Controle de Robô em Ambiente Simulado utilizando Q-Learning

Allan Melquíades, Bernardo Fonseca, Gabriel Veloso, Lucas Pimenta, Luan Borges
Fundamentos de Inteligência Artificial
UFMG

***Abstract*—Este trabalho apresenta a implementação do algoritmo Q-Learning aplicado à navegação de um robô em um ambiente simulado do tipo Grid World. O objetivo do agente é alcançar a saída do laboratório evitando obstáculos que causam penalidades ou o término do episódio. São descritos o ambiente, a função de recompensa, a estratégia de aprendizagem e os resultados obtidos para diferentes configurações de exploração.**

I. Introdução

Problemas de navegação em ambientes com obstáculos são frequentes em robótica móvel e em sistemas de controle autônomos. Nesses cenários, um agente precisa escolher, a cada instante, uma ação entre várias alternativas possíveis, levando em conta as consequências futuras de suas decisões. Em vez de seguir um roteiro fixo, o agente pode aprender, por tentativa e erro, quais caminhos tendem a levá-lo com mais segurança e eficiência ao objetivo.

Neste trabalho, estudamos um método baseado em recompensas sucessivas para que um robô aprenda a se deslocar em um ambiente discreto do tipo *Grid World*, com dimensões 4×4. O laboratório simulado contém áreas seguras, regiões de "lama" (com maior custo de travessia), zonas "tóxicas" (que encerram o episódio com falha) e uma saída final. O algoritmo *Q-Learning* ajusta iterativamente uma função de valor associada a pares estado–ação, de forma que, ao longo das interações, o agente passe a preferir ações que resultam em maior recompensa acumulada.

II. Modelagem do Problema

A. Ambiente e Estados ($\mathcal{S}$)

O ambiente consiste em um grid discreto de dimensões 4×4. Cada célula do grid representa um estado s, identificado por coordenadas (x, y), em que x representa a coluna e y a linha, variando de 1 a 4.

A configuração visual do ambiente, incluindo a posição do agente e dos obstáculos, é apresentada na Fig. 1.

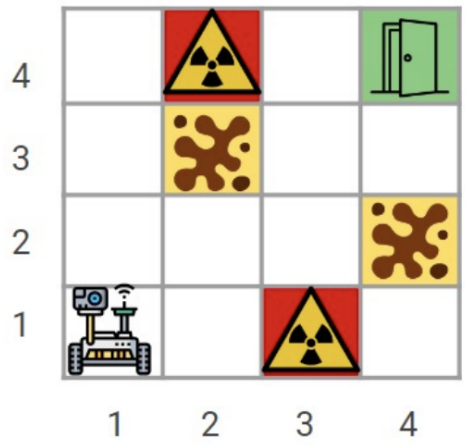

Fig. 1: Representação do ambiente simulado (Grid 4x4). O robô inicia em (1,1); a saída está em (4,4); triângulos representam zonas tóxicas e manchas representam lama.

Identificam-se os seguintes estados especiais:

- **Estado Inicial:** Definido aleatoriamente a cada episódio.
- **Saída (Objetivo):** $(4, 4)$ (estado terminal).
- **Zonas Tóxicas:** $(4, 2)$ e $(1, 3)$ (terminais com falha).
- **Zonas de Lama:** $(3, 2)$ e $(2, 4)$ (alta penalidade).

B. Ações ($\mathcal{A}$)

O agente possui um conjunto de 4 ações discretas disponíveis em cada estado não-terminal:

$$\mathcal{A} = \{\text{Cima}, \text{Direita}, \text{Baixo}, \text{Esquerda}\}. \tag{1}$$

As transições são determinísticas: uma ação move o agente para a célula adjacente correspondente. Caso a ação direcione o agente para fora dos limites do grid, o agente permanece no mesmo estado.

C. Função de Recompensa ($\mathcal{R}$)

A função de recompensa $R(s)$ define o feedback imediato recebido pelo agente. Para incentivar o caminho mais curto e evitar perigos, foram adotados:

- **Movimento Padrão:** -1 (custo de passo).
- **Lama (Mud):** -5.
- **Tóxico:** -20 (penalidade severa e fim do episódio).
- **Saída (Goal):** $+20$ (sucesso e fim do episódio).

$$R(s) = \begin{cases} +20 & \text{se } s = (4,4) \\ -20 & \text{se } s \in \{(4,2),(1,3)\} \\ -5 & \text{se } s \in \{(3,2),(2,4)\} \\ -1 & \text{caso contrário} \end{cases} \quad (2)$$

D. Dinâmica do Episódio

Um episódio inicia com o agente posicionado aleatoriamente e termina quando o agente alcança um estado terminal (saída ou tóxico) ou quando um número máximo de passos é excedido, evitando loops infinitos.

III. Estratégias Adotadas

Para resolver o problema proposto, implementou-se o algoritmo Q-Learning tabular. As subseções a seguir apresentam as principais escolhas de modelagem.

A. Representação da Q-Table

A função valor $Q(s,a)$ foi representada por uma matriz (Q-Table) de dimensões 16×4, em que cada linha corresponde a um dos 16 estados do grid e cada coluna, a uma das ações possíveis (Cima, Direita, Baixo, Esquerda). A tabela foi inicializada com zeros. Para evitar movimentos inválidos, ações que levariam o agente para fora do grid foram marcadas como proibidas e nunca são selecionadas.

B. Atualização dos Valores Q

A atualização segue a equação de Bellman:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R + \gamma \max_{a'} Q(s',a') - Q(s,a) \right], \quad (3)$$

em que s é o estado atual, a a ação tomada, R a recompensa recebida, s' o próximo estado e $\max_{a'} Q(s',a')$ a estimativa do valor ótimo futuro.

C. Política de Seleção de Ações (ϵ-greedy)

Para balancear exploração e explotação, foi adotada a política ϵ-greedy: com probabilidade ϵ, o agente escolhe uma ação aleatória; com probabilidade $1-\epsilon$, escolhe a ação com maior valor $Q(s,a)$. Em caso de empate entre múltiplas ações com mesmo valor máximo, a escolha é feita aleatoriamente entre elas, evitando viés devido à ordem de indexação.

D. Parâmetros de Treinamento

Os hiperparâmetros utilizados são apresentados na Tabela I.

TABLE I: Hiperparâmetros do Algoritmo

Parâmetro	**Valor**
Taxa de Aprendizado (α)	0.2
Fator de Desconto (γ)	0.95
Taxa de Exploração (ϵ)	0.1
Número de Episódios	100
Máximo de Passos por Episódio	10

A cada novo episódio, o estado inicial do agente é sorteado aleatoriamente entre os estados não-terminais, permitindo que o mapa seja explorado a partir de diferentes posições.

IV. Gráficos de Desempenho e Variação de ϵ

A variação do parâmetro ϵ é importante para modular o equilíbrio entre exploração de novos caminhos e explotação do conhecimento adquirido. Para analisar seu impacto, o algoritmo foi executado com três valores: $\epsilon = 0.0$ (política puramente gulosa), $\epsilon = 0.1$ (valor padrão) e $\epsilon = 0.5$ (alta exploração).

A convergência e a qualidade da política aprendida foram avaliadas pela recompensa acumulada por episódio e por sua média móvel, que suaviza o ruído das interações iniciais.

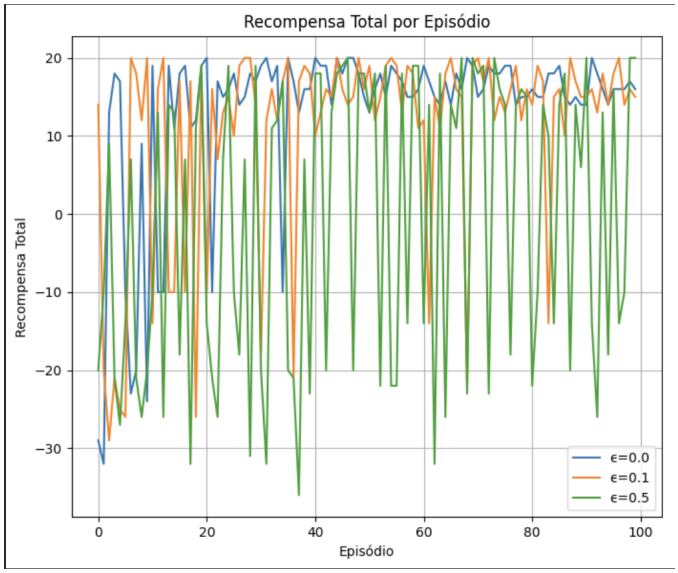

Fig. 2: Recompensa total obtida por episódio para diferentes valores de ϵ.

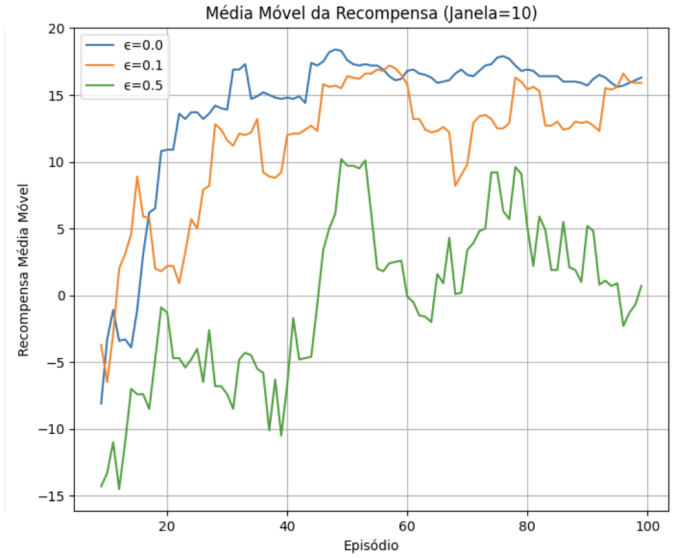

Fig. 3: Média móvel da recompensa total por episódio.

Valores mais altos de ϵ (como $\epsilon = 0.5$) resultam em recompensas mais variáveis devido à alta frequência de ações

aleatórias. O valor intermediário ($\epsilon = 0.1$) apresentou um aumento mais estável da recompensa média, indicando que o agente rapidamente passa a explorar a política aprendida de forma eficiente.

V. Política Ótima Aprendida

A política ótima $\pi^*(s)$ derivada da Q-Table final consiste em selecionar, para cada estado, a ação com o maior valor $Q(s, a)$.

A. Mapeamento Estado-Ação (Grid Policy)

A Tabela II ilustra a direção preferencial aprendida pelo agente em cada célula do ambiente. As setas indicam a ação de maior valor Q ($\arg\max_a Q(s, a)$). Células marcadas como Tóxico ou Lama representam estados terminais ou de penalidade.

TABLE II: Representação da Política Ótima no Grid 4x4

(4,1) ↓	(4,2) **Tóxico**	(4,3) →	(4,4) **Saída**
(3,1) ↓	(3,2) **Lama**	(3,3) →	(3,4) ↑
(2,1) →	(2,2) →	(2,3) ↑	(2,4) **Lama**
(1,1) →	(1,2) ↑	(1,3) **Tóxico**	(1,4) ↑

B. Trajetória Otimizada

Ao executar um teste de validação partindo do estado inicial $(1, 1)$ com $\epsilon = 0$ (política puramente gulosa), o agente segue a trajetória

$$(1, 1) \rightarrow (1, 2) \rightarrow (2, 2) \rightarrow (2, 3) \rightarrow (3, 3) \rightarrow (3, 4) \rightarrow (4, 4)$$

contornando as regiões de lama e as zonas tóxicas em 6 passos até o objetivo.

VI. Discussão dos Resultados

Os resultados obtidos mostram que o algoritmo Q-Learning convergiu para uma solução segura e eficiente. A Q-Table final, para $\epsilon = 0.1$, atribui valores bastante negativos às ações que levam a estados tóxicos ou de lama, criando uma espécie de "zona de repulsão" em torno dessas regiões e favorecendo caminhos mais seguros.

A trajetória aprendida a partir de $(1, 1)$ é compatível com a presença de obstáculos e com o desenho do mapa, e apresenta um número reduzido de passos até o objetivo. Os gráficos de recompensa média indicam que, após os episódios iniciais de exploração, o desempenho estabiliza em torno de valores elevados, o que confirma que o agente aprendeu não apenas a alcançar a saída, mas a fazê-lo evitando penalidades desnecessárias.