FINDING THE SHORTEST PATH USING REINFORCEMENT LEARNING

LINGFENG ZHANG, WENHAN HAO, & TIANMING QIU

Group: applied-rl17 / E3

1 INTRODUCTION

Path planning is a widely-studied and basic problem. Also it has many actual scenarios, for example, for autonomous car navigation and drones for logistics. Considering about distance and traffic situation, the optimal path between start and destination will be figured out. It is related to mathematics, optimal research, and artificial intelligence. Although there are already many existing algorithms to find the best path[1], we shall attempt to solve this problem with reinforcement learning.

This project offers an agent which could find the shortest path between two locations in a unfamiliar road network and thus reduce the cost of time and energy.

2 PROJECT DESCRIPTION

Instead of the real road network in a city or a building, a simulated network will be adopted, which contains several straight lines on the ground as the roads to form the road network with many crossroads. This simulated example map is showed in Figure 1. Arbitrary two points of them could de chose as starting point and destination.

In this project we will use E-puck as the test agent. The E-puck robot will be first placed at the starting point and start to explore the road network. It will make a decision which direction it should go at every crossroad. Once the E-puck reach the destination, it will be replaced at the same starting point. During the learning process, E-pack will take a random direction at each crossroad. Every time E-puck arrive at the destination successfully it will update all the value function of each state. After several trials, the E-puck may find the shortest path to the destination according the latest value functions.

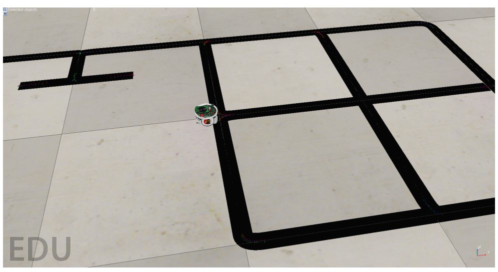

Figure 1: An example of simulate map

3 MODEL

The model part includes how we disgn the states, actions, reward and Q function as well as how the algorithm works.

3.1 State

In this project, we will consider different crossroads in the simulated map as the states. A preliminary conception would include a road network with 8-10 states. We are aware of difficulties in recognizing the different states. In addition, the obstacles will be not considered as states. And in the simulated map we will also set some blind alleys, they can be treated as states but with very poor rewards.

After brainstorming, we have two tentative solution for this problem:

First, a concept of 'photoelectric encoder' would be adopted. We will draw some specific patterns, for example like black-and-white stripes unit, at the crossroads. And we could use the under sensor on E-puck to scan the patterns to determine the current state.

Alternative we could use the front CMOS color camera to recognize the different street signs we put at the crossroads in order to determine the states. On the street signs we decided to write the capital letter such as just 'A', 'B', 'C' and so on. Then we use some computer vision techniques such as feature detection and matching to recognize which pattern the image is. Afterwards the E-puck can understand at which crossroads it is.

Since the image transforming and processing is so computation-assuming that E-puck cannot handle with properly even make the E-puck to halt. So we consider about combination of two solutions. That means we will set some specific patterns on the road ground especially before the crossroad to let the E-puck know it is approaching the crossroad and ready to start the program of image processing. Then E-puck could wait for few seconds at crossroad to judge and make correct decisions during learning process.

3.2 Action

The actions for E-puck would be not difficult. We adopt just 'turn right', ''turn left', 'go straight' and 'backward' naively.

In details, for the 'backward' action we will not accept the method that directly make the wheel rotate inversely to go backward because the E-peck is designed to detect the states always by using its front CMOS color camera. Therefore we ought to design a safe solution to make E-puck turn round and change into the opposite direction and make sure that it would be still in the middle of the road. Unless it would be a little trouble for following activities.

3.3 Reward

Reward setting is always the most difficult part in reinforcement learning algorithm. In this model we will assign a reward '+10' at the destination state and '0' at the other states. Also at the 'blind alley' it will get a very poor reward such as '-100'.

3.4 Q function

We decide to use *Q-Learning*(Watkins, 1989)[2], defined by

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)] \tag{1}$$

And an off-policy TD control algorithm would be adopted. At each iteration step the Q function would be updated and the action would be taken greedily.

4 PROJECT MANAGEMENT

The project will be divided into three part by the type of work: Learning algorithms, recognition algorithm and Simulation in V-rep and physical arena.

- **Learning Algorithms:** An off-policy TD control Q-learning algorithm will be designed for deterministic road network.
- **recognition algorithm:** Find a solution to determine the feature patterns and realize computer vision part algorithm
- **simulation in V-Rep and physical arena:** Familar with the manipulation of V-rep and bild the physical test arena

We will set up several milestones in our futrue project and divide our project into four stages:

- **Stage 1:** familiar with Python and Vrep simulation environment
- **Stage 2:** according the primary conception, we will design a simple map with less than 10 states to test and modify both the modeling and algorithm
- **Stage 3:** confirm the modeling, optimize our algorithm
- **Stage 4:** tranfrom the program from simulation environment to the real E-puck

REFERENCES

[1] S. Koenig and R. G. Simmons, "Complexity analysis of real-time reinforcement learning applied to finding shortest paths in deterministic domains," DTIC Document, Tech. Rep., 1992.

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.