SARSA on FrozenLake

Ugo Marchesini

ugo.marchesini@studio.unibo.it

Abstract

FrozenLake is a game provided by the Gym library to practice the methods of **Reinforced Learning**. Gym provides APIs to act on the environment with an action and return the observable status and reward value. The aim of this project is to apply the **SARSA** method to the **FrozenLake** game and experiment with some variations

1 Problem

FrozenLake is a GridWorld type game where the character can move in 4 directions. If it enters a hole square the episode ends with a reward of 0, if it enters a goal square the episode ends with a reward of +1. If the argument is_slippery = True then the character has a ⅓ chance of completing the action in the direction set by the action, ⅓ chance of performing the action orthogonal to the direction set by the action, this for both directions. There are no rewards for other actions or reaching other squares. The map set for this project is 4X4, but it is customizable.

2 Solution

The chosen solution is SARSA because the size of the observable space which is given by the number of cells (4X4) and the size of the possible actions (4) is such as to use a tabular method. In case the state space is too high, a solution with an approximation function is the only way to go. The choice of the action is done in this project in 2 ways, the first through the ϵ-greedy and the second through the **decreasing** ϵ. The second action selection mode is ϵ-greedy decreasing , i.e. each time the goal is reached ϵ is reduced by a deltaepsilon ($10*10^{-6}$).

When epsilon ($10*10^{-2}$) reaches the minimum value of deltaepsilon the selection is almost greedy. The use of a decreasing ϵ is justified by the convergence of the policy to determinism as the episodes continue.

3 Result

Initializing the QValue array with random values does not lead to convergence, as reported in the table below

Initializing the QValue matrix to 0 and using a fixed ϵ converges quite quickly with a maximum performance of 15% of goals achieved on episodes.

Initializing the QValue matrix to 0 and using ϵ converges faster with a maximum performance of 72% goals achieved on episodes.

Preprint. Under review.

Episode	Random	ϵ-0	Decr-0
0	0	0	0
1	402	885	0
2	423	1466	859
3	432	1523	1700
4	407	1418	2056
5	501	1527	2682
6	482	1516	4664
7	460	1518	6366
8	435	1531	5462
9	465	1467	5775
10	410	1583	6701
11	422	1523	5216
12	400	1487	4889
13	492	1520	6058
14	399	1576	7284
15	441	1469	5636
16	434	1498	6785
17	448	1466	7157
18	394	1497	5659
19	384	1480	7258
20	514	1466	5261

Every line are 10000 episodes.