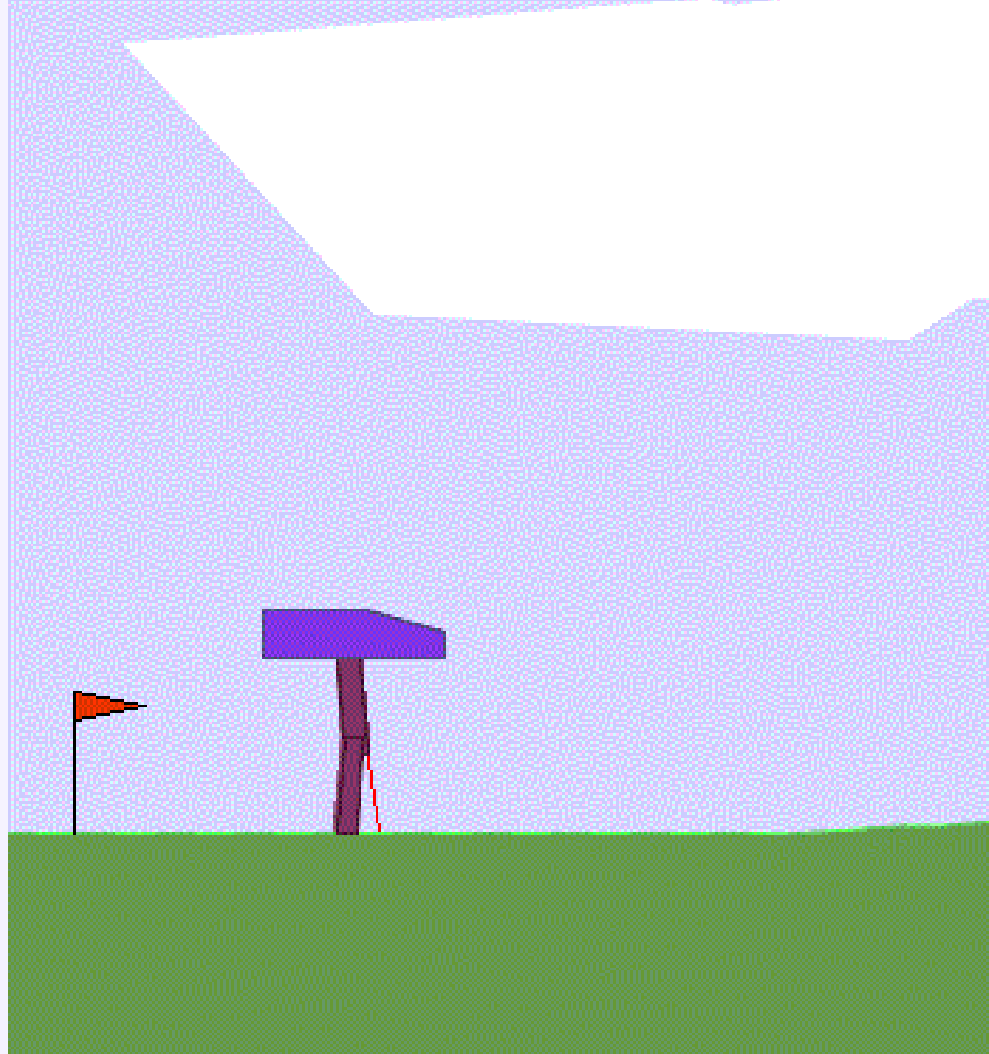


Aprendizado por Reforço em Simuladores de Física Bidimensionais

INF5021 - Matheus Madeira

Introdução

- Simulação física é uma área de extrema relevância para a computação, porém são problemas de grande complexidade.
- Esse trabalho busca servir como mais uma referência de como podemos usar inteligência artificial para resolver essa classe de problemas.
- Agente de aprendizado de reforço para um ambiente de simulação física em 2D.
 - Python, Gymnasium



Objetivos

Objetivo Geral

Desenvolver um agente por aprendizado por reforço para o ambiente Box2D Bipedal Walker.

Objetivos Específicos

- Implementar um agente de aprendizado por reforço
- Comparar o agente implementado com um agente aleatório do ambiente

Trabalhos Relacionados

- Tiago Reck Gambim. *Aprendizado por Reforço em Jogos de Estratégia*

Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) - PUCRS, 2021.

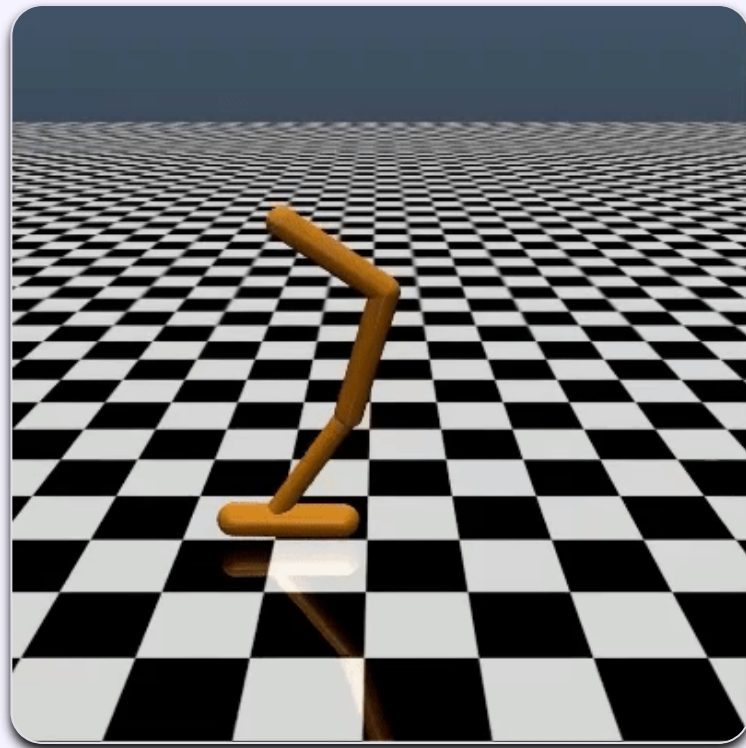
NOTA: ADICIONAR MAIS 2 TRABALHOS RELACIONADOS



Fundamentação Teórica

Aprendizado por reforço

- Aprender através da interação buscando atingir um objetivo.
- Interações entre Agente (tomador de decisões) e o Ambiente.
- Ambiente tem um espaço de ações, e um espaço de observação (estado).
- On-policy, Off-policy
- Q-learning



Fundamentação Teórica

$$\underbrace{Q(s, a)}_{\text{Novo valor}} = \underbrace{Q(s, a)}_{\text{Valor antigo}} + \underbrace{\alpha}_{\text{Taxa de aprendizado}} \left[\underbrace{R(s, a)}_{\text{Recompensa}} + \underbrace{\gamma}_{\text{Taxa de desconto}} \underbrace{\max_{a'} Q'(s', a')}_{\text{Valor máximo esperado, dado o novo estado e todas suas possíveis ações}} - Q(s, a) \right]$$

Metodologia

Gymnasium

- Biblioteca contendo uma diversa coleção de diferentes ambientes para aprendizado por reforço.

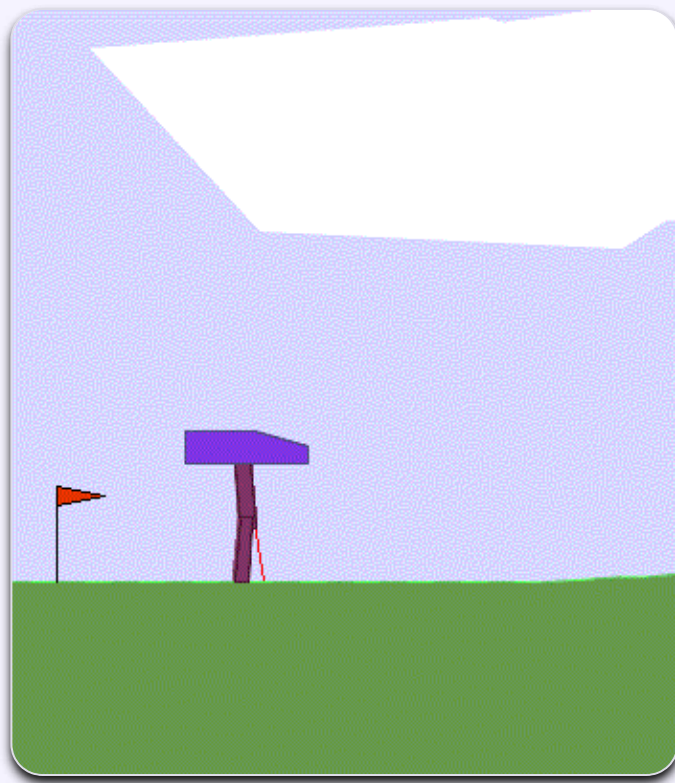
```
import gymnasium as gym
env = gym.make("LunarLander-v2", render_mode="human")
observation, info = env.reset(seed=42)
for _ in range(1000):
    action = env.action_space.sample() # this is where you would insert your policy
    observation, reward, terminated, truncated, info = env.step(action)

    if terminated or truncated:
        observation, info = env.reset()
env.close()
```

Metodologia

Bipedal Walker

- Ambiente com um robô com 4 juntas que caminha num terreno levemente desigual.
- Para resolução é necessário 300 pontos em 1600 passos de tempo.
- Recompensas são obtidas por se mover para frente, totalizando 300 pontos ao final.
 - Se o robô cair, -100 pontos.
 - Aplicar torque ao motor custa uma pequena quantia de pontos.



Metodologia

Bipedal Walker

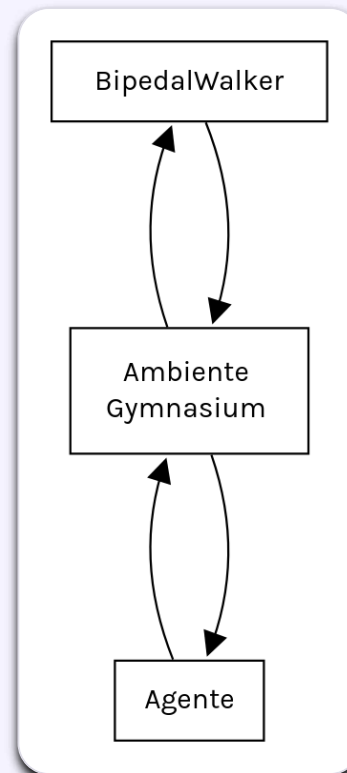
ESTADO CONSISTE DE (I) VELOCIDADE DO ÂNGULO DO CASCO, (II) VELOCIDADE ANGULAR, (III) VELOCIDADE HORIZONTAL, (IV) VELOCIDADE VERTICAL, (V) POSIÇÃO DAS JUNTAS E SUA VELOCIDADE ANGULAR, (VI) PERNAS ESTÃO EM CONTATO COM O CHÃO, E (VII) 10 MEDIÇÕES DE TELÊMETERO.

Espaço de ação	Box(-1.0, 1.0, (4.), float32)
Forma do espaço de observação	(24,)
Espaço de observação Máximo	[3.14, 5, 5, 5, 3.14, 5, 3.14, 5, 5, 3.14, 5, 3.14, 5, 5, 1, 1, 1, 1, 1, 1, 1, 1]
Espaço de observação Mínimo	[-3.14, -5, -5, -5, -3.14, -5, -3.14, -5, 0, -3.14, -5, -3.14, -5, -0, -1, -1, -1, -1, -1, -1, -1, -1]

Metodologia

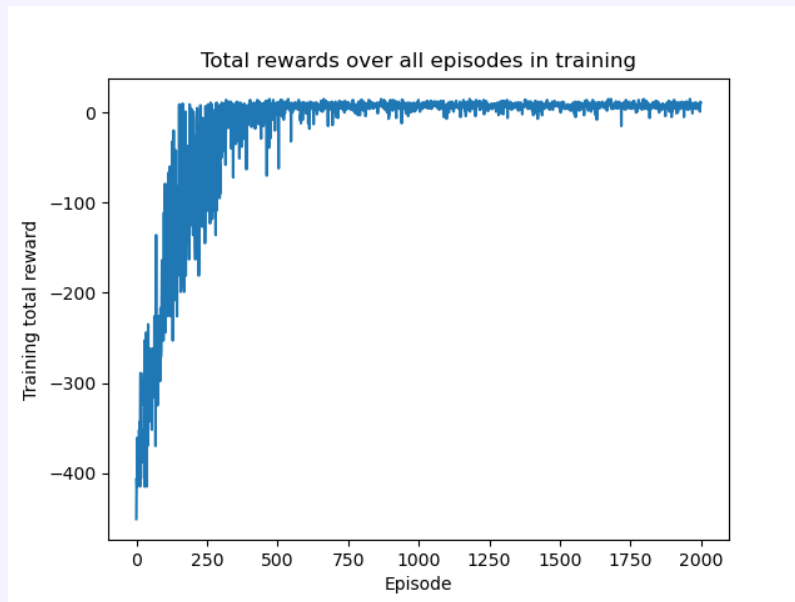
SOLUÇÃO

- Discretização do estado de observação do ambiente
- Definição das variáveis do ambiente necessárias para a solução (NOTA: adicionar assim que terminar)
- Divisão de cada uma dessas variáveis em x (NOTA, ainda em testes) partes iguais
- Etapas:
 - Implementação do Agente ($\alpha = X, \gamma = Y, \epsilon = Z$ (NOTA))
 - Treinamento
 - Comparação



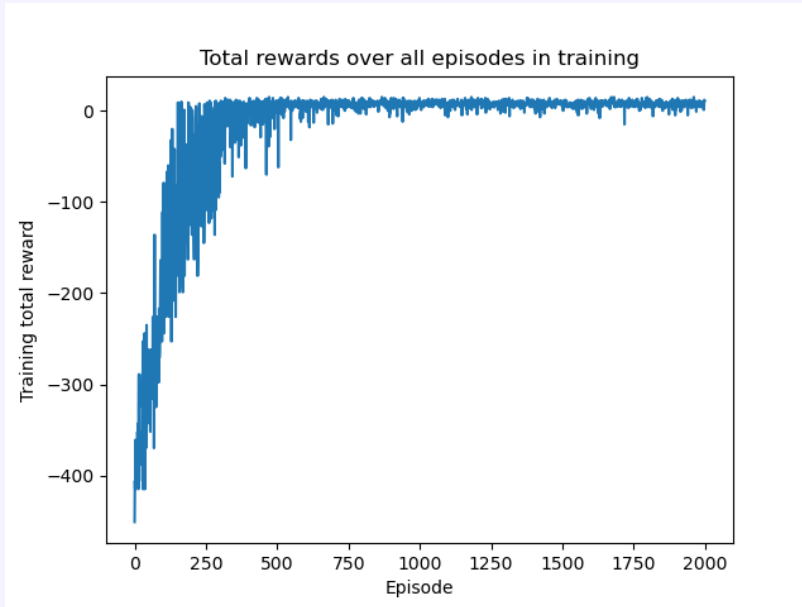
NOTA: SE HOUVER GRANDE MODIFICAÇÃO NA METODOLOGIA
DEVIDO A IMPLEMENTAÇÃO, ATUALIZAR NA FUNDAMENTAÇÃO

Resultados

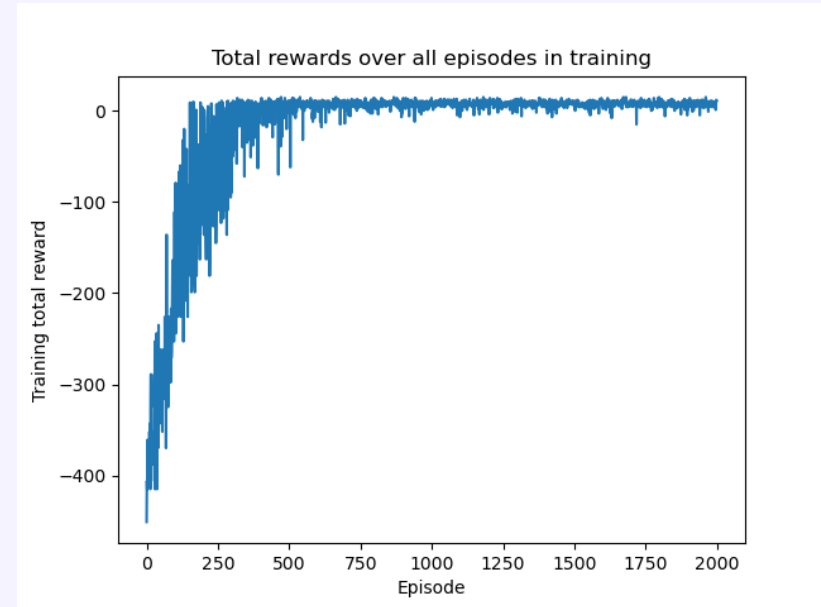


NOTA: Atualizar com resultado concreto médio (Epsilon + recompensa)

Resultados



NOTA: Atualizar com resultado concreto melhor resultado (Epsilon + recompensa)



NOTA: Atualizar com resultado concreto pior resultado (Epsilon + recompensa)

DEMO

Conclusão

- Foi possível obter um maior aprofundamento na área de IA, desenvolvendo habilidades práticas tanto com ferramentas bem utilizadas na área, quanto com a formulação de soluções.
- O resultado do trabalho consegue trazer uma contribuição como mais uma referência de solução para a classe de problemas escolhido.
- Melhorias futuras:
 - Mais implementações para comparação (DQN, PPO, etc)
 - Implementação para ambientes mais complexos

Referências

- Richard S. Sutton and Andrew G. Barto.
Reinforcement Learning: An Introduction.
The MIT Press, Cambridge, MA, 2018.
- Alexander Panin.
Introduction to Reinforcement Learning: On-policy vs off-policy, Nov, 2020.
- Gelana Tostaeva.
Introduction to Q-learning with OpenAI Gym, Abr, 2020.
- Costa Huang.
CleanRL (Clean Implementation of RL Algorithms), Abr, 2020.

Obrigado!

NOTA: UPAR APRESENTAÇÃO FINAL, TIRAR MENSAGEM DE "EM CONSTRUÇÃO DA RAÍZ"

LINK APRESENTAÇÃO: [MATHSMADEIRA.COM/UFRGS/INF5021/](https://mathsmadeira.com/UFRGS/INF5021/)

[GitHub Repo](#)