Reinforcement Learning Project

AIvengers 2023

August 31, 2023

1 Introduction

You have been introduced to some dynamic programming algorithms like Q-Learning to train a model and devise policies based on rewards and punishments. Now it's time for you to get started and implement what you've learned!

The game is simple! imagine a **N-by-N** grid which is filled with integers where N represents the number of rows and columns. Each integer represents the amount of reward you'll get if you enter a specific cell. (of course negative rewards are equivalent to punishments.) You're job is to find the most-suiting route in order to get from the bottom-left cell to the top-right one. However, you may only move 1 cell upwards or rightwards at each step. The data regarding the configuration of the grid is provided in the file '**Grid.xslx**'. You can load this file into your code and start digging around!

Suggestion : Since the 30-by-30 grid would we too large to start with, you could implement the algorithm om smaller grids and then try optimizing the routes for the given size.

-5	-5	-5	-5	-5	10
-1	-1	-1	-1	-1	-1
-1	15	-1	-1	5	-1
-1	-1	-1	-10	-1	-1
-5	-5	-5	-5	-5	-5
-10	-1	-1	-1	-1	-1

Table 1: A simpler 6-by-6 grid

2 Concepts

Worry not if you're not familiar with Q-Learning! the algorithm iteratively tries to enhace a policy in order to maximize the reward. Some concepts you might need in the process can be observed below.

- **States :** Each cell in the grid is a state.
- **Actions :** You may choose to move rightwards or upwards at each step. This decision defines your action which leads to newer state.
- **Q-Values :** Q-Learning revolves around the concept of Q-Values. The Q-Value of a state-action pair represents the expected cumulative reward you can achieve by starting from that state, taking a particular action, and then following a specific policy.
- **Learning rate :** Controls how much the newly-gained information will affect your policy
- **Discount factor :** Controls the importance of future rewards rather than the immediate ones.

3 Briefing

So, you must implement the learning algorithm and improve your policy in order to determine the most rewarding route. The update rule you may use in order to do so is as follows.

$$Q(s,a) = (1-\alpha)\ Q(s,a) + \alpha\Big(R(s) + \gamma\ (max[Q(s^{'},a^{'})])\Big)$$

Where Q represents the Q-Value corresponding to a specific pair of states and actions, α is the learning rate, R represents the immediate reward of a certain state and γ is the discount factor.

Hope you enjoy

AIvengers team