

CEFET/RJ - Maracanã
Bacharelado em Ciência da Computação
Inteligência Artificial
Trabalho 3: Aprendizado por Reforço
Prof. Eduardo Bezerra

Realizamos em sala de aula um tutorial de utilização da ferramenta Open AI Gym (<https://www.learn datasci.com/tutorials/reinforcement-q-learning-scratch-python-openai-gym/>). Neste trabalho, você irá dar continuidade às atividades deste tutorial, ainda no contexto do ambiente “Taxi” utilizado no tutorial.

Informações Preliminares

As três primeiras partes desse trabalho solicitam que você implemente variações do algoritmo Q-learning, tomando como ponto de partida a implementação básica fornecida no tutorial. Você deve implementar cada uma dessas variações em um classe separada (cujo nome é indicado nas partes abaixo). Além disso, cada uma dessas classes deve conter os seguintes métodos:

- construtor para permitir configurar as informações necessárias para o treinamento (e.g., valores de alpha, epsilon e gamma, no caso do Q-learning básico).
- método `treinar(epochs)`. (sem parâmetros) para realizar o treinamento do agente. Quando for invocado, esse método deve receber a quantidade de episódios por meio do parâmetro `epochs`. O valor desse parâmetro deve ser usado no lugar da quantidade fixa de episódios, que atualmente está definida como igual a 100000 (cem mil) no código fornecido no tutorial.
- método `evaluate(epochs)`. Esse método tem o propósito de avaliar o modelo que foi treinado previamente com o método `treinar`. O método `evaluate` deve encapsular a lógica fornecida no tutorial na seção “*Evaluating the agent*”. Quando for invocado, esse método deve receber a quantidade de episódios por meio do parâmetro `epochs`. Ao final de sua execução, esse método deve produzir a quantidade

média de passos por episódio, assim como a quantidade média de penalidades por episódio.

Parte 1 - QLearningBasico

Transforme o código fornecido no tutorial em um módulo de funções em Python que pode usar vários ambientes. Em particular, você deve encapsular o código do algoritmo Q-learning básico fornecido no tutorial (na seção “*Training the Agent*”) em uma classe denominada `QLearningBasico`.

Parte 2 - QLearningComFuncaoExploracao

Vimos em sala de aula duas abordagens para fazer com que um agente equilibre os aspectos de exploração (*exploration*) e de aproveitamento (*exploitation*) do ambiente: epsilon-greedy e funções de exploração. O tutorial apresenta a implementação da abordagem epsilon-greedy. Nesta parte do trabalho, você deve implementar uma variação do algoritmo Q-learning fornecido no tutorial para utilizar uma função de exploração, em vez da abordagem epsilon-greedy. Para isso, crie uma nova classe `QLearningComFuncaoExploracao` para conter a nova versão do algoritmo Q-learning. Descreva no seu relatório a sua função de exploração.

Parte 3 - QLearningComDecaimento

O algoritmo Q-learning possui diversos hiperparâmetros: alfa (taxa de aprendizado), gama (fator de desconto) e épsilon (probabilidade de o agente explorar uma ação do ambiente). Na implementação do Q-learning fornecida no tutorial, os valores desses hiperparâmetros são fixados inicialmente e não mudam durante o treinamento. Nesta parte do trabalho, você deve implementar uma variação do Q-learning que seja capaz de ajustar os valores dos hiperparâmetros alfa, gama e épsilon ao longo dos episódios de treinamento. O texto do tutorial fornece dicas acerca de como esses ajustes podem ser feitos (veja a seção “Hyperparameters and optimizations”). Tome como ponto de partida a implementação que você fez para da classe `QLearningBasico`. Defina essa versão do algoritmo em uma classe denominada

`QLearningComDecaimento`. Essa classe deve também conter os métodos `treinar` e `avaliar`.

Parte 4 – Experimentos de comparação

Nesta parte do trabalho, você deve comparar as diferentes variações do Q-learning que você implementou nas partes anteriores. Sua comparação deve gerar uma tabela similar à fornecida no tutorial (na seção “*Comparing our Q-learning agent to no Reinforcement Learning*”). Contudo, em vez de comparar o Q-learning com um agente aleatório, você deve apresentar os resultados de execução de suas três variações do algoritmo Q-learning, a saber;

- `QLearningBasico`
- `QLearningComFuncaoExploracao`
- `QLearningComDecaimento`

Apresente em sua tabela métricas computadas ao longo de 100 episódios, para cada uma das três variações.

O que deve ser entregue

Você deve produzir um relatório (em formato PDF) descrevendo de que forma implementou cada uma das variações solicitadas do Q-learning, assim como o quadro comparativo com os resultados (solicitados na parte 4). Entregue também os seguintes arquivos:

- `q_learning_basico.py` (`QLearningBasico`)
- `q_learning_com_func_exploracao.py` (`QLearningComFuncaoExploracao`)
- `q_learning_com_decaimento.py` (`QLearningComDecaimento`)
- `parte4.py`: script que produz o que é solicitado na parte 4 deste trabalho.

Você deve compactar todos os arquivos do trabalho em um arquivo denominado `SEU_NOME_COMPLETO.zip`. Esse deve ser o arquivo a ser submetido pelo Moodle.