Simulated Humanoid Robot Control With Reinforcement Learning

Luis Guilherme G. Aguiar[1], Takashi Yoneyama [1] e Marcos R. O. A. Maximo [2]

[1]Divisão de Engenharia Eletrônica, Instituto Tecnológico de Aeronáutica (ITA)
[2]Laboratório de Sistemas Computacionais Autônomos (LAB-SCA), Divisão de Ciência da Computação (IEC), Instituto Tecnológico de Aeronáutica (ITA)

Exame de Tese, 29/06/2018

Roteiro

Roteiro

Motivação

- Usando DQN, uma IA aprendeu a jogar 49 jogos Atari.
- Paper na Nature (Mnih et al. 2015)

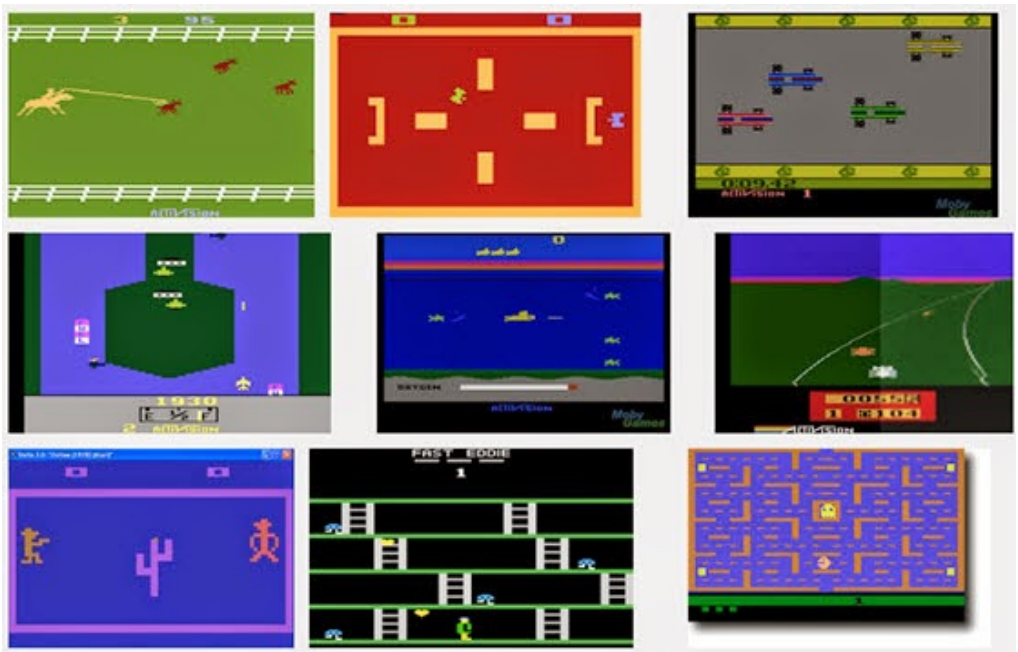

Figura 1: DQN e diferentes jogos Atari.

Motivação

- No mesmo ano, AlphaGo derrotou Lee Sedol, campeão mundial de Go.
- Em 2017, (Silver et al. 2017) introduziu Alpha Go zero.

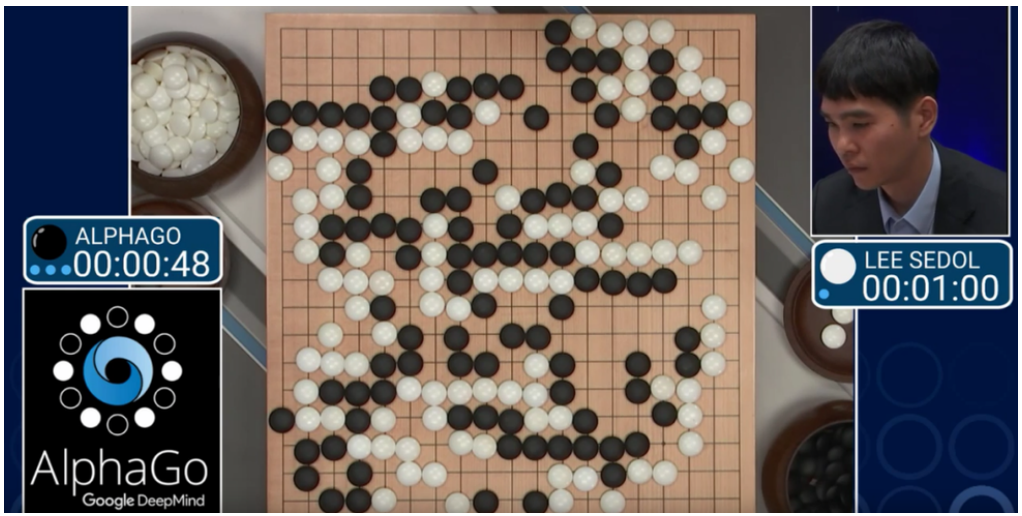

Figura 2: AlphaGo contra Lee Sedol.

Motivação

- Controle de movimento é um grande desafio para IA.
- Em 2018, (Peng et al. 2018) introduziu Deep Mimic.
- Aprendizado de diversos movimentos em ambiente simulado (correr, chutar, acrobacias, etc)

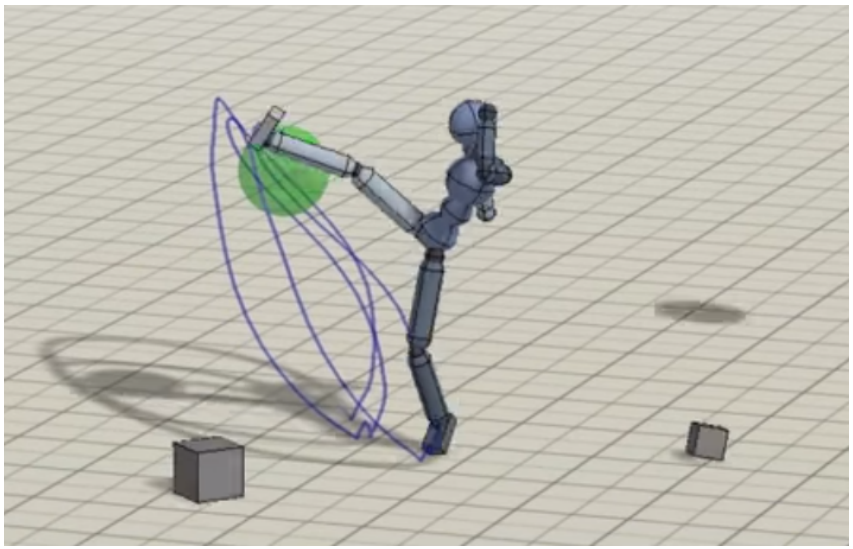

Figura 3: Humanoid chutando em DeepMimic.

- RoboCup 3D Soccer Simulation

Figura 4: Símbolo da Robocup e uma partida da liga Soccer 3D.

Problema

- Chutar uma bola a uma distância final do agente planejada.
- Aprender um comportamento de baixo nível completo.
- Input: estado do agente. Output: sinais de controle para as juntas.

Abordagem

- Técnicas recentes de Deep Reinforcement Learning.
- Inicialmente, aprender por imitação (Peng et al. 2018).
- Curriculum learning (Bengio et al. 2009).
- Fornecer a distância planejada como entrada da política. Treinar para diversos valores.

Revisão Literária

- Técnicas recentes, como DDPG (Lillicrap et al. 2015), TRPO (Schulman et al. 2015), PPO (Schulman et al. 2017).
- Trabalho passado nesse problema: (Abdolmaleki et al. 2016).
- Ainda faz uso de um movimento keyframe. Utilizou CREPS-CMA para aprender $\pi(\theta|s)$.
- (Muzio, Yoneyama e Máximo 2017) desenvolveu comportamentos de alto nivel para movimentos de drible e roubo de bola no SS3D usando DRL.

Roteiro

- Interagir com o ambiente, obter recompensas, aprender.

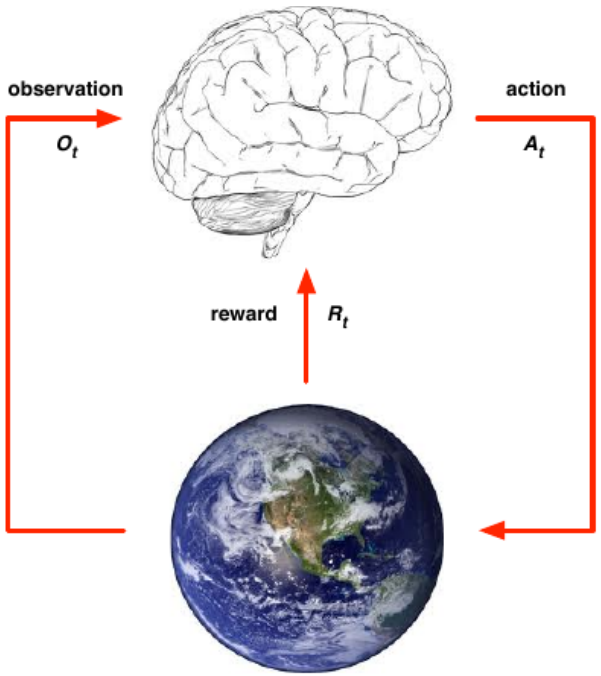

Figura 5: Agente interagindo com o ambiente.

Markov Decision Processes

- Cadeia de Markov $P(S_{t+1}|S_t) = P(S_{t+1} \mid S_1, ..., S_t)$
- Tupla $< \mathbf{S}, \mathbf{A}, P, R, \gamma >$
  - **S** é um conjunto finito de estados que um agente pode assumir.
  - **A** é um conjunto finito de ações que um agente pode tomar.
  - P é a função probabilidade de transição para o estado, representado por $P^a_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$.
  - R é a função de recompensa imediata, com valor esperado dado por $R^a_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$.
  - γ é o fator de disconto, tal que $\gamma \in [0,1]$.

Outros Conceitos

- Retorno total $G_t = R_{t+1} + \gamma R_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$
- Política $\pi(a \mid s) = \mathbb{P}[A_t = a | S_t = s]$
- State value function $v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s]$
- Action value function $q_\pi(s,a) = \mathbb{E}_\pi(G_t \mid S_t = s, A_t = a)$

Algoritmos Clássicos

- Métodos Monte Carlo.
- Temporal-Difference e Sarsa
- Policy Search
- Actor Critic
- (Sutton e Barto 1998)

Roteiro

- Em espaço de ações contínuo, precisamos estimar value functions complexas e não-lineares.

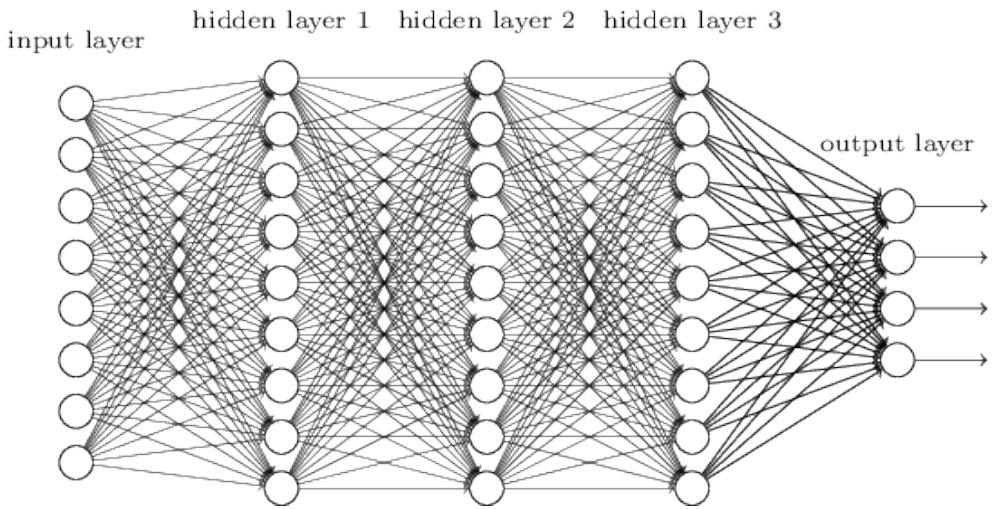

Figura 6: Deep Neural Network.

- Técnicas consagradas.

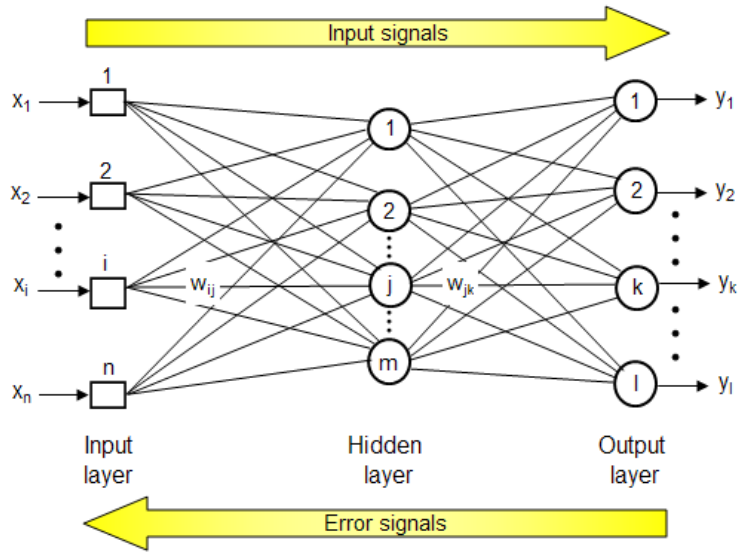

Figura 7: Forward and Backward propagation.

- Otimizações: Regularization, RMSprop, Adam Optimizer

Roteiro

- Representa $\pi_\theta(a|s)$ como uma distribuição $\mathcal{N}(\mu,\sigma)$. μ e σ output da rede neural.
- Vulnerabilidde ao tamanho do passo de atualização de métodos tradicionais em policy gradient.
- Restringir o tamanho do passo a uma região de confiança

$$\underset{\theta}{\text{maximize}}\ \hat{\mathbb{E}}_{s_t,a_t\sim\pi_{\theta_{old}}}\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] \tag{1}$$

$$\text{subject to}\ \hat{\mathbb{E}}_{s_t,a_t\sim\pi_{\theta_{old}}}\left[KL[\pi_{\theta_{old}},\pi_\theta(.|s_t)]\right] \leq \delta \tag{2}$$

- Ainda podemos manter essa restrição, mas através de uma implementação mais simples.
- Otimização de primeira ordem, com uma função objetivo modificada.

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \tag{3}$$

$$L(\theta) = \hat{\mathbb{E}}_t \left[min(r_t(\theta)\hat{A}_t, clip(r_t(\theta),1-\epsilon,1+\epsilon)\hat{A}_t) \right] \tag{4}$$

Roteiro

Ambiente e Ferramentas

- Simulador SimSpark (SimSpark 2004)
- C++
- Python

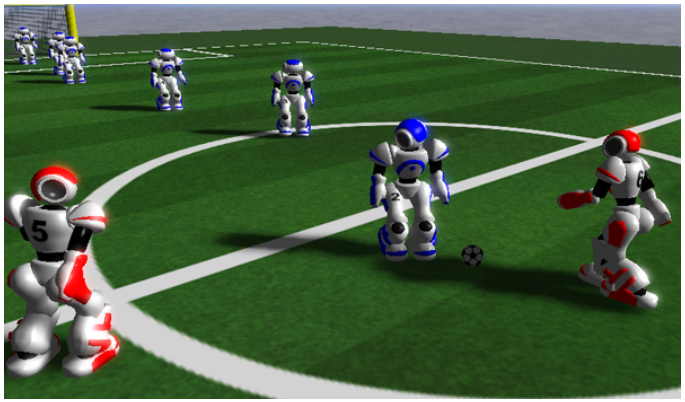

Figura 8: Simulador SimSpark.

Figura 9: Logo TensorFlow.

Figura 10: Logo OpenAI.

- Implementada em (Muzio, Yoneyama e Máximo 2017).

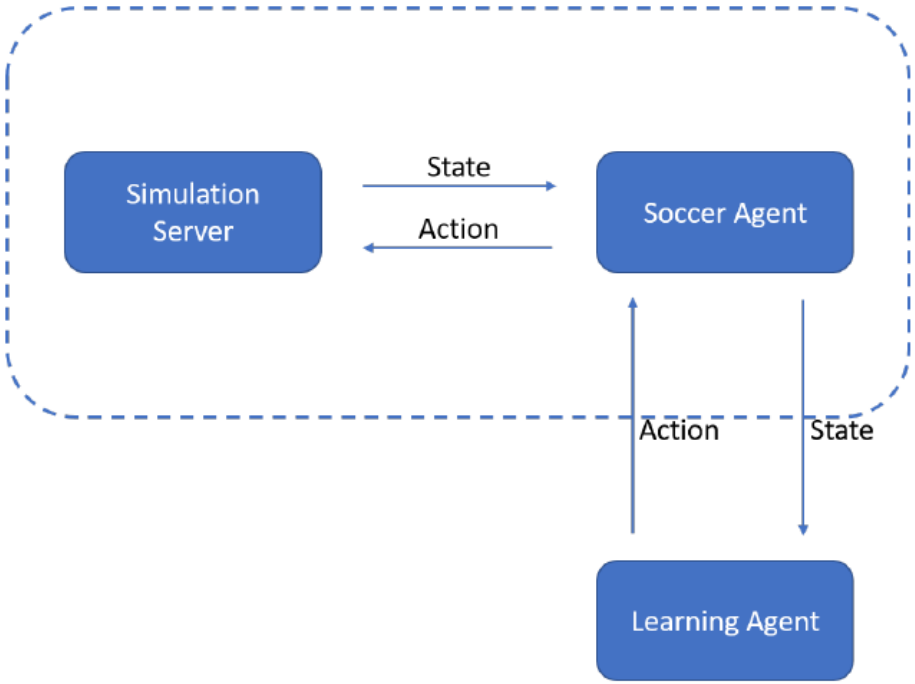

Figura 11: Arquitetura a ser usada.

Roteiro

Feito

- Aprender aprendizado supervisionado.
- Aprender aprendizado por reforço.
- Revisão bibliográfica.
- Implementar técnicas tradicionais de RL para alguns problemas simples de controle da OpenAI gym.

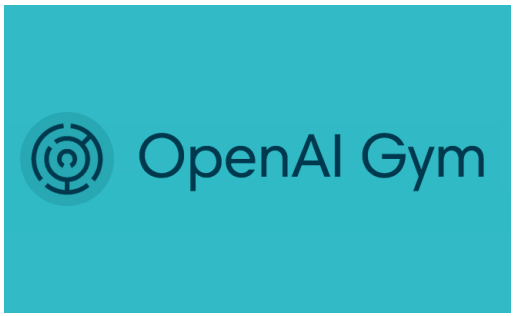

Figura 12: Logo OpenAI Gym.

Problemas Gym

- CartPole com Deep Q-Networks.

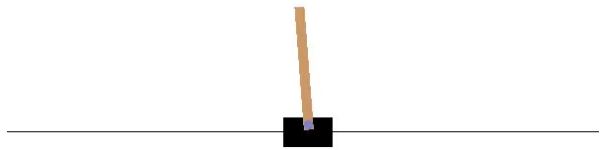

Figura 13: Cart Pole Gym.

- MountainCar com policy gradient.

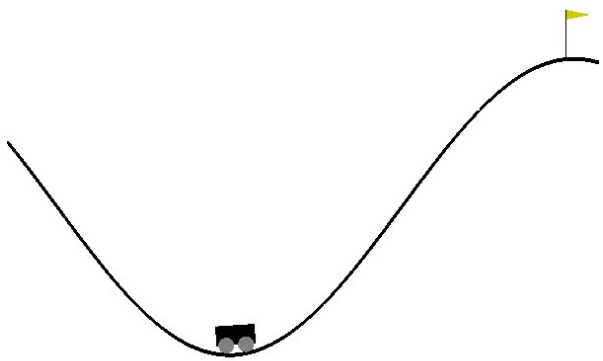

Figura 14: Mountain Car Gym.

Trabalhos Futuros

- Aprender um behavior que imita o movimento Keyframe atual para o chute.
- Inputar uma distância fixa para a política e aprender a alcançar essa distância.
- Aprender para várias distâncias num range discreto.

Bibliografia I

- Abdolmaleki, Abbas et al. (2016). "Learning a Humanoid Kick With Controlled Distance". Em: *20th RoboCup International Symposium, Leipzig, Germany, July 2016*.
- Bengio, Yoshua et al. (2009). "Curriculum Learning". Em: *Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. Montreal, Quebec, Canada: ACM, pp. 41–48. ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553380. URL: http://doi.acm.org/10.1145/1553374.1553380.
- Lillicrap, Timothy P. et al. (2015). "Continuous control with deep reinforcement learning". Em: *CoRR* abs/1509.02971. URL: http://arxiv.org/abs/1509.02971.
- Mnih, Volodymyr et al. (2015). "Human-level control through deep reinforcement learning". Em: *Nature* 518.7540. Letter, pp. 529–533. ISSN: 0028-0836. URL: http://dx.doi.org/10.1038/nature14236.

Bibliografia II

- Muzio, Alexandre, Takashi Yoneyama e Marcos Máximo (2017). "Deep reinforcement learning applied to humanoid robots". (In Portuguese). Bachelor's Thesis. São José dos Campos, SP, Brazil: Instituto Tecnológico de Aeronáutica.
- Peng, Xue Bin et al. (2018). "DeepMimic: Example-Guided Deep Reinforcement Learning of Physics-Based Character Skills". Em:
- Schulman, John et al. (2015). "Trust Region Policy Optimization". Em: *CoRR* abs/1502.05477. arXiv: 1502.05477. URL: http://arxiv.org/abs/1502.05477.
- Schulman, John et al. (2017). "Proximal Policy Optimization Algorithms". Em: *CoRR* abs/1707.06347. arXiv: 1707.06347. URL: http://arxiv.org/abs/1707.06347.
- Silver, David et al. (2017). "Mastering the game of Go without human knowledge". Em: *Nature* 550.

Bibliografia III

Sutton, Richard S. e Andrew G. Barto (1998). *Introduction to Reinforcement Learning*. 1st. Cambridge, MA, USA: MIT Press. ISBN: 0262193981.