**Reinforcement Learning Assignment 2 Report**

In this report I’ll compare the implementation of two different RL algorithms in windy grid world with different actions sets for each algorithm with epsilon-greedy selection policy.

Algorithms: (SARSA, Q-learning)

First actions set: **([‘UP’,’DOWN’,’LEFT’,’RIGHT’])**

Second actions set: **([‘UP’, ‘DOWN’, ‘LEFT’, ‘RIGHT’, ‘UP-right’, ‘UP-left’, ‘DOWN-right’, ‘DOWN-left’])**

Solving windy grid world using SARSA algorithm with different values of alpha and epsilon with implement First action set and the number of episodes is 300:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gamma** | **Alpha** | **Epsilon** | **Total number of steps taken to reach to 30000 Episodes** | **Number of steps to reach to the goal state** |
| 1 | 0.1 | 0.1 | 25264 | 21 |
| 1 | 0.1 | 0.3 | 29659 | 39 |
| 1 | 0.1 | 0.5 | 45277 | 156 |
| 1 | 0.3 | 0.1 | 13680 | 19 |
| 1 | 0.3 | 0.3 | 18681 | 44 |
| 1 | 0.3 | 0.5 | 31986 | 47 |
| 1 | 0.5 | 0.1 | 11321 | 22 |
| 1 | 0.5 | 0.3 | 17164 | 33 |
| 1 | 0.5 | 0.5 | 29975 | 153 |

Findings:

The best value for alpha was 0.3 and for epsilon was 0.1 and it took the minimum number of steps to reach to the goal and it was 19 steps and that’s because the epsilon value makes the most of action selection process is greedy.

The worest value for alpha was 0.1 and for epsilon was 0.5 and it took the minimum number of steps to reach to the goal and it was 156 steps and that because the step size was too small to converge, and the epsilon value balanced the exploration and exploitation action selection process.

I noticed if I increased the value of epsilon the algorithm starts learning after a long time so if I increased the value of epsilon, I need to increase the number of episodes as much as I can to make the agent learn better.

The best value for epsilon was 0.1 because the best actions taken for SARSA algorithm selected as greedy action and the value of alpha was 0.3 because it doesn’t take a big step to converge.

Note: if you run the code again the number of steps to reach the goal with change but the worest policy will be with epsilon = 0.5 and it take the longest time to reach the minimum number of steps.

Solving windy grid world using SARSA algorithm with different values of alpha and epsilon with implement second action set (stochastic king moves) and the number of episodes is 300:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gamma** | **Alpha** | **Epsilon** | **Total number of steps taken to reach to 30000 Episodes** | **Number of steps to reach to the goal state** |
| 1 | 0.1 | 0.1 | 47977 | 82 |
| 1 | 0.1 | 0.3 | 45115 | 27 |
| 1 | 0.1 | 0.5 | 60734 | 20 |
| 1 | 0.3 | 0.1 | 29638 | 35 |
| 1 | 0.3 | 0.3 | 41236 | 122 |
| 1 | 0.3 | 0.5 | 50716 | 121 |
| 1 | 0.5 | 0.1 | 29159 | 20 |
| 1 | 0.5 | 0.3 | 39647 | 186 |
| 1 | 0.5 | 0.5 | 66833 | 226 |

Findings:

The best value for alpha was 0.1 and epsilon is 0.5 but it wasn’t that much better than previous algorithm with different action step and I think that’s because the stochastic movements.

The worest value for alpha was 0.5 and for epsilon was 0.5 and it took the minimum number of steps to reach to the goal and it was 66 steps which was much worest than using SARSA with first action set with same policy, and that’s because the agent explores and exploit action with the same probability also the value of alpha is a little bit big which means a big step, also due to the stochastic movements.

Solving windy grid world using Q-learning algorithm with different values of alpha and epsilon with implement First action set and the number of episodes is 30000:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gamma** | **Alpha** | **Epsilon** | **Total number of steps taken to reach to 300 Episodes** | **Number of steps to reach to the goal state** |
| 1 | 0.1 | 0.1 | 24897 | 29 |
| 1 | 0.1 | 0.3 | 27389 | 41 |
| 1 | 0.1 | 0.5 | 34167 | 68 |
| 1 | 0.3 | 0.1 | 13345 | 20 |
| 1 | 0.3 | 0.3 | 15805 | 29 |
| 1 | 0.3 | 0.5 | 21984 | 116 |
| 1 | 0.5 | 0.1 | 10630 | 18 |
| 1 | 0.5 | 0.3 | 13291 | 25 |
| 1 | 0.5 | 0.5 | 19496 | 39 |

Findings:

The best value for alpha was 0.5 and for epsilon was 0.1 and it took the minimum number of steps to reach to the goal and it was 18 steps which was a great improvement than the previous algorithm with both actions sets.

The worest value for alpha was 0.1 and for epsilon was 0.5 and it took the minimum number of steps to reach to the goal and it was 101 steps which is much worest than SARSA algorithm with the same actions set and the same alpha and epsilon values and I think that is because agent have the same probability with explore and exploit the actions.

I noticed if I increased the value of epsilon the algorithm starts learning after a long time so if I increased the value of epsilon, I need to increase the number of episodes as much as I can.

Solving windy grid world using Q-learning algorithm with different values of alpha and epsilon with implement second action set (stochastic king moves) and the number of episodes is 30000:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Gamma** | **Alpha** | **Epsilon** | **Total number of steps taken to reach to 30000 Episodes** | **Minimum time steps taken to reach to the goal state** |
| 1 | 0.1 | 0.1 | 45137 | 21 |
| 1 | 0.1 | 0.3 | 43413 | 173 |
| 1 | 0.1 | 0.5 | 55830 | 67 |
| 1 | 0.3 | 0.1 | 30660 | 42 |
| 1 | 0.3 | 0.3 | 33054 | 25 |
| 1 | 0.3 | 0.5 | 37317 | 28 |
| 1 | 0.5 | 0.1 | 27207 | 38 |
| 1 | 0.5 | 0.3 | 29008 | 106 |
| 1 | 0.5 | 0.5 | 38039 | 155 |

Findings:

The best value for alpha was 0.1 and for epsilon was 0.1 and it took the minimum number of steps to reach to the goal and it was 21 steps which was not a great improvement than the previous algorithm and I think that’s because the randomness in the action selection process

The worest value for alpha was 0.5 and for epsilon was 0.5 and it took the minimum number of steps to reach to the goal and it was 155 steps which wasn’t worse than using SARSA with first and second action set with same policy.

Last findings:

Q-learning from the following plots After 25-30 episodes, Q-learning started learning better and quickly compared to SARSA when they are solving the grid world using second actions set.

Q-learning succussed to find the shortest path to the goal state with stochastic actions set and the alpha was 0.3 and the epsilon was 0.1 in the other hand SARSA algorithm with the same policy and the same alpha value and the same number of episodes with both actions set wasn’t less than 18 steps, so Q-learning here is considered better since it takes the action with the maximum value for the next state.

I noticed that most of the time best epsilon value is 0.1 because these algorithms are better for to select actions greedily not randomly and the optimal alpha value could be any value between 0.1 and 0.5 but not higher than that.

A picture containing shape

Description automatically generatedA picture containing shape

Description automatically generated

Chart, line chart

Description automatically generatedChart

Description automatically generated

A picture containing shape

Description automatically generatedA picture containing shape

Description automatically generated

Chart

Description automatically generatedChart, line chart

Description automatically generated

You may also want to see the rest of other plots in the code to ensure, also the grid world with the optimal path is also shown in the code.