Instituto Tecnológico de Aeronáutica - ITA
Inteligência Artificial para Robótica Móvel - CT-213
Aluno: Vinícius José de Menezes Pereira

Relatório do Laboratório 11 - Aprendizado por Reforço Livre de Modelo

1. Breve Explicação em Alto Nível da Implementação

1.1. SARSA

SARSA é um algoritmo de aprendizado de máquina por reforço. Nele se utiliza uma política On policy e-greedy que vai aprendendo durante as iterações. Esta política sempre converge para a política ótima para um grande número de iterações, pois além de ser greedy, tem um pequeno fator de aleatoriedade que garante exploração. O aprendizado da função ação-valor Q se dava pela seguinte expressão:

$$Q(S,A) = Q(S,A) + \alpha\big(R + \gamma Q(S',A') - Q(S,A)\big)$$

Sendo onde o alvo o aprendizado é justamente:

$$R + \gamma Q(S',A')$$

1.2. Q-Learning

A grande diferença entre o SARSA e o Q-Learning é que o Q-Learning utiliza uma política off policy para aprender. Assim, se determina a política ótima greedy pi pela observação da política exploratória e-greedy. O aprendizado utiliza a seguinte expressão para atualizar a função ação-valor:

$$Q(S,A) = Q(S,A) + \alpha\left(R + \gamma \max_{a' \in A} Q(S',a') - Q(S,A)\right)$$

A alvo do aprendizado passa a ser:

$$R + \gamma \max_{a' \in A} Q(S',a')$$

Nela, a próxima ação é escolhida de maneira greedy, não sendo necessário atualizar nem utilizar o next_action. A cada episódio, a ação é tomada baseada numa política e-greedy segundo os valores correntes da função ação valor Q. No fim apenas, a política final é descoberta de forma greedy segundo os valores finais da função ação valor.

2. Figuras Comprovando Funcionamento do Código

2.1. SARSA

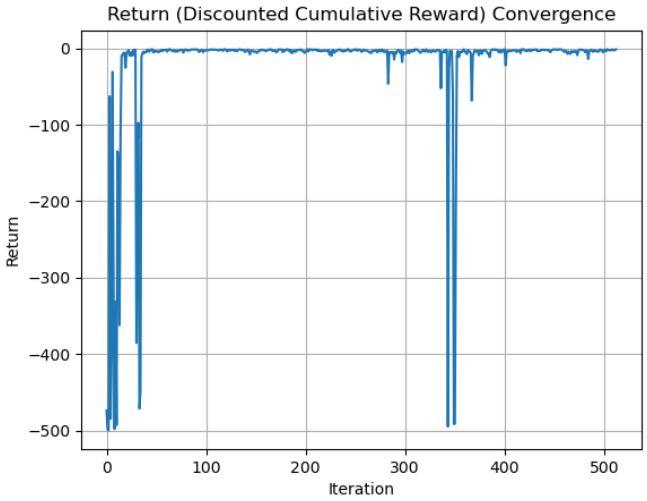
2.1.1. Tabela Ação-Valor e Política *Greedy* Aprendida no Teste com MDP Simples

(saída de test_rl.py)

```
Action-value Table:
[[ -9.29704594  -8.47355633 -10.50303918]
 [-10.43615115  -9.38003657 -11.56236204]
 [-11.07107754 -10.57983127 -11.20986866]
 [-11.6404873  -11.53629684 -12.16826108]
 [-12.41417247 -12.27566413 -12.2676435 ]
 [-11.70244519 -11.9049345  -11.38835743]
 [-11.14789244 -11.4617244  -10.42212335]
 [-10.41631179 -11.35424008  -9.34320435]
 [ -9.35534396 -10.4602997   -8.51648136]
 [ -7.50128728  -8.38169114  -8.39423899]]
Greedy policy learnt:
[L, L, L, L, R, R, R, R, R, S]
```

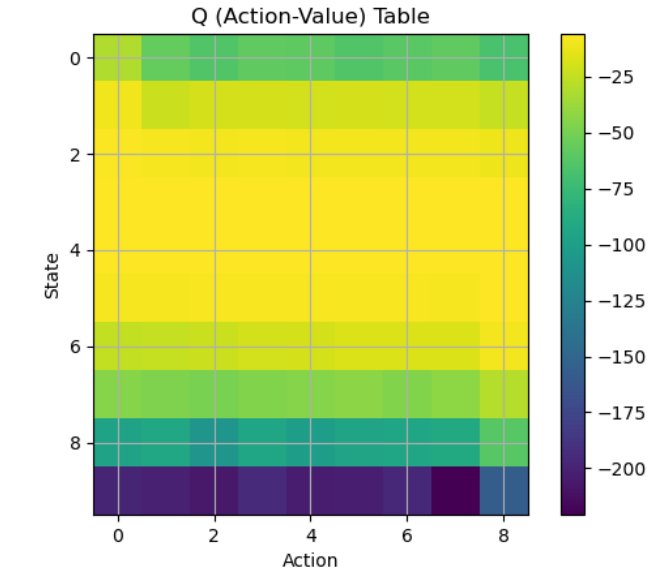
2.1.2. Convergência do Retorno

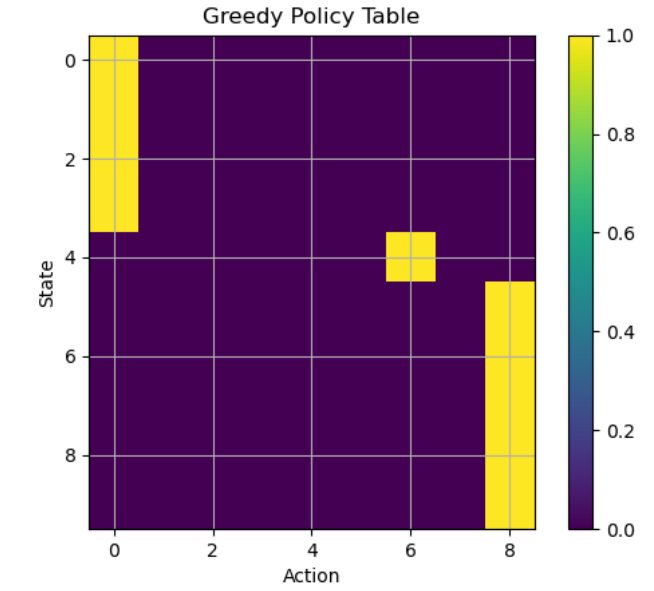
(saída de main.py)



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

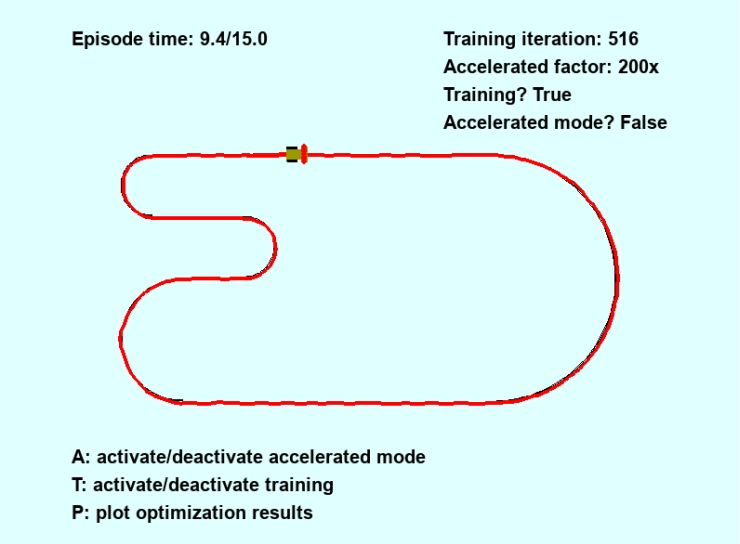
(saída de main.py - *action_value_table and greedy_policy_table*)





2.1.4. **Melhor Trajetória Obtida Durante o Aprendizado**

(saída de main.py)



2.2. Q-Learning

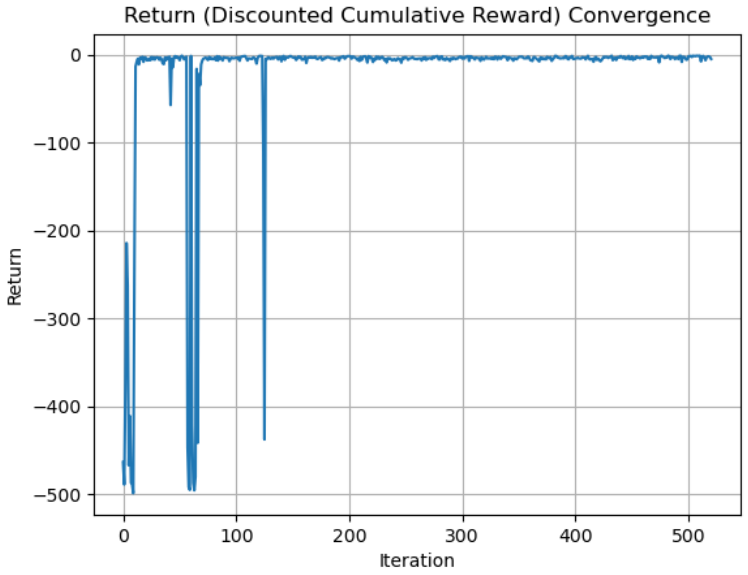
2.2.1. Tabela Ação-Valor e Política *Greedy* Aprendida no Teste com MDP Simples

(saída de test_rl.py)

```
Action-value Table:
[[ -8.96684307  -8.95929389 -10.69079406]
 [-10.39922872  -9.78868087 -11.49794963]
 [-11.29523643 -10.60398953 -12.34118765]
 [-12.18228407 -11.4557451  -90.68531136]
 [-12.91106112 -90.5912236  -11.53398764]
 [-11.14280289 -12.41889203 -10.63968611]
 [-10.33459387 -11.53465027  -9.73939191]
 [ -9.56997012 -10.63736033  -8.83684762]
 [ -8.16056281  -9.76524931  -7.79964041]
 [  0.          -8.07895718  -8.87717464]]
Greedy policy learnt:
[L, L, L, L, R, R, R, R, R, S]
```

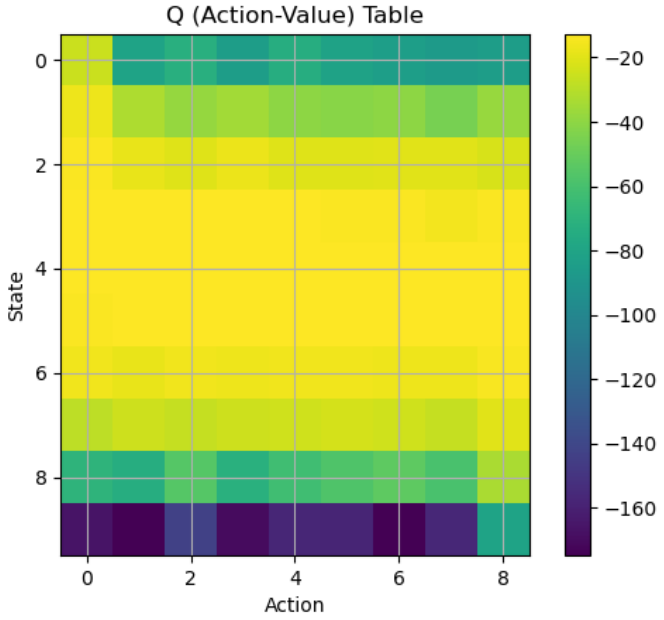
2.2.2. Convergência do Retorno

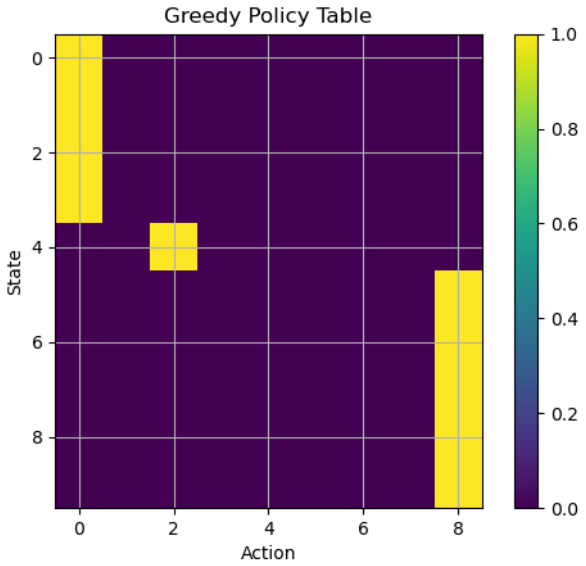
(saída de main.py)



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

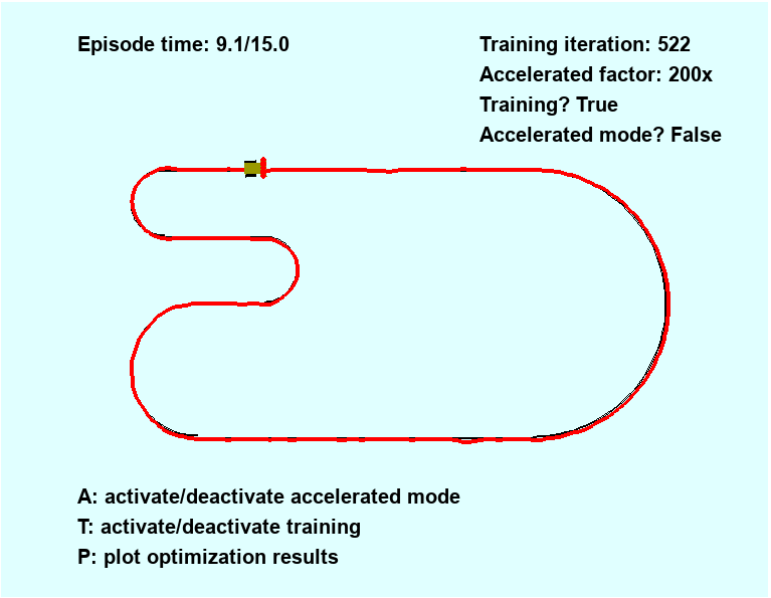
(saída de main.py - *action_value_table and greedy_policy_table*)





2.2.4. Melhor Trajetória Obtida Durante o Aprendizado

(saída de main.py)



3. Discussão dos Resultados

Vemos que as técnicas de aprendizado por reforço obtiveram bons resultados tanto para o teste, quanto para conduzir a trajetória do carrinho pela pista, encontrando políticas ótimas muito parecidas, o que é um bom indicador.

Podemos constatar que no Q-Learning, não há, depois de um tempo, picos nos retornos. Isso acontece porque o algoritmo SARSA foca mais em exploração e em tomar ações aleatórias, enquanto o Q-Learning sempre busca a política ótima greedy, sendo o SARSA mais conservaor e evitando regiões perigosas que impedem a convergência, mesmo que a região esteja mais próxima da greedy.

A política ótima encontrada, como era de se esperar, foi bastante semelhante para os dois algoritmos. A Q-values proporcionalmente era igual para os dois algoritmos, resultando na mesma política ótima.