

Inteligência Artificial – Lista 2

Lucas Martinuzzo Batista

1º. Semestre 2021



Exercícios

- **Nesta segunda lista exploraremos o algoritmo Q-learning. Para isso, teremos um problema de navegação com informações conhecidas por nós, mas que precisarão ser descobertas pelo agente. O objetivo será simular o Q-learning para o agente conseguir identificar uma política “boa”. Consideraremos duas variações de um problema com 12 estados, apenas um estado terminal e ações com confiabilidade exatamente como no exercício visto nas aulas anteriores. As recompensas e confiabilidade das ações serão desconhecidas inicialmente pelo agente.**

Exercício 1

recompensas

Estado inicial

-1	-1	-1	+100
-1	-100	-1	-1
-1	-1	-1	-1

Estado terminal

Exercício 1

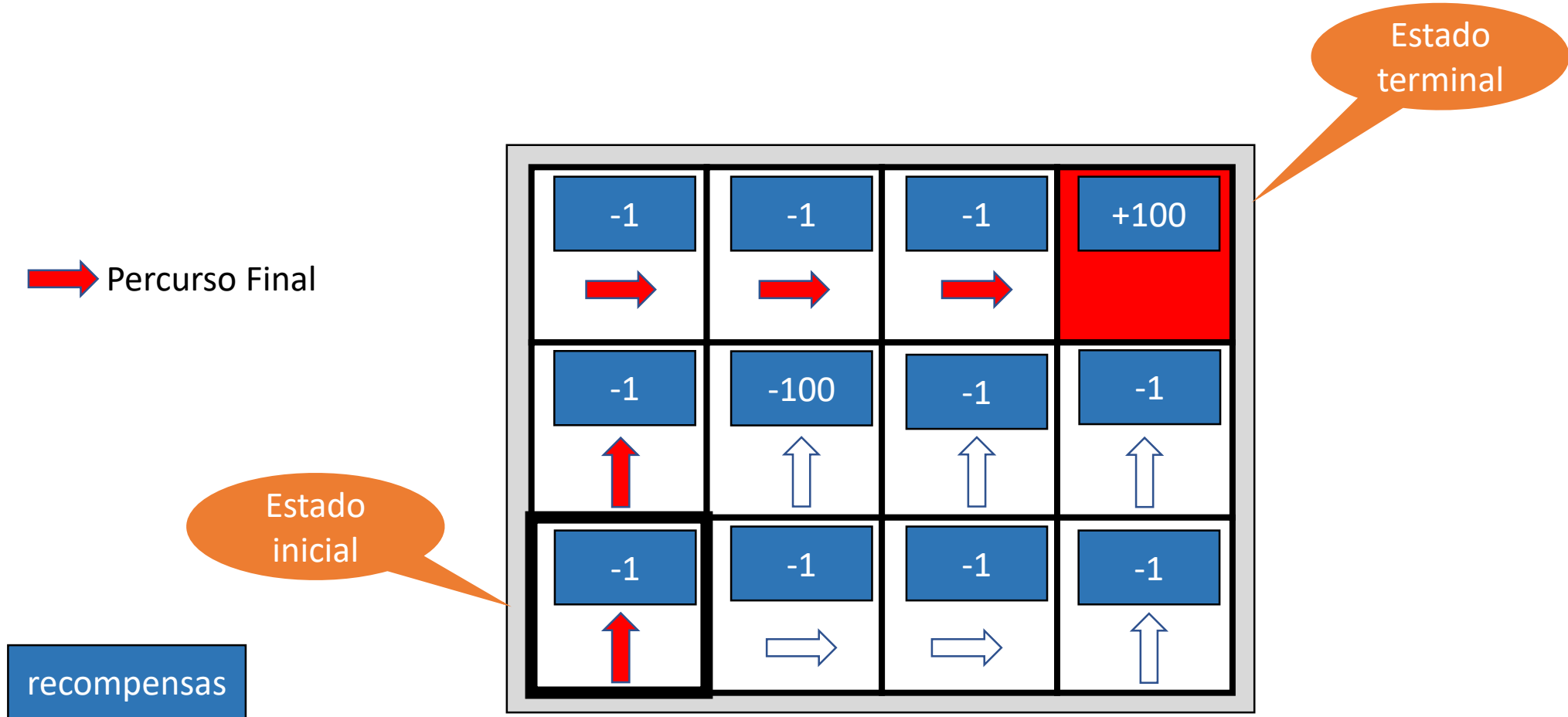
- **O resultado do Q-learning corresponde ao esperado intuitivamente?**

Sim. Apesar da aleatoriedade atrapalhar na velocidade de convergência, ele busca a maior recompensa e a partir daí se direciona mais rapidamente.

- **Quantas rodadas foram necessárias para chegar a este resultado?**

O programa demorou 23 iterações para convergir (ficar 5 episódios com as mesmas escolhas de ação), porém ele se mostrou muito sensível a semente de aleatoriedade do numpy. Mesmo mantendo 10% de chance de ir para um estado aleatório, se eu trocar a semente, ele pode levar 100 iterações, e às vezes até mais.

Ações com maior qualidade ao convergir



Exercício 2

recompensas

Estado inicial

-1	-1	-1	+100
-1	-100	-1	-1
-1	-1	-1	+10

Estado terminal

Recompensa modificada

Exercício 2

- **O resultado do Q-learning corresponde ao esperado intuitivamente?**

Sim. Porém, dependeu ainda mais da aleatoriedade. Se ele descobre o valor com + 10 antes de descobrir o +100, ele fica em loop por um bom tempo até ir para o estado final . Porém, se ele descobre o valor com +100 sem ter passado pelo +10, ele tende a se comportar como no exercício 1 e tende a ir para o +100 muito rapidamente.

É importante frisar que meu algoritmo tem algumas particularidades. A taxa de aleatoriedade, isto é, a chance de ir para um estado aleatório independente do esperado, é de 10%. Além disso, função da taxa de teste exploratório foi modificada (detalhes no próximo slide).

- Em caso negativo, é possível ajustar as informações dos estados para induzir o Q-learning a produzir o resultado desejado?

-

Taxa de teste de aleatoriedade

Assim como comentei com o professor em aula, modifiquei a taxa de teste de aleatoriedade para a seguinte:

$$f = \begin{cases} 1, & N = 0 \\ \frac{Q}{N}, & Q > 0 \\ Q * N, & c.c \end{cases}$$

Desta maneira, caso a Qualidade seja negativa (devido a recompensas negativas), quanto mais ações foram feitas para este estado, mais negativa ela se torna. E caso a qualidade seja positiva, ela é amortecida. Assim, evita-se loops eternos e “ótimos locais”.