Módulo 04 – Tarefa 07

- Disciplina: Reinforcement Learning
- Faculdade: Unyleya / Pós-Graduação em IA
- Aluno: João Luiz Grave Gross

Sumário

1. Enunciado

Considere a aplicação '**Um conto de Natal**' apresentada na última aula. Você deverá implementar a aplicação de acordo com as instruções a seguir, considerando a representação de cidade vista em aula:

Um robô taxista, chamado de agente, deverá ser colocado aleatoriamente em qualquer uma das células e fazer com que ele se mova também aleatoriamente para uma das células vizinhas; a posição dele é chamada de estado. O movimento vai se repetindo e termina em três ocasiões: quando chega nas células 8, 13 ou 16. Se chegar em 8 ou 13 recebe um reforço negativo e chegando em 16 recebe um reforço positivo. É como se fosse concedida uma gorjeta extra por chegar em 16 e uma multa por chegar em 8 ou 13. Quando uma das condições é satisfeita a tentativa acaba, chamamos isso de episódio. O agente então guarda em sua memória informações sobre o caminho que fez, por exemplo, se ele começou em 7 e foi para 8, ele sabe agora que o caminho 7→8 fornece uma recompensa negativa e tentará evitar esta ação (ir de 7 para 8). Todo esse processo é repetido diversas vezes (vários episódios) e o agente vai melhorando sua memória, ou seja, vai aprendendo sobre os melhores caminhos da cidade para chegar à célula 16. O procedimento é o seguinte:

- Sorteia-se uma posição de início aleatória para o taxista: se ele cair em um bloqueio "perde uma vida" e a tentativa acaba; caso contrário ele escolhe aleatoriamente uma das células vizinhas e muda de posição (estado); isso é repetido até encontrar a saída.

Quando muda de estado está aprendendo sobre a cidade e atualiza sua memória (Q) com a equação de aprendizagem Q. Exemplo: se ele foi para a célula 4, "olha" para as células vizinhas (3 e 8), procura nas suas experiências passadas qual das ações "ir de 4 para 3" e "ir de 4 para 8" retornou uma recompensa maior. Isso é representado pelo termo max(Q(s', a')). Se ele nunca executou as ações antes, esta quantidade é zero. A recompensa que o ambiente gera para ele está armazenada em R; são valores que nós definimos. Então o agente faz as contas, soma tudo à memória anterior e avança aleatoriamente para a célula 3 ou 8.

Em cada episódio de nosso experimento, o agente descobre o mundo ao seu redor. Para definir a recompensa a ganhar em cada ação, façamos o seguinte: queremos que o taxista entenda que

a célula 16 contém o destino final, então estar nesta e ali permanecer deve fornecer a maior recompensa, e ir para ela através das ações “12→16” e “15→16” deve fornecer recompensas um pouco menores. Representamos isso, na matriz R, na Figura 1.

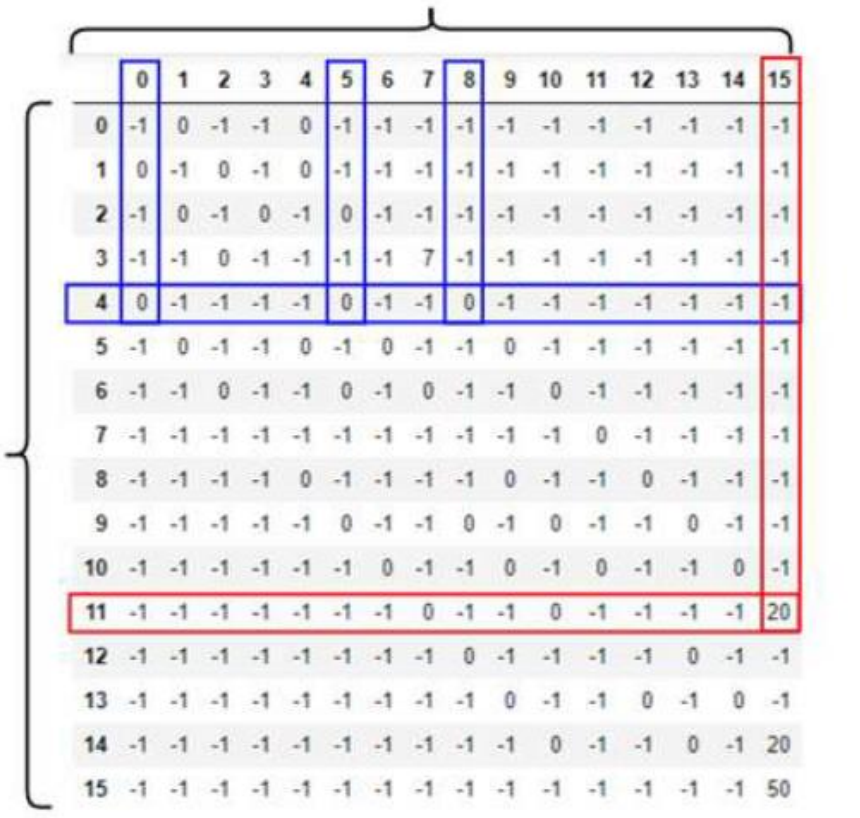
Na matriz R as colunas (zero = célula 1, …, 15 = célula 16) representam o estado e as linhas representam a ação. Exemplo: se o agente estiver na célula 12 (S = 12), e for para a célula 16 (a = 12→16), olhamos no destaque em vermelho que receberá uma recompensa igual a 20. Os valores -1 representam ações proibidas. Exemplo: no destaque em azul, a partir da célula 5, as únicas ações possíveis são a = {5→1, 5→ 6, 5→ 9}. Neste caso com valor zero, ou seja, nenhuma recompensa.

A memória do agente é parecida com essa matriz R. Ela precisa associar um valor alto para as ações boas e baixo para ações ruins. No primeiro episódio, a cidade é completamente nova para o taxista então não há memórias (a matriz contém apenas zeros). Depois de muito treino podemos ver como ele se sai tomando suas próprias decisões.

Ao final da implementação você deverá gerar um relatório e enviar descrevendo todas ações possível que o agente levou até chegar no objetivo final.

Você deverá entregar um arquivo.pdf com o relatório e um arquivo com o código da sua aplicação. A linguagem para implementação é de livre escolha, mas deixo como sugestão a linguagem Python (neste caso, pode enviar um arquivo.py). Não esqueça de descrever os pacotes necessários para executar seu código!

Figura 1. Matriz R.



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	-1	0	-1	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
1	0	-1	0	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
2	-1	0	-1	0	-1	0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
3	-1	-1	0	-1	-1	-1	-1	7	-1	-1	-1	-1	-1	-1	-1	-1
4	0	-1	-1	-1	-1	0	-1	-1	0	-1	-1	-1	-1	-1	-1	-1
5	-1	0	-1	-1	0	-1	0	-1	-1	0	-1	-1	-1	-1	-1	-1
6	-1	-1	0	-1	-1	0	-1	0	-1	-1	0	-1	-1	-1	-1	-1
7	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1
8	-1	-1	-1	-1	0	-1	-1	-1	-1	0	-1	-1	0	-1	-1	-1
9	-1	-1	-1	-1	-1	0	-1	-1	0	-1	0	-1	-1	0	-1	-1
10	-1	-1	-1	-1	-1	-1	0	-1	-1	0	-1	0	-1	-1	0	-1
11	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	0	-1	-1	-1	-1	20
12	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	-1	-1	0	-1	-1
13	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	0	-1	0	-1
14	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	-1	-1	0	-1	20
15	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	50

Fonte: Cabral (2018).

2. Código Fonte

O código fonte está disponível no Notebook “reinforcement_learning_modulo04.ipynb”.

3. Análise

Exemplo:

- A célula 11 foi sorteada como estado inicial;
- O taxista tem quatro opções de ação 11→7, 11→10, 11→15 e 11→ Sorteia a última opção e faz a conta

  Qnovo(s=11,a=12) = (1-1)*Q(s=11,a=12) + 0.5*(0 + 1*0) = 0

- A matriz Q continua com o valor zero para esta ação;
- Na ação seguinte ele vai ao acaso para a saída e recebe uma boa recompensa

  Qnovo(s=12,a=16) = (1-1)*Q(s=12,a=16) + 0.5*(20 + 1*0) = 10

- Agora orobô sabe que, se ele fizer a ação 12→16, receberá uma recompensa. A matriz de memória agora não tem apenas zeros, possui um valor 10 na posição que representa esta ação.
- Após vários episódios a memória estará repleta de números, como esta:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0	75	0	0	81	0	0	0	0	0	0	0	0	0	0	0
1	74	0	77	0	81	0	0	0	0	0	0	0	0	0	0	0
2	0	70	0	47	0	84	0	0	0	0	0	0	0	0	0	0
3	0	0	65	0	0	0	0	0	0	0	0	0	0	0	0	0
4	77	0	0	0	0	85	0	0	83	0	0	0	0	0	0	0
5	0	76	0	0	76	0	87	0	0	83	0	0	0	0	0	0
6	0	0	54	0	0	72	0	0	0	0	99	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
8	0	0	0	0	78	0	0	0	0	96	0	0	0	0	0	0
9	0	0	0	0	0	86	0	0	87	0	102	0	0	70	0	0
10	0	0	0	0	0	0	73	0	0	95	0	111	0	0	86	0
11	0	0	0	0	0	0	0	0	0	0	99	0	0	0	0	131
12	0	0	0	0	0	0	0	0	12	0	0	0	0	49	0	0
13	0	0	0	0	0	0	0	0	0	64	0	0	0	0	83	0
14	0	0	0	0	0	0	0	0	0	0	91	0	0	69	0	88
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	125

Verificando a memória vemos, por exemplo, que, se o agente estiver na célula 11 (linha 10 da tabela), o próximo movimento é escolhido com base no maior número (recompensa) nesta linha; que corresponde ao valor 111 e a ação é ir para a célula 12 (coluna 11 da tabela). Assim, ele se move para 12 e, de lá, a maior recompensa diz a ele para ir para o destino final. Como resultado de tudo isso, não importa onde o agente inicia sua jornada, ele sempre chega ao destino final através do menor caminho existente.