Q-Learn e Deep Learn

Alexandre Vitor Silva Braga

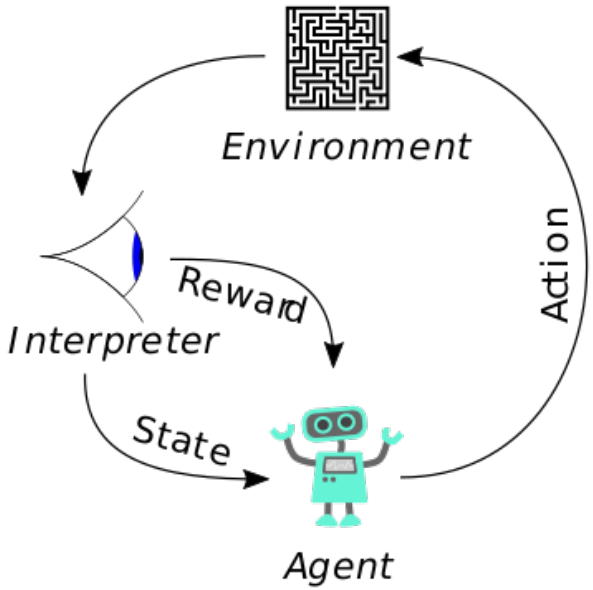
UFJF - 2022.1

30 de outubro de 2022

Table of Contents

Aprendizado por Reforço

- Junto ao Aprendizado Supervisionado e Não Supervisionado constitui um dos 3 paradigmas principais de Machine Learning
- Trata-se da área em que agentes inteligentes tomam ações em um ambiente (environment) a fim de maximizar uma recompensa cumulativa



Q-Learning

- O Q-Learning é um algoritmo **off-policy**, ou seja, não necessita do modelo do environment, por ser independente das ações do agente inteligente.
- O mesmo se baseia em uma tabela Q de valores a serem atualizados, a qual indica a recompensa para as relações estado-ação (state-action)

Q-Table		Actions					
		South (0)	North (1)	East (2)	West (3)	Pickup (4)	Dropoff (5)
States	0	0	0	0	0	0	0
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	328	-2.30108105	-1.97092096	-2.30357004	-2.20591839	-10.3607344	-8.5583017
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	.	.	.	.	.	.	.
	499	9.96984239	4.02706992	12.96022777	29	3.32877873	3.38230603

Q-Learning

- Ademais, como é baseado em valores, atualiza sua função de valor com base em uma equação:

$$Q^{new}(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{current value}} + \underbrace{\alpha}_{\text{learning rate}} \cdot \overbrace{\Big(\underbrace{r_t}_{\text{reward}} + \underbrace{\gamma}_{\text{discount factor}} \cdot \underbrace{\max_a Q(s_{t+1}, a)}_{\text{estimate of optimal future value}} - \underbrace{Q(s_t, a_t)}_{\text{current value}}\Big)}^{\text{temporal difference}}$$
$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\text{new value (temporal difference target)}}$$

- Que representa a soma de:
  - $(1 - \alpha)Q(s_t, a_t)$, o valor atual
  - αr_t, a recompensa no estado $s_t$proporcional a taxa de aprendizado
  - $\alpha\gamma maxQ(s_{t+1}, a)$, a recompensa máxima obtível no estado s_{t+1}

Agentes Inteligentes

- Um agente pode interagir com o ambiente com 1 de 2 maneiras possíveis.
  - A primeira, é utilizando a **Tabela Q** do Q-Learning, aproveitar-se do conhecimento prévio da mesma. Utilizando-a como referência é possível visualizar todas as ações para um determinado estado e selecionar aquela de máximo valor de retorno. Ou seja, o **Exploit** de informações disponíveis
  - A segunda, é tomar uma ação completamente randômica, ao invés de selecionar ações com base na recompensa máxima. Isso permite explorar novas possibilidades, que não seriam descobertas com o simples Exploit de informações anteriores. Ou seja, um **Explore** de informações novas

Epsilon Q-Learning

- É possível equilibrar as táticas de Exploit e Explore usando um parâmetro ϵ, esse novo algoritmo seria denominado Epsilon Q-Learning
- Através de um balanço entre exploração em território desconhecido e aproveitamento de conhecimento prévio, tem como objetivo final obter uma política ótima ou quase ótima, a fim de maximizar uma função de recompensa

Deep Reinforcement Learning

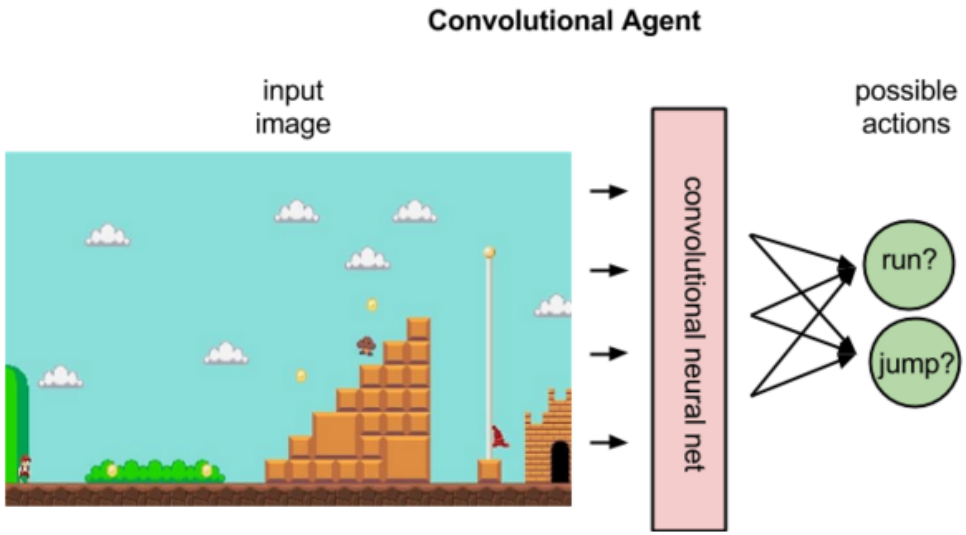
- Pode-se armazenar cada par state-action e sua melhor opção em uma grande tabela, porém o custo computacional seria extremamente grande. Portanto utilizar redes neurais para aproximar funções de valores ou politicas, consequentemente mapeando pares de state-action com Q values pode ser uma solução viável.

DQN

- Como o Q-Learn sofre de instabilidade e divergência quando combinado a funções não lineares alguns algoritmos precisam ser criados para lidar com tal
- O algoritmo DQN (Deep Q-Network) tem como objetivo melhorar e estabilizar o processo de treino do Q-Learning
  - Experience Replay: Todos os passos dos episódios são armazenados em tuplas, em uma memória de replay. Durante o update do Q-Learn, amostras aleatórias são retiradas desse replay, o que melhora a eficiência de dados
  - Periodicamente Atualizar o Target: A função Q é otimizada em direção a um valor alvo que é periodicamente atualizado a certo número de episódios, o que evita oscilações de curto prazo

Algoritmos Deep RL

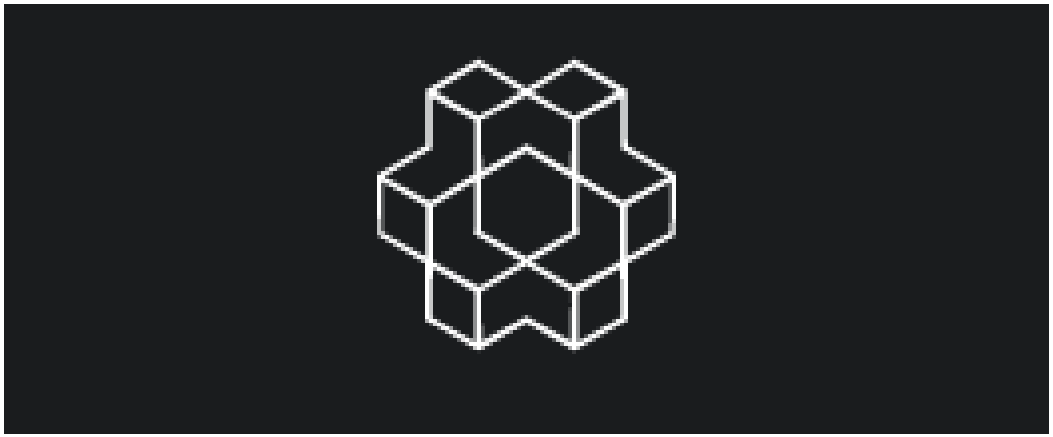
- Logo, ao invés de utilizar uma Tabela Q para armazenar, indexar e atualizar, é muito mais vantajoso treinar redes neurais para predizer quão valiosos os pares são



- Neste seminário veremos alguns algoritmos de Deep RL, como o DQN, e compararemos seus desempenhos

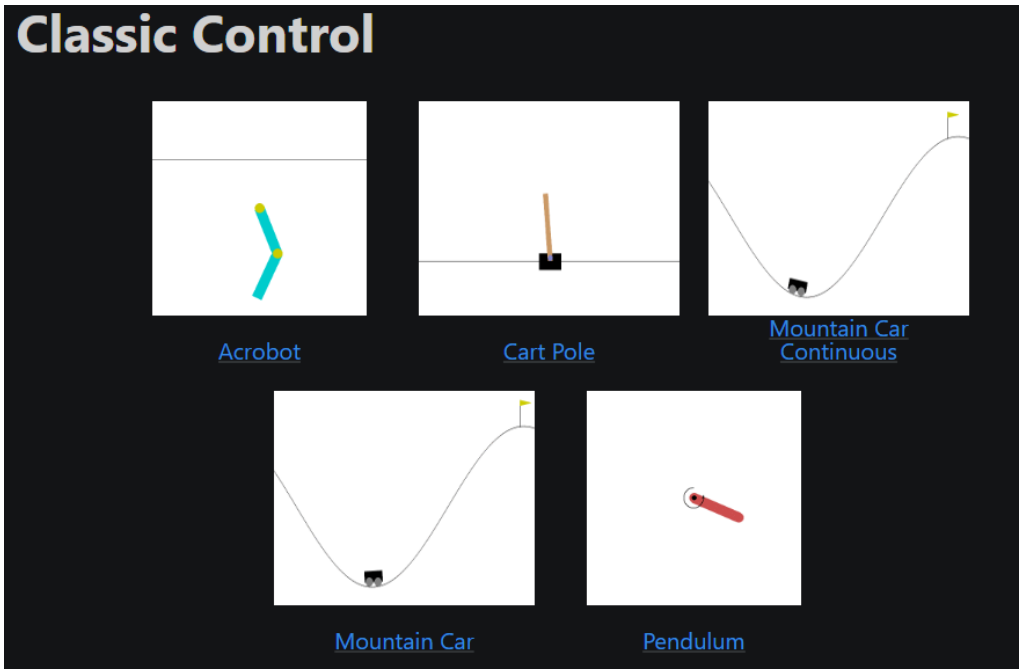
Table of Contents

Open AI Gym



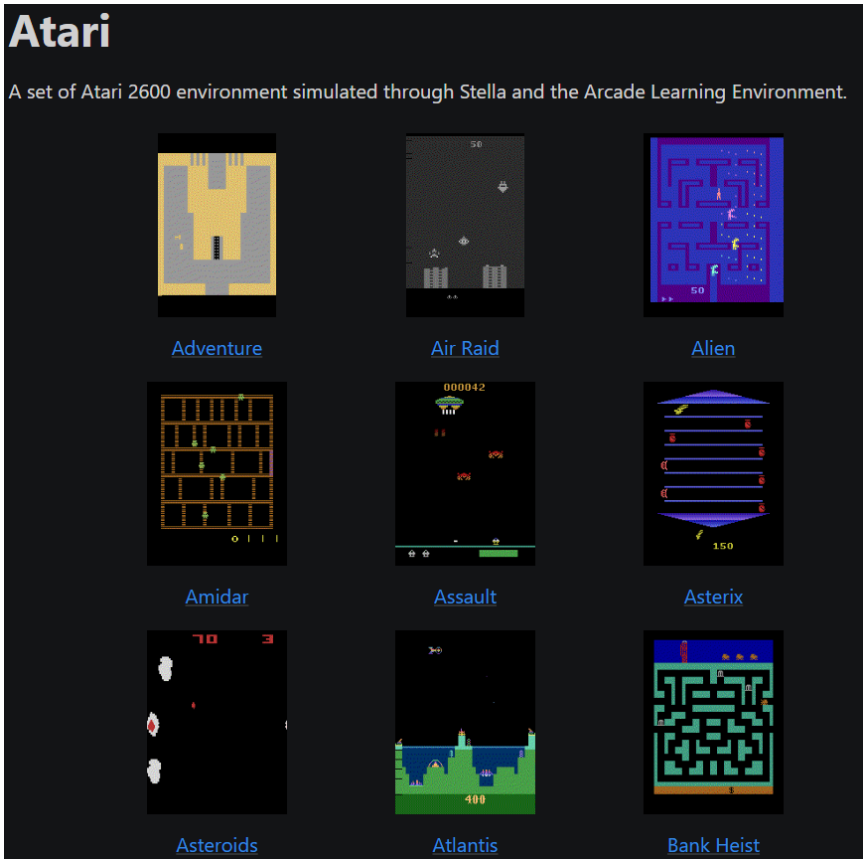
- Trata-se de uma API para Aprendizado por Reforço, possuíndo uma diversa coleção de environments de referência
- Capaz de criar ambientes simulados de complexidade crescente em que agentes podem interagir, permite testar mais facilmente agentes em environments predefinidos ou customizados

Classic Environments



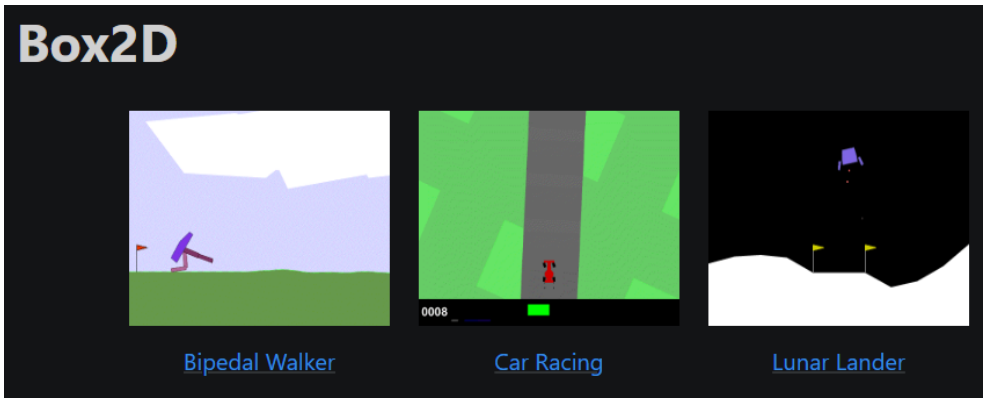
Dentre os environments pré-definidos podemos encontrar, os problemas de controle clássicos

Atari Environments



Podemos encontrar também, os environments de Atari, nos quais o A2C possui benchmarks

Box2D Environment



Por fim, encontramos os enviroments de Box2D, nos quais testaremos o Q-Learn e o compararemos com o A2C

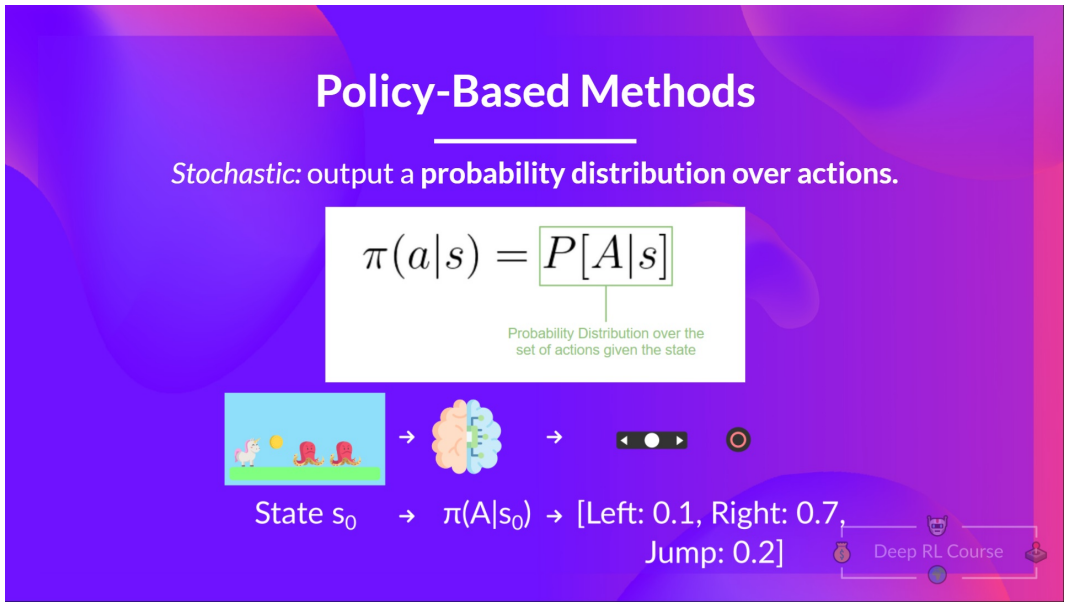
Objetivo

- Avaliar o desempenho dos algoritmo de aprendizado por reforço profundo A2C e DDPG
- Os environments a serem testados pertencem ao pacote Box2D
- Ao final métricas como recompensa total em relação ao número de estados serão comparadas

Table of Contents

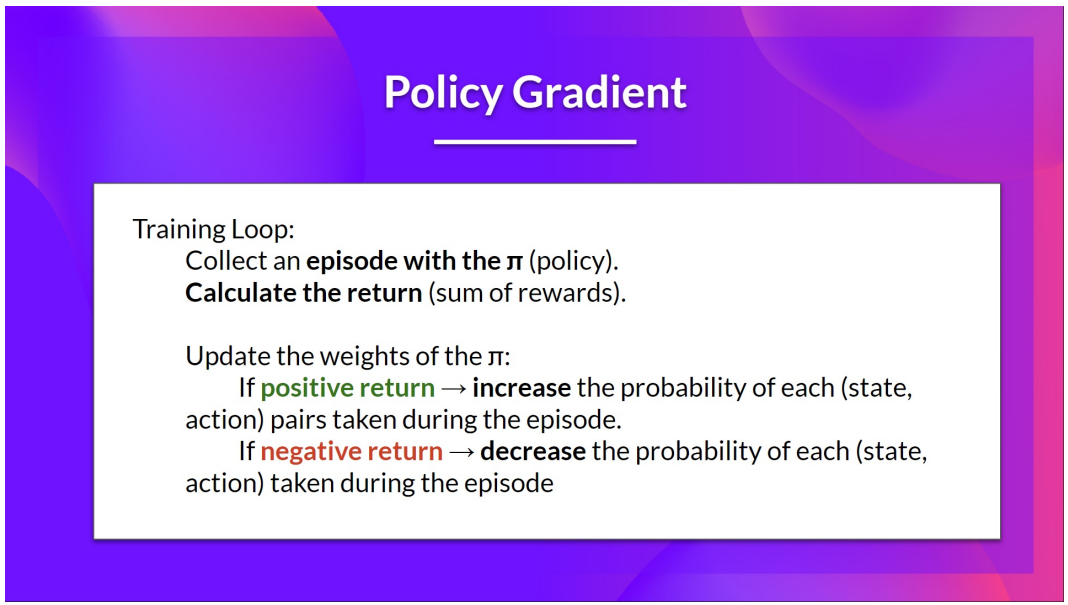
Policy-Based Algorithms

- São algoritmos **Policy-based**, ou seja, desejamos otimizar a política diretamente sem a necessidade de uma função valor, dependendo somente das ações tomadas pelo agente



Método do Gradiente

- Para essa otimização direta, é comum estimar-se os pesos da política ótima usando o método do gradiente
- O objetivo principal desse método é controlar a distribuição da probabilidade das ações, de forma que a ação de melhor retorno seja amostrada mais frequentemente



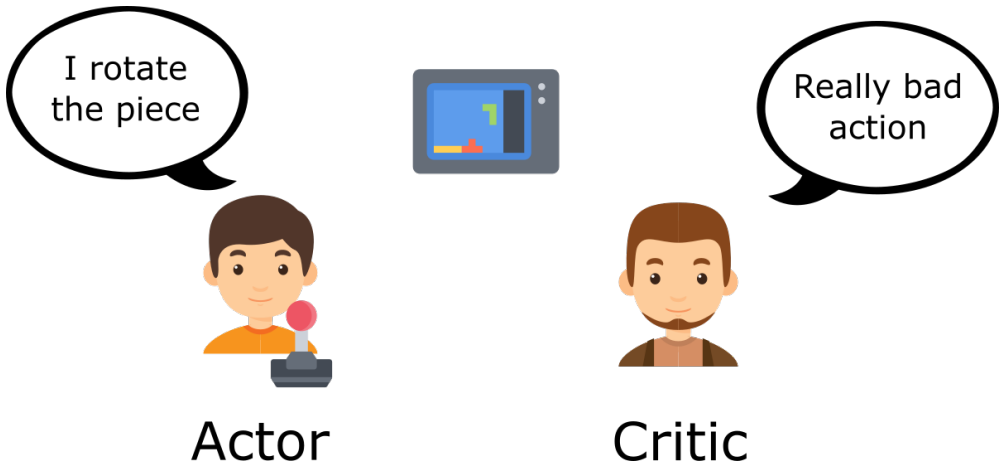
Policy-Gradient

- Métodos baseados em política que fazem uso do gradiente descendente (Policy-Gradient Methods) possuem diversas vantagens sobre outros métodos de Deep Q-Learning
  - É possível estimar a política ideal sem armazenar dados adicionais
  - Podem aprender políticas estocásticas, probabilísticas, permitindo agentes a explorar espaços de estado sem tomar sempre a mesma ação
- Consequentemente, não é preciso implementar a alternância entre exploit e explore como no Epsilon Q-Learning
- Livramo-nos também do problema de perceptual aliasing, quando dois estados parecem similares mas necessitam de ações diferentes
- Por fim, são mais efetivos em espaços contínuos e de dimensões maiores

Actor-Critic

- Dois componentes no Policy-Gradient são o modelo de políticas e a função valor. Conhecendo ambos, a função pode auxiliar a reduzir a variância no método do gradiente
- O método Actor-Critic consiste de 2 modelos, cujos parâmetros são compartilhados

Actor-Critic



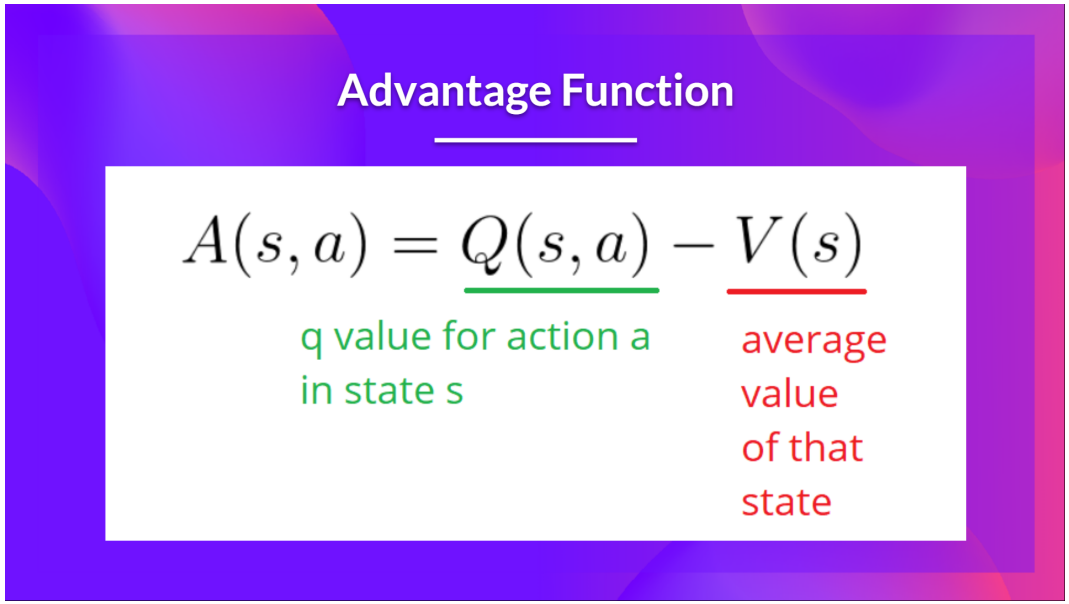
- O Critico, que atualiza os parâmetros da função de valor, indicando quão bom uma ação tomada é: $Q_w(a|s)$
- O Ator, que atualiza os parâmetros da política na direção sugerida pelo critico, controlando como o agente age: $\pi_\theta(a|s)$

Table of Contents

A3C

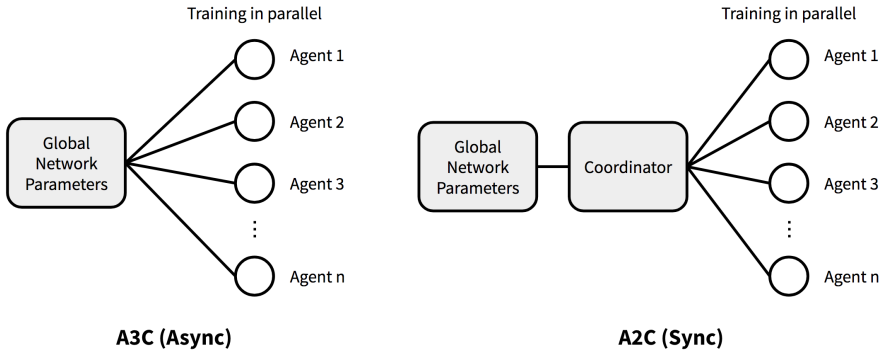
- O algoritmo denominado Asynchronous Advantage Actor-Critic (A3C), utiliza-se de uma função de Advantage, uma medida de quão boa ou ruim certa decisão para determinado estado é em relação à média de todas ações possíveis naquele estado
- Neste caso, passa-se a ter múltiplos criticos, os quais deixam de possuir uma função de valor em troca da função de Advantage. E consequentemente, múltiplos atores são treinados em paralelo, atualizando os parâmetros globais de tempo em tempo

A3C



- Ou seja, temos o cálculo de uma recompensa extra que uma ação produz em relação à média geral do estado. Como possuímos múltiplos atores e criticos, o A3C funciona bem para treinamentos paralelos

A2C



- Já o A2C é a versão síncrona do algoritmo anterior. Se no A3C cada agente se comunicava individualmente, poderia ocorrer de um ator estar em um estado e outro no estado seguinte, logo, não haveria uma garantia de atualização ótima
- Para resolver a inconsistência, o A2C utiliza-se de um coordenador que espera todos os atores paralelos finalizarem seu trabalho, para só então atualizar na iteração seguinte os parâmetros globais

Table of Contents

DPG

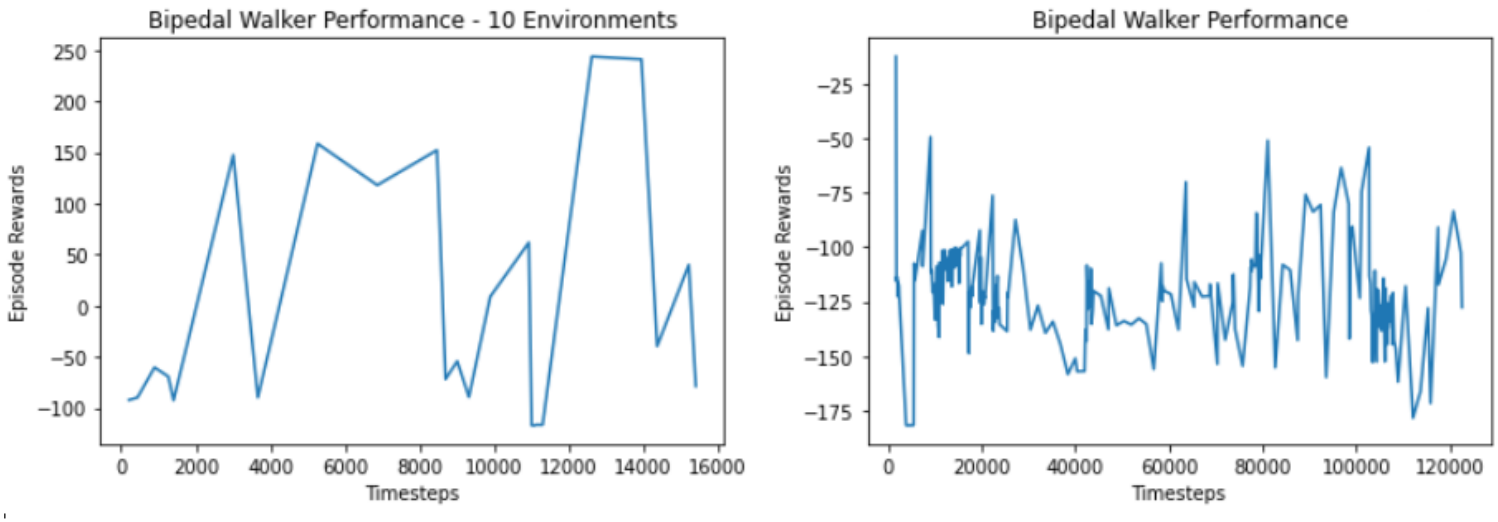
- Outro método baseado em política é o Deterministic Policy Gradient (DPG)
- Diferentemente dos métodos Actor Critic anteriores, nos quais os parâmetros da política eram estocásticos, o DPG a modela como uma decisão determinística

DDPG

- Em foco, o Deep Deterministic Policy Gradient (DDPG), pois o mesmo trata-se de uma combinação do DPG com o DQN
- Utiliza-se das técnicas a seguir:
  - Replay Buffers - Assim como no DQN o buffer que armazena as experiências passadas deve ser grande suficiente para uma boa amostragem mas sem armazenar todas possíveis
  - Target Networks - Diferentemente do DQN, a rede neural com as ações de maior recompensa (alvo) são atualizadas suavemente ao invés de congeladas por um número de episódios
  - DPG com Actor-Critic - Q-Learn não funciona corretamente em espaçoes contínuos, uma aproximação actor-critic baseada no DPG permite que diferentemente do DQN, o DDPG dê certo

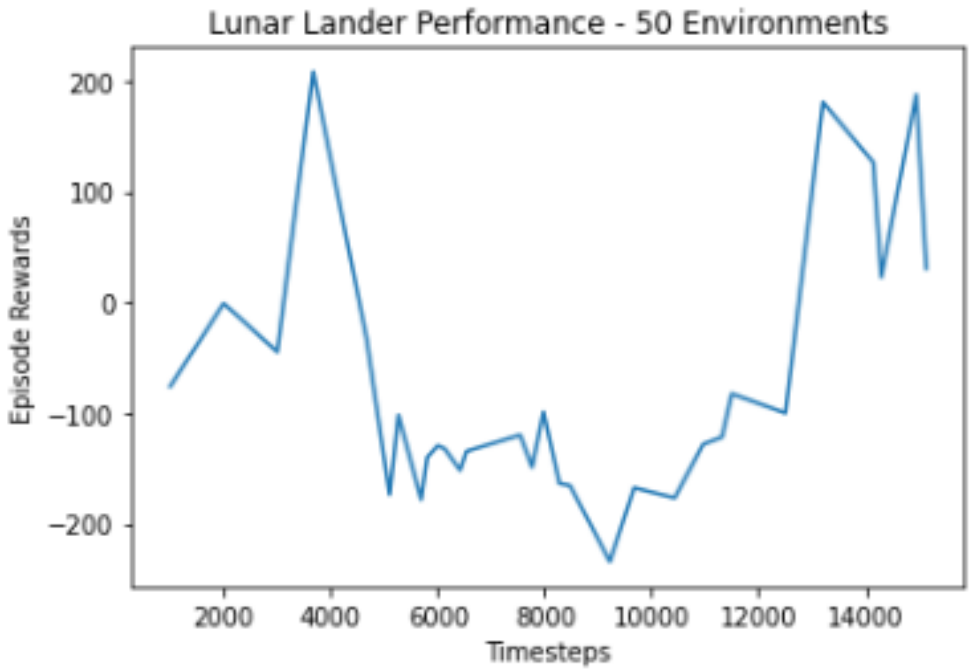
Table of Contents

Bipedal Walker



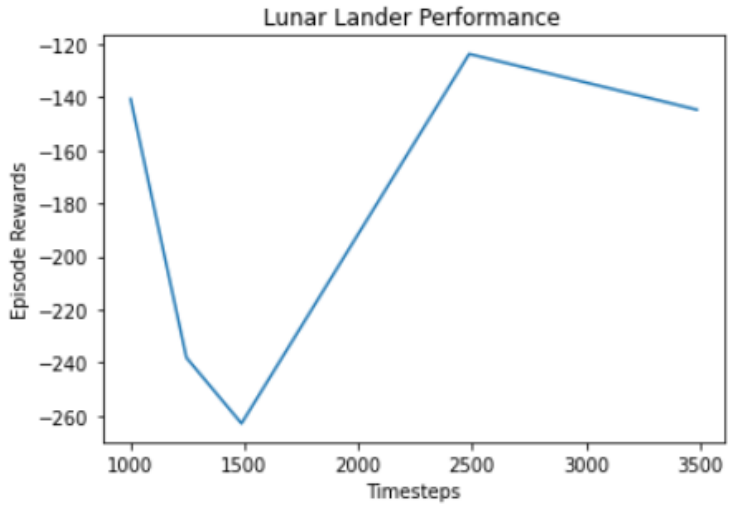
A esquerda, A2C com 10 environments em paralelo, a direita o DDPG

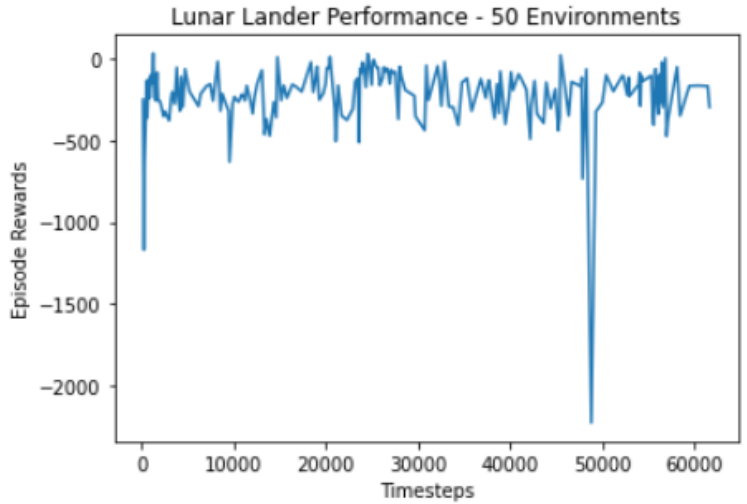
Lunar Lander Discrete



A2C com 50 environments em paralelo, o DDPG não foi capaz de rodar o environment discreto

Lunar Lander Continuous





A esquerda, A2C com 50 environments em paralelo, a direita o DDPG

Table of Contents

Conclusões

- Podemos notar que o A2C por sua capacidade de multiprocessamento de ambientes em paralelo consegue com um número menor de episódios evoluir muito mais rapidamente as recompensas médias dos mesmos
- Além disso, o DDPG não pode ser testado em um ambiente discreto para compararmos, porém, pelos resultados obtidos através do A2C, podemos notar que a estratégia actor-critic em environments discretos pode ser bem promissora
- Por fim, ambos algoritmos de Deep Learn demonstram evoluções na área de aprendizado por reforço. Ademais área ainda possui desafios, mas possui grande potencial, o que evidência a relevância do tema

Referências

- Q-Learning e ε-Q-Learning
  - Q-learning
  - Epsilon Greedy Q-Learning
- Deep Learning
  - Beginner's Guide to Deep RI
  - Deep RL Class
- Policy-Gradient
  - Deep RL Class - Policy-Gradient
  - Policy Gradient Algorithms

Referências

- A3C
  - A3C Paper
  - Understanding Actor Critic Methods
- A2C
  - Deep RL Class - A2C
  - OpenAI A2C
  - A2C Paper
  - A2C Stablebaselines3
- DDPG
  - DDPG Paper
  - DDPG

Referências

- Problemas com Aprendizado por Reforço
  - RL is Hard
  - Faulty Reward Functions
- Redes Neurais para Jogos
  - MariFlow
  - MariFlow Documentation

Obrigado pela atenção !