Trabalho Prático 2
Introdução a Inteligencia Artificial

João Vitor Soares Santos
2023002138

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Belo Horizonte - MG - Brasil

joaosoares@dcc.ufmg.br

1 Introdução

Este trabalho explora conceitos fundamentais de aprendizado por reforço, com foco na aplicação prática do algoritmo de Q-learning [1]. O objetivo principal é implementar e avaliar a eficácia do Q-learning na aprendizagem de políticas ótimas para agentes autônomos em ambientes discretizados. O Q-learning, como um algoritmo off-policy, busca maximizar as recompensas acumuladas ao longo do tempo, ajustando a política do agente com base na interação com o ambiente. Para isso, será realizada a implementação do algoritmo, seguido de uma análise de desempenho em diferentes cenários, com o intuito de comparar os resultados obtidos em termos de eficiência e convergência da política aprendida. Este estudo visa aplicar o Q-learning para resolver problemas de navegação e decisão em ambientes com estados e ações discretas, levando em consideração o impacto de fatores como o fator de desconto e a taxa de aprendizado.

2 Modelagem

A modelagem foi definida em três tópicos principais: a definição de estados, que representam as configurações possíveis do problema; a função sucessora, responsável por gerar novos estados a partir do atual; e as estruturas de dados utilizadas nos algorítimos.

2.1 Estado

Neste trabalho, um **estado** é definido como uma estrutura que contém duas informações essenciais:

1. **Coordenadas x e y**: Representam a posição no mapa discretizado.
2. **Custo/Recompensa**: Indica o custo/recompensa de percorrer a posição no mapa discretizado.

Essa definição de estado é fundamental para que os algoritmos de busca possam navegar eficientemente pela árvore de busca e calcular os custos dos caminhos para construir a solução final.

2.2 Função sucessora

Neste trabalho, os movimentos permitidos para transitar de um estado para o outro são: esquerda, direita, baixo e cima. Dado que o mapa é representado por uma matriz bidimensional, a função sucessora primeiro verifica se é possível fazer cada movimento sem ultrapassar os limites da matriz. Em seguida, verifica-se se o movimento não leva o agente a uma coordenada representando um terreno do tipo parede, que é considerado intransitável. Por fim, a função `get_next_state`, que foi desenvolvida nesse trabalho, retorna um vetor contendo apenas os movimentos válidos — esquerda, direita, cima e baixo — que permanecem dentro dos limites do mapa e não levam a uma coordenada do tipo parede.

2.2.1 Agente

O agente é responsável por selecionar ações com base no estado atual e na política de aprendizado utilizando a função sucessora e métodos aleatórios. Ele aprende por meio da interação com o ambiente, ajustando a matriz Q e suas estratégias para maximizar a recompensa esperada.

2.2.2 Recompensa

A recompensa r é um valor escalar que o agente recebe após realizar uma ação no ambiente. Ela serve como um sinal para o agente avaliar a qualidade de suas ações. A meta do agente é maximizar a soma das recompensas recebidas ao longo do tempo.

Neste trabalho, foram definidas duas tabelas de recompensa distintas para o agente, a fim de avaliar a eficácia de suas ações em diferentes cenários. As duas tabelas de recompensa definidas são:

- **Tabela de Recompensa com Valores Positivos e Negativos (Punições)**: Nesta tabela, as recompensas podem ser tanto positivas quanto negativas.
- **Tabela de Recompensa com Valores Exclusivamente Positivos**: Nesta tabela, todas as recompensas são positivas, indicando que qualquer ação tomada pelo agente é considerada benéfica em alguma medida.

Essas duas tabelas de recompensa permitem a análise de diferentes comportamentos do agente e o impacto das punições no aprendizado. A escolha entre elas depende do tipo de problema e dos objetivos desejados para o treinamento do agente. Em ambos os casos, o agente visa maximizar a soma das recompensas recebidas ao longo do tempo, ajustando suas ações com base nos valores de recompensa associados a cada transição de estado.

2.3 Estruturas de dados

Para este trabalho, foi utilizada a estrutura de dados *matriz* Q, que armazena os valores de $Q(e, a)$ para cada par de estado e e ação a. Esta matriz é fundamental no Q-learning, pois ela guarda as estimativas de valor para as ações que o agente pode tomar em cada estado do ambiente. À medida que o agente interage com o ambiente e recebe recompensas, os valores de $Q(s, a)$ são atualizados para refletir a expectativa de recompensa acumulada a longo prazo.

3 Algorítimos

3.1 Algoritmo Q-Learning

O Q-Learning é um algoritmo de aprendizado por reforço que busca encontrar a política ótima de um agente, maximizando a soma das recompensas ao longo do tempo. Ele utiliza uma tabela Q, onde cada entrada $Q(s, a)$ representa o valor esperado de uma ação a em um estado s, atualizado pela fórmula:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right]$$

O algoritmo Q-Learning implementado neste trabalho utiliza os seguintes parâmetros e configurações:

- **Taxa de aprendizado (α)**: 0.1, que controla a atualização dos valores da tabela Q com base nas recompensas recebidas.
- **Taxa de desconto (γ)**: 0.9, que determina o peso das recompensas futuras em relação às recompensas imediatas.
- **Exploração**: A exploração é realizada usando o método ϵ-greedy, com $\epsilon = 0.1$. Isso significa que 10% das ações são escolhidas aleatoriamente (exploração), enquanto 90% são baseadas na ação com o maior valor Q (exploração).

O agente escolhe ações usando uma estratégia ϵ-greedy, explorando ou explorando com base em ϵ.

O Q-Learning foi aplicado em três modos diferentes:

3.1.1 Modo Padrão

No modo padrão, o agente recebe recompensas constantes, positivas ou negativas, e o objetivo é aprender a política que maximiza a recompensa total.

3.1.2 Modo Positivo

No modo positivo, as ações bem-sucedidas sempre geram recompensas positivas. Em resumo, ações erradas não são punidas, apenas não recompensadas.

3.1.3 Modo Estocástico

No modo estocástico, embora o agente siga uma direção específica baseada na tabela Q, existe uma probabilidade de desvio da trajetória escolhida. Por exemplo, se o agente decide seguir para um estado adjacente, há 10% de chance de ele mover-se para a direita na diagonal e 10% de chance de ir para a esquerda, na diagonal oposta. Esse comportamento simula incertezas no ambiente, forçando o agente a aprender a lidar com a imprevisibilidade nas ações.

Esses três modos permitiram testar o Q-Learning em diferentes cenários, observando o desempenho do agente com diferentes tipos de recompensa.

4 Análise quantitativa

O tempo de execução e a convergência do algoritmo Q-Learning dependem do número de atualizações realizadas durante o treinamento, que está diretamente ligado ao número de passos definidos. O modo de exploração também influencia, já que em um modo determinístico o agente toma decisões mais previsíveis, enquanto em um modo estocástico há uma introdução de aleatoriedade, o que pode aumentar o tempo de execução. A quantidade de atualizações necessárias para a convergência pode variar com a complexidade do mapa, da política gerada e da tabela de recompensas, impactando diretamente a duração do processo.

No modo de recompensas positivas, o algoritmo Q-Learning não convergiu devido à falta de punições adequadas. O agente foi recompensado por ações em áreas de baixo custo, como a grama e a grama alta, sem ser incentivado a explorar de forma eficiente. Isso resultou em um aprendizado lento e na formação de uma política ineficaz, que nem mesmo convergiu. Esse comportamento reforça a ideia de que é necessário equilibrar recompensas positivas e negativas para guiar o agente ao objetivo de maneira eficiente. Como Maquiavel disse, "é melhor ser temido do que respeitado": as punições ajudam a direcionar o agente para escolhas mais acertadas, enquanto as recompensas positivas sozinhas não são suficientes para garantir uma convergência eficaz.

Na Figura 1 é possível ver uma comparação de tempo de execução e quantidade de atualizações feitas na matriz Q em diferentes mapas com uma quantidade de 300 mil episódios. No gráfico é possível ver que o tempo de execução e a quantidade de atualizações do Q-Learning padrão são menores do que no modo estocástico , afinal no modo padrão a exploração controlada e determinística gera menos variação de estados e atualizações, o que impacta diretamente no tempo de execução. Por outro lado, o Q-Learning estocástico introduz variabilidade nas decisões do agente, ao não garantir que uma ação seja tomada, o que acontece em cenários do mundo real.

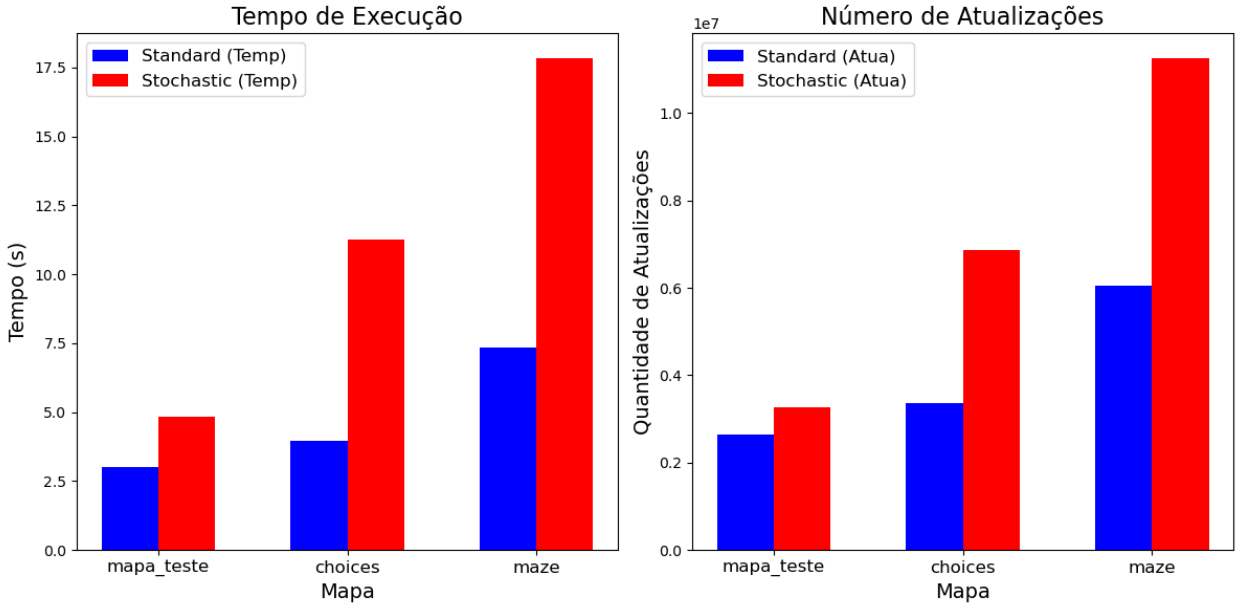

Figura 1: Comparação de tempo de execução e quantidade de execuções em diferentes mapas

Na Figura 2 é possível ver como as políticas finais do modo padrão e estocástico variaram, mesmo que nos minimos detalhes. No Q-Learning Padrão, a política tende a ser mais determinística, com

o agente tomando decisões focadas e eficientes, o que leva a uma convergência mais rápida para o caminho ótimo. Por outro lado, no Q-Learning Estocástico, a política é mais exploratória e menos estável devido à aleatoriedade nas ações e recompensas, resultando em uma aprendizagem mais dispersa e uma maior variação nas decisões tomadas pelo agente.

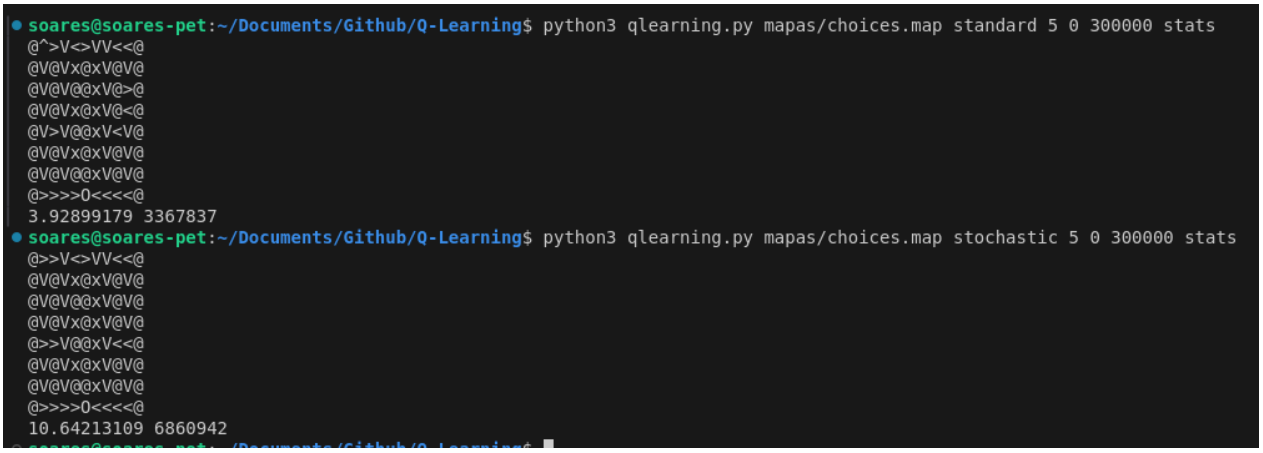

Figura 2: Comparação de políticas

5 Resultados e Considerações Finais

Em resumo, o Q-Learning Padrão se mostrou mais eficiente em termos de tempo de execução e número de atualizações, devido à sua abordagem determinística e controlada. No entanto, o Q-Learning Estocástico, apesar de ser mais lento e menos eficiente, reflete melhor a realidade, já que introduz aleatoriedade nas ações do agente, algo que é comum em cenários do mundo real, onde a incerteza e variabilidade das situações impactam diretamente nas escolhas e nos resultados. Portanto, embora o padrão seja preferível em termos de desempenho, o estocástico é mais representativo de condições práticas.

Por meio da resolução desse trabalho, foi possível praticar conceitos de algoritmos de aprendizado por reforço. Além disso, ficou evidente a relação entre política e as recompensas com o tempo de execução e também a importância de escolher abordagens que melhor se adéquam ao contexto do problema. Esse processo proporcionou não apenas um aprendizado técnico, mas também uma visão mais crítica sobre como escolher a melhor politica para os algorítimos de aprendizado por reforço.

Referências

[1] Wikipedia contributors. *Q-learning*. Accessed: 2025-02-03. 2025. URL: `https://en.wikipedia.org/wiki/Q-learning`.