INSTITUTO DE MATEMATICA E ESTATÍSTICA
UNIVERSIDADE DE SÃO PAULO

XLIX Programa de Verão (2020) - Introdução ao Aprendizado por Reforço

Tópicos Avançados: Desafios de RL

Thiago Pereira Bueno
tbueno@ime.usp.br

IME - USP, 17/02/2019
LIAMF: Grupo PAR (Planejamento e Aprendizado por Reforço)

Aula 5 - Desafios de Aprendizado por Reforço

Agenda

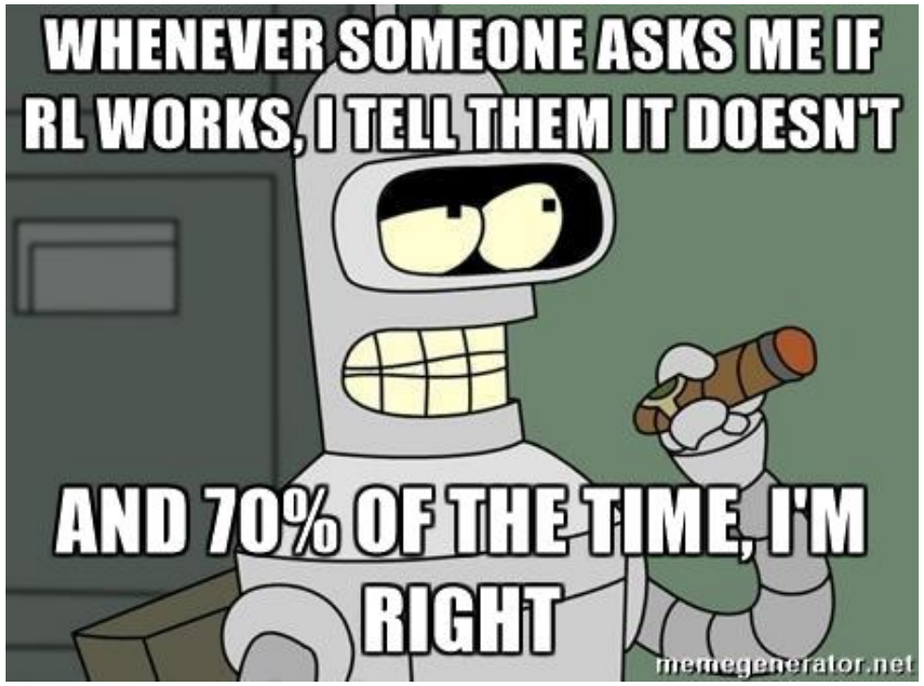
1. Algoritmos de RL:
    A. Eficiência: quanto tempo será necessário para atingir a convergência?
    B. Estabilidade: o treinamento da política irá convergir?

2. Formulação do problema:
    A. Maximização de retorno esperado: é o melhor que podemos fazer?
    B. *Exploration* & *Exploitation:* como incentivar o agente a continuar aprendendo?

Objetivos

- Entender algumas das limitações e dificuldades fundamentais de *Deep RL*
- Familiarizar-se com técnicas avançadas de algoritmos *Actor-Critic*
- Ter uma visão geral sobre diferentes áreas de pesquisa em RL

Principais Desafios de RL

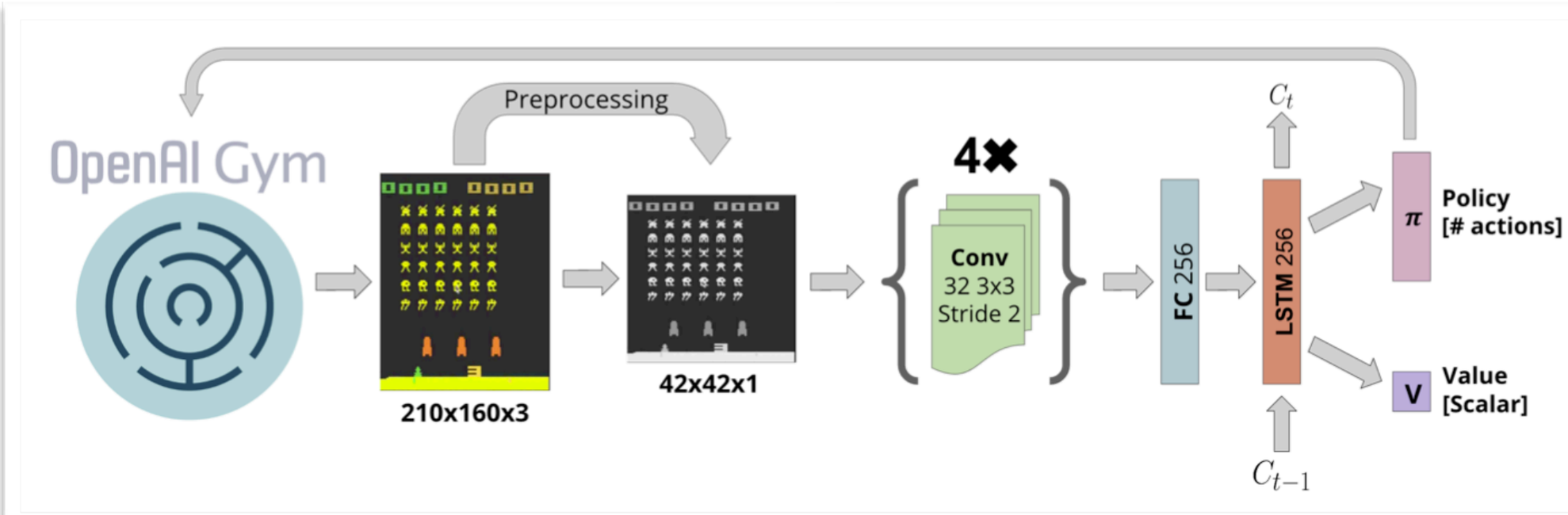


Deep Reinforcement Learning Doesn't Work **Yet**

https://www.alexirpan.com/2018/02/14/rl-hard.html

- Deep RL necessita de uma **enorme quantidade de dados**
- Definir objetivos via "**engenharia de recompensas**" não é nada trivial em boa parte dos casos
- **Ótimos locais** podem ser desafiadores ou até inevitáveis
- **Aprendizado é instável** e resultados difíceis de reproduzir

Revisão: *Actor-Critic*

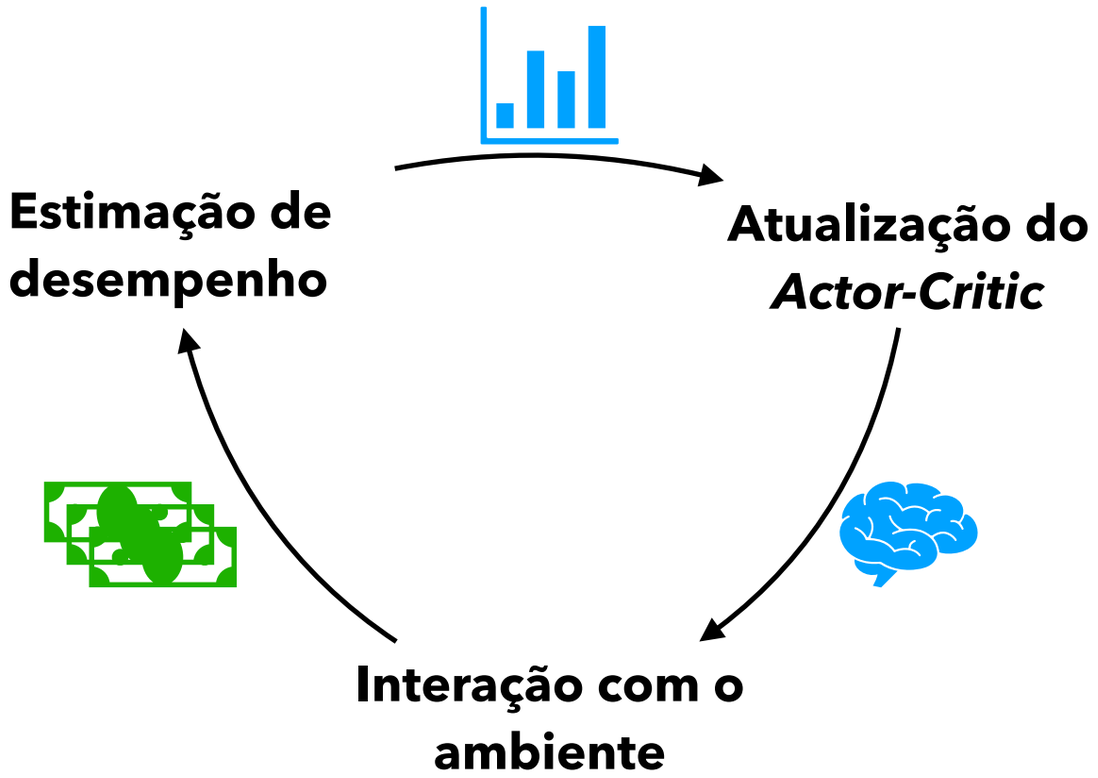


$$[\theta, \phi] \leftarrow [\theta, \phi] + \alpha \nabla_{\theta,\phi} \left(L_{\text{actor}}(\theta) + L_{\text{critic}}(\phi) \right)$$

$$L_{\text{actor}}(\theta) = -\frac{1}{K} \sum_{t=1}^{K} \log \pi_\theta(a_t|s_t) \hat{A}_t^{(n)}$$

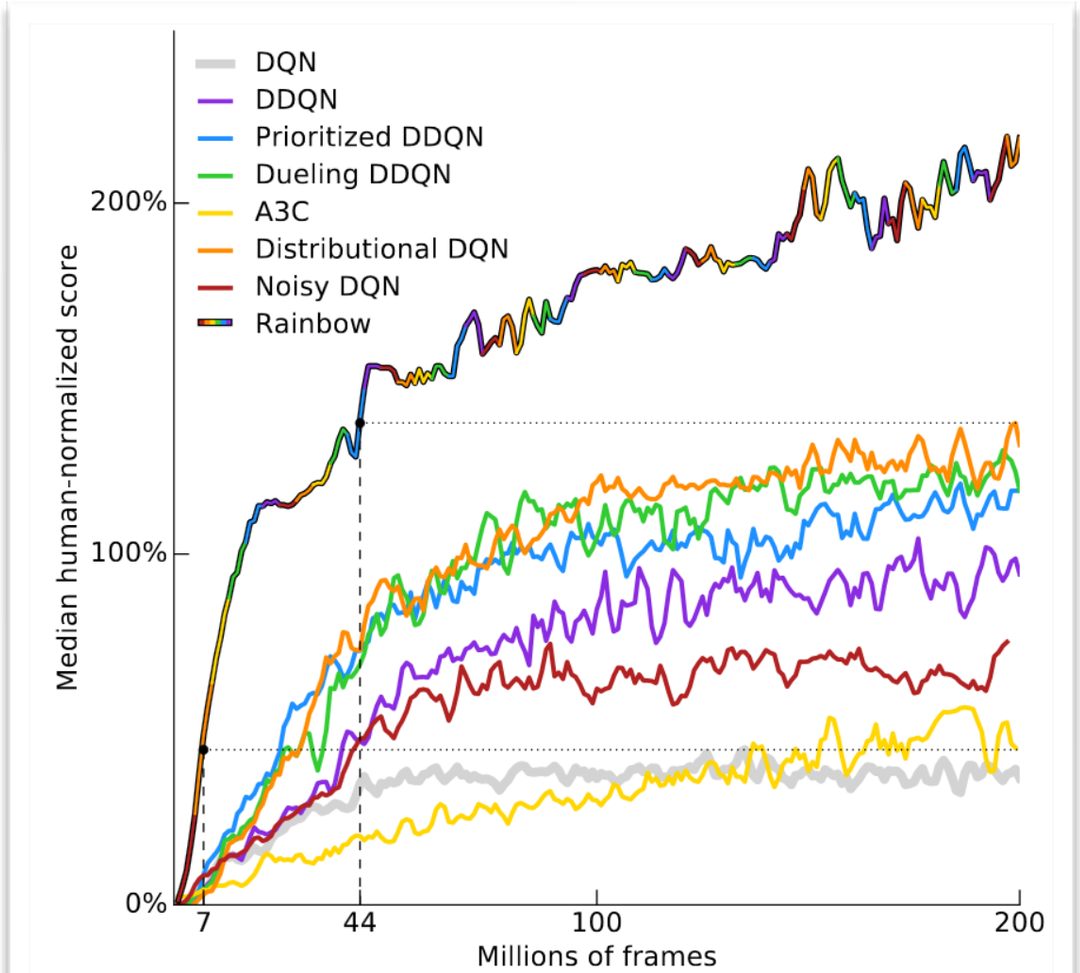
$$L_{\text{critic}}(\phi) = \frac{1}{K} \sum_{t=1}^{K} (V_\phi(s_t) - \hat{R}_t)^2$$

Revisão: *Actor-Critic*



- **Actor-Critic** vem da interpretação intuitiva dos dois principais componentes do agente
  - A política $\pi_\theta(\cdot \mid s)$ recomenda ações para cada estado, portanto é vista como "*actor*"
  - A função $V_\phi(s)$ avalia o retorno esperado sob a política, portanto é vista como "*critic*"

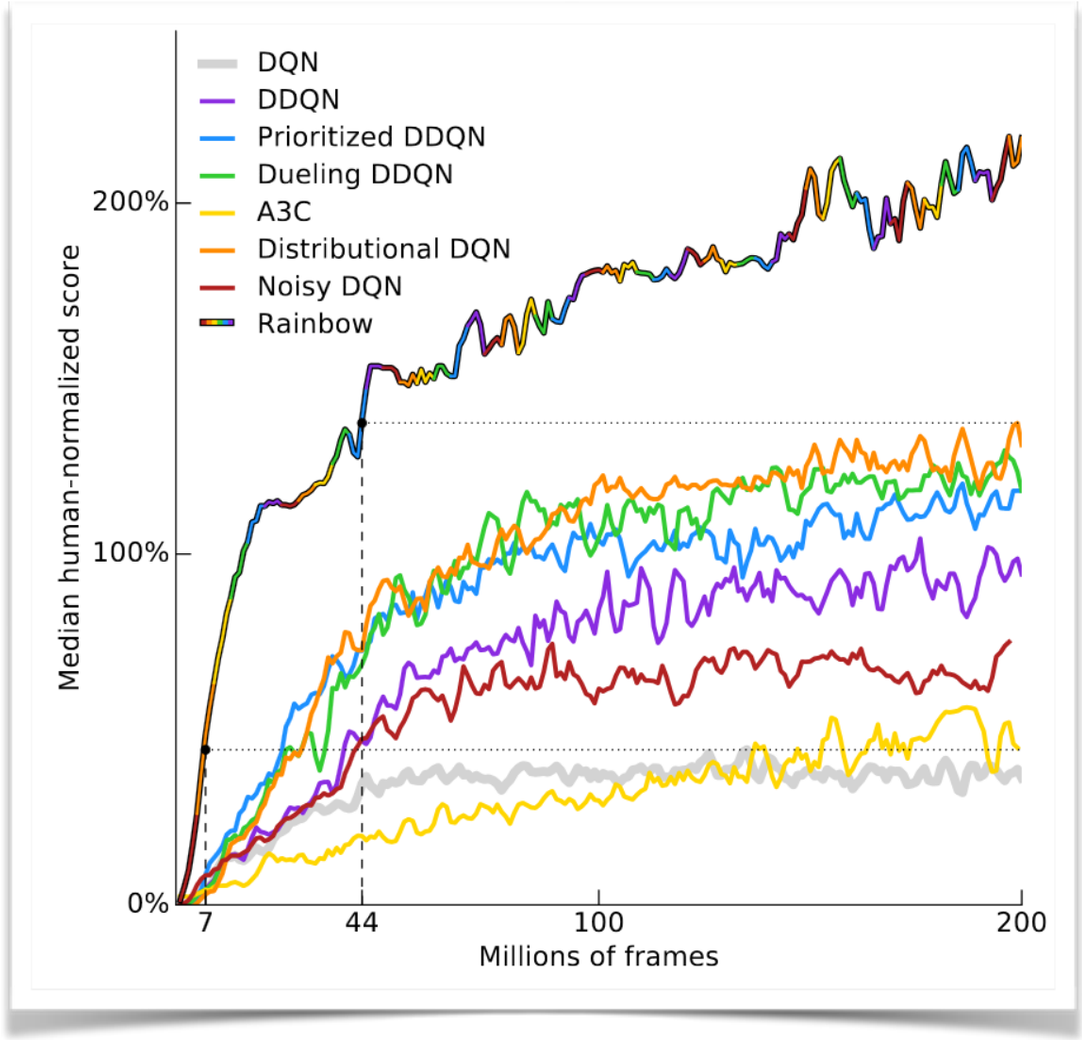
Eficiência computacional (sample complexity)



Rainbow: Combining Improvements in Deep Reinforcement Learning

Eficiência computacional (sample complexity)

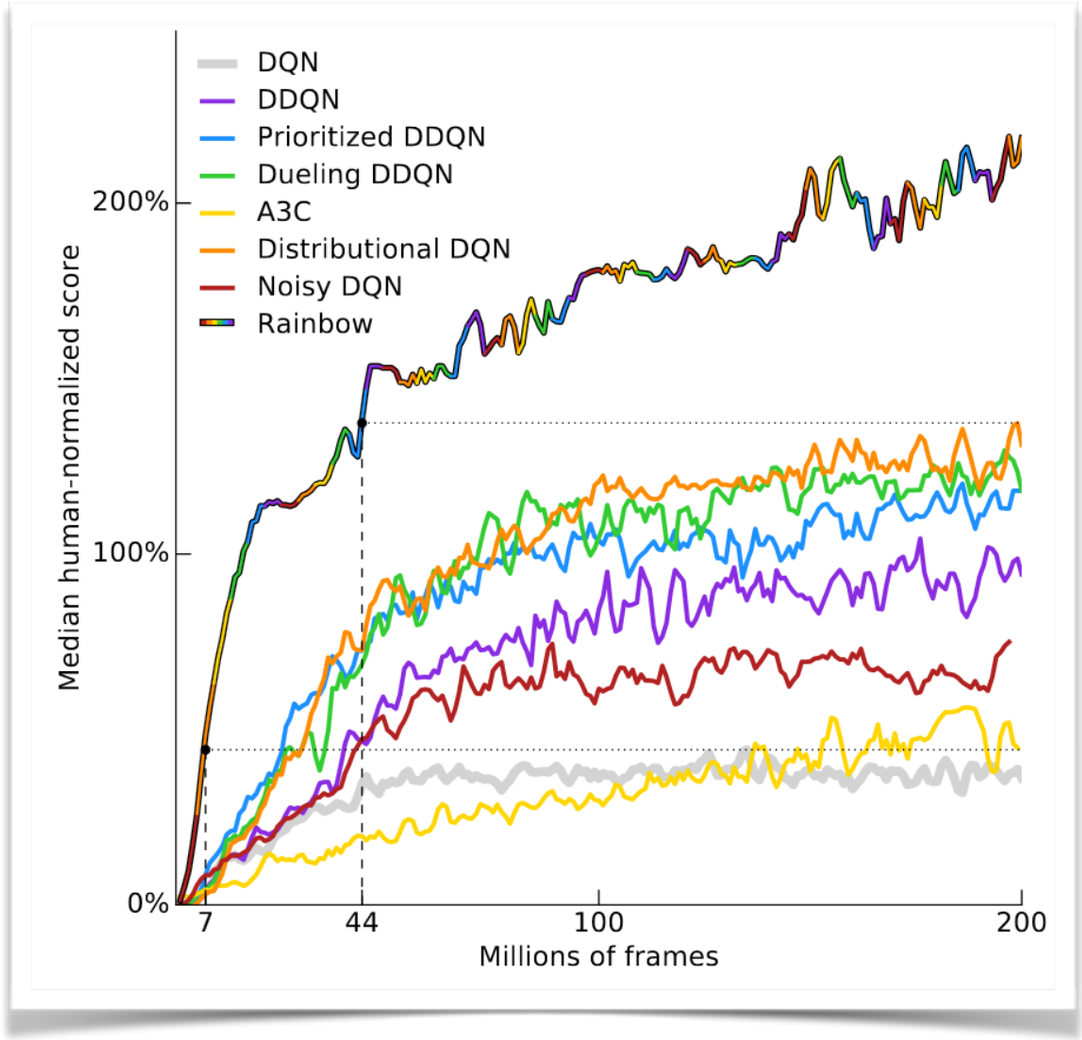
- **Rainbow DQN** é o estado-da-arte em jogos de Atari



Rainbow: Combining Improvements in Deep Reinforcement Learning

Eficiência computacional (sample complexity)

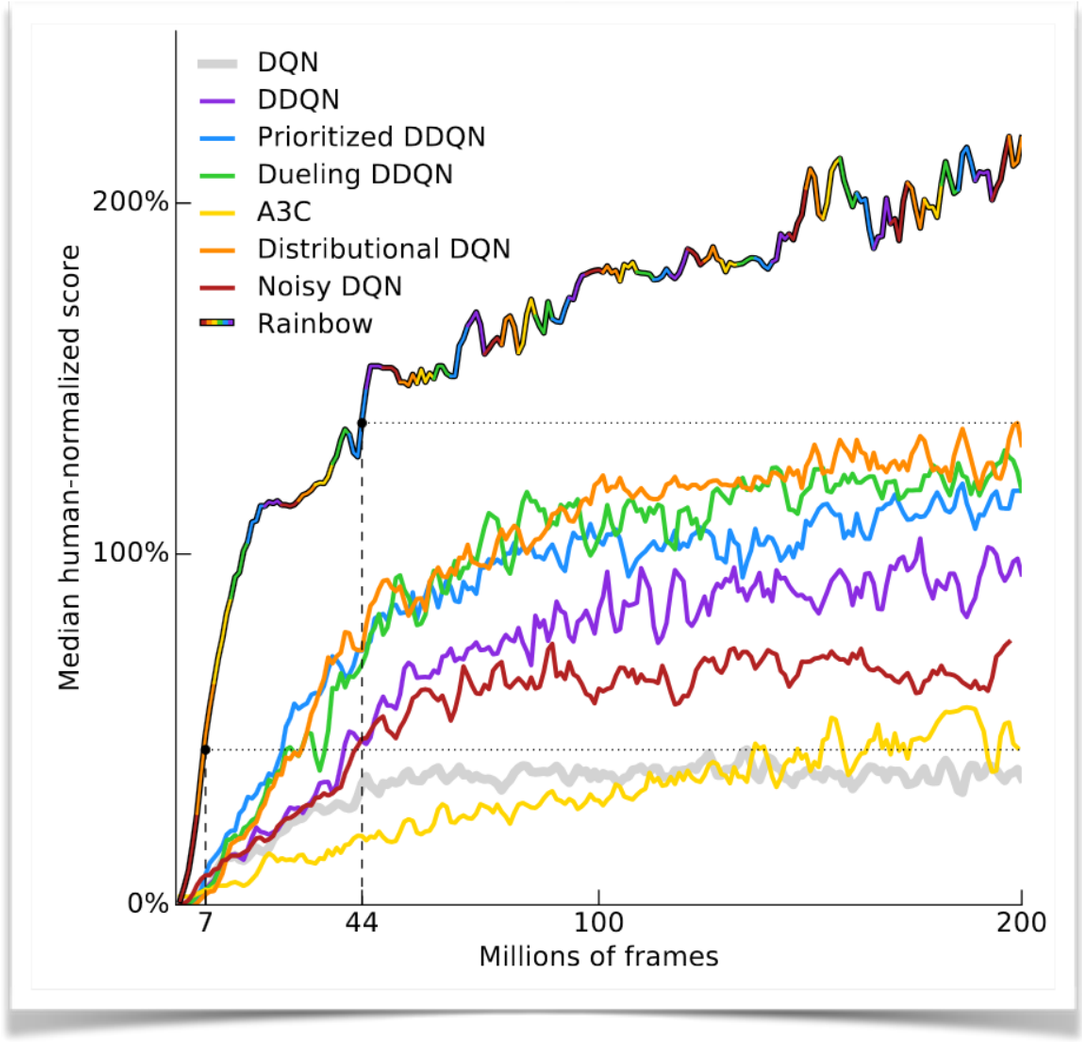
- **Rainbow DQN** é o estado-da-arte em jogos de Atari
- Atingiu desempenho mediano de humanos em 40 dos 57 jogos...



Rainbow: Combining Improvements in Deep Reinforcement Learning

Eficiência computacional (sample complexity)

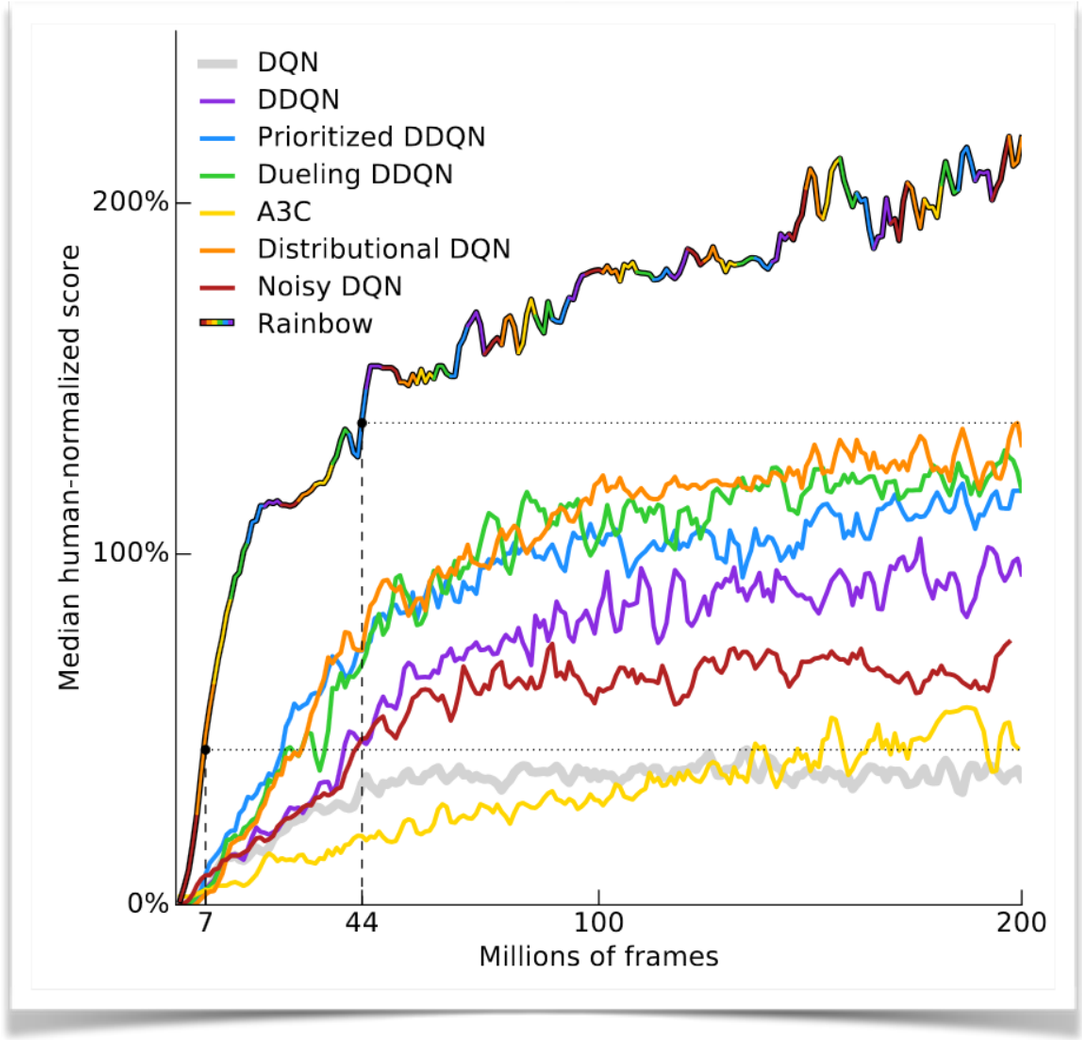
- **Rainbow DQN** é o estado-da-arte em jogos de Atari
- Atingiu desempenho mediano de humanos em 40 dos 57 jogos...
- Entretanto, para atingir esse nível ainda são necessários mais de **18 milhões de *frames***; o que corresponde à **83 horas de jogo**...



Rainbow: Combining Improvements in Deep Reinforcement Learning

Eficiência computacional (sample complexity)

- **Rainbow DQN** é o estado-da-arte em jogos de Atari
- Atingiu desempenho mediano de humanos em 40 dos 57 jogos...
- Entretanto, para atingir esse nível ainda são necessários mais de **18 milhões de *frames***; o que corresponde à **83 horas de jogo**...
- Para a maioria dessas tarefas, um ser humano precisa de poucos minutos de jogo!



Rainbow: Combining Improvements in Deep Reinforcement Learning

Aprendizado *Off-Policy*

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

Aprendizado *Off-Policy*

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

- DQN e outros algoritmos são treinados com dados históricos (*off-policy)*

Aprendizado *Off-Policy*

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

- DQN e outros algoritmos são treinados com dados históricos (*off-policy)*
- Política de coleta de dados é diferente da política que está sendo treinada!

Aprendizado *Off-Policy*

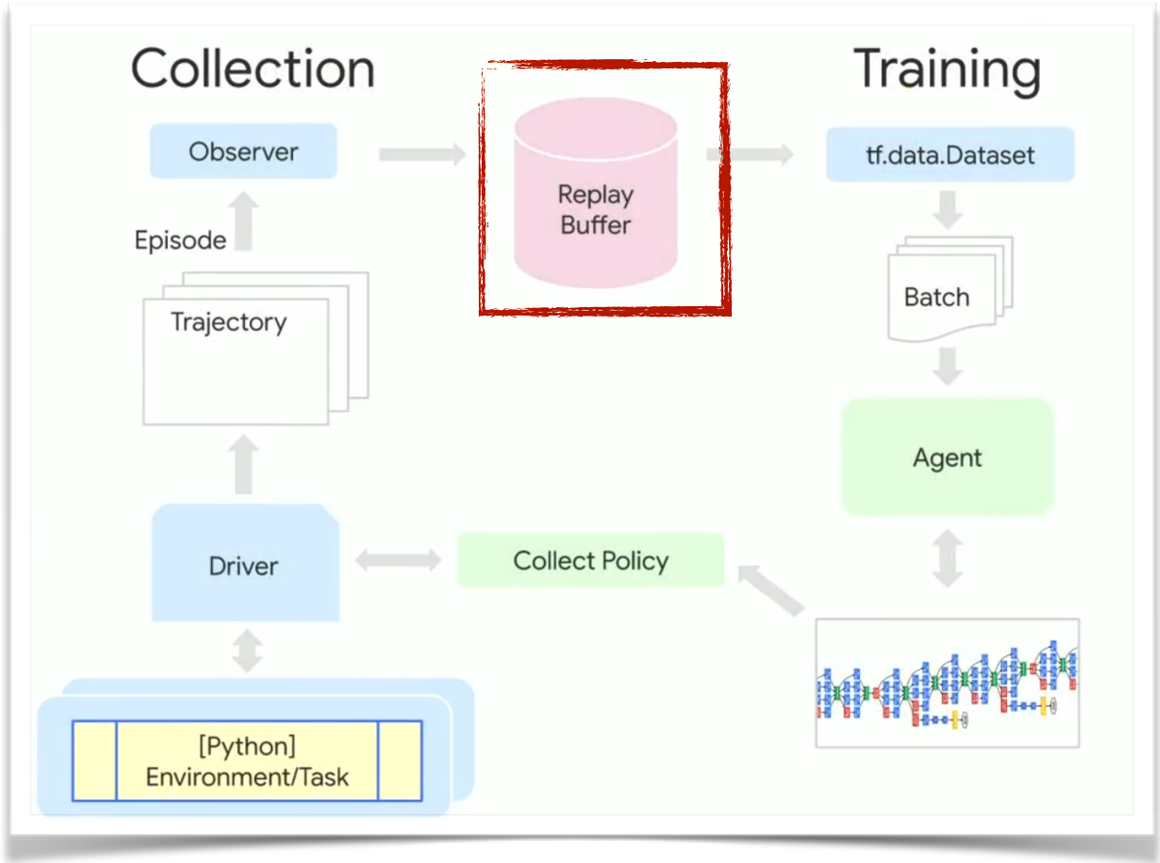
$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

- DQN e outros algoritmos são treinados com dados históricos (*off-policy)*
- Política de coleta de dados é diferente da política que está sendo treinada!
- ***Behavior*** *policy vs* ***Target*** *policy*

Aprendizado *Off-Policy*

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

- DQN e outros algoritmos são treinados com dados históricos (*off-policy)*
- Política de coleta de dados *é* diferente da política que está sendo treinada!
- ***Behavior*** *policy vs* ***Target*** *policy*
- Memórias das trajetórias passadas são salvas em um ***Experience Replay***



https://www.youtube.com/watch?v=tAOApRQAgpc

Eficiência computacional (sample complexity)

gradient-free methods
(e.g. NES, CMA, etc.)

10x

fully online methods
(e.g. A3C)

10x

policy gradient methods
(e.g. TRPO)

10x

replay buffer value estimation methods
(Q-learning, DDPG, NAF, SAC, etc.)

10x

model-based deep RL
(e.g. PETS, guided policy search)

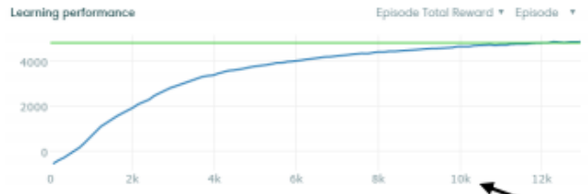
10x

model-based "shallow" RL
(e.g. PILCO)

Evolution Strategies as a Scalable Alternative to Reinforcement Learning

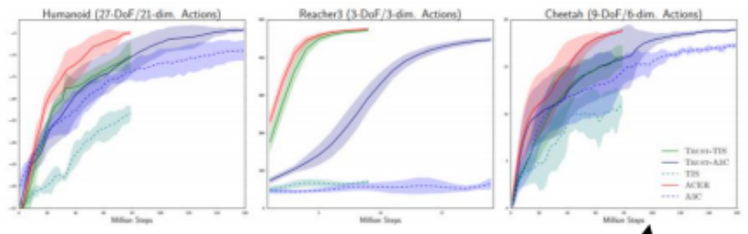
Tim Salimans [1] Jonathan Ho [1] Xi Chen [1] Ilya Sutskever [1]

half-cheetah (slightly different version)



TRPO+GAE (Schulman et al. '16)

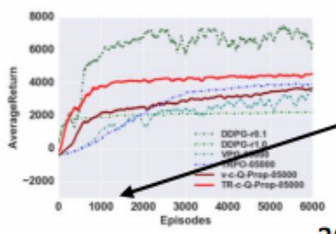
10,000,000 steps
(10,000 episodes)
(~ 1.5 days real time)



Wang et al. '17

100,000,000 steps
(100,000 episodes)
(~ 15 days real time)

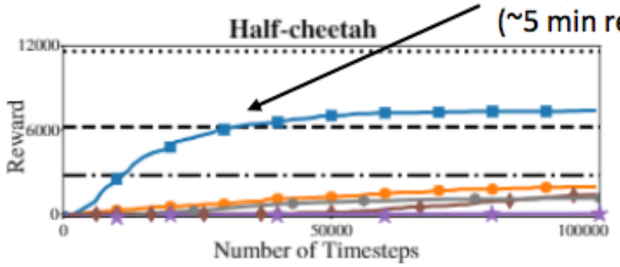
half-cheetah



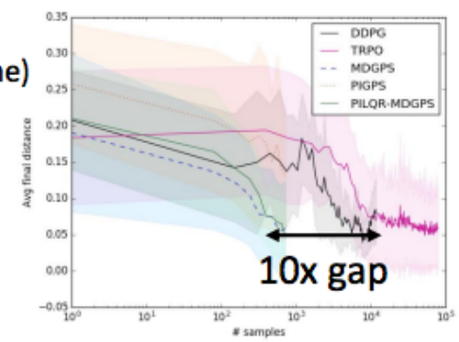
Gu et al. '16

1,000,000 steps
(1,000 episodes)
(~3 hours real time)

30,000 steps
(30 episodes)
(~5 min real time)



Chua et a. '18: Deep Reinforcement Learning in a Handful of Trials



about 20
minutes of
experience on a
real robot

Chebotar et al. '17 (note log scale)

http://rail.eecs.berkeley.edu/deeprlcourse/static/slides/lec-21.pdf

Estabilidade & Convergência: *Policy Optimization*

Estabilidade & Convergência: *Policy Optimization*

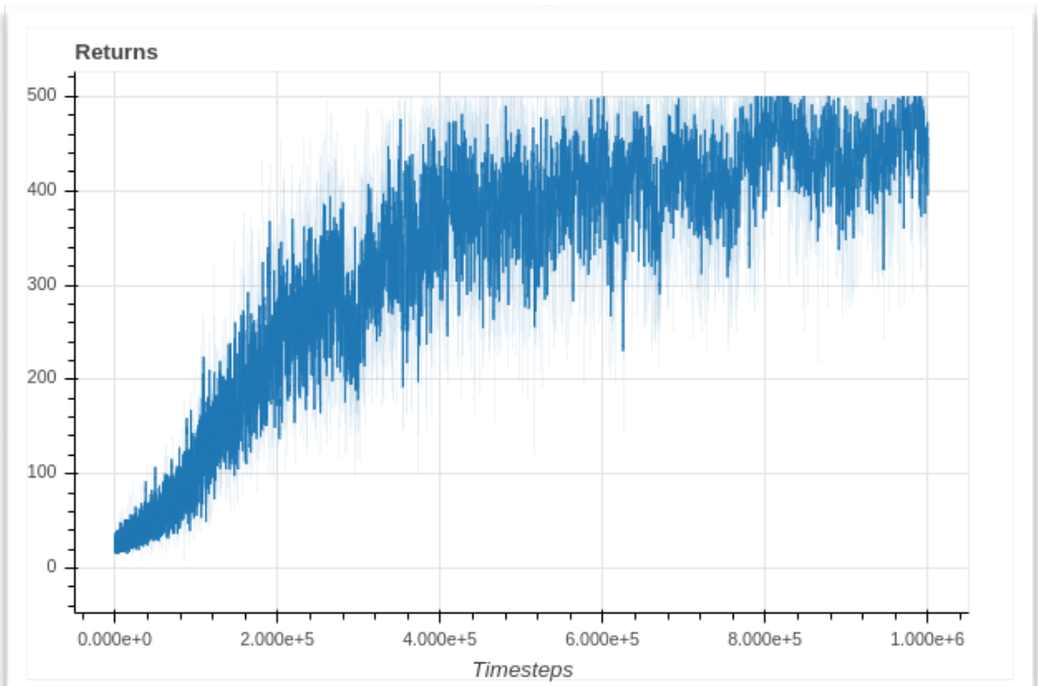
Policy Gradient

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t) R(\tau) \right]$$

Estabilidade & Convergência: *Policy Optimization*

Policy Gradient

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t \mid s_t) R(\tau) \right]$$



Estabilidade & Convergência: *Policy Optimization*

Line Search

Trust Region

Trust Region Policy Optimization (TRPO)

Trust Region

https://arxiv.org/abs/1502.05477

Trust Region Policy Optimization (TRPO)

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta)$$

$$\text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta$$

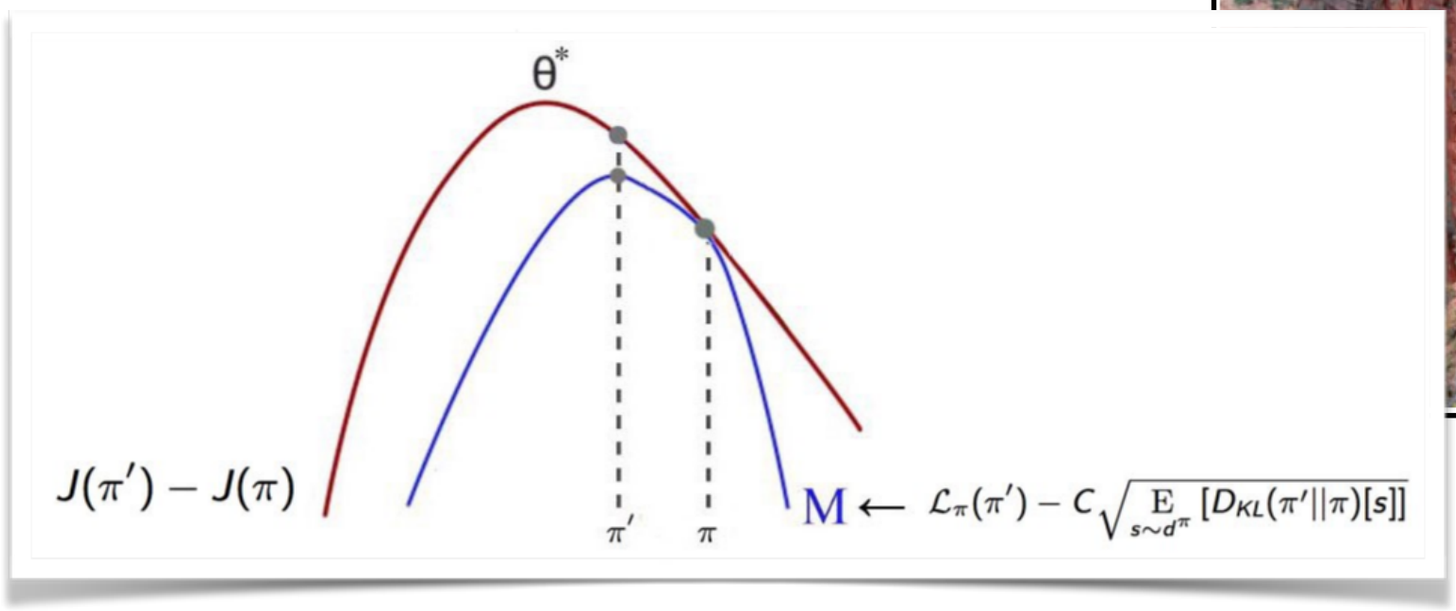
Trust Region

https://arxiv.org/abs/1502.05477

Trust Region Policy Optimization (TRPO)

$$\underset{\theta}{\text{maximize}}\, L_{\theta_{\text{old}}}(\theta)$$

$$\text{subject to } D_{\text{KL}}^{\max}(\theta_{\text{old}}, \theta) \leq \delta$$

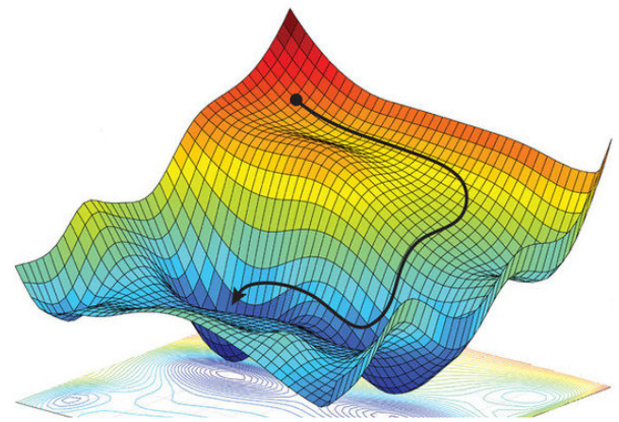


Trust Region

https://arxiv.org/abs/1502.05477

RL como Maximização dos Retornos

$$\pi^* = \arg\max J(\pi) = \arg\max \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T-1} r_{t+1}\right]$$



$$\theta \leftarrow \theta + \alpha\nabla_\theta J(\theta)$$

- Quais são as **premissas básicas** da abordagem de **Maximização dos Retornos**?
- Em quais situações podemos esperar bons resultados para algoritmos "**trial-and-error**" ?

RL como Maximização dos Retornos

https://arxiv.org/abs/1606.01868

- **Recompensas esparsas** ainda são um grande desafio para algoritmos de RL

RL como Maximização dos Retornos

https://arxiv.org/abs/1606.01868

- **Recompensas esparsas** ainda são um grande desafio para algoritmos de RL

RL como Maximização dos Retornos

https://www.youtube.com/watch?v=tlOIHko8ySg

- **Recompensas mal especificadas** podem inviabilizar RL para problemas de tomada de decisão!

RL como Maximização dos Retornos

https://www.youtube.com/watch?v=tlOIHko8ySg

- **Recompensas mal especificadas** podem inviabilizar RL para problemas de tomada de decisão!

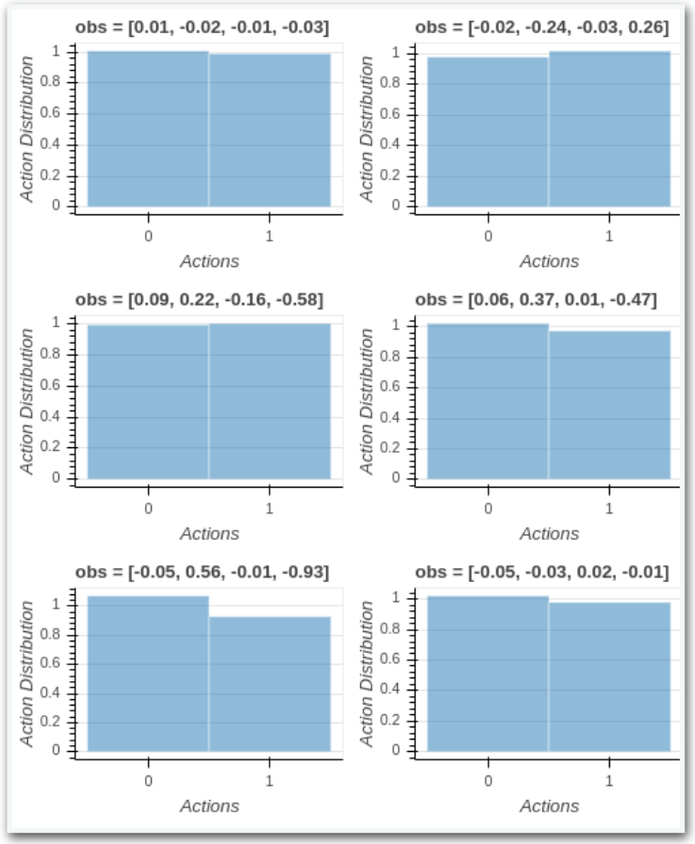
Como incentivar o agente mesmo sem recompensas?

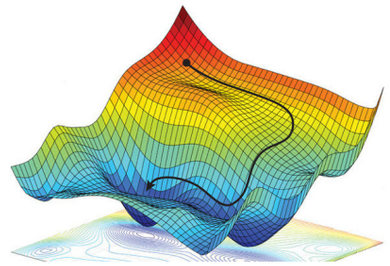
https://www.youtube.com/watch?v=8vNxjwt2AqY

Como incentivar o agente mesmo sem recompensas?

https://www.youtube.com/watch?v=8vNxjwt2AqY

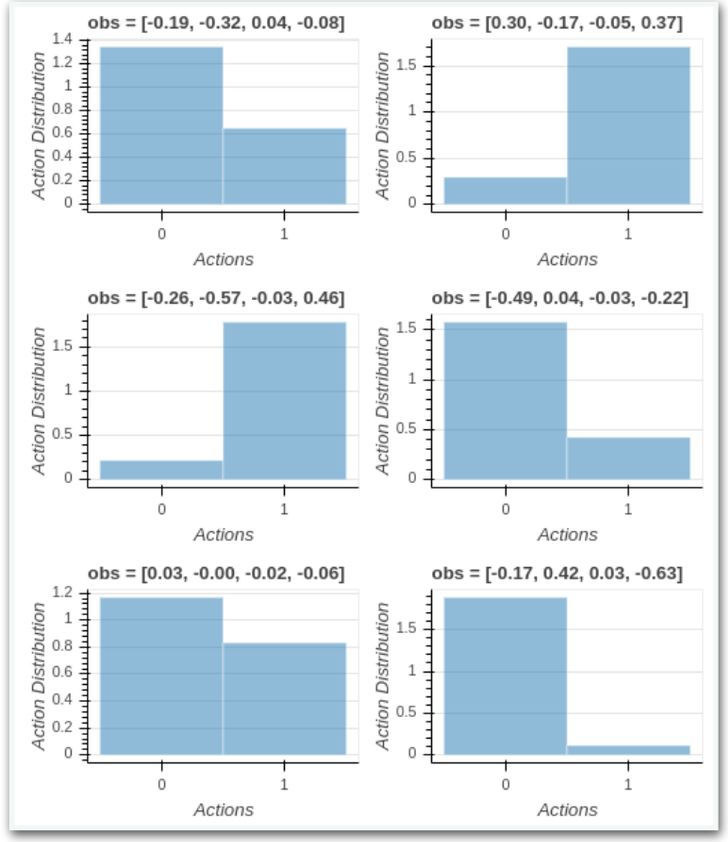
Como incentivar o agente mesmo sem recompensas?





$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$

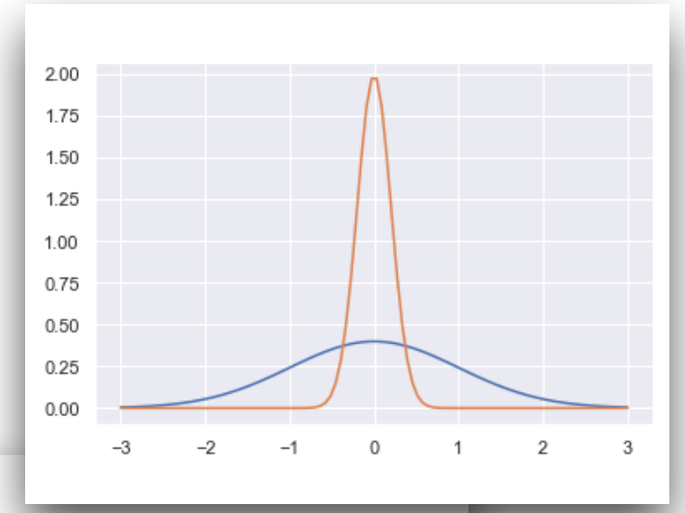
Treinamento

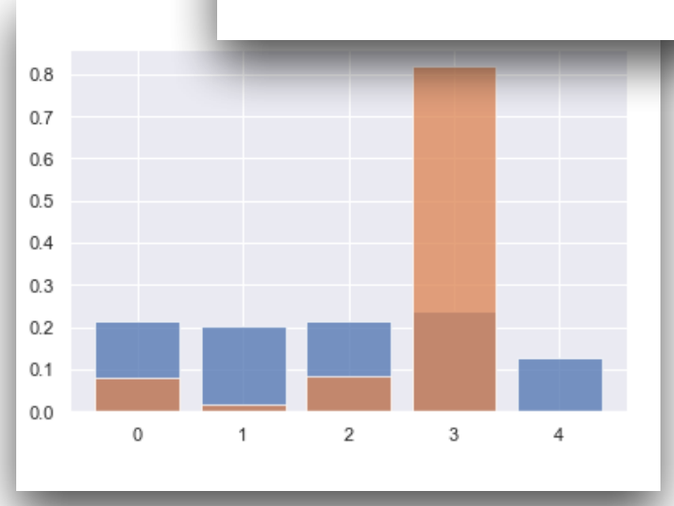


Maximum Entropy (MaxEnt)

$$\pi^*_{\text{MaxEnt}} = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_\pi \left[\sum_{t=0}^{T} r_t + \mathcal{H}(\pi(\cdot|\mathbf{s}_t)) \right]$$

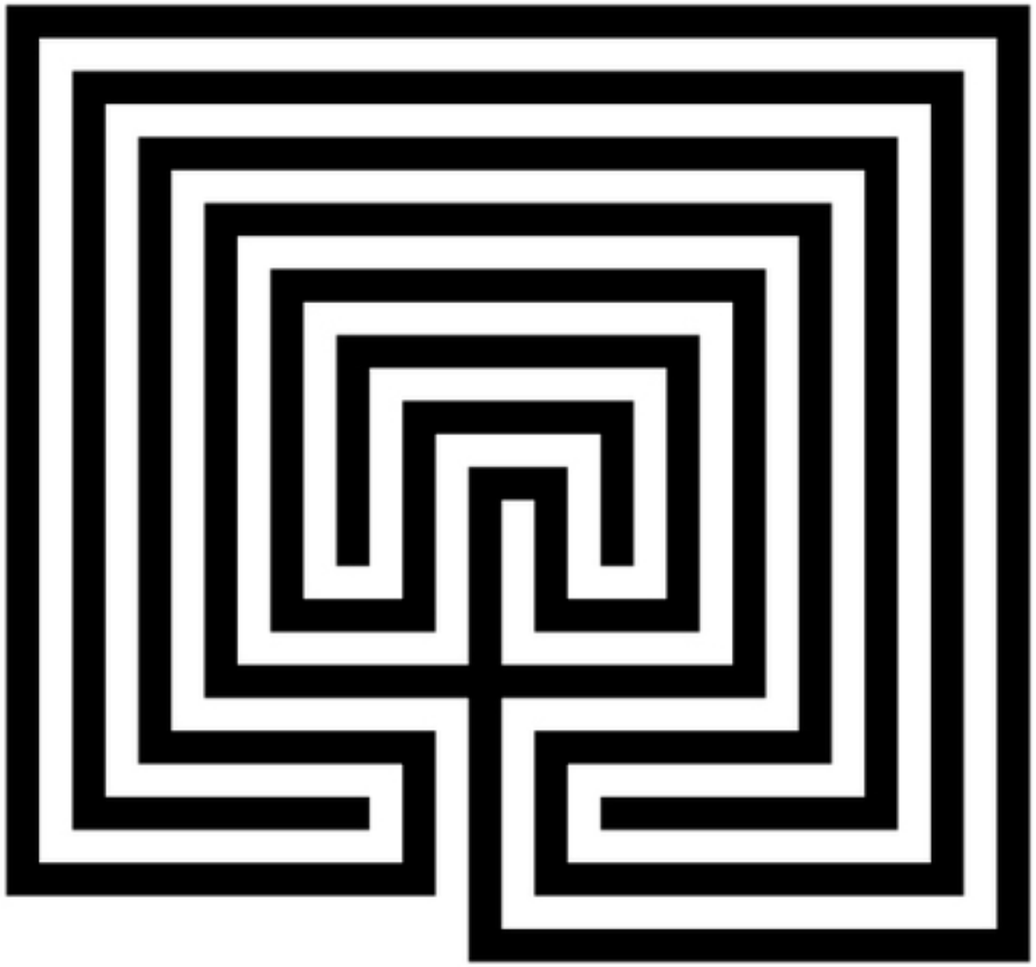
$$\mathcal{H}(p(\mathbf{x})) = -E_{\mathbf{x}\sim p(\mathbf{x})}[\log p(\mathbf{x})]$$



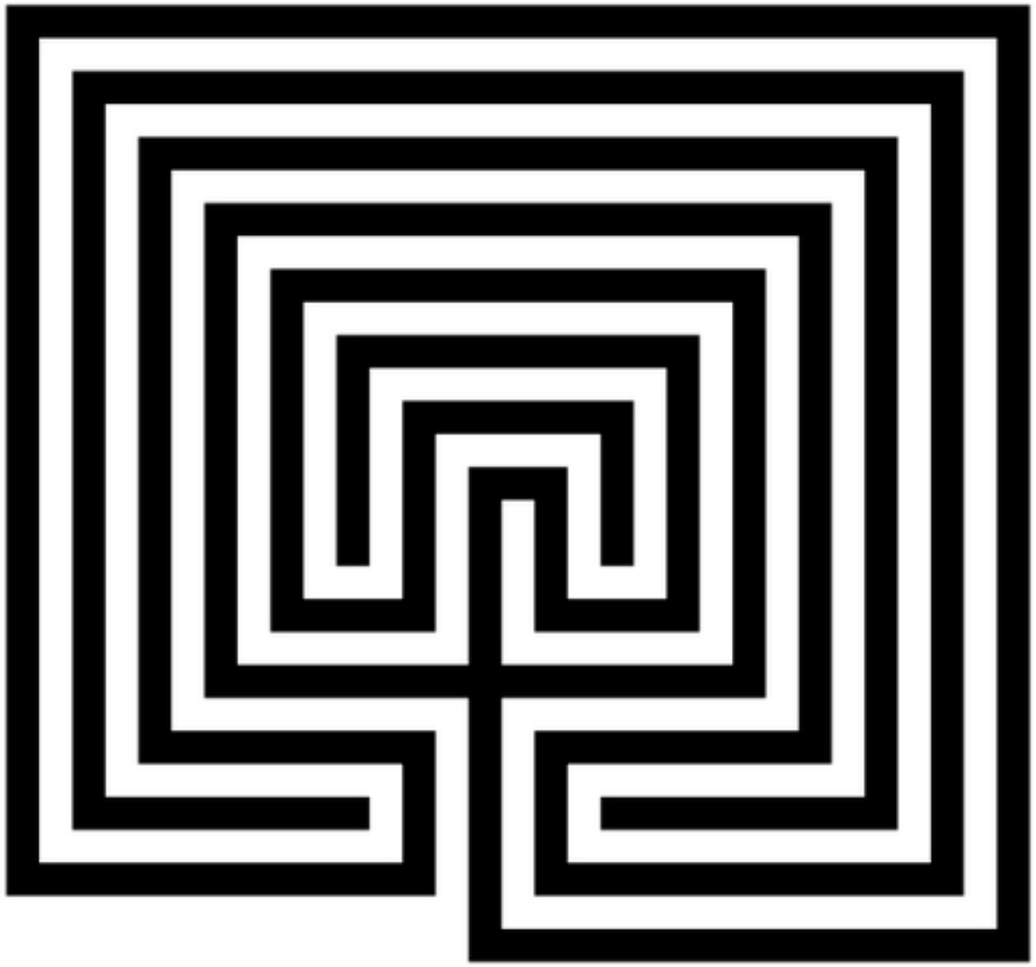


- Dentro da abordagem *MaxEnt* o objetivo é encontrar um comportamento que faça um balanço entre "**compromisso**" e "**adaptabilidade**"

Como explorar um labirinto de forma eficiente?

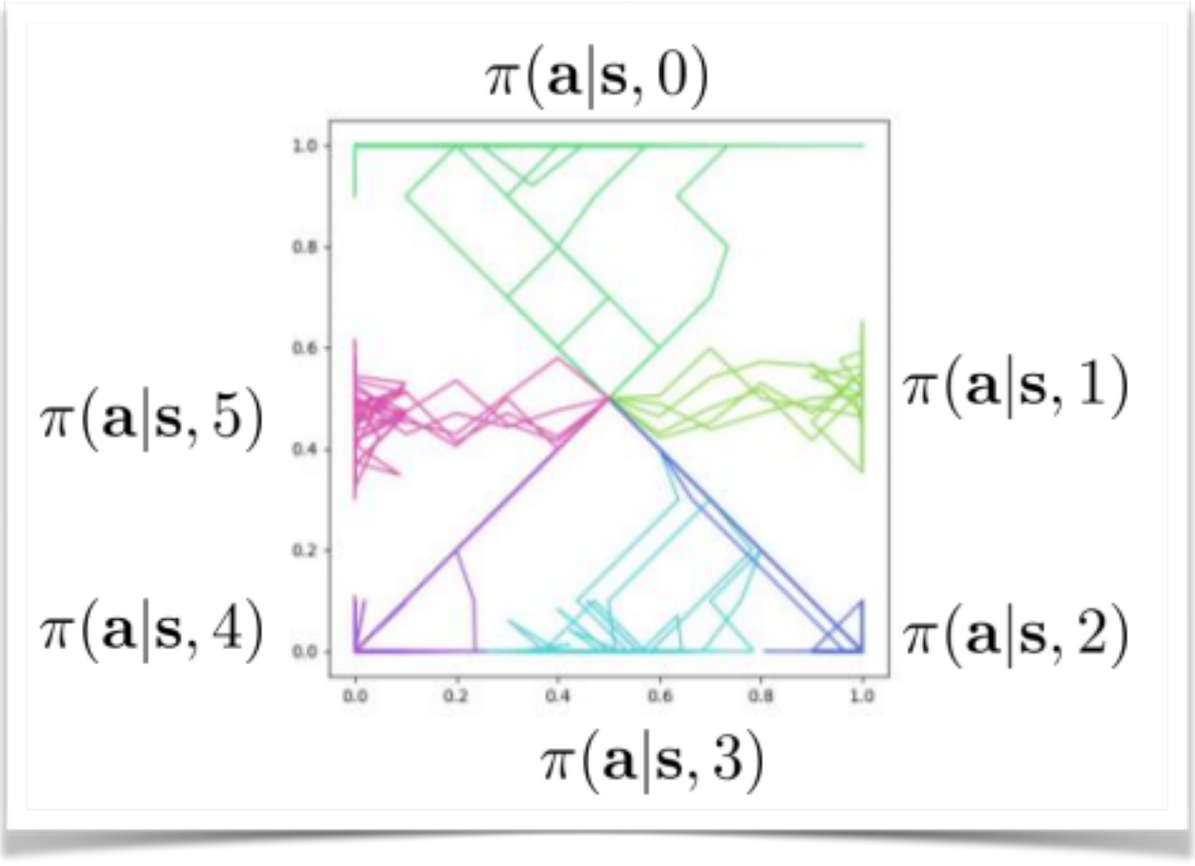


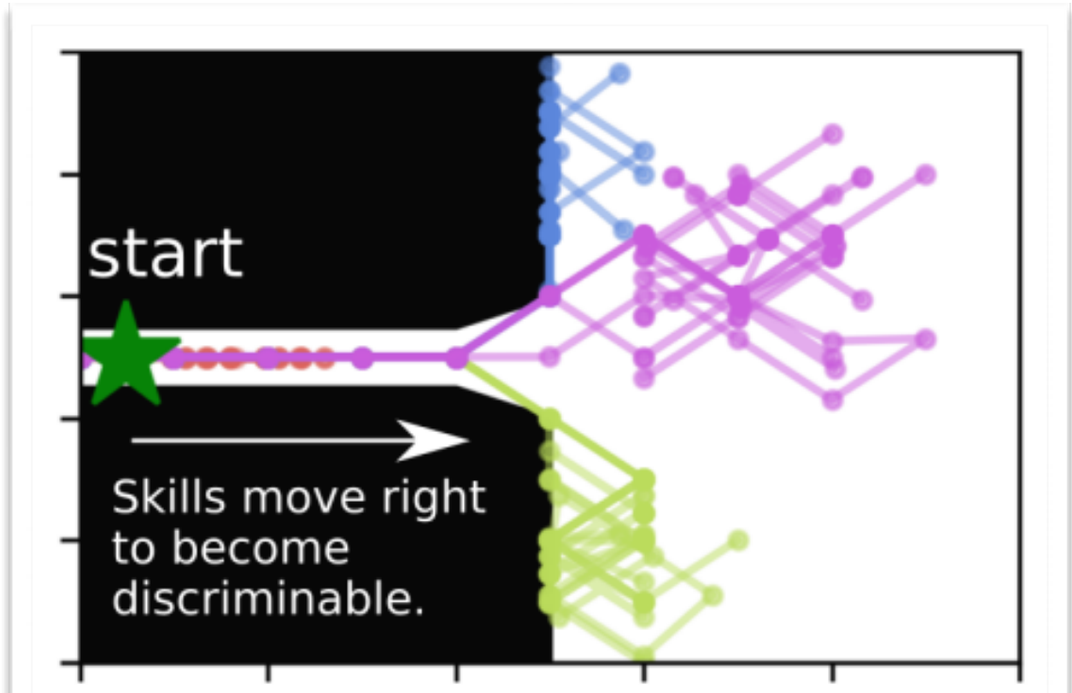
Como explorar um labirinto de forma eficiente?



- **Entropia das ações** não se traduz necessariamente em **entropia da visitação de estados!**

Diversity is All You Need



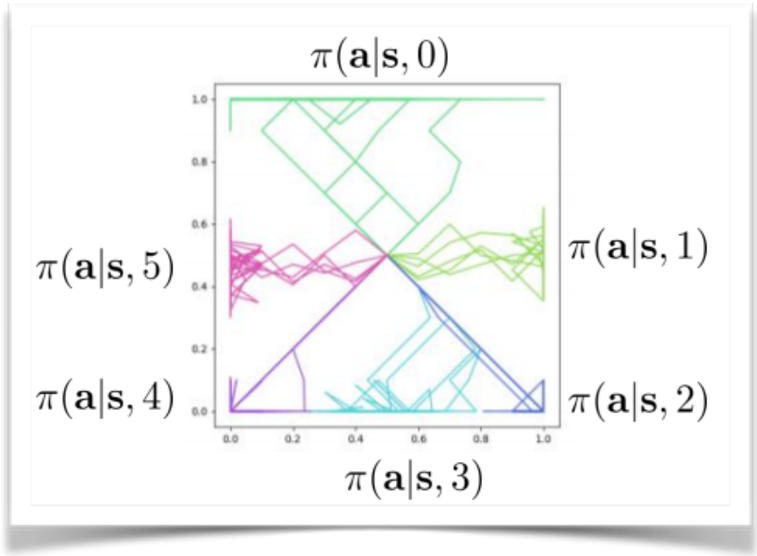


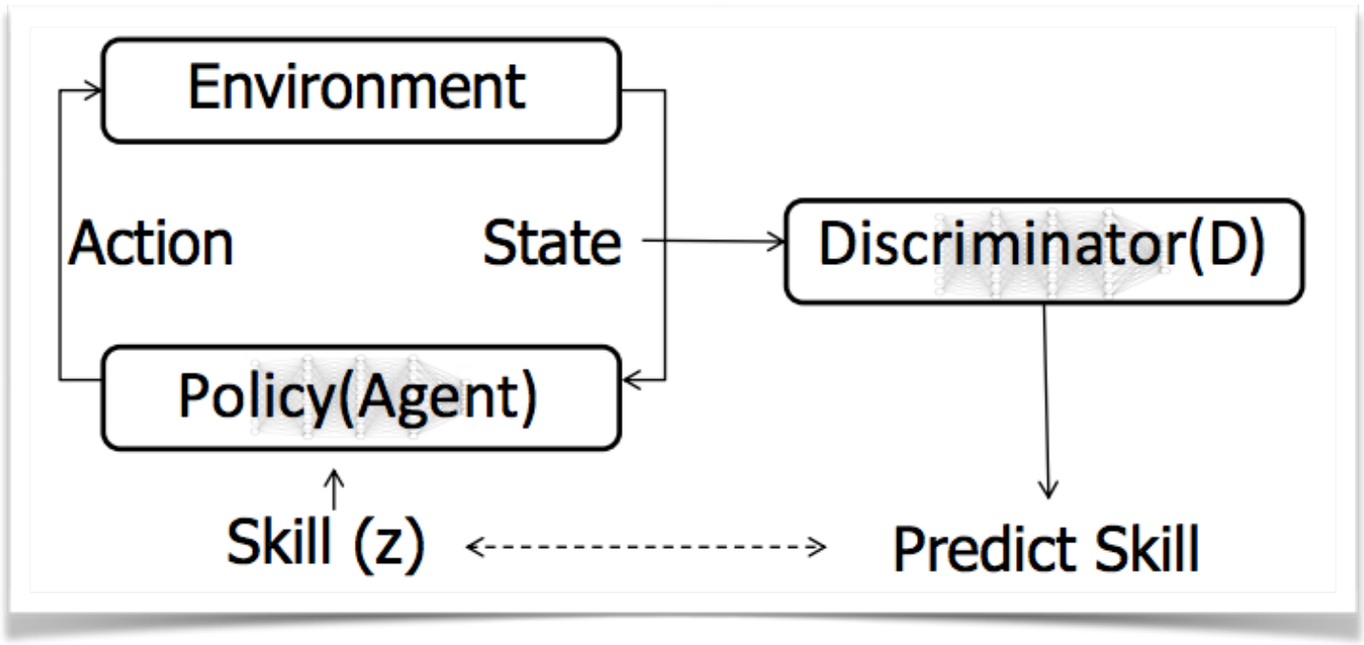
Diversity is All You Need

$$\pi(\mathbf{a}|\mathbf{s}, z) = \arg\max_{\pi} \sum_{z} E_{\mathbf{s}\sim\pi(\mathbf{s}|z)}[r(\mathbf{s}, z)]$$

reward states that are unlikely for other $z' \neq z$

$$r(\mathbf{s}, z) = \log p(z|\mathbf{s})$$





Conclusão

Conclusão

- Aprendizado por Reforço é um área bastante ativa de pesquisa

Conclusão

- Aprendizado por Reforço é um área bastante ativa de pesquisa
- Em teoria, RL pode ser aplicado em um grande número de áreas

Conclusão

- Aprendizado por Reforço é um área bastante ativa de pesquisa
- Em teoria, RL pode ser aplicado em um grande número de áreas
- Grandes resultados experimentais, mas ainda falta entendimento teórico

Conclusão

- Aprendizado por Reforço é um área bastante ativa de pesquisa
- Em teoria, RL pode ser aplicado em um grande número de áreas
- Grandes resultados experimentais, mas ainda falta entendimento teórico

Conclusão

- Aprendizado por Reforço é um área bastante ativa de pesquisa
- Em teoria, RL pode ser aplicado em um grande número de áreas
- Grandes resultados experimentais, mas ainda falta entendimento teórico

- Neste curso introduzimos algumas das ideias fundamentais de RL:

Conclusão

- Aprendizado por Reforço é um área bastante ativa de pesquisa
- Em teoria, RL pode ser aplicado em um grande número de áreas
- Grandes resultados experimentais, mas ainda falta entendimento teórico

- Neste curso introduzimos algumas das ideias fundamentais de RL:
    - **Otimização de Política** baseada em Gradientes

Conclusão

- Aprendizado por Reforço é um área bastante ativa de pesquisa
- Em teoria, RL pode ser aplicado em um grande número de áreas
- Grandes resultados experimentais, mas ainda falta entendimento teórico

- Neste curso introduzimos algumas das ideias fundamentais de RL:
  - **Otimização de Política** baseada em Gradientes
  - Simulação de **Monte-Carlo**; abordagem de "tentativa-e-erro"

Conclusão

- Aprendizado por Reforço é um área bastante ativa de pesquisa
- Em teoria, RL pode ser aplicado em um grande número de áreas
- Grandes resultados experimentais, mas ainda falta entendimento teórico

- Neste curso introduzimos algumas das ideias fundamentais de RL:
  - **Otimização de Política** baseada em Gradientes
  - Simulação de **Monte-Carlo**; abordagem de "tentativa-e-erro"
  - Aproximadores de função com **redes neurais** (política e função Valor)

Conclusão

- Aprendizado por Reforço é um área bastante ativa de pesquisa
- Em teoria, RL pode ser aplicado em um grande número de áreas
- Grandes resultados experimentais, mas ainda falta entendimento teórico

- Neste curso introduzimos algumas das ideias fundamentais de RL:
  - **Otimização de Política** baseada em Gradientes
  - Simulação de **Monte-Carlo**; abordagem de "tentativa-e-erro"
  - Aproximadores de função com **redes neurais** (política e função Valor)
  - **Arquitetura A2C** + *joint loss* + bônus de entropia

Referências

(1) **Deep Reinforcement Learning Doesn't Work Yet**

- https://www.alexirpan.com/2018/02/14/rl-hard.html

(2) **CS 285 - Deep Reinforcement Learning (UC Berkeley)**

- https://www.youtube.com/watch?v=tzieElmtAjs

(3) **Diversity is All You Need:Learning Diverse Skills without a Reward Function**

- https://sites.google.com/view/diayn/home?authuser=0

(4) **Introduction to Deep RL (MIT)**

- https://www.youtube.com/watch?v=zR11FLZ-O9M