Reinforcement and Online learning Coursework Part 2: Dynamic Maze Solving Project

Yu Zhang
Student ID:32050003
yz8n21@soton.ac.uk

1 Introduction

In this coursework, the dynamic maze would be conquered based on Reinforcement Learning(RL). The agent should start from (1.1) and explore the path to the destination(199,199). Meanwhile, the agent only can move in four directions, such as left, right, up and down. In this maze, these are two obstacles, such as walls and fires. The wall cannot be passed through. For the fire, it will be randomly occurred on any positions. Although it also cannot be passed through, it will be extinguished with time. After extinction, this position can be visited. These obstacles will be evaluated by this work. Additionally, more detailed information of this coursework will be saved in GitHub,https://github.com/Yu-Xiu6/COMP6247-Coursework2.git.

2 Algorithm Design

This project will be discussed through Q-learning, Deep Q-Network(DQN) and Dueling DQN. Although there are some problem is Dueling DQN, Q-learning and Deep Q-Network still have better performance.

2.1 Strategies of Walls and Fires

For the walls, it is indestructible and impassable. Therefore, When meeting walls, the agent will choose other available direction. For the fire, the agent will choose two actions, which respectively are eluding it and waiting for its extinction. For eluding it, this will cost too long time to the exploration without any human intervention. Although for the waiting, it is almost like the human intervention, it can effectively descend the probability of going in circles.

2.2 Rewards

At the initial state, the reward is equal to zero. It will be descending one mark on each normal action. If the agent choose stay, it will obtain the punishment with 2 score lost. It is meaningful to encourage the agent to do the movement. Eventually, when reaching the destination, the agent will be rewarded with 100 points.

2.3 Algorithms

According the comparison in Figure 1, the algorithm will select Q-learning, Deep Q-Network(DQN) and Dueling DQN to try to explore the maze.

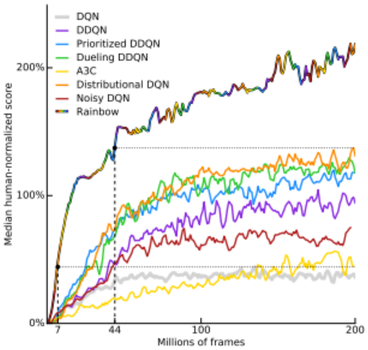

Figure 1: The comparison of model-free algorithms in playing Atari games [1]

Q-learning

In Q-learning, the Q table is updating with the time. Afterwards, the action will be selected through the table of diverse state [2]. For this experiment, the maze will be explored at the first to discover the path to the destination. These features will be loaded into Q-table to improve its learning efficiency. Meanwhile, if meeting fires, the agent will adopt the waiting strategy. After that, the agent will grope the path by itself.

DQN

The DQN is utilized the neutral network based on the Q-learning algorithm, which effectively ascends the variety of the Q-predict and Q-real [3]. In DQN, the agent will elude the fire and wall. Each feature can be well learnt by the algorithm.

Dueling DQN

The dueling DQN utilizes the feature named advantage to determine the Q value of each action through the addition between advantage and Q value of its action [4]. It has the same strategy for fires and walls.

3 Results

The Q-learning and DQN has successfully reach the destination. Whereas, the Dueling DQN is failed in this game, because of some mistakes in codes. The Q-learning costs 27761 episodes in waiting strategy. There is no result for Q-leaning in eluding strategy. The reason is that the agent repeatedly moves between several positions. Meanwhile, the DQN costs 38973 episodes.

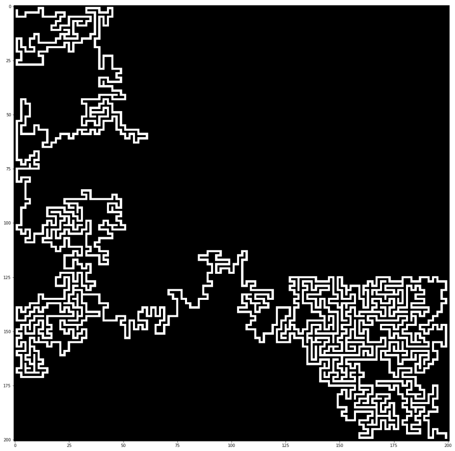

Figure 2: The trace of the agent

4 Analysis

From the gained result, the Q-learning is better than the DQN. However, they adopt the diverse strategies. For the same strategy, the DQN is better, which verifies the comparison result from [1].

4.1 Future Work

In future work, the reward strategy should be enhanced. Each position should be neatened into one buffer. If the agent always appears in this position for a time, the agent will obtain a severe punishment over time. In this situation, it seems that there is a high probability that agent has been going in circles. There is not any contribution.

References

[1] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, “Rainbow: Combining improvements in deep reinforcement learning,” in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing atari with deep reinforcement learning,” *arXiv preprint arXiv:1312.5602*, 2013.

[4] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, “Dueling network architectures for deep reinforcement learning,” in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.