



EEEP Joaquim Nogueira - Fortaleza

Reinforcement Learning: Como as máquinas aprendem?

Minho Menezes

September 12, 2019



Apresentação



Olá! Eu sou o Minho.

~ Bacharel em Eng. de Computação (UFC, 2019)

~ Mestrando em Eng. de Teleinformática (UFC, 2019–)

Sou um pesquisador com especialidade em **Aprendizagem de Máquina** e **Teoria do Controle**.

Contatos:



[minhotmog@gmail](mailto:minhotmog@gmail.com)



[@tiominho](https://github.com/tiominho)



[@katchau](https://t.me/katchau)

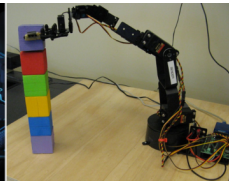
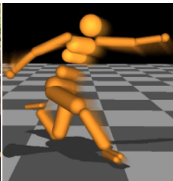
Nessa palestra, iremos conversar sobre...

Reinforcement Learning

⇒ Ramo da **I.A.** que lida com o controle autônomo de *sistemas físicos reais*.

Como programamos
máquinas para realizar
atividades que
exigem inteligência?

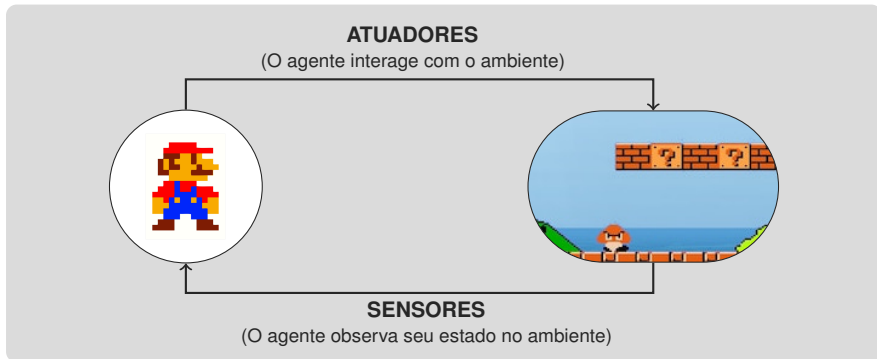
Como ensinar
máquinas a realizar
uma atividade
específica?





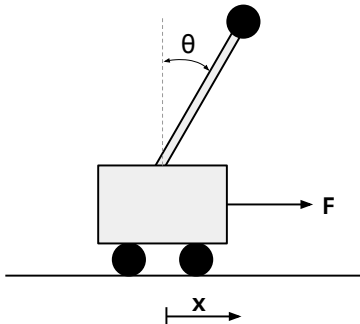
Agentes Inteligentes

O elemento principal é o **Agente Inteligente**. (Também conhecido como "Sistema Dinâmico")



- ~> Está sempre inserido em um **Ambiente** que influencia um no outro.
- ~> Possui um conjunto de **ações** e **estados** possíveis.
- ~> Regidos por **leis bem definidas** (*leis da física, regras de jogo, etc*).

Exemplo: Pêndulo Invertido (ou "Cartpole")



~> **Ações:** [\Leftarrow , \Rightarrow]

~> **Estados:** [θ , x]

Spoiler: vamos ver o Pêndulo aprendendo a se controlar jajá



Aprendizagem do Agente



O nosso Agente é **ONISCIENTE**.

CONTROLE POR PLANEJAMENTO

1. Informamos ao Agente seu **estado de partida** e o seu **estado objetivo**
2. O Agente calcula uma **trajetória** \rightsquigarrow (uma série de ações)
3. O Agente começa a aplicar sua **Lei de Controle** e recalcula a trajetória se houver algum desvio do que foi planejado

Vantagens:

- ▶ Controle da melhor qualidade possível
- ▶ Algoritmos rápidos e implementáveis em chip

Desvantagens:

- ▶ Necessidade de **Modelos Dinâmicos**
- ▶ Altamente complexo para o programador

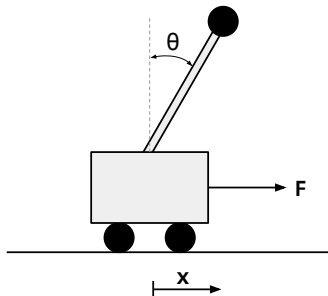
Alguns algoritmos...

\rightsquigarrow LQR, LQG, MPC, NMPC, iLQR, LoopShaping, DynamicProgramming, SMC, IPM, ...



Modelos Dinâmicos não são simples...

na verdade, essa é uma das maiores áreas de TODA a ciência



Modelo:

$$\begin{cases} \ddot{x} = \frac{2a}{2 - m\cos^2(\theta)} \left(ml\dot{\theta}^2 \sin(\theta) - 0.5mg\cos(\theta)\sin(\theta) + u - f_c(\dot{x}) + \frac{1}{2l}\cos(\theta)d_{Mf}\dot{\theta} \right) \\ \ddot{\theta} = \frac{1}{2l - ml\cos^2(\theta)} \left(g\sin(\theta) - mla\dot{\theta}\cos(\theta) + a\cos(\theta)(u - f_c(\dot{x})) - \frac{d_{Mf}\dot{\theta}}{ml} \right) \end{cases}$$

Ação->Estado: $x(t) = e^{At}x(t_0) + \int_0^t e^{A(t-\tau)}Bu(\tau)d\tau$



O nosso Agente é **MOTIVADO**.

CONTROLE POR RECOMPENSA

1. Criamos um **Sistema de Recompensas** para o Agente
2. O Agente verifica seu estado e...
 - ▶ (*Exploração*) Tenta fazer uma ação que tenha **boa recompensa**, ou
 - ▶ (*Exploitação*) Faz qualquer ação **aleatória**
3. Se a ação der recompensa, memoriza a relação (Estado \rightarrow Ação) como boa.
4. Repete o processo até o Agente aprender todas as relações, ou você perder a paciência.

Vantagens:

- ▶ Não precisa de modelos, só de treino
- ▶ Simula a forma como seres humanos aprendem

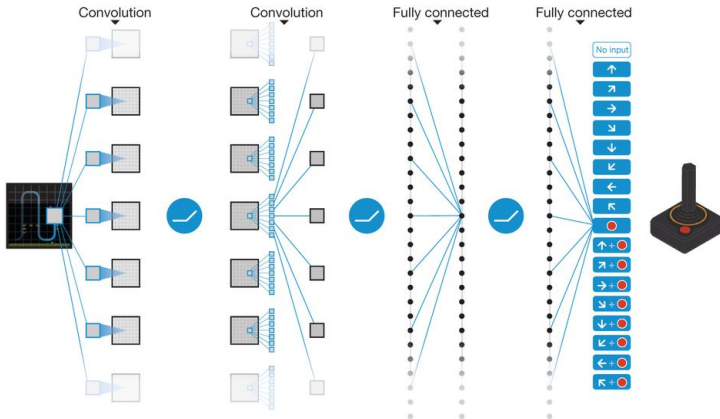
Desvantagens:

- ▶ Treinamento é demorado e caro
- ▶ O resultado dificilmente será perfeito

Alguns algoritmos...

↪ Q-Learning, TD, SARSA, ADP, E2ERL, Policy Gradients, DeepQN, ...

Redes Neurais: codificação biológica de (*Atuadores, Sensores*)



! Showcase !

Muito obrigado!



Perguntas?