neural control systems. *IEEE Expert*, *10*(3), 23–27. https://doi.org/10.1109/64.393139

[20] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., … Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, *550*, 354.

[21] Stanley, K. O. (2007). Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, *8*(2), 131–162. https://doi.org/10.1007/s10710-007-9028-8

[22] *Starcraft: Broodwar*. (1998). Retrieved from https://starcraft.com/

[23] Szita, I. (2012). Reinforcement Learning in Games. In M. Wiering & M. van Otterlo (Eds.), *Reinforcement Learning* (Vol. 12, pp. 539–577). https://doi.org/10.1007/978-3-642-27645-3_17

[24] Togelius, J., Preuss, M., Beume, N., Wessing, S., Hagelback, J., & Yannakakis, G. N. (2010). Multiobjective exploration of the StarCraft map space. *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, 265–272. https://doi.org/10.1109/ITW.2010.5593346

[25] Togelius, J., Yannakakis, G. N., Stanley, K. O., & Browne, C. (2011). Search-Based Procedural Content Generation: A Taxonomy and Survey. *IEEE Transactions on Computational Intelligence and AI in Games*, *3*(3), 172–186. https://doi.org/10.1109/TCIAIG.2011.2148116

[26] Wender, S., & Watson, I. (2008). Using reinforcement learning for city site selection in the turn-based strategy game Civilization IV. *2008 IEEE Symposium On Computational Intelligence and Games*, 372–377. https://doi.org/10.1109/CIG.2008.5035664

[27] Wender, S., & Watson, I. (2012). Applying reinforcement learning to small scale combat in the real-time strategy game StarCraft:Broodwar. *2012 IEEE Conference on Computational Intelligence and Games (CIG)*, 402–408. https://doi.org/10.1109/CIG.2012.6374183