Using Reinforcement Learning to Decide What's for Dinner

SUBMISSION #0006
IBBB-2019

Abstract

This paper shows a method of how to use reinforcement learning to decide what's for dinner. Every time I go for dinner with my friends, they always ask me : "What's for dinner? ". Oh, what a hard problem that almost every one does not know the answer. So am I. I have been confused about this problem for a long time. I'll try to use RL method to solve the problem in this paper.

1. Introduction

Recently DRL methods have been applied in many fields such as video games, Go, Robotics and so on. "What's for dinner" is really a hard question(Everyone was plagued by this question?). We will try to apply RL method to solve this hard problem.

2. Problem Modeling

We first build a model of the problem. Every time we need to make a decision on what's for dinner, we have some choices, for example, YueXiang, mlxg, noodles... These choices can be seen as actions that we can take. So the action space A is made of all kinds of food that we can choose.

$$A = \{YueXiang, mlxg, mlt, noodles...NaN\} \tag{1}$$

Here we add 'NaN' to the action space because the initial state need a representation for no action before. What we have had for dinner since a few days ago can be modeled as the state(Here we only keep three days records for simplicity).

$$s_t = (a_{t-3}, a_{t-2}, a_{t-1}) \tag{2}$$

Now we have the state space S. The reason we use what we have had for dinner as state is that maybe we don't want to eat same things for several days, and we are willing to choose what we haven't eat for a long time. After we take an action, we can give a score to the food, which can be seen as the reward. So we can model the problem as an MDP. Now we can use reinforcement learning to solve this problem!

3 Algorithm

It's easy to solve an MDP with little state space and action space. So we may not need a neural network as a function approximator. Here we use table based Q-learning algorithm.

Algorithm 1: Decide what's for dinner with Q-learning(DWDQ)

Output: A Q-table
initialize Q-table terms to zero,
  $s_0 = (NaN, NaN, NaN)$;
while *True* **do**
  $s_t = (s_{t-1}[1], s_{t-1}[2], a_{t-1})$;
  Go Rock Paper Scissors;
  if *I win* **then**
    Choose an action randomly;
  else
    $a_t = \max_a Q(s_t, a)$ based on the Q-table;
  end
  Go for dinner and get the score r_t;
  Update Q-table:
  $Q(s_t, a_t) = Q(s_t, a_t) + \alpha[r_t + \gamma \max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))]$
end

Here we use the game 'Rock Paper Scissors' to decide whether we choose the greedy action based on the Q-table or choose an action randomly. By doing this, we avoid the process of generate a random number, which is too hard for human. As the training process going on, you may want to lower the exploration rate, you can choose a random action after you continue winning the game twice to get a lower exploration possibility.

If you want to improve the performance of the algorithm, you can keep more records of what you have had for dinner as state information. But this will cause the Q-table to be very large. Maybe you will need a neural network and use DRL methods such as DQN. Then there will be another problem: You may not have enough samples to train your net. (Maybe you can connect "yourselves" who are in the different parallel universes to get more experience, just like you open many envs to collect samples when you train your own DRL algorithm.)

4 Experiment

I will keep doing experiments to verify the feasibility of the algorithm in the next two years.

5 Conclusion

In this paper, we propose an algorithm to solve a human problem –What's for dinner. We also propose a method that leveraging the game 'Rock Paper Scissor' to decide if we use the greedy action or not, skillfully avoiding generating random number. However, We don't know whether the algorithm works because we don't have enough time to test it. We will keep doing experiments to verify our algorithm in the future.

Acknowledgement

I would like to give my sincere gratitude to Yi Ma and Jiewen Zhao. If they don't always go for dinner with me, I may not find this interesting subject, which is so meaningful. And they also give me a lot help in daily life. I express my heartfelt thanks to Yi Ma for giving me a lot advice on my IBBB paper, which is very critical. Finally, I would like to give my sincere thanks to the organizer of IBBB for giving us such a good platform to use our imagination.