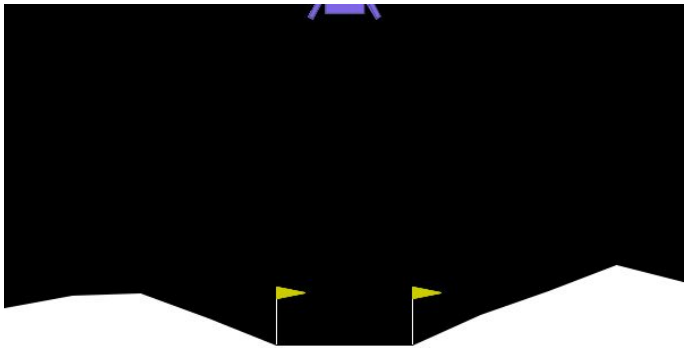
U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Francisco Ribeiro | up202104797
Marisa Azevedo | up202108624

Customizing OpenAI Gym Lunar Lander

Introdução aos Sistemas Inteligentes e Autónomos



Ambiente

Este ambiente simula um problema clássico de otimização da trajetória de um foguetão. O combustível é infinito, permitindo que o agente aprenda a voar e aterrar. O foguetão Lander é projetado com uma força inicial aleatória aplicada ao centro de massa. O objetivo é obter uma pontuação mínima de 200 pontos por episódio para considerá-lo uma solução.

Existem quatro **ações** discretas:

- Não fazer nada
- Ligar motor de orientação esquerdo
- Ligar motor de orientação direito
- Ligar motor principal

O **estado** é um vetor 8D que representa:

- As coordenadas da posição do foguetão
- As velocidades linear de eixos, x e y, e, também, a velocidade angular
- Ângulo
- Contato das pernas com o solo.

Ambiente

Para cada passo, a **recompensa**:

- aumenta/diminui quanto mais próximo/distante o lander está da plataforma de aterragem.
- aumenta/diminui quanto mais lento/rápido o lander está se movendo.
- diminui quanto mais inclinado (ângulo não horizontal) o lander estiver.
- aumenta em 10 pontos para cada perna em contato com o solo.
- diminui em 0,03 pontos a cada frame que o motor lateral está ligado.
- diminui em 0,3 pontos a cada frame que o motor principal está ligado.

O episódio recebe uma recompensa adicional de -100 ou +100 pontos para colisão ou aterragem segura, respetivamente.

Modificações

Recompensa 1

Quando terminado:

- Dentro do espaço de aterragem, valorizamos:
  - O agente por estar na zona de aterragem.
  - As pernas estarem em contato com o solo, e a velocidade e o ângulo de aterragem não serem demasiado bruscos.
  - O agente estar no centro da zona de aterragem.
- Fora do espaço de aterragem penalizamos, visto que não cumpriu o objetivo

Durante o episódio é feita uma pequena penalização, para certificar que a aterragem é feita o mais rápido possível.

Recompensa 2

Esta estrutura de recompensas é muito semelhante à *recompensa 1*. A única diferença é que não penalizamos o agente por estar no ar.

Recompensa 3

Esta estrutura de recompensa é muito idêntica à *recompensa 1*, a diferença entre elas é que quando o módulo de aterragem está no ar, o módulo de aterragem é menos penalizado do que a *recompensa 1*.

Algoritmos Escolhidos

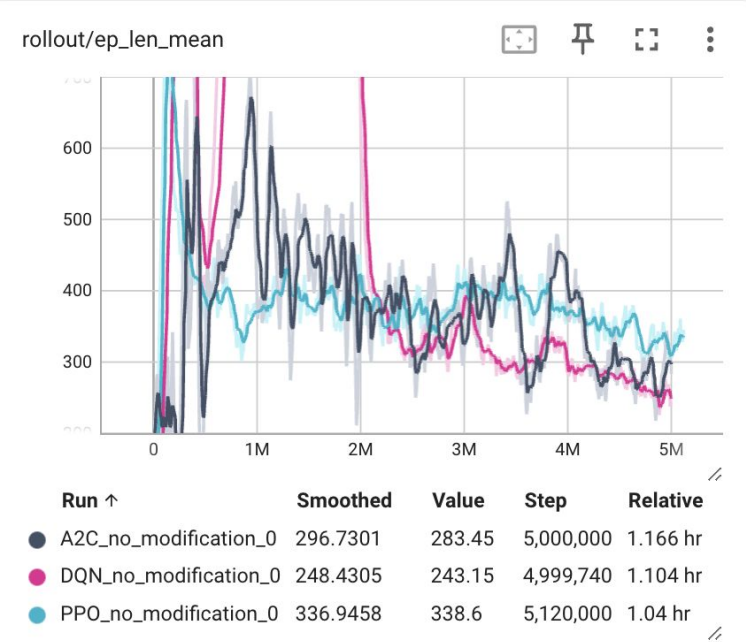
Decidimos experimentar três algoritmos diferentes: **PPO**, **A2C** e **DQN**

Para avaliar os algoritmos, utilizámos a duração de um episódio e a recompensa de um episódio. O gráfico de duração porque queremos que o módulo de aterragem aterre o mais rapidamente possível, e utilizámos o gráfico de recompensa para ver e analisar o estado e a estabilidade de um treino.

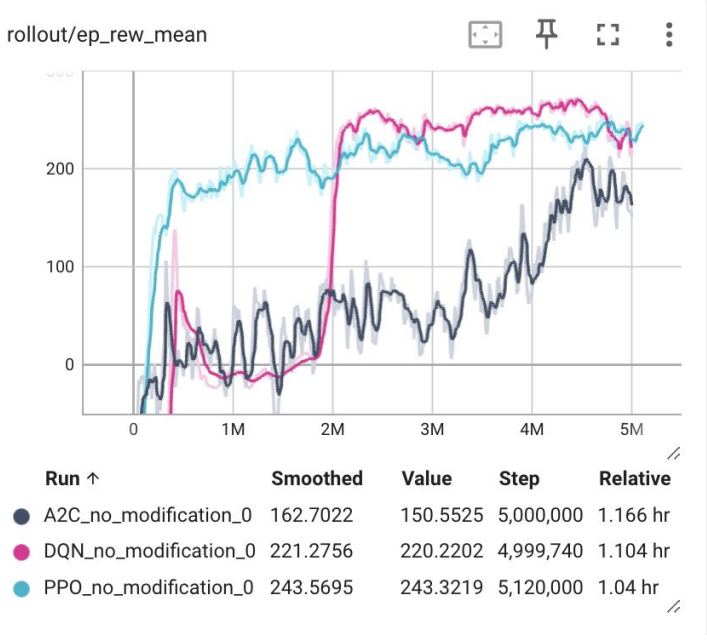
Resultados

Notamos que o algoritmo Proximal Policy Optimization (PPO) foi ligeiramente melhor.

Por isso, decidimos treinar o agente com ele, utilizando as três estruturas de recompensas diferentes criadas por nós (mencionadas acima) que achámos que fariam sentido.

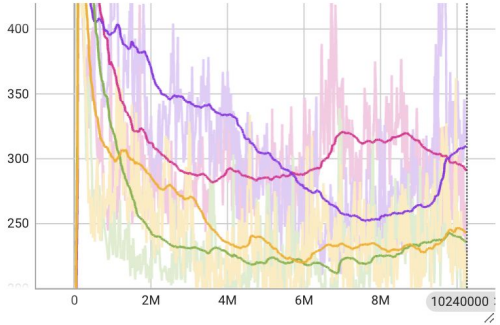


Run ↑	Smoothed	Value	Step	Relative
A2C_no_modification_0	296.7301	283.45	5,000,000	1.166 hr
DQN_no_modification_0	248.4305	243.15	4,999,740	1.104 hr
PPO_no_modification_0	336.9458	338.6	5,120,000	1.04 hr



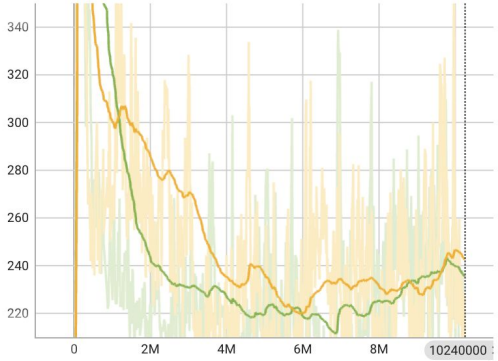
Run ↑	Smoothed	Value	Step	Relative
A2C_no_modification_0	162.7022	150.5525	5,000,000	1.166 hr
DQN_no_modification_0	221.2756	220.2202	4,999,740	1.104 hr
PPO_no_modification_0	243.5695	243.3219	5,120,000	1.04 hr

rollout/ep_len_mean



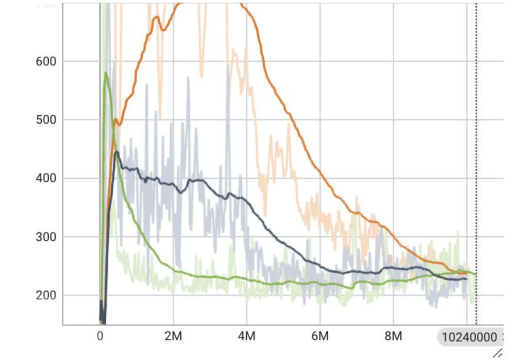
Run ↑	Smoothed	Value	Step	Rela
PPO_no_modification_10M_0	242.4926	206.05	10,240,000	3.31
PPO_reward_1_10M_0	309.8306	327.99	10,240,000	3.37
PPO_reward_2_10M_0	290.39	230.87	10,240,000	3.34
PPO_reward_3_10M_0	235.4261	202.44	10,240,000	2.85

rollout/ep_len_mean



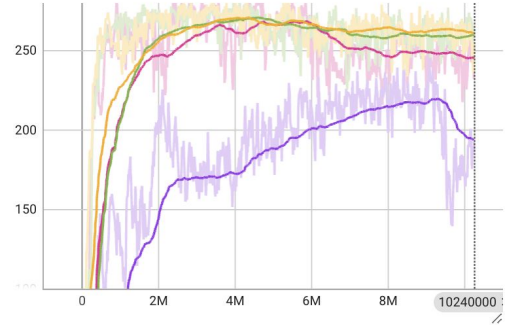
Run ↑	Smoothed	Value	Step	Rela
PPO_no_modification_10M_0	242.4926	206.05	10,240,000	3.31
PPO_reward_3_10M_0	235.4261	202.44	10,240,000	2.89

rollout/ep_len_mean



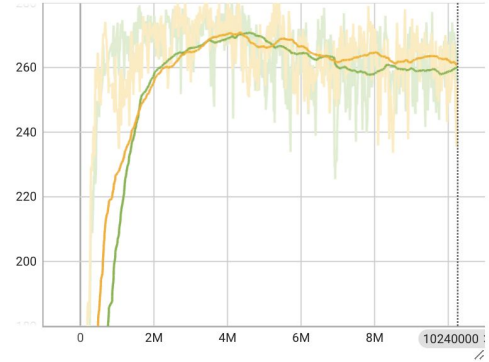
Run ↑	Smoothed	Value	Step	Relative
A2C_reward_3_10M_0	228.4001	242.26	10,000,000	1.802 hr
DQN_reward_3_10M_0	237.7494	243.48	9,999,876	2.013 hr
PPO_reward_3_10M_0	235.4261	202.44	10,240,000	2.891 hr

rollout/ep_rew_mean



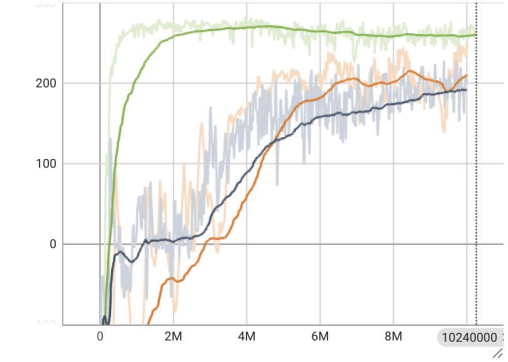
Run ↑	Smoothed	Value	Step	Re
PPO_no_modification_10M_0	260.7368	254.3159	10,240,000	3.3
PPO_reward_1_10M_0	193.992	192.7067	10,240,000	3.3
PPO_reward_2_10M_0	245.3129	233.9543	10,240,000	3.3
PPO_reward_3_10M_0	259.9332	265.1743	10,240,000	2.8

rollout/ep_rew_mean



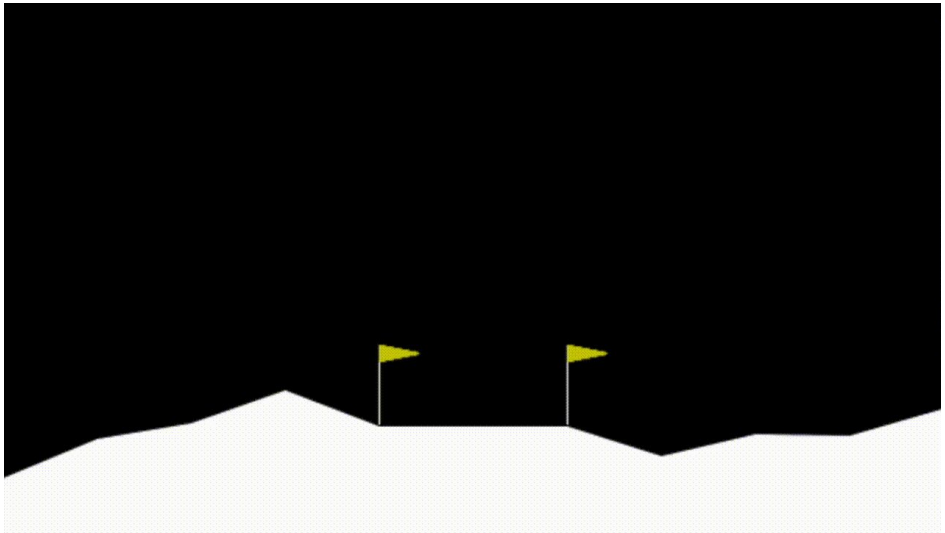
Run ↑	Smoothed	Value	Step	Re
PPO_no_modification_10M_0	260.7368	254.3159	10,240,000	3.3
PPO_reward_3_10M_0	259.9332	265.1743	10,240,000	2.8

rollout/ep_rew_mean

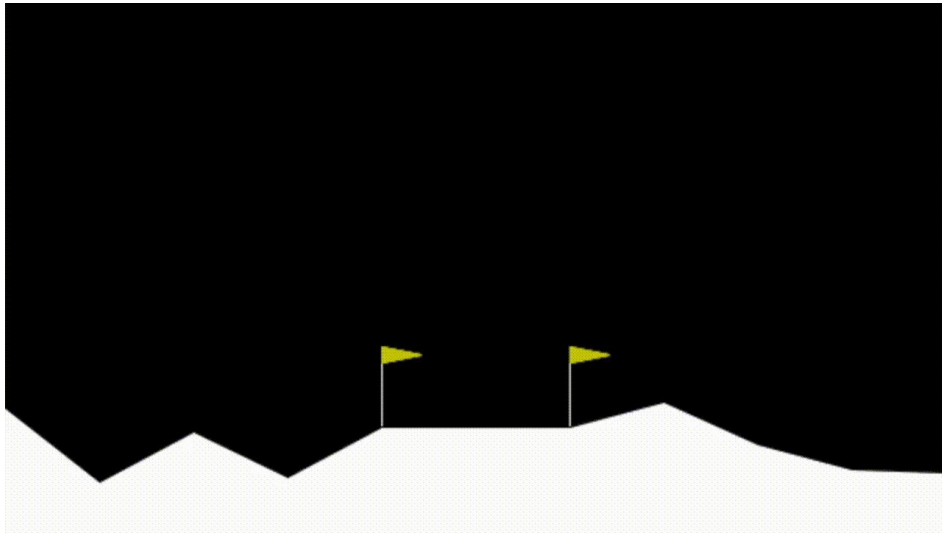


Run ↑	Smoothed	Value	Step	Relative
A2C_reward_3_10M_0	192.5956	204.4394	10,000,000	1.802 hr
DQN_reward_3_10M_0	210.9206	240.5432	9,999,876	2.013 hr
PPO_reward_3_10M_0	259.9332	265.1743	10,240,000	2.891 hr

Resultados



Sem modificações



Com modificações

U.PORTO
FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

Francisco Ribeiro | up202104797
Marisa Azevedo | up202108624

Conclusão

Com esta análise, verificámos que o nosso modelo com alterações melhorou ligeiramente em comparação com o modelo sem alterações. Como a duração de um episódio é maioritariamente mais curta e as recompensas permanecem lineares e constantes e, fora disso, também achamos que a nível de renderização, o nosso modelo por norma parece-se comportar melhor, consideramos que as nossas alterações foram positivas e melhoraram o modelo.