The Combination of Flappy Bird and Reinforcement Learning Using Q-learning

Xinlu Dong [1,*,†], Xiaole Fan [2,†], Chenhao Sun [3,†], Zhengyang Xu [4,†]

[1] Xi'an Maple Leaf International School, Xi'an, China

[2.] Clifford International School, Guangzhou, China

[3.] High School Attached to Northeast Normal University, Jilin, China

[4.] Legacy Christian Academy, Saskatoon, Canada

* Corresponding Author Email: 2019121056@students.mapleleafedu.com

[†] These authors contributed equally.

Abstract. The development of artificial intelligence has expanded application fields gradually. In recent years, the combination of artificial intelligence and games attracted much attention. In this case, reinforcement learning is often chosen as an effective method to let the computer play the game by itself. In this study, the Q-learning algorithm from reinforcement learning was applied to Flappy Bird. There are the important factors of Q-learning, including state (S), action (A), reward (R), policy (π), time (t), Epsilon Greedy policy and Q-table. After that, a python class called "bot" was used, and it is used as an intelligent agent in the project. In order to implement the Q-learning algorithm, the state of each element of the game was adjusted continuously through the "mainGame" function of the Flappy Bird game. Finally, the survival reward was set to 1 and the death reward to -1000 to increase the survival rate. In addition, coins with different reward values were added to increase the difficulty of training. After training, the survival rate of the bird is improved, and it is clear that the reward value of gold coins will affect the agent's choice tendency. To combine artificial intelligence and games means that computers can be trained to deal with the complex and changing situations in games, and the progress will affect the application of artificial intelligence in real life more deeply.

Keywords: Flappy Bird, Reinforcement Learning, Q-learning, Artificial Intelligence.

1. Introduction

Artificial intelligence is a branch of computer science that attempts to make machines understand human intelligence and respond similar behaviors [1]. The researches about this field include robotics, language recognition, image recognition, natural language processing, and expert systems. Since its inception, artificial intelligence has been an industry with unlimited potential. When the theory and technology of artificial intelligence grow more advanced, the application of it is also expanded. Applying artificial intelligence to video games is another area that is full of developing prospects. Currently, e-sports is a popular activity with large commercial value. E-sports players need a lot of game training every day. However, the labor cost of its training is huge, because it is very difficult to hire a sparring partner or coach whose strength is comparable to the top professional players. If artificial intelligence is used as a sparring partner, a lot of money and time can be saved. What is more, we can train a "sparring player" with comparable strength by collecting the data of the world's top e-sports players.

Reinforcement learning is very suitable for the situation. It is a branch of machine learning, which mainly uses agents to interact with the environment to learn policies that can achieve good results [2]. In fact, combining reinforcement learning with games has yielded a lot of achievements. In 2015, the British company DeepMind proposed a reinforcement learning Deep Q-network based on deep neural network, which achieved a game level comparable to that of humans in 49 Arcade games such as Space Invaders, Arkanoid, and table tennis [3]. Microsoft also lead a project using deep reinforcement learning to train an artificial intelligence to play the four-player board game Mahjong [4].

In this project, we chose Flappy Bird as the training object and trained it through the Q-learning algorithm. This is because the game mechanism of Flappy Bird is not too complicated, which is convenient for us to perform multiple training. Secondly, the reward mechanism in the Q-learning algorithm allows the "birds" we trained to learn how to achieve high scores and avoid death, thereby increasing the survival rate [5]. In addition, we added a new "Gold Coin System" on top of the original game, which increase the difficulty and fun of the game. In this game, the important hypothesis of the research question is that the possibility that the intelligent agent chooses to hit a gold is proportional to the reward value, since the agent might not want to take risk of crashing into an obstacle when the reward might not be able to make up the punishment.

Here is how reinforcement learning and Flappy Bird were connected. First, a code was downloaded from a blog in GitHub [6]. In this code, Q-learning and reinforcement learning were used to train the intelligent agent to play the game Flappy Bird. The agent has to avoid crashing into the ground and pipes in order to gain points, which is the reward, when it is playing Flappy Bird. Otherwise, the bird character dies and ends a round of game, and a punishment is given to the agent. Also, in order to make the intelligent agent learn more efficiently, if the agent crashes into an obstacle at the same place as before, it would get more punishment.

In the second step, an additional reward is added in. Some "gold coins," which the original Flappy Bird game did not contain, were added in to be an extra condition for an intelligent agent to gain more points. Meanwhile, positive values were assigned to the coins, so the agent was able to get positive feedback from crashing into them, instead of getting rewards from a single condition. Also, during the experiments, different reward values for the coins were assigned to see the difference between their results. Finally, through this experiment, we made the computer have the intelligence to play Flappy Bird by itself successfully. At the same time, it also have own judgment in facing the different rewards. The high scores obtained by the computer also proved the applicability of artificial intelligence in games. It can be used to detect potential problems and difficulties in the game to help designers design better games to adapt the common players' game level.

2. Methodology

2.1. Introduction for Flappy Bird

In 2013, with the gradual maturity of game development technology, the content of the game became more and more diverse to gain more popularity. In such an environment, Dong Nguyen, a Vietnamese man, designed a very concise game in a period of two or three days and obtained a huge success. In January 2014, the game became the most popular free application on iTunes in the United States and China. This game was even called as the "Drug of the App Store [7]."

The gameplay includes controlling the bird to fly and avoiding the green pipes; if the bird hits an obstacle, the game will end. Each time the bird flies through a set of pipes, the player earns a point. The game is known for its difficulty, with most players scoring in the single digits [8].

Because of its difficulty, Flappy Bird was criticized by The Huffington Post, stating that the game was a "crazy annoying, difficult and frustrating game" that "combined a super steep difficulty curve, poor boring picture quality and blunt action [9]." However, with the help of a methodology called Q-learning, the bird can easily get hundreds of points, and it can even become immortal.

2.2. Employed Q-learning Model

Q-learning is an off-policy and value-based method that uses a temporal difference learning method and approaches to train its action-value function [10]. There are several important elements concerning the Q-learning. They are state (S), action (A), reward (R), policy (π), time (t), Epsilon Greedy policy and Q-table. The aim of using the Q-learning is to help the agent get the highest score possible. After the agent select and execute an action, it will get a reward. The sum of rewards is:

$$U_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \cdots \quad (1)$$

Where U_t represents the sum of rewards, R_t is the reward in different t, and γ is the discount factor that can help the agent to become more visionary instead of just focusing on immediate interests.

The agent will tend to choose actions that can lead to a higher total reward. The Epsilon Greedy policy is the basis for choosing an action, which can make sure that the algorithm can use the known and explore the unknown. To illustrate the process of Q-learning: in the initial S, the agent chooses an A using a policy derived from the Q-table and the Epsilon Greedy. Then, the agent takes action A and observes R and S' (a new state). Finally, Q(S, A) is updated. The process repeats. The formula is:

$$Q^{new}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \times [r_t + \gamma \times Q^* - Q(s_t, a_t)] \quad (2)$$

Where α is the learning rate, Q^* is the estimate of optimal future value, and $[r_t + \gamma \times Q^* - Q(s_t, a_t)]$ is the temporal difference.

To implement the Q-learning algorithm, a Python class called "Bot" was used. This Python class represents the intelligent agent in this machine learning project. The "mainGame" function, which runs the Flappy Bird game, continuously calls the "act" function in the "Bot" class. The state of the game includes the position of the bird, the position of the pipes, and the vertical velocity of the bird. Using a call to the "act" function, the state of the game and whether a coin is obtained are given to the intelligent agent. Then, the agent will make a choice of whether the bird will flap by checking the q-values corresponding to the state of the game. If the value of flapping is larger than the value of not flapping, the agent will choose to flap. Otherwise, the agent will choose not to flap.

The q-values are updated each time when a game is lost, using a call to the "update_scores" function in the "Bot" class. This function evaluates each action that the agent took, and it changes each corresponding q-value according to the reward given to the agent.

The reward for surviving in the game was set to positive one point, and the reward for losing a game was set to negative one thousand points. We left these two variables unchanged, while we used different reward values for getting a coin in the game. The values of zero, ten, thirty, fifty, eighty, one hundred, and five hundred were used. Also, the learning rate was set to zero point seven, and the discount factor was set to one.

Each time after we trained the agent, we ran a test of one hundred games. We coded another Python program to gather data from these games of Flappy Bird played by the agent. We used this program to calculate the average number of coins the agent obtained in each game, the average number of pairs of pipes the agent avoided in each game, and the ratio of the average number of coins to the average number of pairs of pipes.

3. Results and discussion

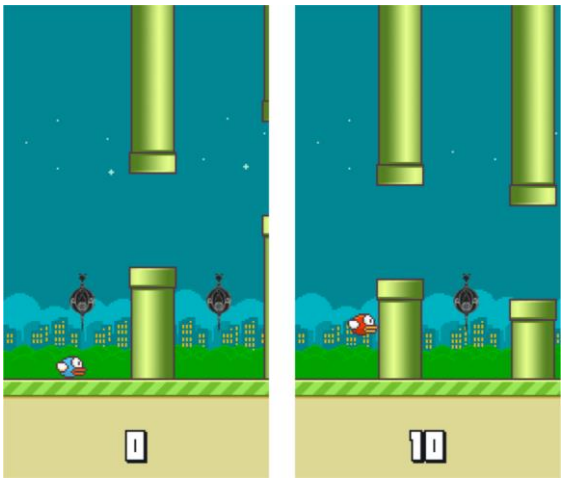

Figure 1. Screenshots of gameplay with an intelligent agent that has not been trained

The left side of Figure 1 shows a bird crushing into the ground, and the right side shows a bird crushing into a pipe. In each of the two cases above, a game is lost.

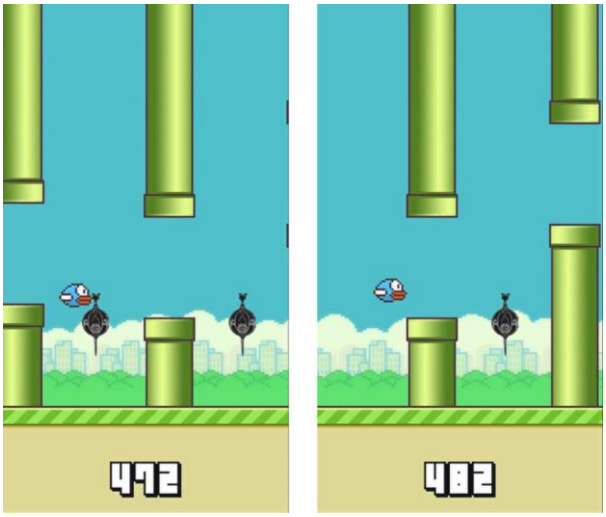

Figure 2. Screenshots of gameplay with an intelligent agent that has been trained (with the reward value of a coin set to fifty)

In Figure 2, the bird successfully gets a coin while avoiding crushing into an obstacle at the same time.

Table 1. Performance of intelligent agent after five thousand iterations of training using different reward values of a coin

Reward value of a coin	Average number of coins	Average number of pairs of pipes	Ratio of average number of coins to average number of pairs of pipes
0	58.09	325.53	0.17845
10	30.88	217.29	0.14211
30	20.34	174.54	0.11653
50	31.65	134.38	0.23553
80	16.94	64.6	0.26223
100	29.72	123.4	0.24084

Table 1 shows that as the reward value of a coin increases, the average number of pairs of pipes tends to decrease. This means that increasing the reward value of a coin tends to make the bird survive for a shorter period of time. Also, when the reward value of a coin is set to ten, thirty, or five hundred, the ratio of the average number of coins to the average number of pairs of pipes is lower than the ratio when the coin has no value. On the other hand, when the reward value of a coin is between fifty and one hundred, this ratio becomes higher.

When the bird goes across a pair of pipes, it will obtain a survival reward of 37 points (the ratio of the interval to the speed of the bird multiplied by the reward of survival of the bird). According to the above tables, when the reward of coins is lower than that of the survival reward considerably, the agent is not attracted by the coins. In addition, as the reward of the coins come close to or exceed the 37, the coins will disturb the bird, which means that the agent tries its best to gain the coins and die for getting coins, so the ratio become higher, even the average number of pipes and the average number of coins become lower. However, when the reward of the coins is equal to 100, the data is too abnormal and therefore disregarded.

The Q-Learning methodology encourages the agent to get a higher reward is the main factor leading to this experimental result.

4. Conclusions

In this work, we tried to solve the problem that what an AI would choose to do in a more complex environment with more reward conditions. We developed a "Gold Coin System" and added it in the game Flappy Bird. Then, a code of an AI that train agent to play Flappy Bird by using Q-Learning and Reinforcement Learning was download from internet and an extra reward is added, and its value is changed to collect different data. According to the samples we collected, since the ratio of the passed pipes and hit coins can show the choice tendency of the agent, a result came out. When the reward value of the coins is close to or more than 37, it would affect the agent: it would tend to hit the coins to gain extra point.

In addition, the downsides of the experiment might be that the number of different types of reward values are not enough, since only one extra type of reward, which changed during the experiment, was added in. Also, the insufficiency of collected data might lead to the inaccurate results. For future research, we would try to add more variables and collect more samples. Moreover, we would try to do more experiments about the research question based on other games.

References

[1] IBM Cloud Education. Artificial Intelligence (AI) [R]. Retrieved from https://www.ibm.com/cloud/learn/what-is-artificial-intelligence, 2020

[2] Piyush Verma, Stelios Diamantidis. What is Reinforcement Learning? [R] Retrieved from https://www.synopsys.com/ai/what-is-reinforcement-learning.html, 2021

[3] Gibney, E. Game-playing software holds lessons for neuroscience [J]. Nature 518, 465–466 (2015). https://doi.org/10.1038/518465a

[4] Li, Junjie, et al. Suphx: Mastering mahjong with deep reinforcement learning [R]. arXiv preprint arXiv:2003.13590 (2020).

[5] Simplilearn. What Is Q-Learning: The Best Guide to Understand Q-Learning Lesson 23 of 33 [R], 2020 Retrieved from https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-q-learning

[6] Cihan, ZeeWanderer. Flappy Bird Bot using Reinforcement Learning in Python [R]. https://github.com/chncyhn/flappybird-qlearning-bot, 2019.

[7] Mike Bertha, Philly.com. Everything you need to know about your new favorite cell phone game, 'Flappy Bird' [R]. Philly.com. 2012.

[8] Ingenito, Vince. Flappy Bird Review [R]. IGN. Archived from the original on February 23, 2014. Retrieved February 26, 2014.

[9] Flappy Bird Tips: How to Get a High Score Without Cheats [R]. Huffingtonpost.co.uk. 2014.

[10] Melo, Francisco S. Convergence of Q-learning: A simple proof [J]. Institute Of Systems and Robotics, Tech. Rep (2001): 1-4.