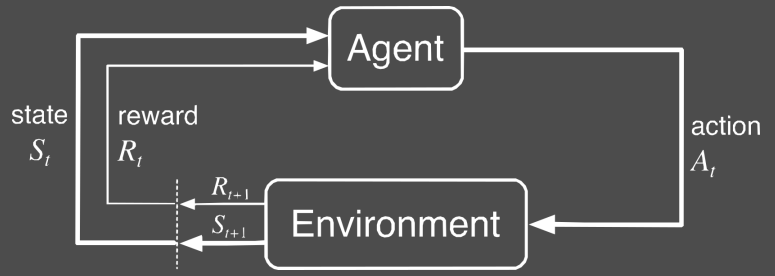
Aprendizado por Reforço

AULA - 3

Diferença Temporal e Deep RL

Retrospectiva do último episódio

- Formalização da relação Agente-Ambiente
  - Transição
  - Dinâmicas do Ambiente



$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$$

- Retorno com Desconto (*Discounted Return*)

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

$$R_t = r_{t+1} + \gamma R_{t+1}$$

- Função de Valor

$$V_\pi(s) = \mathbb{E}_\pi[R_t | s_t = s]$$

$$Q_\pi(s, a) = \mathbb{E}_\pi[R_t | s_t = s, a_t = a]$$

Retrospectiva do último episódio

- Equação de Bellman

$$V_\pi(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r_{t+1} + \gamma V_\pi(s')]$$

- V, Q, e *pi* ótimos

$$V_*(s) = \max_a \sum_{s',r} p(s', r|s, a)[r_{t+1} + \gamma V_*(s')]$$

$$Q_*(s, a) = \sum_{s',r} p(s', r|s, a)[r_{t+1} + \gamma \max_{a'} Q_*(s_{t+1}, a')]$$

Temporal-Difference Learning

Temporal-Difference Learning

- Forma de atualizar sua estimativa de valor
- Objetivo: Estimar o retorno
- Quanto eu errei na minha estimativa do retorno?
- Quanto desse erro eu vou usar para atualizar minha estimativa atual?

$$V(s) \leftarrow V(s) + \alpha[R_t - V(s)]$$

TD Learning

- Métodos de Monte Carlo

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots + \gamma^{T-(t+1)} r_T$$

- Bootstrapping

$$R_t = r_{t+1} + \gamma V(s_{t+1})$$

$$V(s) \leftarrow V(s) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s)]$$

- Permite atualização após um único passo no ambiente
- Análogo para Q(s,a)

SARSA

- On-Policy
  - Atualizações feitas com experiências adquiridas a partir da política atual

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

$$(s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1})$$

- Política (exemplo): ε-greedy
  - Há uma probabilidade ε de se escolher uma ação aleatória e (1-ε) de se seguir Q(s,a)

Q-Learning

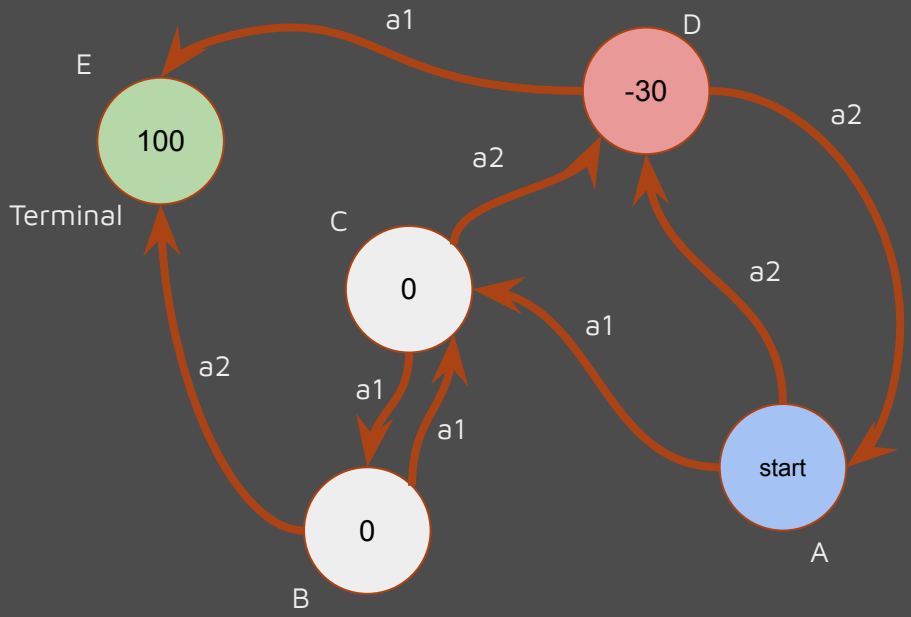
- Off-Policy
  - Atualizações feitas com experiências adquiridas a partir de qualquer política

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

- Independe da política de decisão
  - Presume-se que a melhor ação sempre será escolhida
- Política ainda define os estados visitados
- Requerimento para Convergência em Q^*:
  - Que todos os estados continuem a ser visitados

Q-Learning

$$Q = \begin{matrix} \\ a1 \\ a2 \end{matrix} \begin{matrix} \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$



Faremos algumas interações na mão

$$Q = \begin{matrix} \\ a1 \\ a2 \end{matrix} \begin{matrix} \begin{matrix} A & B & C & D & E \end{matrix} \\ \begin{bmatrix} 81 & 81 & 90 & 100 & - \\ 51 & 100 & 51 & 72.9 & - \end{bmatrix} \end{matrix}$$

Q-Learning (Algoritmo)

- Inicializar Q(s,a) com zeros
- *while(t<MAX_STEPS)*
  - processar estado (s)
  - selecionar ação (a)
  - executar a no ambiente
  - t=t+1
  - receber recompensa (r) e estado (s’)
  - atualizar Q(s,a)
  - s=s’
- *end*

Dimensionalidade

- Quanto mais estados, e mais ações:
- Maior é a nossa tabela Q
- Mais demorado fica de explorar todas as combinações
- *Curse of Dimensionality*

- Solução: Aproximadores de Funções não Lineares?

Deep RL

Redes Neurais Artificiais

+

Aprendizado por Reforço

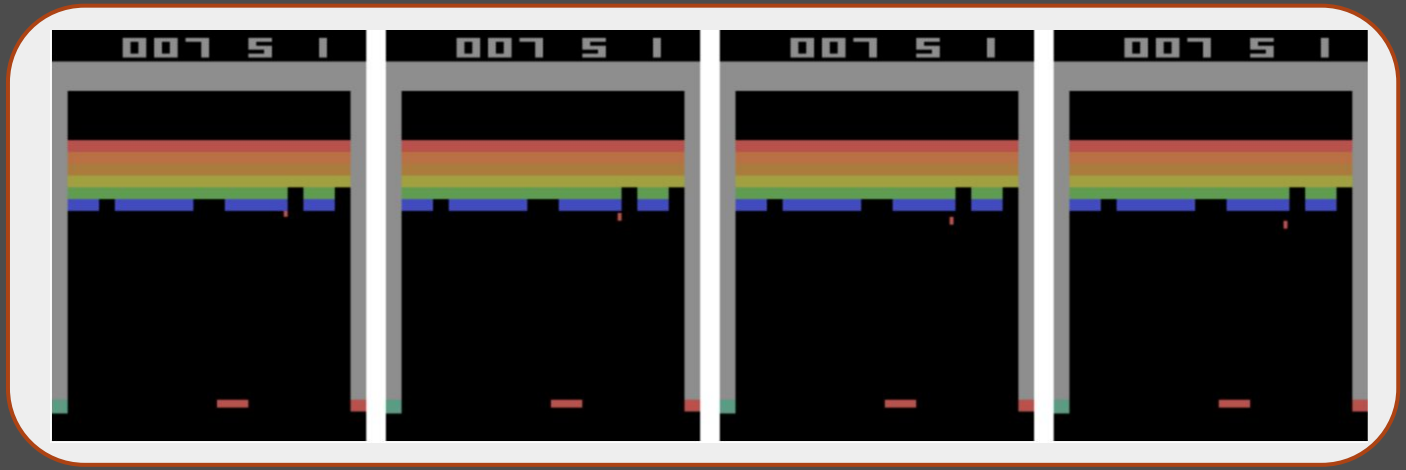
NÃO FUNCIONA

Por que não funciona?

- Instabilidade no treinamento
- Lembrando da representatividade necessária em problemas de aprendizado supervisionado
  - Batches aleatórios
  - Amostra representativa dos dados reais

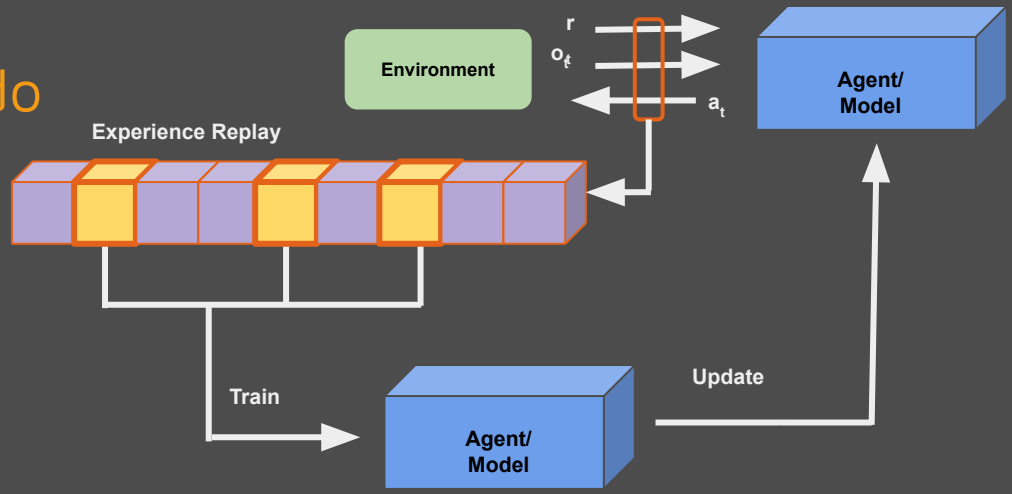
De onde vem a instabilidade?

- Em reforço as tarefas são sequenciais
- Observações subsequentes são muito parecidas
- Gradientes da função de perda ficam muito parecidos



Como Resolver?

- **Experience Replay**
- Coleta de experiências/transições
- Armazenamento no *Buffer*
- Coleta aleatória de dados do *Buffer*
- Treino da rede com batch

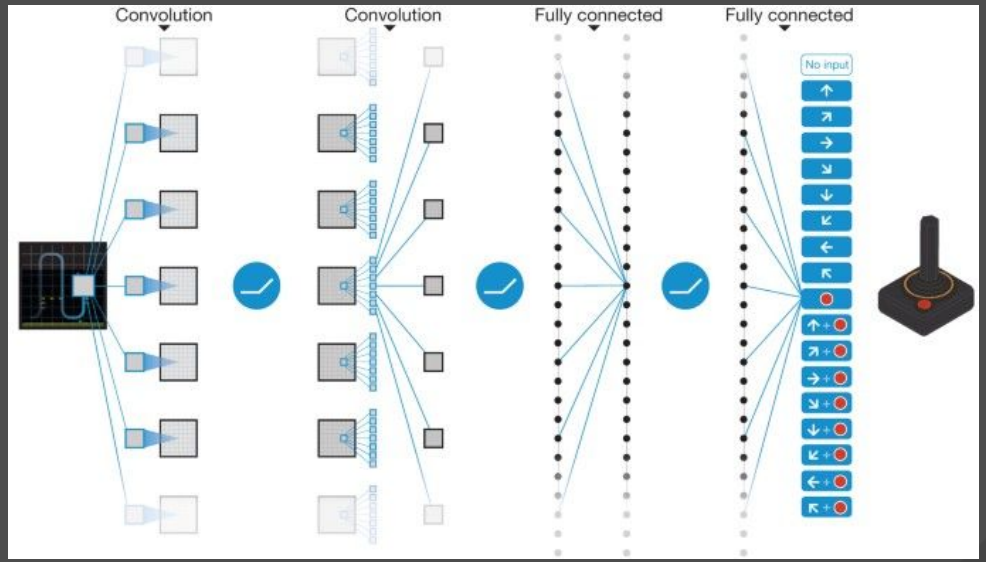


Agora Funciona?

Não, ainda não...

Deep Q Network

- Aprender jogos de ATARI 2600 com RL
  - Estados representados por imagens
    - Frames empilhados para noções de velocidade e direção
  - Ações discretas



Human-level control through deep reinforcement learning (Mnih, 2015)
https://www.nature.com/articles/nature14236

Playing Atari with deep reinforcement learning (Mnih, 2013)
https://www.cs.toronto.edu/~vmnih/docs/dqn.pdf

Gradiente

- Como calcular o gradiente para atualizar os pesos da rede?

ALVO PREDIÇÃO

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)]$$

ERRO

- Função de perda como expectativa do erro

$$L(\theta) = \left(r + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta) \right)^2$$

The Target Network

- Alvo e Predição dependem dos pesos (theta)

$$L(\theta) = \left(r + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta) \right)^2$$

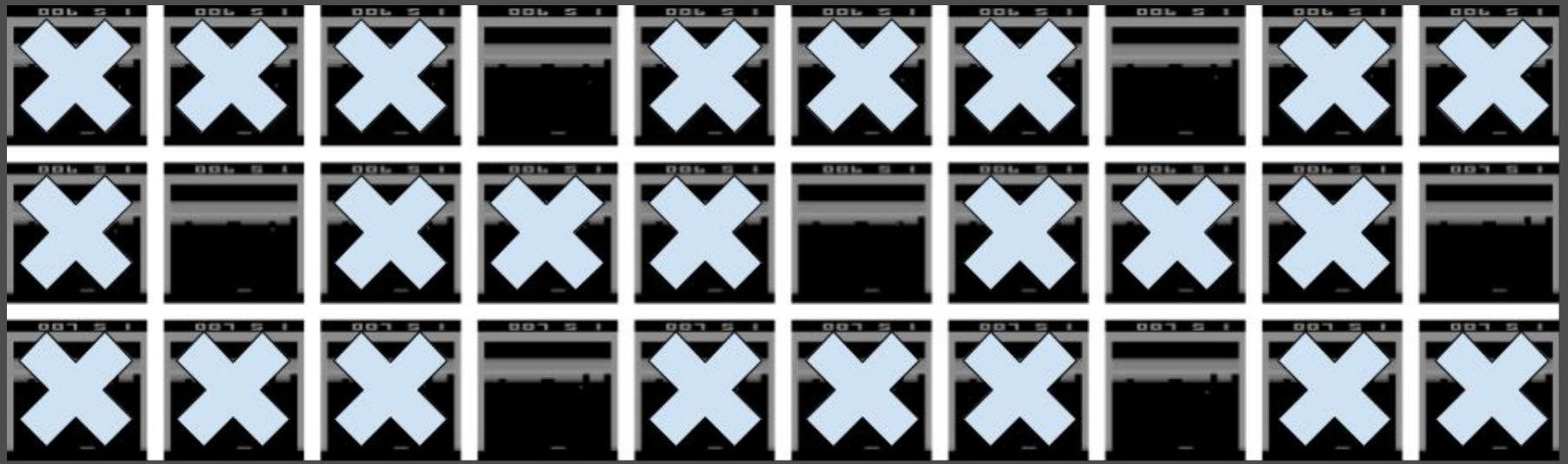
- Uma mudança nos parâmetros muda o alvo
- Target Network (phi) = uma versão anterior de (theta)

$$L(\theta) = \left(r + \gamma \max_{a'} Q(s', a'; \phi) - Q(s, a; \theta) \right)^2$$

The Target Network

- Atualizada regularmente (a cada C passos)
- Atualizações mais espaçadas
  - Alvos mais estáveis/fixos
  - Alvos desatualizados/Incorretos
- Atualizações menos espaçadas
  - Alvos mais instáveis
  - Alvos mais corretos

Frame Skipping



Frame Skipping

- Frames mais espaçados = Menos parecidos
- Tempo para ação “ter efeito” no ambiente
- Acelera k vezes a execução de episódios
  - É mais rápido executar o ambiente que executar a rede

DQN

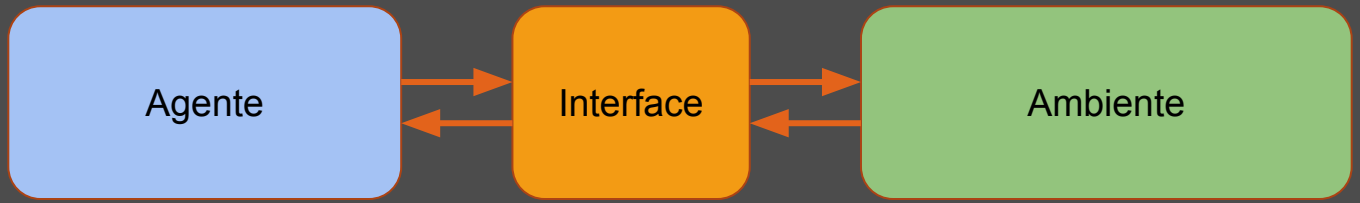
- Experience Replay
- Target Network
- Frame Skipping
- Frame Stacking

Problemas do DQN

- Tamanho do Buffer
  - ~1 milhão de experiências
  - Alto uso de RAM
- Tempo de convergência
  - Treinar camadas convolucionais é caro
  - Demora para que experiências novas encham o buffer

OpenAI Gym

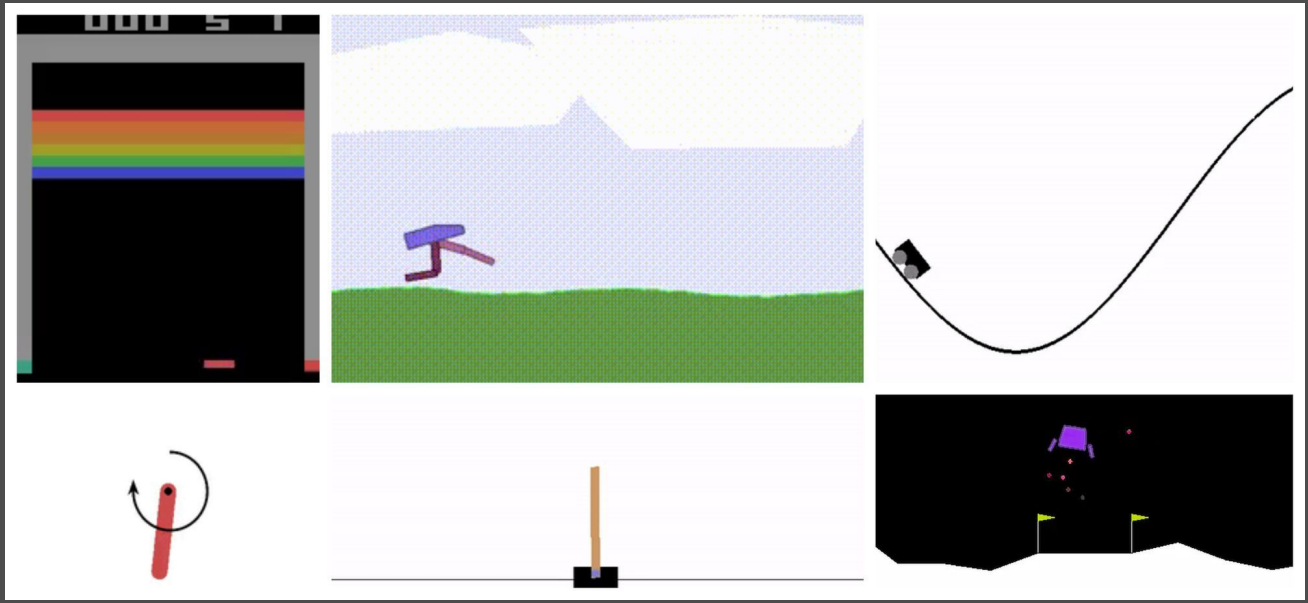
Interface Padronizada Agente-Ambiente




```
env = gym.make("Breakout-v0")
env.action_space.n
Out[...]: 4
env.env.get_action_meanings()
Out[...]: ['NOOP', 'FIRE', 'RIGHT', 'LEFT']
env.observation_space
Out[...]: Box(210, 160, 3)

obs = env.reset()
reward, next_obs = env.step(actions)
env.render() # show video
```

Ambientes Prontos



The End

Richard S. Sutton and Andrew G. Barto - Reinforcement Learning: An Introduction - Second Edition

Capítulo 6 (TD-Learning)