Multiple AI Competition in Self Developed Game
Term 2 Report
ESTR4998/4999

April 13, 2021

Partners:	XIAO Tianyi
	LUO Lu
Instructor:	Prof. Andrej Bogdanov

Abstract

(some abstract)

1 Background

1.1 Previous Progress

In the last semester, we have implemented the preliminary version of the game environment and Q-Learning agent. In this section, we will briefly reiterate our previous progress before introducing new progress and our opinions.

1.1.1 Environment

When we started to draft our proposal, original considerations for designing our environment included complexity, scalability and implemental difficulty. We hoped to design an environment which is easy to implement, complex enough to distinguish the power of different algorithms and can be reused for different game modes, including 1 player, 1 versus 1, and multi-players against multi-players.

Therefore, we designed a game similar to football, and the graph user interface for 1 versus 1 version can be found in figure 1. In this environment, each agent controls one player which is represented as a blue(team 0) or an orange(team 1) rounded rectangle. And the purpose of players is to shoot the ball, which is represented as a white ball, to goal of opponent team, which is represent as white rectangle in the left border(for team 0) and right border(for team 1).

The environment can output a list to represent the game state. For 1 versus 1 mode, the list will contain the position of the current player, the position of the opponent player and the position of the ball.

The environment can receive a list with size equal 2 from a player as input, the first element represents the direction and the second element represents the action, shoot or not.

After receiving an action, we can call the reward function to calculate the reward of the given action, and return the reward as a scalar number to that agent.

1.1.2 Agent

In the last semester, we have finished the design of Q-Learning agent by following the algorithm provided in the background theory section. Moreover, our conclusion of the last term is that our game is too complicated to be trained by Q-Learning even for 1 vs 1 mode, even if we simplified input states.

The reason to simplify states is that the number of raw game states is an enormous figure even without considering the action space. The simplify function will replace the positions of opponent player and ball by logarithms of relative positions(distances) with base 10, which will compress the range of these 4 parameters from several hundred to 7. This will significantly reduce the size of Q-Table and make some simpler situations trainable, and we will discuss this later.

2 Implementation

2.1 Agents Design

2.1.1 Reinforcement learning

3 Result

4 Discussion

References

[1] Silver, David, et al. ”Mastering the game of Go with deep neural networks and tree search.” nature 529.7587 (2016): 484-489.

[2] Mnih, Volodymyr, et al. ”Playing atari with deep reinforcement learning.” arXiv preprint arXiv:1312.5602 (2013).

[3] Berner, Christopher, et al. ”Dota 2 with large scale deep reinforcement learning.” arXiv preprint arXiv:1912.06680 (2019).

[4] Zoph, Barret, and Quoc V. Le. ”Neural architecture search with reinforcement learning.” arXiv preprint arXiv:1611.01578 (2016).

[5] Mnih, Volodymyr, et al. ”Human-level control through deep reinforcement learning.” nature 518.7540 (2015): 529-533.