Making Decisions Like Human, the Application of Deep Learning Algorithms in Game Testing

Xiaokai Wang

Tianjin TEDA Maple Leaf International School, China

Abstract. Artificial intelligence (A.I.) is currently used widely in testing of video games. A.I. had surprisingly performance in board games such as chess and go. How does it perform in modern digital games? Currently, A.I. is at trials of checking the operation of the game and figured out bugs by simulations. However, the algorithms that were applied to the first generation of A.I cannot perform well in current digital games due to completely different constraints. Hence, Deep Learning algorithms are implemented on artificial intelligences or agents that are created for the digital game testing. Deep learning algorithms not only let A.I. can simulate how a human player is thinking when they are playing a digital game, but they also let A.I. can gain experience from billions of simulations based on the extreme speed of calculating. This study would give analysis on performance of A.I. or agents with deep learning algorithms, and show the limitations and barriers of automation of video game testing. Eventually, we explained why automated video game testing is still far away from us.

Keywords: Deep-learning Algorithms; Digital Games; Game Testing; Agents.

1. Introduction

Game, originally indicates activities humans do for fun. As the complexity of games increases, game also indicates the process of players of a game against each other. A.I. has beaten human masters in almost all board games. Then the diversity of modern video games and great complexity of video games make them a great environment to test algorithms. And many artificial intelligences are constructed and trained as a player of games to test them (Serafim et.al, 2017). However, algorithms used in board games such as minimax are not suitable for digital games. This is mainly because the expectation of digital games could not be determined because of the difficulties of quantification of expectation in digital games. Therefore, many new algorithms were invented.

Serafim et.al (2017) has applied Neural Networks (a form of deep learning algorithm) in 3-dimensional first-person shooting game. Serafim et.al (2017) has claimed that the agent is doing what they expect it to operate. Besides, the agent not only learns how to operate the character to kill enemies, but it also learning how to predict how its enemies will act. This demonstrates that the A.I. with Neural Networks has primary consciousness to think easily in the game.

Nevertheless, the limitations of artificial intelligence also show a barrier of automation of game testing. Ariyurek et.al (2019) used an automated agent with algorithms that simulate humans to test a puzzle game. The results showed that the agent performs very similar with human testers in a 9*9 map yet its performance became worse than human testers as the size of the map increases. It is very difficult for artificial intelligences to test games in the current stage. Politowski et.al (2022) has found main issues of existed papers related to automated video game testing. Finally, the time consumption of A.I. increase tremendously when the agent is put into a relatively harder game mode, it takes maximally about 90 lifespans of training to makes the agent performs stably (Serafim et.al ,2017).

This study contains mainly 3 sections. In the literature review part, a brief introduction of game testing criteria and the brief introduction of algorithms used in the agents will be illustrated. Next, we will give analysis on the ideals and limitations in the finding and discussion part. Finally, conclusion is drawn in the last sections.

2. Literature Review

Since the methods of testing video games is different from other applications (Politowski et.al, 2021). Ramadan & Hendradjaya (2014) has proposed testing criteria with nine disciplines (Fig.1).

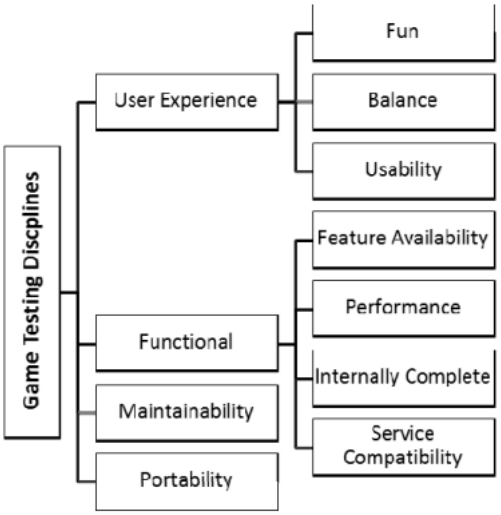


Fig 1. Proposed game testing criteria (Ramadan & Hendradjaya, 2014)

Among this 9 factors, fun, balance and usability belong to user experience. Feature abailablity, Internally complete and maintainability are related to the operation of codes. Performance, Service compatibility and portablility depend on the type of the code that used to develop the game. Ramadan & Hendradjaya (2014) have also claimed that this standard is a combination of two existed software quality model standards, ISO/IEC 25010 & IEEE STD-1061 and Fullerton's model for games. Additionally, Ramadan & Hendradjaya (2014) also emphasized on the three subfactors of user experience.

Currently, artificial intelligences are used as agents to test the operation of the game. Skinner & Walmsley (2019) has proclaimed five A.I. deep learning methodologies, unsupervised learning, reinforcement learning, neural networks, Convolutional Neural Networks and Recurrent Neural Network. These are the mythologies that are mostly used in game testing.

Mohammed et.al (2016) has claimed that in unsupervised learning, data are unclassified and the output is uncertain, hence A.I. has to collect data and analyze them independently. Therefore, A.I. needs to develop criteria of classification to make sure that similar data are gathered. Namratha, & Prajwala(2012) has claim this process clustering.

Skinner & Walmsley (2019) states that reinforcement learning is alike to conditioned reflexes in organisms. The teaching of reinforcement learning is to nurture a certain reaction of A.I. in a specific situation. Therefore, A.I. can make the best choice in similar situations.

Neural Networks are a form of deep learning algorithms (Skinner & Walmsley, 2019). It contains an input layer of neurons, several hidden layers in the middle, and the final output layer of neurons, and each connection between two neurons hold its weight (Wang, 2003) (see in Fig.2).

According to Zehedi (1991), different layers, nodes, connections are built to simulate human's brain. Once data are entered the networks, A.I. will give them a weight to decide whether these are the results it wants (Gurney, 2018). The process of information transporting through the nodes in different layers simulates the judgments that are made by human brains.

Convolutional Neural Networks (CNN) are another expansion of multi-layer Neural Networks. According to Albawi et.al (2017), there are four layers in the CNN, convolutional layer, non-linearity layer, pooling layer and fully connected layer. Convolutional layer separates a whole image to smaller parts in order to reduce the number of parameters (Albawi et.al, 2017). For instance, the total amount of parameters that are input to a neural network without CNN is the resolution of the game times the Framerate Per Second (FPS). And FPS of a digital game is at least 30 (Ramadan & Hendradjaya, 2014), which results in a huge number of parameters that are input into a Neural Network, especially those high-resolution games. Non-linearity layer suppresses the generation of outputs once it reaches

a threshold (Albawi et.al, 2017). Pooling layer tries to compress the size of samples in order to reduce the complexity of the further layers. Then the nodes in pooling layers are connected to every node in the first layer of fully connected layer. Therefore, the fully connected layer is very similar to normal Neural Networks (Albawi et.al, 2017). This algorithm utilizes the number of pixels between blanks in a picture or video as the data enter in the Neural Network (Stutz, 2014). This algorithm plays the roles of visual system and transcript information from images or videos into signals enter the Neural Networks.

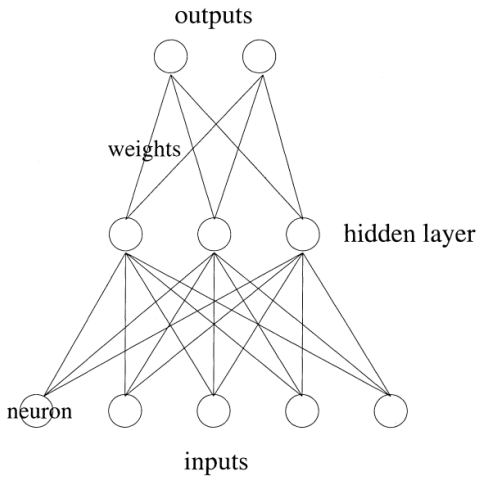


Fig 2. Structure of Neural Networks (Wang, 2003)

Recurrent Neural Network is a modified Neural Network that added a delay layer in the hidden layers (Boden, 2002). The delay layer will result in the calculation of the weights of how the change of values in neurons or weights of connections between neurons will affect the final output. Then it starts to iterate and changing the weight of connections between neurons until the output is what we expected (Boden, 2002). Skinner & Walmsley (2019) have also mentioned that Reecurrent Neural Networks are used in writing and speaking recognition.

Deep learning algorithms have made artificial intelligences being able to transfer images of games into models that could be analyzed. They allow artificial intelligences to play from constrained traditional board games to more flexible multiplayer digital games.

Nevertheless, Politowski et.al (2022) has concluded the four main issues of papers that are about utilizing A.I. and agents to realize the automation of game testing.

1. The digital games used in the researches are not similar with digital games in the society.
2. The description of the application of automation of game testing is ambiguous.
3. The data are collected manually after agents or A.I. finished testing.
4. The source code is not publicized.

3. Finding & Discussions

Artificial intelligence and agents have shown great potentials on debugging the digital games. Skinner & Walmsley (2019) have mentioned that deep learning algorithms are used in many video game agents to examine whether they are complex enough. Politowski (2021) has claimed that game-testers have to challenge themselves to find all the bugs and abnormality of the games. Using artificial intelligent to test video games has simplified the whole testing process. Machado et.al (2018) has pointed that the testers perform better in debugging when assisted by an A.I in dexterity-required games. The potentials of artificial intelligence or agents used in game testing are mainly concentrated at the efficiency of debugging and giving feedback and the reduction of number of testers.

In the stage of initial game testing, the programmers will check the operation of the game to figure out whether the game needs improvements on complexity and debug the whole program. However, not all bugs can be discovered by normal operations. Many of them need players to trigger some special conditions. For instance, bugs could appear in the calculation of collision size and physics engine, which are two main parts in a 3-dimensional game. When two entities are colliding with each other, the models might stuck together and begin to twitch. These bugs are very hard to discover because it does not appear in all collisions of any two entities in the game. It happens mostly between two entities that are not supposed to collide in the normal gaming process. Therefore, testers might not discover all bugs in a video game. And there will be bug reports even a digital game is in the Open Beta. Additionally, the bug reports of players might not be expressed clearly since they can only take a snapshot or using descriptive language yet the programmers are struggling at looking it up in the codes. Utilizing an artificial intelligence to test games can avoid gaining limited data from limited players.

Despite that A.I. or artificial have a great ideal upper limit, the limitations have exposed during the researches Politowski et.al (2022).

Ariyurek et.al (2019) has discovered how a generated humanlike agent could find bugs in a puzzle solving game. In the experiment, humanlike agents perform similar with human testers in a small map. However, when the size of map increases, the performance of agents is worse than human testers. Despite of that, the author has mentioned that the agent uses linear features so it can only find a unique solution of a multi-solution puzzle. Additionally, the agent only used Markov Decision Process-Inverse Reinforcement Learning program in the agents (Ariyurek et.al, 2019). This means that the whole program uses iteration and finally output target data. The amount of calculations increases exponentially when the complexity of the game increases because of the iterative structure of the agent. Similarly, Machado et.al (2018) also concerned about the inconsistency of A.I.'s performance in a greater-size level with more complexity. The consistency of artificial intelligence and agents are not stable due to increase of the difficulty of digital games.

Furthermore, game testing still cannot leave human testers because of the record of data and the discipline of user experience. Politowski et.al (2022) figured that the data are collected by the testers. The difficulty of collecting output of agents or A.I. manually decrease the efficiency of automated testing. Besides, the time consumption in training of A.I. or agents is another major issue (Politowski et.al, 2022). In Serafim et.al's (2017) research, the agent takes 40 lifespans to be trained to play a simple 3D shooter game in an easy mode and the time increase to 70 lifespan in a harder mode. Sometimes it even takes approximately 90 lifespans to get a consistent performance. Modern video games are flexible, players could choose any mode they would like to play. The time consumption of A.I. or agents learning all modes of a complicated modern digital game would be enormously large.

Also, Politowski et.al (2021) has proposed that the criteria of game testing are different from the testing of other types of software. Game testing is particularly focused on the experience of gaming. In Ramadan & Hendradjaya's (2014) survey, players were asked to score a mark on gameplay, graphics, sound, storyline, and players' interactions with a scaling from 1 to 10. This means that the testing needs a lot of feedbacks from players or testers. Virtually, most game companies or studios give priority on the game experience (Ramadan & Hendradjaya, 2014). A stable, glitch-and-bug-free is a standard to measure a good game, yet the creative gamemodes and storylines, epic graphics and immersive user experience can bring more fun to players (Ramadan & Hendradjaya, 2014). And thus, factors that determine user's experience when playing the game are very conclusive. However, even though deep-learning algorithms can stimulate the thinking process of a human, agents are unable to give comments or feedback on any of these disciplines. In other words, A.I. and agents cannot determine whether a game has fulfilled most players' genuine requirements on the game.

In a nutshell, similarly with what Politowski et.al (2022) has claimed, the automation of game testing by A.I. or agents with deep-learning algorithms is still far-fetched nowadays. The papers done to test automated game testing lack practicality, on the other hand, the increase of number of papers

study about automation of game testing (Fig.3) has shown that automation of digital game testing is ideally feasible.

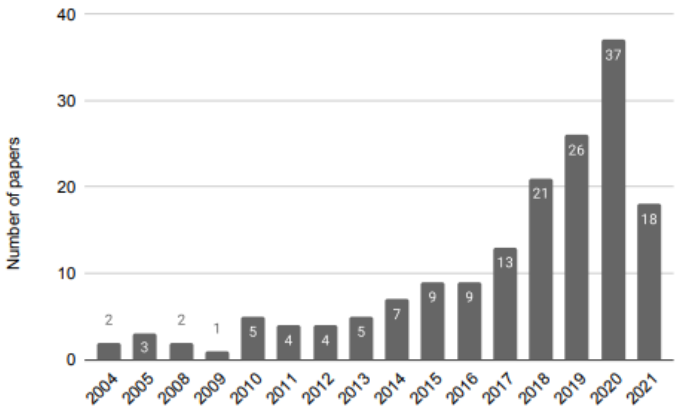


Fig 3. The histogram of papers that are related to "automated video game testing" (2004-2021) (Politowski et.al, 2022)

4. Conclusion

In conclusion, deep-learning algorithms can ideally automate the whole process of game testing and debugging. However, current algorithms cannot support the whole process of game testing and do not perform exceptionally well in existed researches.

This paper is an exploratory study that researches the application of deep-learning algorithms in game. The performance of A.I. or agents with deep-learning algorithms in the reviewed papers is not quite well yet shows a great potential for game testing. Current methodologies applied to A.I. need improvements to adapt a more complicated, flexible gaming circumstance. However, this study has a limitation that it only discussed several types of deep-learning algorithms that are mostly used in the A.I.

The results of literature review show a special standard for video game testing and the principles of deep-learning algorithms, but the results from the researches are not surprising. In addition, the researchers lack the confidence of the application of deep-learning algorithms in the real life.

We recognize that the application of deep-learning algorithms in video game testing is actually far from the automation of video game testing. Especially on the practicality of testing, the algorithms ought to be improved the flexibility of the A.I. so it could reacts faster when facing more complex games. Yet the emergence of researches so far in this subject indicates that we are not supposed to just focus on the current failures of automated game testing system.

References

[1] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In 2017 international conference on engineering and technology (ICET) (pp. 1-6). Ieee.

[2] Ariyurek, S., Betin-Can, A., & Surer, E. (2019). Automated video game testing using synthetic and humanlike agents. IEEE Transactions on Games, 13(1), 50-67.

[3] Boden, M. (2002). A guide to recurrent neural networks and backpropagation. the Dallas project, 2(2), 1-10.

[4] G. Skinner and T. Walmsley, "Artificial Intelligence and Deep Learning in Video Games A Brief Review," 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), 2019, pp. 404-408, doi: 10.1109/CCOMS.2019.8821783.

[5] Gurney, K. (2018). An introduction to neural networks. CRC press.

[6] Machado, T., Gopstein, D., Nealen, A., Nov, O., & Togelius, J. (2018, July). Ai-assisted game debugging with cicero. In 2018 IEEE Congress on Evolutionary Computation (CEC) (pp. 1-8). IEEE.

[7] Mohammed, M., Khan, M. B., & Bashier, E. B. M. (2016). Machine learning: algorithms and applications. Crc Press.

[8] Namratha, M., & Prajwala, T. R. (2012). A comprehensive overview of clustering algorithms in pattern recognition. IOSR Journal of Computer Engineering, 4(6), 23-30.

[9] Politowski, C., Guéhéneuc, Y. G., & Petrillo, F. (2022). Towards Automated Video Game Testing: Still a Long Way to Go. arXiv preprint arXiv:2202.12777.

[10] Politowski, C., Petrillo, F., & Guéhéneuc, Y. G. (2021, May). A survey of video game testing. In 2021 IEEE/ACM International Conference on Automation of Software Test (AST) (pp. 90-99). IEEE.

[11] Ramadan, R., & Hendradjaya, B. (2014, November). Development of game testing method for measuring game quality. In 2014 International Conference on Data and Software Engineering (ICODSE) (pp. 1-6). IEEE.

[12] Serafim, P. B. S., Nogueira, Y. L. B., Vidal, C. A., & Neto, J. B. C. (2017, November). Towards playing a 3d first-person shooter game using a classification deep neural network architecture. In 2017 19th Symposium on Virtual and Augmented Reality (SVR) (pp. 120-126). IEEE.

[13] Stutz, D. (2014). Seminar report: understanding convolutional neural networks. Fakultät für Mathematik, Informatik und Naturwissenschaften Lehr-und Forschungsgebiet Informatik VIII.

[14] Wang, S. C. (2003). Artificial neural network. In Interdisciplinary computing in java programming (pp. 81-100). Springer, Boston, MA.

[15] Zahedi, F. (1991). An introduction to neural networks and a comparison with artificial intelligence and expert systems. Interfaces, 21(2), 25-38.