ROB311 - TD3 - Apprentissage pour la Robotique

Araujo Belén, Victor
De Carvalho Ferreira Sula, Julia

October 2019

Q Learning

Q-Learning is a free model algorithm that belongs to the family of reinforcement learning algorithms. It is considered off policy, as it takes in to account random actions, from each it is also able to learn.

Shortly , this algorithm is based in the construction of the q-table, in each combination action-state has a respective value or reward. This table is initialized at zero.

The agent has two types of interaction with its in environment: exploitation or exploration. The first on consists in using the q-table in order to chose the action with the higher reward, as the other consists in taking an random action as to get to know the environment/discovering new states.

Finally, the update of the q-table is done based in equation 1, in each Q_{new} is the new q-table value, α the learning rate, Q_{old} the old q-table value , r the reward and γ the discounted factor.

$$Q_{new} = (1-\alpha)Q_{old} + \alpha(r + \gamma max Q_{new}) \tag{1}$$

In order to implement this algorithm in a pacman game, there are 6 main functions that allow to follow the q-learning principle. They are:

- **init()**
  This is the initialization of the q-table for a q-agent. Through a counter, the table is initialized with all values zero and the key for accessing the values are the action and state linked to it
- **getQValue()**
  This function returns the q-value for a pair action-state if it exists .
- **computeValueFromQValues()**
  In order to find the max q-value for a state, this function compares the q-value for all possible action combined with a defined states and returns it maximum value.
- **computeActionFromQValues()**
  This function is very similar to the above, it also search the action that combined to a defined state offers the best q-value, but instead of returning the q-value it returns the action.
- **getAction**
  In order to chose the type of interaction the q-learning agent will have with the environment a random number is created if it is above a threshold ϵ, the exploration mode is deployed ,else it is the exploitation mode.Consequently, it is possible to elect the ration exploitation/exploration by tuning ϵ.
- **update()**
  The q-table (counter structure) is updated according to equation 1 with the help of the *computeValueFromQValues()* function