2020

StarCraft II
Micro-Control Agent

SLS GROUP PROJECT GUIDE

Zhang Zekun
Tao Xiyan
Xiao Yuchao

HTTPS://GITHUB.COM/2020-IRS-G12/????

table of Contents

ABSTRACT

In StarCraft II game competition micro controlling plays a important role in winning the game. Exquisite micro controlling can reverse the unfavorable situation in game competition also can expand the advantage. "Scatter marines to defeat Banelings" is one of most famous micro controlling in game of Terran vs. Zerg. In this paper we use Deep Q-learning Network to explore a better strategy in marines against Banelings.

Key words: Deep Q-learning, DQN, Reinforcement learning, StarCraft2, pysc2, PyTorch, self-learning system

1. INTRODUCTION

1.1. StarCraft II

StarCraft II, one of the most famous game real-time strategy video game, is developed and published by Blizzard Entertainment. It is the sequel of StarCraft and the Brood War expansion pack which is also the most famous RTS game and published in 1998. The game is split into three installments. The first version: Wings of Liberty (the base game) is published in July 2010 for Microsoft Windows and Mac OS X. After three years, Blizzard published its second standalone expansion pack: Heart of the Swarm. No more than 3 years, the third standalone expansion pack: Legacy of the Void was released on November 10, 2015. In November 22, 2016, Blizzard published the final mission expansion pack: Nova Covert Ops.

The key words of RTS game are "Strategy", "Real-time" and "Control". The whole process of the game consists of real-time strategy and controlling. For winning the game, we need some global strategy to make sure that we are in right direction and we also need some partial strategy for the perfect details, moreover the accurate and meticulous controlling is the key to guarantee strategy execution. Based above reason, RTS game can be explored in depth from two aspects of control and strategy, therefore different players can have different game understanding in those two aspects, which is the most important reason in players having different game level. In addition, StarCraft II has three species from the original game StarCraft: Protoss, Terran, and Zerg, which are has themself own character differently. Moreover, each specie has more than a dozen types of units and different types of units have mutual restraint relationship. all above make the game can be played as players' wish and can develop so many types of strategy, so that highly increased the gameplay of StarCraft II. Even though this game is 10 years old, it still has a large number of player and many Worldwide competition because of the gameplay and competitive character of RTS games.

Computer can do very quick, accurate and meticulous controlling in which even the most top human player also cannot do it. In the e-sports industry, especially

in the field of RTS games, we often use APM (Action Per Minute) to evaluate the level of control of the player. According to records and investigations in StarCraft II competition, the South Korea player Park Sung-joon is noted for the record APM of 818 and there are few players can achieve APM more than 400 in StarCraft II. However, computer can achieve APM 7000 easily which is never can human achieve. This makes a lot of control impossible for humans to be implemented by the computer, such as transferring the target to a transport unit (the target will be judged as Not attackable) within 0.2 seconds before the delayed attack reaches the target to avoid the attack.
In November 2016, Alphabet's DeepMind branch announced a collaboration with Blizzard to create "a useful testing platform for the wider AI research community." In August 2017, Blizzard published StarCraft II API and DeepMind also published the python package Pysc2 which can be very easy to use to design a StarCraft II agent. From then on, StarCraft II became another battlefield of intelligent agent competition.

1.2. Game AI

In a broad sense, game AI refers to responding to game input commands. Some game AIs can only return limited preset outputs for limited inputs, and many of these preset outputs are defined based on the developer's knowledge. Therefore, some game AI does not contain real AI technology, and many in the field of AI have argued that video game AI is not true intelligence, but an advertising buzzword used to describe computer programs that use simple sorting and matching algorithms to create the illusion of intelligent behavior while bestowing software.
However, with the development of AI technology, software agents with real AI significance began to appear. In 1997, IBM's Deep Blue computer defeated Garry Kasparov in chess. There is no doubt that this is a great advancement in game AI. Another case where computers defeated top human players is that Google’s intelligent software robot AlphaGo defeated Go master Ke Jie in 2017. If the AI agent of board games is the first show of AI in the game field, then the AI agent of RTS games is a testament to the power of AI in the game field. In 2018, AlphaStar of the DeepMind team defeated the world's top players in StarCraft II with a score of 5:0. Compared with board games, RTS games have a more complicated environment and richer strategies, which have extremely high requirements on agent performance.

2. RELATED WORK

2.1 StarCraft II API and Deep Mind Pysc2

The StarCraft II API, which provides access to in-game state observation and unit control, is published by Blizzard Enterprise. The API provides a perspective for the computer to observe the game. We can retrieve the status returned by the API to obtain the coordinates, health, and other status information of each unit and building. Compared with the player's perspective, this perspective can obtain game data more clearly and intuitively. This advantage can avoid the problem of human players losing the game due to observation errors (Professional players caused their base to be destroyed by their own workers due to observation errors, and finally lost the game). Compared with humans, computers have clearer and more intuitive information acquisition, faster and more precise control of games, so what we need to do is to let computers learn strategies to deal with various situations. StarCraft II API provides a good environment for us to create intelligent agent. The StarCraft II API suite includes these functions:

1. Script AI API
2. Image-based AI API (including layers)
3. Documentation, sample code and sample robot
4. Support offline AI vs. AI games
5. Video pack of 1v1 ladder game
6. Support Windows and Mac
7. Full-featured Linux API package

By using these resources not only can we design and train our own agent by fighting against AI or players but also can we use the replay from the top worldwide competition to train our agent. The release of StarCraft II API will undoubtedly bring the intelligent game Agent to a new level and make StarCraft II a new AI research and competition field.

Released together with StarCraft II API, there is DeepMind's PySC2 package. PySC2 is a StarCraft II python learning environment developed by DeepMind and Blizzard based on StarCraft II API. It exposes Blizzard Entertainment's StarCraft II Machine Learning API as a Python RL Environment.

2.2 AlphaStar

In 2011, the founder of DeepMind Technologies Demis Hassabis called StarCraft "the next step up" after games like Go. In March 2016, after AlphaGo defeat the world champion Go player, Hassabis publicly mulled building an AI for StarCraft. In November 2016, DeepMind and Blizzard announced a formal collaboration and alongside a plan to release an open development environment for bots in Q1 of 2017. In August 2017, DeepMind and Blizzard released development tools to assist in bot development as well as data from 65,000 past games which can be used to train the agent. On 19 December 2018, DeepMind's bot defeated "a top professional player", Grzegorz "MaNa" Komincz, 5-0. DeepMind announced the bot, named "AlphaStar", on 24 January 2019. Some one argued that AlphaStar still had unfair advantages: "AlphaStar has the ability to make its clicks with surgical precision using an API, whereas human players are constrained by the mechanical limits of computer mice". Starting in July 2019, the new, constrained version of AlphaStar anonymously competed against players who "opted in" on the public 1v1

European multiplayer ladder. By the end of August 2019, AlphaStar had attained "grandmaster level", ranking among the top 0.2 percent of human players.
Because of the high complexity of the game in StarCraft II, the process of discovering strategies by AI will take a long time. Therefore, the training method of AlphaStar first adopts the strategy and control that imitates the top players of human beings, that is, the training is conducted by using the game videos of the top players. This allows the Agent to learn more new strategies as soon as possible. The next step is to let the Agent fight, the purpose of this step is to eliminate the limitations of human players.

3. FORMULATED APPROACH

3.1 StarCraft II Game Environment Obtaining

3.1.1 Select the research question

Strategy is the most important part in StarCraft II. Excellent strategy allows players to easily win. Strategy is not only the overall strategic deployment of the game, but also the placement of buildings, the planning of resource use, and the tactical arrangements for each battle Due to the mutual restraint between the attack mode and the armor mode in StarCraft II, under the research and development of the world's top players, the strategic deployment strategy for confrontation between different races has a relatively stable system. Therefore, under certain racial confrontation conditions, some classic and effective unit selection and tactical strategies are used. For example: Terran vs. Zerg often uses the combination of Marine and Medivac Dropship, while Zerg uses Zergling and Baneling as a response. Such unit combination confrontation will last more than 50% of the time in the game, and the key to winning the game is to reduce the battle loss as much as possible in the battle loss exchange time after time. In the middle of the game, whoever loses less in battle will master the rhythm of the game and have the chips to win the game.
Based on the above description, we decided to explore the question of what tactical strategy the player adopts to minimize the battle loss when Marine is fighting the combination of Zergling and Baneling. We plan to use two methods of human confrontation computer and Intelligent Agent confrontation computer to conduct research, and finally train to obtain an intelligent agent for the combination of Marine confrontation Zergling and Baneling.

3.2.2 StarCraft II map selection

We chose a battle simulation map. At the beginning of the game, 9 Marines will appear on one side of the map, and 4 Baneling and 5 Zergling will appear on the other side of the map. In this simulated confrontation, the player will control 9 Marines to destroy Baneling and Zergling, and every time a wave of Zerg

units are destroyed, another wave of Zerg units will be refreshed, but Marine will not refresh. The goal of the confrontation is to use limited Marine to eliminate as many Zerg units as possible. And every time you refresh, Marines and Zerg units will appear in new locations.
Baneling's attack method is self-explosion, causing 16 (+19 vs light armor) damage to enemy units within the 2.2 range (Marine is light armor). The difficulty of this simulated confrontation map is how to disperse Marines and eliminate Zerg units as quickly as possible.

Fig. 1 Schematic diagram of three types of units.
Left: Marine picture; Middle: Zergling picture; Right: Baneling picture.

This map is a more realistic simulation of the small-scale encounter in the early stage of the Terran vs. Zerg match. Such a battle may also occur in the later stage of the game, when the small-scale airborne troops of Terran attack the Zerg base. This kind of battle occupies a dominant position in the exchange of battle losses.
The status of the map is shown in the figure below:

Fig. 2 Schematic diagram of the battle problem map.

3.1.3 Map information acquisition

In the battle, we need to control 9 Marines, and the environment of each Marine is different, so we decided to control each Marine individually. The information we need is the entire environment faced by our chosen Marine, that is, the entire map information includes the location information of friendly Marine and enemy Zerg units.

3.2 StarCraft II Agent established

3.2.1 Marines action

In the battle, each Marine's behavior can be divided into three categories: move, attack, and stop. After subdividing various behaviors, movement can be divided into movement in different directions or movement with different targets. Attacks can be divided into attacks on different types of units or attacks on units at different distances. We set Marine behavior to attack Zergling, attack Baneling, move to the enemy, move to 8 directions, and stop, a total of 12 action modes.

3.2.2 Deep Q-Learning Network

During the battle, the map information is complex, different Marines face different environments, and the selected action strategies are also different, so the end-to-end learning method is more suitable for this situation. Based on the above, we choose the Deep Q-Learning Network model to build our StarCraft II agent. The network input is a map matrix of (64, 64, 1), and the output is the selection of 12 actions. Using DNQ can make Agent automatically generate policies, avoiding the mistake of ignoring information when setting policies manually. The neural network structure is shown in the following table:

Name	Type	Output Shape	Activate	Parameter
Conv1	Conv2D	(32, 32, 32)	ReLU	2080
Conv2	Conv2D	(64, 16, 16)	ReLU	32832
Conv3	Conv2D	(64, 8, 8)	ReLU	32928
Flatten	Flatten	4096	-	-
Dense1	Dense	256	ReLU	1048832
Dense2	Dense	12	Softmax	3072

Table 1 DQN network structure

3.2.3 System execution flow chart

We designed the training process for the Agent:

1. Initialize the map and agent
2. First obtain the information of the battle problem map, calculate whether there is a Marine in danger, if it is, select the marine in the most dangerous situation, if not, select a marine at random.
3. Then enter the map information into DQN to obtain and execute this Marine action.
4. Then update the map information, get the Reward and update the network.
5. Finally, judge whether the termination condition is reached, if not, return to the step 2 to select Marine. If the termination conditions are met, the training ends.

The training flowchart of this Agent is as follows:

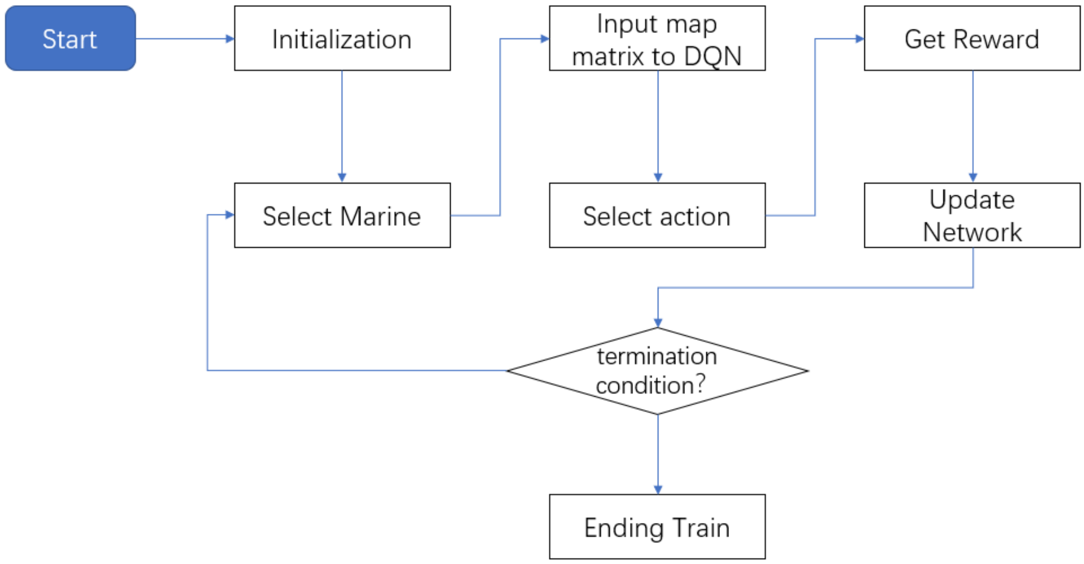

Fig. 3 DQN training flowchart.

4. EXPERIMENTAL RESULTS

At the beginning of training, the Agent will give Marine various instructions. After a period of training, the Agent gradually learned to send a part of Marine to attract attacks from Zerg units.

Fig. 4 Schematic diagram of the initial stage of model training.

In the mid-training period, in addition to sending units to attract attacks, the Agent will also begin to retreat part of the Marine to form gradient firepower.

Fig. 5 Schematic diagram of the mid-stage of model training.

From the picture, we can see that the Marines' positions are arc-shaped, so that the Marine's long-range firepower advantage can be most effectively used. After a period of training, the Agent learned to attack Baneling first, and try not to let Baneling approach as much as possible.

Fig. 5 Schematic diagram of the later stage of model training.

From the figure we can see that there are more Zerglings left, but Baneling only has one left. This shows that our StarCraft II Agent is gradually generating strategies to deal with such a confrontational environment. After 1000 cycles of training, we have obtained an ideal model.

5. CONCLUSION

We use DQN to build the Smart Agent of StarCraft II. Through training, the Agent generates a tactical strategy for Marine against the combination of Zergling and Baneling. After training, the Agent learned to send a part of the Marine to attract the Zerg unit to attack, and at the same time move a part of the Marine backwards to make the Marine's position present an arc, maximizing the use of Marine's long-range firepower advantage. And choose Baneling as the priority attack target, as far as possible to avoid Baneling self-explosion at close range to cause a lot of combat losses.

Reference

[1] Vinyals, Oriol & Ewalds, Timo & Bartunov, Sergey & Georgiev, Petko & Vezhnevets, Alexander & Yeo, Michelle & Makhzani, Alireza & Küttler, Heinrich & Agapiou, John & Schrittwieser, Julian & Quan, John & Gaffney, Stephen & Petersen, Stig & Simonyan, Karen & Schaul, Tom & Van Hasselt, Hado & Silver, David & Lillicrap, Timothy & Calderone, Kevin & Tsing, Rodney. (2017). StarCraft II: A New Challenge for Reinforcement Learning.

[2] Vinyals, Oriol & Babuschkin, Igor & Chung, Junyoung & Mathieu, Michael & Jaderberg, Max & Czarnecki, Wojtek & Dudzik, Andrew & Huang, Aja & Georgiev, Petko & Powell, Richard & Ewalds, Timo & Horgan, Dan & Kroiss, Manuel & Danihelka, Ivo & Agapiou, John & Oh, Junhyuk & Dalibard, Valentin & Choi, David & Sifre, Laurent & Sulsky, Yury & Vezhnevets, Sasha & Molloy, James & Cai, Trevor & Budden, David & Paine, Tom & Gulcehre, Caglar & Wang, Ziyu & Pfaff, Tobias & Pohlen, Toby & Yogatama, Dani & Cohen, Julia & McKinney, Katrina & Smith, Oliver & Schaul, Tom & Lillicrap, Timothy & Apps, Chris & Kavukcuoglu, Koray & Hassabis, Demis & Silver, David. (2019). AlphaStar: Mastering the Real-Time Strategy Game StarCraft II