**CS 580 Reinforcement Learning**

**HW3**

**Yang Zhang 11529139**

**Part I. Implementation of Temporal Difference Learning (Q-learning)**

Result:

Q\_values = [[64.3758021855065, 53.1833014089657, 62.377878329379755, 75.29999999999997], [73.98083425501605, 75.29821978383154, 64.74304878912916, 86.99999999999997], [86.99999999999997, 75.29999999999997, 75.29999999999997, 99.99999999999999], [0.0, 0.0, 0.0, 0.0], [64.76999999999995, 46.76361438961865, 54.48989497463296, 54.610376600454934], [0.0, 0.0, 0.0, 0.0], [86.99999999999997, 64.76999999999995, 75.29999999999997, -99.99999999999999], [0.0, 0.0, 0.0, 0.0], [55.29299999999995, 46.76369999999995, 46.76369999999995, 55.29299999999995], [55.29299999999995, 55.29299999999995, 46.76369999999995, 64.76999999999995], [75.29999999999997, 64.76999999999995, 55.29299999999995, 55.29299999999995], [-84.35968651000002, 53.29308507345464, 64.76999999999995, 54.80497857256528]]

Convert q\_value to policy:

policy = [3, 3, 3, 0, 0, 0, 0, 0, 0, 3, 0, 2]

The policy from Q-learning is the optimal policy (e=0.3, a=0.01)

**Part II. Implement eligibility traces and show how different value of lambda change the speed of learning (Sarsa with eligibility trace )**

|  |  |
| --- | --- |
| **Iterations to converge (average of 6 runs)** | **Lambda** |
| 330211 | 0.9 |
| 159017 | 0.8 |
| 45040 | 0.7 |
| 148748 | 0.6 |
| 1518 | 0.5 |
| 535 | 0.4 |

From the table, we can see that in general, with smaller lambda comes to faster learning.

**Part III. Compare on-policy vs. off-policy (Sarsa vs. Q-learning)**

|  |  |
| --- | --- |
| **Method** | **Iterations to converge (average of 6 runs)** |
| **Sarsa** | 37974 |
| **Q-learning** | 1730 |

Off-policy (Q-learning) works better in this domain. The reason is that this domain is quite sample, so that it is easy to pass value of terminal states to non-terminal states. Therefore, off-policy update is faster to converge than on-policy update.

**Part IV. How different values of alpha and epsilon affects learning**

|  |  |  |
| --- | --- | --- |
| **Alpha** | **Epsilon** | **Iterations to converge** |
| 0.1 | 0.1 | 75483 |
| 0.2 | 0.2 | 17878 |
| 0.3 | 0.3 | 8103 |
| 0.4 | 0.4 | 4770 |
| 0.5 | 0.5 | 119 |
| 0.6 | 0.6 | 1964 |
| 0.7 | 0.7 | 1355 |
| 0.8 | 0.8 | 435 |
| 0.9 | 0.9 | 636 |

From the table, we can see that in general, with bigger alpha and bigger epsilon come with faster learning.