INSTITUTO DE MATEMATICA E ESTATÍSTICA
UNIVERSIDADE DE SÃO PAULO

VERÃO
IME-USP

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

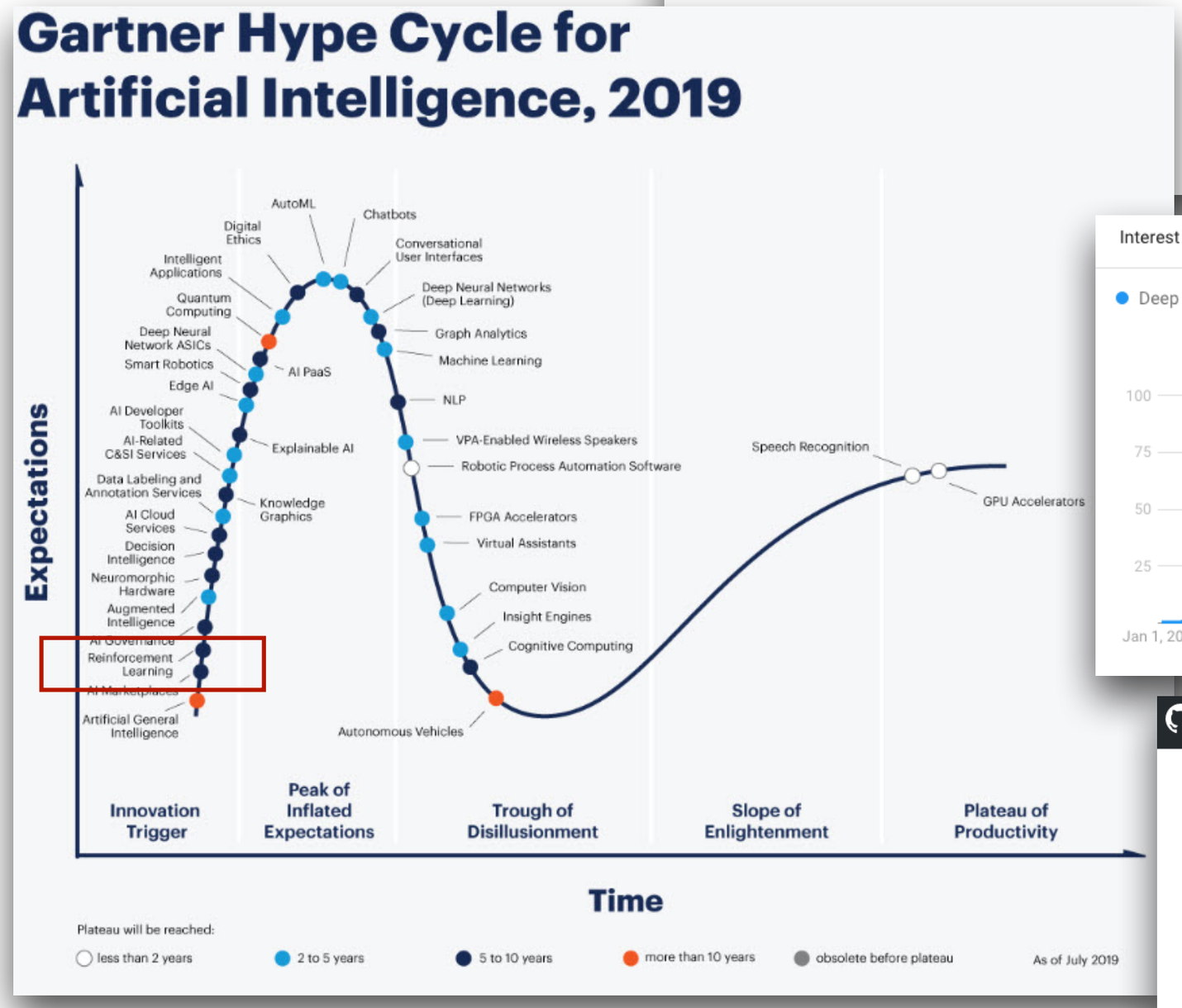
Introdução ao Aprendizado por Reforço

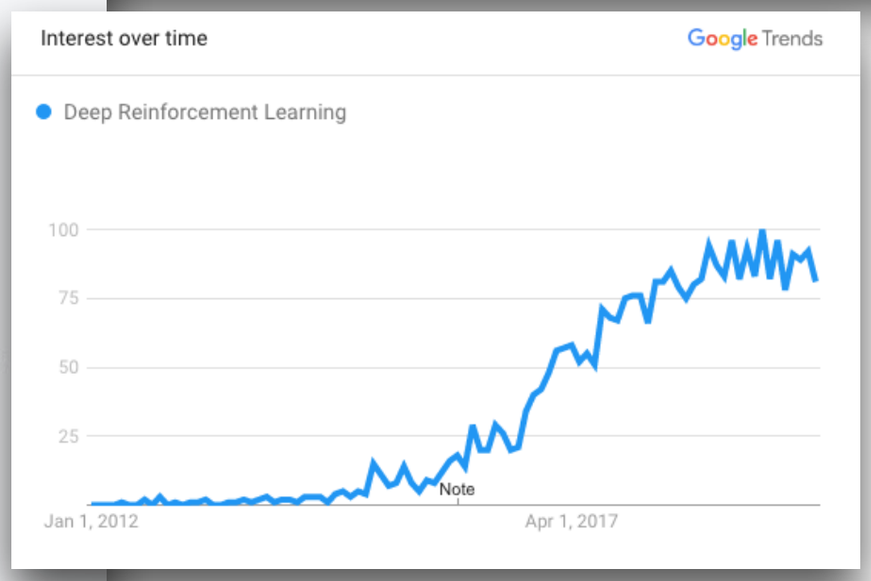
Thiago Pereira Bueno
tbueno@ime.usp.br
IME - USP, 10/02/2019

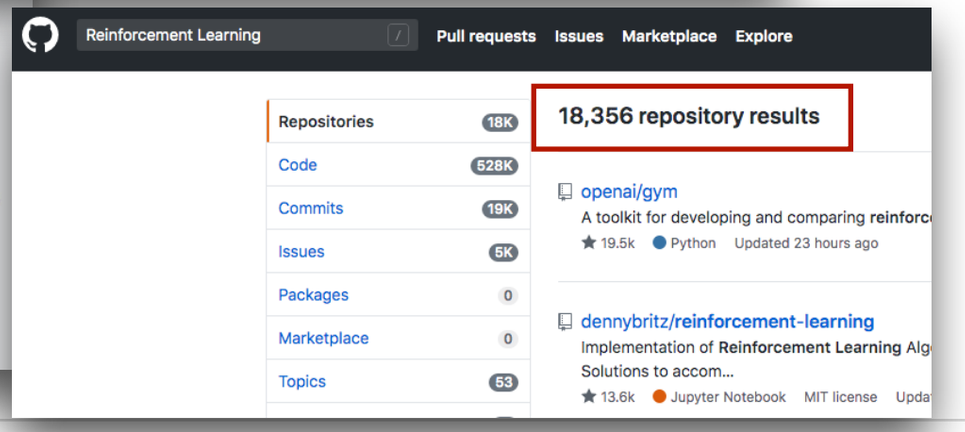
Google's DeepMind AI Just Taught Itself To Walk

https://www.youtube.com/watch?v=gn4nRCC9TwQ

Aprendizado por Reforço em 2019





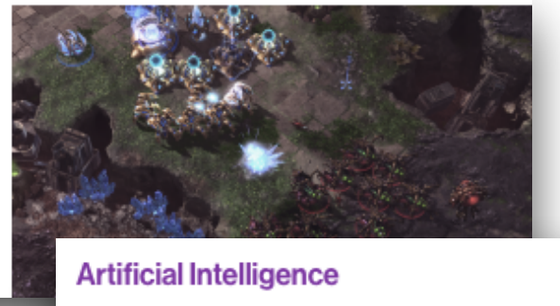


Grandes avanços em 2019 ...

Artificial Intelligence

DeepMind's AI has now outcompeted nearly all human players at StarCraft II

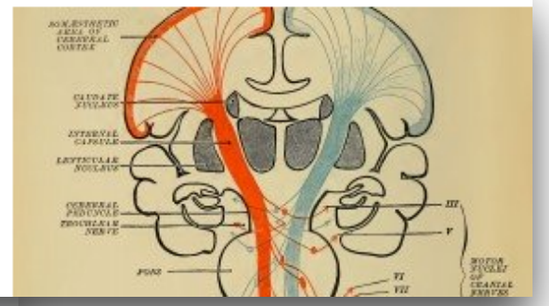
AlphaStar cooperated with itself to learn new strategies for conquering the popular galactic warfare game.



Artificial Intelligence

An algorithm that learns through rewards may show how our brain does too

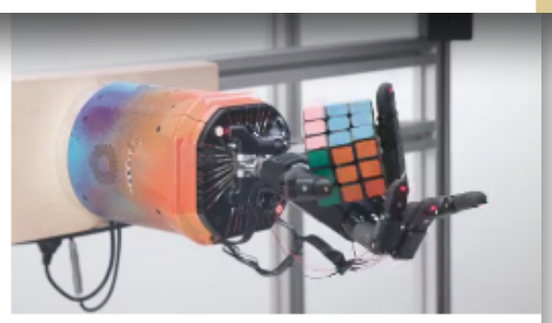
By optimizing reinforcement-learning algorithms, DeepMind uncovered new details about how dopamine helps the brain learn.



Artificial Intelligence

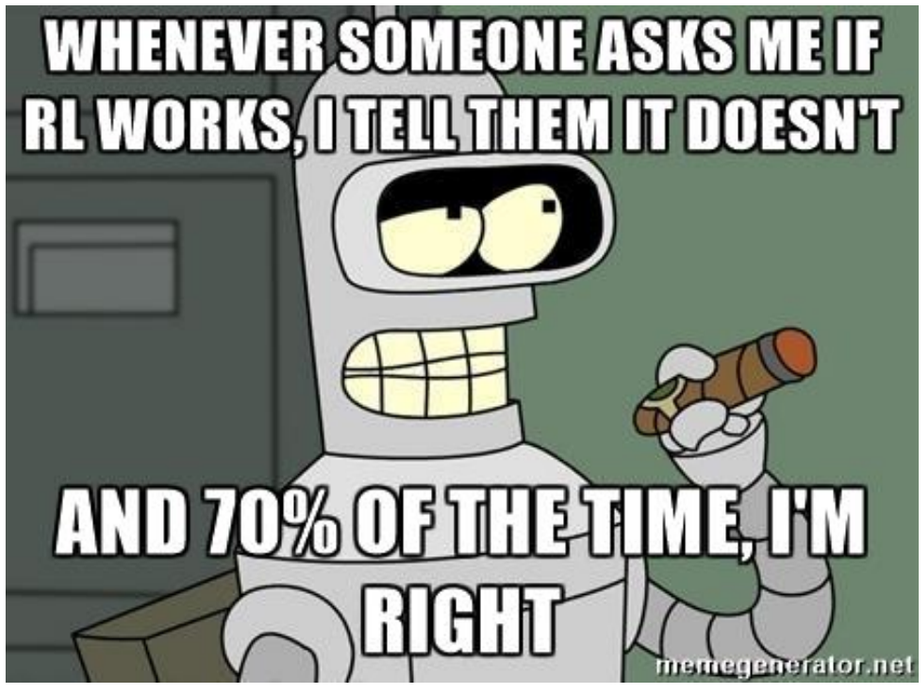
A robot hand taught itself to solve a Rubik's Cube after creating its own training regime

Researchers at OpenAI have developed a new method for transferring complex manipulation skills from simulated to physical environments.



https://www.technologyreview.com/search/?s=Reinforcement+Learning

... mas ainda há muitos desafios pela frente!



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
- *"Overfitting"* ainda é um problema em aberto
- Aprendizado é instável e resultados difíceis de reproduzir

O que esperar desse curso?

- 5 aulas (teoria + prática):
    1. Introdução ao Aprendizado por Reforço (RL)
    2. Policy Gradients (REINFORCE)
    3. Funções Valor e técnicas de redução de variância
    4. Actor-Critic (A2C) e Generalized Advantage Estimation (GAE)
    5. Tópicos Avançados (TBD)

O que esperar desse curso?

- **Parte prática**: OpenAI Gym, TensorFlow 2.0 + Keras, NumPy, Bokeh
  - API do OpenAI Gym
  - Implementação de redes neurais via API de modelos do Keras
  - Treinamento de modelos via diferenciação automática no TensorFlow
  - Monitoramento de experimentos e visualização do desempenho de agentes

Aula 1

Agenda

1. Aprendizado por Reforço (RL) & MDPs
2. *Deep* RL = *Deep Learning* + RL
3. Aproximadores de função em RL
4. *Deep* RL: arcabouço algorítmico

Objetivos

- Familiarizar-se com os objetivos e formato do curso
- Ter uma ideia geral sobre possíveis aplicações de RL
- Aprender os conceitos básicos e vocabulário de RL
- Entender as diferenças entre RL e *Supervised Learning* (SL)

Aprendizado por Reforço: visão geral

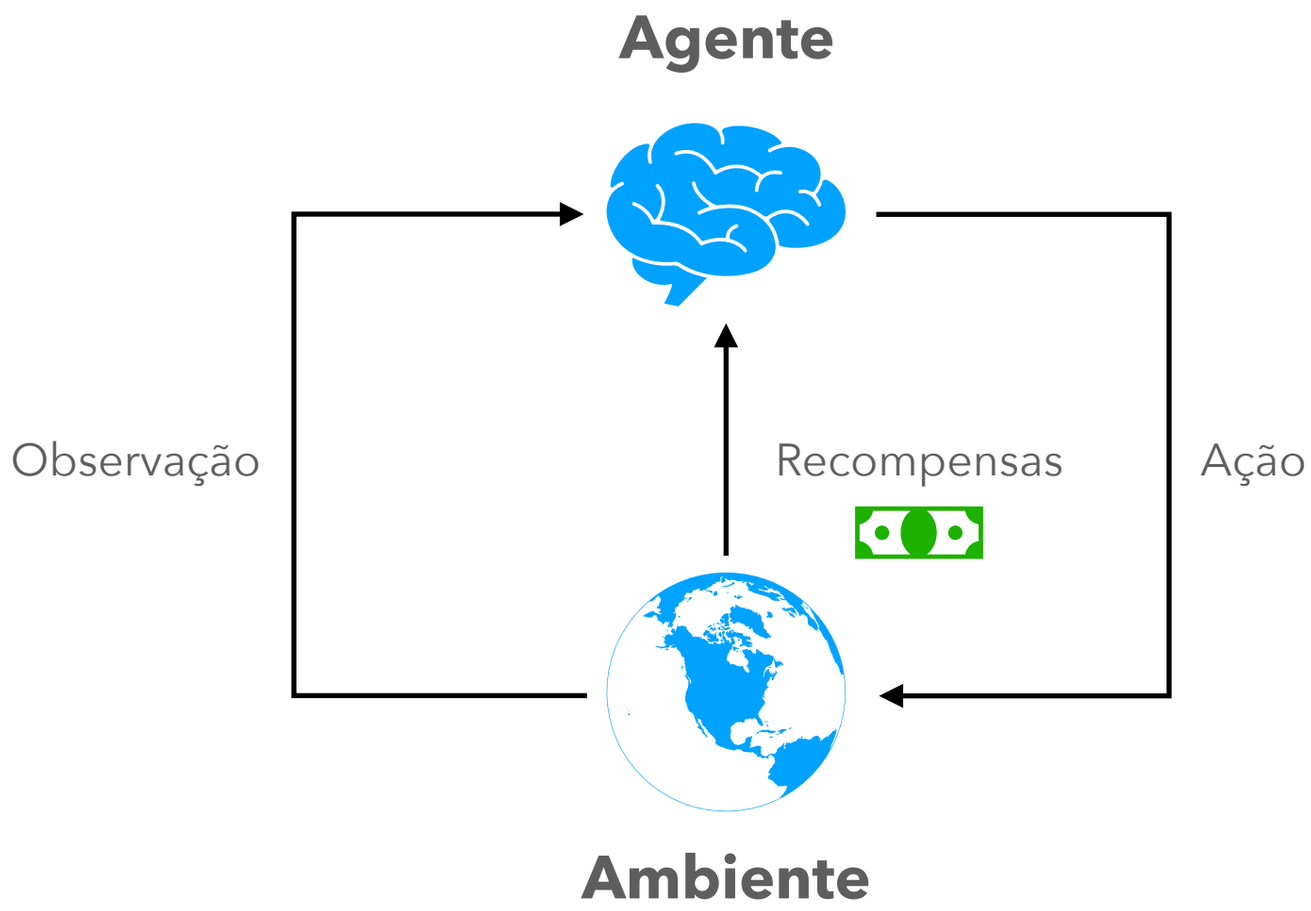
Ciclo de interação Agente-Ambiente

Um agente …

(1) interage com o ambiente;

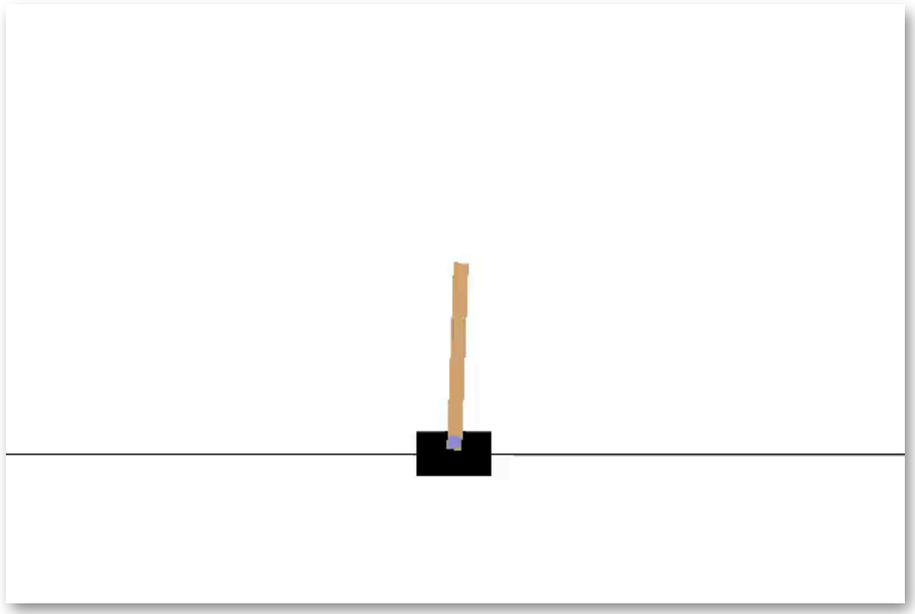
(2) coleta experiências; e

(3) aprende com seus erros e acertos!



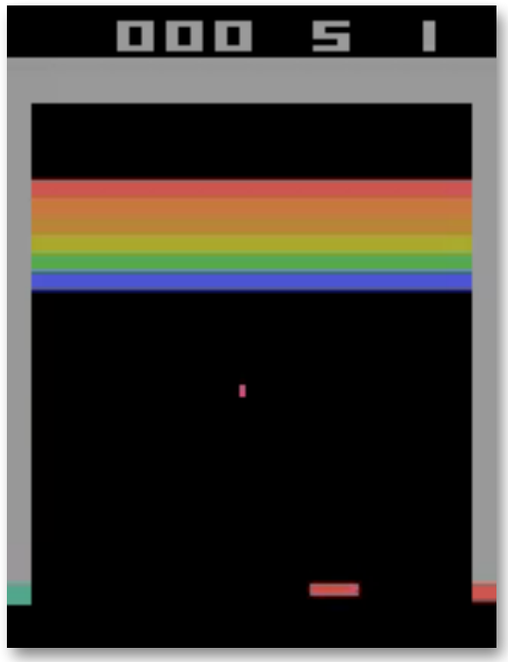
Aprendizado por Reforço: Exemplos (1/2)

CartPole-v1



- **Objetivo:** manter o mastro na vertical por 200 passos
- **Estado:** posição e velocidade (angular) do mastro e do carro
- **Ação:** mover o carrinho para esquerda ou direita
- **Recompensa:** +1 para cada passo que o mastro não cai
- **Término:** o mastro cai (> 12 graus) ou o carro sai da tela
- **Solução:** retorno acima 195 por 100 episódios consecutivos

Breakout-v0



- **Objetivo:** maximizar o *score* do jogo
- **Estado:** image RGB de *shape*=(210, 160, 3)
- **Ação:** número {0, 1, 2, 3}; mover ou não a barra
- **Recompensa:** *score* do jogo (e.g., tijolos quebrados)
- **Término:** jogador perde todas as "vidas"
- **Solução:** maximizar o *score* médio por 100 episódios

Aprendizado por Reforço: Exemplos (2/2)

- Policy gradient methods for **robotics** (Peters and Schall, 2006)
- **Crowdfunding Dynamics Tracking**: A Reinforcement Learning Approach (Wang, Zhang, Liu et al, 2019)
- Developing Multi-Task **Recommendations** with Long-Term Rewards via Policy Distilled Reinforcement Learning (Liu, Li, Xie et al, 2019)
- An Efficient Deep Reinforcement Learning Model for **Urban Traffic Control** (Lin, Dai, Li et al, 2018)
- Universal **quantum control** through deep reinforcement learning (Niu, Boixo, Smelyanskiy et a, 2019)
- Practical Deep Reinforcement Learning Approach for **Stock Trading** (Xiong, Lil, Zhong et al, 2018)
- A REVIEW ON DEEP REINFORCEMENT LEARNING FOR **FLUID MECHANICS** (Garnier, Viquerat, Rabault et al, 2019)
- SquirRL: Automating **Attack Discovery on Blockchain** Incentive Mechanisms with Deep Reinforcement Learning (Hou, Zhou, Ji et al, 2019)
- Which Channel to Ask My Question?: **Personalized Customer Service** Request Stream Routing using Deep Reinforcement Learning (Liu, Long, Lu et al, 2019)

Aprendizado por Reforço e MDPs

Um **Processo de Decisão Markoviano** (*Markov Decision Process)* é definido por:

$$\mathcal{S}, \mathcal{A}, \rho, P, R$$

conjunto de estados
$\mathbf{s}_t \in \mathcal{S}$

conjunto de ações
$\mathbf{a}_t \in \mathcal{A}$

distribuição inicial
$\mathbf{s}_0 \sim \rho(\mathbf{s}_0)$

função de transição
$\mathbf{s}_{t+1} \sim P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$

Função recompensa
$r_{t+1} = R(s_t, a_t)$

Aprendizado por Reforço e MDPs

Tempo discreto $t = 0, 1, 2, ...$

Propriedade de *Markov*

- *"O futuro é independente do passado dado o presente"*

$$P(\mathbf{s}_{t+1} \mid (\mathbf{s}_0, \mathbf{a}_0), (\mathbf{s}_1, \mathbf{a}_1), \ldots, (\mathbf{s}_t, \mathbf{a}_t)) = P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$$

Futuro **Passado** **Presente**

Dinâmica estacionária

- *"Hoje não é diferente de amanhã"*

$$\forall t, t' = 0, 1, \ldots$$
$$P(\mathbf{s}_{t+1} = \mathbf{s}' | \mathbf{s}_t = \mathbf{s}, \mathbf{a}_t = \mathbf{a}) = P(\mathbf{s}_{t'+1} = \mathbf{s}' | \mathbf{s}_{t'} = \mathbf{s}, \mathbf{a}_{t'} = \mathbf{a})$$

Aprendizado por Reforço e MDPs

Trajetória (episódio)

$$\tau_{0:T} = (s_0, a_0, r_1, s_1, a_1, r_2, s_2, \ldots, s_{T-1}, a_{T-1}, r_T, s_T)$$

Retorno

$$R(\tau_{0:T}) = r_1 + r_2 + \cdots + r_T = \sum_{t=0}^{T-1} r_{t+1}$$

Política (estocástica)

$$\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t)$$

Função Objetivo

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

Aprendizado por Reforço: definição de "problema"

Dado um ambiente (modelado por um MDP) $\mathcal{S}, \mathcal{A}, \rho, P, R$

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

Tendo acesso somente a amostras

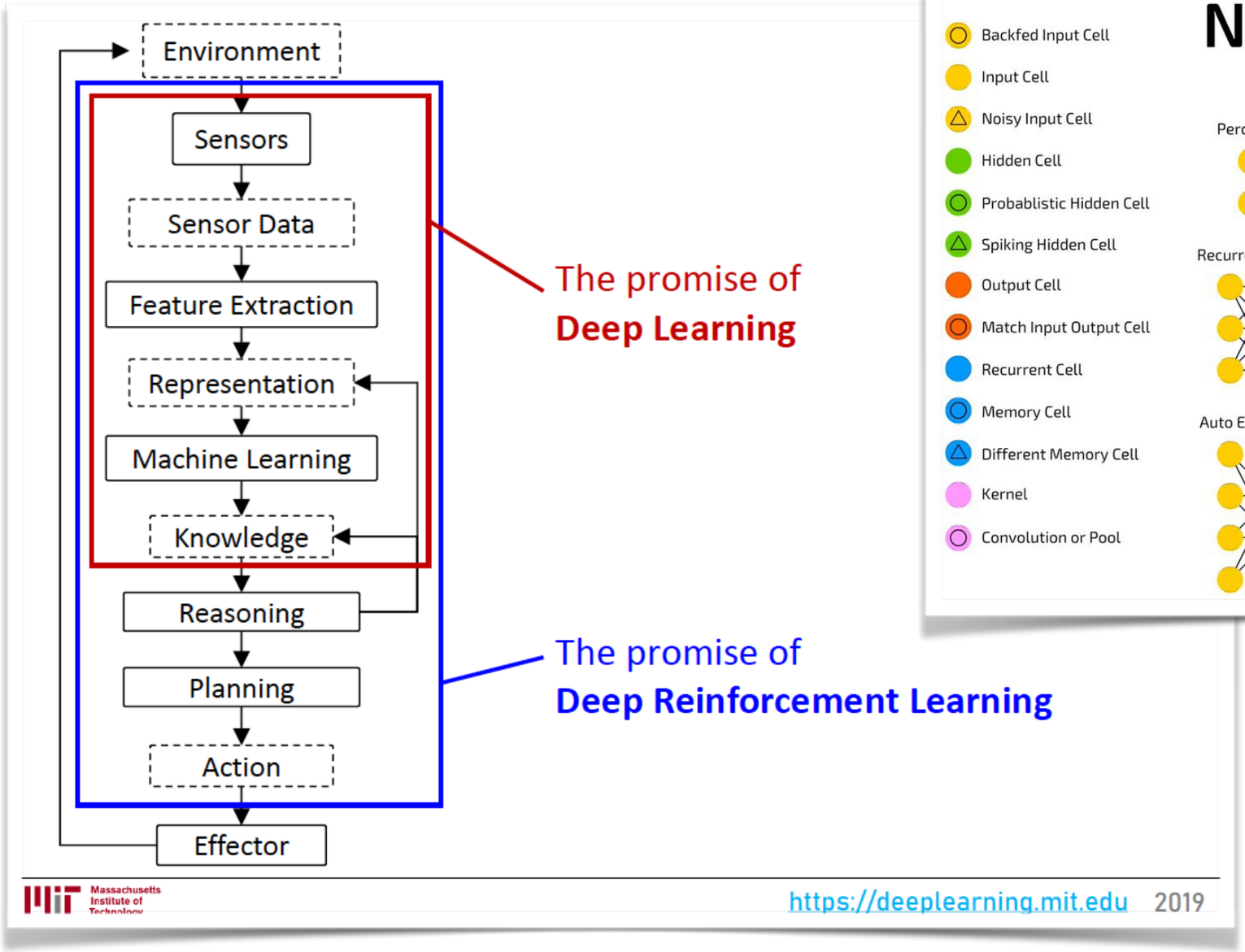
$$\mathbf{s}_0 \sim \rho(\mathbf{s}_0)$$

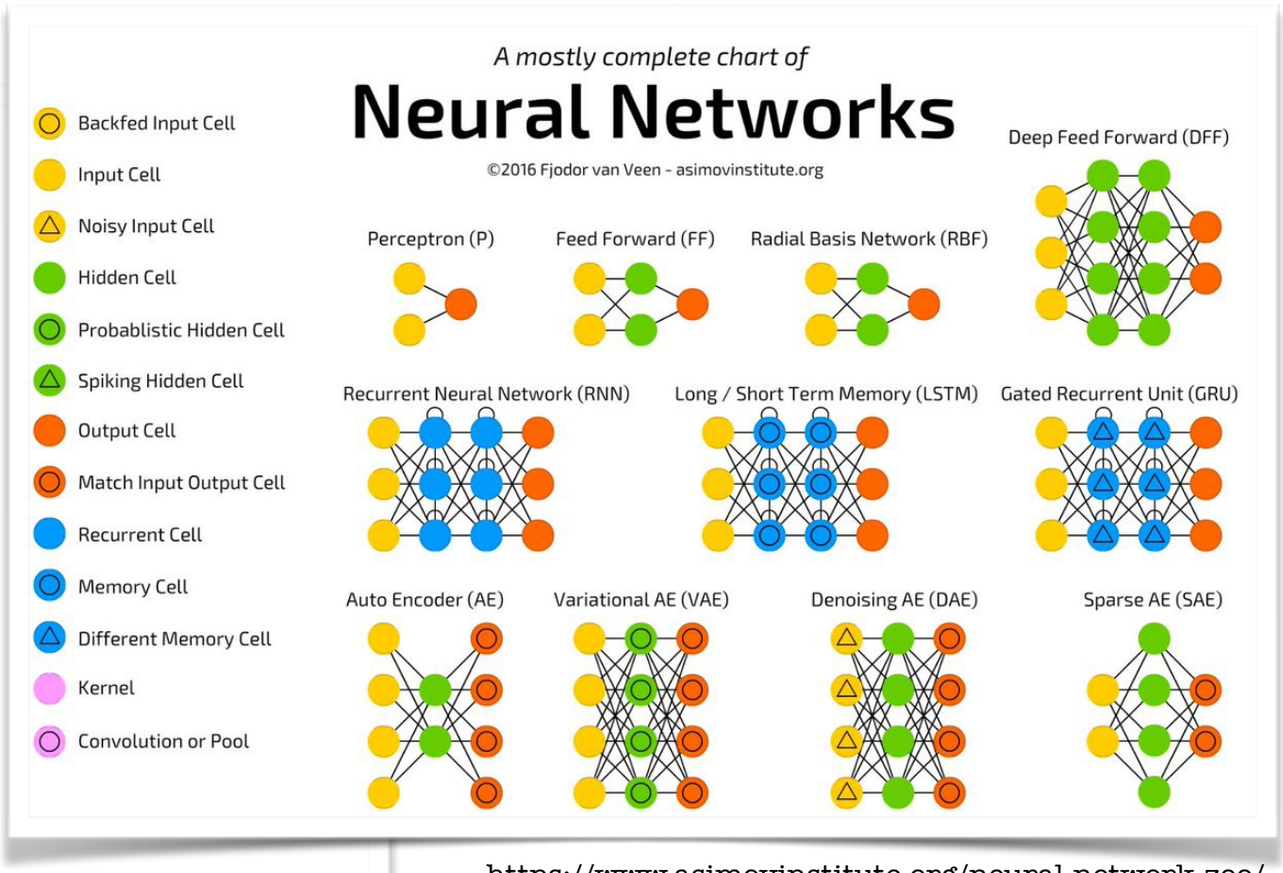
$$\mathbf{s}_{t+1} \sim P(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$$

$$\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t)$$

$$r_{t+1} = R(s_t, a_t)$$

Deep RL = Deep Learning + RL





https://www.asimovinstitute.org/neural-network-zoo/

Deep RL = Deep Learning + RL

Tesla Autopilot - Neural Networks

"Apply cutting-edge research to train deep neural networks on problems ranging from perception to control. Our per-camera networks analyze raw images to perform semantic segmentation, object detection and monocular depth estimation. Our birds-eye-view networks take video from all cameras to output the road layout, static infrastructure and 3D objects directly in the top-down view. Our networks learn from the most complicated and diverse scenarios in the world, iteratively sourced from our fleet of nearly 1M vehicles in real time. ***A full build of Autopilot neural networks involves 48 networks that take 70,000 GPU hours to train*** 🔥*. Together, they output 1,000 distinct tensors (predictions) at each timestep."*

https://www.tesla.com/autopilotAI

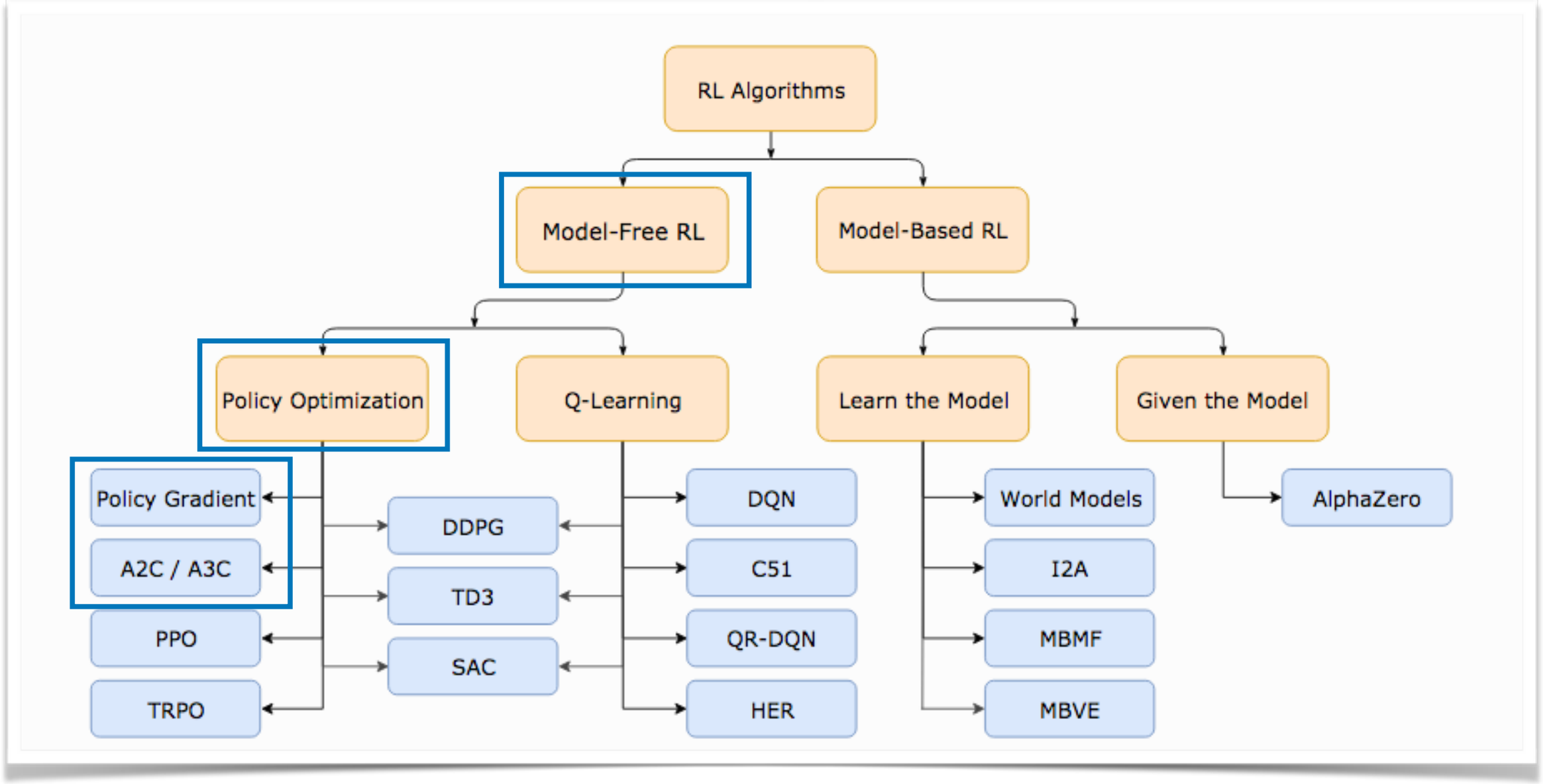
Aproximadores de função em RL

- **Pólitica:** mapeamento entre estados e ações $\pi_\theta(\mathbf{a}_t \,|\, \mathbf{s}_t)$

- **Função Valor:** estimador dos retornos esperados $V_\phi(\mathbf{s}) \approx \mathbb{E}_{\tau\sim\pi}\left[R(\tau) \mid \mathbf{s}_0 = \mathbf{s}\right]$

- **Modelo:** dinâmica do ambiente e/ou recompensas $P_w(\mathbf{s}' \,|\, \mathbf{s}, \mathbf{a}) \approx P(\mathbf{s}' \,|\, \mathbf{s}, \mathbf{a})$

Deep RL Zoo - Taxonomia de Algoritmos de RL



https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html#id20

Aprendizado por Reforço Vs. Aprendizado Supervisionado

- Não há “oráculos” :
    - Sem acesso explícito às respostas certas (i.e., nenhum *target* ou *label* é fornecido)
- *Feedback* esparso e/ou atrasado:
    - Maior parte do tempo o agente recebe pouca informação para melhorar seu desempenho
- Geração de dados:
    - Não há noção clara de “*datasets*”
    - Se a política do agente se altera, a distribuição das experiências do agente também muda

RL: Otimização de Política - Algoritmo

Algoritmo 2 Otimização de Política

Entrada: parâmetros da política θ; taxa de aprendizado α

1: **enquanto** não satisfeito **faça**

2: Colete trajetórias com a política atual, $\tau_1, \tau_2, \ldots, \tau_N \sim \pi_\theta$

3: Calcule os retornos de cada trajetória,

$$R_k = \sum_{t=0}^{T^{(k)}-1} r_t^{(k)}$$

4: Estime o desempenho da política,

$$J(\pi_\theta) \approx \frac{1}{N} \sum_{k=1}^{N} R_k$$

5: Compute gradientes e atualize os parâmetros da política,

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

6: **fim enquanto**

7: **devolve** π_θ

Referências

(1) **Reinforcement Learning: An Introduction** (Sutton & Barto 2018, 2nd Edition)

- Capítulo 1 (http://incompleteideas.net/book/RLbook2018.pdf)

(2) **OpenAI Spinning Up**

- https://spinningup.openai.com/en/latest/spinningup/rl_intro.html

(3) **Challenges of Real-World Reinforcement Learning** (Dulac-Arnold, Mankowitz, and Hester, 2019)

- https://arxiv.org/abs/1904.12901

(4) **Reinforcement Learning Applications** (Li, 2019)

- https://arxiv.org/abs/1908.06973