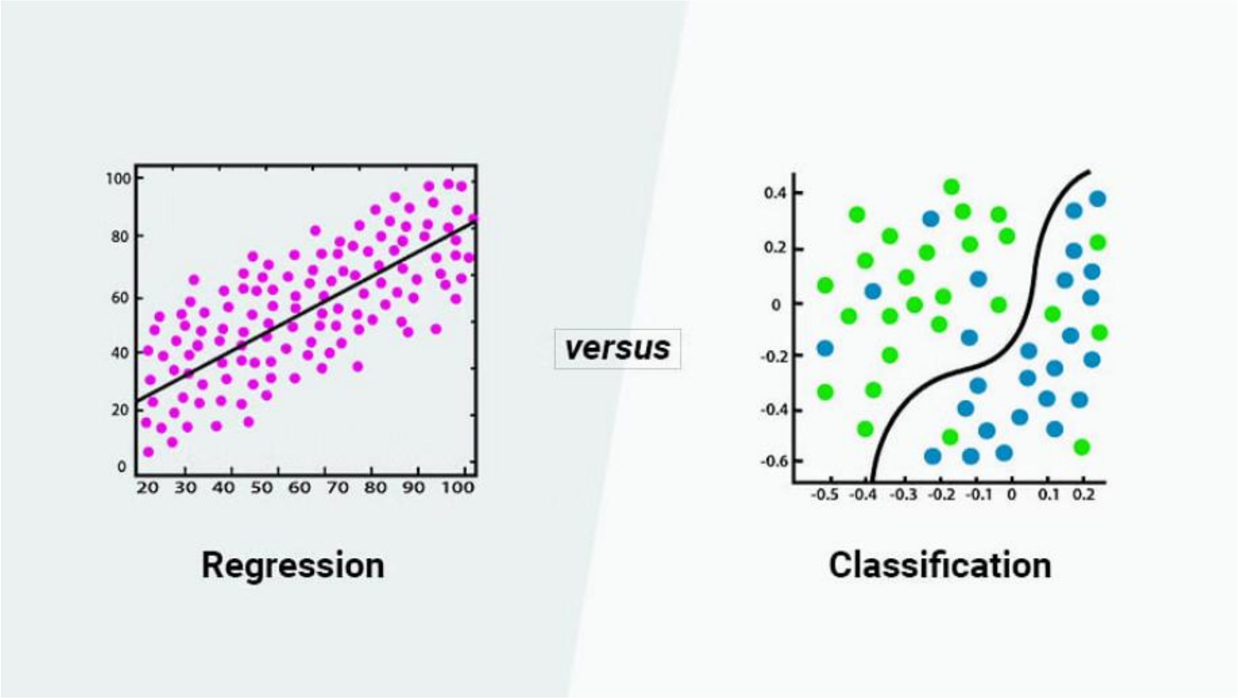
Aprendizado por reforço

Visão Geral

Paradigmas de aprendizado de máquina

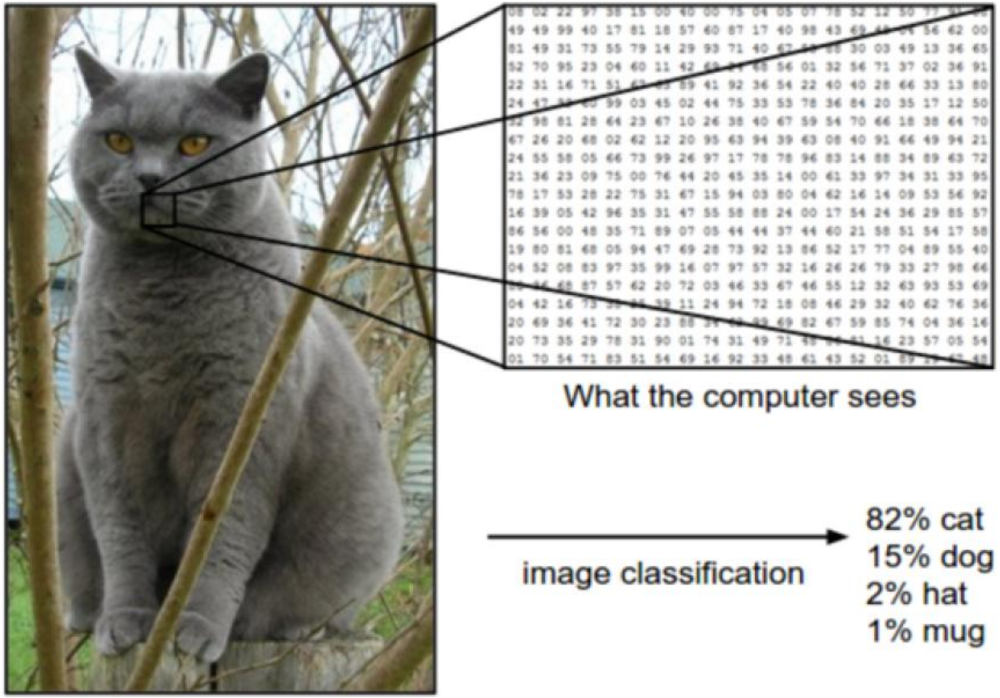
- Supervisionado (tem rótulos): Classificação, regressão.
- Não Supervisionado (não tem rótulos): Agrupamento (*clustering*), redução de dimensionalidade, detecção de anomalias, etc.
- Reforço: O agente aprende interagindo com o ambiente (baseado em recompensa).

Aprendizado Supervisionado



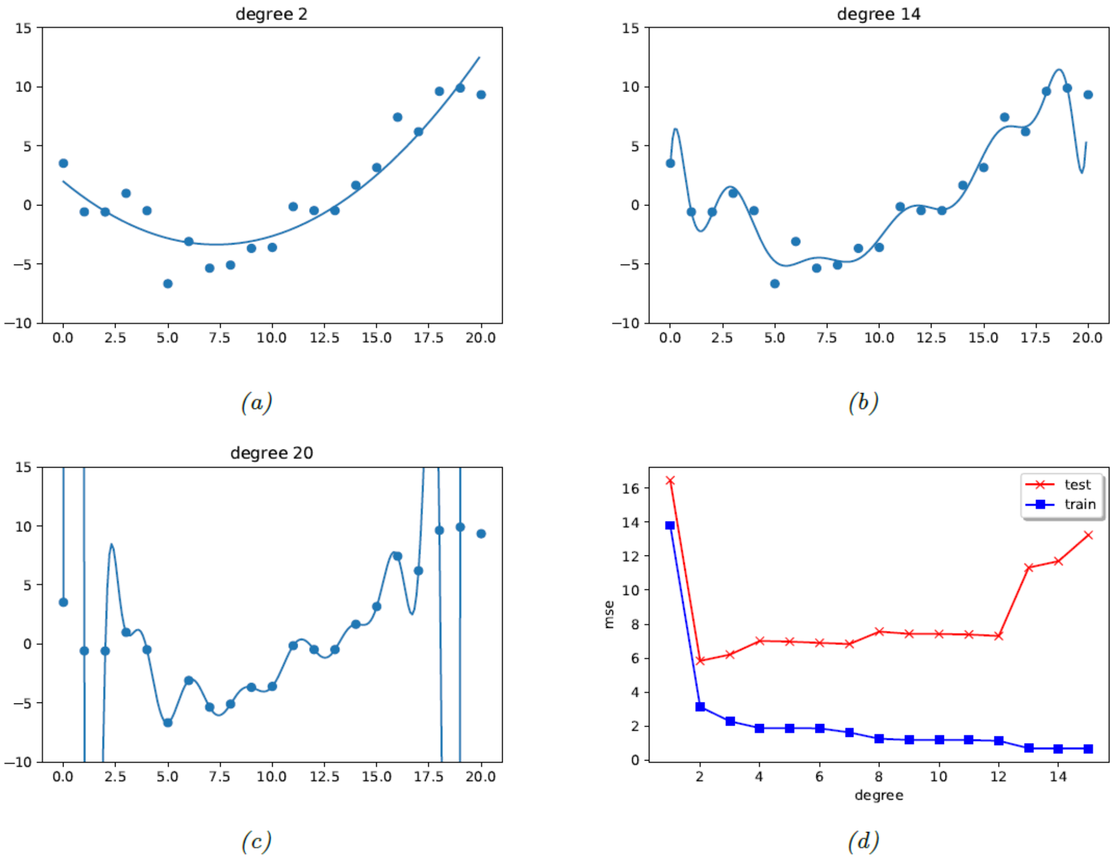
https://www.simplilearn.com/regression-vs-classification-in-machine-learning-article

Classificação



https://cs231n.github.io/classification/

Regressão

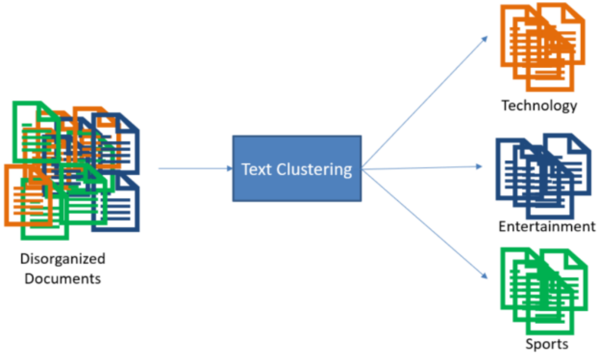


Kevin P. Murphy. Probabilistic Machine Learning: An introduction, MIT Press, 2022.

Aprendizado não-supervisionado

(atributos, não há rotulos)

https://www.linkedin.com/pulse/cluster-analysis-marketing-techniques-methods-use-cases-chetan-yadav



https://machinelearninggeek.com/text-clustering-clustering-news-articles/

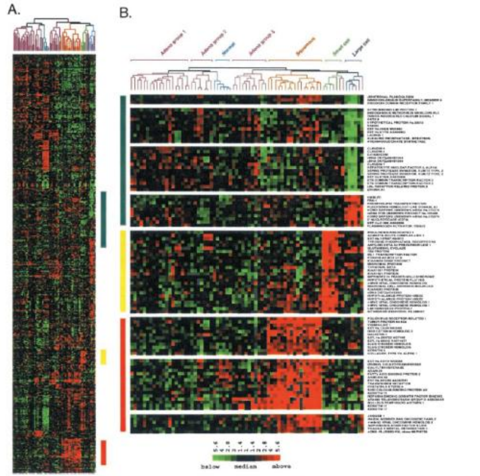
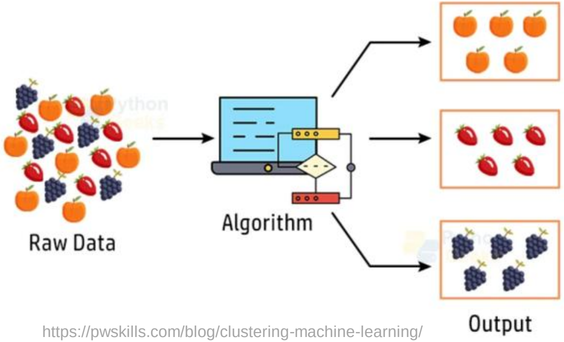
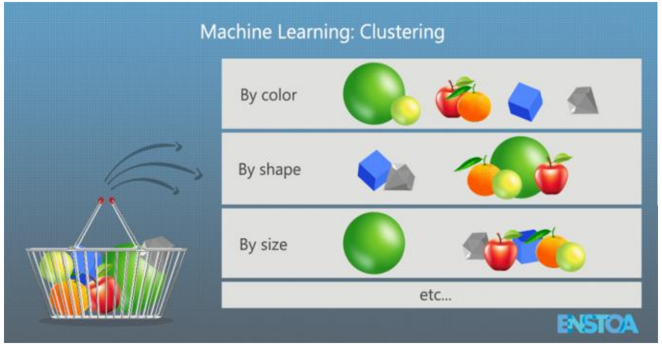


Fig. 2. Squamous, small cell, and large cell lung tumors express a unique set of genes. (A) Hierarchical clustering sorted 918 cDNA clones and 73 lung tissues based on similarity in gene expression. Gene clusters relevant to lung tumor types were extracted from the larger cluster of 918 clones in the regions indicated by the colored bars and expanded on the right to include gene names. A row in the cluster indicates expression of a specific gene across all 73 lung tissues. A column indicates the tissue in which the gene is expressed. Red, green, and black squares indicate that expression of the gene is greater than, less than, or equal to the median level of expression across all 73 lung tissues, respectively. Gray represents missing or poor quality data. (B) (Top) Gene clusters relevant to large cell tumors (blue bar). (Middle) Gene clusters relevant to small cell tumors (yellow bar). (Bottom) Gene clusters relevant to squamous lung tumors (red bar). The scale bar reflects the fold increase (red) or decrease (green) for any given gene relative to the median level of expression across all samples.

M. Garber, et al.. Diversity of gene expression in adenocarcinoma of the lung. Proceedings of National Academy of Sciences, vol. 98, pp. 13784-13789, 2001



https://pwskills.com/blog/clustering-machine-learning/



https://enstoa.com/blog/machine-learning-construction-how-clustering-data-can-improve-processes-part-2-of-2

Aprendizado por reforço

(Agente ⇔ Ambiente)

(Estado, Ação, Recompensa)

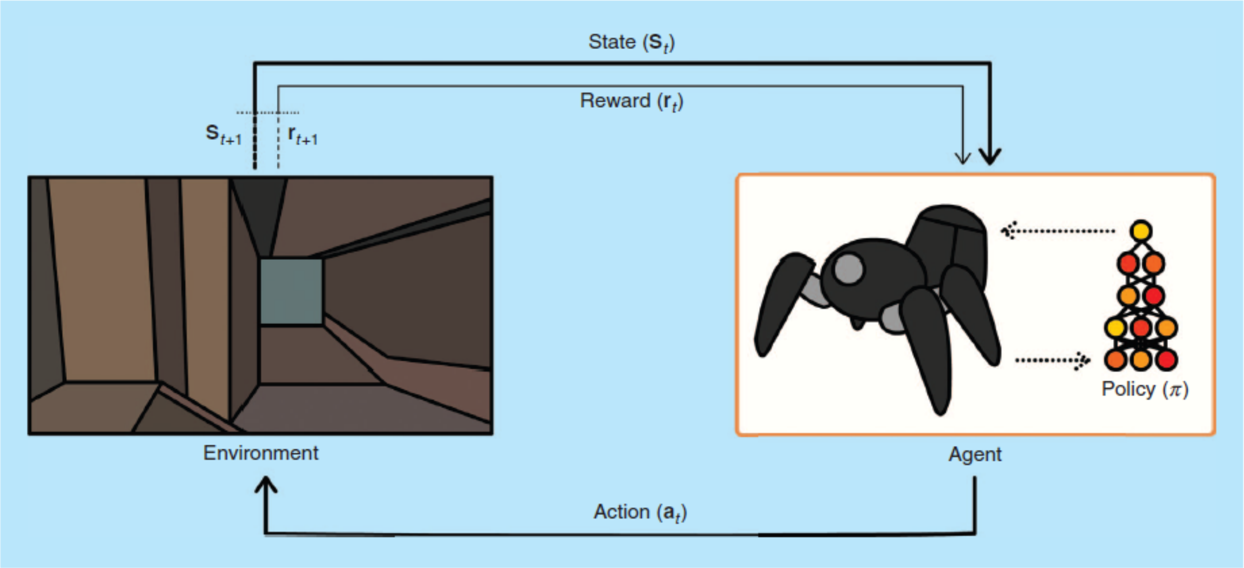


FIGURE 2. The perception-action-learning loop. At time t, the agent receives state $\mathbf{s}_t$ from the environment. The agent uses its policy to choose an action $\mathbf{a}_t$. Once the action is executed, the environment transitions a step, providing the next state, $\mathbf{s}_{t+1}$, as well as feedback in the form of a reward, r_{t+1}. The agent uses knowledge of state transitions, of the form $(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}, r_{t+1})$, to learn and improve its policy.

K. Arulkumaran et al., "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.

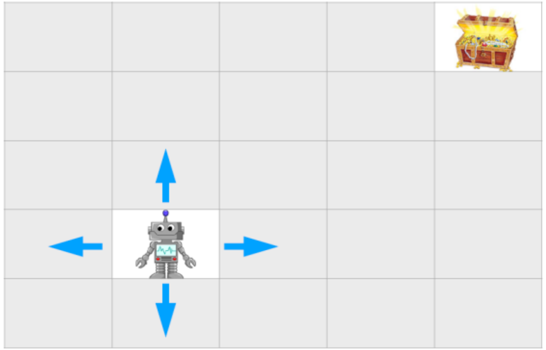


Fig. 1 Example of a simple grid world where RL techniques can be used to optimally reach a goal state from any starting position

E. F. Morales et al., "A survey on deep learning and deep reinforcement learning in robotics with a tutorial on deep reinforcement learning," *Intell. Serv. Robot.*, vol. 14, no. 5, pp. 773–805, Nov. 2021.

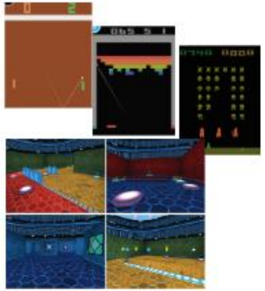
Aprendizado por reforço

Kohl and Stone, 2004

Ng et al, 2004

Tedrake et al, 2005

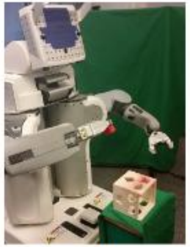
Kober and Peters, 2009



Mnih et al, 2015
(A3C)

Silver et al, 2014
(DPG)
Lillicrap et al, 2015
(DDPG)

Schulman et al,
2016 (TRPO + GAE)



Levine*, Finn*, et
al, 2016
(GPS)

Silver*, Huang*, et
al, 2016
(AlphaGo)

John Schulman & Pieter Abbeel – OpenAI + UC Berkeley

https://simons.berkeley.edu/sites/default/files/docs/6453/201703xxsimons-representations-deep-rl.pdf

Aprendizado por reforço

DeepMind

[DeepMind Atari]

[DeepMind AlphaGo]

[DeepMind AlphaStar]

[Manobras drone]

IEEE TRANSACTIONS ON NEURAL NETWORKS, VOL. 14, NO. 4, JULY 2003 929

Helicopter Trimming and Tracking Control Using Direct Neural Dynamic Programming

Russell Enns and Jennie Si

Abstract—This paper advances a neural-network-based approximate dynamic programming control mechanism that can be applied to complex control problems such as helicopter flight control design. Based on direct neural dynamic programming (DNDP), an approximate dynamic programming methodology, the control system is tailored to learn to maneuver a helicopter. The paper consists of a comprehensive treatise of this DNDP-based tracking control framework and extensive simulation studies for an Apache helicopter. A trim network is developed and seamlessly integrated into the neural dynamic programming (NDP) controller as part of a baseline structure for controlling complex nonlinear systems such as a helicopter. Design robustness is addressed by performing simulations under various disturbance conditions. All designs are tested using FLYRT, a sophisticated industrial scale nonlinear validated model of the Apache helicopter. This is probably the first time that an approximate dynamic programming methodology has been systematically applied to, and evaluated on, a complex, continuous state, multiple-input–multiple-output nonlinear system with uncertainty. Though illustrated for helicopters, the DNDP control system framework should be applicable to general purpose tracking control.

Index Terms—Approximate dynamic programming, helicopter flight control, helicopter trim, neural dynamic programming.

neuro-dynamic programming [2], adaptive critics [3], and so forth. Recently and most often, it has been referred to as approximate dynamic programming (ADP) [4]. This paper is not in a position to discuss which name fits the field the most. Rather, we consider techniques that converge to an (approximately) optimal policy over time in a nonlinear stochastic decision and control problem. Particularly, in this paper, we show that a recently proposed learning control framework [8], still under the theme of neural network, can solve very complex problems such as tracking of Apache helicopter.

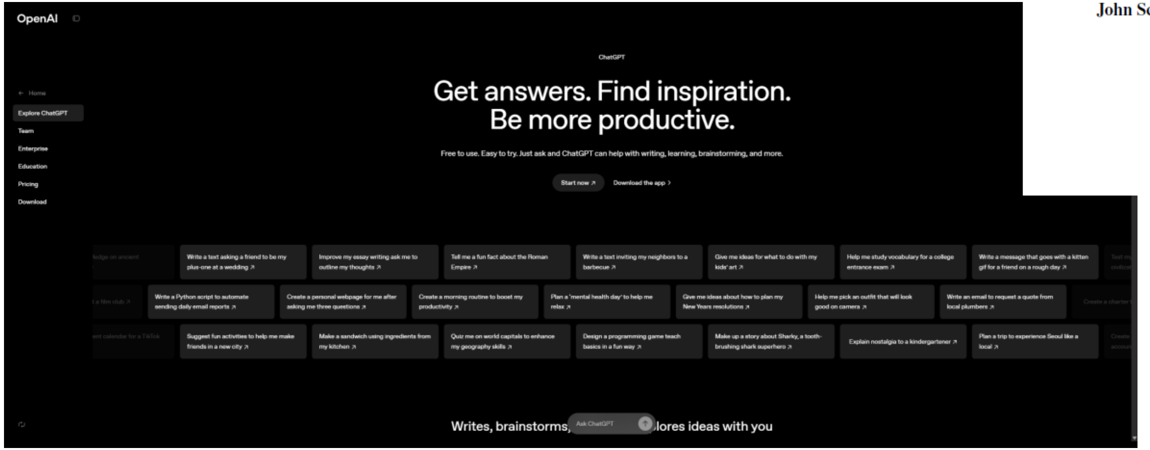
For the ease of discussion, the terms "discrete-event" approaches and "continuous-state" approaches are used to discuss solutions of ADP. The former refers to the fact that controls/actions are obtained by search algorithms and the problems are discrete event in nature. The latter refers to the fact that (approximate) gradient information is used in value function approximation and action generation, and the problems can be in both continuous or discrete-state spaces.

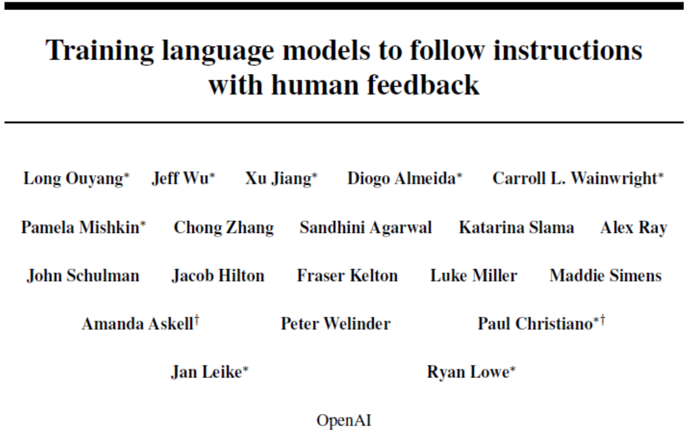
Until very recently [5], generalization problems remain a major hurdle in reinforcement learning community when

R. Enns and Jennie Si, "Helicopter trimming and tracking control using direct neural dynamic programming," in IEEE Transactions on Neural Networks, vol. 14, no. 4, pp. 929-939, July 2003.

Aprendizado por reforço

ChatGPT treinado usando aprendizagem por reforço a partir de feedback humano (RLHF).





Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*

Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

Amanda Askell† Peter Welinder Paul Christiano*†

Jan Leike* Ryan Lowe*

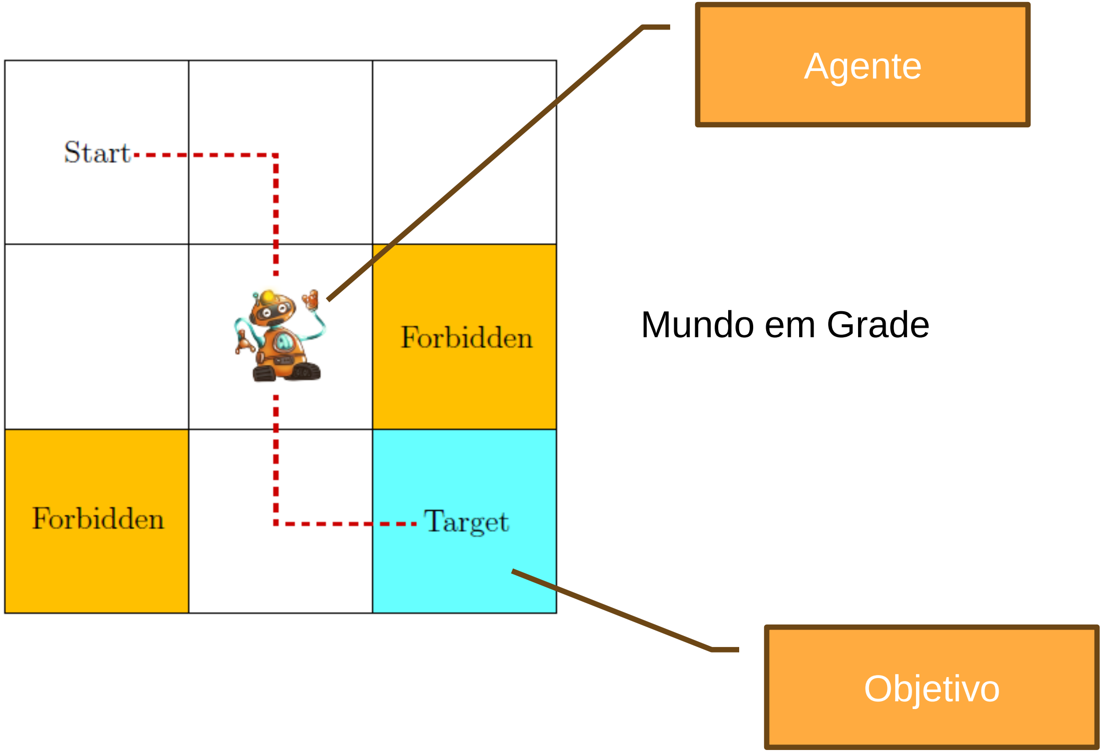
OpenAI

L. Ouyang et al., "Training language models to follow instructions with human feedback," Adv. Neural Inf. Process. Syst., vol. 35, no. NeurIPS, 2022.

Conceitos Básicos

Tarefa não-trivial quando o agente não tem nenhuma informação a priori sobre o ambiente!

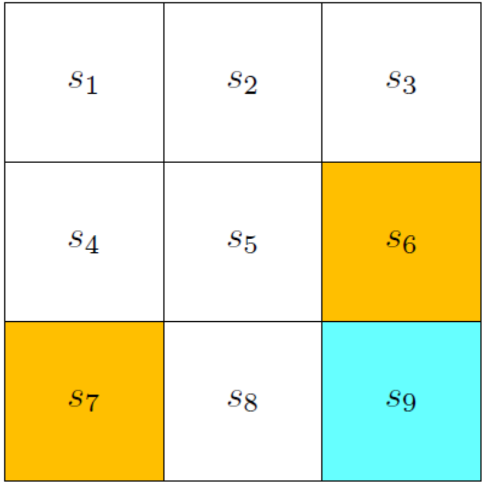
Queremos encontrar uma política para alcançar o alvo.



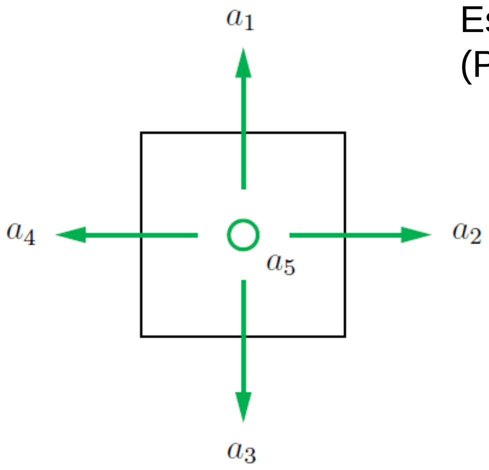
Mundo em Grade

Conceitos Básicos

- Estados e ação



(a) States

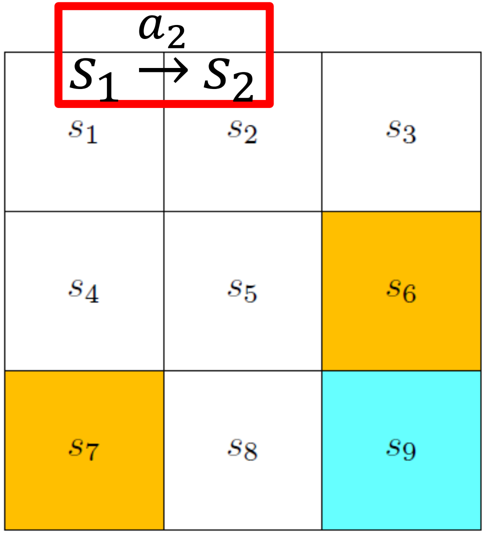


(b) Actions

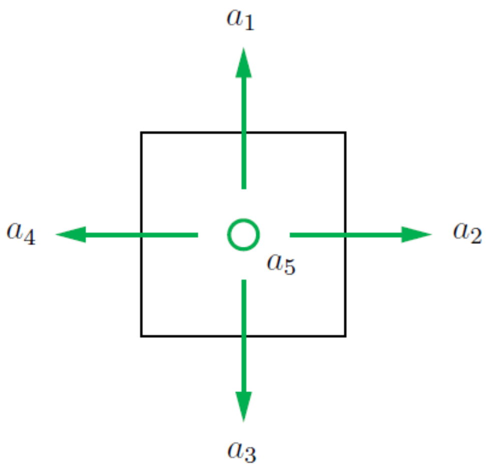
Espaço de estados: $S = \{s_1, \dots, s_9\}$
Espaço de ações: $\mathrm{A} = \{a_1, \dots, a_5\}$
(Pode ser uma função do estado $A(s_i)$)

Conceitos Básicos

- Transição de estados



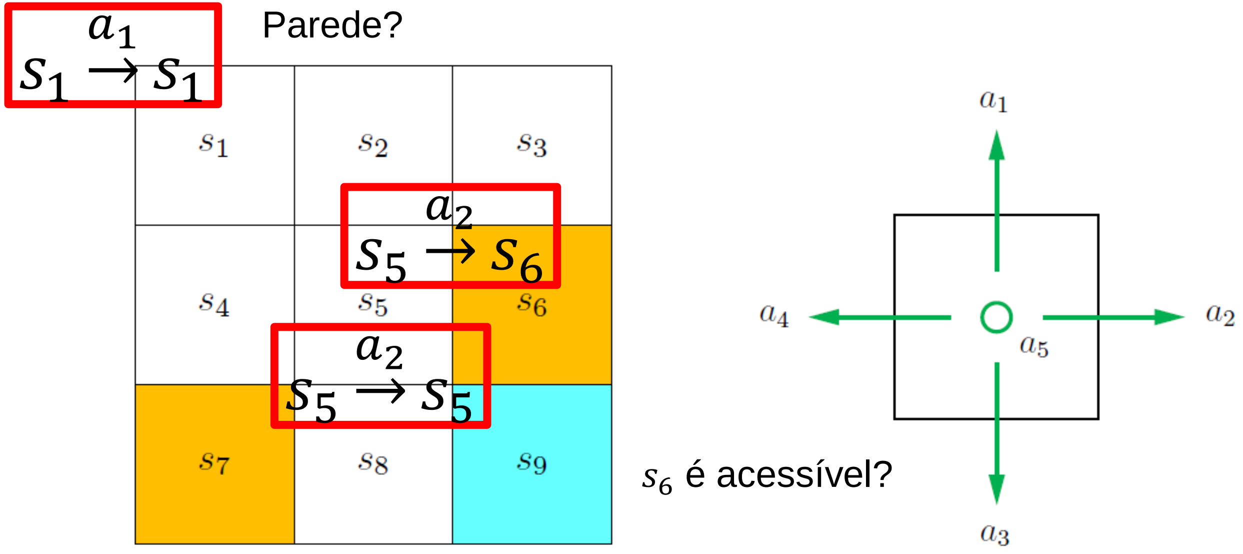
(a) States



(b) Actions

Conceitos Básicos

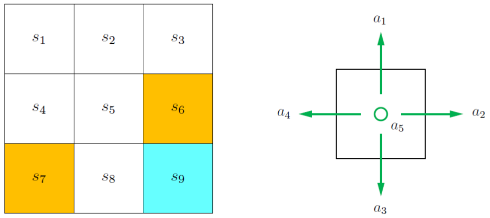
- Transição de estados



(a) States (b) Actions

Conceitos Básicos

- Transição de estados

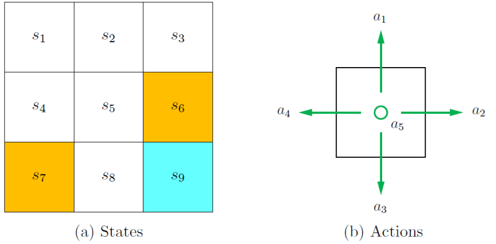


(a) States (b) Actions

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (still)
s_1	s_1	s_2	s_4	s_1	s_1
s_2	s_2	s_3	s_5	s_1	s_2
s_3	s_3	s_3	s_6	s_2	s_3
s_4	s_1	s_5	s_7	s_4	s_4
s_5	s_2	s_6	s_8	s_4	s_5
s_6	s_3	s_6	s_9	s_5	s_6
s_7	s_4	s_8	s_7	s_7	s_7
s_8	s_5	s_9	s_8	s_7	s_8
s_9	s_6	s_9	s_9	s_8	s_9

Conceitos Básicos

- Transição de estados (probabilidade condicional)



	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (still)
s_1	s_1	s_2	s_4	s_1	s_1
s_2	s_2	s_3	s_5	s_1	s_2
s_3	s_3	s_3	s_6	s_2	s_3
s_4	s_1	s_5	s_7	s_4	s_4
s_5	s_2	s_6	s_8	s_4	s_5
s_6	s_3	s_6	s_9	s_5	s_6
s_7	s_4	s_8	s_7	s_7	s_7
s_8	s_5	s_9	s_8	s_7	s_8
s_9	s_6	s_9	s_9	s_8	s_9

Determinístico!

$p(s_1|s_1,a_2) = 0$

$p(s_2|s_1,a_2) = 1$

$p(s_3|s_1,a_2) = 0$

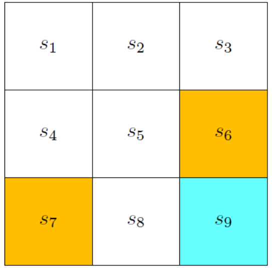
$p(s_4|s_1,a_2) = 0$

$p(s_5|s_1,a_2) = 0$

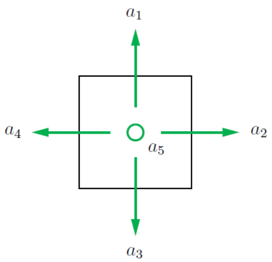
Mas pode ser estocástico
(se por exemplo $p(s_5|s_1,a_2) > 0$)!

Conceitos Básicos

- Política ($\pi(a|s)$):
    - indica que ação o agente deve tomar em cada estado.
    - Seguir uma política gera uma trajetória



(a) States



(b) Actions

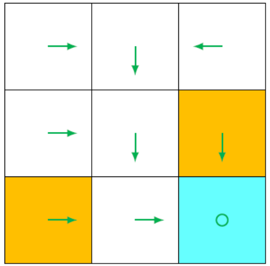
$$\pi(a_1|s_1) = 0$$

$$\pi(a_2|s_1) = 1$$

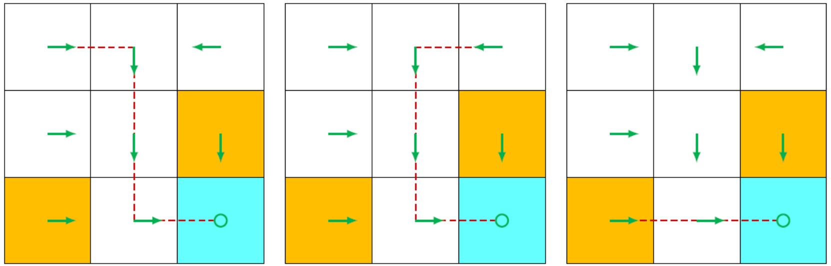
$$\pi(a_3|s_1) = 0$$

$$\pi(a_4|s_1) = 0$$

$$\pi(a_5|s_1) = 0$$



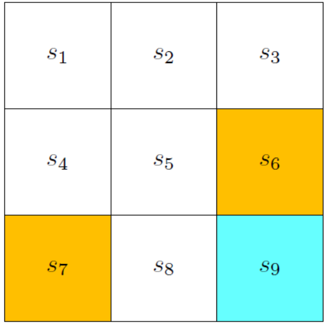
(a) A deterministic policy



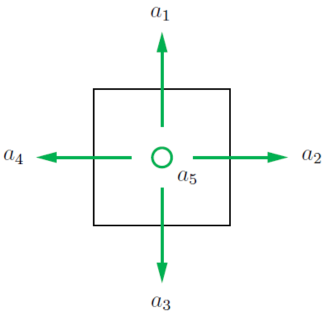
(b) Trajectories obtained from the policy

Conceitos Básicos

- Política ($\pi(a|s)$) em geral são estocásticas



(a) States



(b) Actions

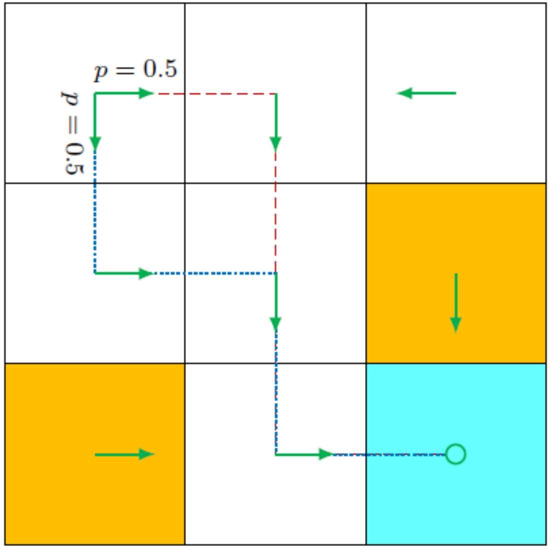
$$\pi(a_1|s_1) = 0$$

$$\pi(a_2|s_1) = 0.5$$

$$\pi(a_3|s_1) = 0.5$$

$$\pi(a_4|s_1) = 0$$

$$\pi(a_5|s_1) = 0$$



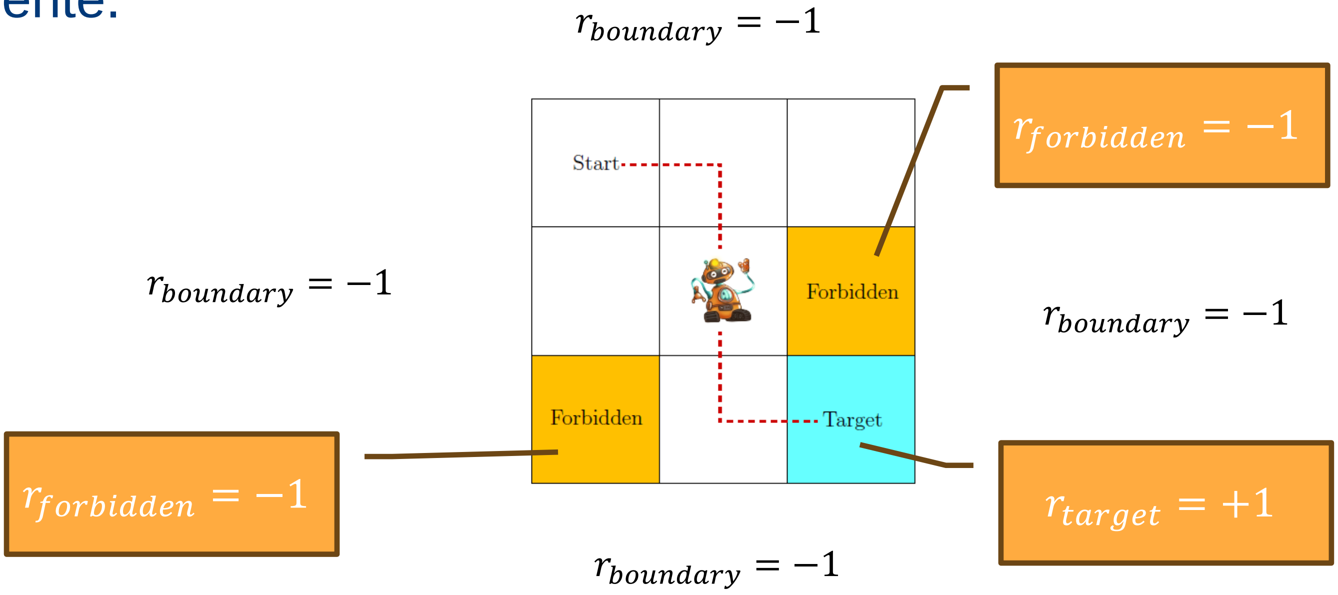
Conceitos Básicos

- Política ($\pi(a|s)$): representação tabular

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (still)
s_1	0	0.5	0.5	0	0
s_2	0	0	1	0	0
s_3	0	0	0	1	0
s_4	0	1	0	0	0
s_5	0	0	1	0	0
s_6	0	0	1	0	0
s_7	0	1	0	0	0
s_8	0	1	0	0	0
s_9	0	0	0	0	1

Conceitos Básicos

- Recompensa ($r(s, a)$): depois de executar uma ação em um dado estado o agente recebe uma recompensa como um feedback do ambiente.



Conceitos Básicos

- Recompensa ($r(s,a)$): representação tabular

	a_1 (upward)	a_2 (rightward)	a_3 (downward)	a_4 (leftward)	a_5 (still)
s_1	r_{boundary}	0	0	r_{boundary}	0
s_2	r_{boundary}	0	0	0	0
s_3	r_{boundary}	r_{boundary}	$r_{\text{forbidden}}$	0	0
s_4	0	0	$r_{\text{forbidden}}$	r_{boundary}	0
s_5	0	$r_{\text{forbidden}}$	0	0	0
s_6	0	r_{boundary}	r_{target}	0	$r_{\text{forbidden}}$
s_7	0	0	r_{boundary}	r_{boundary}	$r_{\text{forbidden}}$
s_8	0	r_{target}	r_{boundary}	$r_{\text{forbidden}}$	0
s_9	$r_{\text{forbidden}}$	r_{boundary}	r_{boundary}	0	r_{target}

$p(r = -1 | s_1, a_1) = 1, \qquad p(r \neq -1 | s_1, a_1) = 0$ Determinístico!

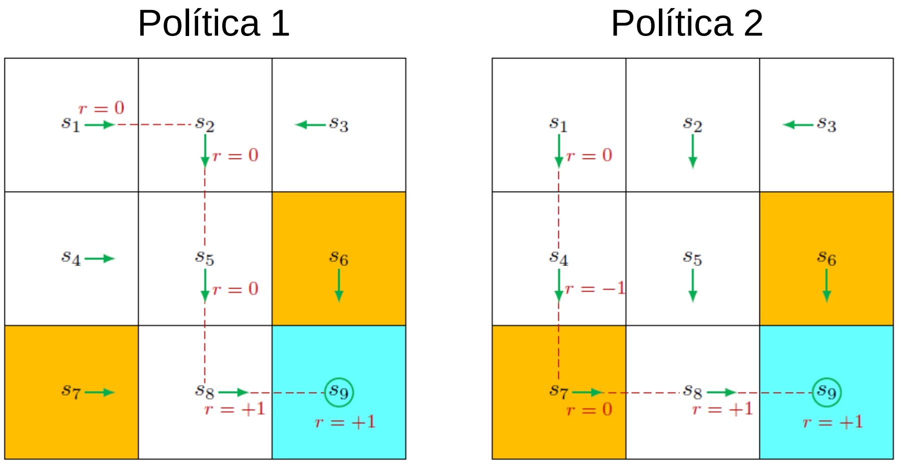
Mas em geral, pode ser estocástico!

Conceitos Básicos

- Recompensa ($r(s, a)$):
    - recompensa imediata vs. total de recompensas

Conceitos Básicos

- Trajetórias, retornos e episódios
  - Trajetória: cadeia de estado-ação-recompense
  - Retorno: soma de todas as recompensas coletadas ao longo da trajetória. Usado para avaliar políticas



(a) Policy 1 and the trajectory (b) Policy 2 and the trajectory

Política 1

$$s_1 \xrightarrow[r=0]{a_2} s_2 \xrightarrow[r=0]{a_3} s_5 \xrightarrow[r=0]{a_3} s_8 \xrightarrow[r=1]{a_2} s_9$$

$$retorno = 0 + 0 + 0 + 1 = 1$$

Política 2

$$s_1 \xrightarrow[r=0]{a_3} s_4 \xrightarrow[r=-1]{a_3} s_7 \xrightarrow[r=0]{a_2} s_8 \xrightarrow[r=1]{a_2} s_9$$

$$retorno = 0 - 1 + 0 + 1 = 0$$

Conceitos Básicos

- Trajetórias, retornos e episódios
    - As ações tomadas devem ser determinadas pelo retorno (recompensa total) ao invés da recompensa imediata.
    - Trajetórias infinitas:

$$s_1 \underset{r=0}{\overset{a_2}{\rightarrow}} s_2 \underset{r=0}{\overset{a_3}{\rightarrow}} s_5 \underset{r=0}{\overset{a_3}{\rightarrow}} s_8 \underset{r=1}{\overset{a_2}{\rightarrow}} s_9 \underset{r=1}{\overset{a_5}{\rightarrow}} s_9 \underset{r=1}{\overset{a_5}{\rightarrow}} s_9 \cdots$$

    - Problema:

$$retorno \ = 0 + 0 + 0 + 1 + 1 + 1 + \cdots = \infty$$

Conceitos Básicos

- Trajetórias, retornos e episódios
  - As ações tomadas devem ser determinadas pelo retorno (recompensa total) ao invés da recompensa imediata.
  - Trajetórias infinitas:

$$s_1 \underset{r=0}{\overset{a_2}{\rightarrow}} s_2 \underset{r=0}{\overset{a_3}{\rightarrow}} s_5 \underset{r=0}{\overset{a_3}{\rightarrow}} s_8 \underset{r=1}{\overset{a_2}{\rightarrow}} s_9 \underset{r=1}{\overset{a_5}{\rightarrow}} s_9 \underset{r=1}{\overset{a_5}{\rightarrow}} s_9 \cdots$$

  - Problema:

$$retorno \ = 0 + 0 + 0 + 1 + 1 + 1 + \cdots = \infty$$

  - Solução: retorno descontado (introdução de um fator de desconto $\gamma \in (0,1)$):

$$retorno\ descontado = 0 + \gamma 0 + \gamma^2 0 + \gamma^3 1 + \gamma^4 1 + \gamma^5 1 + \cdots$$
$$retorno\ descontado = \gamma^3 (1 + \gamma + \gamma^2 + \cdots\ )$$
$$retorno\ descontado = \gamma^3 \frac{1}{1-\gamma}$$

Conceitos Básicos

- Terminologia: o agente pode parar em **estados terminais**. A trajetória resultante é chamada de **episódio.**
- Episódios no caso de políticas/ambientes estocásticos vs. determinísticos

Referências

- Shiyu Zhao. Mathematical Foundations of Reinforcement Learning. Springer Singapore, 2025. [capítulo 1]
  - disponível em: https://github.com/MathFoundationRL/Book-Mathematical-Foundation-of-Reinforcement-Learning

Slides construídos com base no livro supracitado, o qual está disponibilizado publicamente pelo seu respectivo autor.