Deep SARSA

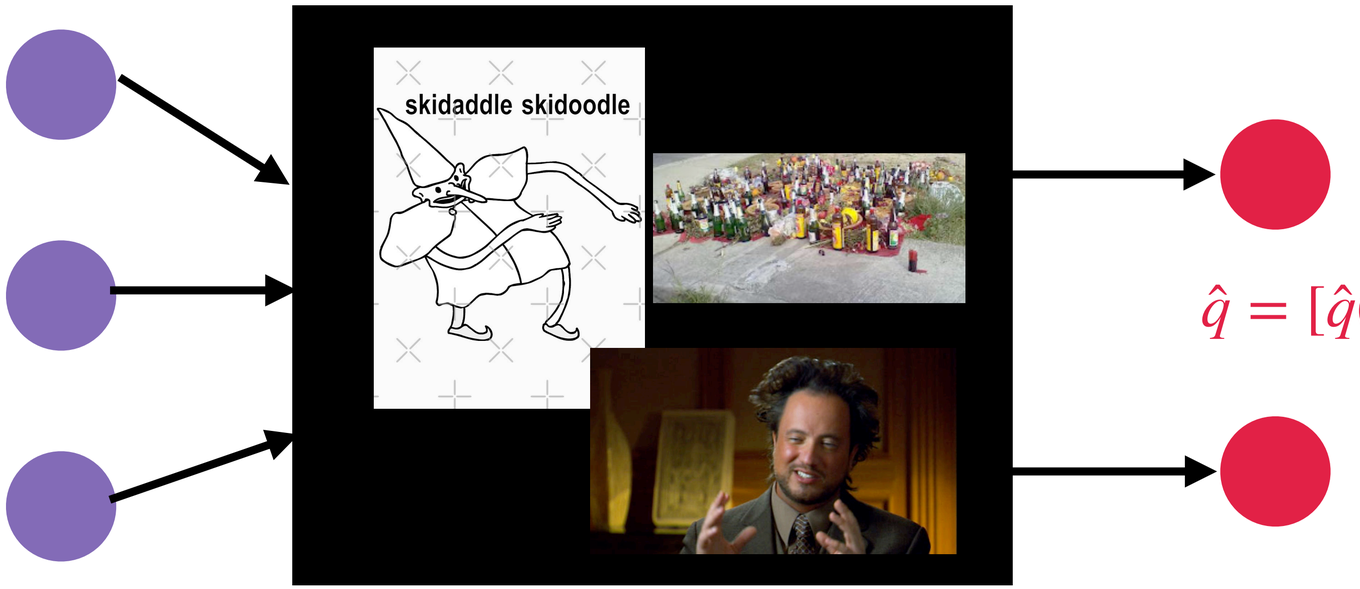
Introdução a redes Neurais

Informações sobre
O ambiente

Probabilidade de
Ações

Introdução a redes Neurais

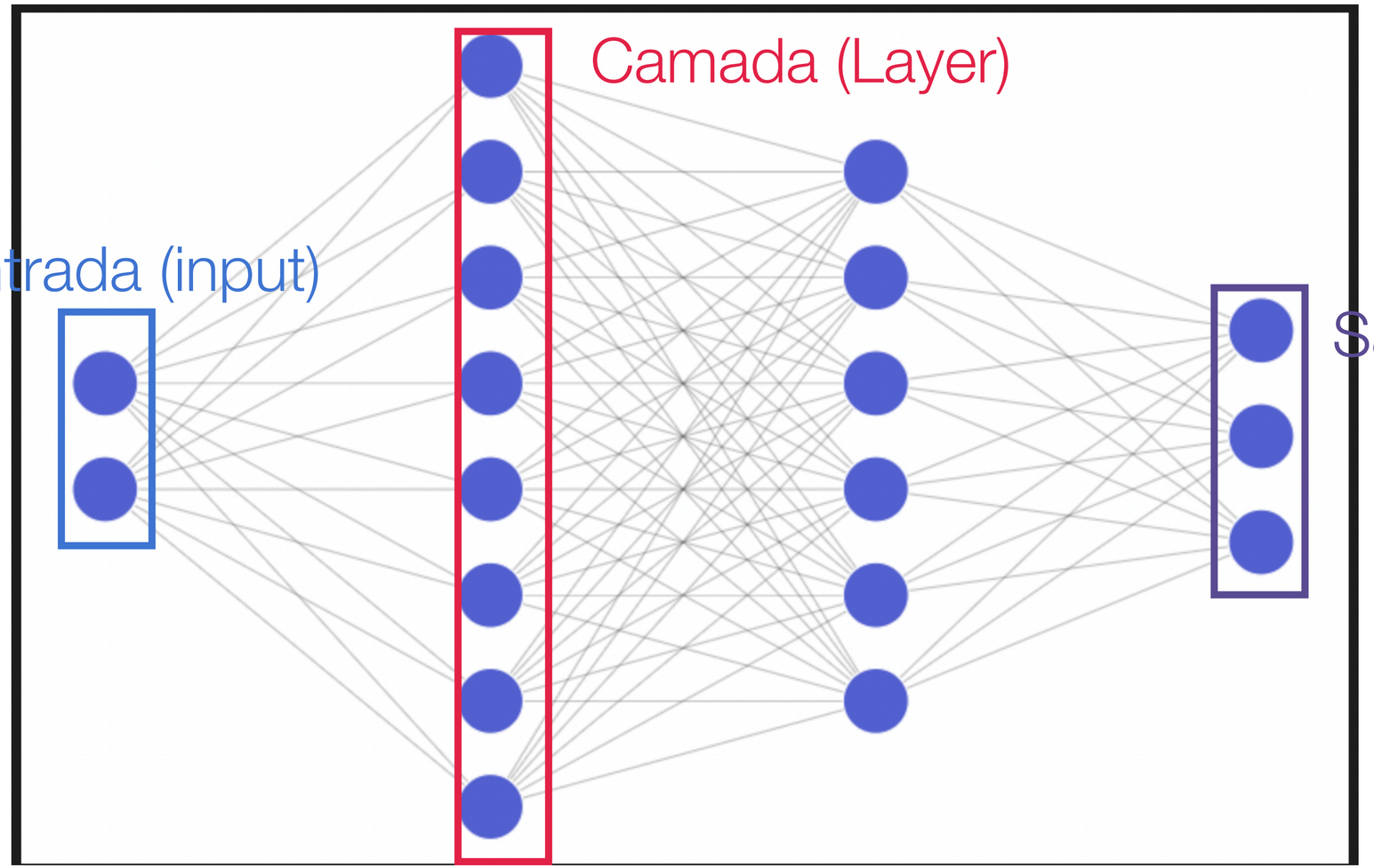
$s = [s_1, s_2]$



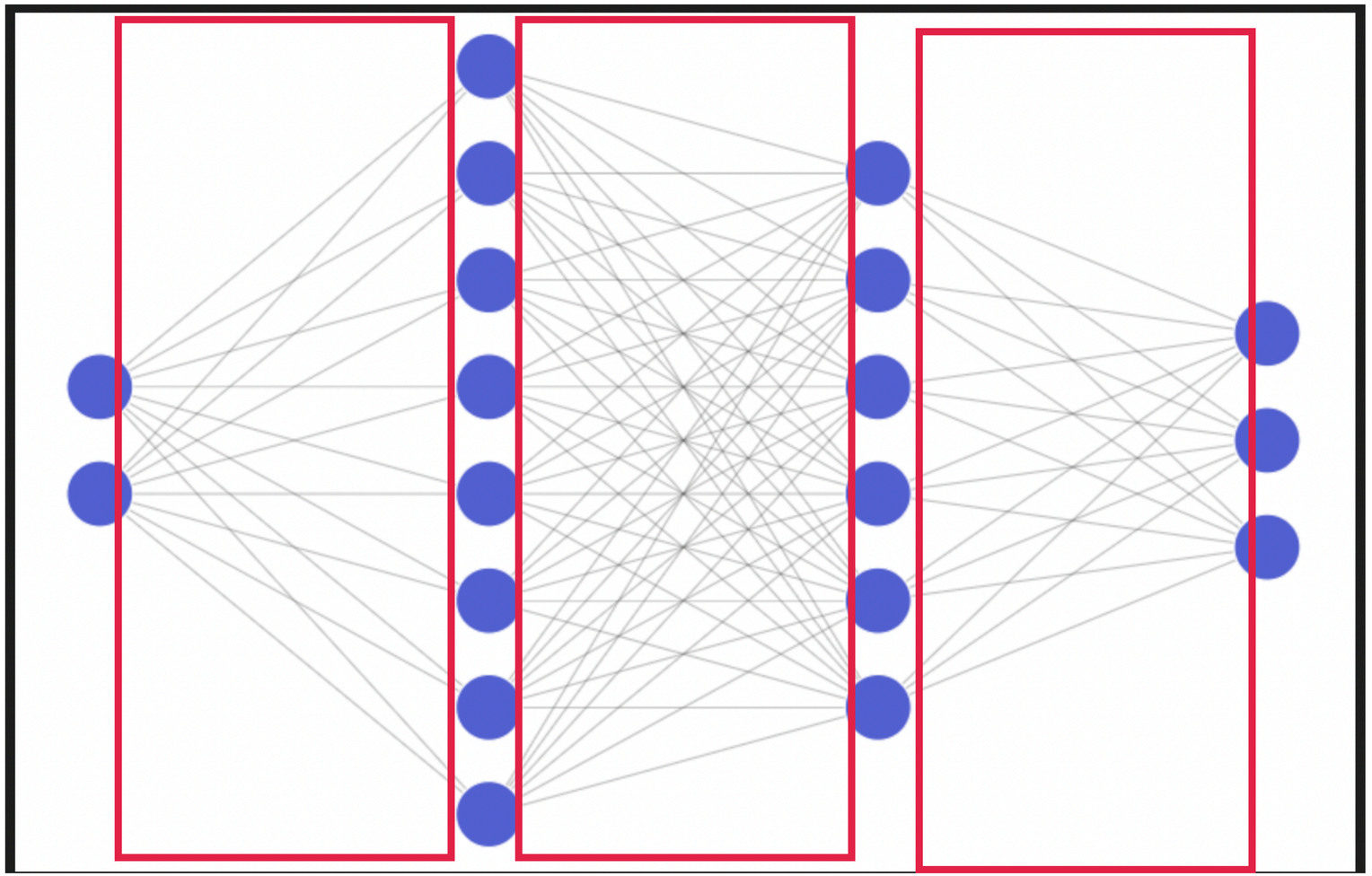
$\hat{q} = [\hat{q}(s, a_1), \hat{q}(s, a_2), \hat{q}(s, a_3)]$

Introdução a redes Neurais

Introdução a redes Neurais

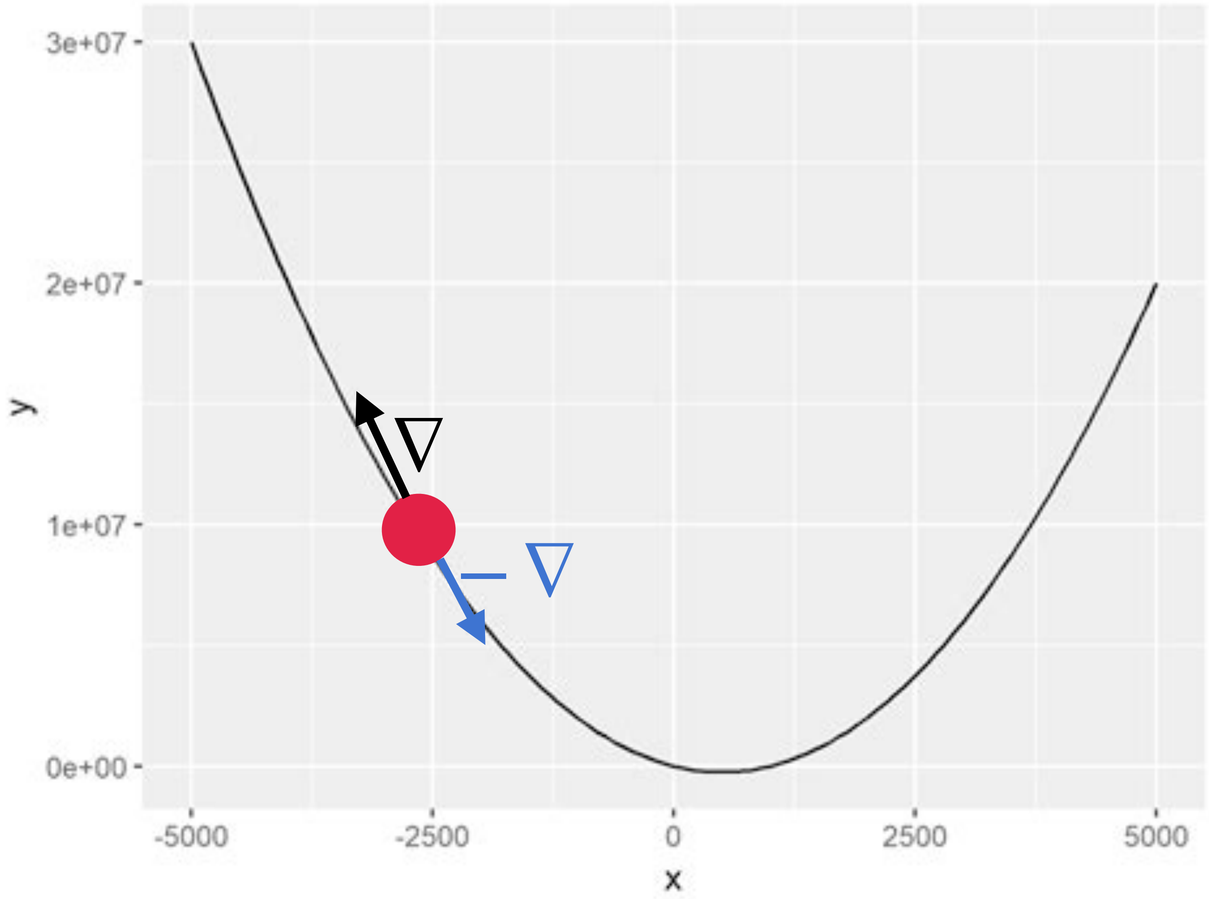


Introdução a redes Neurais



Pesos w

Cálculo do Gradiente



Deep SARSA

- SARSA + Redes neurais
- Como nos estados contínuos são infinito, só é necessário saber dos valores $q(s, a)$ deste momento.
- Aproxima os valores $\hat{q}(s, a)$ das ações disponíveis no estado atual s através da rede neural

Deep SARSA

Algorithm 1 Deep SARSA

Input: α learning rate, ϵ random action probability,
    γ discount factor,
Initialize q-value parameters θ and target parameters $\theta_{targ} \leftarrow \theta$
$\pi \leftarrow \epsilon$-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 $A_t \sim \pi(S_t)$
        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 SARSA

Algorithm 1 Deep SARSA

Input: α learning rate, ϵ random action probability,
    γ discount factor,
Initialize q-value parameters θ and target parameters $\theta_{targ} \leftarrow \theta$
$\pi \leftarrow \epsilon$-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 $A_t \sim \pi(S_t)$
        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 $\hat{q}(s, a|\theta)$

Agente interage com o ambiente

Deep SARSA

Algorithm 1 Deep SARSA

Input: α learning rate, ϵ random action probability,
    γ discount factor,
Initialize q-value parameters θ and target parameters $\theta_{targ} \leftarrow \theta$
$\pi \leftarrow \epsilon$-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 $A_t \sim \pi(S_t)$
        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 $\hat{q}(s, a|\theta)$

Atualização dos

Parâmetros da rede

Baseado nas experiencias

Processo de otimização

- Erro médio quadrado (MSE):

$$L(\theta) - \frac{1}{N}\sum_{i=1}^{N}[y_i - \hat{y}_i]^2$$

Minimização dos erros das estimativas

Processo de otimização

- No nosso caso:

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

Batch

Objetivo

Estimativa

$$y_i = R_i + \gamma \hat{q}(S'_i, A'_i | \theta_{targ})$$

Valor que queremos levar
As estimativas

$$y_i = \hat{q}(S_i, A_i | \theta)$$

Estimativa do valor $q(s, a)$

Processo de otimização

- Calculo do gradiente da função L em respeito aos pesos w:

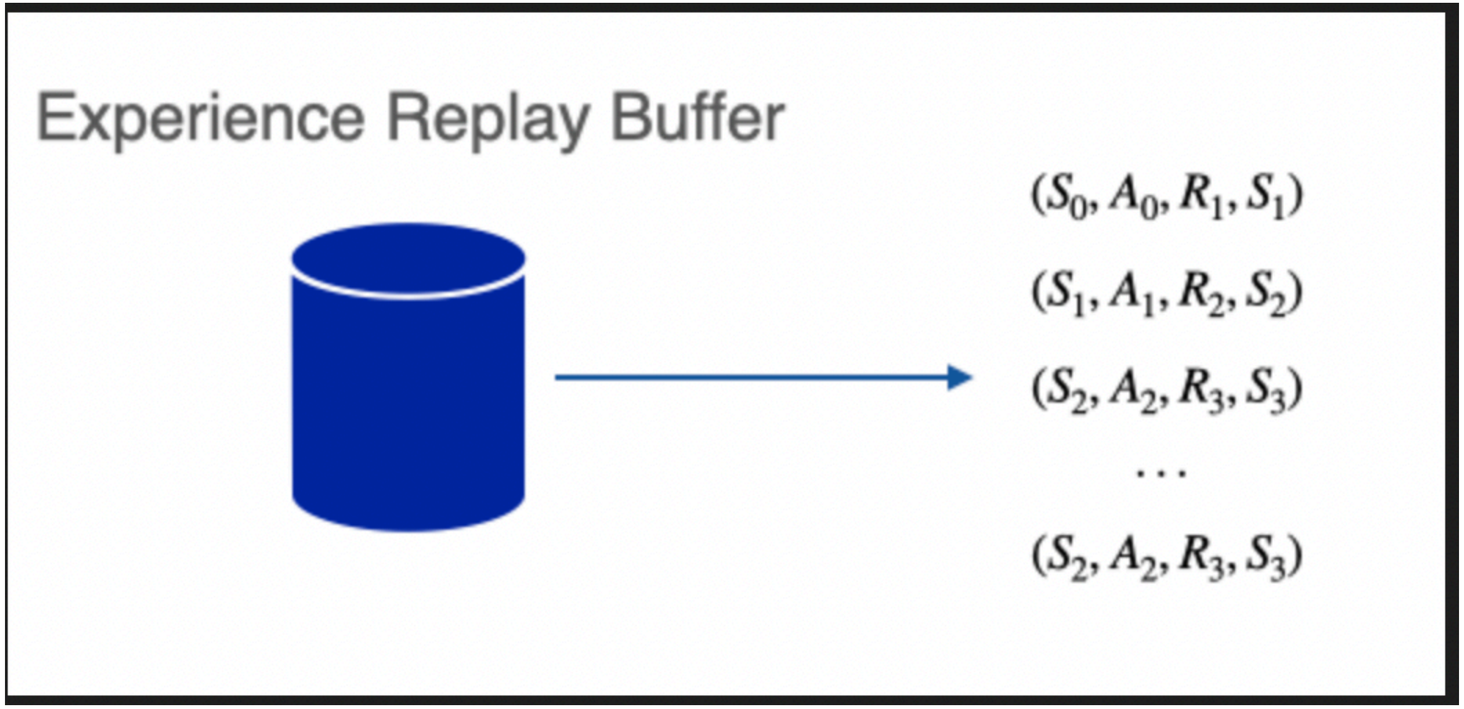
$$\nabla L(\theta) = [\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \cdots, \frac{\partial L}{\partial w_n}]$$

Com o SGD atualiza-se por um pequeno passo

$$\theta \leftarrow \theta - \alpha \nabla L(\theta)$$

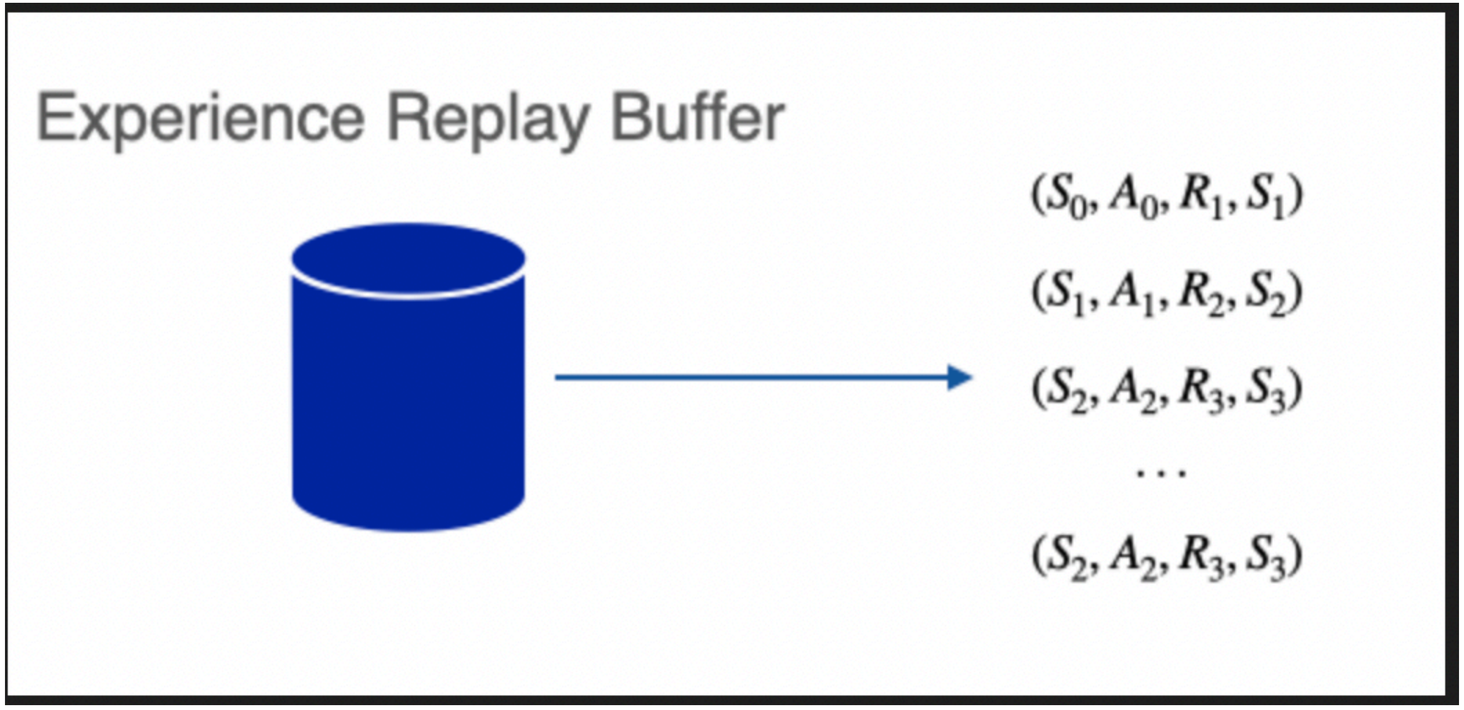
Memória de replay

- Memória que guarda as transições de estados que o agente observa ao tomar as ações



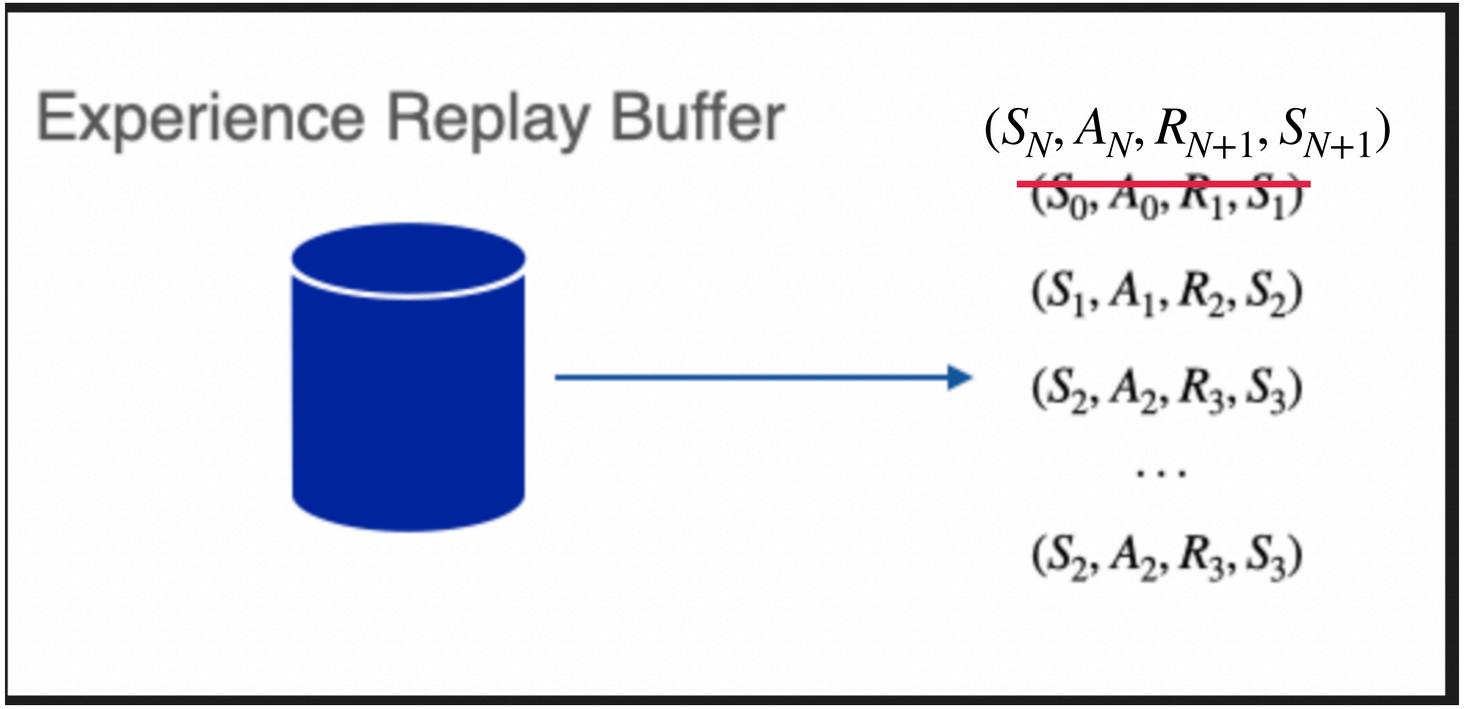
Memória de replay

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



Memória de replay

- A memória possui um espaço limitado e quando enche, os valores mais antigos são trocados pelos mais novos



Rede Neural Alvo

- Processo de aprendizado instável
- Quando um valor $q(s, a)$ muda, todos os valores serão afetados
- Quando se muda $\hat{q}(S_i, A_i \mid \theta)$, também modifica-se o valor alvo de $\hat{q}(S'_i, A'_i \mid \theta_{targ})$
- O alvo (Target) também precisa ser estável!

Rede Neural Alvo

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

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

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

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

Deep SARSA

Algorithm 1 Deep SARSA

Input: α learning rate, ϵ random action probability,
    γ discount factor,
Initialize q-value parameters θ and target parameters $\theta_{targ} \leftarrow \theta$
$\pi \leftarrow \epsilon$-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 $A_t \sim \pi(S_t)$
        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 $\hat{q}(s, a|\theta)$

Deep SARSA

Abra o Notebook
Cap_9_Deep_SARSA.ipynb