Reinforcement Learning use cases

Mohamed ZOUIDINE and Mohammed KHALIL

Master of Informatics & Telecommunication

Mohammed V University, RABAT

Abstract

In this work, we will study two basic algorithms of reinforcement learning (dynamic programming and Q-Learning), a theoretical study (Markov decision process, Bellman equation) will be presented as well as practical examples.

Tow different examples of GridWorld are illustrated. Each one is based on the Markov decision process theory, where an agent tries to compute an optimal value function in order to find an optimal policy to reach predefined objectives.

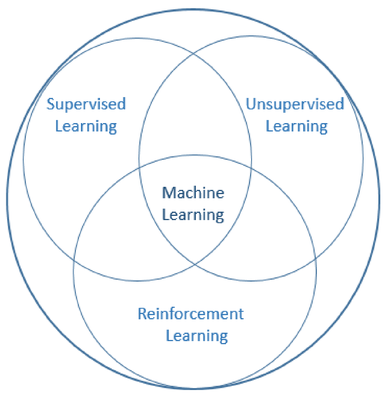
For the first examples we will use dynamic programming methods, and for the last ones we will use Q-Learning to find the optimal policy.

Keyword: Reinforcement Learning, Dynamic Programming, Q-Learning, GridWorld, Markov decision process, Bellman equation

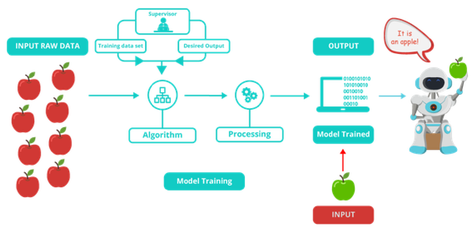
Introduction

The idea that we learn by interacting with our environment is probably the first to occur to us when we think about the nature of learning. When an infant plays, waves its arms, or looks about, it has no explicit teacher, but it does have a direct sensorimotor connection to its environment. Exercising this connection produces a wealth of information about cause and effect, about the consequences of actions, and about what to do in order to achieve goals. Throughout our lives, such interactions are undoubtedly a major source of knowledge about our environment and ourselves. Whether we are learning to drive a car or to hold a conversation, we are acutely aware of how our environment responds to what we do, and we seek to influence what happens through our behavior Learning from interaction is a foundational idea underlying nearly all theories of learning and intelligence. In the field of machine learning, the current trend of researchers is oriented towards a new approach based on interaction learning. This approach is called **reinforcement learning**.

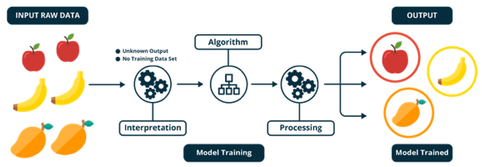
Machine Learning



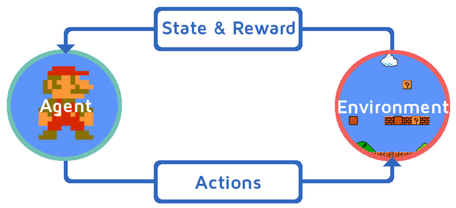
Supervised Learning



Unsupervised Learning

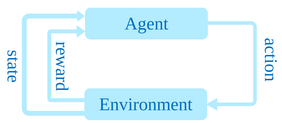


Reinforcement Learning



The theoretical bases of RL

Markov Decision process



- Dynamics of the MDP:

$$p(s', r|s, a) = \Pr\{S_t = s', R_t = r|S_{t-1} = s, R_{t-1} = r\}$$

- The expected rewards for state-action pairs:

$$r(s, a) = \mathbb{E}[R_t|S_{t-1} = s, A_{t-1} = a] = \sum_{r\epsilon\mathcal{R}} r \sum_{s\prime\epsilon\mathcal{S}} p(s', r|s, a)$$

- Discounted return:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 G_t + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- The policy:

$$\pi(a|s) = \Pr\{A_t = a|S_t = s\}$$

- The value function:

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s\,] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \Big/ S_t = s\right]$$

- The action-value function for policy π:

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \Big/ S_t = s\,, A_t = a\right]$$

Bellman equation

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\,[r + \gamma v_\pi(s')]$$

Exploitation and Exploration

Practical examples of RL

Dynamic Programming

The term dynamic programming (DP) refers to a collection of algorithms that can be used to compute optimal policies given a perfect model of the environment as a Markov decision process (MDP).

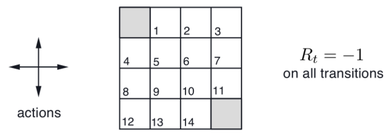
- Policy Evaluation:

$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)\,[r + \gamma v_k(s')]$$

- Policy Improvement:

$$\pi'(s) = \arg\max_a \sum_{s',r} p(s', r|s, a)\,[r + \gamma v_\pi(s')]$$

Consider the 4*4 GridWorld:



$R_t = -1$ on all transitions

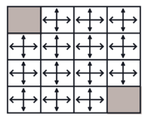
The nonterminal states are $\mathcal{S} = \{1, 2, \ldots, 14\}$

There are four actions possible $\mathcal{A} = \{up, down, left, right\}$

$p(6, -1|5, right) = 1 \quad p(7, -1|7, right) = 1 \quad p(10, -1|5, right) = 0$

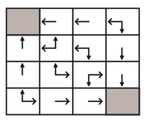
➢ k= 0

0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0



➢ k= ∞

0.0	-14.	-20.	-22.
-14.	-18.	-20.	-20.
-20.	-20.	-18.	-14.
-22.	-20.	-14.	0.0



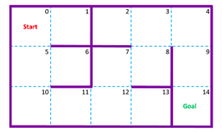
Q-Learning

Q-learning is a model-free reinforcement learning algorithm. The goal of Q-learning is to learn a policy, which tells an agent what action to take under what circumstances. It does not require a model (hence the connotation "model-free") of the environment, and it can handle problems with stochastic transitions and rewards, without requiring adaptations.

- Q-value:

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

Consider the 5*3 GridWorld example:



The nonterminal states are $\mathcal{S} = \{0, 1, 2, \ldots, 14\}$

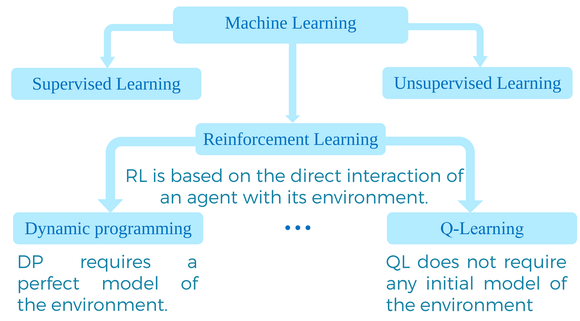
There are four actions possible $\mathcal{A} = \{up, down, left, right\}$

$r(s) = -0.1\ \forall s\epsilon\mathcal{S}\{14\}, \qquad r(14) = 10$

➢ Q-Table:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1	0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	1.25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.62	0.00	2.50	0.00	0.00	0.00	0.62	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	1.25	0.00	0.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00	0.00
5	0.01	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
7	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.62	0.00	0.00	0.00	0.16	0.00	0.00
8	0.00	0.00	0.00	1.25	0.00	0.00	0.00	0.31	0.00	0.00	0.00	0.00	0.00	0.00	0.00
9	0.00	0.00	0.00	0.00	2.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	10.00
10	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.00	0.00
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.00	0.16	0.00	0.00
12	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.31	0.00	0.00	0.00	0.08	0.00	0.08	0.00
13	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.16	0.00	0.00
14	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Conclusion



Perspectives