

LunarLander-DQN

Apresentação do projeto de
aprendizado de reforço



Introdução ao Projeto

O projeto utiliza o ambiente Gymnasium para treinar um agente no jogo LunarLander-v3.

O modelo de aprendizado escolhido é o DQN (Deep Q-Network), que é baseado em redes neurais para aprendizado por reforço.

O ambiente é criado na linha com `env = gym.make("LunarLander-v3", render_mode="rgb_array")`



Configuração do Ambiente

Para garantir que todas as dependências do projeto sejam mantidas organizadas e não interfiram em outras aplicações do sistema, foi criado um ambiente virtual. Isso é essencial para evitar conflitos entre versões de bibliotecas e garantir a reproduzibilidade dos experimentos.

```
python3 -m venv ambiente  
source ambiente/bin/activate
```



Parâmetros de Treinamento

total_timesteps: Define o número **total de interações do agente com o ambiente**. Quanto maior esse valor, maior a quantidade de dados utilizados para aprendizado.

learning_rate: Controla a **taxa de aprendizado da rede neural**. Um valor muito alto pode fazer o modelo aprender rapidamente, mas de forma instável. Um valor muito baixo pode tornar o aprendizado lento.

gamma: Representa o **fator de desconto para recompensas futuras**. Valores próximos de 1 incentivam o agente a considerar recompensas a longo prazo.

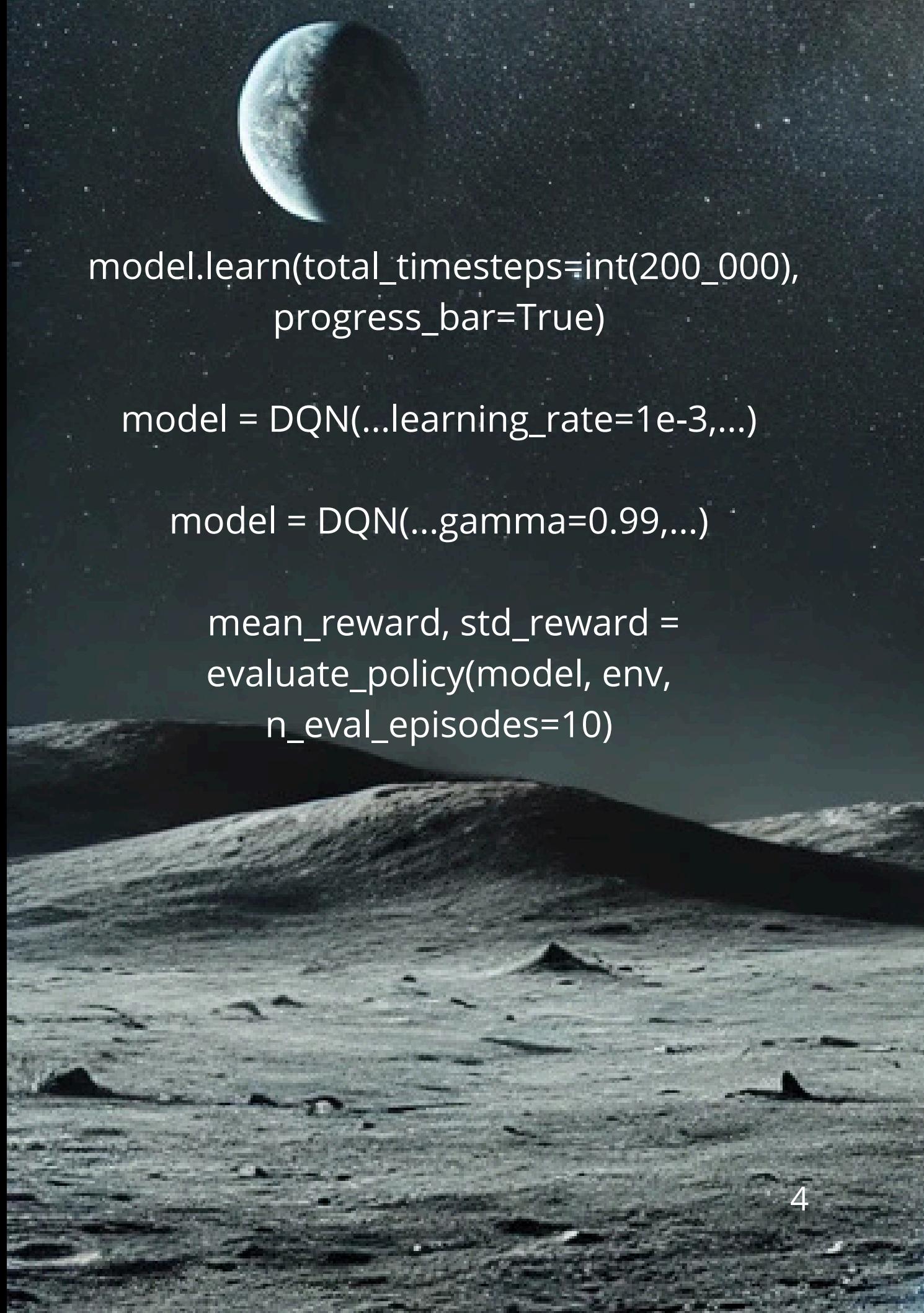
n_eval_episodes: Define o **número de episódios usados para avaliar o modelo após o treinamento**.

```
model.learn(total_timesteps=int(200_000),  
           progress_bar=True)
```

```
model = DQN(...learning_rate=1e-3,...)
```

```
model = DQN(...gamma=0.99,...)
```

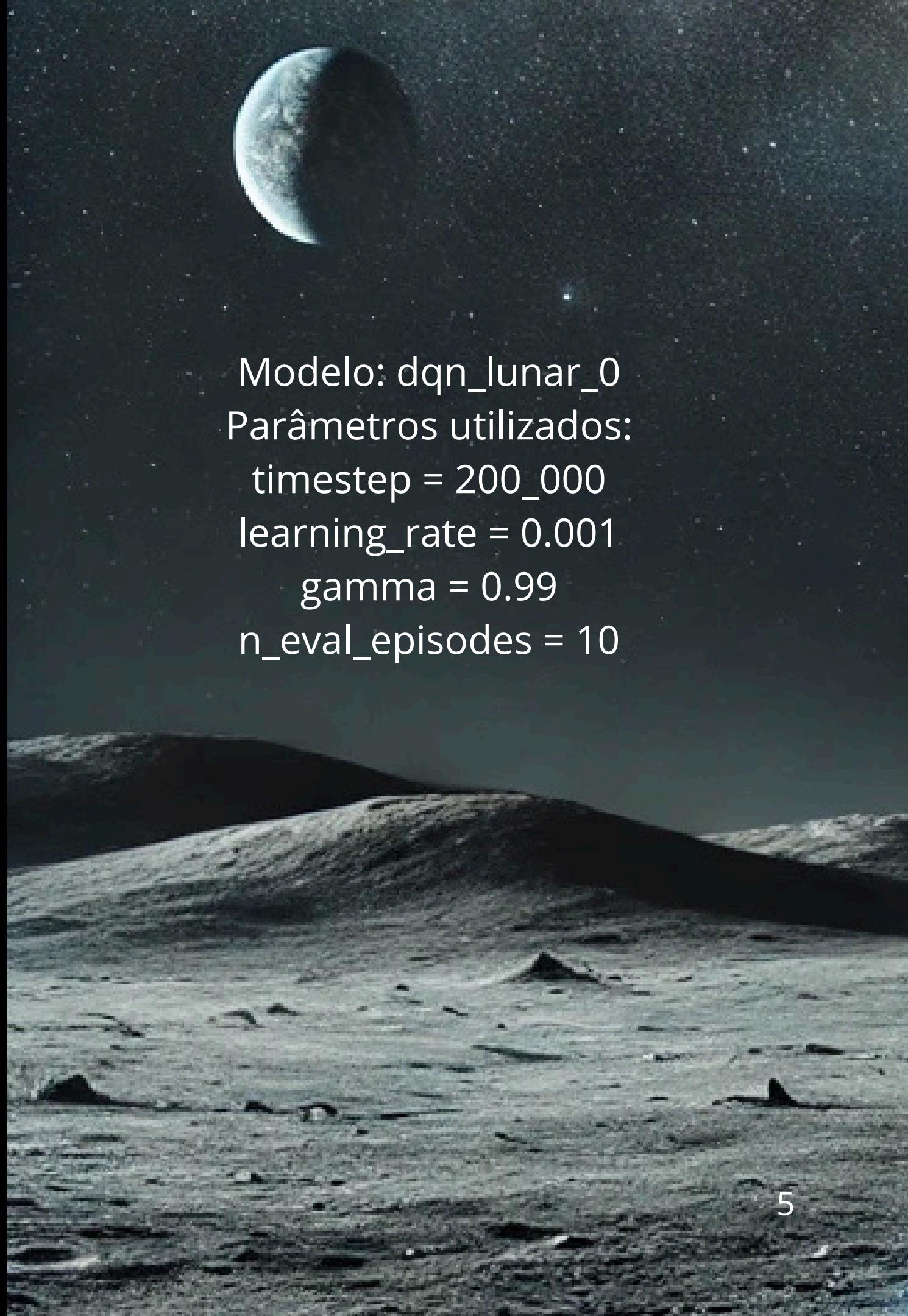
```
mean_reward, std_reward =  
evaluate_policy(model, env,  
n_eval_episodes=10)
```



Resultados do Primeiro Teste

O primeiro teste foi realizado com um conjunto padrão de parâmetros e um tempo de treinamento moderado. O objetivo era verificar se o agente conseguia aprender a pousar o módulo lunar de forma eficiente.

A **média** é **-128.26** e o desvio **padrão** é **73.87**. Isso indica que, na maioria das observações, os dados estão distribuídos em torno de -128.26, mas podem variar significativamente (cerca de 73.87 unidades para mais ou para menos). A **grande dispersão** sugere que há uma variação considerável nos dados.



Modelo: dqn_lunar_0
Parâmetros utilizados:
timestep = 200_000
learning_rate = 0.001
gamma = 0.99
n_eval_episodes = 10

Resultados do Segundo Teste

No segundo teste, foram feitas algumas modificações nos parâmetros do modelo para tentar melhorar o desempenho do agente. O tempo de treinamento foi aumentado e a taxa de aprendizado foi ajustada.

A **média** é **-364.51** e o **desvio padrão** é **50.96**. Neste caso, a média é mais baixa do que no primeiro conjunto de dados, e a variação é menor, o que sugere que os dados estão mais concentrados em torno da média de -364.51.

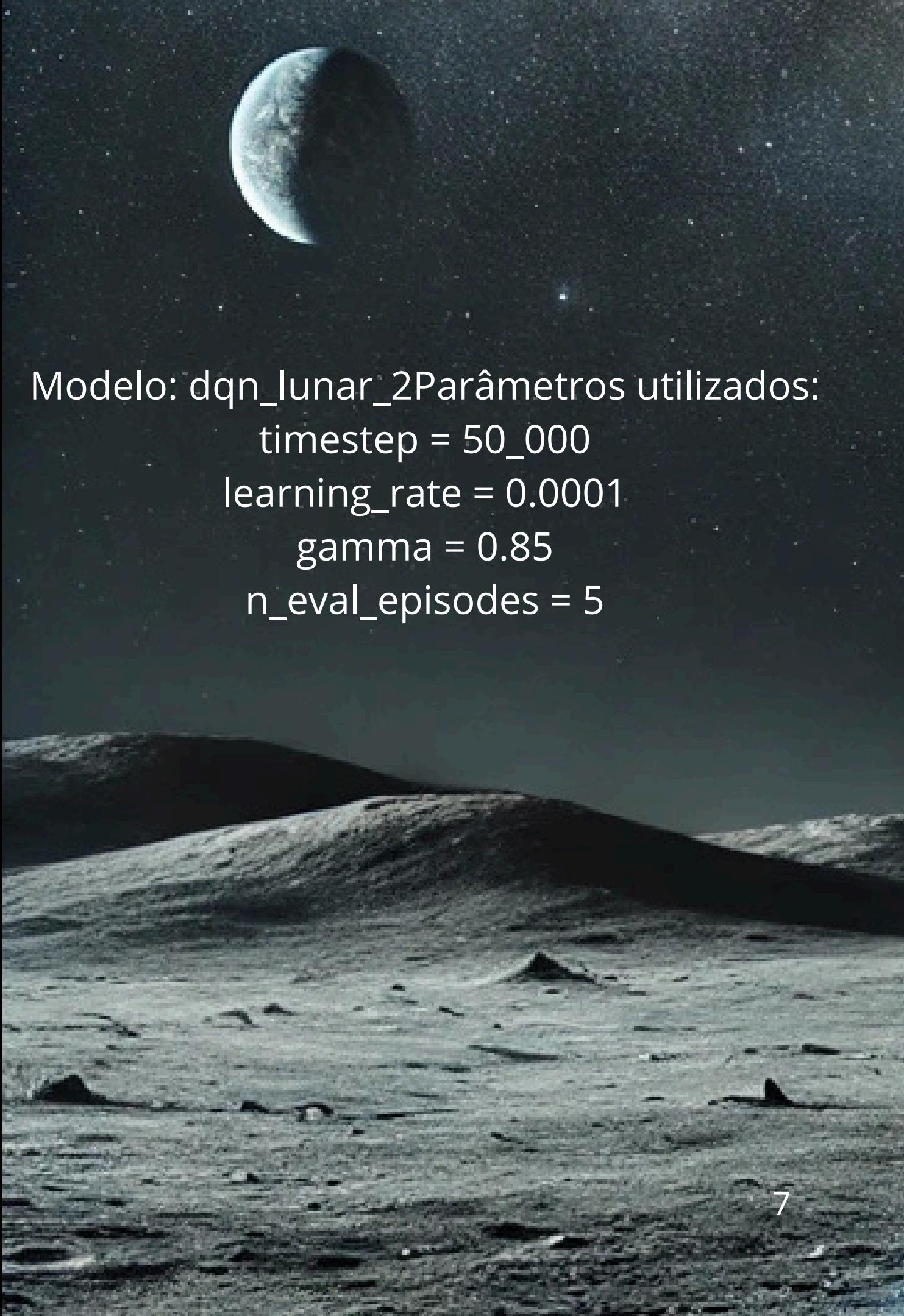


Modelo: dqn_lunar_1
Parâmetros utilizados:
timestep = 500_000
learning_rate = 0.005
gamma = 0.999
n_eval_episodes = 50

Resultados do Terceiro Teste

No terceiro teste, optou-se por um **treinamento mais curto** e um ajuste mais conservador dos parâmetros para verificar se um menor tempo de treinamento poderia trazer resultados positivos.

A **média** é **-91.90** e o **desvio padrão** é **29.75**. Esse valor possui uma **média maior** do que o primeiro e o segundo conjuntos, com a **menor dispersão** entre os três.



Modelo: dqn_lunar_2
Parâmetros utilizados:
timestep = 50_000
learning_rate = 0.0001
gamma = 0.85
n_eval_episodes = 5

Visualização do Modelo

Para analisar o desempenho do agente de forma visual, foi carregado o modelo treinado e executado no ambiente. Permitindo observar as **decisões tomadas pelo agente em tempo real**. O modelo carrega a rede neural salva e executa as ações no ambiente de forma autônoma.

O ambiente "LunarLander-v3" foi carregado através da função `gym.make()`, com a opção **render_mode="human"** ativada, permitindo a **renderização gráfica** das ações do agente durante a execução.



Conclusão

O primeiro resultado, com média de -128.26 e desvio padrão de 73.87, demonstrou ser o melhor desempenho, pois permitiu que a nave tivesse mais estabilidade e realizasse um pouso correto. Apesar da variação relativamente alta, a média se manteve dentro de um intervalo que favoreceu um controle mais eficiente durante a descida. Os outros resultados apresentaram maiores dificuldades, seja pela média muito baixa, como no segundo caso, indicando um desempenho pior, ou pela menor dispersão do terceiro caso, que, embora mais consistente, não proporcionou a mesma precisão no pouso.

