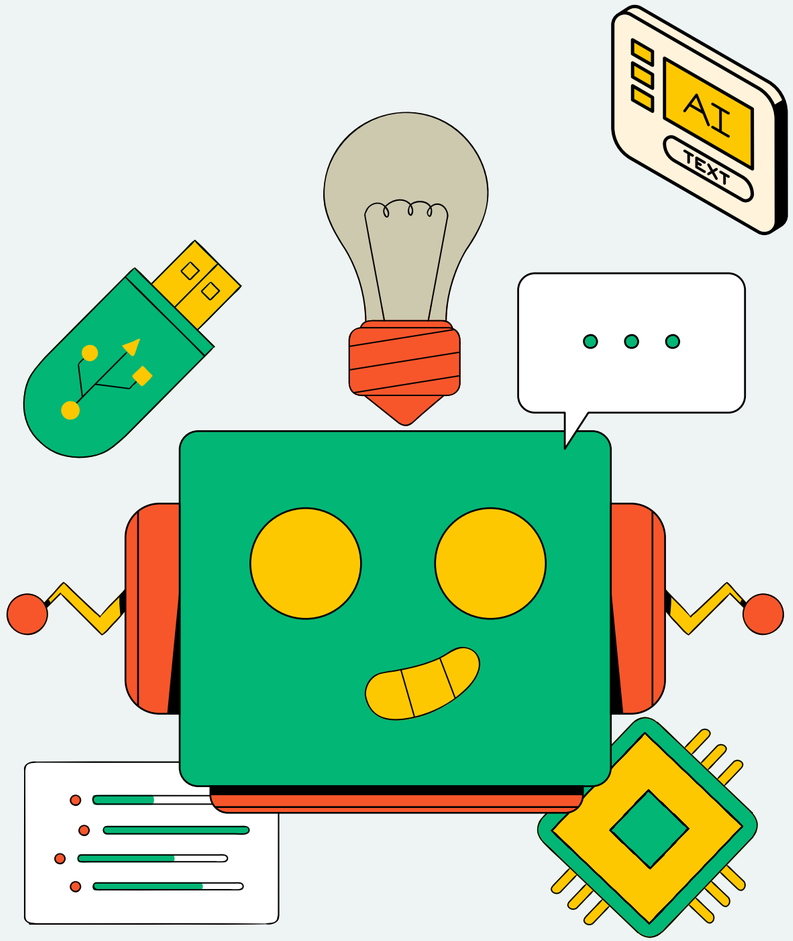
PUC Minas

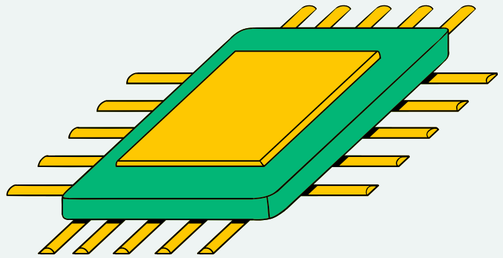


INTELIGÊNCIA ARTIFICIAL

APRENDIZADO POR REFORÇO: DEEP Q-NETWORKS

GRUPO:

ANDRÉ LUIS, BRENO, CAIO, GIUSEPPE, MATHEUS, VINÍCIUS E YAN



TÓPICOS

PUC Minas

- Introdução
- História e Motivação
- Ideia Geral do Algoritmo
- O que possibilitou o DQN?
- Definição Formal do 1º Algoritmo
- Conclusão e Impacto

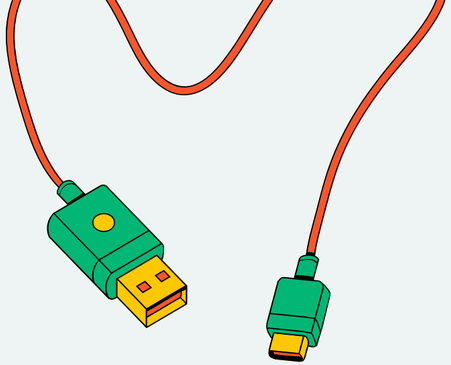
INTRODUÇÃO

PUC Minas

Deep Q-Learning é uma técnica de aprendizado de máquina que une rede neural convolucional com aprendizado por reforço para ensinar uma inteligência artificial a obter sucesso em um ambiente recebendo apenas imagens dele como entrada

Enquanto a **rede** é responsável pela **visualização da imagem, detecção de características e tomada de decisão**, o **aprendizado por reforço** faz a **avaliação da ação executada** por meio da recompensa recebida a cada instante.

HISTÓRIA E MOTIVAÇÃO



PUC Minas

01

ANTES DO DEEP Q-NETWOROK (DQP)

Q-learning é um algoritmo de aprendizado por reforço que ajuda um agente a aprender a tomar decisões ótimas em um ambiente, estimando o valor de cada ação em cada estado. Este, funcionava só com tabelas, inviável para estados de alta dimensão.

02

DEEPMIND (2013-2015)

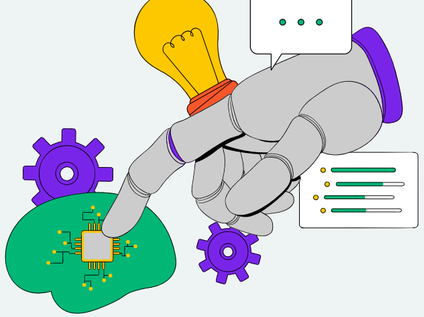
- Artigo inicial: **Playing Atari with Deep Reinforcement Learning** (2013)
- Artigo formal: **Human-level control through deep reinforcement learning** (2015, Nature)

03

CONTRIBUIÇÃO HISTÓRICA

IA aprende a jogar Atari só olhando a tela, sem codificar regras. Essa foi a primeira vez que uma rede neural foi usada com sucesso em um ambiente de aprendizado por reforço de alta complexidade, superando até mesmo jogadores humanos em vários jogos.

IDEIA GERAL DO ALGORITMO



PUC Minas

1. O agente observa uma **imagem** (estado atual do jogo).
2. Essa imagem é processada por uma **CNN (Rede Neural Convolucional)**, que extrai as principais características visuais.
3. A rede retorna os valores estimados Q(s, a) para cada ação possível (Q-Learning).
4. O agente escolhe a ação com maior valor (ou explora aleatoriamente com uma estratégia como ε-greedy).
5. A ação é executada, e o agente recebe uma recompensa e observa o novo estado.
6. A experiência (estado, ação, recompensa, novo estado) é armazenada em uma **memória de replay**.

Durante o treinamento, o agente amostra essas experiências da memória e atualiza os pesos da rede com base na diferença entre o valor estimado e o alvo, calculado com base no Q-learning.

O QUE POSSIBILITOU O DQN?

PUC Minas

EXPERIENCE REPLAY

- **Armazena experiências anteriores** em um buffer.
- Durante o treinamento, **seleciona um mini-batch aleatório**.
- **Quebra correlações temporais** e melhora a eficiência de aprendizado.

TARGET NETWORK:

- Uma **cópia da rede principal** (chamada rede-alvo) é usada para **calcular os valores-alvo de Q**.
- Ela é atualizada com menos frequência, **estabilizando o processo de atualização dos pesos**.

E-GREEDY:

- Estratégia de exploração: com probabilidade ε, o **agente escolhe uma ação aleatória**.
- Com probabilidade 1 – ε, **escolhe a ação com maior valor Q estimado**.
- **ε é decrescido** ao longo do tempo.

DEFINIÇÃO FORMAL DO PRIMEIRO ALGORITMO

PUC Minas

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

INPUT

Imagens ou pixels.

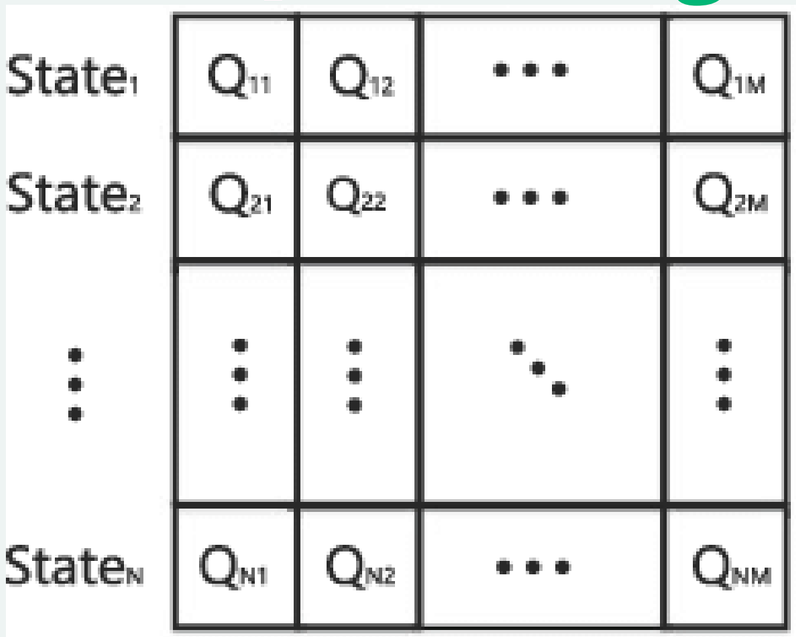
OUTPUT

Ação Ótima

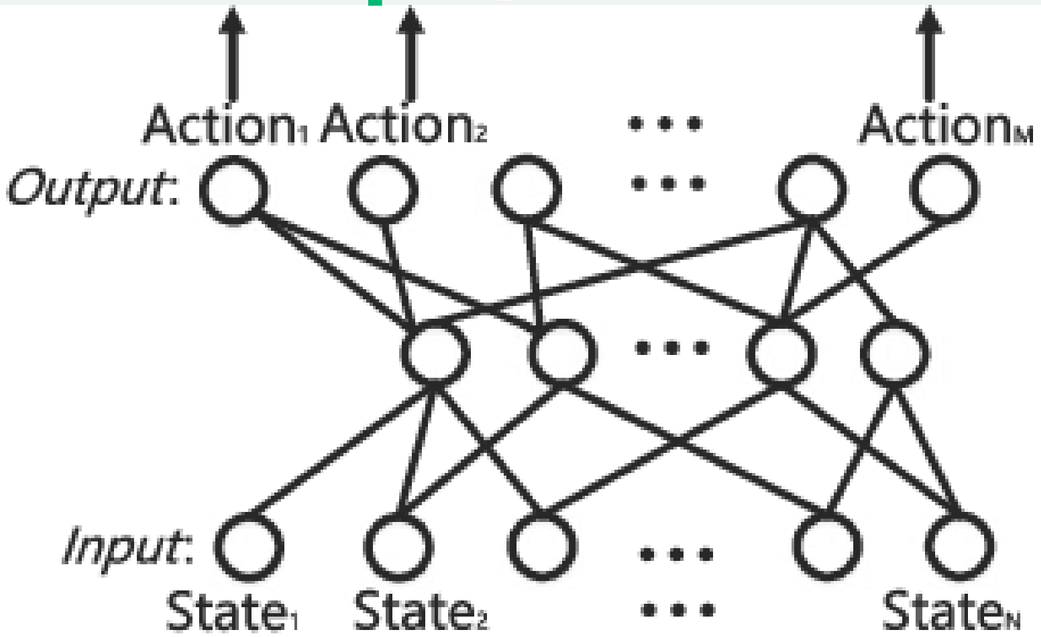
COMPARAÇÃO VISUAL

PUC Minas

Q-Learning



Deep Q-Network



CONCLUSÃO E IMPACTO

PUC Minas

O DQN foi um divisor de águas no Aprendizado por Reforço. Ele mostrou que é possível aplicar redes neurais profundas em tarefas de decisão complexas, mesmo com entradas de alta dimensão como imagens

Double DQN

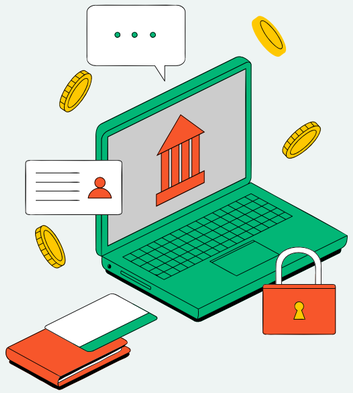
Dueling DQN

Rainbow DQN

Utilizando o DQN: **Notebook - Colab**

PUC Minas

REFERÊNCIAS BIBLIOGRÁFICAS



Artigo de 2013 – DeepMind:
MNHI, Volodymyr et al. Playing Atari with Deep Reinforcement Learning. arXiv preprint arXiv:1312.5602, 2013. Disponível em: https://arxiv.org/abs/1312.5602. Acesso em: 31 maio 2025.

Artigo de 2015 – DeepMind (Nature):
MNHI, Volodymyr et al. Human-level control through deep reinforcement learning. Nature, v. 518, p. 529–533, 2015. DOI: 10.1038/nature14236. Disponível em: https://www.nature.com/articles/nature14236. Acesso em: 31 maio 2025.

Trabalho de Conclusão de Curso – USP (2018):
TAVARES, Vikttor Cruz. Uma introdução ao Aprendizado por Reforço Profundo: estudo e implementação do algoritmo Deep Q-Network. Universidade de São Paulo, Instituto de Matemática e Estatística, 2018. Disponível em: https://bccdev.ime.usp.br/tccs/2018/viktt/monografia_revisada.pdf. Acesso em: 31 maio 2025.

PUC Minas

MUITO OBRIGADO

André Luis Silva
Breno Pires
Caio Diniz
Giuseppe Cordeiro
Matheus Fagundes
Vinícius Miranda
Yan Sabarense