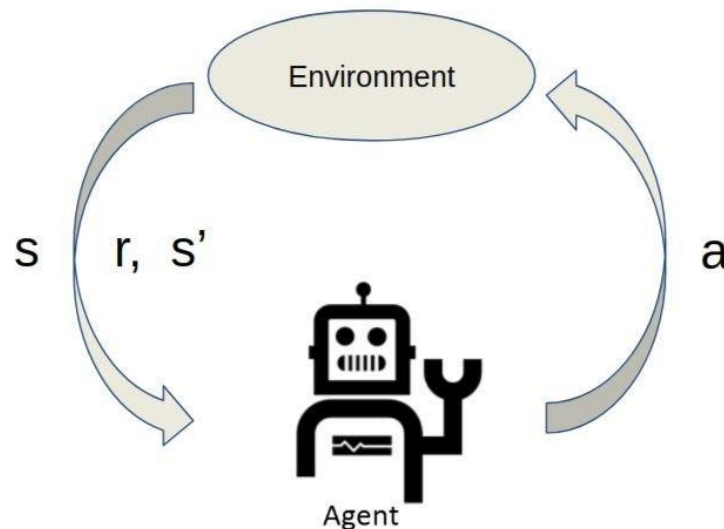


*Fundamentos de Linguagem Python para Análise de Dados e Data Science*

# Fundamentos de Linguagem Python Para Análise de Dados e Data Science

## Algoritmo Q-Learning

O Q-Learning é um algoritmo de aprendizado por reforço baseado em valor, que visa aprender uma função de valor-ação, chamada de função Q, para estimar o valor esperado de executar uma ação em um estado específico e seguir uma política ótima a partir daí. A função Q é representada como  $Q(s, a)$ , onde  $s$  é o estado e  $a$  é a ação. O objetivo do Q-Learning é aprender a política ótima  $r(s)$  que maximiza a recompensa acumulada ao longo do tempo.



O algoritmo Q-Learning segue um processo iterativo e pode ser descrito nos seguintes passos:

**Inicialização:** Inicialize a tabela Q com zeros (ou valores pequenos) para todos os pares estado-ação ( $s, a$ ).

**Interação com o ambiente:** O agente interage com o ambiente repetidamente, executando episódios até que o algoritmo convirja ou um limite de episódios seja atingido. Em cada episódio:

- Observe o estado atual ( $s$ ) do ambiente.
- Selecione uma ação ( $a$ ) com base na política atual, como a política  $\epsilon$ -greedy, que explora ações aleatórias com probabilidade  $\epsilon$  e explora a ação com o maior valor  $Q(s, a)$  com probabilidade  $1 - \epsilon$ .
- Execute a ação ( $a$ ) selecionada e observe a recompensa ( $r$ ) e o próximo estado ( $s'$ ) resultante.
- Atualize a tabela Q usando a regra de atualização do Q-Learning:

$$Q(s, a) = Q(s, a) + \alpha * (r + \gamma * \max_{a'} Q(s', a') - Q(s, a))$$



## *Fundamentos de Linguagem Python para Análise de Dados e Data Science*

---

onde  $\alpha$  é a taxa de aprendizado,  $\gamma$  é o fator de desconto e  $\max_a Q(s', a)$  é o valor máximo de  $Q$  para o próximo estado ( $s'$ ) e todas as ações possíveis ( $a$ ).

e. Atualize o estado atual:  $s = s'$ .

f. Repita os passos b-e até que o episódio termine (por exemplo, o agente atinge um estado terminal ou um limite de passos é atingido).

Ao final do processo de aprendizado, o agente terá aprendido uma aproximação da função  $Q$  ótima, que pode ser usada para tomar decisões que maximizem a recompensa acumulada. Para escolher a ação ótima em um estado  $s$ , o agente simplesmente seleciona a ação  $a$  que possui o maior valor  $Q(s, a)$ .

Confira o passo a passo do algoritmo no Deep Learning Book:

<https://www.deeplearningbook.com.br/algoritmo-de-agente-baseado-em-ia-com-reinforcement-learning-q-learning/>