Agente Orientado por Objetivos Usando Deep Q-Learning e PettingZoo para Métodos de Busca

Vitor Leandro Machado, Rodrigo Lucas Rosales

Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie

São Paulo – SP - Brasil

10409358@mackenzista.com.br, 10365071@mackenzista.com.br

Abstract. *Currently, we have algorithms like BFS and A* that are responsible for finding the best path in a known scenario. But what if we consider real environments? Imagine an automated robot responsible for cleaning a room, an autonomous motorcycle making deliveries, or even Joe Delivery, who could be delivered to your home by a robot. How would these agents behave? How can we reinforce their learning and measure success rates, efficiency, and effectiveness? Our work proposes creating a navigation agent capable of navigating unfamiliar routes, gradually learning, through reinforcement, to find the best path. The result is an intelligent agent that handles unexpected events, optimizes routes, and moves efficiently from origin A to destination B..*

Resumo. *Atualmente, temos algoritmos como BFS e A* responsáveis por buscar caminhos em grafos ou redes. Mas, e se pensarmos em demais ambientes? Imagine, por exemplo, um robô autônomo responsável pela limpeza de um ambiente ou um veículo autônomo realizando entregas em uma cidade. Como esses agentes se comportariam? Como podemos reforçar seu aprendizado e medir taxas de sucesso, eficiência e eficácia? Nosso trabalho propõe criar um agente de navegação capaz de percorrer trajetos desconhecidos, aprendendo gradativamente, por meio de reforço, a encontrar o melhor caminho. O resultado é um agente inteligente que lida com imprevistos, otimiza percursos e se movimenta de forma eficiente de uma origem A até um destino B.*

1. Introdução

Nesta seção separamos a introdução em 5 etapas onde explicamos e contextualizamos algumas capacidades de nosso agente.

1.1 Contextualização

Em um mundo moderno, onde temos cada vez mais ações tomadas por robôs e IA, com o intuito de automatizar tarefas, por que não pensarmos em tarefas intrínsecas ao nosso cotidiano? Como a entrega de um pedido no iFood, o pedido feito no Zé Delivery, ou um robô inteligente que efetua toda a limpeza de uma casa. O tema abordado não é nenhum pouco futurista e já convivemos com ele em alguns ambientes.

A exemplo do robô responsável pela limpeza: compramos uma miniatura de robô inteligente que faz todo o mapeamento do ambiente e atua na remoção de pó e sujeiras; alguns deles até conseguem passar pano no chão. Isso nos mostra o quão poderosos são esses recursos e nos leva a explorar cada vez mais sua capacidade.

O intuito não é substituir uma tarefa humana ou o próprio ser humano, mas sim pensarmos: como posso ter um auxiliar para uma determinada tarefa? E se, por acaso, eu estiver muito ocupado, posso contar com alguém para fazer isto por mim? Esta crítica e pensamento se aplicam à nossa justificativa.

1.2 Justificativa

Podemos explorar recursos que se tornam relevante diante da necessidade de automatizar tarefas que envolvem deslocamento em ambientes complexos ou desconhecidos. Esse tipo de agente aprende a tomar decisões por meio de um processo de acertos e erros, aprimorando continuamente sua capacidade de atingir um destino específico, do ponto A ao ponto B, mesmo diante de obstáculos ou caminhos desconhecidos.

Esse aprendizado adaptativo permite que o agente identifique rotas mais eficientes, evite obstáculos e corrija seus próprios erros, sem depender de intervenção humana constante. Além disso, ele demonstra como recursos inteligentes podem auxiliar em tarefas cotidianas de transporte, entrega ou logística, aumentando a eficiência e reduzindo a possibilidade de falhas.

A criação de um agente de navegação também abre espaço para simulações em ambientes variados, possibilitando estudar comportamentos, testar estratégias de tomada de decisão e explorar soluções que poderiam ser aplicadas em robótica, veículos autônomos ou sistemas de entrega automatizados. Dessa forma, nosso trabalho busca compreender e demonstrar o potencial de agentes inteligentes capazes de aprender e se adaptar

1.3 Objetivo

Criamos um agente que seja capaz de andar por percursos desconhecidos e aprender através de acertos e erros.

A ideia é treinarmos o agente para atingir uma faixa ideal de sucesso, que colocamos como 90% dos caminhos percorridos. Claro que é uma tarefa árdua, dado que o agente teria de conhecer todos os caminhos possíveis para atingir essa faixa de 90%.

Mas pensamos em simular os ambientes sobre um grid, contendo obstáculos. Conforme o agente avança e aprende, pretendemos aumentar o grid, como se estivéssemos ampliando seu mapa de conhecimento, ou seja, o caminho percorrido. A ideia é chegarmos a um grid de 500x500, representando no final uma simulação de entrega nos arredores de uma determinada região.

1.4 Opção do projeto

Optamos por seguir com os percursos em grid, visando o treinamento do agente por meio de grades, aumentando sua complexidade conforme o agente faz o reconhecimento e obtém uma certa taxa de sucesso no ambiente.

2. Fundamentação Teórica

O estudo aborda uma variedade de conceitos e tópicos, que são apresentados de forma resumida nas subseções seguintes.

2.1 Aprendizado por Reforço

O aprendizado por reforço é um processo baseado no mecanismo de tentativa e erro, no qual um agente interage com o ambiente e ajusta seu comportamento a partir das consequências de suas ações. A cada interação, o agente recebe um retorno em forma de recompensa ou punição: recompensas são atribuídas quando a ação realizada contribui para alcançar o objetivo esperado, enquanto punições ocorrem quando a ação conduz a resultados indesejados. Dessa forma, o agente aprende progressivamente a identificar quais estratégias são mais eficazes para atingir sua meta. Com o tempo e a exposição a diferentes cenários, ele aprimora sua tomada de decisão, aumentando a probabilidade de sucesso em situações cada vez mais complexas.

2.2 Agentes Inteligentes

Um agente inteligente é uma entidade que identifica o seu ambiente, toma ações de forma autónoma para atingir objetivos e pode melhorar o seu desempenho através da aprendizagem automática ou adquirindo conhecimento. Segundo **Bringsjord, Selmer; Govindarajulu, Naveen Sunda (2020)** os agentes inteligentes operam com base numa função objetiva, que encapsula os seus objetivos. Eles são projetados para criar e executar planos que maximizem o valor esperado desta função após a conclusão.

Por exemplo, um agente de aprendizado por reforço tem uma função de recompensa, que permite aos programadores moldar e atingir o seu comportamento desejado [Wolchover, Natalie 2020].

O agente inteligente, é aquele que adota a melhor ação possível perante uma situação, estando presente na resolução duma infinidade de problemas dos utilizadores comuns. Hoje, a internet conta com diversas iniciativas que utilizam agentes, desde sites que comparam preços de bens de consumo, a mecanismos de buscas inteligentes, que navegam dentro das páginas web, apresentando o resultado da busca classificado pelo grau de precisão e relevância dos assuntos.

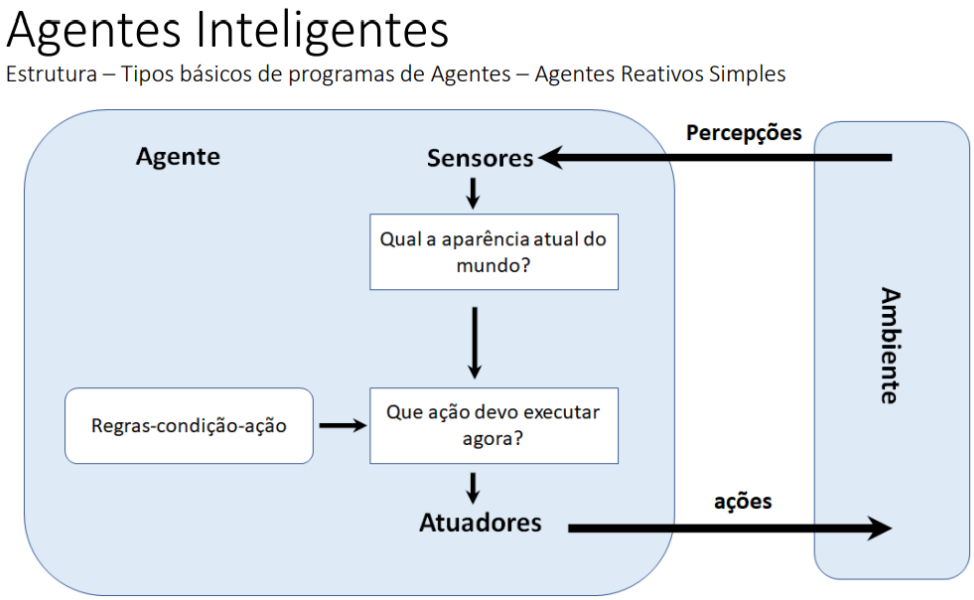


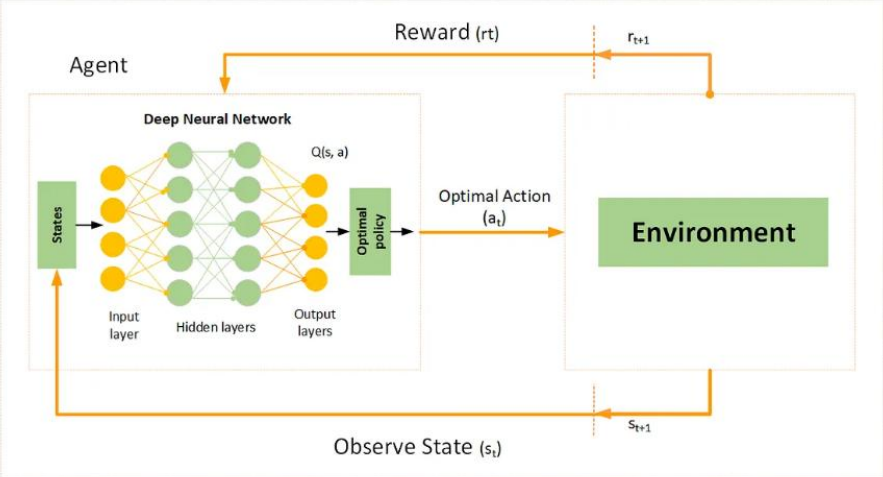
Figura 1 - Imagem estrutura agente inteligente - PDF '02_IA_AgentesInteligentes_aula.pdf' , professor Ivan

2.3 Deep Q-Learning ou Deep Q Network (DQN)

Uma das técnicas mais conhecidas no campo, aprendizado por reforço, é o **Q-Learning**, algoritmo baseado em valores (value-based), no qual o agente aprende uma função de valor de ação Q(s,a)Q(s, a)Q(s,a), que estima o retorno esperado ao executar a ação '**a**' em um estado '**s**'.

No entanto, o Q-Learning tradicional enfrenta limitações quando aplicado a problemas de alta dimensionalidade, como ambientes com grandes espaços de estados (ex.: jogos de Atari ou ambientes de robótica). Para superar esse desafio, surge o **Deep Q-**

Learning (DQN), é uma extensão do algoritmo básico de Q-Learning. O Deep Q-Learning supera essa limitação substituindo a tabela Q por uma rede neural que pode aproximar os valores de Q para cada par estado-ação



Estrutura do DQN

Figura 2 - Rede Deep Q-Learning (DQN) retirada de Amin (2020)

2.4 Treinamento

O agente será treinado atráves de erro e acerto, seguindo o modelo baseado na aprendizagem por reforço. A ideia é expormos o agente a cenários variados e permitir que ele siga aprendendo conforme erra. Iremos capturar suas métricas e medir sua taxa de sucesso conforme avança.

A ideia é atingirmos um cenário onde o agente consiga atingir o objetivo com maestria e minimizando colisões e optando por caminhos ótimos.

2.5 Petting Zoo

Desenvolvida por Terry et al. (2021), a **biblioteca PettingZoo** é uma extensão da API Gym, criada por Brockman et al. (2016), que adiciona suporte a ambientes multiagente enquanto mantém as funcionalidades e padrões já estabelecidos pelo Gym, incluindo wrappers, gerenciamento de ações, passos e observações. Embora o PettingZoo seja projetado para múltiplos agentes, ele também pode ser utilizado em cenários com **apenas um agente**, mantendo a compatibilidade com a API e permitindo que experimentos sejam facilmente escaláveis para múltiplos agentes no futuro.

O uso do PettingZoo neste trabalho tem como objetivo **padronizar o ambiente**, garantindo compatibilidade com diferentes algoritmos de aprendizado por reforço e mantendo a possibilidade de expansão para múltiplos agentes futuramente. Além disso, a biblioteca facilita a replicabilidade de experimentos, a comparação de resultados e a implementação de wrappers e estratégias de treinamento já existentes na API Gym, mesmo em cenários de agente único.

3. Descrição do problema

Com o avanço das máquinas autônomas e a popularização de agentes de inteligência artificial, surge a necessidade de desenvolver soluções que atuem como aliados do ser humano em diferentes contextos de trabalho e lazer. Um exemplo simples é o robô de limpeza, que auxilia nas tarefas domésticas. Porém, podemos expandir esse conceito para profissões como motoboys, entregadores ou até mesmo investidores: como criar agentes capazes de gerar valor e executar tarefas mesmo enquanto o usuário descansa, viaja ou realiza outras atividades?

Nosso projeto busca explorar esse cenário por meio da implementação de um **agente inteligente baseado em métodos de busca**, capaz de percorrer rotas e caminhos de forma eficiente para resolver problemas específicos de clientes. Um caso prático seria a entrega de produtos, como pizzas ou bebidas (ex.: Zé Delivery), utilizando veículos autônomos (motos ou carros), nos quais o agente atua como responsável pelo planejamento e execução do trajeto de maneira autônoma e otimizada.

4. Ética e Responsabilidades no uso da IA

Nesta seção abordamos como pretendemos aplicar e seguir práticas recomendas quanto ao uso da IA. Abrangendo como a solução será concebida, treinada, validada e, eventualmente, aplicada em contextos reais.

4.1. Transparência e Explicabilidade

Os modelos de IA, principalmente aqueles baseados em aprendizagem por reforço podem, muitas vezes, serem interpretados como ‘caixas-pretas’, onde nem sempre é simples entender por que uma decisão foi tomada.

Avaliamos possíveis riscos onde em aplicações críticas (por exemplo, navegação em ambientes reais) decisões inesperadas possam gerar comportamentos perigosos

Pretendemos seguir práticas para o desenvolvimento, olhando algumas diretrizes impostas pelo MIT no link: https://mittechreview.com.br/desafios-eticos-tecnicos-ia-autonoma/?srsltid=AfmBOorIReudo9rt_q48DW0W_6ToPEZPxLe4VfLc49rC0Ge7v2OwR5jK

4.2 Segurança e Confiabilidade

Para reduzir falhar é importante expor o agente a cenários variados, casos extremos, com intuito de reduzir falhas. Importante atuar com cenários de baixa, média e alta complexidade. Medindo como o agente se comporta, visando criar limites de segurança no ambiente e mecanismos de fallback que evitem riscos so usuário ou ao entorno.

4.3 Privacidade

Se a solução for aplicada em ambientes que coletam dados (ex.: mapas reais, sensores, câmeras), surge a preocupação com a **privacidade de informações.** Para garantir não

expor nenhuma informação dos usuários, iremos adotar um metodologia de anomização e coleta mínima de dados

5. Conteúdo, Origem e Preparação dos Dados

O projeto não conta com um dataset externo tradicional (como imagens ou tabelas). Os dados são **gerados dinamicamente pelo ambiente de simulação** construído em **PettingZoo**, no qual o agente é treinado para navegar em um grid até alcançar um ponto-alvo, desviando de obstáculos.

Cada episódio do treinamento gera trajetórias compostas por **estados, ações, recompensas e próximos estados**. Esses registros equivalem ao que seria um dataset tradicional em problemas supervisionados, mas aqui são produzidos em tempo real à medida que o agente interage com o ambiente.

O conjunto de informações coletadas durante as interações do agente com o ambiente pode ser descrito da seguinte forma:

1. O agente perde ponto a cada passo. Com intuito de que ele siga o caminho mais curto e chegue ao objetivo o mais rápido que puder
2. Caso o agente colida com obstáculos, perderá 5 pontos.
3. Caso atinja o objetivo final, ganhará 100 pontos

A estrutura corresponde, adota ao formato clássico da função Q, do Q-Learning.

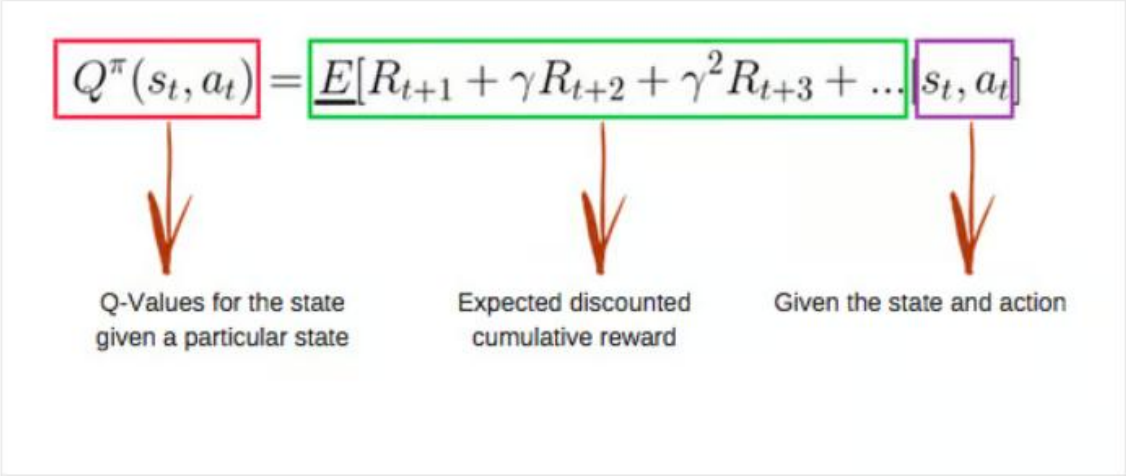


Figura 3 - Imagem de freecodecamp.org

A análise exploratória ocorre através dos dados gerados por simulação. A medida que o agente é exposto a ambientes e cenários onde precisar encontrar a estratégia ideal para atingir a meta.

Elencamos através de 4 métricas para entender a evolução do agente, conforme é aumentada gradativamente as simulações:

- Distribuição de recompensas ao longo dos episódios (para verificar se o agente está aprendendo);
- Frequência de colisões com obstáculos;
- Quantidade média de passos até atingir o objetivo;
- Taxa de sucesso (% de episódios em que o agente chega ao destino).

6. Resultados

Nesta seção, apresentamos os resultados que pretendemos obter com o agente ao final do desenvolvimento deste trabalho

6.1 Capacidade de Navegação Eficiente

- O agente deve ser capaz de alcançar consistentemente o ponto de destino (B) a partir da origem (A) em diferentes configurações de ambiente.
- Espera-se que a trajetória escolhida seja ótima ou próxima do caminho mais curto, minimizando o número de passos

6.2 Desvio de Obstáculos

- O agente deve aprender a evitar colisões com obstáculos, adaptando sua trajetória de acordo com a disposição do ambiente.
- A taxa de colisão deve diminuir ao longo do treinamento, indicando aprendizado efetivo.

6.4 Métricas Quantitativas Esperadas

Entre os principais indicadores de desempenho a serem alcançados:

- Taxa de sucesso: percentual de episódios em que o agente chega ao objetivo (>90% após treinamento).
- Recompensa acumulada média: valores positivos crescentes ao longo do treinamento, refletindo aprendizado consistente.
- Número médio de passos por episódio: convergindo para valores próximos ao caminho ótimo.
- Redução da taxa de colisão: idealmente próxima de 0 após convergência.

6.5 Visualização e Evidências

Para demonstrar os resultados, pretendendemos apresentar:

- Curvas de aprendizagem: evolução da recompensa média e da taxa de sucesso por episódio.

- Heatmaps de trajetórias: mostrando o padrão de movimentação do agente no grid.
- Comparação antes/depois do treinamento: evidenciando a diferença entre um agente aleatório e o agente treinado.

7. Referências

AWAN, Abid Ali. *Uma introdução ao Q-Learning: um tutorial para iniciantes*. DataCamp, 24 abr. 2024. Disponível em: https://www.datacamp.com/pt/tutorial/introduction-q-learning-beginner-tutorial. Acesso em: 26 set. 2025.

LUU, Quang Trung. *Q-Learning vs. Deep Q-Learning vs. Deep Q-Network*. Baeldung, 12 fev. 2025. Disponível em: https://www.baeldung.com/cs/q-learning-vs-deep-q-learning-vs-deep-q-network. Acesso em: 26 set. 2025.

Q-learning. *Wikipedia*. Disponível em: https://en.wikipedia.org/wiki/Q-learning. Acesso em: 28 set. 2025.

KOVALCHUK, Gregory. *A Beginner's Guide to Q-Learning: Understanding with a Simple Gridworld Example*. Medium. Disponível em: https://medium.com/@goldengrisha/a-beginners-guide-to-q-learning-understanding-with-a-simple-gridworld-example-2b6736e7e2c9. Acesso em: 28 set. 2025.

An Introduction to Q-Learning (Reinforcement Learning). FreeCodeCamp. Disponível em: https://www.freecodecamp.org/news/an-introduction-to-q-learning-reinforcement-learning-14ac0b4493cc/. Acesso em: 27 set. 2025.

PettingZoo Documentation. PettingZoo. Disponível em: https://pettingzoo.farama.org/. Acesso em: 20 set. 2025.

AMIN, Samina. *Deep Q-Learning (DQN)*. Medium. Disponível em: https://medium.com/@samina.amin/deep-q-learning-dqn-71c109586bae. Acesso em: 27 set. 2025.

FARAMA. *Gymnasium Basics: Criação de Ambientes*. Disponível em: https://gymnasium.farama.org/introduction/basic_usage/. Acesso em: 28 set. 2025.

SILVA, Lucas Souza; OLIVEIRA, Ivan Carlos Alcântara de. *Ambiente Multiagente para aplicação de Técnicas de Aprendizado por Reforço na Análise de Fluxo de Atendimento em Locais Fechados*. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade Presbiteriana Mackenzie, São Paulo, 2023. Disponível em: biblioteca mackenzie. Acesso em: 26 set. 2025.

WIKIPÉDIA. Agente inteligente. Disponível em: https://pt.wikipedia.org/wiki/Agente_inteligente. Acesso em: 27 set. 2025

STANFORD ENCYCLOPEDIA OF PHILOSOPHY. Artificial Intelligence (Summer 2020 Edition). Disponível em: https://plato.stanford.edu/archives/sum2020/entries/artificial-intelligence/#InteAgenCont. Acesso em: 27 set. 2025

8. Bibliografia

PYKES, Kurtis. *Reinforcement Learning: An Introduction With Python Examples*. DataCamp. Disponível em: https://www.datacamp.com/tutorial/reinforcement-learning-python-introduction. 2025.

FARAMA. *PettingZoo Documentation*. Disponível em: https://pettingzoo.farama.org/. 2025.

AWAN, Abid Ali. *Uma introdução ao Q-Learning: um tutorial para iniciantes*. DataCamp, 24 abr. 2024. Disponível em: https://www.datacamp.com/pt/tutorial/introduction-q-learning-beginner-tutorial. 2025.

AMIN, Samina. *Deep Q-Learning (DQN)*. Medium, 14 set. 2024. Disponível em: https://medium.com/@samina.amin/deep-q-learning-dqn-71c109586bae. 2025.

NEVES, Enzo Cardeal. *Aprendizado por Reforço #1 — Introdução*. Turing Talks, 23 fev. 2020. Disponível em: https://medium.com/turing-talks/aprendizado-por-refor%C3%A7o-1-introdu%C3%A7%C3%A3o-7382ebb641ab. 2025.