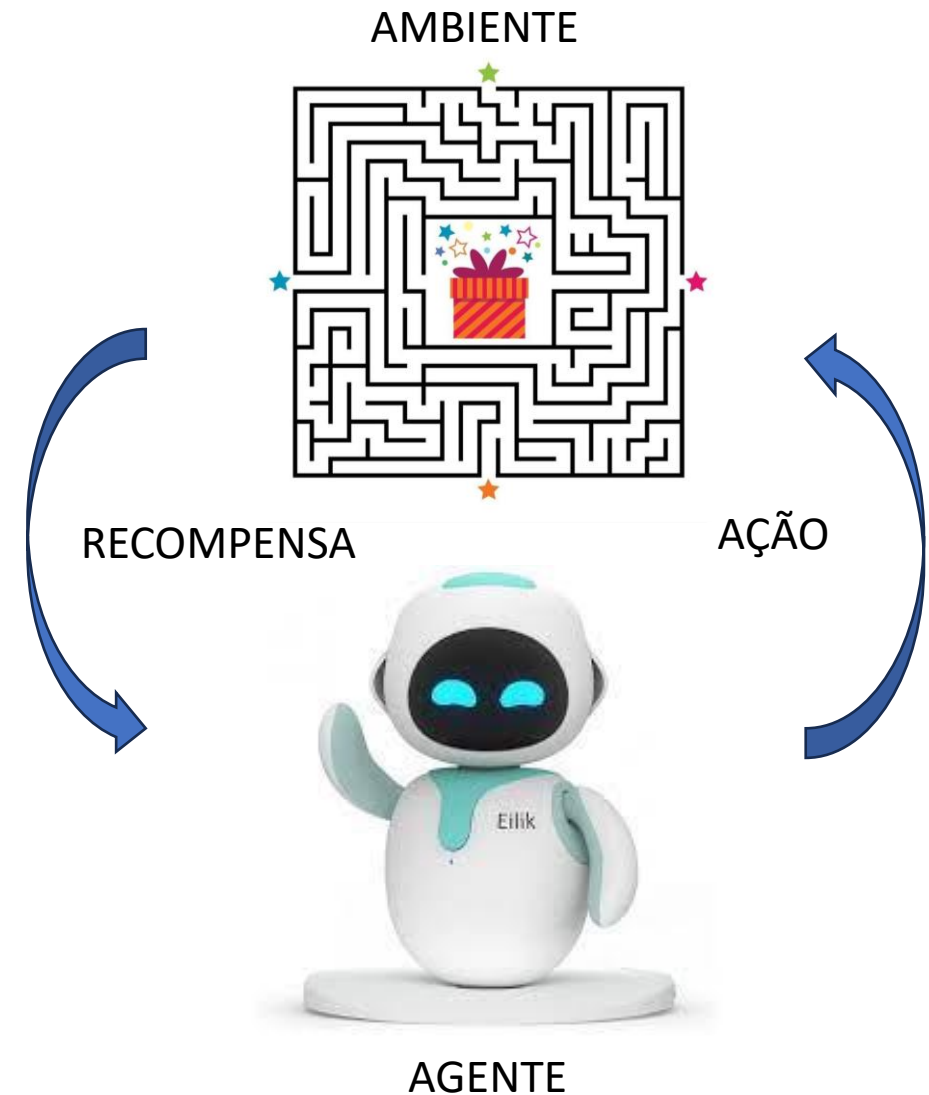


TP558 - Tópicos avançados em Machine Learning: Deep Q-Learning





Introdução

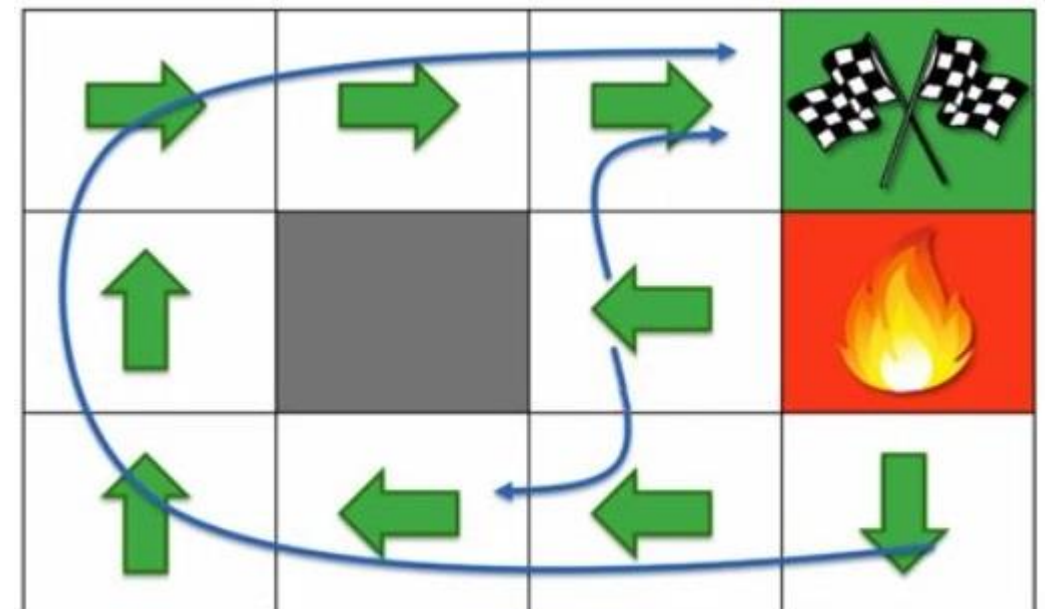
- Deep Q-Learning é uma técnica de aprendizado de máquina que combina o algoritmo Q-Learning com redes neurais profundas.
- É utilizada em problemas de aprendizado por reforço, nos quais um agente aprende a executar ações em um ambiente para maximizar uma recompensa cumulativa ao longo do tempo.



Introdução

- O algoritmo Q-Learning é uma forma de aprendizado por reforço que envolve aprender uma função de valor de ação chamada de função Q, que associa pares de estado-ação a valores representando a utilidade esperada dessas ações.
- Tradicionalmente, o Q-Learning é implementado com tabelas de valores Q, onde cada estado-ação tem uma entrada na tabela.

V=0.71	V=0.74	V=0.86	
V=0.63		V=0.39	
V=0.55	V=0.46	V=0.36	V=0.22



Introdução

- No entanto, o Deep Q-Learning é uma técnica específica dentro do campo da inteligência artificial, e sua importância é destacada em várias áreas:
 - **Aplicações em Jogos:** Jogos de Atari e jogos de tabuleiro;
 - **Saúde e Medicina:** Otimizar tratamentos médicos;
 - **Sistemas de Navegação Autônoma:** Tomada de decisões de navegação, ajudando a evitar obstáculos;
 - **Controle de Sistemas Complexos:** Ações sequenciais em ambientes dinâmicos e complexos;

Fundamentação teórica

APRENDIZADO POR REFORÇO

O Aprendizado por Reforço é um paradigma de aprendizado de máquina inspirado na psicologia comportamental, no qual um agente aprende a executar ações em um ambiente para maximizar uma recompensa cumulativa.

Fundamentação teórica

Processo

O **agente** é o sistema de inteligência artificial que está aprendendo a interagir com o ambiente. Ele observa o estado atual do ambiente e toma decisões sobre quais ações tomar.

O **ambiente** é tudo com o qual o agente interage. Pode ser físico (como um robô navegando em um ambiente real) ou virtual (como um programa de computador jogando um jogo).

Fundamentação teórica

O **estado** representa a configuração atual do ambiente em um determinado momento. É a informação relevante para a tomada de decisão do agente.

Uma **ação** é uma escolha feita pelo agente em resposta ao estado atual do ambiente. O agente seleciona a ação que ele acredita ser mais vantajosa com base em sua política de decisão.

Fundamentação teórica

A **recompensa** é um sinal de feedback que o agente recebe do ambiente após realizar uma ação. Ela indica o quão boa ou ruim foi a ação tomada em relação ao objetivo do agente. O objetivo do agente é maximizar a recompensa cumulativa ao longo do tempo.

A **política** é a estratégia que o agente utiliza para escolher ações com base nos estados do ambiente. Ela mapeia estados para ações e é ajustada ao longo do tempo à medida que o agente aprende.

Fundamentação teórica

O **agente aprende** a melhor política através de tentativa e erro. Ele explora diferentes ações e observa as recompensas resultantes, ajustando sua política com o objetivo de maximizar as recompensas futuras.

O agente enfrenta um dilema entre **explorar** novas ações para descobrir novas informações e explorar ações conhecidas para maximizar recompensas imediatas. Encontrar um equilíbrio entre exploração e exploração é fundamental para o sucesso do aprendizado por reforço.

Fundamentação teórica

Q-LEARNING

Q-Learning é um algoritmo de aprendizado de reforço que visa aprender uma política ótima para controlar um agente em um ambiente desconhecido e estocástico.

Ele é frequentemente utilizado em problemas de tomada de decisão sequencial, nos quais o agente interage com o ambiente de maneira iterativa, recebendo feedbacks em forma de recompensa.

Fundamentação teórica

CONCEITOS DE FUNCIONAMENTOS

Inicialização: Inicialize a tabela Q com valores arbitrários ou zeros para todos os pares estado-ação possíveis.

Escolha de ação: Selecionar uma ação para ser executada no estado atual. Isso pode ser feito usando uma política de exploração que equilibra a exploração de novas ações com a exploração das ações já conhecidas.

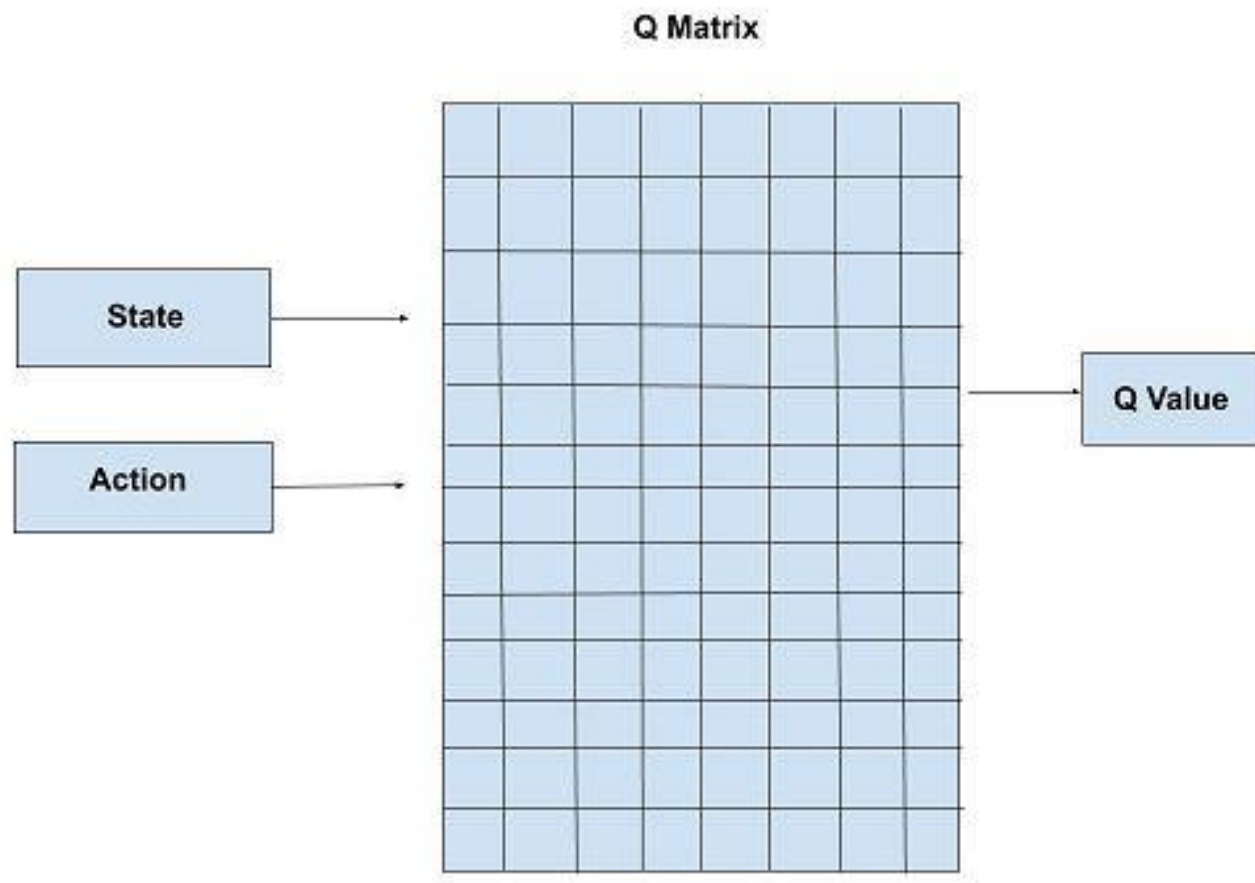
Fundamentação teórica

Execução da ação: Executar a ação escolhida no ambiente e observar a recompensa e o próximo estado.

Atualização do Q-Value: Usar a equação de atualização do Q-Value para atualizar o valor Q do par estado-ação, levando em consideração a recompensa recebida, o valor Q do próximo estado e a taxa de aprendizado.

Iteração: Repetir os passos 2 a 4 até que um critério de parada seja alcançado, como um número máximo de iterações ou até que a convergência seja alcançada.

Fundamentação teórica



Fundamentação teórica

CONCEITO DE REDES NEURAIIS

As redes neurais são uma classe de modelos computacionais inspirados no funcionamento do cérebro humano. Essa estrutura é capaz de aprender a partir de dados e realizar tarefas complexas, como reconhecimento de padrões, classificação, previsão e tomada de decisão.

Arquitetura e funcionamento

COMBINAÇÃO DE REDES NEURAIIS COM Q-LEARNING

A combinação de redes neurais com o algoritmo Q-learning é uma abordagem interessante para resolver problemas de aprendizado por reforço em ambientes complexos e de alta dimensionalidade.

Arquitetura e funcionamento

- Em vez de manter uma tabela Q explícita (que pode ser inviável em ambientes com um grande número de estados), podemos usar uma rede neural para representar a função Q. Isso é conhecido como "Deep Q-Network" (DQN).
- A entrada da rede neural é uma representação do estado do ambiente, e a saída é um vetor de valores Q para cada ação possível no estado atual.
- Durante o treinamento, a rede neural é ajustada para minimizar a diferença entre os valores Q preditos e os valores Q reais, calculados usando a equação de Bellman.

Arquitetura e funcionamento

EQUAÇÃO DE BELLMAN

$$Q(s, a) = r(s, a) + \gamma \max_a Q(s', a)$$

Q = Quantidade

s = Estado

a = ação

r = Recompensa

γ = Fator de desconto

Arquitetura e funcionamento

- A exploração do espaço de ações pode ser feita usando uma política ϵ -greedy, onde uma pequena fração ϵ das vezes uma ação aleatória é escolhida, enquanto o restante do tempo a ação com o maior valor Q é escolhida.
- O processo de treinamento envolve a coleta de experiências (estado, ação, recompensa, próximo estado) e a utilização dessas experiências para atualizar os pesos da rede neural, usando um algoritmo de otimização como o gradiente descendente.

Arquitetura e funcionamento

REDE NEURAL PROFUNDA (DNN)

Uma Rede Neural Profunda (DNN) é uma forma de rede neural artificial composta por várias camadas de neurônios, com cada camada se comunicando com a próxima.

Arquitetura e funcionamento

Camada de Entrada

- É a camada inicial da rede onde os dados de entrada são alimentados.

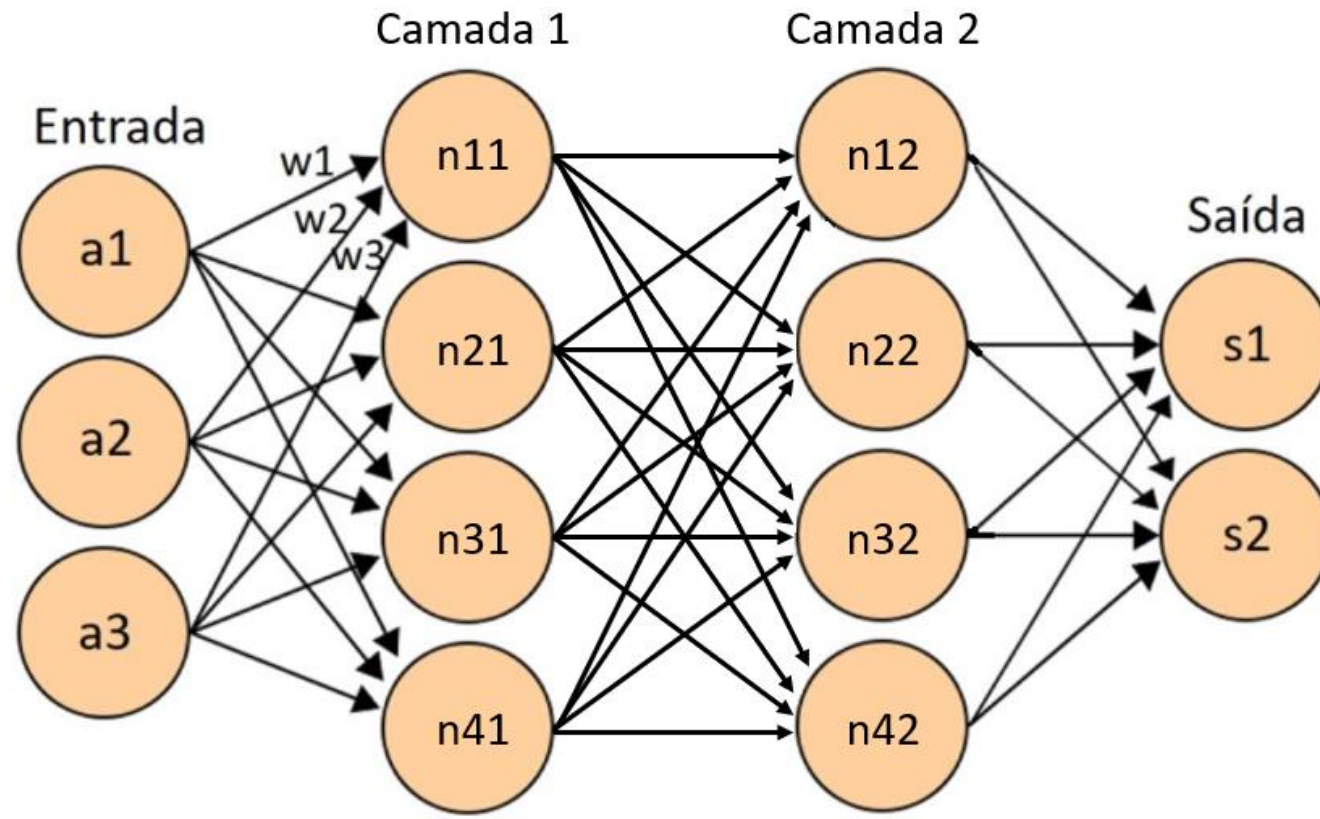
Camadas Ocultas

- São as camadas intermediárias ou ocultas, onde realizam cálculos sobre os dados de entrada.

Camada de Saída

- É a camada final da rede onde os resultados são gerados.

Arquitetura e funcionamento



Arquitetura e funcionamento

Funções de Ativação

- Cada neurônio em uma camada oculta aplica uma função de ativação aos resultados da soma ponderada das entradas

Pesos e Viés

- Cada conexão entre neurônios é associada a um peso que controla a força da conexão

Função de Perda

- Durante o treinamento, os pesos da rede são ajustados para minimizar a função de perda

Algoritmo de Otimização

- O algoritmo usado para ajustar os pesos da rede durante o treinamento, com o objetivo de minimizar a função de perda.

Arquitetura e funcionamento

FUNÇÃO “Q” EM REDE NEURAL

A função Q (também chamada de função de valor de ação) é usada para avaliar a qualidade de uma ação em um determinado estado.

Ela atribui um valor numérico a cada par (estado, ação), representando a "utilidade" esperada de escolher essa ação enquanto estiver no estado correspondente.

A função Q pode ser representada como uma tabela (para problemas com um número finito de estados e ações) ou, mais comumente, por uma função aproximadora, como uma Rede Neural .

Arquitetura e funcionamento

TREINAMENTO DA DNN

No treinamento, a rede neural é atualizada para minimizar uma função de perda, que mede a diferença entre os valores Q previstos pela rede e os valores Q reais observados. A função de perda comum é o erro quadrático médio (MSE - Mean Squared Error) entre os valores Q previstos e os valores alvo.

Os valores alvo são calculados usando a equação de Bellman, que é uma equação de recursão que relaciona o valor Q de um estado e uma ação ao valor Q do próximo estado e à melhor ação subsequente.

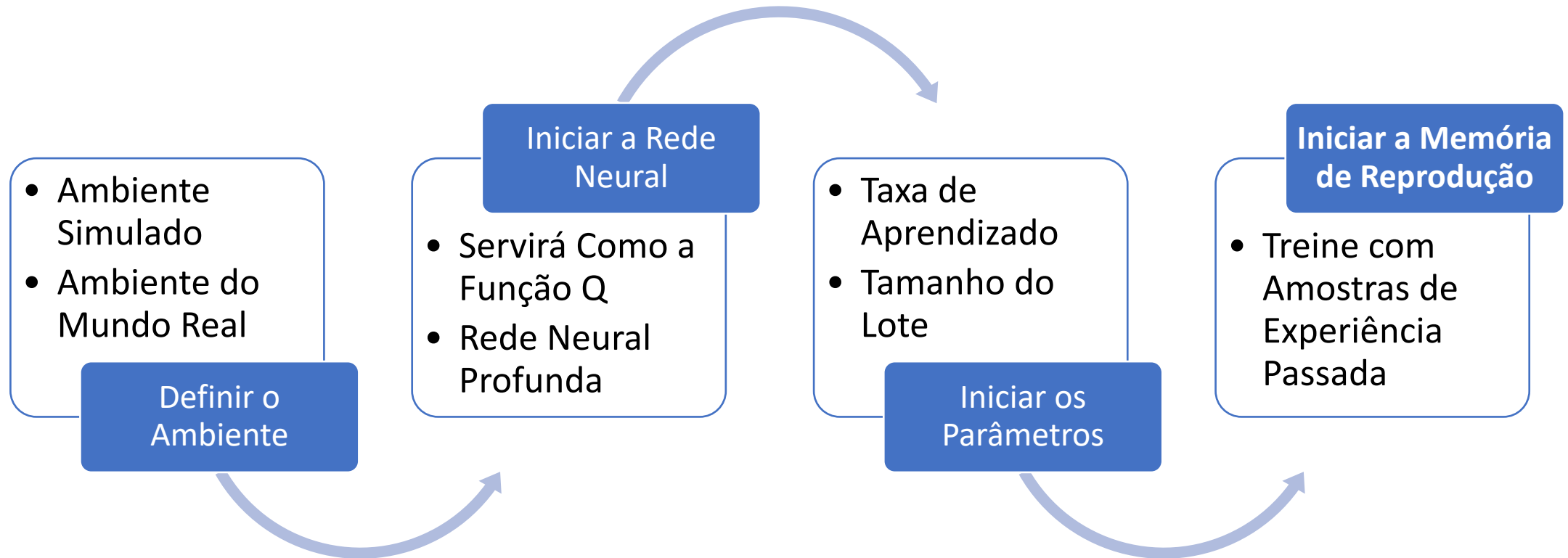
Arquitetura e funcionamento

TREINAMENTO DA DNN

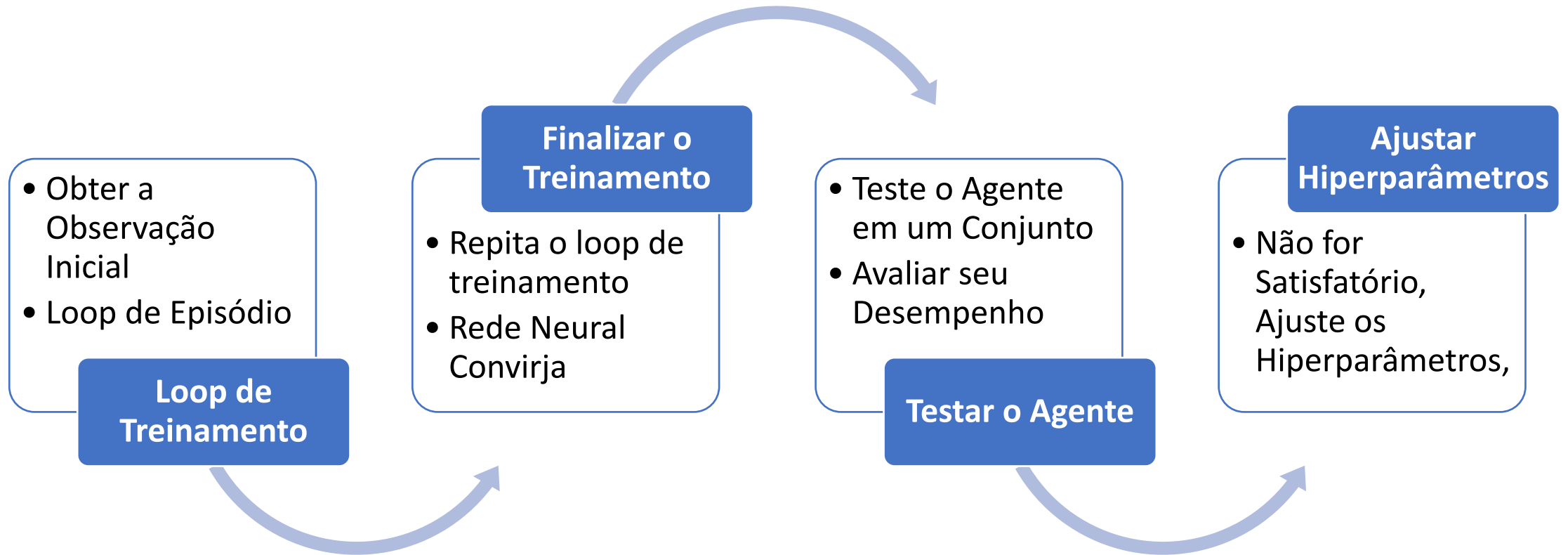
Durante o treinamento, a rede neural é ajustada iterativamente usando gradient descent para minimizar a diferença entre os valores Q previstos e os valores alvo.

Treinamento e otimização

PROCESSO BÁSICO DE TREINAMENTO EM DEEP Q-LEARNING



Treinamento e otimização



Treinamento e otimização

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

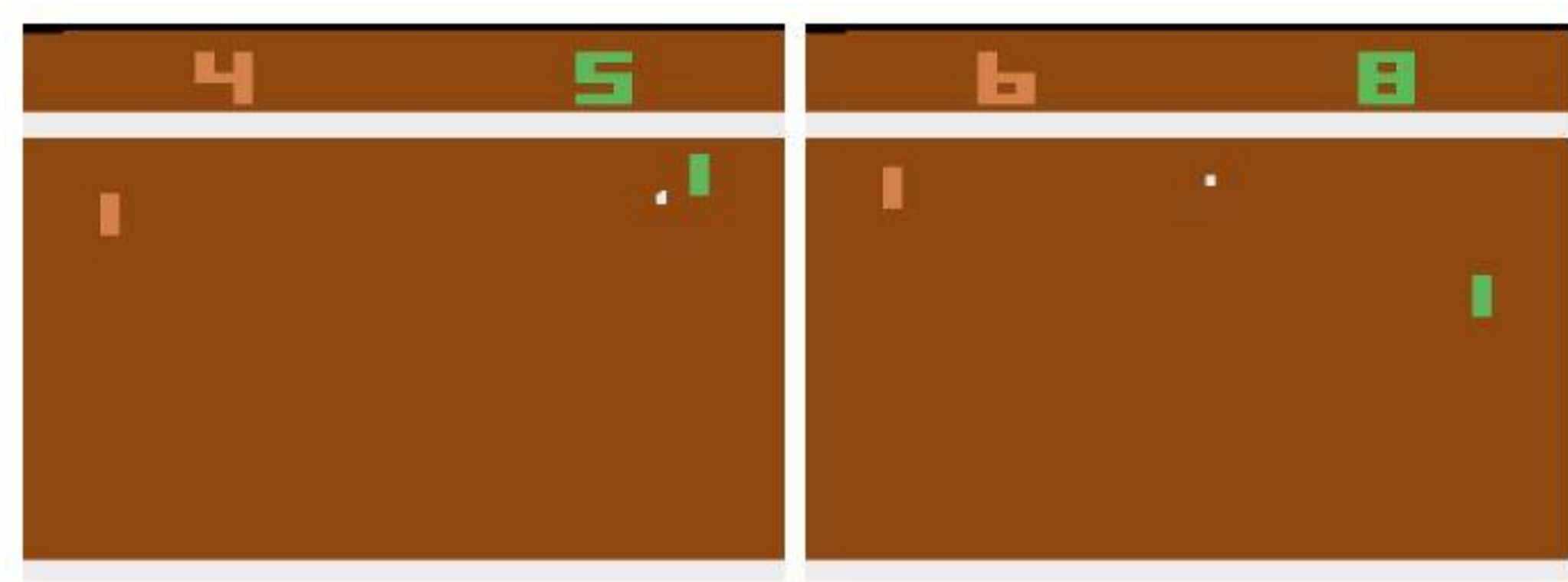
- $Q(s, a)$ é o valor Q para o estado s e a ação a .
- α é a taxa de aprendizado.
- r é a recompensa recebida após a execução da ação a no estado s .
- γ é o fator de desconto.
- s' é o próximo estado após executar a ação a .
- a' é a próxima ação escolhida no próximo estado s' .
- $\max(Q(s', a'))$ é o valor Q máximo para o próximo estado.

Treinamento e otimização

- As ações possíveis A são mover a barra que o jogador controla para cima ou para baixo.
- As recompensas $R(S; A; S_0)$ são recebidas quando a bola chega ao fim da tela do lado esquerdo ou direito, gerando uma positiva se chegar do lado do adversário e negativa se chegar do lado do jogador.
- As probabilidades de transição $P(S; A; S_0)$ são as probabilidades de o jogo estar em um estado S , por exemplo com a bola sendo rebatida pelo jogador, e transitar para algum outro estado futuro S_0 , como marcar um ponto, após tomar uma ação A , como mover a barra para cima.

Treinamento e otimização

PONG



Vantagens e Desvantagens

VANTAGENS

- Alta capacidade de generalização;
- Capacidade de aprender a partir de grandes quantidades de dados;
- Potencial para lidar com problemas complexos e de grande escala;

DESVANTAGENS

- Requer grande poder computacional
- Sensível à inicialização e hiperparâmetros
 - Instabilidade durante o treinamento

Exemplo(s) de aplicação

- **Jogos de Vídeo:** (Sucesso na aprendizagem de jogos de Atari);
- **Robótica:** (Controlar o movimento de robôs em ambientes complexos e dinâmicos);
- **Gerenciamento de Energia:** (Otimizar o consumo de energia em edifícios);
- **Controle de Tráfego:** (Otimizar o fluxo de tráfego, minimizar congestionamentos e reduzir o tempo de viagem);

Comparação com outros algoritmos

Q-Learning Clássico

Usa tabelas de pesquisa para armazenar os valores Q para todos os pares estado-ação possíveis.

Policy Gradient Methods

São baseados na otimização direta da política

A3C

Utiliza múltiplos agentes (atores) em paralelo para explorar e coletar experiências

Perguntas?

Referências

- <https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf>
- [https://keras.io/examples/rl/deep q network breakout/](https://keras.io/examples/rl/deep_q_network_breakout/)
- [https://www.tensorflow.org/agents/tutorials/0 intro rl?hl=pt-br](https://www.tensorflow.org/agents/tutorials/0_intro_rl?hl=pt-br)
- <https://proceedings.mlr.press/v120/yang20a>
- <https://ojs.aaai.org/index.php/AAAI/article/view/11757>

<https://forms.gle/62Bv1yh13WTHxXTr8>

Obrigado!