Deep Q-Learning

Samara Ribeiro Silva
Instituto Tecnológico de Aeronáutica, Laboratório de Inteligência Artificial para Robótica Móvel (CT-213). Professor Marcus Ricardo Omena de Albuquerque Máximo, São José dos Campos, São Paulo, 12 de julho de 2021.

Para implementar a rede neural utilizou-se as camadas conforme a tabela 1 e como pode-se observar no summary da figura 1.

Layer	**Neurons**	**Activation Function**
Dense	24	ReLU
Dense	24	ReLU
Dense	`action_size`	Linear

Tabela 1: arquitetura da rede neural usada para aproximar a função ação-valor $\hat{q}(s, a)$.

A escolha da ação do ε-greedy foi implementada de maneira semelhante ao laboratório anterior. Gerou-se um número aleatório entre 0 e 1 e caso esse número fosse menor que o ε, então o valor retornado é um número entre 0 e o tamanho do vetor de ação (*action_size*) caso contrário retorna-se o índice do maior valor do *array model.predict(state)[0]*.

Foi implementada também uma função de recompensa intermediária. As seguintes fórmulas foram utilizadas como base:

$$r_{modified} = r_{original} + (position - start)^2 + velocity^2$$

$$r'_{modified} = r_{modified} + 50 * 1\{next\_position \geq 0.5\}$$

Figura 1: Summary do modelo implementado em Keras

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 24)                72
_________________________________________________________________
dense_1 (Dense)              (None, 24)                600
_________________________________________________________________
dense_2 (Dense)              (None, 3)                 75
=================================================================
Total params: 747
Trainable params: 747
Non-trainable params: 0
_________________________________________________________________
Loading weights from previous learning session.
```

Figura 2: Média do *score* para o *evaluate*.

```
episode: 1/30, time: 163, score: 42.9904, epsilon: 0.0
episode: 2/30, time: 180, score: 37.0908, epsilon: 0.0
episode: 3/30, time: 94, score: 42.191, epsilon: 0.0
episode: 4/30, time: 157, score: 36.9984, epsilon: 0.0
episode: 5/30, time: 197, score: 37.1213, epsilon: 0.0
episode: 6/30, time: 153, score: 40.2426, epsilon: 0.0
episode: 7/30, time: 156, score: 36.9792, epsilon: 0.0
episode: 8/30, time: 153, score: 40.2121, epsilon: 0.0
episode: 9/30, time: 107, score: 44.3052, epsilon: 0.0
episode: 10/30, time: 153, score: 40.2226, epsilon: 0.0
episode: 11/30, time: 186, score: 37.1031, epsilon: 0.0
episode: 12/30, time: 86, score: 40.5513, epsilon: 0.0
episode: 13/30, time: 175, score: 37.0733, epsilon: 0.0
episode: 14/30, time: 99, score: 43.0924, epsilon: 0.0
episode: 15/30, time: 85, score: 40.2758, epsilon: 0.0
episode: 16/30, time: 85, score: 40.0656, epsilon: 0.0
episode: 17/30, time: 200, score: -6.40895, epsilon: 0.0
episode: 18/30, time: 172, score: 37.0587, epsilon: 0.0
episode: 19/30, time: 149, score: 39.0164, epsilon: 0.0
episode: 20/30, time: 186, score: 37.1042, epsilon: 0.0
episode: 21/30, time: 153, score: 40.2448, epsilon: 0.0
episode: 22/30, time: 159, score: 42.125, epsilon: 0.0
episode: 23/30, time: 148, score: 38.8463, epsilon: 0.0
episode: 24/30, time: 155, score: 41.0143, epsilon: 0.0
episode: 25/30, time: 200, score: -15.2897, epsilon: 0.0
episode: 26/30, time: 157, score: 36.9858, epsilon: 0.0
episode: 27/30, time: 85, score: 40.2287, epsilon: 0.0
episode: 28/30, time: 153, score: 40.242, epsilon: 0.0
episode: 29/30, time: 200, score: -12.4389, epsilon: 0.0
episode: 30/30, time: 149, score: 39.0187, epsilon: 0.0
Mean return:  34.475412573494395
```

Nas figuras 2 pode-se encontrar a média dos *scores* recebidos durante o *evaluate*. Pode-se observar pelo gráfico da figura 3 que o treinamento foi bem sucedido, obtendo sucesso antes do 100 episódio conforme requisito. Já na figura 4, é possível concluir que o resultado obtido foi satisfatório obtendo uma taxa de sucesso de 90 %.

Figura 3: Recompensa acumulada para o treinamento.

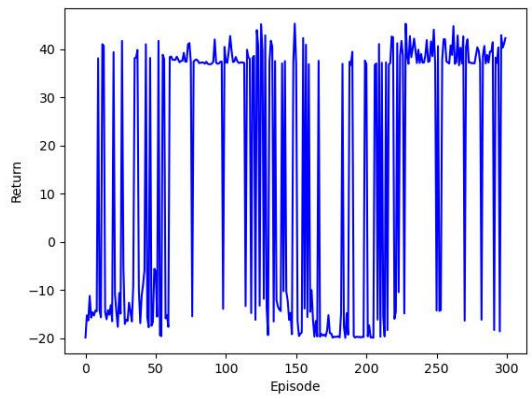


Figura 4: Recompensa acumulada para o *evaluate*.

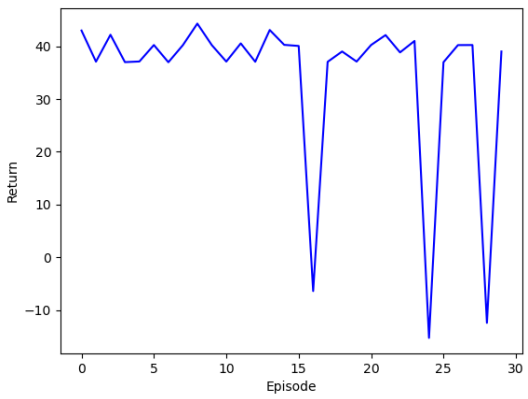


Figura 5: Representação em cores da tabela de greedy-policy calculada.

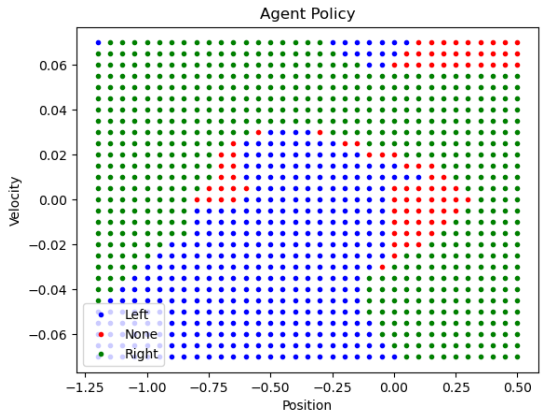
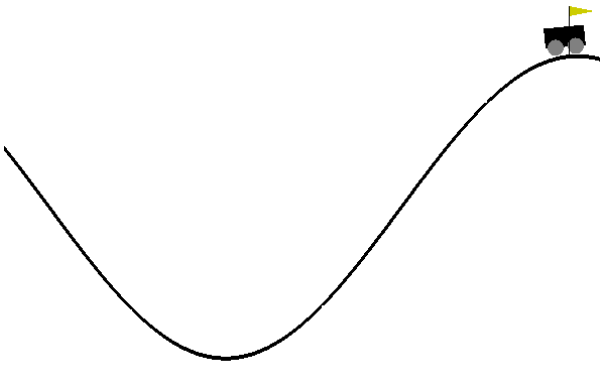


Figura 6: Trajetória do *Mountain Car* de linha após aprendizado com Q-Learning



Na figura 5, é possível observar a decisão tomada pelo *Mountain Car* de acordo com a velocidade e posição. Pode-se observar que há uma tendência de ir para a direita com velocidade acima de zero ou para posição maior que zero.