Computação Natural - TP4

Jéssica Lemos (a82061)

1 Implementação e resultados

À semelhança da implementação deste trabalho realizada inicialmente e também a resolução do exercício proposta foram implementados os dois algoritmos, SARSA e Q-learning, sendo que foi necessário fazer alterações ao código fornecido. Nesta adaptação optei por manter uma penalização no movimento, cujo valor é 0.1. Assim sendo tornou-se necessário efetuar as alterações causadas pela mudança do mapa que se refletiu nas três matrizes, *rewards*, *actions* e *states*. Estas alterações refletem-se devido à alteração das dimensões do mapa bem como à existencia de uma parede no estado 5 pelo que não é possível efetuar-se movimento para este. Para além disso, o estado 3 é no qual existe uma recompensa de +1 e no 7 uma penalização de -1. Por fim podemos visualizar os dados através de gráficos como o apresentado na Figura 1.

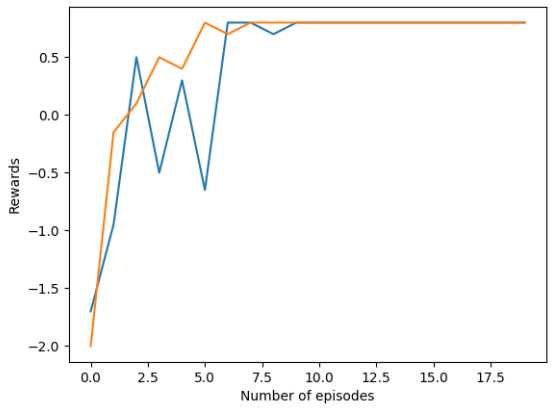

Figura 1: Episódios vs recompensas

Para este caso, consideramos os valores apresentados de seguida, sendo que tendo um número máximo de episódios 20, conseguimos constatar que o SARSA converge com 10 episódios enquando o Q-learning com 8.

```
MAX_N_EPISODES = 20
LEARNING_RATE = 0.7
GAMMA_DISCOUNT = 0.7
EPSILON = 0.3
EPSILON_DEGRADATION = 0.03
```

Agora alterando o número máximo de episódios para mais do dobro, ou seja 50, constatamos que o SARSA convergiu mais rapidamente, em 7 episódios mantendo o mesmo comportamento o Q-learning. Sendo que esta alteração não é muito significante tendo em conta a aleatoriedade do mesmo.

Assim sendo, mantendo o número inicial de episódios (20) passamos agora a verificar as alterações decorrentes da alteração do *learning rate* para um valor consideravelmente mais baixo, como

por exemplo 0.2. Sendo que como podemos constatar pela Figura 2 o número de episódios necessários para convergir em ambos os algoritmos aumentou, 14 episódios no caso do SARSA e 12 no Q-learning, o que já seria expectável.

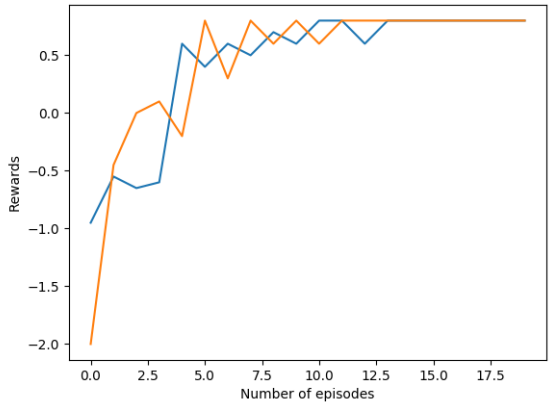

Figura 2: Episódios vs recompensas com *learning rate = 0.2*

O contrário acontece quando o valor é muito elevado como 0.9, sendo que ambos convergem em menos episódios. No exemplo apresentado na Figura 3, em 5 episódios no SARSA e 7 no Q-learning.

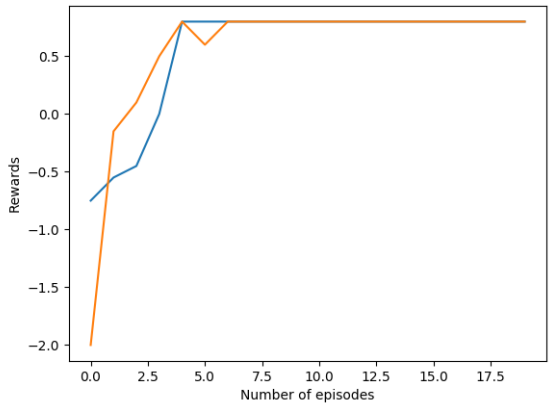

Figura 3: Episódios vs recompensas com *learning rate = 0.9*

Passamos agora a verificar as alterações causadas pelo valor de *gamma discount*, mantendo o valor de *learning rate* tendo em conta que não é uma valor muito elevado e tem comportamento semelhante a valores intermédios. Com um valor menor, como 0.2 não verificamos grande modificação no comportamente, convergindo em 8 episódios no SARSA e 7 no Q-learning. Semelhante para um valor de 0.9, em que o SARSA converge em 6 episódios e 8 no Q-learning.

Quanto ao *epsilon* nas condições iniciais, aumentando o seu valor o número de episódios necessários para convergir o SARSA também aumenta, sendo que com um valor de 0.5 obtemos o gráfico da Figura 4, em que o SARSA converge em 12 episódios e o Q-learning em 8.

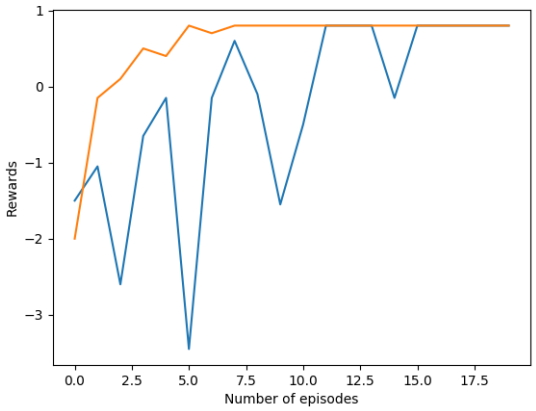

Figura 4: Episódios vs recompensas com *epsilon = 0.5*

Por fim, quanto ao *epsilon degradation* a alteração deste valor não tem impacto significativo dado que mesmo com um valor muito superior como 0.9, o número de episódios necessários são 8, tanto para o SARSA como o Q-learning.

2 Dificuldades

As dificuldades passaram pela fase inicial da implementação dos dois algoritmos, que foi demorada e com o aparecimento de vários problemas que tiveram por base a implementação dos vários valores como o *epsilon* e *gamma*. Contudo, a adaptação do código fornecido para a resolução deste problema foi uma etapa simples sendo mais demorado retirar conclusões sobre a alteração dos valores pedidos.