Reinforcement Learning

Sao Mai Nguyen

2022

Contents

1 Introduction **1**

2 Reinforcement learning : Definitions and Goals **2**
2.1 Intuition 2
2.2 Formal definition 2
2.3 Policy and Value 3

3 Q Learning **4**
3.1 How to choose an action? 4

4 SARSA **4**

5 Travail attendu **4**
5.1 Structuration du code 4
5.2 Objectifs du projet 4

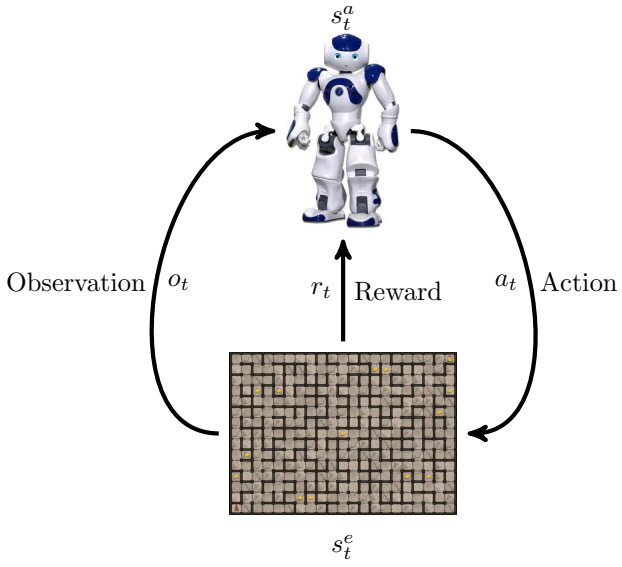
6 Moyens d'évaluation **5**
6.1 Rapport 5
6.2 Clarté du code 5
6.3 Présentation 5

1 Introduction

The aim of this tutorial is to give basis of Reinforcement Learning to achieve the project IN104 at ENSTA ParisTech. The goal of this project is to implement one or several algorithms in one or several environments.

Reinforcement learning is used when we have a clear idea of what we want, but not exactly how we can achieve it. For example if I play chess, I know I want to checkmate my opponent, but I don't know exactly which action to perform every time it is my turn.

2 Reinforcement learning : Definitions and Goals



2.1 Intuition

In the Reinforcement Learning (RL) approach, an agent interacts with an environment. The environment produces a state s_t at each timestep t, and when receiving the current state s_t, the agent reacts with an action a_t. The agent acts according to a policy $\pi(a_t|s_t)$, which represents the probability to take an action a_t when being in state s_t (in a deterministic environment, $\pi(s_t) = a_t$). After the agent has taken the action a_t from state s_t, the environment provides a new state s_{t+1} alongside a reward r_t.

The goal of the agent is to maximize the cumulative rewards over its lifetime. At each time step the agent receives a reward and the objective is to maximize the reward not on one precise timestamp but on the whole trajectory, i.e. in the future until the end of the horizon. Maximizing the long-term future reward : what does it mean? It means that we want to choose the actions at each time step so as to maximize what I receive at time $t + 1$ but also at time $t + 2$, $t + 3$, the action that i choose at time t influences the reward that I receive now but also in the future.

The *discounted return* is the sum of the rewards with a discount factor : $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma R_{t+4} + ...$ where γ, the discount factor, is a real number between 0 and 1. Mainly it means that we are going to look at the sum of the rewards the future rewards from our time t so once i've done action t then i'll receive all these rewards t plus r_{t+1} etc and I'm going to sum them but i'm not going to sum them equally depending on when I receive the reward. If I will receive ten dollars I'd rather receive them now than in ten years.

γ close to 1 means that we are quite fine if we receive the reward now or later whereas γ small is that we want the return to to be as soon as possible.

With this idea of cumulative return or discounted return, we are going to define two functions that we call value functions the first is the value of the state given a policy.

2.2 Formal definition

A RL problem s defined by a set of variables:

- The first is a state space where the states are in a discrete or continuous space.

- Then we have a the action space where the actions are the same they are in they can be either discrete or continuous.
- Then we have $\mathcal{T}$ the transition function or the transition operator so in general the transition operator is a function from state action state and another another integer to a real number between 0 and 1; so mainly it's a probability distribution of the state, the action and the next state, in other words, it's a function of state at time t, action at time t and state that time $t+1$. They are often written under the form of a conditional probability : the probability of states s_{t+1} equal to s' given that at time t you had s_t equal s and a_t equal to a.
- the horizon over which the agent will act so it can be seen as the lifetime of the agent. The horizon H is an integer from 0 to infinity : if we consider that the lifetime is infinity then h is infinity.
- a reward function r, representing the reward for that is going to be to be received by the agent. A reward function is a function from state action states and horizon to a real number.
- Then you have a constant called the discount factor. It is a general written γ it's a real number between zero and 1.

2.3 Policy and Value

To maximize the cumulative return or discounted return, we are going to define the policy.

A policy is a function that gives you the actions that you want to do so that the agent wants to do so it's in general a probability function of the state.

Given a policy that we denotes π we're going to be able to define the value $V_\pi(s)$ s as the expectation of this discounted return given that at time t I'm in state s and in the future until infinity all the actions that i'm going to use are given by the policy which is a probability density function π so this function v π is a function from the state to the real number the real number being the expectation of the return.

The value of a state action pair given a policy is the expectation of this return given that at time t i'm at state s and i've chosen action a and for the times at $t+1$ to infinity i will choose actions according to the policy π. The function Q_π depends on the state and the action.

Policy π

- A policy describes the agent behavior
- Map from state to action
- Deterministic : $a = \pi(s)$
- Stochastic : $\pi(a|s) = P[a_t = a|s_t = s]$

Q-Value Function Q

- prediction of future reward
- Evaluates the goodness of state-action pairs
- Action selection using the value function
- $Q_\pi(s, a) = \mathbb{E}_\pi[\sum_k \gamma^k r_{t+k+1}|s_t = s, a_t = a]$

Q-values defines 'how good' is a pair state-action. For instance, in a gridworld, if the agent is in a given square s, it needs to know if it must go up, down, left or right. We can represent the Q-Value function with a table of dimension $|\mathcal{S}| \times |\mathcal{A}|$.

The objective of the project if to Fill this Q table for different environments and with different algorithms. And once we have the Q table, we know how the agent must behave to get the maximum return.

Below are the pseudo-code of some algorithms.

3 Q Learning

Algorithm 1 Q-Learning for estimating π^*

Parameter $\alpha \in [0, 1]$ learning rate, $\epsilon > 0$
1. INITIALIZATION
$Q(s, a) \in \mathbb{R}, \forall s \in \mathcal{S}, a \in \mathcal{A}$ arbitrarily except that $Q(terminal, .) = 0$

2. TRAINING
loop for each episode:
  Initialise $s \in \mathcal{S}$
  repeat for each step of episode :
    Choose a using policy derived from Q
    Take action a, observe r, s'.
    $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma max_a Q(s', a) \ - \ Q(s, a)]$
    $s \leftarrow s'$
  until s is terminal
end loop

3.1 How to choose an action?

In Q-learning, there are 2 main methods to choose an action from Q.

The first way, is called ϵ-Greedy: the agent chooses a random action with probability ϵ, otherwise choose an action which maximises $Q(s, .)$.

The first manner is called the Botzmann exploration : the agent chooses an action randomly with a probability proportional to $exp(Q(s, a))$.

4 SARSA

Sarsa is another algorithm to learn the Q-values. The name Sarsa comes from the quintuple $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$ needed for the algorithm

5 Travail attendu

5.1 Structuration du code

Veuillez structurer votre code avec un dossier environnement où différents environnements sont implémentés, et un dossier agents, où plusieurs algorithmes sont implémentés.

5.2 Objectifs du projet

Implémenter un ou plusieurs algorithmes d'apprentissage par renforcement pour résoudre le déplacement dans un labyrinthe. Implémenter d'autres environnements. Pour aller plus loin :

Algorithm 2 Sarsa for estimating Q^*

Parameter $\alpha \in [0, 1]$ learning rate, $\epsilon > 0$
1. INITIALIZATION
$Q(s, a) \in \mathbb{R}, \forall s \in \mathcal{S}, a \in \mathcal{A}$ arbitrarily except that $Q(terminal, .) = 0$

2. TRAINING
loop for each episode:
    Initialise $s \in \mathcal{S}$
    Choose a from s using policy derived from Q
    repeat for each step of episode :
        Take action a, observe r, s'.
        Choose a' from s' using policy derived from Q
        $Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma Q(s', a') \ - \ Q(s, a)]$
        $s \leftarrow s', a \leftarrow a'$
    until s is terminal
end loop

- Rendu graphique
- Faire décroitre le taux d'exploration

6 Moyens d'évaluation

6.1 Rapport

- Description de l'algo utilisé
- Problèmes rencontrés
- Résultats et interprétation

6.2 Clarté du code

- Code (très) bien commenté
- Noms de variables et de fonctions cohérents
- Readme : description du projet + comment lancer le projet
- Git : commits réguliers et bien commentés

6.3 Présentation

8 minutes de présentation, 6 minutes de question.