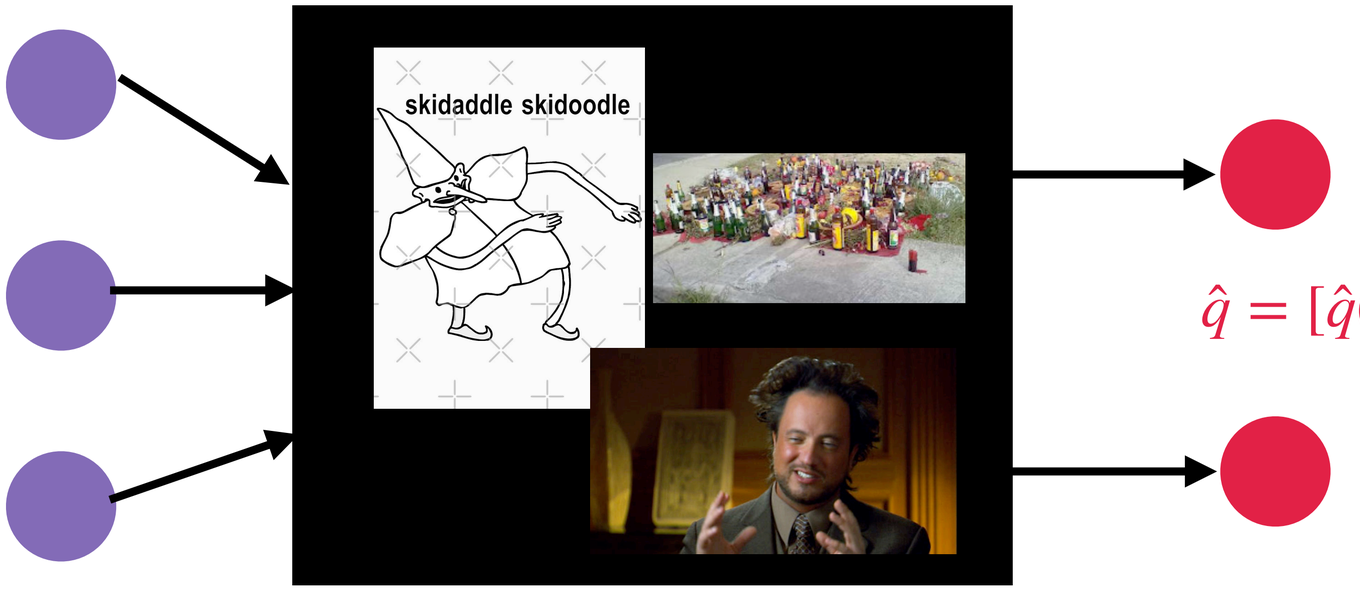
Deep Q-Learning

Deep Q-Learning

- Q-Learning + Redes Neurais

$s = [s_1, s_2]$



$\hat{q} = [\hat{q}(s, a_1), \hat{q}(s, a_2), \hat{q}(s, a_3)]$

Deep Q-Learning

- Q-Learning + Redes Neurais
- Estado s como um input, valores $\hat{q}(s, a)$ como outputs
- Como no Deep SARSA, a alvo é lidar com estados contínuos
- *Off Policy*

Deep Q-Learning

Algorithm 1 Deep Q-Learning

Input: α learning rate, ϵ random action probability,
    γ discount factor,
Initialize q-value parameters θ and target parameters $\theta_{targ} \leftarrow \theta$
$b \leftarrow \epsilon$-greedy policy w.r.t $\hat{q}(s, a|\theta)$
$\pi \leftarrow$ greedy policy w.r.t $\hat{q}(s, a|\theta)$
Initialize replay buffer B
for episode $\in 1..N$ **do**
    Restart environment and observe the initial state S_0
    for $t \in 0..T-1$ **do**
        Select action $A_t \sim b(S_t)$
        Execute action A_t and observe S_{t+1}, R_{t+1}
        Insert transition $(S_t, A_t, R_{t+1}, S_{t+1})$ into the buffer B
        $K = (S, A, R, S') \sim B$
        Select actions $A' \sim \pi(S')$
        Compute loss function over the batch of experiences:

$$L(\theta) = \frac{1}{|K|} \sum_{i=1}^{|K|} \left[R_i + \gamma \hat{q}(S'_i, A'_i|\theta_{targ}) - \hat{q}(S_i, A_i|\theta)\right]^2 \quad (1)$$

    end for
    Every k episodes synchronize $\theta_{targ} \leftarrow \theta$
end for
Output: Near optimal policy π and q-value approximations $\hat{q}(s, a|\theta)$

Deep Q-Learning

Algorithm 1 Deep Q-Learning

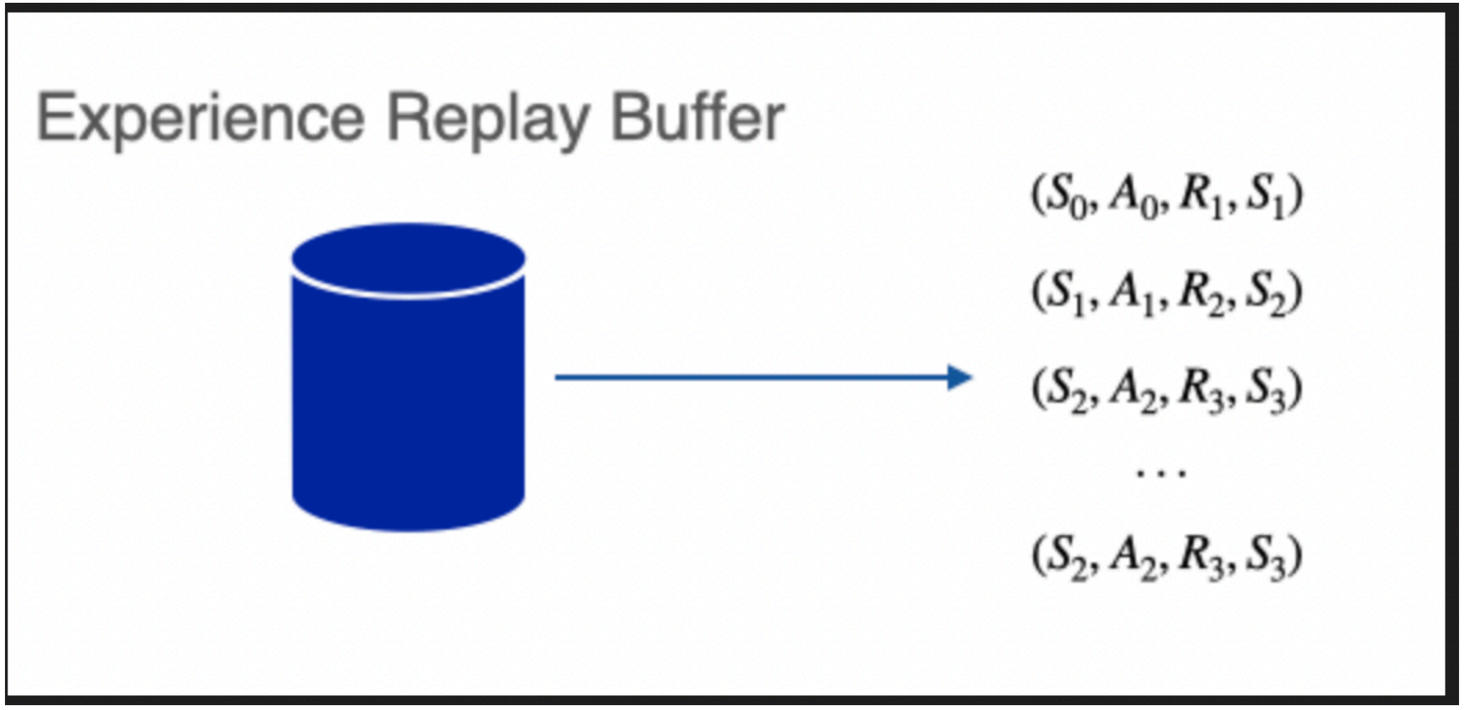
Input: α learning rate, ϵ random action probability,
    γ discount factor,
Initialize q-value parameters θ and target parameters $\theta_{targ} \leftarrow \theta$
$b \leftarrow \epsilon$-greedy policy w.r.t $\hat{q}(s, a|\theta)$
$\pi \leftarrow$ greedy policy w.r.t $\hat{q}(s, a|\theta)$
Initialize replay buffer B
for episode $\in 1..N$ **do**
    Restart environment and observe the initial state S_0
    for $t \in 0..T-1$ **do**
        Select action $A_t \sim b(S_t)$
        Execute action A_t and observe S_{t+1}, R_{t+1}
        Insert transition $(S_t, A_t, R_{t+1}, S_{t+1})$ into the buffer B
        $K = (S, A, R, S') \sim B$
        Select actions $A' \sim \pi(S')$
        Compute loss function over the batch of experiences:

$$L(\theta) = \frac{1}{|K|} \sum_{i=1}^{|K|} [R_i + \gamma \hat{q}(S'_i, A'_i|\theta_{targ}) - \hat{q}(S_i, A_i|\theta)]^2 \quad (1)$$

    end for
    Every k episodes synchronize $\theta_{targ} \leftarrow \theta$
end for
Output: Near optimal policy π and q-value approximations $\hat{q}(s, a|\theta)$

Memória de replay

- Para atualizar a rede neural, escolhe-se aleatoriamente um lote de transições da memória de replay



Rede Neural Alvo

- Fazemos uma copia da rede neural para calcular o alvo, essa rede não é atualizada com o gradiente.

$$\theta_{targ} \leftarrow \theta$$

$$L(\theta) - \frac{1}{|K|} \sum_{i=1}^{|K|} [R_i + \gamma \hat{q}(S'_i, A'_i \,|\, \theta_{targ}) - \hat{q}(S_i, A_i \,|\, \theta)]^2$$

$$\theta_{targ} \leftarrow \theta_k$$