

**ASSIGNMENT-4**

**ACADEMIC INTEGRITY STATEMENT:**

"I (We) certify that the code and data in this assignment were generated independently, using only the tools and resources defined in the course and that I (we) did not receive any external help, coaching or contributions during the production of this work."

**PART-I REPORT:**

**1.Describe the environment that you defined. Provide a set of actions, states, rewards, main objective, etc**.

At first, we imported pip install gym then imported the various libraries such as pandas, numpy, matplot.lib and we also imported the spaces from gym. The environment we have taken is a 4x4 grid with objectives as Start, End, 2 rocks and 2 batteries. The rewards for the batteries and the rocks is +5 and -5 along with the observational space and the action space having max time steps as 5. We have taken initializer, reset, step and render. The actions are the left, right, top and bottom. We have 16 states of matrix. The main objective is to get to the end state with the maximum reward.

**States** = 16 states [(0,0),(0,1),(0,2),(0,3),(1,1),(1,2),(1,3),(1,4),(2,1),(2,2),(2,3),(2,4),(3,1),(3,2),(3,3),(3,4),(4,1),(4,2),(4,3),(4,4)]

**Actions** = [**3** = left, **2** = right, **0** = down, **1**= up]

**Rewards**= [-5, +5, -5, +5]

**Objective=** To reach the goal state with maximum reward.

**Note**: The Battery (Parrot Green), Rock(black), Start(Darkish green), End(Yellow) are defined with the various unique colors.

**2.Provide visualization of your environment**.

The basic Visualization with 0 rewards is shown below:

Chart, bar chart

Description automatically generated

**3.Runarandomagentforatleast10timestepstoshowthattheenvironmentlogic is defined correctly. Print the current state, chosen action, reward and return your grid world visualization for each step**.

We have designed a certain path for an anonymous thing to travel. The path is listed below performing various actions and rewards and at the we get final reward which is the maximum.

path = [2,2,2,0,3,3,0,2,0,2]

Action = [Right,Right,Right,Down,Left,Left,Down,Right,Down,Right]

Chart

Description automatically generated with medium confidenceA picture containing chart

Description automatically generatedChart, bar chart

Description automatically generatedChart

Description automatically generated

Chart

Description automatically generated with low confidenceChart

Description automatically generated with medium confidenceChart, bar chart

Description automatically generatedChart

Description automatically generated

**Chart, bar chart

Description automatically generatedChart, bar chart

Description automatically generated**

So we got an final reward of 16 after reaching it to the end.

**3.SafetyinAI:Write a brief review explaining how you ensure the safety of your environment**.

AI Safety is a collective phrase for the ethics we should follow to avoid machine learning system mishaps, as well as unanticipated and negative behavior that may result from bad design of real-world AI systems.

Besides this we took care by not going outside of the grid and ensuring the safety to our environment.

We ensured the safety for our environment by following the principles below:

1.Using a human centered design approach

2.Identifying multiple metrics to assess training and monitoring.

3.Continue to monitor and update the system after deployment.

**PART-II:**

**1.Show and discuss the results after applying SARSA to solve the environment defined in Part I**.

The SARSA algorithm is a slightly modified version of the well-known Q-Learning algorithm.

SARSA is known as on process learning algorithm.

Firstly we need to create the environment,

A picture containing text

Description automatically generated

Initializing different parameters in sarsa

Text

Description automatically generated with low confidence

Defining utility functions to be used in learning process

Graphical user interface, text, application

Description automatically generated

Learning the Q value



In the conclusion sarsa is more effective when compared to the previous model in part 1 as we have used 1000 episodes with alpha as 0.1, gamma as 0.6 and epsilon as 0.9

A screenshot of a computer

Description automatically generated with low confidence

**2.Provide a plot for epsilon decay**

**Chart

Description automatically generated**

**3.Provide a plot for the total reward per episode**

**Chart

Description automatically generated**

**4.Provide the valuation results. Run your environment for at least 10 episodes, where the agent chooses only greedy actions from the learnt policy. Plot should include the total reward per episode.**

**Graphical user interface, text, application, chat or text message

Description automatically generated**

**Chart, line chart

Description automatically generated**

**5.** **Give your interpretation of the results.**

The rewards per episodes is increasing when compared to the previous sarsa algorithm and the epsilon decay value has been reduced.

**6.** **Briefly explain these tabular methods: SARSA and Q-learning. Provide their update functions and key features.**

**SARSA:** SARSA (state–action–reward–state–action) is a machine learning reinforcement learning algorithm for learning a Markov decision process policy. Rumer and Niranjan offered the moniker "Modified Connectionist Q-Learning" in a technical note. Rich Sutton submitted the alternate name SARSA, which was only noted as a footnote. This name simply reflects the fact that the main function for updating the Q-value is dependent on the agent's current state "S1," the action "A1," the reward "R" the agent receives for taking that action, the state "S2" the agent enters after taking that action, and finally the next action "A2" the agent chooses in its new state. SARSA is an abbreviation for the quintuple (st, at, rt, st+1, at+1). Depending on which time step the prize is explicitly awarded, some writers use a slightly different approach and write the quintuple (st, at, rt+1, st+1, at+1). Temporal Difference(TD) Update to improve the agent’s behavior.

**Q-learning:** Q-learning is a model-free reinforcement learning technique for determining the worth of a certain action in each state. It can handle issues with stochastic transitions and rewards without requiring adaptations and does not require a model of the environment (thus "model-free"). Starting from the present state, Q-learning discovers an optimum policy in terms of maximizing the anticipated value of the total reward across all consecutive steps. *Q*-learning can identify an optimal action selection policy for any given FMDP, given infinite exploration time and a partly-random policy.

**7. Provide the analysis after tuning at least two hyperparameters from the list above.**

In the hyper paarameter1 we tuned gamma (discount factor) value as 0.89, 0.79 and 0.69.

In the second hyper parameter (2) we tuned episodes as 400,450,300

**8.** **Try at least 3 different values for each of the parameters that you choose. Provide the reward graphs and your explanation for each of the results. In total you should have at least 6 graphs and your explanations. Make your suggestion on the most efficient hyperparameters values for your problem setup.**

With gamma as 0.89

**Chart

Description automatically generated**

In the above graph we took the values alpha = 0.01, gamma = 0.89, epsilon= 0.9 and episodes= 500

Now tuning gamma as 0.79

Chart

Description automatically generated

In the above graph we took the values alpha = 0.01, gamma = 0.79, epsilon= 0.9 and episodes= 500

With gamma as 0.85

Chart

Description automatically generated

In the above graph we took the values alpha = 0.01, gamma = 0.85, epsilon= 0.9 and episodes= 500

With episodes as 400

Chart

Description automatically generated

In the above graph we took the values alpha = 0.01, gamma = 0.99, epsilon= 0.9 and episodes= 400

With episodes as 450

Chart

Description automatically generated

In the above graph we took the values alpha = 0.01, gamma = 0.99, epsilon= 0.9 and episodes= 450

With episodes as 300

Chart, histogram

Description automatically generated

In the above graph we took the values alpha = 0.01, gamma = 0.99, epsilon= 0.9 and episodes= 300

We choose that the hyper parameters with episodes as 300 is best and most efficient for the given model.

|  |  |  |
| --- | --- | --- |
| Team member | Assignment part | Contribution |
| SriHarsha Gullapalli(gullapal) | Part1 | 50% |
| Likesh krishna(likeshkr) | Part2 | 50% |
|  |  |  |

**References:**

1.Piazza

2.<https://www.geeksforgeeks.org/sarsa-reinforcement-learning/>

3.<https://medium.com/swlh/introduction-to-reinforcement-learning-coding-sarsa-part-4-2d64d6e37617>

4. <https://en.wikipedia.org/wiki/Q-learning>

5.<https://en.wikipedia.org/wiki/State–action–reward–state–action>