Reinforcement Learning for Robot Routing and Testing against Uncertainty

Mohammad Hasan Olyaei Torqabeh[1], **Ali Olyaei Torqabeh**[2]

[1] Faculty of Electrical Engineering, Sadjad University of Technology, Mashhad, Iran;
mh.olyaei123@sadjad.ac.ir

[2] Department of Computer Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran;
ali.olyaei@mail.um.ac.ir

Abstract

This paper discusses the routing of a robot in an environment (4x12) by Reinforcement learning. In this environment, there are 48 locations (squares) where 10 squares of these 48 squares are burned and the agent must learn to choose a path that does not hit the flames. Our expectation of the agent is that it can recognize two paths. The first path is a safe path to avoid flames as far as possible and the second path is the fastest path. Simulation of this issue has been done with MATLAB software. After learning the agent, uncertainty has been added to the environment. The uncertainty in this problem is defined as adding new fire squares, and the simulation results show that, ultimately, the agent has been able to detect this change in the environment and suggest another new safe path.

Keywords: Reinforcement learning, Robot, Routing, Uncertainty, MATLAB

Introduction

Today, many issues are defined globally by scientists. One of the best ways to solve problems is to use Artificial Intelligence and Machine Learning. In general, machine learning topics are divided into three types of learning with a supervisor, without a supervisor, and a reinforcement learning [1].

Reinforcement Learning (RL) is a way in which an agent, by doing the trial-and-error and receiving the reward, learns that choose the best action in the present state, and if the environment is dynamical and change, he must be able to himself the environment [2]. Selection methods of action in each state, named "policy" that have different types [3]. RL has a specific framework and flowchart. This framework [4] and flowchart [5] is shown in Figure 1 and 2, which consists of 7 main components: State, action, reward, Environment, policy, Q-Table and update section.

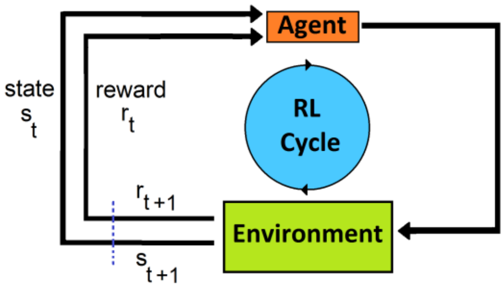

Figure 1. RL Framework

The Reinforcement Learning Cycle (RLC) is such that, first, the current state of the agent is considered. According to Q-table, one of the actions is selected and the agent goes to the next state. Depending on the next state, it receives a reward and re-runs the RLC.

The flowchart that shown in Figure 2 explains that after receiving the rewards, the Q-table is updated. With higher the number of learning steps, the Q-table values will be closer to the optimal values of Q-table and the solution of problem will be more accurate.

The learning stop condition is to reach agent to the goal.

Problem description

The environment considered in this paper is a plane with dimensions of 4x12 squares and agent is a robot that must be learned from the starting point to the Goal point.

This agent can move in the four main directions up, down, left and right, that these represent Actions Vector (AV) of the problem. 10 squares of this plane have been burnt and contain fire and the robot must learn to stay away from these squares. If a robot hits the fire, it receives -100 reward and returns to the starting point, and if it does a normal move and does not hit the fire, it receives -1 reward and if it reaches the goal point, it receives 10 reward. In Equation 1, the Reward Function (RF) is shown.

$$\mathrm{RF} = \begin{cases} \mathrm{R(Fire)} = -100 \\ \mathrm{R(NoFire)} = -1 \\ \mathrm{R(Goal)} = 10 \end{cases} \qquad (1)$$

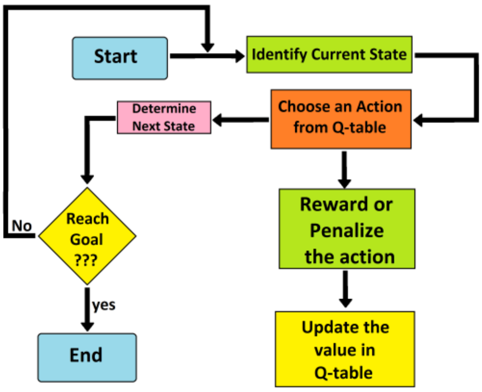
Start
Identify Current State
Choose an Action from Q-table
Determine Next State
Reward or Penalize the action
Reach Goal ???
No
yes
End
Update the value in Q-table

Figure 2. RL Flowchart