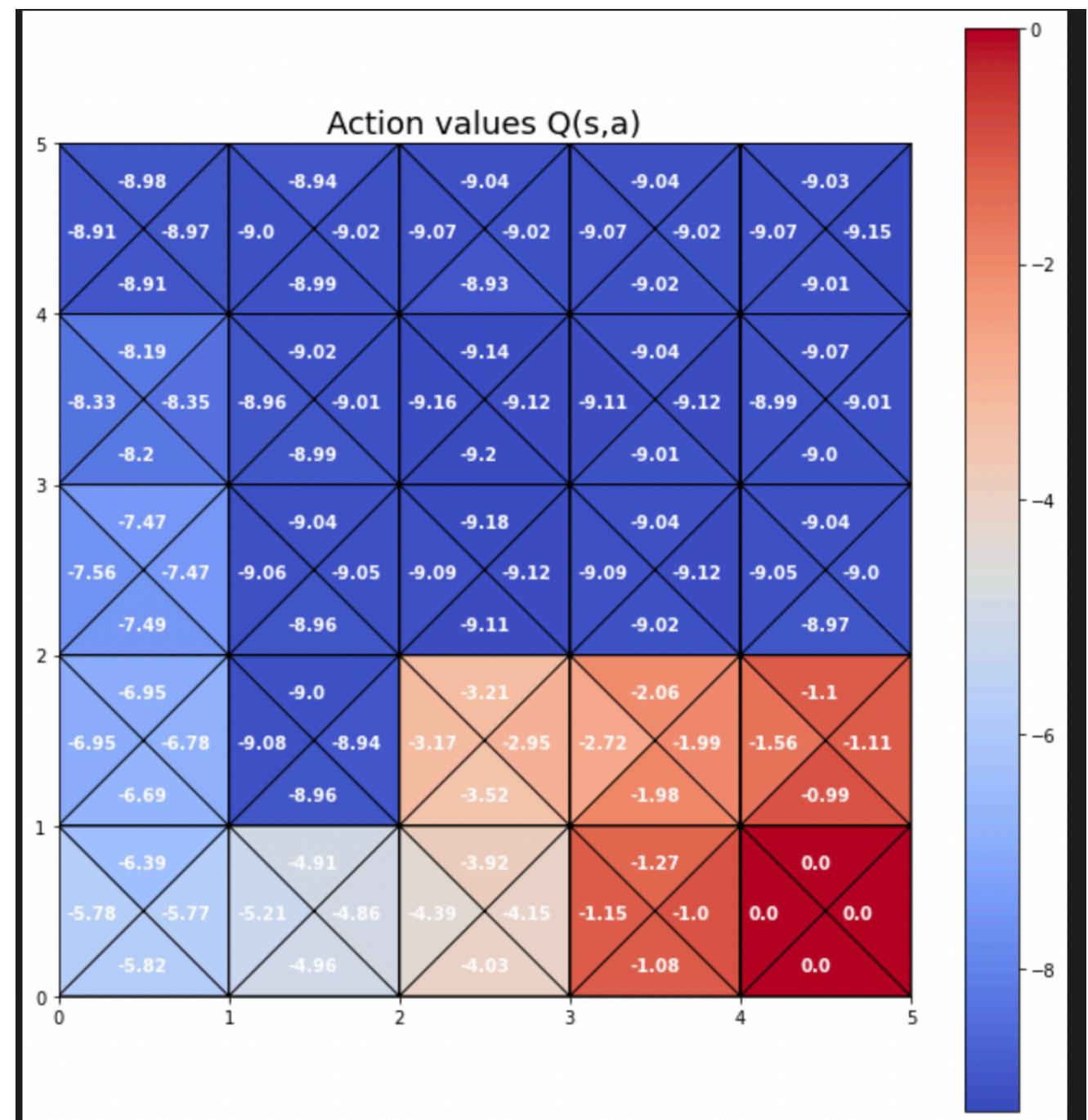


REINFORCE

Até agora

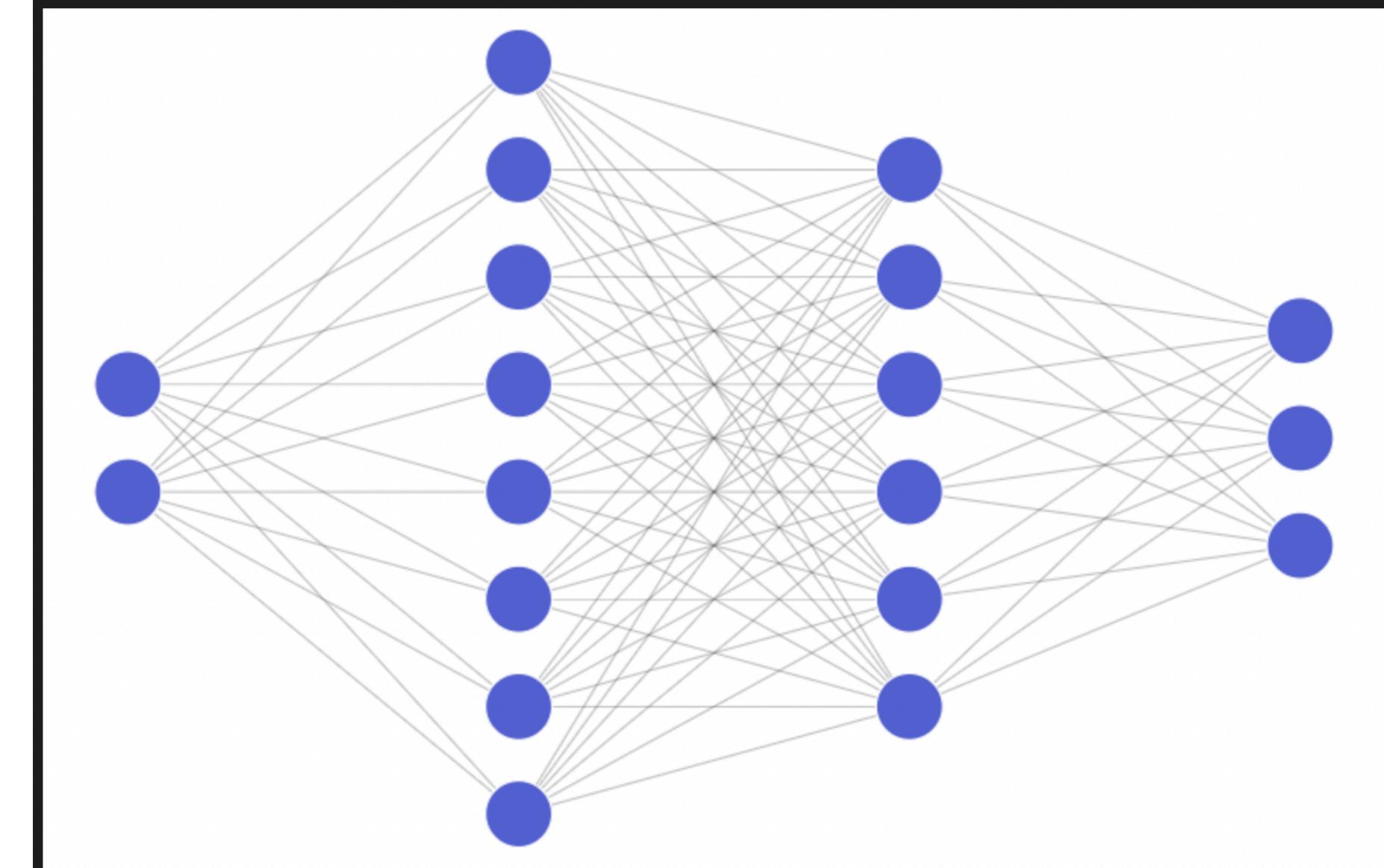
Tabela de Valores

$$a = \operatorname{argmax}_a Q(s, a)$$



Aproximadores de Função

$$a = \operatorname{argmax}_a \hat{q}(s, a | \theta)$$



Gradiente de Políticas

- Família de algoritmos que não se baseia nos valores $Q(s, a)$
- Aproximação de função para estimar a probabilidade de tomar cada ação

$$\pi(a | s, \theta) \in [0, 1]$$

Gradiente de Politicas

Plot twist: a rede neural é a política esse tempo todo



Vantagens

- Métodos baseados em valores $q(s, a)$ são ineficientes em representar políticas estocásticas de uma forma simples

Política gulosa

$$\pi(a'|s) = \begin{cases} 1 & \text{se } a' = \operatorname{argmax}_a \hat{q}(s, a | \theta) \\ 0 & \text{otherwise} \end{cases}$$

Política ϵ -gulosa

$$\pi(a'|s) = \begin{cases} 1 - \epsilon + \epsilon_r & \text{se } a' = \operatorname{argmax}_a \hat{q}(s, a | \theta) \\ \epsilon_r & \text{otherwise} \end{cases}$$

Vantagens

- A probabilidade de se escolher uma ação aumenta em pequenos incrementos, ao invés de pular de 0 para 1

$$a \sim \pi(s | \theta)$$

Performance de uma política

- Para melhorar a política é necessário pode compara-las
- Para isso, se define uma medida de performance $J(\theta)$
- Se $J_{\pi_1}(\theta) > J_{\pi_2}(\theta)$ então considera-se $J_{\pi_1}(\theta)$ melhor que $J_{\pi_2}(\theta)$

Performance de uma politica

- Objetivo: achar o conjunto de valores θ que maximizem a performance de politica J

$$\pi^\star(a | s, \theta) = \operatorname{argmax}_\theta J(\theta)$$

Performance de uma politica

- A partir de agora vamos utilizar a experiencia do agente com o ambiente para estimar a performance da politica como $\hat{J}(\theta)$

$$\hat{J}(\theta) \sim J(\theta)$$

Performance de uma política

- Stochastic Gradient Ascent (SGA) é utilizado para aproximar os valores ótimos de θ

$$\theta_{t+1} = \theta_t + \alpha \nabla \hat{J}(\theta)$$

Onde:

$$\nabla \hat{J}(\theta) = \left[\frac{\partial \hat{J}(\theta)}{\partial \theta_1}, \frac{\partial \hat{J}(\theta)}{\partial \theta_2}, \dots, \frac{\partial \hat{J}(\theta)}{\partial \theta_n} \right]$$

O teorema do gradiente da politica

- Definimos performance como e deriva-se o valor $v_\pi(S)$:

$$J(\theta) = v_\pi(S_0)$$

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s, a) \nabla \pi(a | s, \theta)$$

O teorema do gradiente da política

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_{\pi}(s, a) \nabla \pi(a | s, \theta)$$

Gradiente da performance

Distribuição de estados
Seguindo π

Par de valores $q(s, a)$
Seguindo π

Gradiente da probabilidade
De escolher ação a

```
graph TD; A["\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_{\pi}(s, a) \nabla \pi(a | s, \theta)"] --> B["Gradiente da performance"]; A --> C["Distribuição de estados  
Seguindo \pi"]; A --> D["Par de valores  $q(s, a)$   
Seguindo  $\pi$ "]; A --> E["Gradiente da probabilidade  
De escolher ação  $a$ "]
```

REINFORCE

Gradiente
De
Politica

+

Monte
Carlo

REINFORCE

- O algoritmo vai utilizar o SGA aproximando o gradiente de performance $\nabla \hat{J}(\theta)$ utilizando amostras coletadas do ambiente através de tentativa e erro

$$\theta_{t+1} = \theta_t + \alpha \nabla \hat{J}(\theta)$$

REINFORCE

- É necessário obter amostrar do ambiente na qual a esperança matemática vai ser aproximadamente a seguinte expressão:

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s, a) \nabla \pi(a | s, \theta)$$

REINFORCE

- Gradiente e regra:

$$\nabla \hat{J}(\theta) = \gamma^t G_t \frac{\nabla_{\pi}(A_t | S_t, \theta_t)}{\pi(A_t | S_t, \theta_t)}$$

$$\theta_{t+1} = \theta_t + \alpha \gamma^t G_t \frac{\nabla_{\pi}(A_t | S_t, \theta_t)}{\pi(A_t | S_t, \theta_t)}$$

REINFORCE

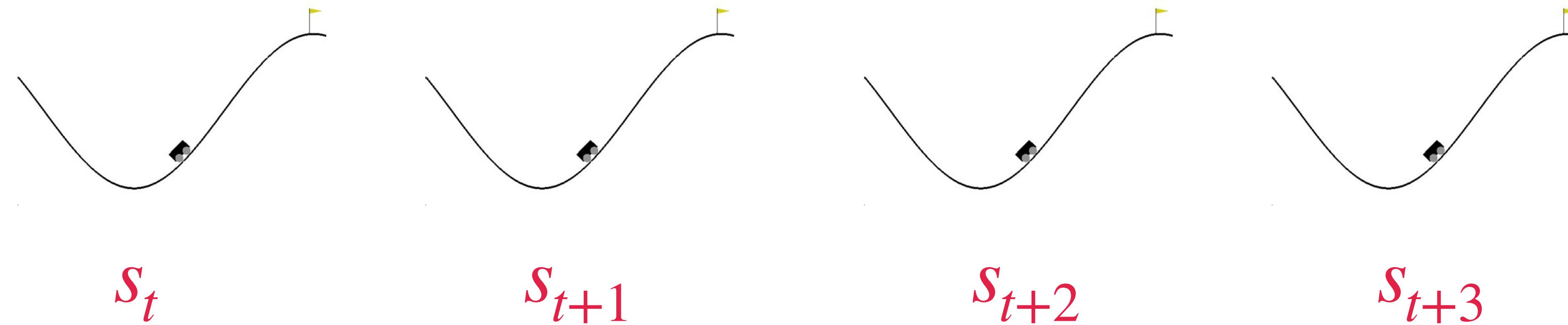
- Simplifica-se a regra mais ainda:

$$\nabla \ln \pi(A_t | S_t, \theta) = \frac{\nabla_{\pi}(A_t | S_t, \theta_t)}{\pi(A_t | S_t, \theta_t)}$$

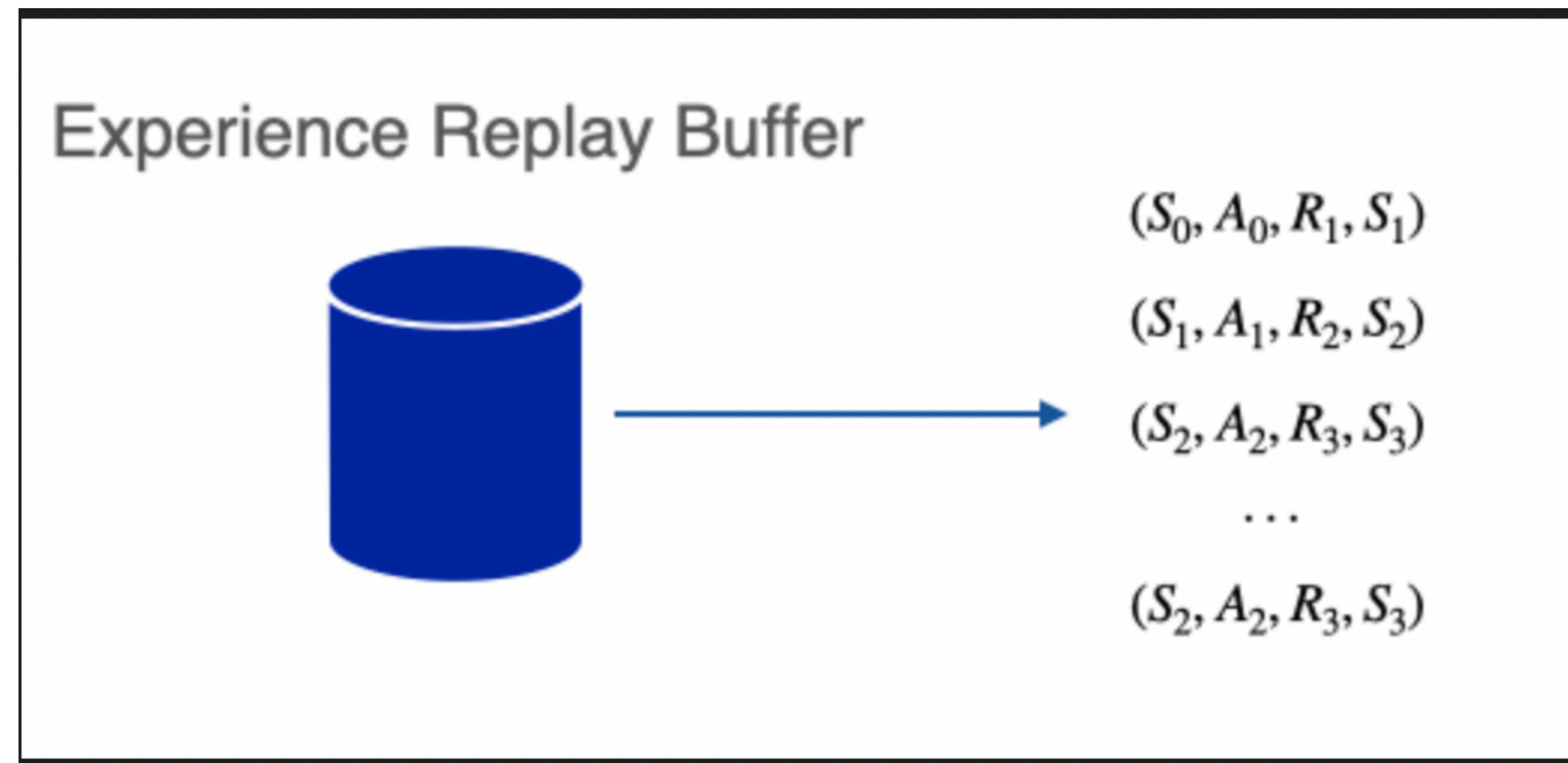
$$\theta_{t+1} = \theta_t + \alpha \gamma^t G_t \nabla \ln \pi(A_t | S_t, \theta_t)$$

Aprendizado Paralelo

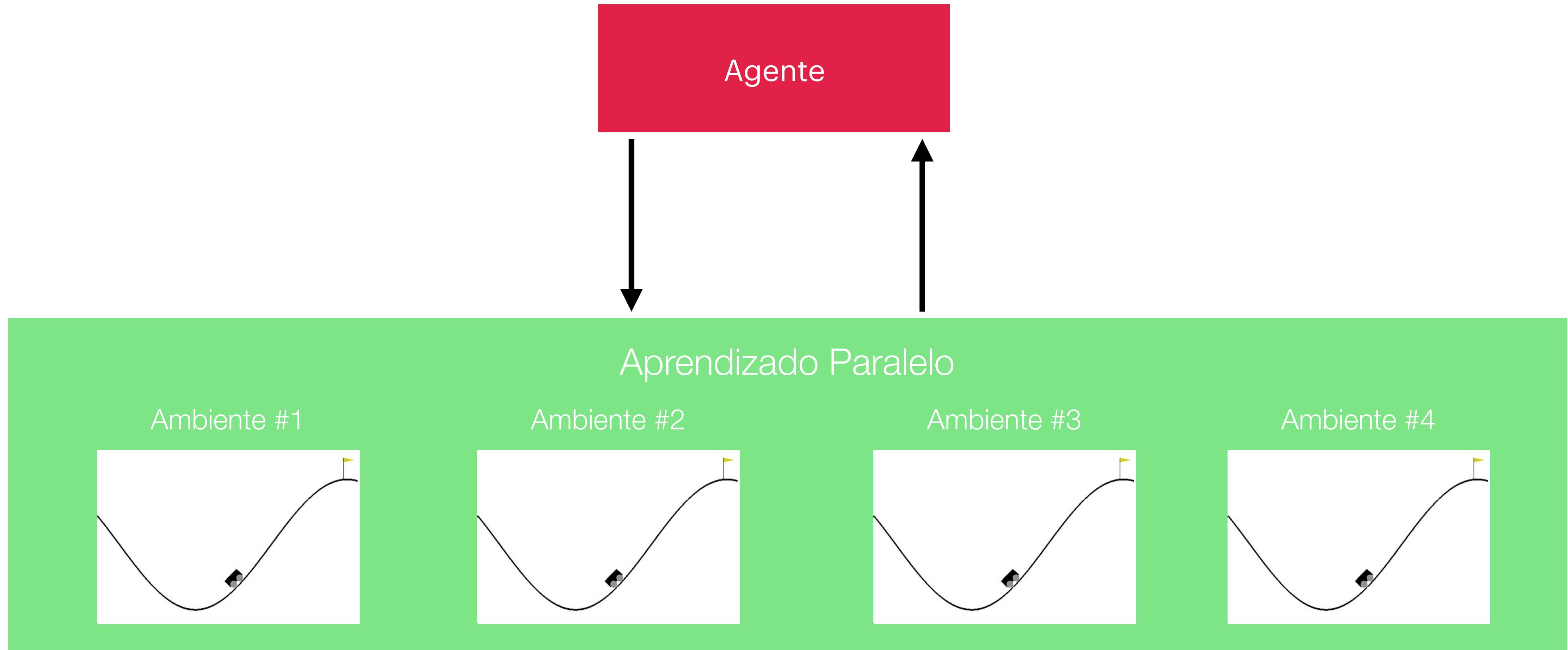
- Problema: estados adjacentes são muito parecidos



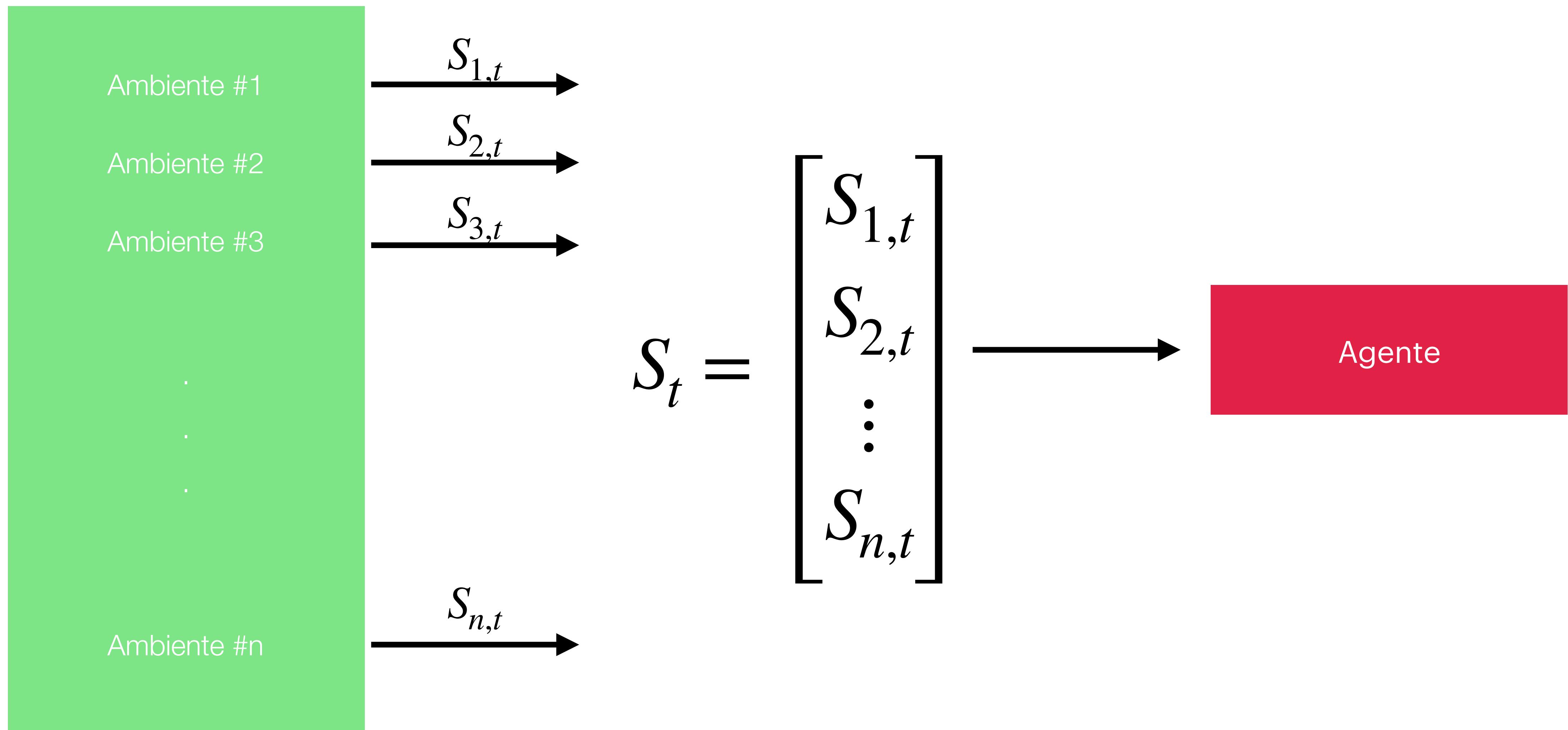
Aprendizado Paralelo



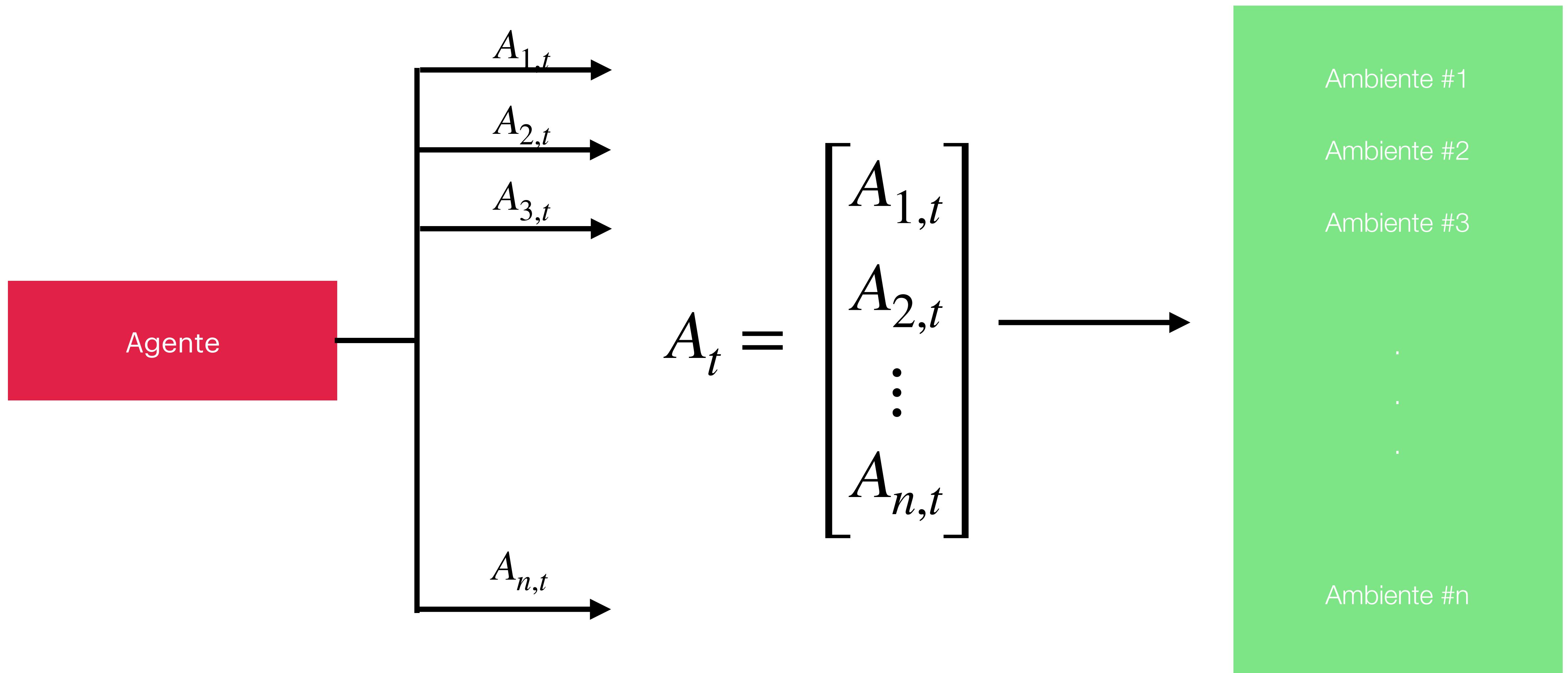
Aprendizado Paralelo



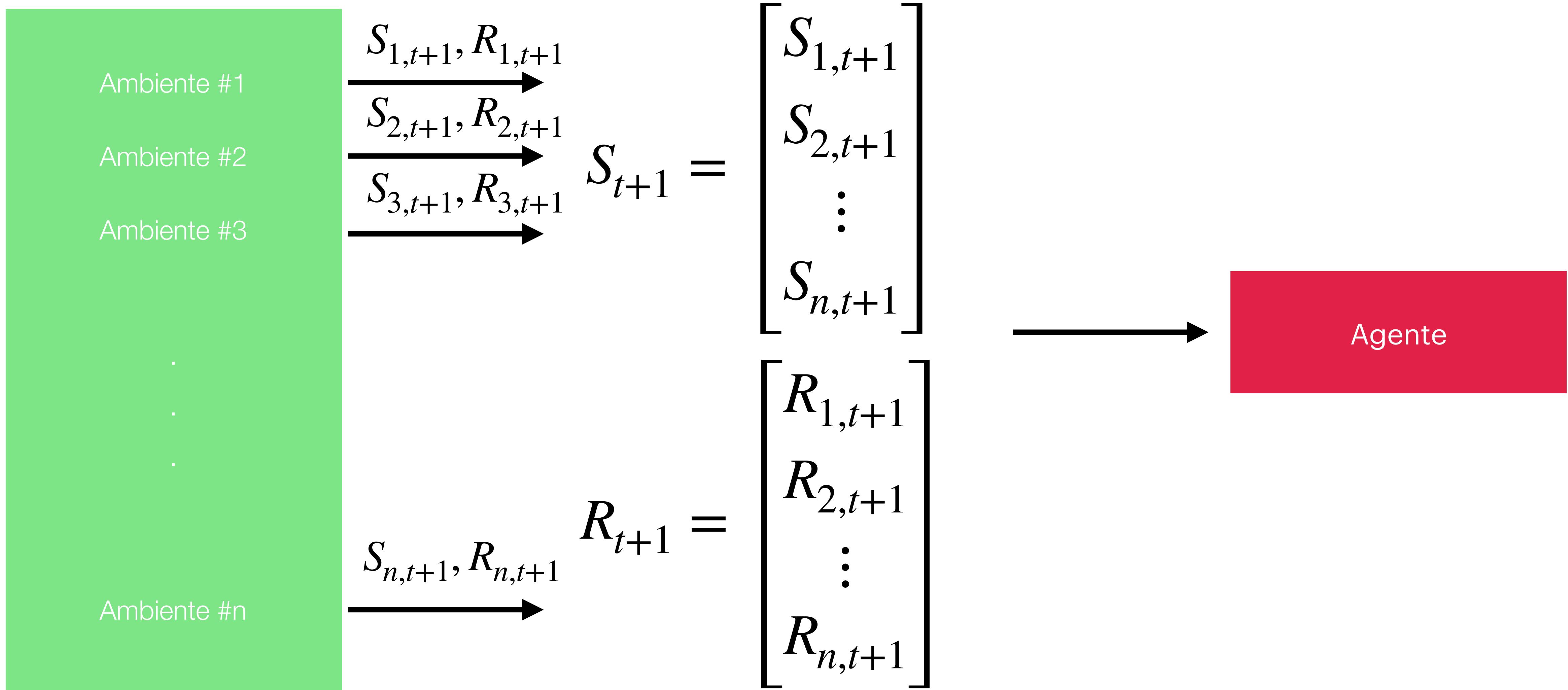
Aprendizado Paralelo



Aprendizado Paralelo



Aprendizado Paralelo



Estratégia de Exploração

- É necessário manter o fator de exploração do agente
- Não se tem mecanismos como a política ϵ -gulosa
- Como vamos incorporar mecanismos de exploração na rede neural?

Estratégia de Exploração

- Vamos incentivar o agente a manter a entropia da política o mais alto possível

$$H(X) = - \sum_{x \in X} p(x) \cdot \ln p(x)$$

Estratégia de Exploração

- Entropia: o nível de incerteza de uma variável aleatória

$$H(X) = - \sum_{x \in X} p(x) \cdot \ln p(x)$$

Estratégia de Exploração

- Exemplo: Moeda viciada

$$p(X = x_1) = 1$$

$$p(X = x_2) = 0$$

$$H(X) = - [1 \cdot \ln(1) + 0 \cdot \ln(0)] = 0$$



Estratégia de Exploração

- Exemplo: Moeda não viciada

$$p(X = x_1) = 0.5 \quad p(X = x_2) = 0.5$$

$$H(X) = -[0.5 \cdot \ln(0.5) + 0.5 \cdot \ln(0.5)] \approx 0.6931$$



Estratégia de Exploração

- Entropia de uma política: incerteza de uma ação ser escolhida em um estado
- Manter a entropia alta incentiva a exploração

$$H(X) = - \sum_{a \in A_t} \pi(a | S_t) \cdot \ln \pi(a | S_t)$$

Estratégia de Exploração

- Adicionamos o fator de entropia na função a ser maximizada

$$\theta_{t+1} = \theta_t + [\alpha \gamma^t G_t \nabla \ln \pi(A_t | S_t, \theta_t) + \beta \nabla H(\pi)]$$

Estratégia de Exploração

- Vantagens:
 - Exploração
 - Robustez
 - Refinamento da Política

Algoritmo

Algorithm 1 REINFORCE

- 1: **Input:** α learning rate, γ discount factor.
- 2: Initialize parallel environments E
- 3: Initialize policy parameters θ
- 4: **for** episode in 1..N **do**
- 5: Use $\pi(s|\theta)$ to collect $|E|$ trajectories: $S_0, A_0, R_1, \dots, R_T$
- 6: $G = \vec{0}$
- 7: **for** t = T-1..0 **do**
- 8: $G = R_t + \gamma G$
- 9: Compute entropy regularization: $H_t = -\sum_a \pi(a|S_t) \ln \pi(a|S_t)$
- 10: $\hat{J}(\theta) = \gamma^t G \ln \pi(A_t|S_t, \theta) - H_t$
- 11: $\theta = \theta + \alpha \nabla \hat{J}(\theta)$
- 12: **end for**
- 13: **end for**
