Universidade Federal de Minas Gerais

Engenharia de Sistemas

Relatório de controle de um robô em um ambiente simulado usando aprendizado por reforço (Q-Learning)

Fundamentos de Inteligência Artificial

Professores: Cristiano Castro e João Paulo Lara

Alunos:
Áquila Oliveira Souza — 2021019327
Arthur Jorge — 2022055718
Felippe Veloso Marinho — 2021072260
Jefferson Pereira de Souza — 2022099049
Josoé Santos Queiroz — 2019026982

Belo Horizonte, MG
9 de dezembro de 2025

Sumário

1 Introdução

O objetivo deste trabalho é documentar a implementação do algoritmo Q-learning seguindo uma política ϵ-greedy para ensinar um agente a navegar em um laboratório simulado e encontrar a saída com o mínimo de passos possível, evitando obstáculos que possam atolá-lo ou destruí-lo.

2 Fundamentação Teórica: Q-Learning

O Q-Learning é um algoritmo de Aprendizado por Reforço *model-free* (livre de modelo) e *off-policy*. O objetivo do algoritmo é aprender uma função de valor de ação $Q(s, a)$, que estima a recompensa acumulada esperada ao executar uma ação a em um estado s e, posteriormente, seguir uma política ótima.

A base do algoritmo é a Equação de Bellman para a atualização iterativa dos valores Q. A regra de atualização utilizada a cada passo de tempo é dada por:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[R + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \tag{1}$$

Onde:

- $Q(s, a)$: Valor atual estimado para o par estado-ação.
- α (taxa de aprendizado): Determina o quanto as novas informações substituem as antigas ($0 < \alpha \leq 1$).
- R: Recompensa imediata recebida após a ação.
- γ (fator de desconto): Determina a importância das recompensas futuras ($0 \leq \gamma \leq 1$).
- $\max_{a'} Q(s', a')$: A estimativa da melhor recompensa futura possível a partir do novo estado s'.

O algoritmo garante a convergência para os valores ótimos $Q^*(s, a)$ desde que todos os pares estado-ação sejam visitados infinitas vezes e a taxa de aprendizado decaia apropriadamente, permitindo que o agente derive uma política ótima $\pi^*(s) = \arg\max_a Q(s, a)$.

3 Modelagem do Problema

3.1 Definição do Ambiente e Recompensas

O espaço de estados é um grid 4×4. As ações do agente são: ir para CIMA, DIREITA, BAIXO e ESQUERDA, desde que os limites do grid permitam. Por exemplo, na posição $(1, 1)$, ele só pode ir para cima ou para a direita.

A função de recompensa é definida da seguinte forma:

- Cada passo para um estado vazio gera uma recompensa de -1.
- Se pisar na lama, a recompensa é -5.
- Se pisar na substância tóxica, a recompensa é -20 e o episódio termina (estado terminal negativo).
- Se encontrar a saída, a recompensa é $+20$ e o episódio termina (estado terminal positivo).

Os estados terminais representam os locais onde as substâncias tóxicas se encontram e também a porta de saída do laboratório, conforme ilustrado na Figura 1.

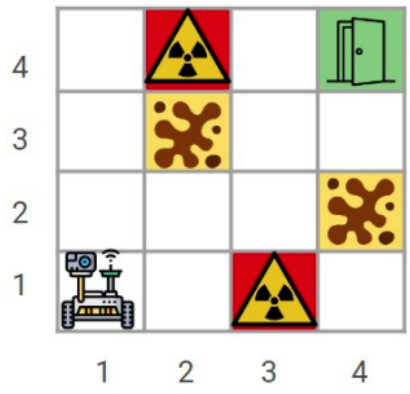


Figura 1: Representação do ambiente do laboratório (Grid 4×4).

4 Estratégias Adotadas

4.1 Implementação Computacional

As características do ambiente foram traduzidas para o código da seguinte maneira:

- **Espaço de estados:** Formado por 16 células, indexadas por coordenadas (x, y) onde $x, y \in \{1, 2, 3, 4\}$.
- **Espaço de ações:** Definido pela lista `actions = ["CIMA", "DIREITA", "BAIXO", "ESQUERDA"]`, mapeados respectivamente para os índices 0, 1, 2 e 3.
- **Dinâmica:** Foi definida a função `rollout` que move o agente uma célula na direção escolhida. Não há verificação de borda na dinâmica de movimento em si dentro desta função; no entanto, as ações inválidas são prevenidas marcando-as com $-\infty$ na tabela Q.
- **Recompensas (`get_reward`):**
  - *Estados terminais negativos (Tóxicos/Radioativos):* As posições $(4, 2)$ e $(1, 3)$ retornam -20.
  - *Obstáculos (Lama):* As posições $(3, 2)$ e $(2, 4)$ retornam -5.
  - *Estado terminal positivo (Saída):* A posição $(4, 4)$ retorna $+20$.
  - *Passo comum:* Retorna -1 para os demais estados.
- **Condições de parada:** Os episódios encerram ao atingir uma recompensa de $+20$ ou -20, ou ao ultrapassar o limite de passos (`limit = 10`).

4.2 Hiperparâmetros

A tabela Q é inicializada com zeros para ações válidas e $-\infty$ para ações inválidas (bordas). Os hiperparâmetros utilizados foram:

- Taxa de aprendizado (α): 0.2
- Fator de desconto (γ): 0.95
- Máximo de iterações por episódio: 10

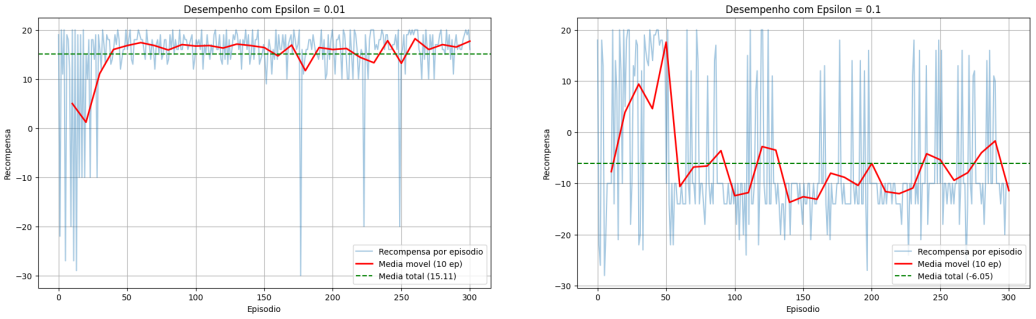
4.3 Política ϵ-greedy

Para a seleção de ações, adotou-se a política ϵ-greedy. O agente escolhe a ação com maior valor Q com probabilidade $1-\epsilon$ (explotação) e uma ação aleatória com probabilidade ϵ (exploração). Conforme especificado no enunciado, em caso de empate nos valores Q durante a escolha gulosa (A^*), o desempate é feito de forma aleatória entre as melhores ações.

5 Análise e Resultados

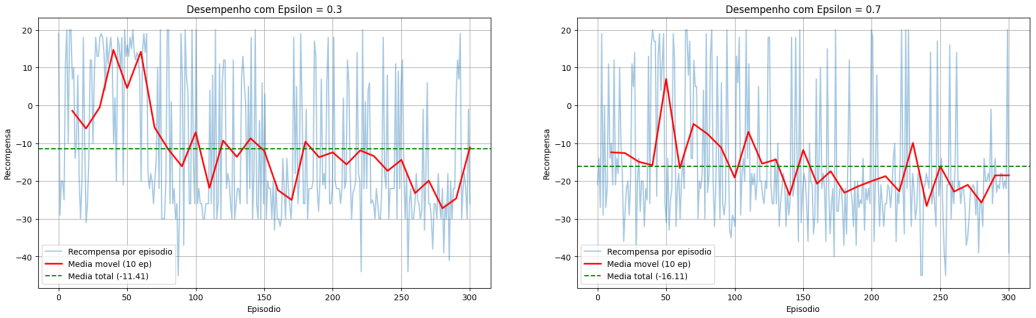
5.1 Variação do Parâmetro ϵ

Abaixo apresentamos os gráficos de recompensa acumulada por episódio e a média móvel (janela de 10 episódios) para diferentes valores de ϵ. O número de episódios foi fixado em 300 para melhor visualização da estabilidade a longo prazo.



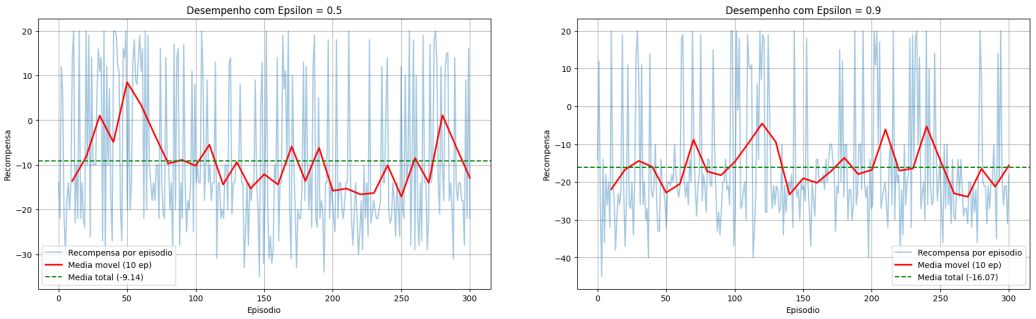
(a) $\epsilon = 0.01$ (Muita Explotação) (b) $\epsilon = 0.1$ (Padrão sugerido)

Figura 2: Comparação de convergência: Baixa exploração vs Padrão.



(a) $\epsilon = 0.3$ (b) $\epsilon = 0.7$

Figura 3: Comparação de convergência com exploração moderada a alta.



(a) $\epsilon = 0.5$ (Equilibrado) (b) $\epsilon = 0.9$ (Muita Exploração)

Figura 4: Impacto de alta taxa de exploração na convergência.

Os experimentos foram realizados variando o parâmetro ϵ em $\{0.01, 0.1, 0.3, 0.5, 0.7, 0.9\}$. As médias totais de recompensa obtidas foram:

- $\epsilon = 0.01$: Média de 15.11 (Melhor desempenho)
- $\epsilon = 0.1$: Média de -6.05
- $\epsilon = 0.3$: Média de -11.41
- $\epsilon = 0.5$: Média de -9.14
- $\epsilon = 0.7$: Média de -16.11
- $\epsilon = 0.9$: Média de -16.07 (Pior desempenho)

Os dados mostram que valores muito baixos de exploração ($\epsilon = 0.01$) resultaram nas melhores médias globais. Isso ocorre devido à natureza perigosa do ambiente: como as penalidades para erros são severas (-20 para o tóxico e -5 para a lama), qualquer movimento aleatório indesejado após o aprendizado do caminho ótimo reduz drasticamente a pontuação acumulada. Com $\epsilon \geq 0.5$, o agente colide frequentemente com obstáculos, explicando as médias negativas.

5.2 Política Ótima Encontrada

Abaixo apresentamos a tabela Q final e a política ótima derivada.

Estado	CIMA	DIREITA	BAIXO	ESQUERDA
(1, 1)	10.48	-0.76	$-\infty$	$-\infty$
(1, 2)	12.41	-4.00	$-\infty$	-0.60
(1, 3)	0.00	0.00	$-\infty$	0.00
(1, 4)	8.43	$-\infty$	$-\infty$	-4.00
(2, 1)	-0.66	12.58	-0.79	$-\infty$
(2, 2)	1.83	14.29	-0.36	1.86
(2, 3)	16.10	-1.00	-4.00	2.36
(2, 4)	-0.36	$-\infty$	-0.93	14.29
(3, 1)	-1.00	-1.00	10.93	$-\infty$
(3, 2)	-4.00	16.06	-0.20	-0.24
(3, 3)	18.00	7.85	2.83	0.00
(3, 4)	19.85	$-\infty$	1.10	-0.20
(4, 1)	$-\infty$	-4.00	8.43	$-\infty$
(4, 2)	$-\infty$	0.00	0.00	0.00
(4, 3)	$-\infty$	20.00	3.22	0.00
(4, 4)	$-\infty$	$-\infty$	0.00	0.00

Tabela 1: Valores Q finais aprendidos (Média de 300 episódios, $\epsilon = 0.01$)

(4,1) $\downarrow$ BAIXO	(4,2) **TÓXICO**	(4,3) $\rightarrow$ DIREITA	(4,4) **SAÍDA**
(3,1) $\downarrow$ BAIXO	(3,2) **LAMA**	(3,3) $\rightarrow$ DIREITA	(3,4) $\uparrow$ CIMA
(2,1) $\rightarrow$ DIREITA	(2,2) $\rightarrow$ DIREITA	(2,3) $\uparrow$ CIMA	(2,4) **LAMA**
(1,1) $\uparrow$ CIMA	(1,2) $\uparrow$ CIMA	(1,3) **TÓXICO**	(1,4) $\uparrow$ CIMA

Tabela 2: Política Ótima obtida com $\epsilon = 0.01$

6 Conclusão

O algoritmo Q-Learning foi capaz de convergir para uma solução ótima no ambiente do laboratório simulado. A introdução da fundamentação teórica de Bellman permitiu compreender como os valores explodiram em magnitude devido à formulação acumulativa, mas ainda assim preservaram a ordem de preferência correta para a navegação.

A análise da variação do parâmetro ϵ evidenciou o dilema exploração-explotação: taxas muito altas de exploração impedem a estabilização da recompensa máxima, enquanto taxas muito baixas aceleram a convergência mas aumentam o risco de mínimos locais. Para este ambiente específico, um $\epsilon = 0.01$ provou ser o mais eficiente.