EXERCÍCIO 6 - APRENDIZAGEM POR REFORÇO

Bruno Andreghetti Dantas*

Email: brunoandreghetti@gmail.com

Abstract— The purpose of this exercise is to experiment with the Q-learning algorithm to teach an agent how to navigate a maze to find a treasure. The policy learned by the agent is defined using a state-to-action table.

Keywords— Reinforcement Learning, Maze, Q-learning, Finite Markov Decision Process, Python.

Resumo— Esse exercício visa experimentar com o algoritmo Q-learning para treinar um agente na navegação através de um labirinto a fim de encontrar um tesouro. A política aprendida é representada utilizando uma tabela que leva de um estado para a próxima ação.

Palavras-chave— Aprendizagem por Reforço, Labirinto, Q-learning, Processo de Decisão de Markov Finito, Python.

1 Introdução

O aprendizado através da experimentação é um dos principais métodos utilizados pelos seres humanos para dominar a interação com o ambiente no qual eles estão inseridos. Seja para manter uma conversação ou dirigir um veículo, estamos sempre atentos às condições do nosso ambiente e tentamos influenciar essas condições através do nosso comportamento. (Sutton and Barto, 2018)

A aprendizagem por reforço é uma abordagem computacional com o objetivo de simular esse processo pelo qual aprendemos a realizar tarefas. Diferentemente da aprendizagem supervisionada e da não-supervisionada, a aprendizagem por reforço não exige qualquer conhecimento prévio do ambiente no qual o agente está inserido.

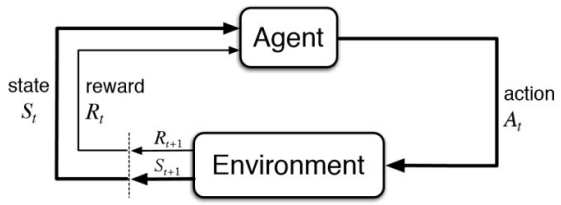

Figura 1: Malha fechada de um sistema de aprendizagem por reforço.

O Q-learning é um dos principais algoritmos da área. Ele consiste em aprender uma política de ação para cada estado. Esse algoritmo visa obter uma política que maximize a recompensa futura. Deste modo, a atualização da política de ação para um dado estado é dada pela Equação 1:

$$Q(s_t, a_t) \leftarrow (1-\alpha)Q(s_t, a_t)+\alpha(r+\gamma \max_a Q(s_{t+1}, a)) \quad (1)$$

Sendo s_t e a_t o estado e a ação no instante t, respectivamente, α a taxa de aprendizagem, γ o fator de desconto e $Q(s, a)$ a recompensa esperada ao tomar uma ação a partir de um dado estado.

A taxa de aprendizagem α determina a importância das novas informações adquiridas pelo agente a cada ação. Se $\alpha = 0$, as políticas do agente não são atualizadas e portanto ele toma suas decisões baseado no seu conhecimento prévio. Se $\alpha = 1$, o agente sempre considera a última experiência obtida ao tomar suas decisões. (Sutton and Barto, 2018)

O fator de desconto γ é um fator multiplicativo que reduz a relevância dos ganhos futuros. Um valor baixo para γ faz com quue o agente se torne "imediatista", incapaz de agir em favor de uma alta recompensa futura. Em contrapartida, valores de γ iguais ou acima de 1 podem fazer com que o treinamento divirja.

Há ainda o parâmetro ϵ, que representa a probabilidade de, durante o treinamento, o agente escolher uma ação aleatória em vez de selecionar a melhor opção da tabela. Esse parâmetro evita a convergência prematura do treinamento permitindo que o agente explore novos caminhos.

2 Metodologia e Desenvolvimento

Para desenvolvimento do exercício foi utilizado a linguagem Python e a plataforma Jupyter. O código-fonte provido foi adaptado a fim de atender às necessidades do exercício.

2.1 O Labirinto

O ambiente utilizado para desenvolvimento do exercício foi um labirinto bidimensional de tamanho arbitrário. No labirinto há o agente a ser treinado e um objetivo, ou "tesouro", onde o agente obtém a recompensa máxima. O código provido para o exercício se utiliza de um algoritmo gerador que garante a existência de apenas 1 caminho até o objetivo. Para tornar o exercício mais interessante, foram retiradas algumas das paredes do labirinto a fim de aumentar o número de caminhos válidos a serem encontrados.

Cada episódio do treinamento é encerrado quando o agente chega até o tesouro ou quando

o número máximo de passos dados pelos agente é atingido. O agente inicia cada episódio no canto inferior esquerdo do labirinto e o tesouro sempre é posicionado próximo ao canto superior direito, a fim de maximizar a distância mínima entre o agente e o tesouro.

As recompensas obtidas pelo agente para cada evento ocorrido estão descritas na Tabela 1.

Evento	Recompensa
Tentar se mover em direção a uma parede ou para fora do labirinto	-1
Se mover para um novo ponto válido do labirinto	0
Alcançar o objetivo em N passos	$\frac{10}{N}$

Tabela 1: Eventos do ambiente e suas recompensas.

Para esse exercício, foram gerados labirintos de tamanho 10×20 e 40×40. Todas as combinações de parâmetros foram utilizadas com o mesmo labirinto a fim de tornar a comparação do desempenho mais justa.

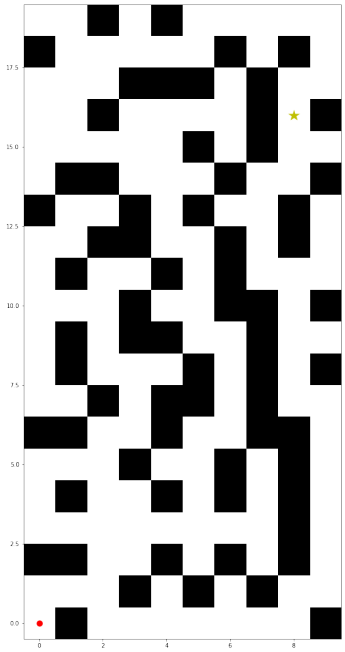

Figura 2: Labirinto 10×20 utilizado para teste dos parâmetros.

2.2 A Tabela Q

Para representar a política do agente, foi utilizada uma tabela de tamanho SxA, sendo S o número de estados possíveis para o agente e A o número de ações possíveis em cada estado. Em particular, para o labirinto, a tabela possui $S = X * Y$ e $A = 4$, sendo X e Y iguais à largura e à altura do labirinto, respectivamente.

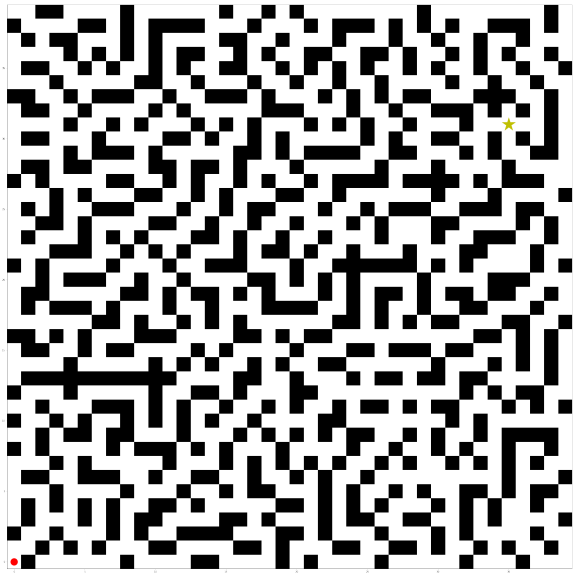

Figura 3: Labirinto 40×40 utilizado para teste dos parâmetros.

Em cada elemento da tabela há a expectativa de recompensa futura para uma determinada ação a tomada a partir de um estado s.

Para uma tabela corretamente preenchida - ou seja, para um agente corretamente treinado - o caminho ótimo para chegar ao objetivo é dado pela sequência de ações de maior expectativa de recompensa futura em cada um dos estados alcançados pelo agente no trajeto.

2.3 Escolha dos Parâmetros

Para observação do comportamento do algoritmo, foi selecionado um conjunto de valores a serem testados para os parâmetros, dispostos na Tabela 2.

Parâmetro	Valores
α	0.2, 0.5, 1
γ	0.9, 0.95
ϵ	0, 0.2, 0.5
$X \times Y$	10×20, 40×40

Tabela 2: Valores testados para os parâmetros.

3 Resultados

Foram testadas todas as combinações de parâmetros para cada um dos labirintos das Figuras 2 e 3 e os resultados foram dispostos na Tabela 3. O número de épocas representa a primeira época em que o agente obteve o melhor desempenho durante o treinamento. O número de passos foi medido sempre com $\epsilon = 0$ após o término do treinamento,

a fim de se obter o caminho considerado ótimo pelo agente.

Parâmetros	# de épocas		# de passos	
	10×20	40×40	10×20	40×40
$\alpha = 0.2, \gamma = 0.9, \epsilon = 0$	91	356	30	84
$\alpha = 0.5, \gamma = 0.9, \epsilon = 0$	36	196	30	86
$\alpha = 1, \gamma = 0.9, \epsilon = 0$	33	159	30	84
$\alpha = 0.2, \gamma = 0.95, \epsilon = 0$	32	514	30	84
$\alpha = 0.5, \gamma = 0.95, \epsilon = 0$	54	151	42	96
$\alpha = 1, \gamma = 0.95, \epsilon = 0$	70	150	30	90
$\alpha = 0.2, \gamma = 0.9, \epsilon = 0.2$	224	1334	30	90
$\alpha = 0.5, \gamma = 0.9, \epsilon = 0.2$	122	1890	30	84
$\alpha = 1, \gamma = 0.9, \epsilon = 0.2$	89	840	30	89
$\alpha = 0.2, \gamma = 0.95, \epsilon = 0.2$	495	839	30	84
$\alpha = 0.5, \gamma = 0.95, \epsilon = 0.2$	245	1924	42	84
$\alpha = 1, \gamma = 0.95, \epsilon = 0.2$	413	1615	30	90
$\alpha = 0.2, \gamma = 0.9, \epsilon = 0.5$	236	1214	30	84
$\alpha = 0.5, \gamma = 0.9, \epsilon = 0.5$	284	1615	30	84
$\alpha = 1, \gamma = 0.9, \epsilon = 0.5$	216	1976	30	84
$\alpha = 0.2, \gamma = 0.95, \epsilon = 0.5$	467	1338	30	84
$\alpha = 0.5, \gamma = 0.95, \epsilon = 0.5$	161	752	30	84
$\alpha = 1, \gamma = 0.95, \epsilon = 0.5$	625	1699	30	84

Tabela 3: Resultados para todas as combinações de parâmetros.

Ao final de cada treinamento, é possível visualizar um mapa de calor indicando os caminhos por onde o agente mais passou como na Figura 4 - e portanto os pontos onde sua política foi melhor ajustada.

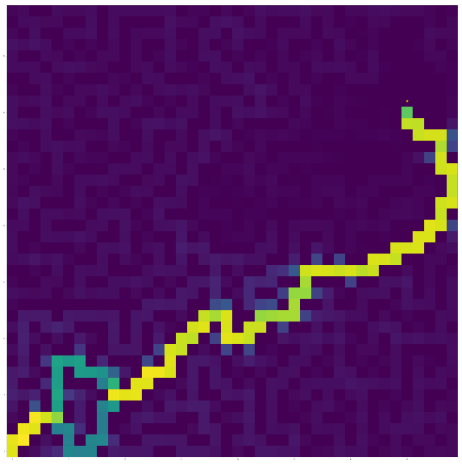

Figura 4: Mapa de calor para um dos treinamentos do labirinto 40x40.

Na bifurcação em azul da Figura 4, o agente aprendeu dois caminhos igualmente eficientes e alternou entre eles ao acaso em função de ϵ.

Também é possível visualizar a política aprendida pelo agente através de setas como na Figura 5, em que setas verdes representam recompensa positiva, setas vermelhas representam recompensa negativa e o tamanho das setas é proporcional à magnitude da recompensa esperada.

A progressão do agente ao longo do treinamento também é observável através da medição da quantidade de passos necessários para que ele complete o labirinto a cada episódio. Caso o agente não consiga atingir o objetivo antes do número de passos limite, é atribuído um número de passos infinito para esse episódio.

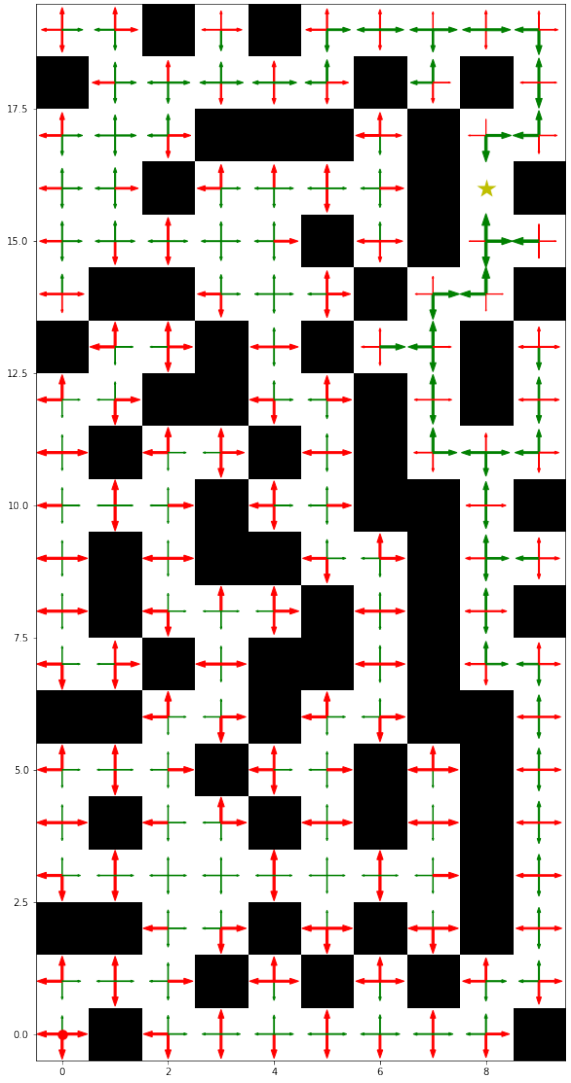

Figura 5: Setas indicando as políticas aprendidas pelo agente.

Um aspecto interessante, observável através das Figuras 6 e 7, é o efeito do parâmetro ϵ no treinamento. No caso em que $\epsilon = 0$, o treinamento converge e se mantém constante após obter qualquer caminho válido até o objetivo (Figura 6). Já quando $\epsilon > 0$, como por exemplo na Figura 7 onde $\epsilon = 0.5$, o agente continua explorando o ambiente após encontrar uma rota válida, possibilitando a descoberta de um caminho melhor.

Para o labirinto 10×20, como visto na Tabela 3, o caminho ótimo visto na Figura 8 consiste de 30 passos:

Algumas configurações convergiram para o caminho subótimo de 42 passos visto na Figura 9:

4 Análise e Conclusões

O algoritmo Q-learning de aprendizagem por reforço se mostrou capaz de obter com considerável

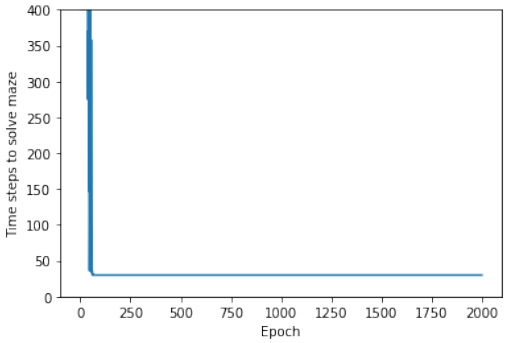

Figura 6: Progressão do agente com $\epsilon = 0$.

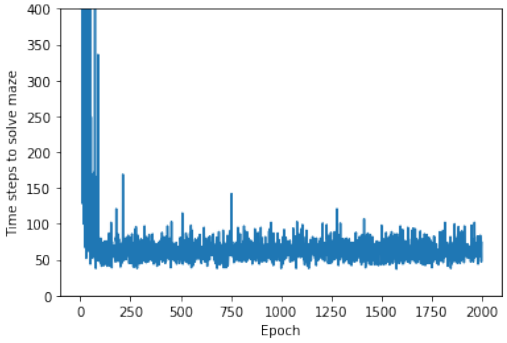

Figura 7: Progressão do agente com $\epsilon = 0.5$.

agilidade a solução ótima para ambos os labirintos. Para algumas combinações de parâmetros, uma solução boa, porém subótima foi obtida.

A velocidade dessa implementação do Q-learning se deu pelo fato de que foi utilizada uma tabela para descrever a política de ação. Em sistemas contínuos, é necessário discretizar as variáveis de estado para adotar uma abordagem como esta ou utilizar um aproximador universal de funções para mapear os estados contínuos para as estimativas de recompensa futura.

Foi observada também a importância de balancear a exploração para obtenção de novas informações acerca do ambiente e o aproveitamento do conhecimento já adquirido para tomada de decisão acerca do próximo passo a ser tomado.

A aprendizagem por reforço é uma área em constante evolução e, por funcionar sem muita informação prévia sobre o problema a ser resolvido, é a área de pesquisa mais próxima atualmente da inteligência artificial genérica, capaz de resolver uma ampla gama de problemas - e não apenas um problema específico para o qual ela foi desenvolvida.

Referências

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement learning: An introduction*, MIT press.

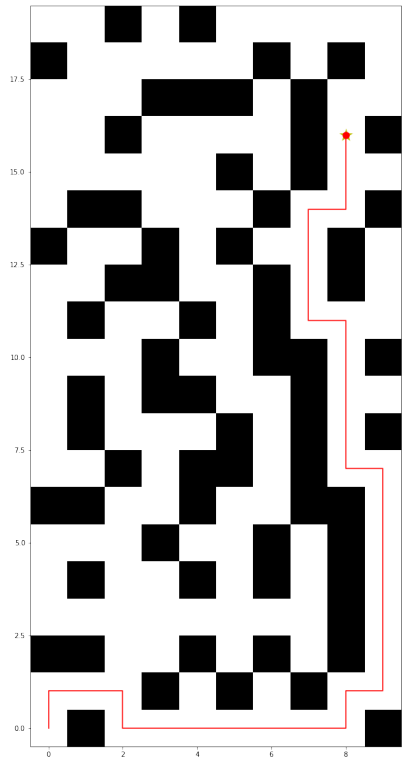

Figura 8: Solução ótima para o Labirinto 10×20.

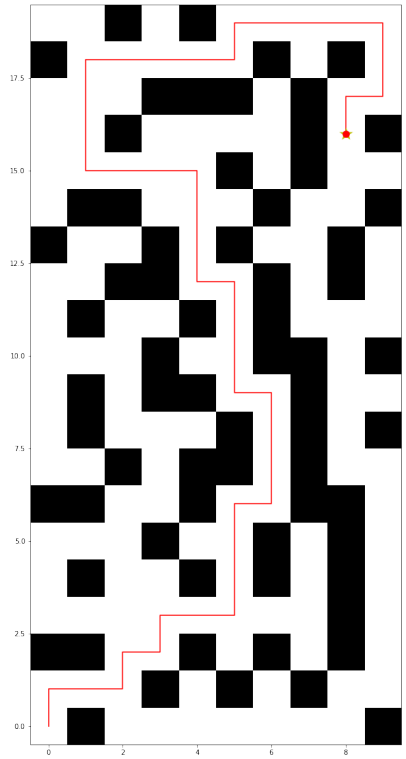

Figura 9: Solução subótima para o Labirinto 10×20.