(7) Introdução ao Aprendizado por Reforço

Redes Neurais e Arquiteturas Profundas

Moacir A. Ponti
CeMEAI/ICMC, Universidade de São Paulo
MBA em Ciência de Dados

www.icmc.usp.br/~moacir — moacir@icmc.usp.br

São Carlos-SP/Brasil – 2020

Agenda

Agenda

Paradigmas de aprendizado de máquina

Supervisionado

Dados:
$(x, y) \in X, Y$ com anotações Y

Objetivo: aprender relação entre X e Y
$f : X \rightarrow Y$

Exemplo:

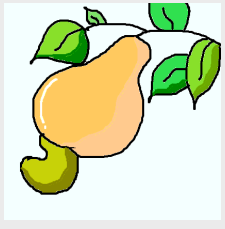
isso é um caju

Não-supervisionado

Dados: $(x) \in X$

Objetivo:
particionar X
por similaridade ou estrutura dos dados

Exemplo:

esses dois exemplos são semelhantes

Por reforço

Dados: $(s, a) \in S, A$

Objetivo:
Maximizar recompensa futura ao longo de passos

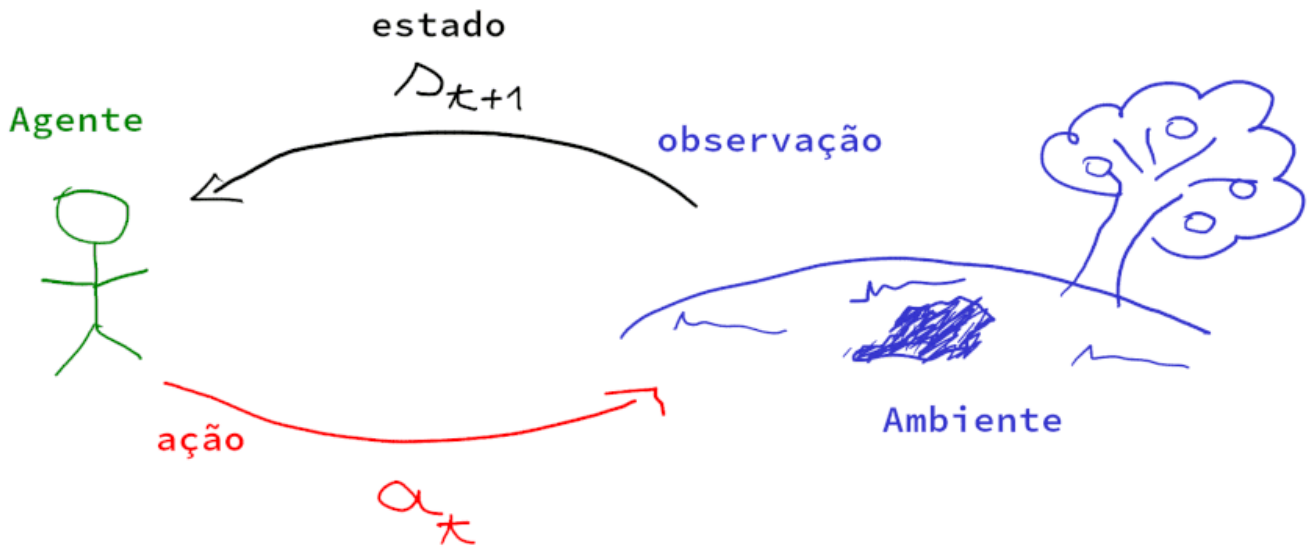
Exemplo:

coma isso pois te manterá vivo

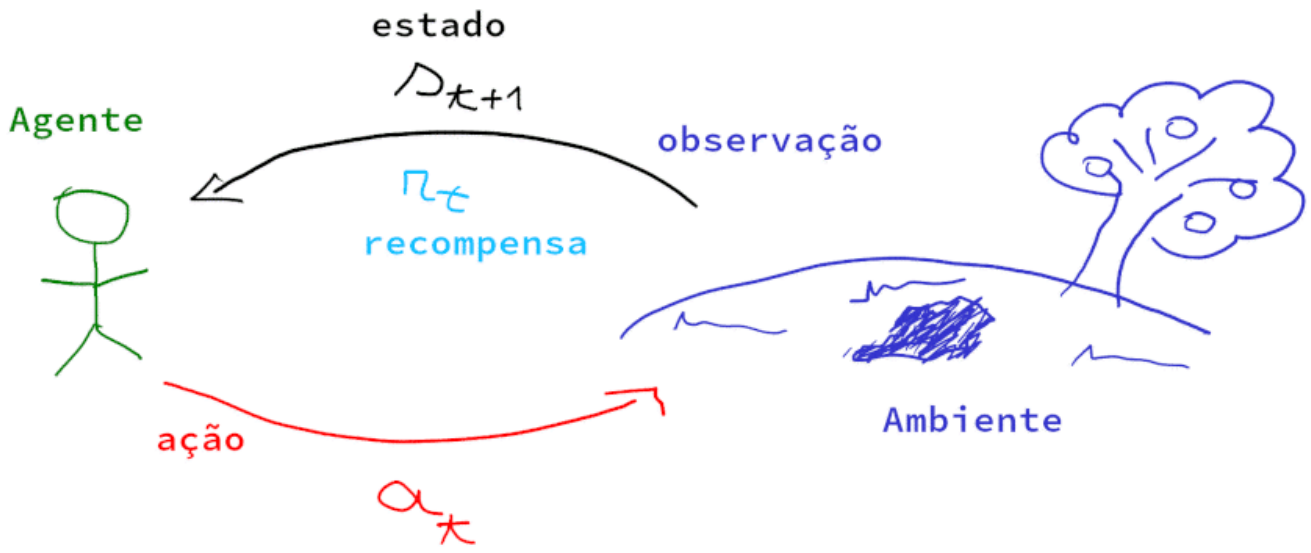
Aprendizado por reforço

Aprendizado por reforço

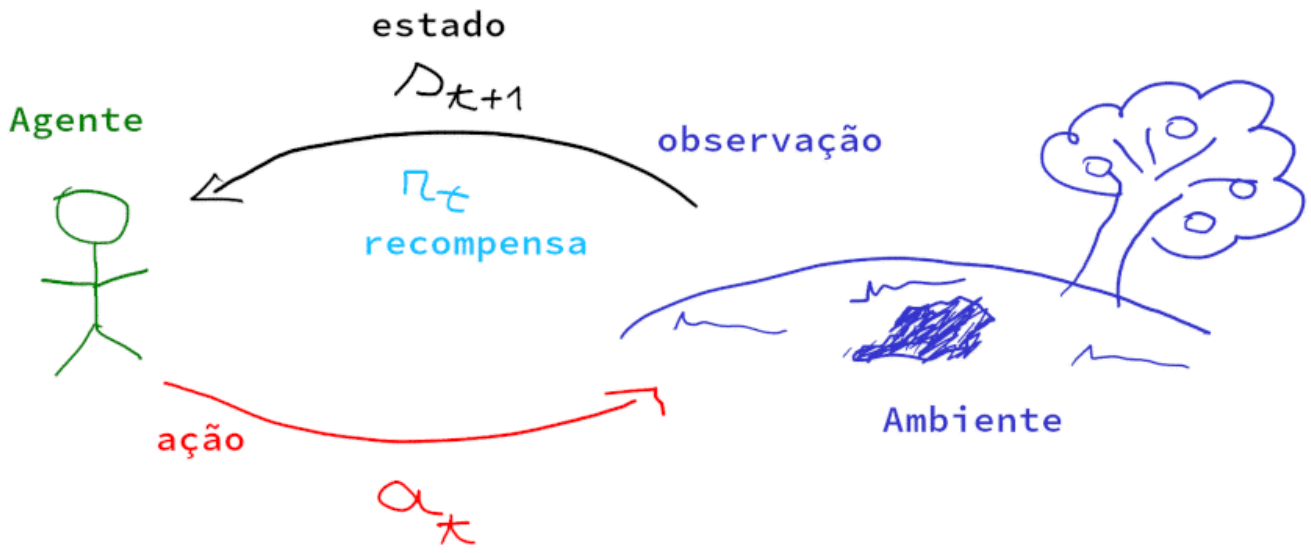
Aprendizado por reforço



Aprendizado por reforço



Aprendizado por reforço



Aprendizado por reforço

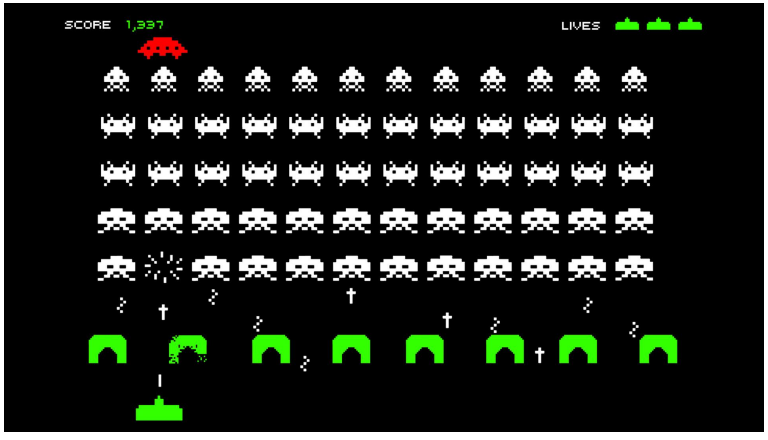
Otimizar com objetivo de: **tomar boas decisões** ou aprender **boas estratégias**

Consequências atrasadas: **decisões atuais podem impactar a longo prazo**

Aprender tomando decisões e falhando por meio de recompensas, sem saber se outra ação seria melhor ou não

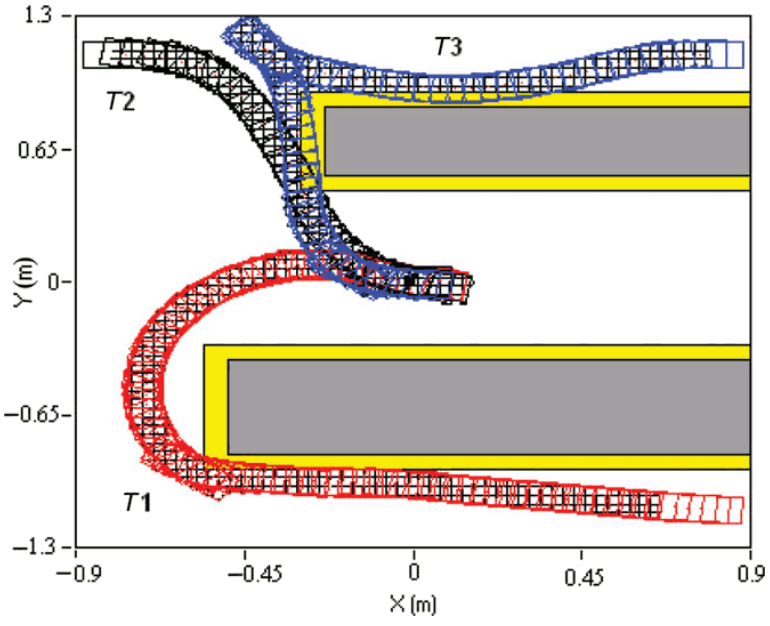
Generalização: uma política mapeia **experiências passadas em uma ação**

Aplicações: jogos



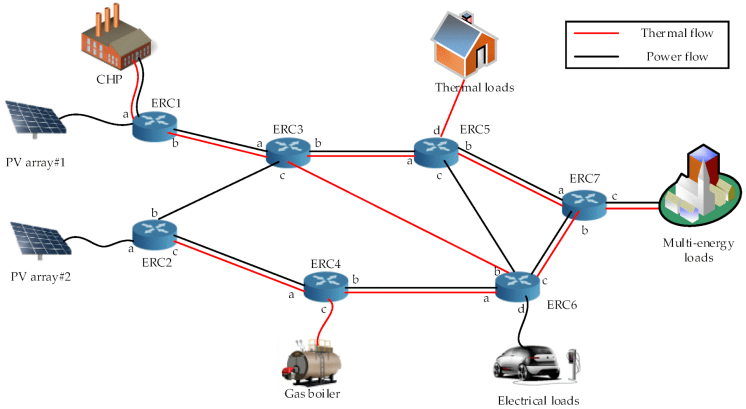
DeepMind: "Playing Atari" and "AlphaGo"

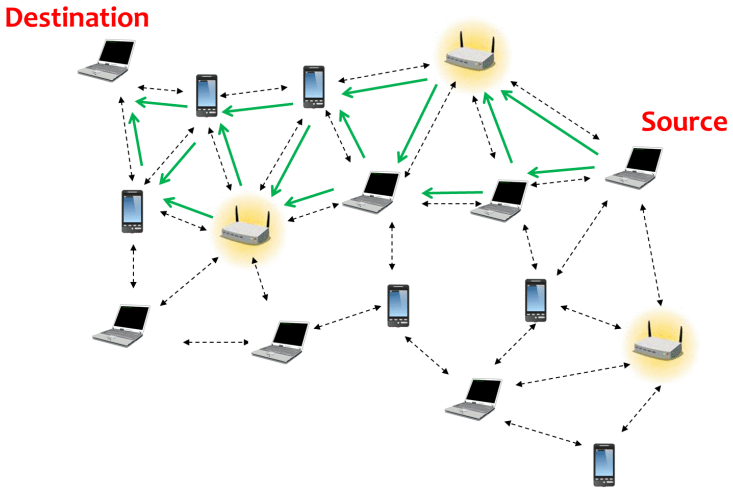
Aplicações: robótica móvel



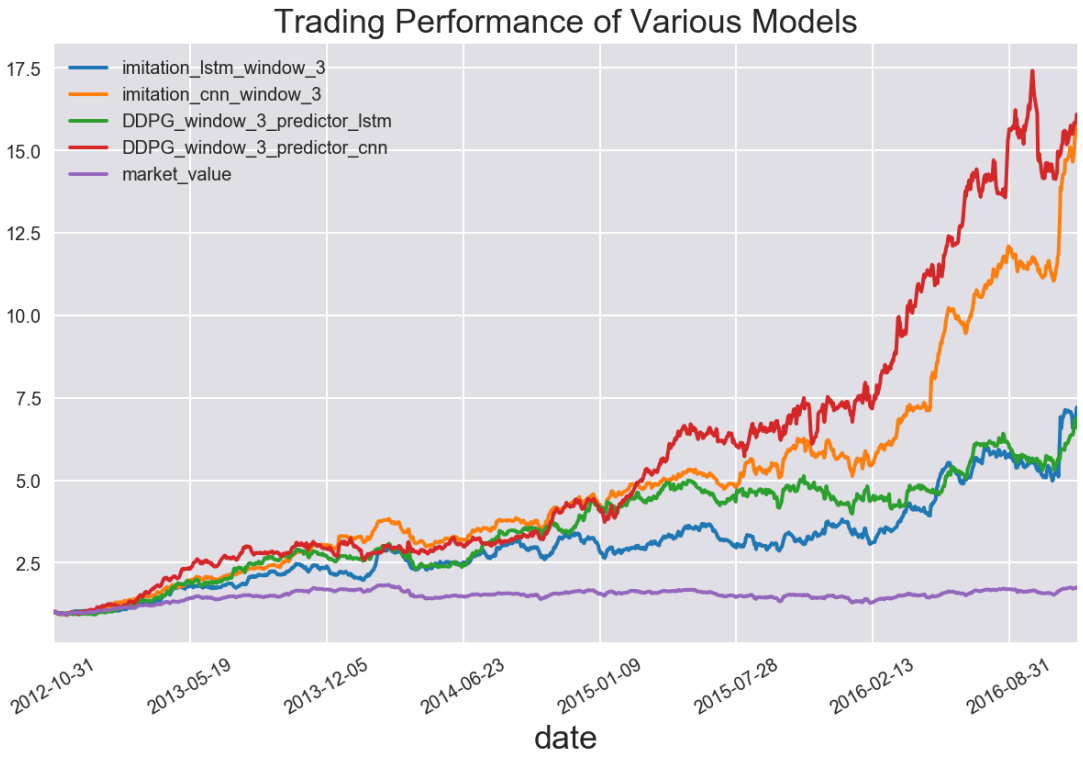
https://www.youtube.com/watch?v=fv-oFPAqSZ4

Aplicações: infraestrutura





Aplicações: trading



Agenda

Agenda

Agente e Ambiente

- Ação a_t
- Recompensa r_t
- Observação o_t

Ações sequenciais

Objetivo: selecionar ações para maximizar a recompensa total futura

Ações podem ter consequências de longo prazo

- Atraso no retorno/feedback
- Pode ser preciso sacrificar recompensa imediata para ganhar mais a longo prazo

Recompensa

- r_t é um valor (sinal) que indica a recompensa no tempo t

Definição: hipótese da recompensa

Todos os objetivos podem ser descritos pela maximização da expectativa da recompensa acumulada

Recompensa: exemplos

Manobrar um drone

- Positivo: seguir trajetória desejada
- Negativo: desviar trajetória ou cair

Jogar arcade contra o computador

- Positivo: ganhar pontos
- Negativo: perder vida

Gerenciar operações de compra e venda na bolsa

- Positivo: operação resultou em lucro
- Negativo: operação resultou em prejuízo

Aprendizado por reforço e mecanismo de recompensas

Recompensa total é calculada pela soma de recompensas *futuras*

$$R_t = \sum_{i=t}^{\infty} r_i$$

Aprendizado por reforço e mecanismo de recompensas

Recompensa total descontada: soma de recompensas *futuras ponderadas*

$$R_t = \sum_{i=t}^{\infty} \gamma^i r_i = \gamma^t r_t + \gamma^{t+1} r_{t+1} + \cdots + \gamma^{t+n} r_{t+n} + \cdots$$

$\gamma \in [0, 1]$ dá menos peso a recompensas a longo prazo

Estado do Ambiente/Environment

Representação interna/privada do ambiente $s_t^{(e)}$

- dados usados pelo ambiente para selecionar a próxima observação e recompensa
- comumente não visível pelo agente, que tem acesso apenas à observação/recompensa

Estado do Agente

Representação interna do agente $s_t^{(a)}$

- dados utilizados para selecionar a próxima ação
- geralmente modelamos $s_t^{(a)} = f(h_t)$

Histórico e Estados

Histórico é uma sequência: $h_t = a_1, o_1, r_1, \ldots, a_t, o_t, r_t$

- Agente: seleciona ações
- Ambiente: seleciona observações e recompensas

Estado

Informação usada para determinar o próximo passo, formalmente estado é uma função do histórico:

$$s_t = f(h_t)$$

Estado de Markov

Estado que carrega toda a informação útil do histórico

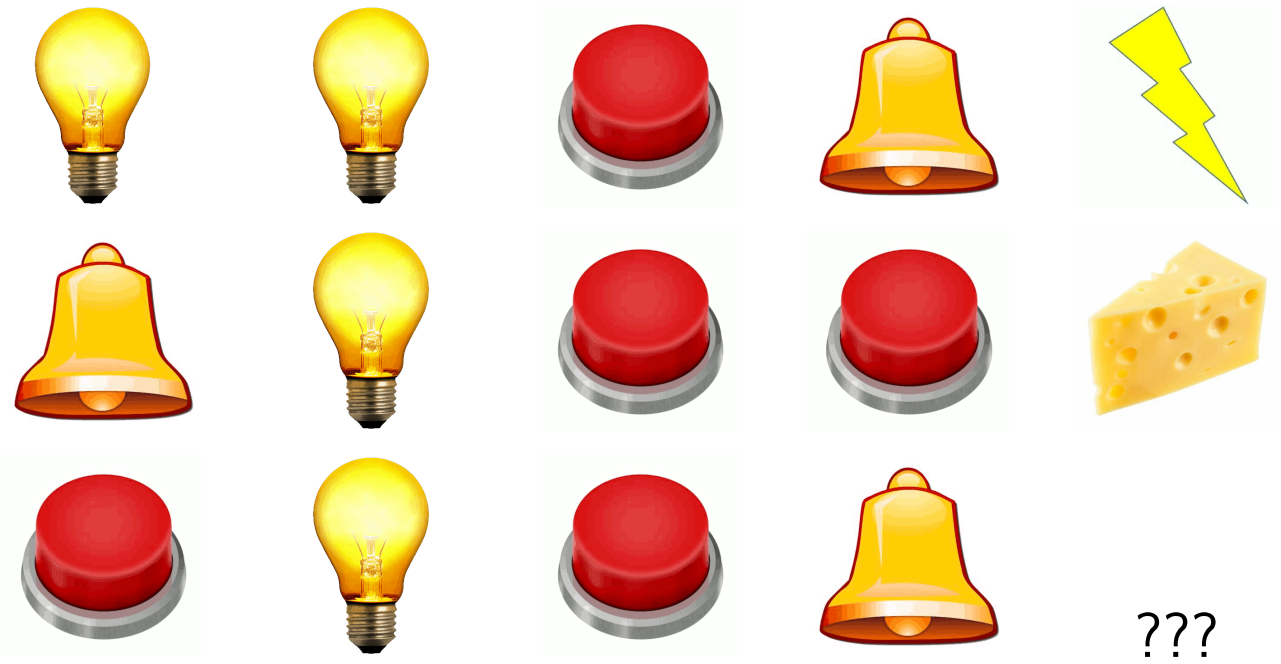
Estado de Markov

Estado s_t para o qual temos a propriedade:

$$P[S_{t+1}|S_t] = P[S_{t+1}|S_1, \ldots S_t]$$

- O estado é uma estatística suficiente do histórico (estados anteriores são "descartáveis")
- h_t (histórico) é um estado de Markov

Exemplos de estados do agente



Considerando o estado do agente:

- últimas 3 observações em sequência...
- frequência de luz, alarme e apertar botão...
- sequência completa de observações...

Completude da observação

Observação completa

- Agente observa o estado do ambiente diretamente

$$o_t = s_t^{(a)} = s_t^{(e)}$$

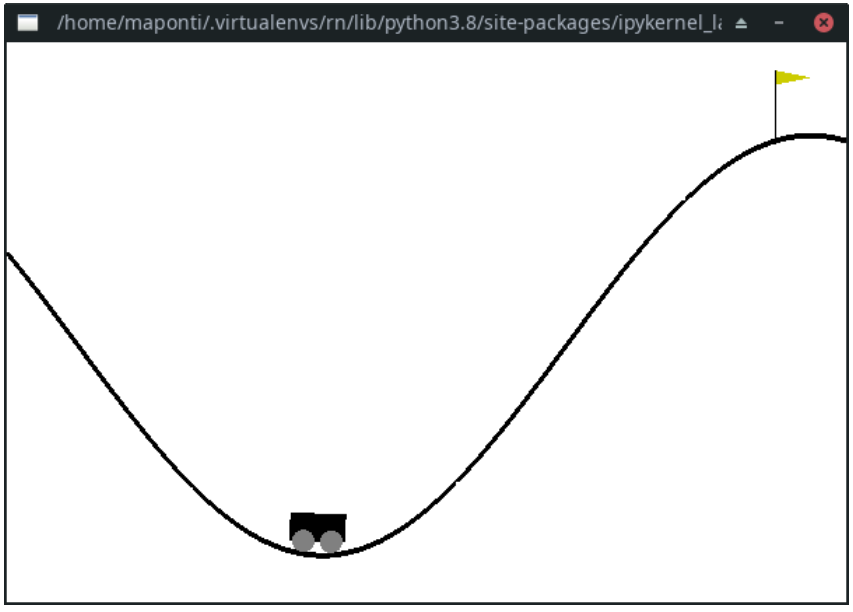
- Markov Decision Process (MDP)

Completude da observação

Observação parcial

- Agente observa o estado do ambiente indiretamente
- Deve construir sua própria representação, exemplos:
  - histórico completo: $s_t^{(a)} = h_t$
  - probabilidades do estado do ambiente
    $s_t^{(a)} = (P[s_t^{(e)} = s^{(1)}], \ldots, P[s_t^{(e)} = s^{(n)}])$
  - unidade recorrente $s_t^{(a)} = s_{t-1}^{(a)} W_s + o_t W_o$
- Partially Observed Markov Decision Process (POMDP)

Exemplos: mountaincar



- Observação: posição e velocidade
- Ações: mover para esquerda, mover para direita, não acelerar
- Recompensa: +1 para posição da bandeira, -1 para outras posições
- Estado terminal: posição da bandeira

Exemplos: 21 / Blackjack

- Observação: soma das cartas do jogador, apenas uma carta da banca virada para cima
- Ações: pedir carta, parar
- Recompensa: valor positivo para vitória, valor negativo para derrota
- Estado terminal: jogador para ou estoura 21

Agenda

Política / Policy

- Define o comportamento do agente
- Mapa do estado para a ação
  - Determinístico: $a = \pi(s)$
  - Probabilístico/estocástico: $a = \pi(a|s) = P[a|s]$

Exploration vs Exploitation

- Exploration: buscar mais informação sobre o ambiente
- Exploitation: fazer uso da informação conhecida para maximizar recompensa

Modelo

- Descreve uma forma de prever o que acontecerá com o ambiente
- **Transições**: $\mathcal{P}$ prediz o próximo estado
  $\mathcal{P}^a_{ss'} = P[s'|s, a]$
- **Recompensa**: $\mathcal{R}$ prediz a próxima recompensa
  $\mathcal{R} = E[r|s, a]$

Função valor / objetivo

- **Predição** da recompensa futura
- Avalia a "qualidade" do estado s

$$v_\pi(s) = E_\pi[r_t + \gamma_1 r_{t+1} + \gamma_2 r_{t+2} + \cdots | s_t = s]$$

- "discounted future reward": quanto mais futuro, menor sua importância

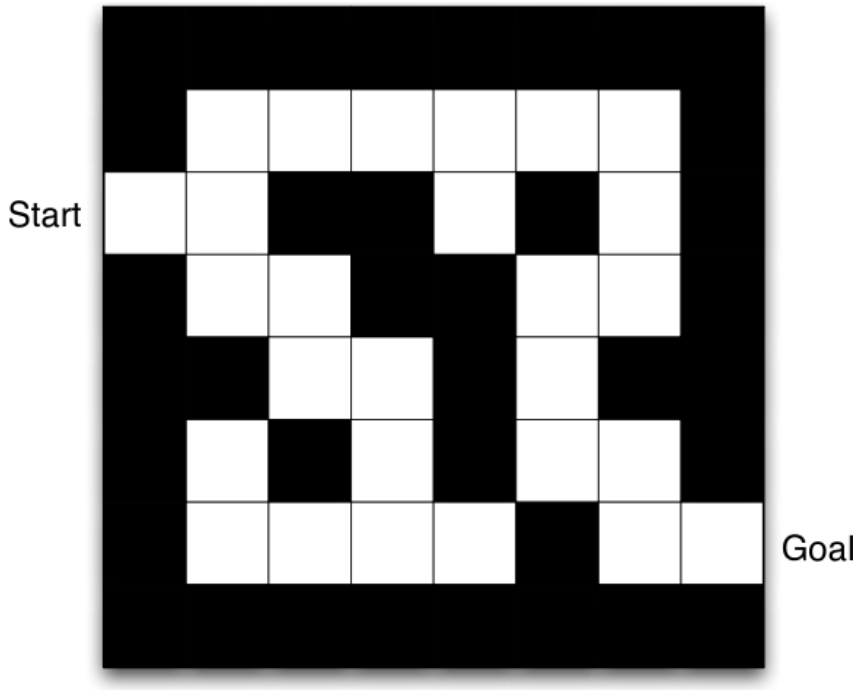
Função Q

$$R_t = \sum_{i=t}^{\infty} \gamma^i r_i = \gamma^t r_t + \gamma^{t+1} r_{t+1} + \cdots + \gamma^{t+n} r_{t+n} + \cdots$$

$$Q(s_t, a_t) = E[R_t | s_t, a_t]$$

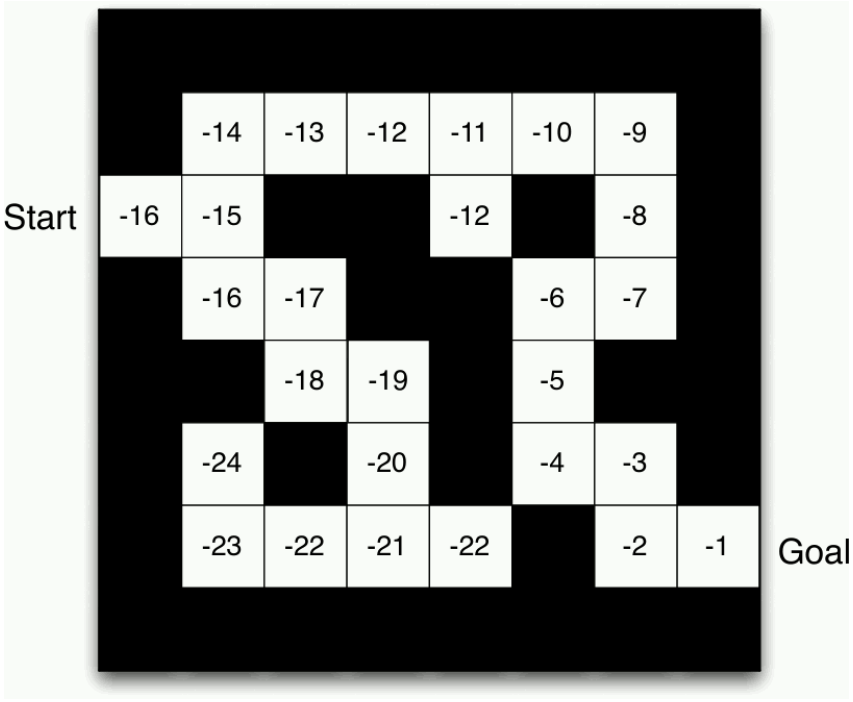
A **recompensa futura total esperada** que um agente no estado s pode receber ao executar ação a

Labirinto



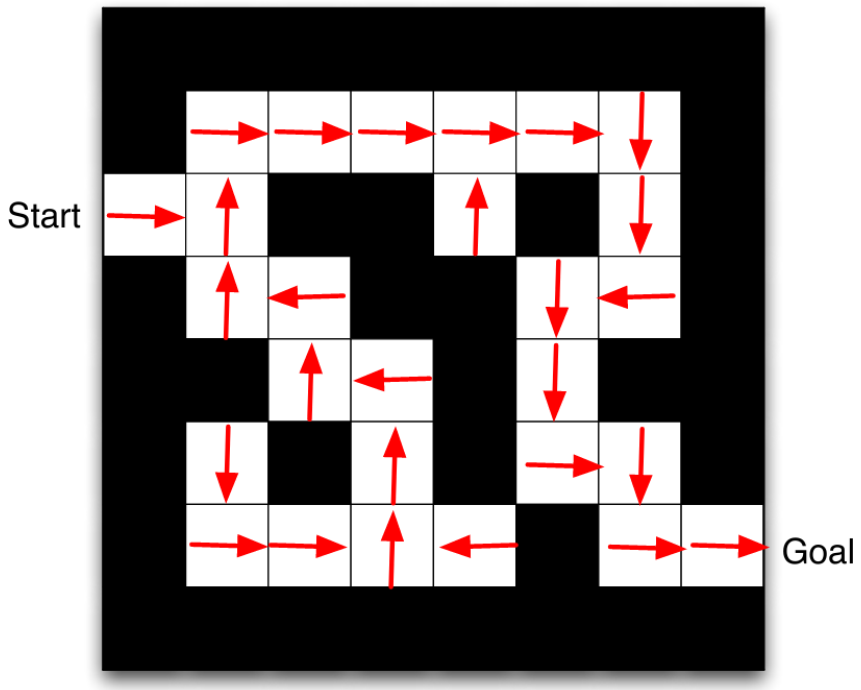
- Recompensas: -1 por tempo
- Ações: direções
- Estado: localização do agente

Labirinto: função valor



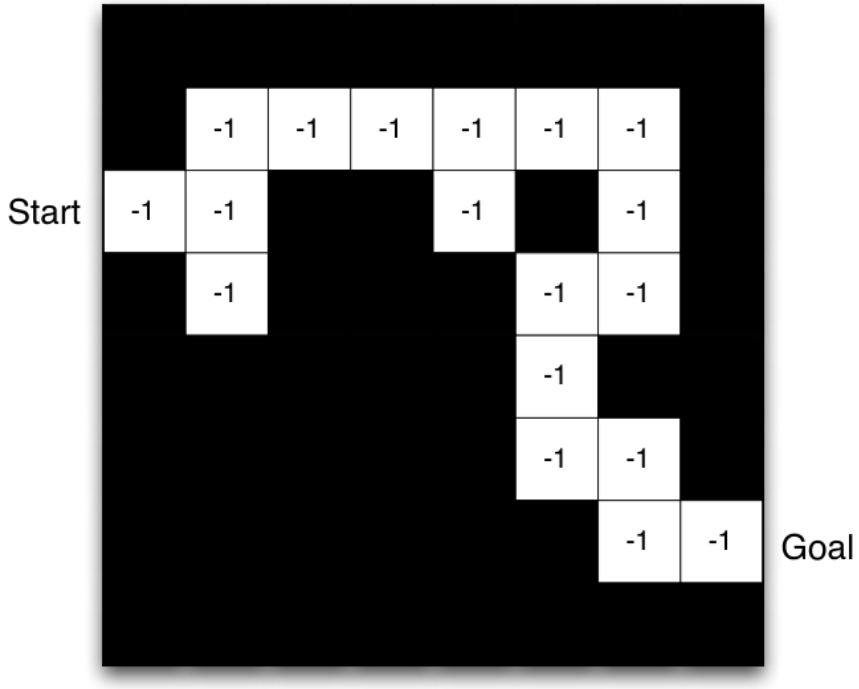
- número que representa $v_\pi(s)$ de cada estado possível

Labirinto: política



- setas representam políticas $\pi(s)$ para cada estado s, i.e. direções para cada posição

Labirinto: modelo



- layout em grid que representa o modelo de transição
- números representam a recompensa imediata a partir de cada estado s

Função Q

$$Q(s_t, a_t) = E[R_t | s_t, a_t]$$

Assuma que a função Q está disponível:

- agente precisa de uma **política** (*policy*) $\pi(s)$ para inferir a melhor ação possível no estado s
- **estratégia básica**?
  - ação que maximiza Q:

$$\pi(s) = \arg\max_a Q(s, a)$$

- **Mas como calcular** Q?

Algoritmos para aprendizado por reforço

Value Learning

- Obter $Q(s, a)$
- $\pi(s) = \arg\max_a Q(s, a)$

Policy Learning

- Obter $\pi(s)$
- Amostrar $a \sim \pi(s)$

Algoritmos para aprendizado por reforço

Value Learning

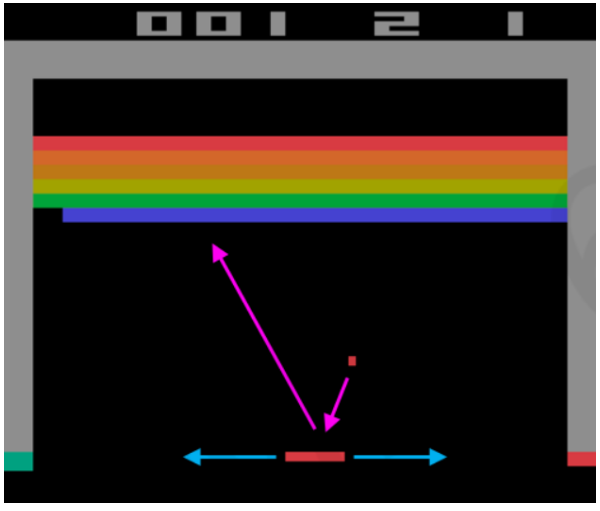
- Encontrar $Q(s, a)$
- $\pi(s) = \arg\max_a Q(s, a)$
- "avaliar o futuro, dada uma política atual"

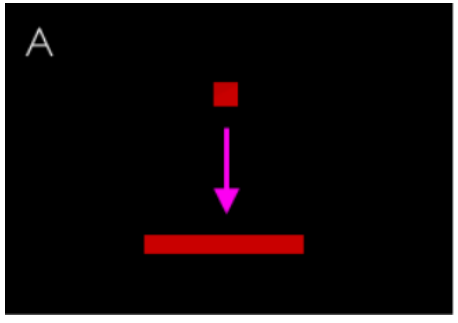
Policy Learning

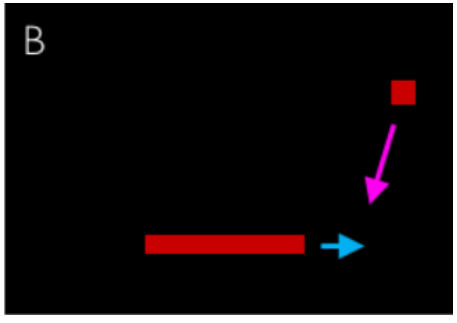
- Encontrar $\pi(s)$
- Amostrar $a \sim \pi(s)$
- "otimizar o futuro, encontrando a melhor política"

Agenda

Value learning: Breakout Atari





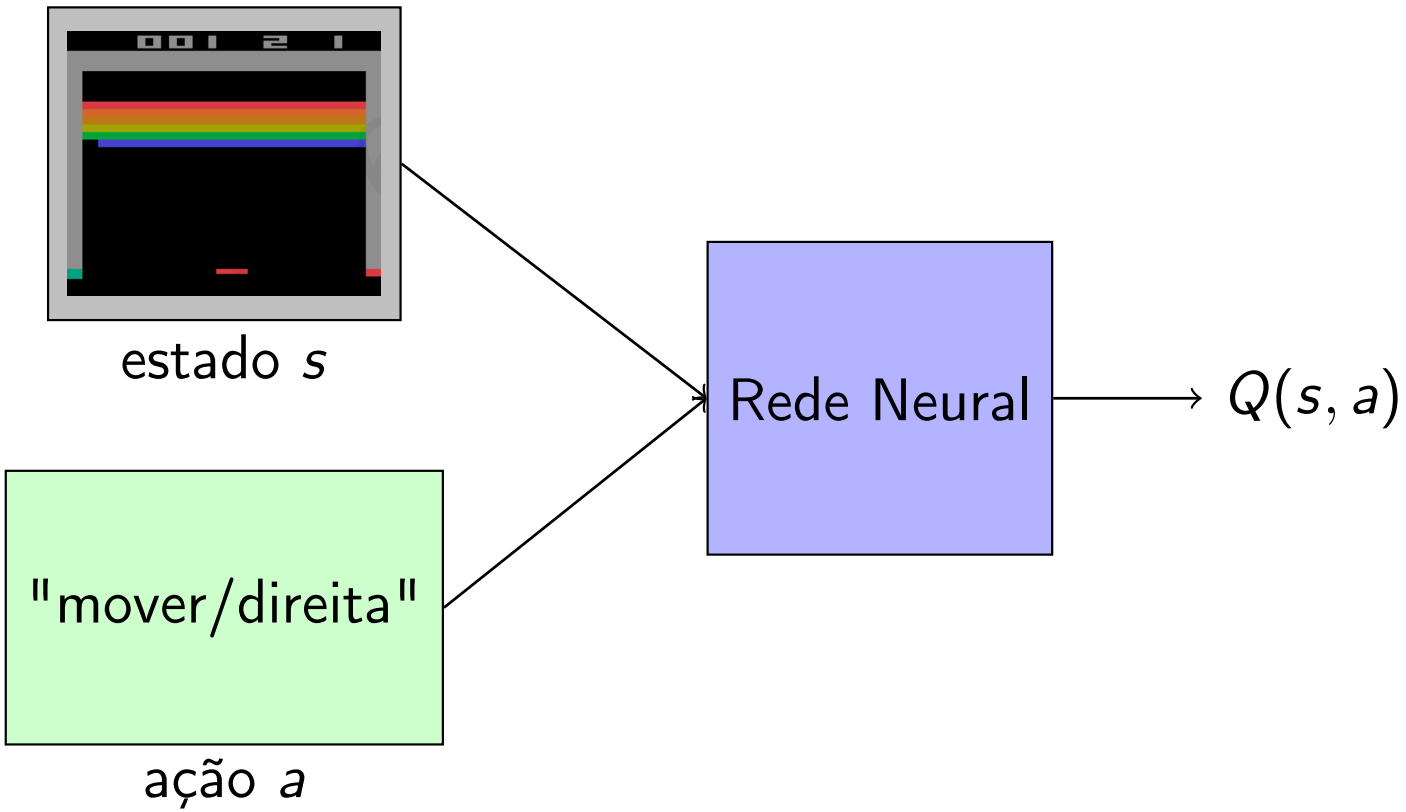


- Qual par (s, a) possui maior valor?

Alexander Amini and Ava Soleimany. MIT 6.S191: Introduction to Deep Learning.

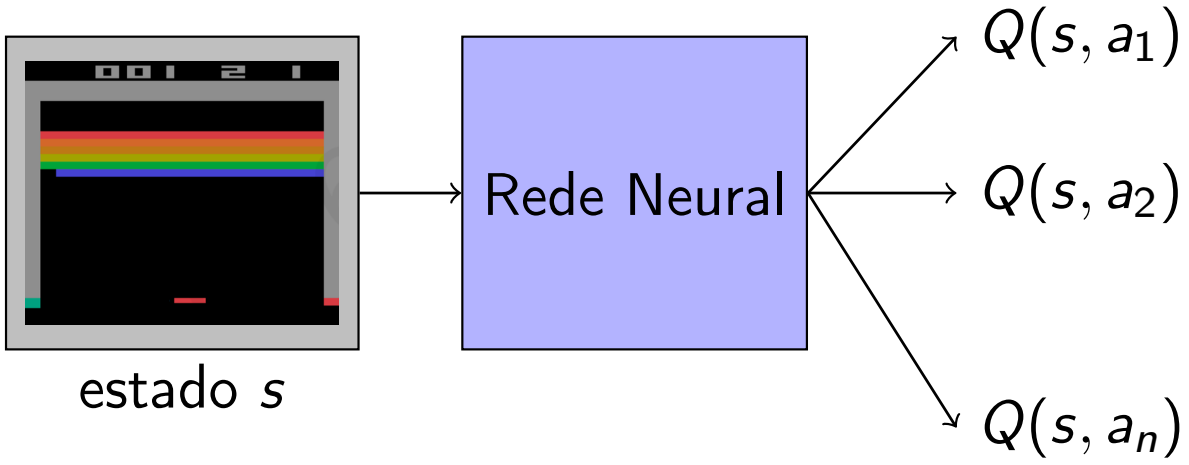
IntroToDeepLearning.com

Q-Learning



- Executa por muitos "episódios" até encontrar um estado terminal
- Ao final teremos a predição do valor para 1 ação avaliada/escolhida por estado

Q-Learning



- Ao final teremos a predição do valor para cada ação avaliada por estado

Q-Learning: treinamento

$$Q(s, a; \Theta) \approx Q^*(s, a)$$

O alvo/target é

$$r + \gamma \max_{a'} Q(s', a')$$

- recompensa atual r
- a recompensa ao executar a melhor ação a' em cada estado sucessor $s \rightarrow s'$

Q-Learning: treinamento

Alvo (exige atingir o estado terminal):

$$r + \gamma \max_{a'} Q(s', a')$$

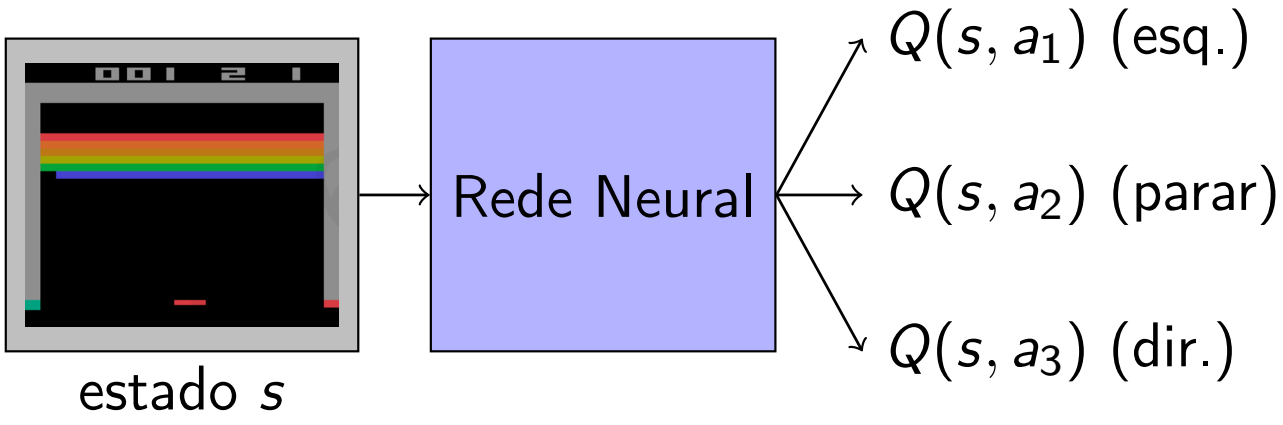
Predição:

$$Q(s, a)$$

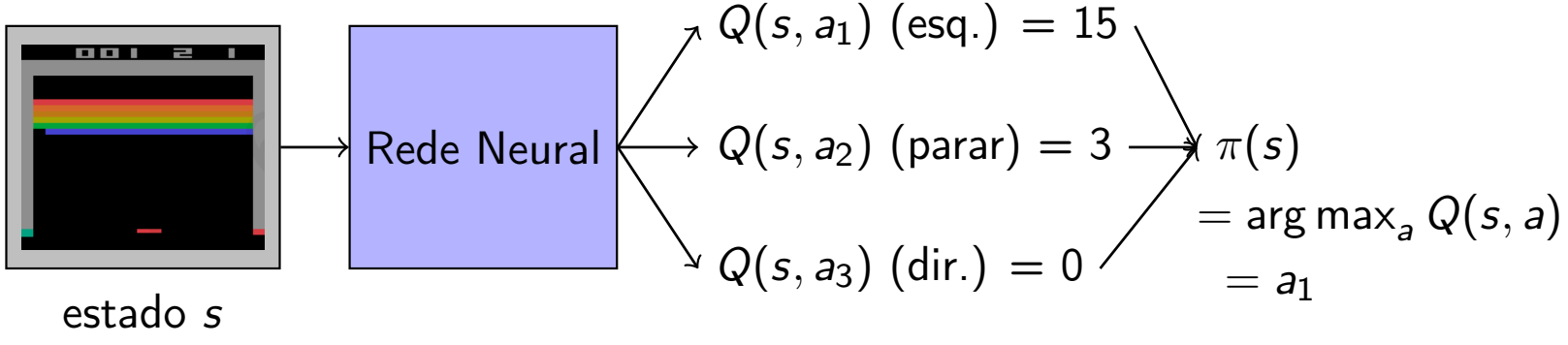
Função de custo/perda

$$\mathcal{L} = E[||r + \gamma \max_{a'} Q(s', a') - Q(s, a)||^2]$$

Q-Learning: treinamento



Q-Learning: treinamento



Deep Q-Networks

Limitações

Complexidade

- Implementa espaços de ação discretos nativamente
- Contínuos são possíveis mas menos naturais nessa abordagem

Flexibilidade

- Política determinística maximizando recompensa via função Q
- Não aprende política estocástica

Agenda

Algoritmos para aprendizado por reforço

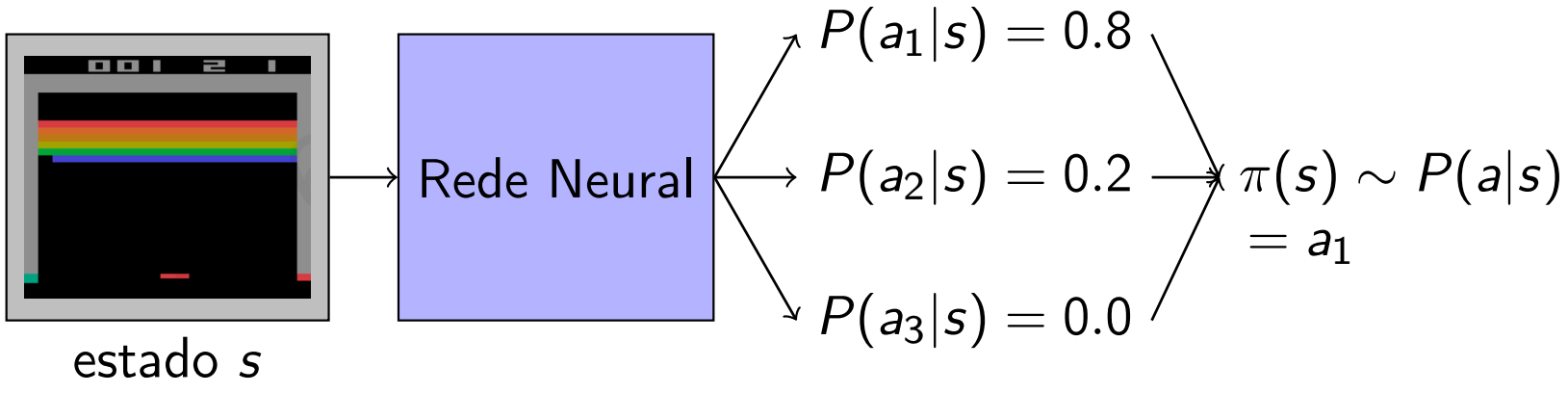
Value Learning

- Encontrar $Q(s, a)$
- $\pi(s) = \arg\max_a Q(s, a)$
- "avaliar o futuro, dada uma política atual"

Policy Learning

- Encontrar $\pi(s)$
- Amostrar $a \sim \pi(s)$
- "otimizar o futuro, encontrando a melhor política"

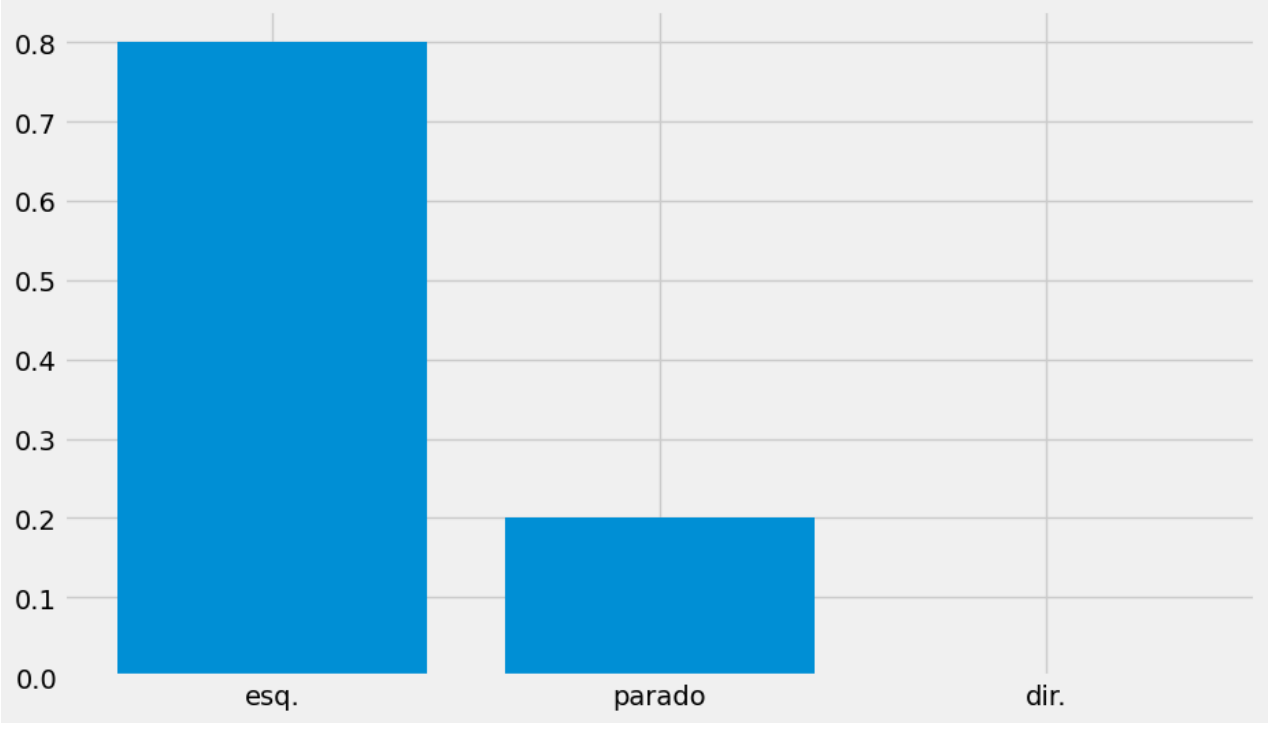
Policy Learning



- Otimizar diretamente a política $\pi(s)$
- Probabilidades de obter o maior valor
- Ainda que a_1 seja mais provável há chances de obter a_2

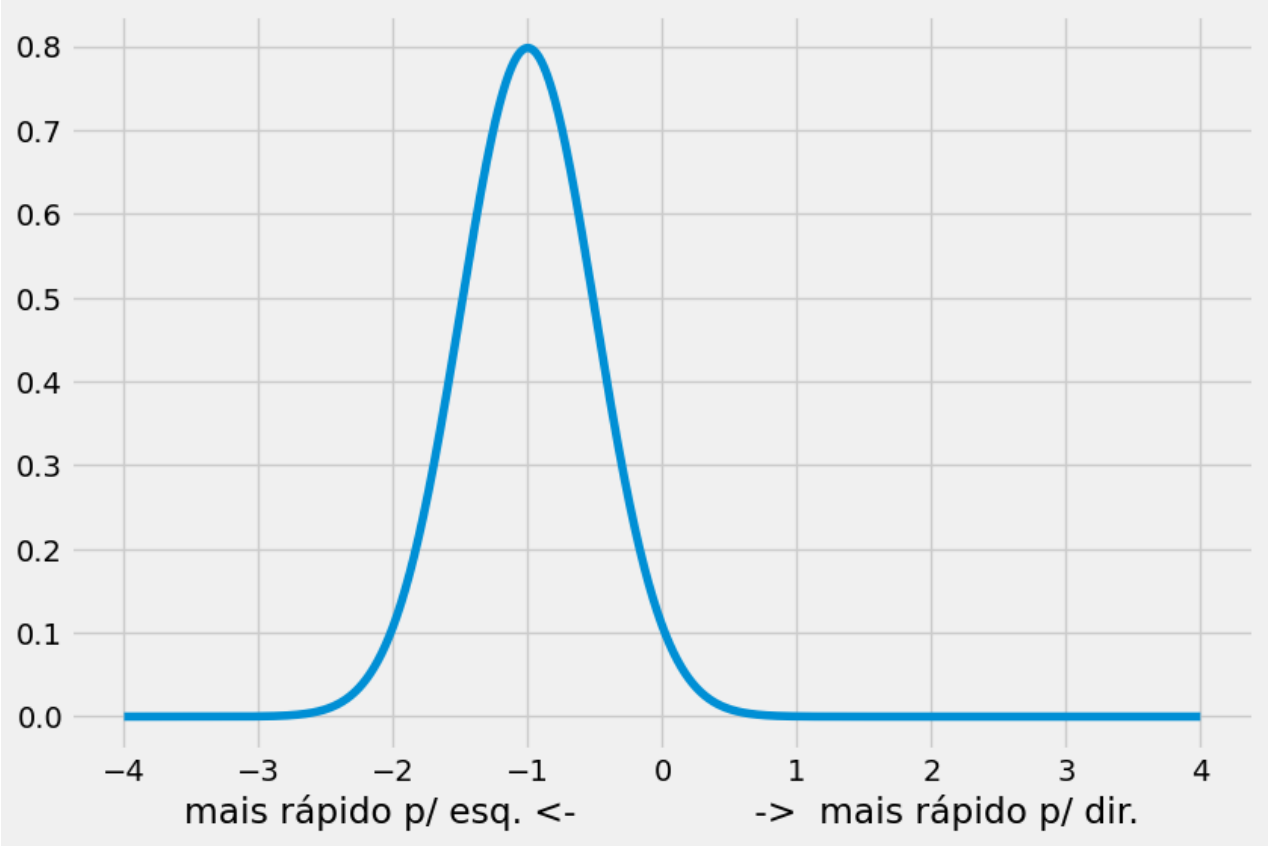
Policy learning: espaço de ações discreto

Em qual direção movimentar?

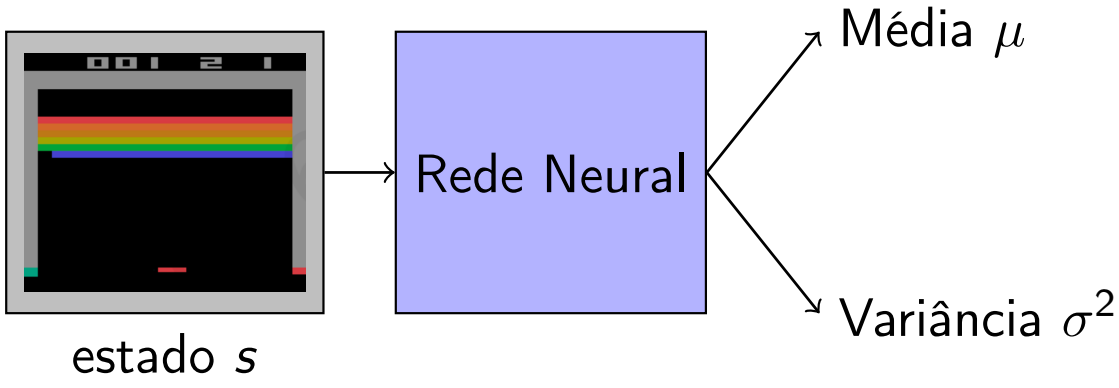


Policy learning: espaço de ações contínuo

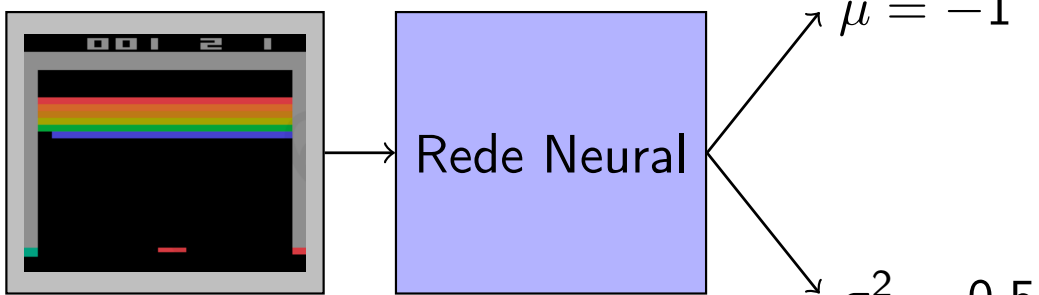
Quão rápido se movimentar?



Policy Learning



Policy Learning



$$P(a|s) = \mathcal{N}(\mu, \sigma^2)$$

$$\pi(s) \sim \mathcal{N}(-1, 0.5)$$

$$= -1.15$$

Policy learning: treinamento com exemplo

Algoritmo de treinamento

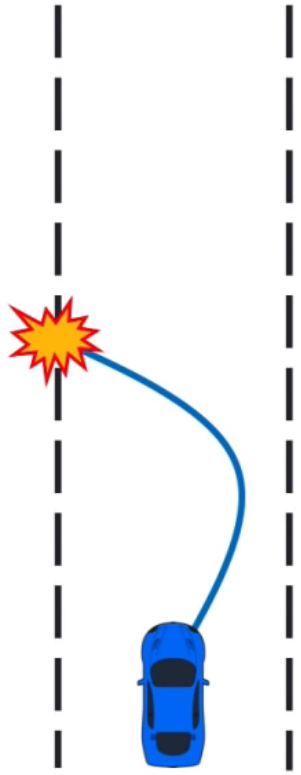
- Inicializar agente
- Executar política até estado terminal
- Armazenar: estados, ações e recompensas
- Reduzir probabilidade de ações com baixo r
- Aumentar probabilidade de ações com alto r

Alexander Amini and Ava Soleimany. MIT 6.S191: Introduction to Deep Learning. IntroToDeepLearning.com

Policy learning: treinamento com exemplo

Algoritmo de treinamento

- Inicializar agente
- Executar política até estado terminal
- Armazenar: estados, ações e recompensas
- Reduzir probabilidade de ações com baixo r
- Aumentar probabilidade de ações com alto r

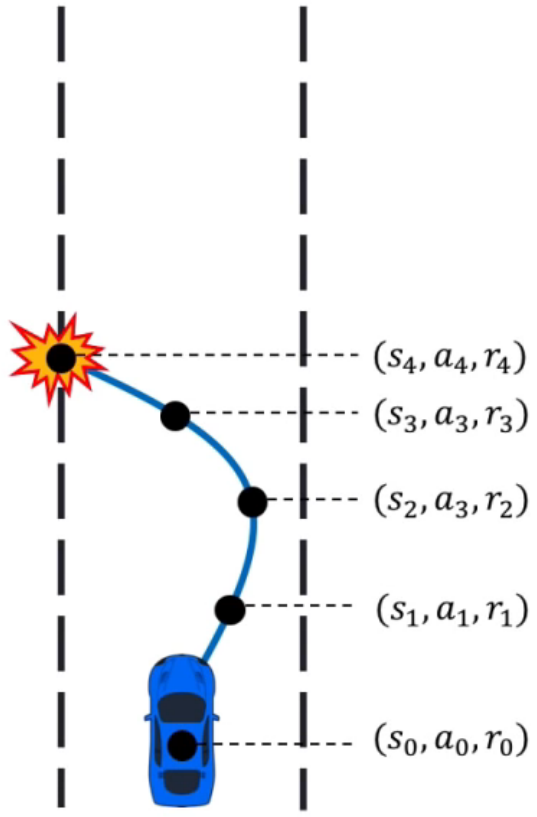


Alexander Amini and Ava Soleimany. MIT 6.S191: Introduction to Deep Learning. IntroToDeepLearning.com

Policy learning: treinamento com exemplo

Algoritmo de treinamento

- Inicializar agente
- Executar política até estado terminal
- Armazenar: estados, ações e recompensas
- Reduzir probabilidade de ações com baixo *r*
- Aumentar probabilidade de ações com alto *r*

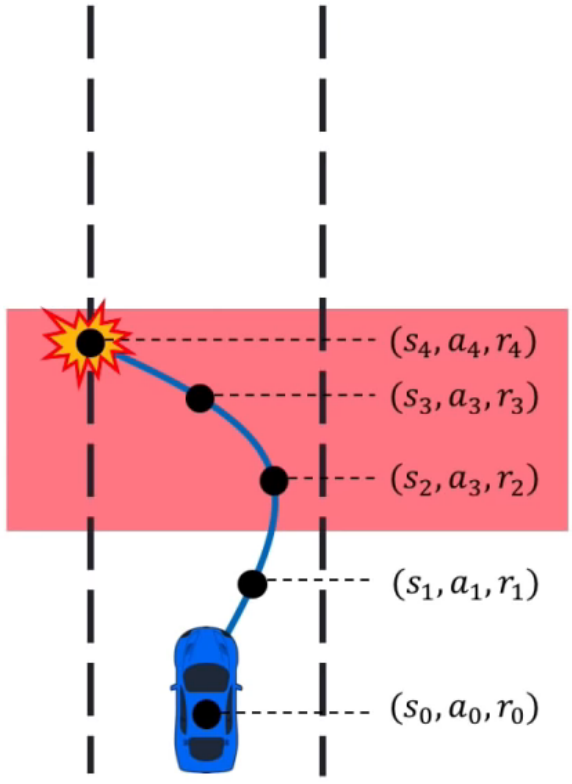


Alexander Amini and Ava Soleimany. MIT 6.S191: Introduction to Deep Learning. IntroToDeepLearning.com

Policy learning: treinamento com exemplo

Algoritmo de treinamento

- Inicializar agente
- Executar política até estado terminal
- Armazenar: estados, ações e recompensas
- Reduzir probabilidade de ações com baixo *r*
- Aumentar probabilidade de ações com alto *r*

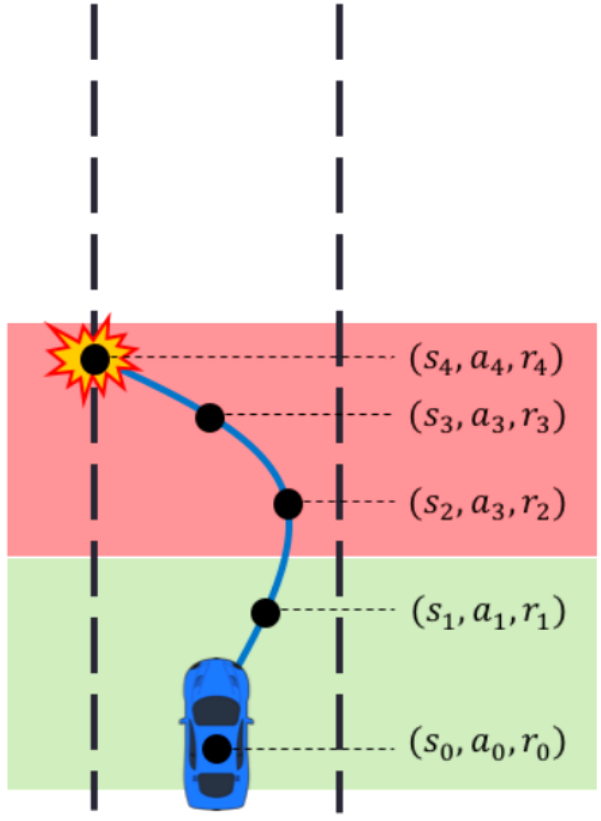


Alexander Amini and Ava Soleimany. MIT 6.S191: Introduction to Deep Learning. IntroToDeepLearning.com

Policy learning: treinamento com exemplo

Algoritmo de treinamento

- Inicializar agente
- Executar política até estado terminal
- Armazenar: estados, ações e recompensas
- Reduzir probabilidade de ações com baixo *r*
- Aumentar probabilidade de ações com alto *r*

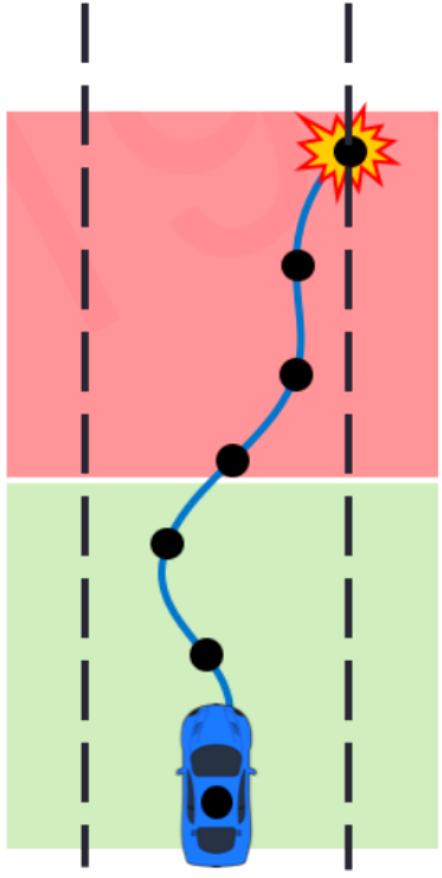


Alexander Amini and Ava Soleimany. MIT 6.S191: Introduction to Deep Learning. IntroToDeepLearning.com

Policy learning: treinamento com exemplo

Algoritmo de treinamento

- Inicializar agente
- Executar política até estado terminal
- Armazenar: estados, ações e recompensas
- Reduzir probabilidade de ações com baixo r
- Aumentar probabilidade de ações com alto r

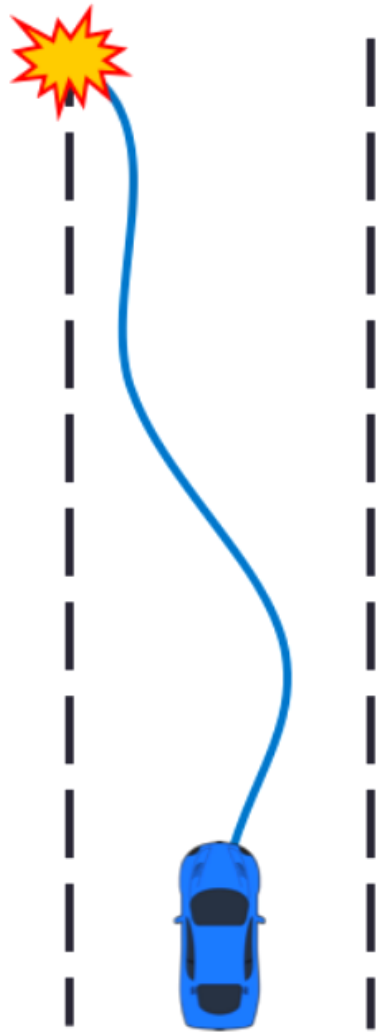


Alexander Amini and Ava Soleimany. MIT 6.S191: Introduction to Deep Learning. IntroToDeepLearning.com

Policy learning: treinamento com exemplo

Algoritmo de treinamento

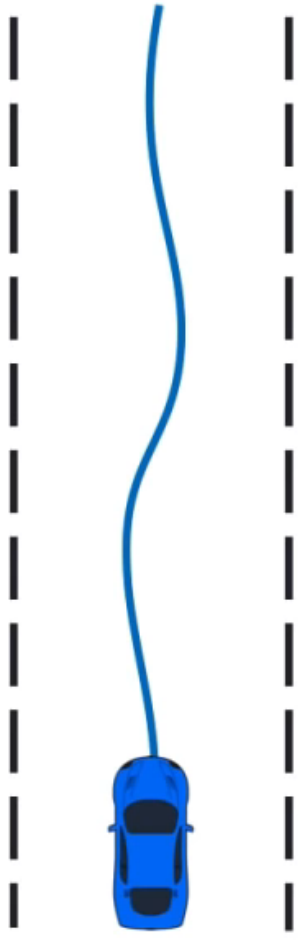
- Inicializar agente
- Executar política até estado terminal
- Armazenar: estados, ações e recompensas
- Reduzir probabilidade de ações com baixo r
- Aumentar probabilidade de ações com alto r



Policy learning: treinamento com exemplo

Algoritmo de treinamento

- Inicializar agente
- Executar política até estado terminal
- Armazenar: estados, ações e recompensas
- Reduzir probabilidade de ações com baixo r
- Aumentar probabilidade de ações com alto r



Alexander Amini and Ava Soleimany. MIT 6.S191: Introduction to Deep Learning. IntroToDeepLearning.com

Policy learning: treinamento com exemplo

Algoritmo de treinamento

- Inicializar agente
- Executar política até estado terminal
- Armazenar: estados, ações e recompensas
- Reduzir probabilidade de ações com baixo r
- Aumentar probabilidade de ações com alto r

Função de perda/custo

$$-\log P(a_t|s_t)R_t$$

- Probabilidades x recompensas

Alexander Amini and Ava Soleimany. MIT 6.S191: Introduction to Deep Learning. IntroToDeepLearning.com

Considerações finais

- Paradigma que pode ser usado em cenários de agentes atuando em ambientes
- Algoritmos gerais: Q-learning x Policy gradients
- Algoritmo Actor Critic: também vale a pena procurar
- Aplicações: robótica, carros autônomos, outros
- Desafio 1: modelar problemas num paradigma menos convencional
- Desafio 2: implementar e colocar em produção sistemas reais

References

- R. Sutton; A. Barto. Reinforcement Learning: an introduction
  http://incompleteideas.net/book/RLbook2020.pdf
- D. Silver. Introduction to Reinforcement Learning
  https://deepmind.com/learning-resources/
  -introduction-reinforcement-learning-david-silver
- A. Karpathy. Deep Reinforcement Learning: Pong from Pixels.
  http://karpathy.github.io/2016/05/31/rl/
- MIT 6.S191. Introduction to Deep Learning.
  http://introtodeeplearning.com/