Inteligência Artificial para Robótica Móvel

CT-213

Instituto Tecnológico de Aeronáutica

Relatório do Laboratório 11 - Aprendizado por Reforço Livre de Modelo

Leonardo Peres Dias

20 de junho de 2025

Sumário

1 Breve Explicação em Alto Nível da Implementação

A implementação desenvolvida define uma classe base `RLAlgorithm`, e suas especializações `Sarsa` e `QLearning`.

A tabela de ação-valor (*Q-table*) é representada por uma matriz `q` de dimensão `(num_states, num_actions)`. Cada entrada `q[s, a]` corresponde à execução da ação a no estado s sob a política corrente.

1.1 SARSA

O algoritmo SARSA segue uma política *on-policy*, ou seja, utiliza a mesma política ε-greedy tanto para seleção de ações quanto para atualização dos valores. A função `learn` atualiza a Q-table de acordo com a seguinte equação:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma Q(s',a') - Q(s,a)\right]$$

onde a' é a próxima ação escolhida de acordo com a política ε-greedy.

1.2 Q-Learning

Já o Q-Learning é um algoritmo *off-policy*, ou seja, aprende assumindo que sempre será tomada a melhor ação futura, mesmo que a execução atual use uma política ε-greedy. A atualização da Q-table segue a equação:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$$

Ambos os algoritmos utilizam funções auxiliares para seleção de ações (`epsilon_greedy_action`, `greedy_action`) e para gerar a política determinística greedy em forma tabular a partir da Q-table final (`compute_greedy_policy_as_table`).

2 Figuras Comprovando Funcionamento do Código

2.1 SARSA

2.1.1. Tabela Ação-Valor e Política *Greedy* Aprendida no Teste com MDP Simples

Tabela 1: Tabela Ação-Valor Q(s,a) aprendida pelo SARSA

Estado s	**Q(s, Ação = L)**	**Q(s, Ação = S)**	**Q(s, Ação = R)**
0	-9.33	-8.60	-10.58
1	-10.47	-9.64	-11.55
2	-11.21	-10.50	-11.66
3	-11.82	-11.56	-12.08
4	-12.38	-12.32	-12.31
5	-11.88	-11.94	-11.44
6	-10.89	-11.59	-10.48
7	-10.58	-11.47	-9.42
8	-9.46	-10.33	-8.65
9	-7.64	-8.71	-8.57

Política Greedy Aprendida:

$$\pi = [\mathrm{L}, \mathrm{L}, \mathrm{L}, \mathrm{L}, \mathrm{R}, \mathrm{R}, \mathrm{R}, \mathrm{R}, \mathrm{R}, \mathrm{S}]$$

2.1.2. Convergência do Retorno

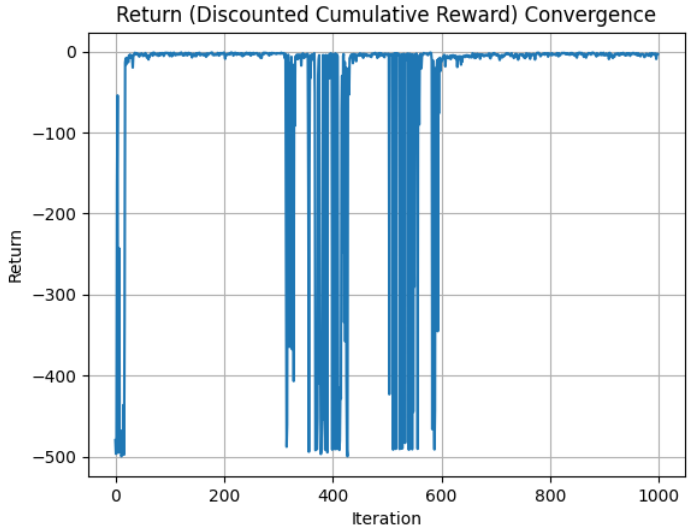

Figura 1: Convergência do Retorno no SARSA

2.1.3. Tabela Q e Política Determinística que Seria Obtida Através de *Greedy*(Q)

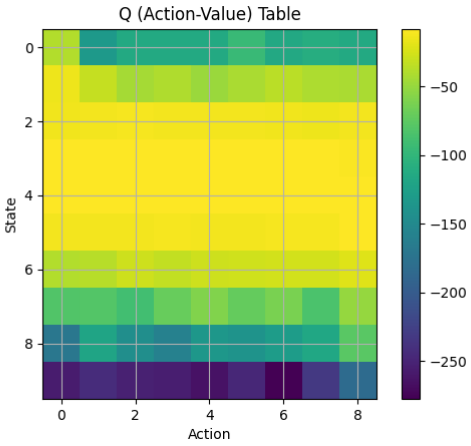

Figura 2: Tabela Q aprendida pelo SARSA

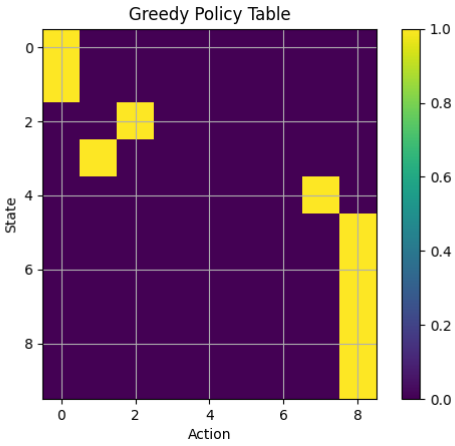

Figura 3: Política determinística Greedy obtida

2.1.4. Melhor Trajetória Obtida Durante o Aprendizado

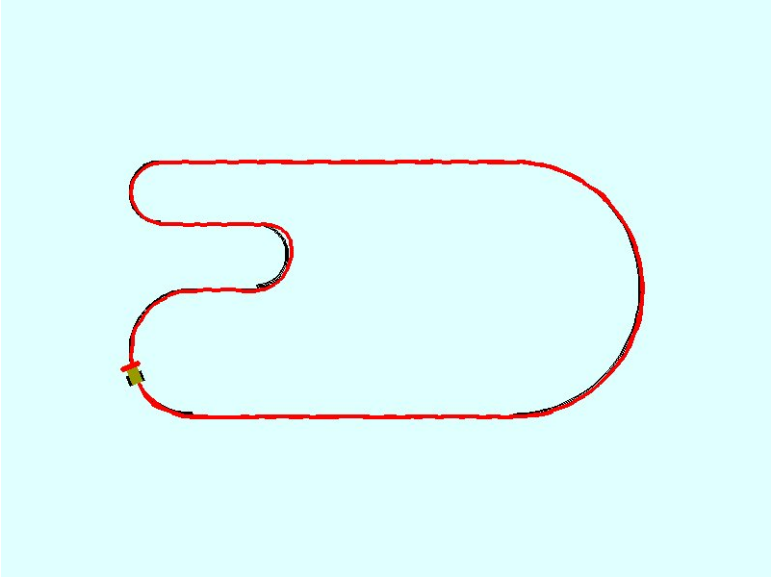

Figura 4: Melhor Trajetória Obtida pelo SARSA

2.2 Q-Learning

2.2.1. Tabela Ação-Valor e Política *Greedy* Aprendida no Teste com MDP Simples

Tabela 2: Tabela Ação-Valor Q(s,a) aprendida pelo Q-Learning

Estado s	**Q(s, Ação = L)**	**Q(s, Ação = S)**	**Q(s, Ação = R)**
0	-1.99	-1.00	-2.97
1	-2.97	-1.99	-3.94
2	-3.45	-2.97	-4.55
3	-4.14	-3.94	-4.49
4	-5.19	-4.89	-4.89
5	-4.25	-4.60	-3.94
6	-3.75	-4.11	-2.97
7	-2.97	-3.91	-1.99
8	-1.99	-2.97	-1.00
9	0.00	-0.99	-0.99

Política Greedy Aprendida:

$$\pi = [\mathrm{L}, \mathrm{L}, \mathrm{L}, \mathrm{L}, \mathrm{L}, \mathrm{R}, \mathrm{R}, \mathrm{R}, \mathrm{R}, \mathrm{S}]$$

2.2.2. Convergência do Retorno

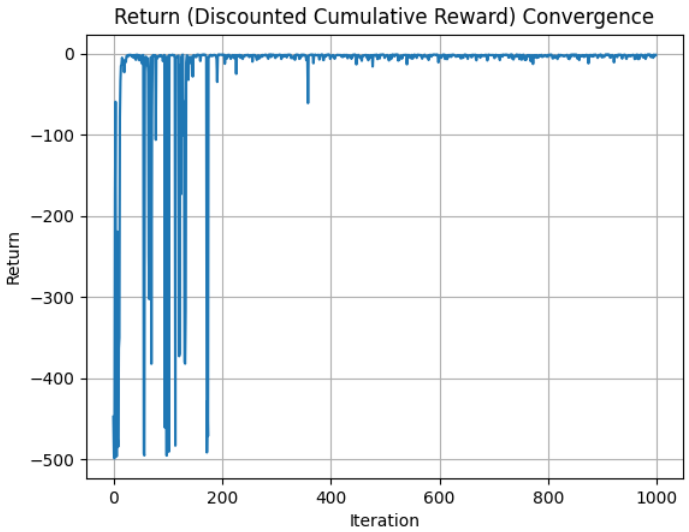

Figura 5: Convergência do Retorno no Q-Learning

2.2.3. Tabela Q e Política Determinística que Seria Obtida Através de *Greedy*(Q)

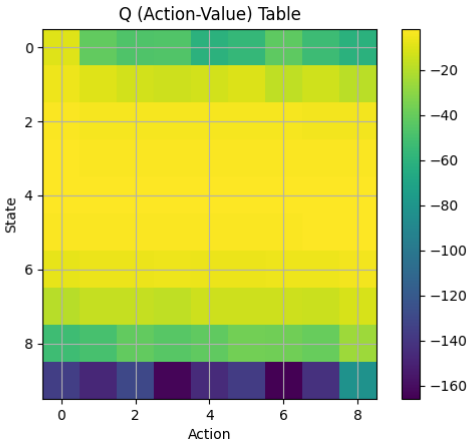

Figura 6: Tabela Q aprendida pelo Q-Learning

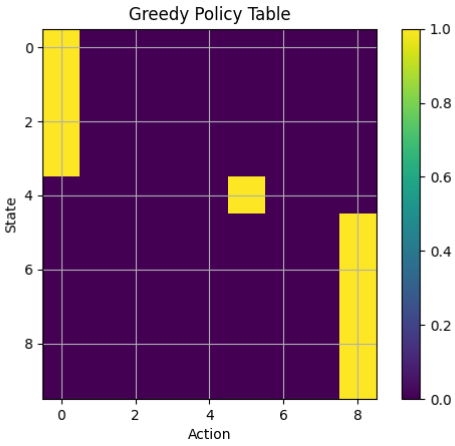

Figura 7: Política determinística Greedy obtida

2.2.4. Melhor Trajetória Obtida Durante o Aprendizado

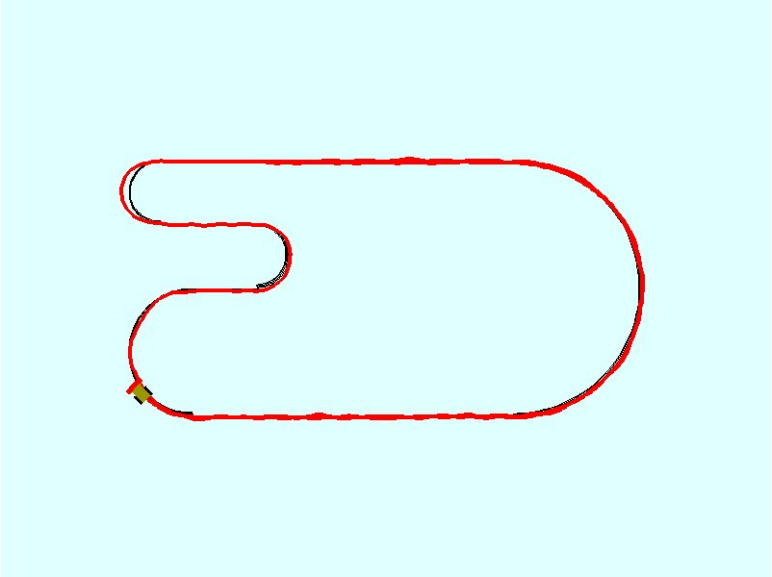

Figura 8: Melhor Trajetória Obtida pelo Q-Learning

3 Discussão dos Resultados

No experimento com o MDP simplificado — um labirinto unidimensional — tanto o algoritmo SARSA quanto o Q-Learning convergiram para políticas *greedy* que resolvem a tarefa de forma ótima. Ambas as políticas levam o agente da posição inicial até o objetivo com o menor número possível de passos. No entanto, apesar de serem ótimas, as políticas aprendidas diferem: por exemplo, no estado central do labirinto, a ação ótima aprendida pelo SARSA foi virar à direita, enquanto o Q-Learning optou por virar à esquerda — ambas válidas devido à simetria do ambiente. Essa diferença reflete o caráter *on-policy* do SARSA, que tende a aprender com base no comportamento atual, e o aspecto *off-policy* do Q-Learning, que favorece estimativas mais agressivas das melhores ações. Além disso, os valores nas tabelas de ação-valor mostram que o Q-Learning atribui valores mais altos (menos negativos), indicando uma política mais otimista em relação ao retorno esperado.

Nos testes realizados com o robô seguidor de linha, observamos que ambos os algoritmos — SARSA e Q-Learning — foram capazes de aprender trajetórias que completam o percurso de forma eficaz. Contudo, a convergência do retorno apresentou comportamentos distintos: o SARSA exibiu maior instabilidade ao longo das iterações, com episódios de retorno significativamente negativos, especialmente nas fases iniciais do aprendizado. Isso reflete seu caráter *on-policy*, que o torna mais sensível à política exploratória. Por outro lado, o Q-Learning convergiu de forma mais estável e rápida, beneficiando-se de seu viés *off-policy* e natureza mais otimista. As tabelas de ação-valor ilustram essa diferença, com valores mais consistentes e homogêneos para o Q-Learning. No geral, apesar de trajetórias semelhantes terem sido obtidas ao final do treinamento, o Q-Learning demonstrou maior robustez e confiabilidade durante o processo de aprendizado.