Inteligência Artificial 2019-1
Documentação Trabalho Prático II
Aprendizado por reforço

Ronald Davi Rodrigues Pereira
ronald.pereira@dcc.ufmg.br

Universidade Federal de Minas Gerais

25 de Junho de 2019

1 Introdução

Pac-man [1] é um jogo de *arcade* desenvolvido pela Namco em 1980. Esse jogo ficou muito famoso na época e ainda é um dos maiores clássicos de jogos *arcade* dos anos 80. O jogo é composto por um personagem controlável principal (o "Pac-Man"ou "Puckman", originalmente no Japão) em que seu objetivo é capturar todas as pastilhas contidas em um labirinto enquanto desvia e foge de vários outros personagens não controláveis (os fantasmas).

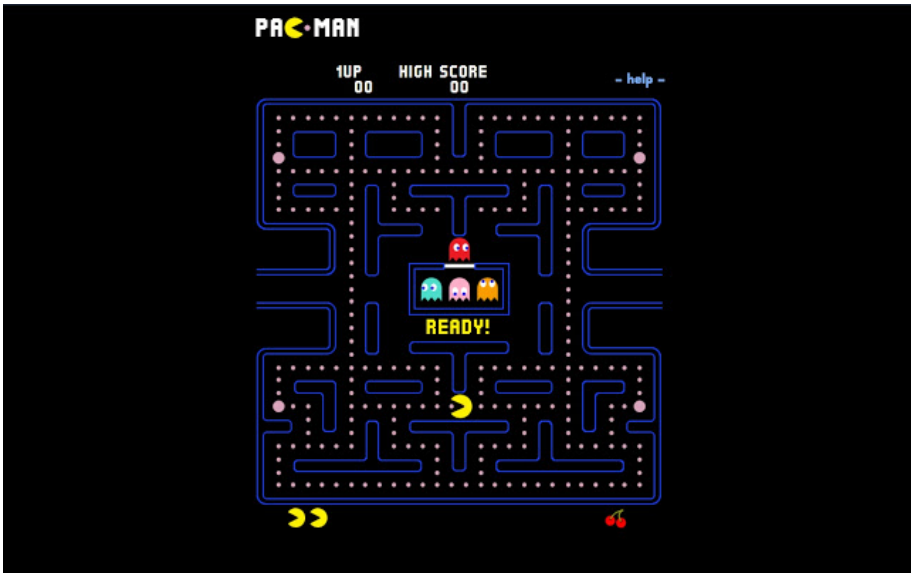

Figura 1 – Exemplo de uma tela do jogo Pac-man

O objetivo desse trabalho foi realizar a implementação de um algoritmo de aprendizado por reforço utilizando o Q-Learning [2] em um labirinto simplificado e estático

com fantasmas também estáticos. O cenário é um mundo bidimensional, representado por uma matriz de caracteres. A pastilha é representada por 0 (zero), um fantasma por & (e comercial), uma parede por # (cerquilha), e um espaço vazio por − (traço), como no exemplo abaixo

```
######
#---0#
#-#-&#
#----#
######
```

2 Modelagem e Implementação

A implementação foi realizada na linguagem Python 3.7.3, contendo um arquivo *pacmaze.py* que executa a leitura da entrada, montagem das estruturas utilizadas e execução do algoritmo do Q-Learning.

2.1 Estados

Cada estado (espaço vazio "−"no labirinto) é composto por um array de tamanho 4 x n x m, sendo n e m o número de linhas e colunas no labirinto, respectivamente. Essa decisão de implementação foi feita para representar os quatro valores possíveis de Q para cada estado possível de se movimentar (cima, baixo, esquerda e direita).

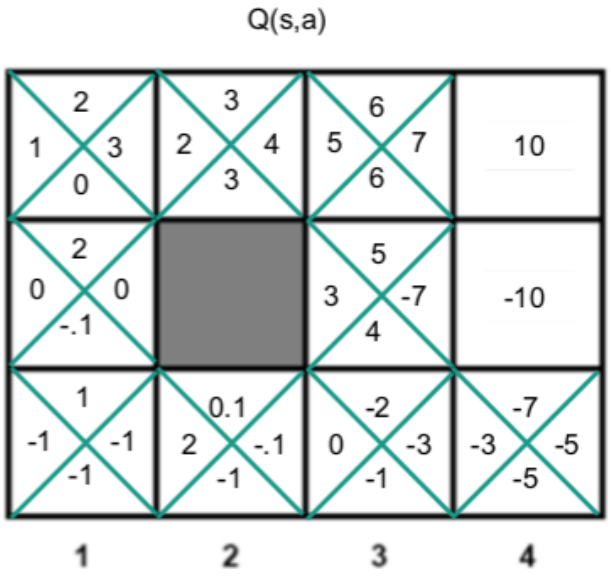

Figura 2 – Exemplo ilustrativo dos valores de Q(s,a) para uma determinada instância do Pacmaze

2.2 Q-Learning

A função de execução do aprendizado por reforço utilizando o Q-Learning se baseia em uma atualização dos valores de $Q(s, a)$, sendo s o estado atual do Pac-man e a a ação

que ele deve tomar. A função de atualização do valor de $Q(s, a)$ pode ser descrita como

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a' \in A} Q(s', a') - Q(s, a)] \quad (1)$$

e o algoritmo de atualização pode ser descrito a partir do pseudo-algoritmo:

Q(s,a) = 0 para todos os estados e ações
s ← estado inicial aleatório
repita:
    se aleatório() < ϵ: a ← ação_aleatória()
    se não: a ← argmax$_a Q(s, a)$
    Execute a ação a, observe o retorno r e o próximo estado s'
    Atualize Q(s,a) de acordo com a equação (1)
    s ← s'

3 Testes

Nessa seção, serão apresentados os resultados da política ótima encontrada para os arquivos de teste disponibilizados pelo professor, bem como os comandos que a executam.

3.1 pacmaze-01-tiny.txt

```
./qlearning.sh input/pacmaze-01-tiny.txt 0.3 0.9 100000
```

```
###############
#RRRRROLLLLLL#
#######U##U#U#
#RRRRRRU&RU#U#
#U##########&#
#ULLLLLLLLLLU#
###############
```

3.2 pacmaze-02-mid-sparse.txt

```
./qlearning.sh input/pacmaze-02-mid-sparse.txt 0.3 0.9 100000
```

```
####################
#DDRDLDDDDDDDDL#RDD#
#RRR&LR&LR&LLLL#RRD#
#UUUULUULRUULLU#U#D#
#U###############U#D#
#ULLLLLLLLLLRRRU#D#
##################D#
#DLDDLLRR&LLLDDDDDD#
```

```
#D&DD########RRR&LL#
#DLLLLLLLLLLRRRUUUU#
#D#################U#
#D#RRRDD#O#RRRRRRRU#
#RRUU#RRRU#RRUUUUUU#
####################
```

3.3 pacmaze-03-tricky.txt

```
./qlearning.sh input/pacmaze-03-tricky.txt 0.3 0.9 100000
```

```
####################
#DLLL&RRRDLLLL&RRRD#
#D&&&&&&&D&&&&&&&&D#
#DDDRRRD&D&DDDLLDDL#
#DDD&DRD&D&DDL&DDLL#
#RRRRRRD&D&DLLLLLLL#
#RRDDD&RRDLL&DDDLLL#
#RRRRRRU&D&ULLLLLLL#
#URU&UUU&D&UUL&UULL#
#RUURURU&O&UULLLUUU#
####################
```

4 Análise Experimental

Para uma análise experimental exploratória foram utilizados diferentes valores de parâmetros de entrada de $\alpha \in \{0.1, 0.5, 0.9\}$ (taxa de aprendizado) e de ϵ-greedy $\in \{0.1, 0.5, 0.9\}$ (fator de exploração) para um mesmo número de iterações do algoritmo ($N = 1000$), de modo a garantir a convergência do algoritmo para esses exemplos utilizados, repetidos por 100 vezes para cada exemplo.

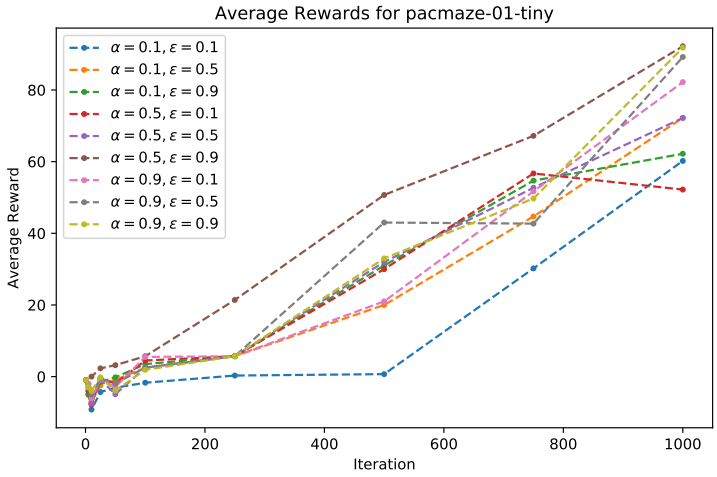

Figura 3 – Recompensas médias para pacmaze-01-tiny

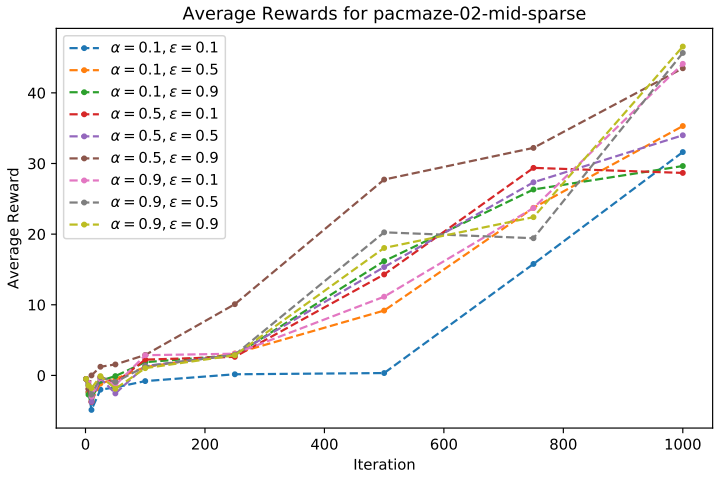

Figura 4 – Recompensas médias para pacmaze-02-mid-sparse

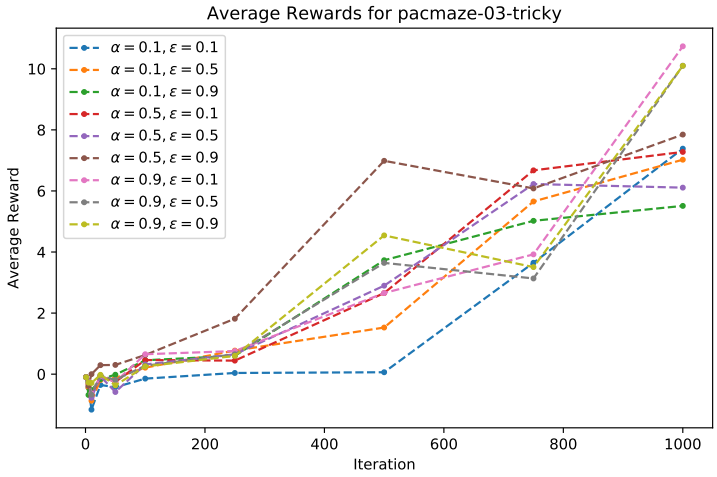

Figura 5 – Recompensas médias para pacmaze-03-tricky

De modo geral, podemos concluir que um baixo α (taxa de aprendizado) juntamente com um baixo ϵ-greedy (fator de exploração) não consegue explorar o mapa e, consequentemente, obteve as menores recompensas acumuladas médias de todas as variações de parâmetros. Uma outra conclusão possível é que um alto α gera uma maior variância das recompensas médias, justamente pelo fato de considerar mais a alteração do valor de $Q(s, a)$ pela parcela que ele a multiplica. A variação do ϵ-greedy não obteve uma expressividade muito grande nos testes realizados, de modo que o comportamento de suas variações não demonstrou um comportamento comum entre as diferentes execuções dos testes.

5 Dificuldades encontradas

A implementação foi muito facilitada pelo fato de eu ter estudado a matéria e compreendido todo o algoritmo antes de começar a criar o código. Desse modo, não obtive dificuldades significativas na implementação da prática proposta. A única dificuldade que eu enfrentei foi a decisão de implementação da representação da matriz dos Q-values, de modo que diferentes estruturas de dados levariam a uma facilidade maior de visualização. Foi tomada a decisão de representar essa matriz como 4 camadas, sendo uma pra cada ação possível. No mais, não foi encontrada mais nenhuma dificuldade e o algoritmo foi implementado em sua totalidade com sucesso.

6 Referências Bibliográficas

[1] WIKIPEDIA. Pac-man. https://en.wikipedia.org/wiki/Pac-Man. Acessado em 25 de Junho de 2019.

[2] WIKIPEDIA. Q-learning. https://en.wikipedia.org/wiki/Q-learning. Acessado em 25 de Junho de 2019.