



Cassiana Fagundes da Silva

Sistemas Inteligentes



Sistemas Inteligentes

Cassiana Fagundes da Silva

IESDE BRASIL
2021

© 2021 – IESDE BRASIL S/A.

É proibida a reprodução, mesmo parcial, por qualquer processo, sem autorização por escrito da autora e do detentor dos direitos autorais.

Projeto de capa: IESDE BRASIL S/A.

Imagem da capa: Benvenuto Cellini/ IRINA SHI /Shutterstock
Envato Elements

CIP-BRASIL. CATALOGAÇÃO NA PUBLICAÇÃO
SINDICATO NACIONAL DOS EDITORES DE LIVROS, RJ

S579s

Silva, Cassiana Fagundes da

Sistemas inteligentes / Cassiana Fagundes da Silva. - 1. ed. - Curitiba
[PR] : IESDE, 2021.

92 p.

Inclui bibliografia

ISBN 978-85-387-6711-4

1. Inteligência artificial. 2. Inteligência computacional. I. Título.

20-67933

CDD: 006.3

CDU: 004.8

Todos os direitos reservados.



IESDE BRASIL S/A.

Al. Dr. Carlos de Carvalho, 1.482. CEP: 80730-200

Batel – Curitiba – PR

0800 708 88 88 – www.iesde.com.br

Cassiana Fagundes da Silva

Doutoranda em Engenharia de Produção e Sistemas pela Pontifícia Universidade Católica do Paraná (PUCPR). Mestre em Computação Aplicada pela Universidade do Vale do Rio dos Sinos (UNISINOS/RS). Graduada em Ciência da Computação pela Universidade de Santa Cruz do Sul (UNISC/RS). Professora no ensino superior ministrando as disciplinas de Linguagens de Programação, Estrutura de Dados, Sistemas Operacionais, Engenharia de Software, Análise e Modelagem de Sistemas, Inteligência Artificial, Inteligência de Negócios, entre outras.

Vídeos em QR code!



Agora é possível acessar os vídeos do livro por meio de QR codes (códigos de barras) presentes no início de cada seção de capítulo.

Acesse os vídeos automaticamente, direcionando a câmera fotográfica de seu smartphone ou tablet para o QR code.

Em alguns dispositivos é necessário ter instalado um leitor de QR code, que pode ser adquirido gratuitamente em lojas de aplicativos.

SUMÁRIO

- 1 Introdução à inteligência artificial 9**
 - 1.1 Conceitos de inteligência artificial 9
 - 1.2 Evolução e histórico da Inteligência Artificial 14
 - 1.3 Aplicações e áreas da Inteligência Artificial 17
 - 1.4 Agentes inteligentes 21

- 2 Resolução de problemas 26**
 - 2.1 Resolução de Problemas 26
 - 2.2 Resolução de problemas por meio de busca 30
 - 2.3 Busca em profundidade e busca em largura 33
 - 2.4 Busca e funções heurísticas 36

- 3 Sistemas baseados em conhecimento 41**
 - 3.1 Fundamentos e tipos de aplicações 42
 - 3.2 Estrutura geral de um sistema baseado em conhecimento 45
 - 3.3 Representação do conhecimento 48
 - 3.4 Aquisição do conhecimento 54

- 4 Aprendizado de máquina 58**
 - 4.1 Conceitos e paradigmas de aprendizado 58
 - 4.2 Aprendizado simbólico 62
 - 4.3 Aprendizado conexionista 67
 - 4.4 Aprendizado probabilístico 73

- 5 Computação evolutiva 77**
 - 5.1 Fundamentos da computação evolucionária 77
 - 5.2 Algoritmos genéticos 79
 - 5.3 Programação genética 83
 - 5.4 Algoritmos evolutivos 86

- 6 Gabarito 89**

APRESENTAÇÃO

▶ Vídeo



À medida que os recursos computacionais foram expandindo em processamento e memória, determinadas áreas passaram a exercer um papel fundamental para a comunidade científica, bem como para as empresas e as indústrias. Uma dessas áreas é a de sistemas inteligentes, que possibilitam por meio de um conjunto de recursos computacionais o desenvolvimento de programas capazes de aprender e de tomar decisões de modo automático e independente, simplesmente levando em consideração experiências anteriores.

Em outras palavras, podemos definir os sistemas inteligentes como programas capazes de se adaptarem a novos ambientes, raciocinar e entender sobre determinadas situações e relacioná-las entre si, da mesma forma que é possível reconhecer a verdade e aprender com base em suas próprias experiências.

Nesse viés, no primeiro capítulo discorreremos sobre os conceitos da Inteligência Artificial, apresentando sua evolução ao longo das últimas décadas e, principalmente, destacando os tipos de aplicações que podem ser desenvolvidas e suportadas por essa inteligência. Não obstante, ainda buscaremos entender quais as principais características e propriedades desejáveis que um determinado agente inteligente necessita exercer para resolver determinado problema.

O segundo capítulo aborda o conteúdo intitulado de **Resolução de problemas**. Sabemos que a resolução de problemas no mundo real se dá de modo bastante distinto do mundo computacional, isso é decorrente do espaço de busca, ou seja, do espaço que é necessário delimitarmos no ambiente programável para que uma determinada solução possa ser encontrada. Caso contrário, o custo e a complexidade para encontrar uma solução tornam-se inviáveis, especialmente, à nível de hardware, memória e processamento das máquinas.

Ainda sobre resolução de problemas em espaço de busca, no Capítulo 2, estudaremos vários tipos de métodos de busca por solução, nos quais estão: a busca em largura ou profundidade,

ambos com conceitos similares aos vistos em outras disciplinas, como Estrutura de dados.

O terceiro capítulo propõe reflexões sobre as estruturas e os tipos de aplicações que adotam os Sistemas Baseados em Conhecimento (SBC). Nesse sentido, para a construção de um SBC, inicialmente, compreendemos os fundamentos e os componentes existentes em sua estrutura e as possíveis formas de representar conhecimento, bem como as fases necessárias para a aquisição do conhecimento.

Diante da importância de aprimorar continuamente o desenvolvimento de programas que pensem de modo inteligente, o quarto capítulo visa estabelecer os fundamentos do aprendizado de máquina e seus diferentes paradigmas: simbólico, conexionista, probabilístico, com base em exemplos.

Por fim, no último capítulo, abordaremos os conceitos da computação evolutiva, em que destacamos como os princípios da teoria da evolução, o darwinismo, podem ser replicados e modelados em modelos matemáticos, de maneira a permitir que problemas com maior complexidade possam ser resolvidos de modo mais fácil do que pelos métodos tradicionais de programação linear etc.

Esperamos que seu estudo não termine por aqui. Com base neste material, realize as atividades propostas, leia os textos indicados e faça você mesmo sua pesquisa para aprofundar os conhecimentos aqui apresentados.

Bons estudos!

1

Introdução à inteligência artificial

Desde o surgimento dos computadores em nossas vidas e cada vez mais em nossas rotinas, há investimentos no ramo das pesquisas científicas que visam aproveitar a capacidade de processamento das máquinas com o objetivo de que estas simulem, de modo mais aproximado possível, o comportamento humano. Há 60 anos, projetos desse ramo proporcionaram mais uma vertente para a ciência da computação: a Inteligência Artificial (IA).

A Inteligência Artificial é conhecida como uma área interdisciplinar, pois adota em grande parte de seus conteúdos uma quantidade de conhecimentos advindos de outras áreas, por exemplo, cálculo e lógica matemática. Além disso, para o uso e para a aplicação de determinadas técnicas e de métodos, nessa área, é necessária plena compreensão dos fundamentos, principalmente de áreas como filosofia, linguística, psicologia e biologia, conceitos amplamente abordados em temas, em métodos e em técnicas utilizadas.

Podemos afirmar que os estudos e as pesquisas dessa vertente científica buscam métodos ou dispositivos computacionais que simulem, além do comportamento humano já citado, a capacidade dos indivíduos de resolver problemas. Ou seja, desde as primeiras pesquisas criadas utilizando Inteligência Artificial, pesquisadores visam implementar, computacionalmente, sistemas que pensem de maneira racional e semelhante aos seres humanos.

1.1 Conceitos de inteligência artificial

▶ Vídeo

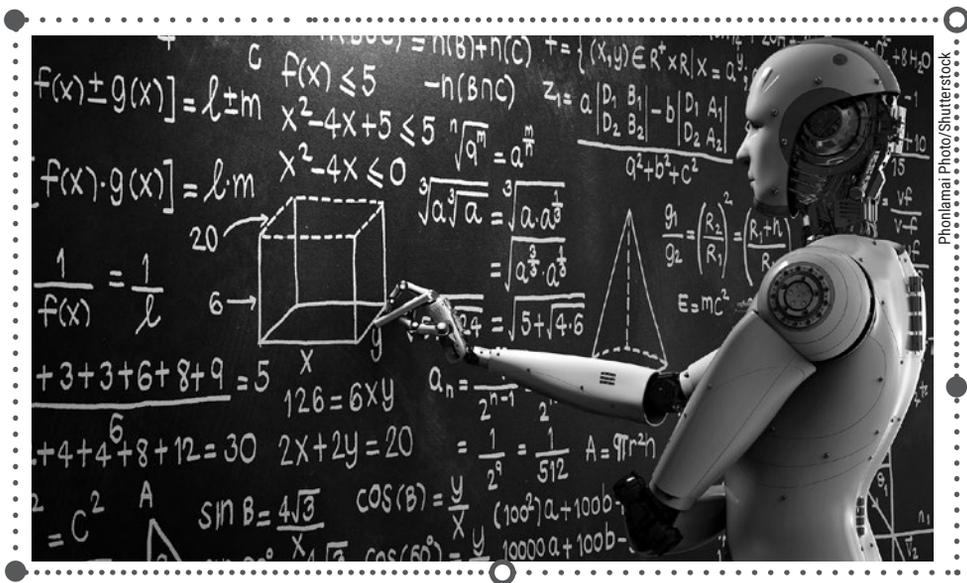


A Inteligência Artificial é uma área que está atraindo grandes interesses e chamando atenção de muitos pesquisadores ao longo dos anos e de sua história. Para muitos, as aplicações de IA podem ser bastante simples, uma vez que os filmes de ficção científica têm apresentado robôs e humanoides com comportamentos inteligentes e muito seme-

lhantes aos dos seres humanos. No entanto, mesmo com essas ideias já inseridas na sociedade, a criação de máquinas desse estilo continua sendo muito avaliada e estudada.

O desafio da construção dessas máquinas e desses robôs é consequência da ideia geral que permeia a IA, em que o objetivo principal é construir uma máquina artificialmente inteligente incorporando programas e equipamentos capazes de tomar decisões próprias para problemas de um determinado domínio para o qual foi desenvolvida, conforme ilustrado na Figura 1, em que um robô seria capaz de desenvolver um problema matemático.

 **Figura 1**
Inteligência Artificial



Vale ressaltar que, inicialmente, a IA apresentava em seus propósitos a reprodução do pensamento humano, isto é, tinha como objetivo principal reproduzir as faculdades humanas, como a criatividade, o uso da linguagem e o autoaperfeiçoamento. No entanto, notamos que o conceito de Inteligência Artificial ainda é bastante difícil de ser definido e, nesse sentido, continua sendo um conceito que dispõe de muitas interpretações, as quais podem ser conflitantes.

Para melhor compreendermos os conceitos da Inteligência Artificial vamos estudá-la, a princípio, de modo individual, ou seja, veremos, separadamente, o significado de cada um dos termos: *artificial* e *inteligência*.

Entendemos em um primeiro momento que o termo *artificial* representa algo que não é natural. Porém, no contexto científico que trazemos aqui, podemos considerar que artificial, nesse caso, tem como propósito representar habilidades humanas ou ações produzidas por humanos. Em outras palavras, podemos definir como algo que implica em uma imitação.

Já o termo *inteligência* pode ser compreendido como um conceito de consciência, identidade e mente envolvidos em um único tipo: a inteligência do ser humano. Dessa forma, a inteligência pode ser vista como uma injeção para que determinada máquina possa desenvolver um trabalho ou uma tarefa que necessitaria, em tese, de um cérebro humano. Quando pensamos na Inteligência Artificial em seu contexto, Copin (2010) acredita que ela possa ser definida como um estudo dos sistemas que agem de tal maneira que, para um observador qualquer, a máquina parece de fato ser inteligente.

Com base nesses conceitos, para que uma máquina seja considerada artificialmente inteligente deve-se levar em conta as seguintes capacidades: aprendizado, raciocínio, solução de problemas, percepção, compreensão da linguagem natural e principal objetivo da IA.

O princípio do aprendizado é permitir que, por meio do conhecimento, a máquina seja capaz de aprender com seus próprios erros, enquanto o raciocínio deve proporcionar à máquina meios possíveis de se chegar a uma conclusão conforme o problema apresentado.

Na solução de problemas, a máquina deve ser capaz de resolver o problema para o qual foi destinada e criada, por exemplo: para um problema de rotas, a máquina buscará dentro do espaço de estados quais são os caminhos que apresentam menor custo. Já a percepção está diretamente relacionada à tomada de decisão no ambiente no qual está inserida, ou seja, baseia-se em suas próprias percepções referentes ao meio. No caso da linguagem natural, as máquinas precisam ser capazes de compreender a linguagem escrita ou falada e, com base nisso, tomar decisões. Por fim, o objetivo da IA baseia-se, inicialmente, na construção de dispositivos e/ou de máquinas que reduzam os esforços humanos e, posteriormente, realizem tarefas de modo inteligente e preciso.

Russell e Norvig (2010) definem a Inteligência Artificial com base em quatro categorias, conforme o Quadro 1 elucidada. Na categoria dos

? Curiosidade

O raciocínio, normalmente, é apresentado de duas formas: dedutivo e indutivo. No raciocínio dedutivo, nós, seres humanos, derivamos 100% a precisão da conclusão em relação ao problema informado. Já o raciocínio indutivo incorre em 50% de chances de uma conclusão ser correta.

sistemas que pensam como seres humanos, Alan Turing (1950) desenvolveu um teste que chamou de *Teste de Turing*, que teve como objetivo fornecer uma definição operacional de inteligência. De acordo com Russell e Norvig (2010), o Teste de Turing foi criado para testar a capacidade de uma máquina para exibir comportamento inteligente equivalente a um ser humano, ou indistinguível deste.



Quadro 1

Definições de Inteligência Artificial em quatro categorias

Sistemas que pensam como seres humanos

“O novo e interessante esforço para fazer os computadores pensarem (...) máquinas com mentes, no sentido total e literal.”
(HAUGELAND, 1985)

“[Automação de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado.”
(BELLMAN, 1978)

Sistemas que pensam racionalmente

“O estudo das faculdades mentais pelo uso de modelos computacionais.”
(CHARNIAK; MCDERMOT, 1985)

“O estudo das computações que tornam possível perceber, raciocinar e agir.” (WINSTON, 1992)

Sistemas que atuam como seres humanos

“A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (KURZWEILL, 1990)

“O estudo de como os computadores podem fazer tarefas que hoje são melhor desempenhadas por pessoas.” (RICH; KNIGHT, 1991)

Sistemas que atuam racionalmente

“Inteligência Computacional é o estudo do projeto de agentes inteligentes.”
(POOLE *et al.* 1998)

“IA está relacionada a um desempenho inteligente de artefatos.” (NILSSON, 1998)

Fonte: Russell; Norvig, 2010, p. 5.

Ao colocar em ação esse teste, Turing (1950) definiu o comportamento inteligente como a habilidade de um sistema alcançar um

desempenho similar ao de um ser humano em todas as tarefas cognitivas, de modo que fosse possível enganar uma pessoa que o estivesse interrogando ou propondo determinado problema. O teste era composto de um computador a ser interrogado por uma pessoa sem que ela o visse, e o resultado final aprovaria o computador se, durante todo o teste, a pessoa que estivesse fazendo as perguntas não fosse capaz de identificar quem estaria respondendo: um computador ou uma pessoa.

No entanto, de acordo com Russell e Norvig (2010), para a realização efetiva do teste, o computador deveria apresentar as seguintes capacidades:

- Processamento de linguagem natural: a comunicação deveria ser efetiva e compreensível em um idioma natural, como o inglês.
- Representação do conhecimento: as informações deveriam ser recebidas, compreendidas, aprendidas e armazenadas.
- Raciocínio automatizado: as informações armazenadas deveriam ser utilizadas com a finalidade de responder às perguntas e tirar novas conclusões da situação.
- Aprendizado de máquina: as conclusões deveriam ser direcionadas para se adequarem às novas circunstâncias e para detectar ou extrapolar padrões.

Em contrapartida, o principal objetivo dos **sistemas que raciocinam como seres humanos** é compreender como esse raciocínio é feito por estes. Muitos pesquisadores têm se baseado na introspecção e nas experiências psicológicas como teorias para desenvolverem tais sistemas. Dispondo de teorias suficientemente precisas acerca do funcionamento da mente humana, torna-se possível expressar essas teorias em um programa de computador. Newell e Simon (1961), desenvolveram o GPS com o objetivo de comparar os passos de raciocínio seguidos pelo programa com os passos seguidos por várias pessoas na resolução dos mesmos problemas.

Com relação aos **sistemas que pensam racionalmente**, em 1965 existiam alguns programas que, com o tempo e a memória necessários, era possível buscar na descrição de um problema a solução, utilizando uma notação lógica. No entanto, caso a solução não fosse encontrada, o programa continuaria procurando por tempo indeterminado, ou seja, infinitamente.

A Glossário

agentes racionais: "são aqueles que agem para alcançar o melhor resultado ou, quando há incerteza, o melhor resultado esperado" (RUSSELL; NORVIG, 2010, p. 6).

☆ Importante

É importante distinguir a IA fraca e a IA forte de métodos fracos e métodos fortes. Para Coppin (2010), métodos fracos são aqueles que não contam com qualquer conhecimento ou compreensão do mundo e dos problemas que estão sendo solucionados. Enquanto os métodos fortes usam conhecimento de mundo e do problema que está sendo solucionado para agir. A abordagem dos métodos fortes é essencial para solucionar problemas complexos do mundo real usando a Inteligência Artificial.

Já **os sistemas que agem racionalmente**, também chamados de **agentes racionais**, agem com o propósito de atingir um dado conjunto de objetivos ou de crenças. Para Russell e Norvig (2010), um agente é uma entidade que percebe o ambiente no qual está inserido por meio de sensores e afeta esse mesmo espaço por meio de atuadores.

Para agir racionalmente, um agente tem alguns modos específicos de pensar que o auxiliam a identificar (inferir) qual deve ser a ação correta a ser utilizada para atingir os objetivos propostos. Por outro lado, existem situações nas quais não há uma ação correta a ser tomada, mas que exigem que algo seja feito. Em alguns casos, agir racionalmente não significa apenas inferir a ação por meio de um processo de pensamento racional, mas entender os objetivos necessários para atingir um propósito.

Diante das quatro categorias apresentadas por Russell e Norvig (2010), entende-se que a Inteligência Artificial ainda pode ser descrita em duas linhas de pensamento conhecidas como: *IA forte* e *IA fraca*.

A Inteligência Artificial forte diz respeito à criação da forma de inteligência a ser utilizada em um computador para que seja possível a ele raciocinar e resolver problemas de modo autoconsciente. Já a Inteligência Artificial fraca diz respeito a uma inteligência que não é capaz de, verdadeiramente, raciocinar e de resolver problemas; exemplo: um computador inteligente agiria como se tivesse inteligência, mas não seria capaz de ser detentor de uma autoconsciência ou da noção de si. Ainda, outro exemplo de IA fraca é a aplicação do Teste de Turing.

1.2 Evolução e histórico da Inteligência Artificial

▶ Vídeo



O interesse pela construção de máquinas capazes de realizar tarefas consideradas inteligentes é antigo e tem sido sustentado com o desenvolvimento dos hardwares disponíveis em cada época, como é o caso do ábaco e das primeiras calculadoras mecânicas fundamentadas em rodas dentadas e utilizadas desde o século I, na Alemanha (ARTERO, 2009).

Na antiguidade, a IA incorporou teorias e pensamentos de filósofos e de cientistas gregos. No entanto, o primeiro avanço concreto dessa vertente foi realizado no século IV a.C., por Aristóteles. O filósofo criou uma teoria conhecida como *lógica silogística* e, com base nela, desenvolveu-se o primeiro sistema de raciocínio dedutivo formal.

Com a Segunda Guerra Mundial e os avanços tecnológicos, vários instrumentos foram construídos com o objetivo de contribuir para o combate bélico. Para que os cálculos de bomba atômica fossem pensados com precisão, um computador foi construído e sua arquitetura foi nomeada com o sobrenome de seu fundador: Von Neumann.

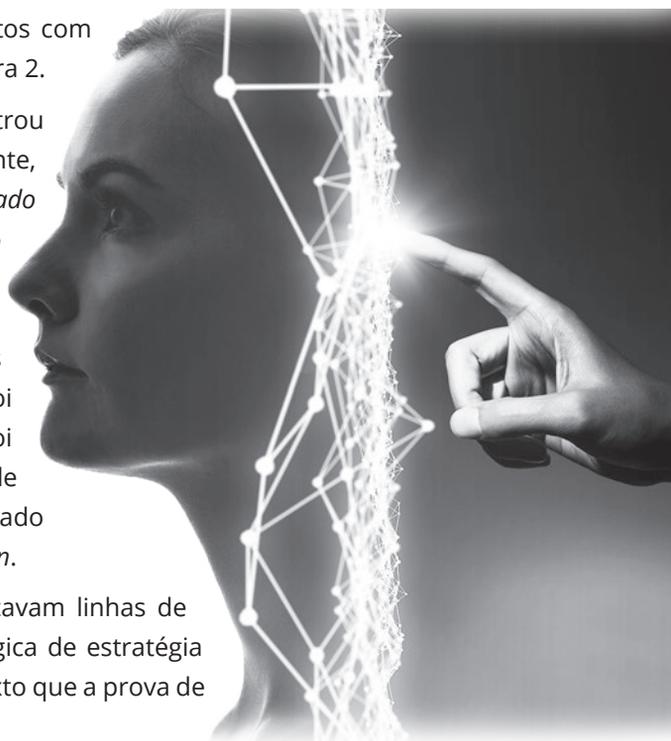
Em 1943, o primeiro trabalho reconhecido na área de IA foi publicado por McCulloch e Pitts. Nesse trabalho foi proposto um modelo de neurônios artificiais em que cada um era caracterizado como ativo ou inativo. Essa classificação se dava por meio da troca de ativação das ocorrências em respostas às estimulações de um número de neurônios vizinhos. Um neurônio artificial é um modelo simplificado e simulado do neurônio real; além disso, suas características básicas são a adaptação e a representação de conhecimentos com base em conexões, conforme ilustrado na Figura 2.

No ano de 1949, Donald Hebb demonstrou uma regra de aprendizado que, posteriormente, ficou conhecida na literatura como *aprendizado de Hebb*. Essa regra permitiu uma atualização simples e que modificaria as intensidades de conexões entre os neurônios. Já a construção do primeiro simulador de redes neurais artificiais do primeiro neurocomputador foi considerada um sucesso. Essa rede neural foi construída com base no modelo matemático de McCulloch e Pitts (1943) e na teoria do aprendizado de Hebb (1949) e recebeu o nome de *perceptron*.

A partir da década de 1950, estudos utilizavam linhas de pesquisas psicológicas para a utilização da lógica de estratégia com finalidades matemáticas e foi nesse contexto que a prova de teoremas passou a ser desenvolvida.

Em 1950, Alan Turing apresentou uma visão completa a respeito da Inteligência Artificial por meio do artigo em que divulgava o Teste de Turing que havia realizado. No ano de 1951, Marvin Minsky e Dean Edmonds desenvolveram o primeiro computador de rede neural – chamado *SNARC*. Em 1956, John MacCarthy, Minsky e demais pesquisadores iniciaram estudos relacionados à teoria de autômatos, às redes neurais e à inteligência.

 **Figura 2**
Rede neural artificial



metamorworks/Shutterstock

 **Vídeo**

Para saber mais a respeito da teoria desenvolvida por Allan Turing, recomendamos o documentário sobre ele de mesmo nome.

Disponível em: <https://www.youtube.com/watch?v=x2AXca1kPQk>.
Acesso em: 26 nov. 2020.

Na década de 1960, conhecida como uma época clássica, o objetivo da IA era simular a inteligência humana com o propósito de obter programas capazes de solucionar qualquer problema. Grande parte dos trabalhos e das pesquisas desenvolvidas nessa década foram fracassadas devido à alta complexidade dos problemas computacionais apresentados. No entanto, um programa desse período se tornou conhecido: o GPS (*General Problem Solver*), projetado para imitar protocolos humanos de resolução de problemas, incorporando a abordagem/categoria de pensar de maneira humana.

Na era romântica, caracterizada pelo período de 1970 a 1980, a IA apresentava um objetivo mais modesto, restringindo-se apenas à simulação do comportamento humano em situações restritas. Surgiram, então, os sistemas especialistas com o intuito de atingir um determinado domínio. Contudo, para o desenvolvimento deles era necessário ter uma base de conhecimento grande para a resolução de um determinado problema, gerando, novamente, alguns fracassos de implementação.

Os anos 1980 ficaram conhecidos como a *idade moderna da Inteligência Artificial*, em que foi possível observar o amadurecimento em relação às dificuldades enfrentadas nas décadas anteriores. Esse amadurecimento foi percebido, principalmente, por meio das tentativas de simulação da inteligência humana e pelo sucesso no desenvolvimento de sistemas especialistas. Em 1982, as redes neurais artificiais foram reconhecidas pelo físico John Hopfield, o qual provou a simulação de um modelo físico por um modelo matemático fundamentado na teoria de redes neurais.

Ainda nessa época, o modelo de rede neural *Backpropagation* foi criado e trouxe em sua essência uma ampliação do modelo perceptron, anteriormente desenvolvido. Já nos anos 1990, a IA apresentou grandes avanços em suas pesquisas, ampliando trabalhos desenvolvidos e focados no aprendizado de máquina, nos sistemas tutores inteligentes, no raciocínio baseado em casos, no processamento de linguagem natural, nos jogos, na realidade virtual, na robótica, nas redes neurais, entre outros.

Ainda durante esses anos ocorreu a explosão da criação de redes neurais artificiais, a qual resultou em uma grande quantidade de aplicações sendo desenvolvidas por meio desses modelos. Em vista disso,

essas redes passaram a ser comparadas às técnicas correspondentes da estatística em aplicações de reconhecimento de padrões e do aprendizado de máquina. Com base nessas aplicações também surgiu a técnica de mineração de dados, esta possibilitava a extração de algum tipo de conhecimento útil em grande volume de dados.

Por sua vez, no início do século XX, a IA começou a surgir de maneira mais similar ao que observamos atualmente devido ao surgimento de princípios matemáticos e ao uso do termo *robô*.

Já a partir de então, a IA se mostrou revolucionária não somente no âmbito de conteúdo, como também no que diz respeito à metodologia. Notamos que essa vertente está sendo utilizada em grande escala nas indústrias, possibilitando a automatização inteligente de determinadas tarefas; além da utilização de robôs ou braços mecânicos que possibilitam maior segurança e produtividade na realização dessas atividades.

As mudanças e as perspectivas na IA foram modificando ao longo dos anos, desde as primeiras ideias de criações de inteligências semelhantes à do ser humano até tornar os computadores mais eficazes por meio de softwares intelectuais que auxiliam em várias atividades, faz da IA uma área de grandes representações tanto em linguagem como em notações.

Leitura

Para saber mais sobre a evolução histórica da Inteligência Artificial ao longo dos anos, bem como o que foi desenvolvido nos principais anos de cada década, leia a seção “1.1 - Uma breve história da Inteligência Artificial” do livro *Inteligência Artificial Teórica e Prática*, de Amilr Olivette Artero.

ARTERO, A. O. São Paulo: Editora Livraria da Física, 2009.

1.3 Aplicações e áreas da Inteligência Artificial —

Vídeo



Na atualidade existe uma vasta quantidade de áreas compondo a macro área da Inteligência Artificial, uma vez que ela é decorrente de uma série de disciplinas que, ao mesmo tempo que compartilha uma abordagem essencial para a solução de problemas, também está preocupada com a diversidade de aplicações.

Diante desse cenário, várias são as áreas de atuação e aplicação da IA. Dentre elas destacam-se: a robótica, a visão computacional, o processamento de linguagem natural, os sistemas especialistas, o reconhecimento de padrões, as bases de dados inteligentes, a prova de teoremas, o aprendizado de máquina, as redes neurais, os algoritmos genéticos, os jogos etc.

Já as aplicações da IA na **robótica** se caracterizam por dispositivos robóticos capazes de realizar tarefas que exigem grande força e precisão.

Atualmente, grande parte das aplicações robóticas têm se desenvolvido nas indústrias, principalmente, após o surgimento dos conceitos da Indústria 4.0, os quais se encontram fortemente ligados à conectividade, à ciência de dados e à internet das coisas, elementos esclarecidos na Figura 3.



Figura 3

Aplicação de Inteligência Artificial na Robótica

PopTika/Shutterstock



Além disso, existem trabalhos que precisam ser feitos em ambientes perigosos, como em águas profundas nas plataformas de petróleo ou, ainda, em uma expedição no espaço lunar. Mesmo a habilidade das tarefas sendo de grande valia, as aplicações robóticas buscam inserir em seus dispositivos cada vez mais algum tipo de inteligência que os tornam capazes de tomar determinadas decisões imprescindíveis em circunstâncias mais arriscadas.

Outra aplicação de IA muito adotada nos últimos anos devido ao crescimento da internet é aquela que envolve o **processamento de linguagem natural**, isto é, a criação de programas que sejam capazes de entender e de reproduzir a linguagem humana. Os autores Russell e Norvig (2010) afirmam que a habilidade de utilizar e de compreender a linguagem natural não apenas parece ser um aspecto fundamental da inteligência humana, assim como a sua automação teria um impacto inacreditável sobre a facilidade de utilização e de eficácia dos próprios computadores.

Dentro dessa área existem várias aplicações que envolvem a tradução automática de textos. Dentre essas aplicações estão: a

sumarização de textos, os chatbots (Figura 4) e os assistentes virtuais. Essas aplicações permitem interpretar informações em formato de texto ou a voz de uma pessoa para dar repostas pontuais a respeito de determinado problema.

Já as aplicações que envolvem os **sistemas especialistas** são definidas como sistemas que, com base no processamento de uma base de conhecimento, conseguem fornecer respostas sobre um determinado domínio a um usuário. Na prática, essas aplicações são usadas para auxiliar na tomada de decisão em áreas como a medicina, a engenharia, a matemática e a administração. As aplicações em sistemas especialistas tornaram-se as mais populares devido a sua simplicidade de desenvolvimento.

Dentre os sistemas especialistas conhecidos estão: o MYCIN para a área médica; o DENDRAL para resolver problemas de elucidação da estrutura química; e o PROSPECTOR, que permite a localização e o tipo de prováveis depósitos de minérios fundamentados em informações geológicas sobre um sítio. O programa Dipmeter Advisor possibilita interpretar os resultados e os registros de perfuração de poços petrolíferos. Enquanto o XCON trata de problemas de configuração de computadores VAX.

Outra área que envolve uma gama de aplicações é a de **base de dados inteligentes**, pois consiste em adicionar às bases de conhecimento habilidades de raciocinar, de modo que seja possível gerar novos resultados. Essa área compõe as aplicações de mineração de dados, mineração de textos, análise de sentimentos, entre outras.

Figura 4
Aplicação usando o Chatbot



Figura 5
Mineração de dados

cifotart/Shutterstock



As aplicações de IA no que dizem respeito ao **reconhecimento de padrões** envolvem uma grande quantidade de tarefas, as quais estão ligadas desde o reconhecimento das faces, dos gestos, das palavras escritas e faladas, até o próprio ambiente em que estão inseridas. Essa área está diretamente relacionada à visão computacional.

Vídeo

Assista ao documentário *AlphaGO*, publicado pelo canal Deepmind, que explica a construção de um sistema de Inteligência Artificial que venceu o campeão mundial do GO por meio de um algoritmo que aprendeu a jogabilidade do sistema: um jogo de tabuleiro chinês.

Disponível em: <https://www.youtube.com/watch?v=WXuK6gekU1Y>. Acesso em: 26 nov. 2020.

A área de jogos tem adotado muitos recursos de IA na última década, pois proporciona maior realismo ao jogo, além de possibilitar a integração de vários participantes em um único ambiente. Com o uso desses recursos também é possível alterar o comportamento das personagens dependendo do tipo de usuário ou, ainda, mudar sua estratégia após uma derrota, por exemplo. Essa área envolve o uso de estratégias e raciocínio, propícios de seres inteligentes. Os recentes sucessos com base em jogos de computador incluem campeonatos mundiais de gamão, de xadrez e de futebol. Observe, na Figura 6, um exemplo de gráfico realista de um jogo.

Figura 6

Jogo utilizando o recurso de IA



A **prova de teoremas** é uma área de aplicações da Inteligência Artificial que envolve estratégias usadas para as demonstrações empregadas na resolução de problemas, ou seja, consiste na definição de uma sequência lógica de ações que levam de uma situação inicial até o objetivo final. Já a **visão por computador** é uma área em que as aplicações visam a desenvolver a capacidade de extrair informações do ambiente por meio de imagens obtidas dele. No entanto, independentemente da área de aplicação em que os recursos de IA serão adotados, eles podem se tornar um grande desafio.

1.4 Agentes inteligentes

▶ Vídeo



Os agentes inteligentes têm evoluído ao longo das décadas com o conhecimento da área de Inteligência Artificial e, principalmente, por tratarem de um recurso de programação que representa elementos autônomos, os quais têm a capacidade de manipular e de trocar informações e conhecimento.

Eles começaram a ser adotados por pesquisadores e por estudiosos na década de 1980 e são utilizados até hoje, tornando-se um campo de conhecimento amplo e com inúmeras possibilidades de estudo e de aplicação. Na área da informática, esses agentes podem ser aplicados na utilização do correio eletrônico, dando prioridade e organização às mensagens dos usuários.

Outra área de atuação dos agentes inteligentes é o comércio eletrônico. Nele, os vendedores precisam, de alguma forma, atrair o cliente para o seu portal, oferecendo suporte especializado sobre os seus produtos, além de controlar e de realizar o acompanhamento das suas vendas. Dessa forma, um agente inteligente pode ser definido como um *programa de computador*, mas que não precisa, necessariamente, apresentar um comportamento inteligente¹.

1

O termo *comportamento inteligente* é alvo de muitas controvérsias, uma vez que é difícil definir o que é realmente esse tipo de comportamento.

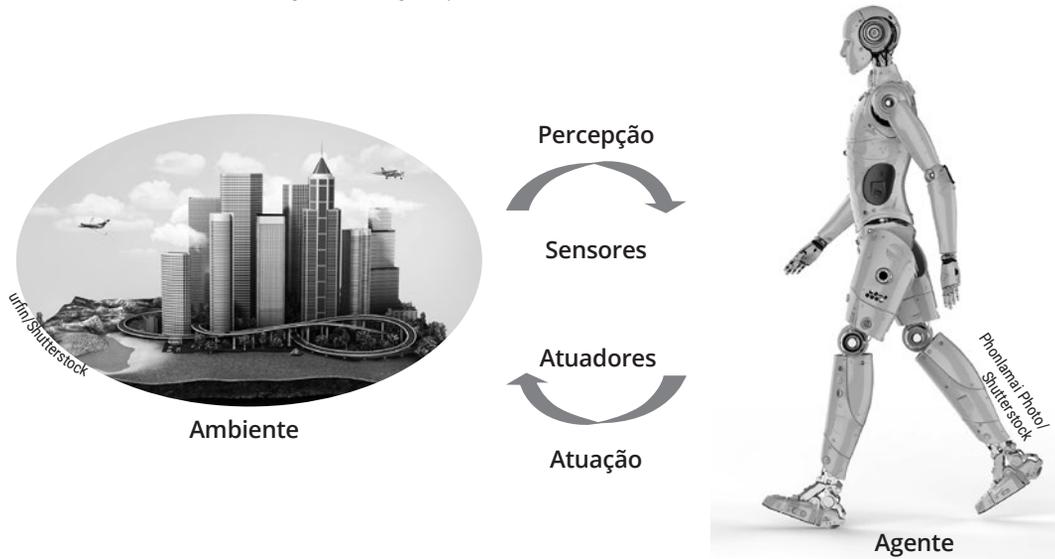
Para Coppin (2010), um agente deve ser capaz de realizar alguma tarefa com o objetivo de auxiliar um usuário humano. Esses agentes podem ser do tipo biológico, como pessoas ou animais; robótico; ou computacional. O autor também considera que eles são ferramentas que podem realizar determinada tarefa em nome de um humano; por exemplo: um simples agente de busca na internet pode ser projetado para enviar consultas a uma série de ferramentas de busca e comprar os resultados. Já para Russell e Norvig (2010), um agente é tudo aquilo considerado capaz de perceber o ambiente por meio de sensores e de agir sobre ele por meio de atuadores. Esses sensores são responsáveis pela base de conhecimento do agente, construída por meio das percepções do agente em relação ao ambiente. Já os atuadores executam as tarefas diretamente no ambiente em que o agente se encontra.

Dentre as suas características principais, os agentes inteligentes se destacam por apresentar atuação autônoma, percepção e adaptação ao ambiente em que foi inserido, bem como por trabalhar em prol dos objetivos a ele definidos.

Diante do exposto, é possível compreender que os agentes inteligentes são capazes de executar determinadas tarefas por meio da percepção, da atuação e da interpretação do seu ambiente.

A Figura 7 ilustra a interação entre um agente e o ambiente por meio de sensores e de atuadores.

 **Figura 7**
Agentes interagindo por meio de sensores e atuadores



Fonte: Adaptada de Russell; Norvig, 2010, p. 34.

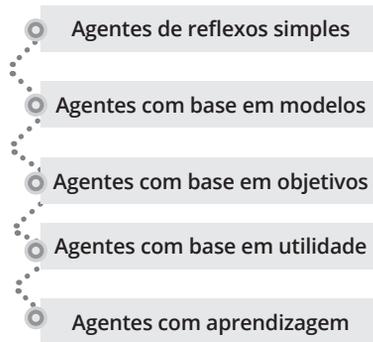
Segundo Russell e Norvig (2010), os agentes humanos possuem como sensores: os ouvidos, os olhos e outros órgãos; e seus atuadores são: as pernas, a voz, os braços, as mãos e outras partes do corpo. Já os agentes robóticos têm como sensores: as câmeras e os detectores de faixa de infravermelho; e como atuadores: os motores.

Para o bom funcionamento dos agentes inteligentes são necessárias algumas características. Os autores elencam algumas delas expostas a seguir.

- Direção: possibilidade de receber diretrizes de usuários e outros agentes.
- Autonomia: comportamento determinado pela própria experiência.
- Persistência: capacidade de operar por longos períodos sem atenção.
- Confiabilidade: desempenho de acordo com expectativas.

- Antecipação: antecipar necessidades, resultados, informações, aprendizagem.
- Atividade: capacidade de iniciar solução de problemas, fusão de informação etc.
- Cooperação: entre sistemas e outros agentes, resolução de conflitos, conflitos etc.

Todo agente é desenvolvido por meio de técnicas computacionais e de algoritmos que especificam se ele agirá individualmente ou coletivamente. Após essas definições é possível especificar como os atuadores e os sensores serão usados para a realização das tarefas. Na literatura, várias são as formas de classificação dos agentes inteligentes, dentre elas estão:



Conforme Russell e Norvig (2010), um agente de reflexo simples, também chamado *agente reativo simples*, se caracteriza por escolher a melhor ação com base naquilo que foi transmitido em determinado momento, sem considerar qualquer outro acontecimento. Os agentes com base em modelos ou agentes reflexivos adotam informações de acontecimentos anteriores para, então, analisar aspectos do estado atual ainda não ocorridos.

Quando nos referimos aos agentes com base em objetivos, para Coppin (2010), esses são mais complexos se comparados aos reativos, pois buscam e seguem determinado objetivo ao invés de seguir um conjunto de regras predeterminadas. Os agentes com base em utilidades estabelecem preferências entre os passos de seus estados, oferecendo um conjunto de possibilidades de execução de determinada ação ao agente (LUGER, 2013).

Para os agentes de aprendizado, componentes conceituais podem ser divididos em: elemento do aprendizado e elemento do desempenho (RUSSELL; NORVIG, 2010). Nessa perspectiva, Coppin (2010) classifica os agentes de acordo com a execução de suas tarefas; no entanto, deve-se levar em consideração que existem agentes híbridos, os quais misturam mais de uma funcionalidade na execução de suas tarefas.



CONSIDERAÇÕES FINAIS

A Inteligência Artificial é uma área de pesquisa da ciência da computação que busca métodos ou dispositivos computacionais simulando a capacidade humana de resolver problemas. Desde sua criação, as pesquisas nessa vertente tentam implementar, computacionalmente, sistemas que pensam e agem racionalmente, além de pensar e de agir de modo semelhante aos seres humanos. É por esse motivo que Russell e Norvig (2010) adotam um conjunto de definições para a IA com base nas categorias apresentadas e na forma como os sistemas pensam ou agem.

Durante a evolução tecnológica e, principalmente, com o surgimento dos computadores nota-se que determinadas décadas foram marcadas pelos grandes adventos da IA, em que se destaca o surgimento do Teste de Turing, o qual teve como objetivo testar a maneira como um computador se comporta quando questionado por uma pessoa. Além disso, outros destaques importantes foram sinalizados, como a criação do primeiro modelo de rede neural e o do computador neural. Outros conceitos de modelos de redes também surgiram nesse período, por exemplo, o modelo de Hebb e o aprimoramento da rede neural Perceptron para *Backpropagation*.

Notamos que durante a evolução histórica várias aplicações e áreas de IA foram expandindo em representatividade e em diversidade de aplicações para a resolução de problemas. Dentre essas vertentes destacam-se os jogos, o reconhecimento de padrões, a visão computacional, o processamento de linguagem natural, o aprendizado de máquina, as redes neurais, os sistemas especialistas, a prova de teoremas, os sistemas inteligentes, entre outras.

Não obstante, quando se desenvolve uma aplicação que adota um recurso inteligente faz-se necessário a utilização de um agente. Este, por sua vez, possibilita a interação entre o ambiente e os usuários da aplicação por meio do uso de atuadores e de sensores. Como exemplo de um agente inteligente cita-se o robô aspirador de pó, que pela implementação de um conjunto de regras e de seus sensores e atuadores torna possível a limpeza de um ambiente de maneira inteligente e automatizada.



ATIVIDADES

1. O que é a Inteligência Artificial?
2. Qual é o objetivo do Teste de Turing?
3. Cite uma área de atuação da IA.
4. O que é um agente?



REFERÊNCIAS

- ARTERO, A. O. *Inteligência artificial: teórica e prática*. São Paulo: Livraria da Física, 2009.
- COPPIN, B. *Inteligência artificial*. Rio de Janeiro: LTC, 2010.
- HEBB, D. O. *The Organization of Behavior: A neuropsychological theory*. New York: Wiley, 1949.
- LUGER, G. F. *Inteligência artificial*. 6. ed. São Paulo: Pearson Education do Brasil, 2013.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, v. 5, p. 115-133, 1943.
- NEWELL, A; SIMON, H. The simulation of human thought. *Current Trends in Psychological Theory*, Pittsburgh, n. 1, p. 152-179, jun. 1961.
- TURING, A. Computing machinery and intelligence. *Mind*. New Series, vol. 59, n. 236, p. 433-460, oct. 1950.

2

Resolução de problemas

A capacidade das pessoas para resolver problemas de diferentes naturezas é um grande indicador de inteligência. Nesse sentido, quando são desenvolvidas aplicações de Inteligência Artificial, a resolução de problemas também é um fator crucial, pois, para que uma tarefa seja executada em um computador, faz-se necessário encontrar estruturas capazes de representar essa resolução, além de fazer o processamento em cima de determinada representação.

Diante disso, quando se opta pela Inteligência Artificial no processo de resolução de problemas é necessário que as estruturas escolhidas sejam representadas, dentre as mais utilizadas na prática estão as árvores e os grafos, para os quais existe um bom conjunto de técnicas destinadas ao seu processamento.

2.1 Resolução de Problemas

▶ Vídeo



Problemas relacionados à Inteligência Artificial (IA) comumente adotam em suas aplicações a resolução de problemas por espaço de busca. Tais situações podem ser resolvidas de duas formas: por procedimentos determinísticos ou por problemas computacionais.

No que tange aos problemas computacionais, podemos observar que nem sempre os problemas do mundo real podem ser computacionalmente resolvidos, uma vez que apresentam várias complexidades. Como alternativa, esses problemas podem ser solucionados por meio de uma busca, por exemplo, busca em largura, busca em profundidade, entre outras (Figura 1).

Problemas que utilizam a IA em suas soluções podem ser representados por grafos e árvores. Os grafos são constituídos por nós e arcos, esses nós representam os vértices ou estados de uma determinada situação que podem ocorrer em diferentes etapas da resolução; enquanto os arcos representam ações que transformam um estado em outro

 **Figura 1**
Análise de um problema

com o objetivo de encontrar uma solução (ARTERO, 2009). Já as árvores são compostas por nó raiz e nós intermediários e/ou folhas (últimos nós na estrutura da árvore).

Um exemplo de resolução de problemas por meio do uso de grafos é citado por Luger (2013): o clássico problema dos missionários e canibais, que têm como objetivo encontrar uma sequência de operações necessárias para que três missionários e três canibais façam a travessia de um rio utilizando apenas um barco com capacidade de transporte para dois passageiros por vez. Além disso, para a resolução desse problema deve-se levar em consideração que o número de canibais jamais pode ultrapassar o número de missionários em qualquer uma das margens do rio.

Outro problema que pode ser solucionado por meio do uso de grafos é o jogo dos quadrinhos, também conhecido como jogo do 8. Esse jogo consiste em um tabuleiro com espaço para nove peças, no entanto existem apenas oito e um espaço vazio, que permite que uma peça seja deslocada para esse espaço (Figura 2).

 **Figura 2**
Jogo do 8

4	5	8
	1	6
7	2	3

1	2	3
4	5	6
7	8	

Fonte: Adaptada de Artero, 2009.

Há também o problema de 8 rainhas. Essa técnica visa posicionar oito rainhas em um tabuleiro de xadrez de modo que nenhuma ataque qualquer outra peça (Figura 3).

Nesse problema, a rainha pode se movimentar no tabuleiro em linhas retas pelas fileiras, colunas e diagonais. Porém, não pode pular por cima de suas próprias peças ou adversárias e capturar a casa ocupada pela adversária. Esse problema é matemático, principalmente

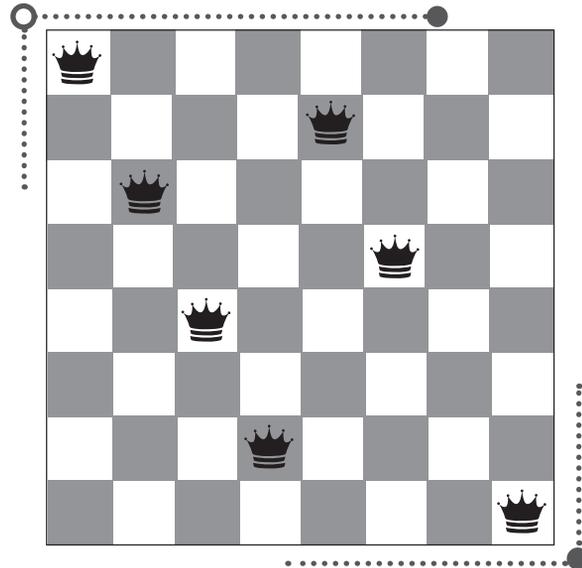


Song_about_summer/Shutterstock.

Análise de um problema por meio de uma representação (analogia) de grafos.

quando é necessário colocar 8 rainhas em um tabuleiro de xadrez, na dimensão 8x8, respeitando o fato de que não podem ocorrer ataques entre si. Como alternativa a essa solução é necessário que duas rainhas não fiquem em uma mesma linha, coluna ou diagonal.

 **Figura 3**
Problema das 8 rainhas



Um exemplo de problema representado por árvores pode ser representado pela distribuição de horários dos professores em uma determinada escola. Neste espaço de busca, em que estão todos os professores e suas disponibilidades de horário, é necessário que sejam percorridas, através dos nós das árvores todas as possíveis soluções para encontrar quais combinações melhor atendem às necessidades do horário em relação à disponibilidade do professor.

Para Coppin (2010), diante dos exemplos apresentados, podemos perceber que um problema pode ser considerado um objetivo que é possível ser alcançado por meio de um conjunto de ações. Enquanto Russell e Norvig (2010) afirmam que um problema pode ser definido pela decisão de sequência de ações e estados a serem considerados, dado um objetivo.

Neste sentido, vale ressaltar que, ao resolver um problema usando como solução a busca, é preciso inicialmente formular o objetivo que se deseja alcançar. Em vista disso, Coppin (2010) define busca como um método que pode ser utilizado por computadores para examinar um

espaço de problemas a fim de encontrar um objetivo. Um espaço de problema também pode ser considerado como um espaço de busca, uma vez que para solucionar o problema é necessário realizar a busca no espaço por um estado objetivo.

Desse modo, de acordo com Russell e Norvig (2010), um problema pode ser formalmente definido com base em quatro componentes:

- Um **estado inicial** que descreve a situação do problema e, principalmente, onde um agente inicia a sua atividade. Dentre as ações disponíveis, para o agente é a **função sucessor**. Entendemos essa função como o que será realizado após uma determinada ação do agente.
- **Espaço de estados** refere-se ao estado inicial e a função sucessora, ou seja, a sequência de passos que pode ser realizada a partir do estado inicial. Já o **caminho** é compreendido como uma sequência de rotas conectadas através de um conjunto de ações realizadas.
- O **teste de objetivo** tem como funcionalidade verificar se o estado é um objetivo, isto é, verificar se o estado chegou ao seu ponto final.
- **Custo do caminho** é quando atribui-se uma função matemática/numérica ao percurso realizado. Assim, torna-se possível, por meio dessa função, escolher o maior ou menor valor de custo, dependendo do problema a ser tratado.

Todos os quatro componentes aqui descritos elucidam um problema e podem ser apresentados em uma única estrutura de dados oferecida como entrada para um algoritmo de resolução de problemas. E a **solução ótima** é dada pelo menor custo do caminho entre todas as soluções apresentadas. Como solução ótima entendemos aquela que apresenta um melhor resultado se comparada as demais opções encontradas, ou seja, nesse exemplo, a solução ótima pode ser identificada como menor caminho ou menor custo do caminho.

Conforme vimos anteriormente, uma das técnicas usadas para a representação de resolução de problemas de busca é por meio de grafos ou árvores. Contudo, é importante diferenciarmos o conceito de espaços de busca e árvores de busca. No exemplo do jogo das 8 rainhas existem apenas 8 estados no espaço de busca, mas um número infinito de caminhos nesse espaço de estados. A árvore de busca, assim como os grafos, também apresenta um número infinito de nós.

! Atenção

Segundo Russell e Norvig (2010), dado um estado particular x , **SUCCESSOR** (x) retorna um conjunto de pares ordenados, em que cada ação é uma das ações válidas no estado x e cada sucessor é um estado que pode ser alcançado a partir de x aplicando-se a ação.

Várias são as maneiras de representação dos nós, mas vale partir do princípio de que um nó é uma estrutura de dados com cinco componentes. Esses são descritos por Russell e Norvig (2010) da seguinte maneira:

- **Estado:** é o espaço de estados a que o nó corresponde.
- **Nó-pai:** é o nó da árvore de busca que gerou esse tipo de nó.
- **Ação:** é a ação que foi aplicada ao pai para gerar o nó.
- **Custo do caminho:** é custo desde o estado inicial até o nó, indicado pelos ponteiros do nó-pai.
- **Profundidade:** é o número de passos ao longo do caminho, considerado desde o estado inicial.

Diante desse cenário em que há possíveis caminhos e probabilidades de resolução, notamos que a complexidade dos problemas é uma questão muito importante a ser observada com base no objetivo, uma vez que pode ocorrer de o número provável de caminhos ser muito extenso ou impossível de ser processado computacionalmente. Um exemplo são os problemas combinatórios de ordem elevada, em que a probabilidade de o problema não ser resolvido é grande. Desse modo, faz-se necessário sempre verificar se o problema maior pode ser fragmentado em problemas menores, resultando em soluções menos complexas.

2.2 Resolução de problemas por meio de busca —

▶ Vídeo



Várias são as técnicas de busca que podem operar sobre as estruturas de árvores e sobre os grafos. No entanto, para que estas possam ser aplicadas, necessitam de um mecanismo de controle, de modo que todos os nós possam ser visitados, evitando, assim, que um mesmo nó seja visitado várias vezes.

Esse controle é comumente realizado com o apoio de uma lista de nós que se deseja visitar. Essa lista consiste em um procedimento genérico em que a cada nó visitado é retirado da lista para que seja aplicada a função teste de objetivo, a fim de verificar se esse corresponde a um nó meta.

Para Coppin (2010), são duas as formas de se fazer busca em uma árvore de objetivos de busca. Essas formas correspondem às abordagens *top-down* e *bottom-up*. A abordagem *top-down* envolve subdivi-

dir inicialmente o problema principal em objetivos menores e, então, subdividir esses objetivos recursivamente em outros menores ainda, e assim sucessivamente. Enquanto a abordagem *bottom-up* envolve determinar inicialmente todos os subobjetivos que sejam necessários à solução de um problema por inteiro e, dessa maneira, começar solucionando os nós de sucesso, e daí para cima até que a solução completa seja obtida.

Também há a busca guiada por dados, conhecida como *encadeamento para frente*, que parte de um estado inicial e usa ações que são permitidas para ir em frente até que o objetivo seja atingido. Já a busca por objetivos, conhecida como *encadeamento para trás*, pode iniciar sua busca no objetivo e voltar para um estado inicial, visualizando quais deslocamentos poderiam ter levado ao estado objetivo.

A busca guiada por objetivos e a busca guiada por dados produzem o mesmo resultado quando aplicadas, porém, dependendo da natureza do problema a ser resolvido, uma das abordagens pode resolver o problema mais rapidamente que a outra, principalmente em situações em que uma das técnicas consegue examinar mais estados que a outra.

Outra abordagem de busca bastante conhecida é a de **gerar e testar**. Essa abordagem é bastante simples e envolve a geração de cada nó no espaço de busca com o objetivo de verificar se ele é um nó objetivo ou não. Caso contrário, o procedimento segue para o próximo nó e a verificação se repete. Essa abordagem é a maneira mais simples de busca e é conhecida como *busca exaustiva*. Isso porque o método de busca não apresenta qualquer conhecimento adicional além de como percorrer a árvore e identificar nós folhas e nós objetivos, fazendo com que cada nó da árvore seja examinado até que seja possível encontrar um objetivo.

O método gerar e testar pode ser aplicado a diversos problemas, além de ser considerado um dos modos pelos quais normalmente as pessoas solucionam os problemas sem que existam informações adicionais sobre como obter determinada solução. Ainda, alguns autores em determinadas literaturas definem o método como uma busca cega, isto é, como consequência do modo como a árvore de busca é percorrida, sem utilizar qualquer informação sobre o espaço de busca.

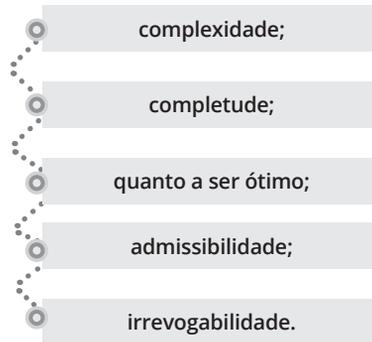
Com base nos exemplos citados, podemos notar que vários são os métodos de busca existentes, mas, em razão disso, é importante

Saiba mais

As abordagens *top-down* e *bottom-up* são usadas em diversas áreas e em diferentes sentidos. Em relação a IA essa abordagem também pode ser usada para conceitos de Web semântica e Ontologias, bastante aplicados em determinadas aplicações da Web. Para saber mais a respeito, leia o artigo *Introdução a Web Semântica*, publicado pelo Devmedia.

Disponível em: <https://www.devmedia.com.br/introducao-a-web-semantica/26181>. Acesso em: 6 jan. 2021.

sempre ter em mente que métodos diferentes trabalham de maneiras diferentes. Por esse motivo existem propriedades importantes que precisam ser respeitadas para que um método se torne útil. Coppin (2010) destaca algumas dessas as propriedades:



Ao se discutir um método de busca, faz-se necessário descrever o quão eficiente ele é em termos de tempo e espaço. Assim, a **complexidade**, no que diz respeito ao tempo de um método de busca, está relacionada à duração de tempos que este leva para encontrar um estado objetivo. Enquanto isso, a complexidade de espaço relaciona-se à quantidade de memória (recursos computacionais) que o método necessita para executar. Nesse sentido, entendemos que um método de busca que se caracteriza por ser ineficiente pode ter um desempenho razoável quando o problema é de teste, porém, para um problema do mundo real, pode ocorrer uma grande diferença entre o desempenho de dois métodos de busca: o mais rápido nem sempre encontrará a melhor solução para o problema. Já aquele que analisa todas as soluções existentes apresenta maior garantia de encontrar a melhor solução, porém torna-se ineficiente devido ao tempo gasto para a essa descoberta.

A propriedade de **completude** é considerada um método de busca completo, uma vez que tem como propósito encontrar um estado objetivo, caso exista. Completude é sempre uma característica desejável, pois ao utilizar um método de busca é importante que o sistema saiba que tal método encontrará uma solução. Já a propriedade **quanto a ser ótimo** ocorre quando o método de busca visa garantir que encontrará a melhor solução existente. Isto é, o método encontrará o caminho que tenha o menor número de passos necessários até o estado objetivo.

Em determinadas situações o termo *ótimo* é usado para descrever um algoritmo que encontra uma solução no menor tempo possível, situação essa que trata do conceito de **admissibilidade**. Assim, dizemos que o algoritmo é admissível caso ele garanta encontrar a melhor solução. Já os métodos de busca **irrevogáveis** são aqueles que encontram soluções subótimas para os problemas e, frequentemente, tendem a ser enganados por **ótimos locais**, mas que não apresentam soluções.

A B Glossário

ótimos locais são soluções que parecem localmente boas, porém são menos favoráveis quando comparadas com outras soluções em outros lugares do espaço de busca.

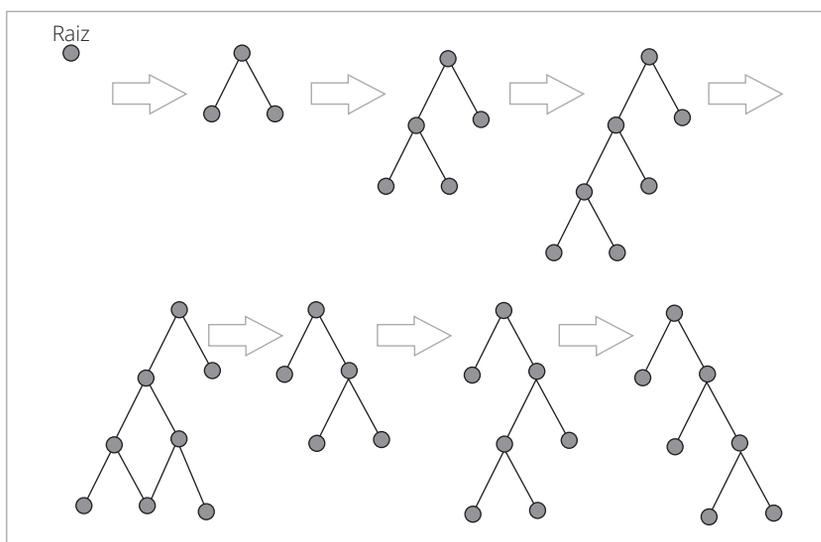
2.3 Busca em profundidade e busca em largura —

Vídeo



De acordo com Russell e Norvig (2010), a busca em profundidade se caracteriza por seguir determinado caminho até a sua maior profundidade antes de ir para o próximo. Um exemplo da busca em profundidade é ilustrado na Figura 4.

Figura 4
Busca em profundidade



Fonte: Adaptada de Coppin, 2010.

Analisando a árvore de busca construída na Figura 4, podemos notar que o início ocorre pelo nó raiz e, na sequência, o caminho é traçado pela esquerda e, posteriormente, para a direita. Nessa técnica, a busca em profundidade se caracteriza por descer pelo caminho mais à esquerda da árvore até encontrar o nó folha. Caso esse seja o estado objetivo, a busca é concluída e será retornado como resultado o sucesso.

? Curiosidade

A busca em largura pode ser usada em um jogo de labirinto em que é necessário encontrar uma determinada saída. Para isso, é atribuído pesos aos caminhos que possibilitam melhor direcionamento em relação ao objetivo.

No entanto, caso a folha não represente o estado objetivo, a busca retrocede ao que tenha um caminho ainda não explorado e repete os passos até que o estado objetivo seja encontrado. A busca em profundidade adota como método o retrocesso cronológico para que seja possível retornar à árvore de busca, uma vez que um caminho sem saída pode ser encontrado.

As aplicações que adotam esse método utilizam computadores para problemas que requerem a busca de um localizador de arquivo em um disco ou por motores de busca do tipo indexação na internet, por exemplo. Dessa forma, dizemos que a busca em profundidade é um exemplo de busca de força bruta ou busca exaustiva, conforme estudamos anteriormente aqui.

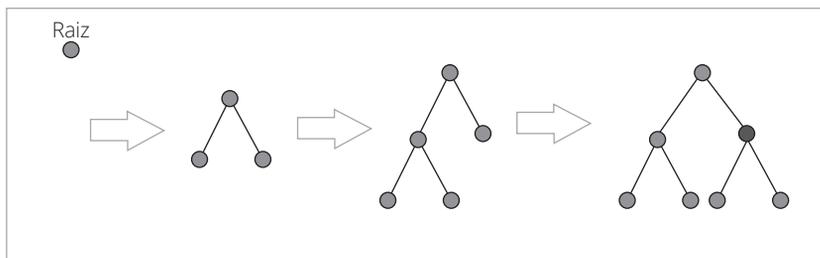
O método de busca em profundidade pode ser implementado por busca em árvore com a utilização de uma fila, conhecida como *LIFO* (*last in first out*), ou seja, o último que chega é o primeiro que sai. Essa busca apresenta requisitos de memória bastante modestos, uma vez que só precisa armazenar um caminho da raiz até um nó folha em conjunto com os nós irmãos restantes, não expandidos de cada caminho (RUSSELL; NORVIG, 2010).

Vale ressaltar que, para problemas em que há grandes descrições de estados, a técnica de busca por profundidade não é a mais indicada, uma vez que pode não apresentar bons resultados. Assim, sua desvantagem encontra-se na possibilidade de realizar uma escolha errada e ficar paralisada ao descer um caminho muito longo quando uma opção diferente levaria a uma solução próxima à raiz da árvore de busca.

Uma alternativa a essa técnica é a busca em largura, que tem como característica percorrer a árvore em largura, não em profundidade. Assim, nesse método a busca se inicia examinando todos os nós um nível abaixo do nó raiz. Caso o estado objetivo seja encontrado, relata-se sucesso. Caso contrário, a busca prossegue pela expansão de caminhos a partir de todos os nós do nível corrente em direção ao próximo nível.

Ao prosseguir pelos caminhos, continua-se examinando os nós em um determinado nível relatando sucesso quando um nó objetivo for encontrado, ou falha, se todos os nós tiverem sido examinados e um nó objetivo não for encontrado (Figura 5).

Figura 5
Busca em largura



A busca inicia pela raiz e segue para o nível inferior, começando a busca pelo nó da esquerda e, posteriormente, o nó que está à direita, que já é o nó referente a busca.

Fonte: Adaptada de Coppin, 2010.

Entendemos que a busca em largura é um método melhor de se utilizar em situações nas quais a árvore pode ter caminhos muitos profundos, principalmente se o nó objetivo estiver em uma parte mais rasa da árvore. A desvantagem dessa técnica é a de que a abordagem não funciona tão bem quando o fator de ramificação da árvore é muito alto.

Além disso, o uso desse tipo de busca também não é recomendado quando todos os caminhos que levam a um nó objetivo têm comprimento parecido. Para esses casos a busca em profundidade é mais indicada, uma vez que pode identificar um nó objetivo quando atingir o final do primeiro nó examinado.

O Quadro 1 ilustra a comparação das buscas em profundidade e em largura.

Quadro 1
Comparação entre busca em profundidade e largura

Cenário	Busca em profundidade	Busca em largura
Alguns caminhos são muito longos ou mesmo infinitos.	Funciona mal	Funciona bem
Todos os caminhos têm comprimentos parecidos.	Funciona bem	Funciona bem
Todos os caminhos têm comprimentos parecidos e todos os caminhos levam a um estado objetivo.	Funciona bem	Desperdício de tempo e memória
Alto fator de ramificação.	O desempenho depende de outros fatores	Funciona precariamente

Fonte: Coppin, 2010, p. 69.

Dessa maneira, podemos concluir que a busca em profundidade é geralmente mais fácil de implementar que a busca em largura e, normalmente, exige menos uso de memória, uma vez que precisa armazenar somente a informação sobre o caminho que está sendo explorado. Já a busca em largura precisa armazenar informações sobre todos os caminhos que atingem a profundidade corrente por onde passa.

Todavia, visando otimizar o processo de buscas e evitar o aumento excessivo de nós a serem visitados com a busca em largura, foi desenvolvida a busca bidirecional. Nessa busca cada nível abaixo do número de nós aumenta drasticamente. Além disso, a técnica visa usar duas buscas em largura, sendo que uma partiria do nó inicial e outra do nó meta; e o processo é interrompido apenas quando as duas se interceptam.

2.4 Busca e funções heurísticas

▶ Vídeo



Funções heurísticas, de acordo com Luger (2013), podem ser definidas como o estudo dos métodos e das regras de descoberta e invenção. Na busca por espaço de estados, heurísticas são formalizadas como regras para escolher ramos/nós em um espaço de estados que apresentam maior probabilidade de levar a uma solução aceitável para o problema em questão.

As técnicas de busca heurística usam algoritmos baseados em conhecimentos que podem ajudar e acelerar a solução do problema. Comumente nessa técnica os parâmetros são incertos e não garantem uma solução ótima, mas, na maioria das vezes, conseguem acelerar o processo, uma vez que as buscas heurísticas exploram as direções que consideram boas.

Vários autores consideram como heurísticas apenas as estratégias não numéricas. Quando são usados valores numéricos para a tomada de decisões, consideramos que a técnica é a de função de avaliação, ou função de custo, uma heurística.

Na função de avaliação cada nó a ser avaliado está associado a um valor que estima a distância entre o estado atual e a meta, servindo para a tomada de decisão.

De acordo com Luger (2013), em Inteligência Artificial os resolvedores de problemas usam heurísticas em soluções em que:

1. Um problema não pode ter uma solução exata/determinista, pois causa ambiguidades inerentes na formulação do problema ou nos dados disponíveis. Como exemplo para essa forma tem-se o diagnóstico médico, em que as heurísticas são usadas para escolher o melhor diagnóstico para o problema em tratamento.
2. Um problema pode ter uma solução exata, mas devido ao seu custo computacional torna-se inviável de resolvê-lo. Um exemplo para essa situação é o jogo de xadrez, em que o espaço de busca é exponencialmente explosivo a nível de recursos computacionais, uma vez que a cada jogada o número de combinações cresce.

Desse modo, entendemos que a heurística é baseada na experiência e na intuição, pois utilizam informações limitadas, como conhecimento sobre a situação atual ou as descrições dos estados abertos que estão na lista. Isto é, as heurísticas nem sempre podem prever o comportamento exato do espaço de estados mais à frente.

Assim, uma heurística pode levar um algoritmo a uma solução subótima, ou então, simplesmente a não encontrar solução. Na literatura existem várias implementações de busca heurística, e entre elas destacam-se a **subida da encosta** e a **programação dinâmica**.

A subida da encosta é uma das formas mais simples de implementar a busca heurística. As estratégias de subida de encosta expandem o estado atual da busca e avaliam os seus filhos, isto é, o “melhor” filho é selecionado para uma expansão futura; nem os seus irmãos nem os genitores são considerados¹ (Figura 6).

Para Artero (2009), a subida da encosta depende do conhecimento, o qual irá estimar um valor relativo à meta e se moverá na direção do nó que apresenta um valor melhor que a do nó atual. No entanto, essa busca apresenta alguns problemas quando todas as direções são iguais. A essa situação dá-se o nome de *platô*; ou, ainda, quando uma direção parece boa, mas em seguida verifica-se que não é, chama-se de *máximo local*.

Quando implementado, o algoritmo da subida da encosta não registra o histórico do processo de subida, impedindo o algoritmo de efetuar uma recuperação de falhas estratégicas.

1

Filhos são os nós que se encontram no nível abaixo do nó raiz de uma árvore. Irmãos, todos os nós disponíveis em determinado nível da árvore. Os nós que os ligam na camada anterior são seus genitores, enquanto folhas ou terminais são os nós que estão disponíveis no último nível de profundidade da árvore e não apresentam filhos.

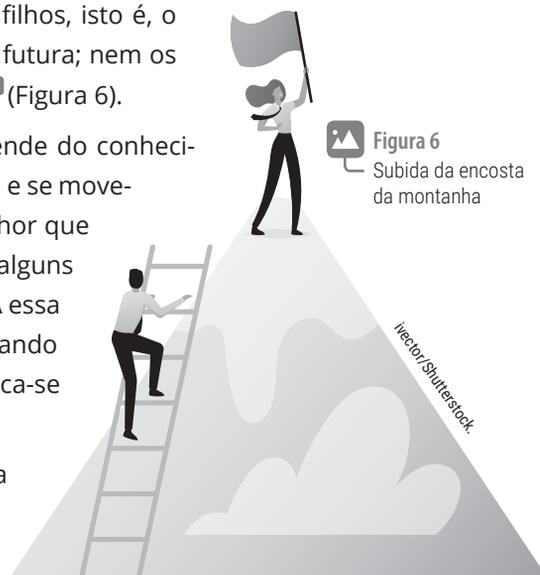
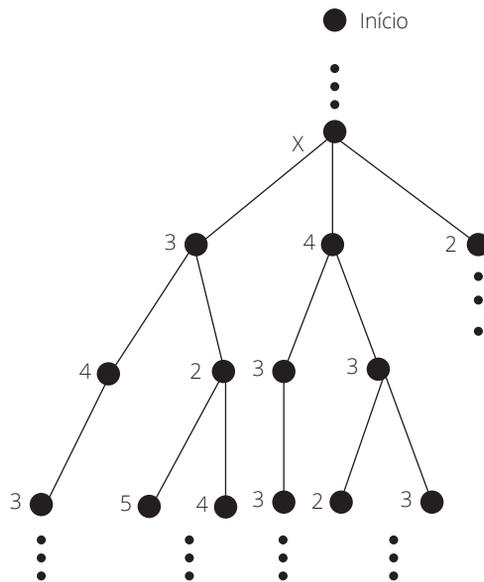


Figura 6
Subida da encosta da montanha

A subida da encosta da montanha leva em consideração a altura da sua posição corrente em cada direção: norte, sul, leste e oeste.

Nesse sentido, um dos problemas da implementação de estratégias da subida é a tendência de ficar preso nos máximos locais. Se elas alcançarem um estado que tenha uma avaliação melhor que qualquer um de seus filhos, o algoritmo tende a fracassar. Se esse estado não for um objetivo, mas apenas um máximo local, há também a possibilidade de o algoritmo não achar a melhor solução. A Figura 7 elucida o máximo local para a subida da encosta com antecipação de três níveis.

 **Figura 7**
Problema do máximo local para a subida da encosta



Fonte: Luger, 2013, p. 107.

No exemplo da Figura 7 temos a situação do máximo local, em que o espaço de busca explorado chega ao estado X e maximiza os valores de estado. As avaliações dos filhos, netos e bisnetos de X demonstram que a subida de encosta pode se confundir com a antecipação de vários níveis.

Já a programação dinâmica (PD), também chamada de *forward-backward* (para frente, para trás), criada por Richard Bellman, em 1956, utiliza como técnica a busca como memória restrita em problemas que abrangem vários subproblemas interagentes e inter-relacionados.

A PD comumente utiliza vários subproblemas já pesquisados e resolvidos dentro da solução do problema maior. Como exemplo, tem-se a solução das séries de Fibonacci. Para Luger (2013), o resultado é um

algoritmo normalmente utilizado para o casamento de sequências, verificações ortográficas e áreas relacionadas ao processamento de linguagem natural.

Outro fator importante é o de que esse tipo de programação requer uma estrutura de dados para registrar os subproblemas relacionados ao atestado que está sendo processado. Para que esse processamento ocorra, um arranjo é usado e o tamanho das dimensões é uma a mais se comparada ao tamanho de cada sequência.

Outras técnicas de busca que não utilizam listas para encontrar a solução de um problema são a têmica simulada e o algoritmo genético. Diferentemente da busca da subida da encosta, que sempre se desloca em direção ao topo e apresenta dificuldades na presença de máximos locais, a Têmica Simulada gera estados com determinada aleatoriedade. Assim, quando o novo estado é melhor que o estado atual, ele passa a ser adotado, porém, mesmo quando o novo estado não é melhor que o estado atual, ainda existe uma probabilidade de ser adotado.

Já a técnica de Algoritmo Genético pode ser compreendida como uma busca similar à técnica da subida da encosta. Contudo, durante o processo os novos estados a partir do cruzamento (combinação) dos estados atuais são considerados mais promissores do que aqueles encontrados na técnica de subida.

Importante

A justificativa para a programação dinâmica é o custo do tempo/espaço na computação.



CONSIDERAÇÕES FINAIS

Todo e qualquer problema do mundo real requer o uso de algum tipo de inteligência para ser solucionado. Isso também ocorre quando se desenvolve determinadas aplicações por meio de computadores para que seja possível executar tarefas específicas e direcionadas. Nesse sentido, as aplicações que utilizam técnicas de Inteligência Artificial geralmente adotam como resolução de problemas a técnica de busca.

A resolução de problemas por meio de busca é comumente utilizada na área de Inteligência Artificial e, também, em várias áreas, como jogos e indexação de pesquisa na web. A forma de representação mais usada para essas aplicações são por meio de grafos e árvores que permitem um conjunto de arcos/nós que estão interligados entre si.

Entretanto, é importante ter em mente que, sempre ao resolver problemas por meio de busca, algumas características precisam ser estabe-

lecidas para o sucesso do processo. Dentre essas características estão a formulação do problema, em que o objetivo a ser alcançado é estabelecido para que posteriormente as ações possam ser descritas. Lembrando que a busca pode ser resolvida por meio de grafos ou árvores e sempre o processo de busca inicia-se pelo estado inicial.

Verificamos também que os métodos de resolução de problemas baseado em busca são classificados em largura e em profundidade. No primeiro, a busca é iniciada pela raiz da árvore e a procura é expandida para os níveis abaixo, até que o estado objetivo seja encontrado. Já a busca em profundidade se diferencia por expandir os nós a partir da raiz, mas sempre escolhendo os estados de maior profundidade do lado de determinado nó.

Além das buscas em profundidade e em largura, outras buscas, como as funções heurísticas, são utilizadas. Essas são aplicadas quando não há uma solução ótima, ou seja, soluções subótimas podem ser usadas de modo satisfatório. Dentre os métodos de busca heurísticas existentes vimos que há a subida da encosta e a programação dinâmica.

Desse modo, podemos concluir que a resolução de problemas em espaço de busca é importante computacionalmente para especificar a melhor forma de como avaliar o desempenho de determinado problema, pois o custo computacional nem sempre se torna viável.



ATIVIDADES

1. Como a busca pode ser usada na resolução de problemas?
2. O que é busca em largura?
3. Cite uma técnica da busca por meio da função heurística.
4. Quais são as propriedades que um algoritmo de busca deve ter?



REFERÊNCIAS

- ARTERO, A. O. *Inteligência artificial: teórica e prática*. São Paulo: Livraria da Física, 2009.
- COPPIN, B. *Inteligência artificial*. Rio de Janeiro: LTC, 2010.
- LUGER, G. F. *Inteligência artificial*. 6 ed. São Paulo: Pearson Education do Brasil, 2013.
- RUSSELL, S.; NORVIG, P. *Inteligência artificial*. Rio de Janeiro: Campus, 2010.

3

Sistemas baseados em conhecimento

O conhecimento pode ser considerado uma informação armazenada por um indivíduo ou por uma máquina para interpretar, identificar, prever e responder adequadamente a questões do mundo real. Em razão disso, é possível compreender que a inteligência e o conhecimento sempre são usados de modo dependente, uma vez que a representação é importante para raciocinar e agregar novos conhecimentos, e não simplesmente para recuperá-los quando necessário.

No entanto, quando se trata de sistemas baseados em conhecimentos que são armazenados em máquinas é preciso que determinados modelos sejam adotados visando garantir a manipulação e a retenção no processamento das informações.

Nesse sentido, para que um conhecimento seja representado por meio de um computador é necessário que escolhas sejam feitas dentro de determinados modelos de representação, que tem como base conhecimentos já existentes, para que seja possível armazenar e inserir as informações corretamente.

Seja por meio de um computador ou de uma pessoa, a manipulação do conhecimento sempre exige uma representação, destacando que diferentes formas facilitam a manipulação por vários agentes (máquinas ou pessoas).

3.1 Fundamentos e tipos de aplicações

▶ Vídeo



Há anos a Inteligência Artificial (IA) vem se aperfeiçoando no que diz respeito à valorização do conhecimento específico de domínio, uma vez que o compreende como um dos pontos fundamentais na resolução de problemas complexos. Isso ocorre devido ao fato de a maioria dos problemas relacionados à IA não possibilitarem estratégias de resolução de problemas gerais mesmo quando heurísticas são adicionadas.

Os sistemas baseados em conhecimento (SBC) surgiram à medida que os avanços de hardware, software e ciência cognitiva foram evoluindo e foi por meio deles que se tornou possível a criação de técnicas e ferramentas baseadas em conhecimento.



Faber14/Shutterstock.

Para Rezende (2003), os sistemas baseados em conhecimento são programas de computadores que utilizam o conhecimento representado explicitamente para a resolução de determinados problemas. Para que isso ocorra, os programas manipulam o conhecimento e a informação de maneira inteligente para que problemas que requerem alto grau de conhecimento e especialização sejam resolvidos com efetividade.

Nesse sentido, vários autores descreveram em suas literaturas um sistema baseado em conhecimento sob duas perspectivas diferentes: a do conhecimento e a simbólica; sendo a primeira processada pelo homem e a segunda pela máquina. Traduzindo para linguagem computacional, compreendemos que o conhecimento se restringe ao que o sistema deve fazer, enquanto a base indica, em termos simbólicos, como o sistema irá proceder.

Para definir um sistema baseado em conhecimento (SBC), é necessário elencarmos algumas características que são atribuídas pela comunidade estudiosa de Inteligência Artificial. De acordo com Rezende (2003), as principais características são:

- questionar o usuário por meio de uma linguagem de fácil compreensão para reunir as informações necessárias à resolução do problema;
- criar uma linha de raciocínio baseada nas informações e no conhecimento para encontrar respostas e soluções satisfatórias;
- explicar o raciocínio usado, caso seja questionado, o sistema precisa ser capaz de interpretar o processo e apresentá-lo de modo compreensível para o usuário do sistema;
- aceitar os erros, uma vez que o sistema pode falhar, mas deve ser desenvolvido para apresentar um desempenho satisfatório.

Para o autor, outras propriedades de SBC podem ser definidas de modo mais operacional:

- todas as informações a respeito do problema precisam ser representadas na base de conhecimento do sistema de modo claro e preciso;
- a base de conhecimento deve ser usada por um agente que seja capaz de entendê-la e interpretá-la, repassando as informações corretamente;
- problemas resolvidos pelos SBC são aqueles em que não se reconhece como solução um problema determinístico, mas que garantem uma solução efetiva.

É possível afirmar que os SBC se diferenciam dos sistemas convencionais no que diz respeito à organização e à forma como incorporam o conhecimento, bem como na sua execução e na impressão que causam aos usuários com os quais interagem. Essas diferenças foram pontuadas e podem ser observadas de modo resumido no Quadro 1.



Quadro 1

Diferenças entre sistemas convencionais e sistemas baseados em conhecimento

Sistemas convencionais	Sistemas baseados em conhecimentos
Estrutura de dados.	Representação do conhecimento.
Dados e relações entre dados.	Conceitos, relações entre conceitos e regras.
Tipicamente usa algoritmos determinísticos.	Busca heurística.
Conhecimento embutido no código do programa.	Conhecimento representado explicitamente e separado do programa que o manipula e interpreta.
Explicação do raciocínio é difícil.	Podem e devem explicar seu raciocínio.

Fonte: Rezende, 2003, p. 18.

A organização dos dados em sistemas convencionais é feita por meio de uma estrutura de dados e está relacionada à arquitetura do computador. Já no SBC essa organização ocorre por meio da representação do conhecimento, que não leva em consideração as informações da máquina. Nos sistemas convencionais, as estruturas de dados apenas representam os dados e as relações existentes entre eles. Essa representação de conhecimento é feita explicitando os domínios dos problemas, suas relações e suas regras de dedução.

Outra diferença em relação aos sistemas convencionais é o uso de algoritmo para realização das funções. Os SBC adotam o método de resolução de problemas por espaço de busca. Destaca-se, ainda, que os sistemas convencionais, tanto no que diz respeito ao conhecimento quanto ao domínio e sobre o método de resolução de problema, são embutidos no código do programa, fazendo um processo oneroso para qualquer modificação ou entendimento. Por outro lado, em um SBC todo o conhecimento é representado de modo separado, ou seja, não está embutido no sistema.

Vale ressaltar que para a construção de um SBC é necessário a realização de um estudo de viabilidade da aplicação dessa tecnologia no domínio em questão. Para isso, devem ser considerados os seguintes fatores: custo, habilidade e disponibilidade do especialista e delimitações do domínio do conhecimento.

O crescimento e a aplicação de SBC têm sido notados em diversas áreas, como medicina, negócios, engenharia, computação, entre outros. Dentre as aplicações desenvolvidas podemos destacar as classes de aplicações que abrangem tarefas nas quais os SBC são utilizados para interpretação, classificação, monitoramento, planejamento e projeto.



Figura 1

Interpretação e monitoramento de dados



tsyhun/Shutterstock

3.2 Estrutura geral de um sistema baseado em conhecimento

▶ Vídeo

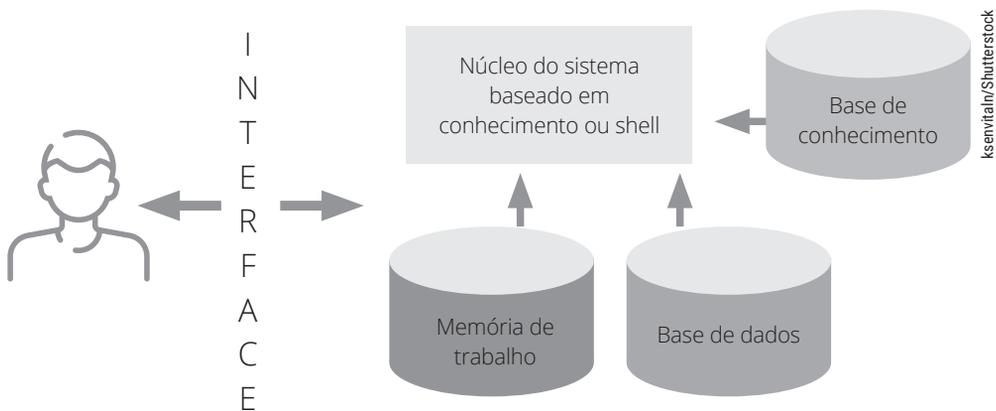


Os sistemas baseados em conhecimento podem apresentar estruturas diferentes, entretanto módulos responsáveis pelo armazenamento da base de conhecimento e pelo mecanismo de inferência devem estar presentes em qualquer estrutura, uma vez que são o núcleo de um SBC. A Figura 2 ilustra a estrutura geral desse sistema.



Figura 2

Estrutura de um sistema baseado em conhecimento



Fonte: Adaptado de Rezende, 2003.

O núcleo do SBC, também conhecido como *shell*, é quem desempenha as principais funções, dentre elas os mecanismos de inferência do sistema. Enquanto isso, a base de conhecimento é responsável por representar todo o conhecimento sobre determinado domínio, ou seja, pelo domínio ao qual determinada aplicação é construída. A construção dessa base é realizada abstraindo as informações do mundo real para um formalismo processável computacionalmente.

+ Saiba mais

Shell é usado para desenvolver vários scripts e acessar determinados serviços do sistema operacional. Para entender um pouco mais a seu respeito, leia o texto *Introdução ao Shell Script no Linux*.

Disponível em: <https://www.devmedia.com.br/introducao-ao-shell-script-no-linux/25778>. Acesso em: 13s jan. 2021.

As conclusões intermediárias de um processo de raciocínio e as interações com o usuário são armazenadas na memória de trabalho. Já a base de dados possibilita a interação com o sistema e com o banco de dados para obtenção e armazenamento de informações sempre que necessário. Por fim, temos a interface com o usuário, responsável pela obtenção da informação junto ao usuário, bem como a apresentação dos resultados e explicações sobre eles.

3.2.1 Núcleo do sistema baseado em conhecimento

Quando se constrói um sistema baseado em conhecimento, utilizamos um núcleo baseado em conhecimento (NSBC) para realizar várias atividades, como: o controle da interação com o usuário ou com o equipamento externo, o processamento do conhecimento usando linha de raciocínio e alguma justificativa das conclusões obtidas com base no raciocínio. O NSBC é composto de três submódulos que realizam funções específicas: módulo coletor de dados (MCD), motor de inferência (MI) e módulo de explicações (ME).

O MCD é responsável por realizar a interação com o usuário e obter as informações sobre o problema em questão. Essas informações são obtidas por meio de um conjunto de perguntas realizadas ao usuário. Quando o motor de inferência é ativado, o MCD realiza as perguntas necessárias ao usuário e as valida conforme as funções preestabelecidas. Já o MI destina-se ao desenvolvimento do raciocínio baseado nas informações obtidas pelo MCD e no conhecimento representando a base de conhecimento.

Nesse contexto, o MI processa a linguagem de representação utilizada na base gerando e percorrendo o espaço de busca sempre que necessário. Enquanto isso, o ME se responsabiliza pelas justificativas das conclusões obtidas e, principalmente, pelos motivos nos quais o SBC fez determinadas perguntas. No ME os principais tipos de explicações são: por que, como, o que acontece se, por que não.

3.2.2 Base de conhecimento

Toda a descrição necessária para a resolução do problema em questão na aplicação fica contida na base de conhecimento. Dentre as funções, se destacam as regras que descrevem as relações no domínio, bem como as heurísticas e/ou os métodos utilizados na resolução dos problemas.

Rezende (2003) define uma base de conhecimento (BC) como um conjunto de representações de ações e de acontecimentos do mundo, sendo cada representação compreendida como uma sentença que é expressa em uma linguagem específica, denominada por Russell e Norvig (2010) como *linguagem de representação do conhecimento*.

Essas linguagens são baseadas em várias técnicas de representação, como: regras de produção, regras semânticas, frames e lógica, que podem ser usadas isoladamente ou, então, combinadas de modo híbrido.

Vale ressaltar que nem sempre o conhecimento existente em uma BC é consistente e preciso. Por isso, ao utilizá-la é necessário analisar as evidências de cada conclusão e então escolher qual deve ser a resposta do SBC.

3.2.3 Memória de trabalho

Nessa área do SBC são registradas todas as respostas fornecidas pelos usuários durante suas interações com o sistema. Além disso, as conclusões e as sequências de passos de raciocínio também são registradas a fim de evitar que o usuário responda mais de uma vez a mesma questão.

Nesse sentido, Rezende (2003) destaca algumas vantagens e características decorrentes da memória de trabalho.

- Possibilita ao usuário conhecer toda sua trilha de raciocínio em relação às conclusões obtidas.
- Evita que as mesmas perguntas sejam encaminhadas ao usuário.
- Evita a sequência de raciocínio repetida para determinadas conclusões intermediárias.

Desse modo, é possível concluir que a memória de trabalho armazena condições iniciais, conclusões e decisões intermediárias, bem como as soluções finais que foram tomadas pelo usuário.

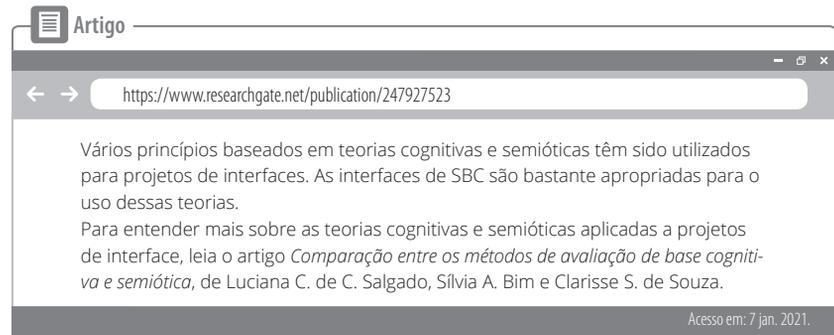
3.2.4 Interface

A interface em um SBC tem a responsabilidade de realizar a interação entre o usuário e o sistema com o objetivo de proporcionar a comunicação e a intermediação entre a representação interna do sistema e a representação central do usuário.

Para Russell e Norvig (2010), as linguagens de interfaces adotadas para a comunicação são mais abstratas se comparadas as utilizadas na repre-

sentação do conhecimento do SBC e, principalmente, bem mais restritas do que as linguagens usadas pelos usuários em suas atividades diárias.

Várias são as linguagens que podem ser utilizadas na interface de um SBC. Por esse motivo, podemos notar que há, em várias situações, uma combinação de diferentes linguagens, pois dessa forma torna-se mais fácil atingir diferentes tipos de públicos.



3.3 Representação do conhecimento

▶ Vídeo



A representação do conhecimento pode ser definida como algo que substitui o fenômeno real, de modo a permitir uma entidade a determinar as consequências de um ato pelo pensamento ao invés de sua utilização (REZENDE, 2003).

Em outras palavras, podemos dizer que a representação do conhecimento é uma maneira de codificar e estruturar o que se sabe a respeito de determinada aplicação. Para que isso ocorra, determinadas características precisam ser seguidas:

- ser compreensível ao ser humano;
- não levar em consideração os detalhes do processador que fará a interpretação do conhecimento;
- permitir sua utilização;
- ser generalizável.

É importante destacarmos que várias são as técnicas existentes de representação do conhecimento (RC). Quando ocorre a RC é necessário que critérios de avaliação também existam para identificar entre as técnicas existentes qual é a melhor a ser utilizada.

Dentre os critérios existentes os principais são: adequação lógica e conveniência notacional. A adequação lógica observa se o formalismo usado consegue expressar o conhecimento que se deseja representar; já a segunda verifica as convenções da linguagem de representação.

3.3.1 Representação lógica

A representação lógica tem sido muito utilizada, desde a década de 1970, em aplicações de Inteligência Artificial, principalmente em aplicações que objetivam derivação da lógica matemática.

A lógica matemática é uma linguagem formal que possui várias regras sintáticas de dedução, além de realizar inferências dedutivas exclusivamente com base no formato sintático de expressões da linguagem.

Uma dessas lógicas é o Prolog, linguagem que utiliza os princípios da lógica e tem seu método de inferência fundamentado no princípio da resolução, além de sua notação ser baseada nas cláusulas de Horn (LUGER, 2013).

Segundo Rezende (2003), a representação do conhecimento em Prolog envolve a descrição dos fatos a respeito dos blocos e a sua disposição, conforme ilustrado no exemplo:

Cor (a, azul).

Cor (b, verde).

Cor (c, branco).

Tamanho (a, grande).

Tamanho (b, médio).

Tamanho (c, pequeno).

Sobre (a, b).

Sobre (b, c).

Acima (X, Y) :- sobre (X, Y).

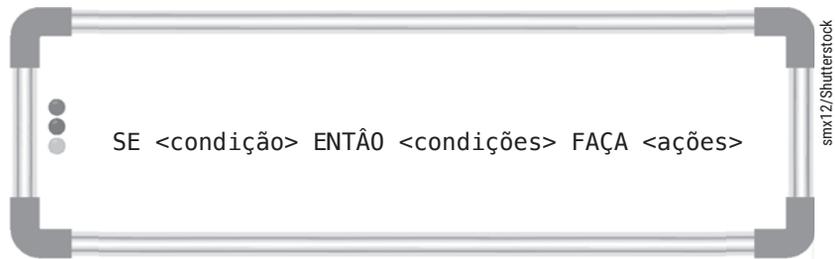
Acima (X, Y) :- sobre (Z, Y), acima (X, Z).

No exemplo, descrito por Rezende (2003), sobre e acima são definidos como predicados e permitem a relação entre os blocos. Enquanto, os predicados cor e tamanho são usados para descrever as características dos blocos.

3.3.2 Regras de Produção

Consideramos que os principais SBC foram construídos e baseados em regras de produção. Essas regras são inspiradas no processo de tomada de decisão humana e no modelo de regras do tipo: se condições então conclusões e ações.

Com base nessas premissas, entendemos que as regras podem expressar relacionamentos lógicos e equivalências de definições para simular o raciocínio humano. Analisando a notação de uma regra de produção, nota-se que a construção ocorre da seguinte forma:



Considera-se que a parte que se refere ao SE gera uma lista de condições a serem satisfeitas; a parte do ENTÃO, por sua vez, cria uma listagem de conclusões; e o FAÇA são as ações que devem ser executadas. De acordo com Coppin (2010), um sistema baseado em regras consiste em diversos componentes e, dentre eles, estão:

- uma base de dados de regras ou base de conhecimento;
- uma base de dados de fatos;
- um interpretador ou mecanismo de inferência.

Ainda segundo o autor, a base de conhecimento de um sistema fundamentado em regras consiste em um conjunto de diretrizes que representam o conhecimento que determinado sistema possui. A base de dados retrata as entradas do sistema que são utilizadas para obter conclusões, e o interpretador é aquele que controla o processo de obtenção das conclusões.

Um exemplo de sistema que utiliza regras de produção é o sistema especialista. Para Coppin (2010), um sistema especialista é um projeto que tem como objetivo o de modelar o comportamento de um especialista em determinada área, como medicina, engenharia ou geografia.

Nesse sentido, os sistemas especialistas baseados em regras são desenvolvidos para serem capazes de utilizar as mesmas regras que os especialistas utilizariam para chegar a determinadas conclusões, baseando-se em um conjunto de fatos apresentados ao sistema. Ou seja, um sistema especialista desenvolvido para diagnóstico médico, teria uma base de conhecimento composta por várias regras relacionadas a doenças, possibilitando ao sistema, quando questionado, apresentar uma solução considerando o conjunto de fatos estipulado.

3.3.3 Redes Semânticas

Compreende-se como rede semântica um grafo rotulado e direcionado, o qual é formado por um conjunto de nós que representam os objetos, bem como por um conjunto de arcos que representam as relações entre esses objetos.

Para Luger (2013), uma rede semântica representa o conhecimento por meio de um grafo, sendo os nós correspondentes aos fatos ou aos conceitos e os arcos como relações ou associações entre conceitos; tanto os nós quanto os arcos são rotulados.

Ainda para o autor, o termo *rede semântica* pode abranger uma família de representações baseadas em grafos, que são diferenciadas pelos nomes que podem ser usados para os nós e arcos, e nas inferências que podem ser oriundas dessas estruturas.

Já Rezende (2003), comenta em sua literatura que as redes semânticas são bem aceitas na representação do conhecimento devido à possibilidade de visualização gráfica das estruturas de conhecimento e suas relações.

3.3.4 Frames

Os frames, também conhecidos como *quadros*, são representações que possibilitam a organização do conhecimento em unidades mais complexas, refletindo a organização de objetos de um determinado domínio.

Para Luger (2013), cada frame pode ser compreendido como uma estrutura de dados bastante similar a uma estrutura de registro tradicional, na qual há informações relevantes sobre entidades estereotipadas. Diante disso, para o autor, as partições dos frames devem conter as seguintes informações:

1. informação sobre a identificação do quadro;
2. relação desse quadro com os demais;
3. descritores de requisitos para um quadro;
4. informação procedimental sobre o uso da estrutura descritiva;
5. informação padrão do quadro;
6. nova informação de instância.

Rezende (2003) afirma que uma das principais características do uso de frames para a representação do conhecimento é a herança de propriedades, na qual uma classe especializada pode herdar todas as propriedades da classe geral. Assim, as associações existentes entre os frames determinam a sua estrutura hierárquica, em que faz a ligação de um frame-pai a um frame-filho. Sendo que um frame-filho pode ser compreendido como uma especialização do frame-pai, por exemplo.

Diante do exposto, a hierarquia de frames permite que dados possam ser armazenados de modo abstrato e aninhado desde que respeite suas propriedades comuns herdadas por meio da hierarquia. Como consequência, evita-se a duplicidade de informações e torna-se mais fácil o código a nível de leitura e manutenção.

3.3.5 Processo de desenvolvimento de um sistema baseado em conhecimento

Para desenvolver um sistema baseado em conhecimento é necessário que seus processos tenham recursos disponíveis e, principalmente, que eles sejam organizados e gerenciados. Desse modo, o gerenciamento de projetos compreende o gerenciamento das atividades e o da configuração, além do gerenciamento dos recursos.

Os métodos convencionais de processo de desenvolvimento de software são conhecidos como cascata, espiral, prototipagem, RAD, incremental, entre outros. O método cascata, também denominado de *sequencial* ou *linear*, é o mais antigo e adota uma sequência de etapas que só podem ser definidas após o término de uma da outra, dificultando seu uso na construção de sistemas complexos.

Já o método incremental, também adota um conjunto de etapas que devem ser seguidas, porém essas são realizadas de modo iterativo e incremental, ou seja, uma primeira versão do sistema é criada considerando todas as

(Continua)

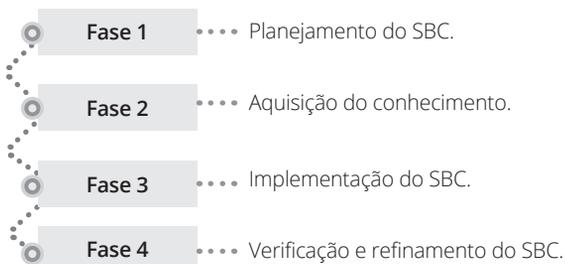
etapas concluídas e validadas, na sequência, um novo incremento é realizado, aumentando as funcionalidades do sistema e novamente passando por todas suas fases de desenvolvimento. O método incremental é bastante usado, pois permite maior interação com o cliente além de possibilitar o desenvolvimento em pequenos módulos e ir aos poucos aumentando a complexidade, porém os módulos finalizados já podem ser entregues e usados pelo cliente.

O RAD é uma adaptação do método cascata, porém com maior interação entre as fases, sendo que uma não precisa necessariamente esperar pelo término de outra para ser realizada. Normalmente esse método é usado quando se tem conhecimento bem específico do que se deseja desenvolver, pois o processo como um todo é realizado em um curto período. Dessa forma, é mais adequado que o método seja usado para atualização de softwares já existentes e não para a criação de algo novo.

O modelo espiral segue a mesma linha do modelo incremental, porém em cada uma de suas sucessivas sequências de etapas, não entrega um modelo pronto ao cliente, e sim permite que o software/programa seja implementado em várias iterações.

O método prototipagem é baseado na construção de uma interface para auxiliar no levantamento de requisitos junto aos clientes, quando esses não tem total domínio do que precisam e das necessidades do sistema. Nesse sentido a interface serve para validação junto com o cliente das funcionalidades do software; após essa definição o protótipo deve ser descartado e iniciado do zero de maneira planejada e seguindo cada uma de suas respectivas etapas.

Rezende (2003) afirma que são vários os estágios de um processo de desenvolvimento de um SBC. Dentre esses estágios destacam-se o estudo de possibilidades, a prototipação rápida, o refinamento do sistema, o teste preliminar do sistema, a qualidade comercial do sistema, a manutenção e a evolução. Com base nisso, pode-se utilizar vários modelos de processo de desenvolvimento de software convencionais, no entanto é necessário que determinadas etapas sejam seguidas.



A primeira fase destina-se a descrever o domínio de conhecimento em relação aos conceitos já relacionados. Já a segunda fase, de aquisição do conhecimento, é o momento em que tudo o que foi planejado na fase 1 é executado, ou seja, é uma fase que se refere à identificação, à conceitualização e à formalização do conhecimento.

Na terceira fase, a implementação do SBC é representada formalmente e adota-se a estrutura de representação do conhecimento escolhida na primeira fase. Por fim, na quarta fase – validação e refinamento do SBC – envolve-se a validação e a verificação do sistema, que podem ser considerados um processo contínuo, isto é, sendo necessário que o sistema funcione corretamente sempre.

Desse modo, compreendemos que dentre as etapas do desenvolvimento de um sistema baseado em conhecimento, a aquisição de conhecimento é a fase mais complexa e difícil, pois é nela que é construída a base de conhecimento que serve como core/núcleo ao funcionamento do SBC.

3.4 Aquisição do conhecimento

▶ Vídeo



Com o avanço tecnológico as informações têm crescido exponencialmente e a forma de as manipular tem se tornado um grande desafio. Vários sistemas inteligentes estão sendo desenvolvidos para que seja possível criar bases de conhecimentos capazes de compreender determinado domínio.

Diante do exposto, várias empresas, em busca do diferencial competitivo, sistematizam seu *know-how* em um sistema inteligente (SI) com o objetivo de tornar o processo mais ágil e efetivo para a tomada de decisão, para responder ao mercado rapidamente e com alto padrão de qualidade.

Pode-se dizer que um SBC é um tipo de SI com foco no processo simbólico, capaz de gerar explicações sobre a linha de raciocínio por trás de cada uma de suas decisões. Em razão disso, é possível perceber que o processo de aquisição do conhecimento é considerado um gargalo na construção dos sistemas baseados em conhecimento, uma vez que necessitam de conhecimentos mais explícitos que os sistemas convencionais.

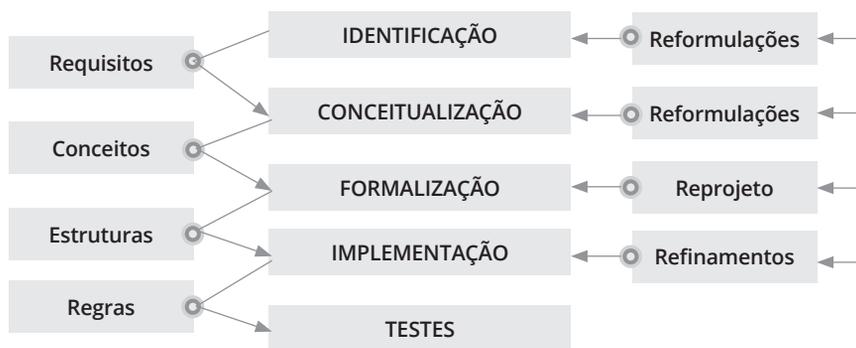
Figura 3
Aquisição do Conhecimento



Rezende (2003) define **aquisição do conhecimento** como a transferência e a transformação do conhecimento especializado em um potencial para a resolução de problemas de alguma fonte para algum programa. Ainda, pode-se definir como um processo de modelagem de problemas e soluções pertinentes a tarefas em um determinado domínio (LUGER, 2013).

Para compreender melhor o processo de aquisição do conhecimento, são descritas as fases que normalmente o compõem do seguinte modo: identificação; conceitualização; formalização; implementação e testes. O modo como essas fases ocorrem, além de a que estão ligadas, podem ser vistas na Figura 4.

 **Figura 4**
Fases do processo de aquisição do conhecimento



Fonte: Adaptada de Rezende, 2003.

A fase de identificação é bastante similar à fase de análise da engenharia de software, na qual o projetista procura entender os requisitos do problema que deve resolver, os dados sobre os quais o sistema deve operar, os critérios para sua avaliação e a forma como o problema deve ser resolvido.

Por sua vez, na fase de conceitualização, o engenheiro do conhecimento torna-se responsável por formular conceitos importantes do problema, que são obtidos por meio de uma revisão bibliográfica e de uma entrevista estruturada, além das relações entre esses conceitos. As relações e os conceitos são caracterizados em termo de domínio como causa-efeito, espaço-tempo e parte-todo.

A fase de formalização concentra-se no processo de modelagem computacional do problema e se preocupa com a natureza do espaço de busca e em como a busca nesse espaço deve ser conduzida. Nessa fase, o engenheiro precisa definir o modelo da tarefa a ser adaptado;

escolher a linguagem de representação para modelar o sistema; definir qual será o espaço de busca e o espaço de soluções do problema; definir os métodos de busca a serem usados; além de identificar possíveis limitações do SBC.

Na fase de implementação a linguagem de programação é selecionada e, então, o SBC é codificado. Como resultado da fase tem-se um protótipo do sistema que é validado pelo especialista. Já na fase de teste verifica-se a conformidade do que foi construído com o que foi especificado; desse modo, o engenheiro do conhecimento e o especialista avaliam o desempenho do SBC por meio de casos de teste.

Como qualquer sistema de software, um SBC também precisa estar atento e, principalmente, atender a novas demandas. Sendo que essas podem variar desde novas funcionalidades a serem cumpridas, a novas formas de realizar uma antiga atividade.

Diante disso, é importante na construção do processo de aquisição do conhecimento que as bases criadas sejam capazes de se adaptar a novas modificações e que tenham maior flexibilidade na forma de suas representações.



CONSIDERAÇÕES FINAIS

O conhecimento é um conceito abstrato que determina um comportamento padrão reconhecível. A representação da informação pode ser usada como solução inteligente de problemas que oferecem soluções difíceis e desafios importantes na área de Inteligência Artificial.

Diante disso, nota-se que os sistemas baseados em conhecimento têm sido utilizados nas últimas décadas quando a formulação do problema a ser resolvido computacionalmente é complexa e existe uma grande quantidade de conhecimento específico sobre o domínio de como resolvê-lo. Esses sistemas são aplicados não somente nas áreas acadêmicas, mas também tem sido úteis em áreas comerciais, como medicina, engenharia, geologia etc.

Um SBC é composto de uma estrutura que contempla o núcleo do sistema, a base de conhecimento, a memória de trabalho, uma base de dados e uma interface para interação com o usuário. Sempre ao representar um conhecimento, esse pode ser lógico ou por regras de produção, frames, entre outros. No caso da representação lógica é adotado uma linguagem formal e matemática, já na parte de regras de produção as regras possibilitam derivar determinados diagnósticos. Os frames ou

quadros são agrupamentos de conhecimento que se referem a mesma coisa e são relevantes a ela.

Como aquisição do conhecimento entendemos o processo de interação entre um engenheiro do conhecimento, responsável por construir um SBC, e o especialista, ou seja, a pessoa que detém o conhecimento.



ATIVIDADES

1. O que é um sistema baseado em conhecimento?
2. Defina o sistema especialista.
3. Quais são as etapas do núcleo do sistema da base do conhecimento (NSBC)?
4. O que é a aquisição do conhecimento e quais são suas fases?



REFERÊNCIAS

COPPIN, B. *Inteligência artificial*. Rio de Janeiro: LTC, 2010.

LUGER, G. F. *Inteligência artificial*. 6 ed., São Paulo: Pearson Education do Brasil, 2013.

REZENDE, S. O. *Sistemas Inteligentes: fundamentos e aplicações*. São Paulo: Manole, 2003.

RUSSELL, S.; NORVIG, P. *Inteligência artificial*. Rio de Janeiro: Campus, 2010.

4

Aprendizado de máquina

A Inteligência Artificial é uma grande área que pode ser aplicada a diferentes frentes, além disso há uma variedade de subáreas, das quais o aprendizado de máquina (AM) encontra-se localizado. O AM é uma subárea da IA que apresenta como principal objetivo criar soluções computacionais capazes de aprender automaticamente.

De modo mais simplificado, o AM pode ser resumido como um programa de computador que consegue individualmente tomar decisões considerando todo o aprendizado adquirido na resolução de problemas similares e já resolvidos. No entanto, para que esse programa possa tomar decisões e/ou aprender com base em experiências anteriores é necessário que um conjunto de características sempre sejam respeitadas como, por exemplo, classificação quanto à uma determinada linguagem de descrição, paradigma ou tipo de aprendizado usado.

Vale lembrar que, quando construímos algum problema que adote o conceito estabelecido pelo AM de criar tomada de decisões de maneira automática, ainda é necessário observar que, nessa subárea, existem vários tipos de AM, bem como algoritmos de inteligência que podem ser adotados em determinado problema. Em outras palavras, para a resolução de um determinado problema usando algoritmos de AM, primeiro, é importante compreendermos as características e limitações do algoritmo para somente então aplicá-lo, pois determinados algoritmos apresentam melhor desempenho quando usados para dadas aplicações do que para outras.

4.1 Conceitos e paradigmas de aprendizado

▶ Vídeo



A capacidade de aprender é necessária e deve fazer parte de qualquer sistema que possua inteligência de modo geral. Dessa forma, como aprendizado podemos considerar as mudanças realizadas em

um determinado sistema de modo a melhorar seu desempenho na segunda vez que ele for repetir a mesma tarefa. Além disso, é importante destacar que o aprendizado não deve apenas melhorar o desempenho de uma tarefa repetida, mas precisa melhorar o desempenho em tarefas semelhantes no determinado domínio aprendido.

Vários são os paradigmas de aprendizado de máquina (AM), dentre eles destacam-se o aprendizado simbólico, conexionista, estatístico, baseado em exemplos e evolutivo (REZENDE, 2003).

O **aprendizado simbólico** se caracteriza por aprender por meio da construção de representações simbólicas e pela análise de exemplos e contraexemplos desses conceitos. Normalmente, as representações são realizadas na forma de uma expressão lógica, árvores de decisão, redes semânticas, regras etc.

O **aprendizado conexionista** faz uso de redes neurais artificiais, que adotam um modelo matemático simplificado e inspirado no modelo biológico do sistema nervoso humano. Sua representação envolve unidades interconectadas assim como os neurônios naturais do cérebro humano. Por outro lado, o **aprendizado estatístico** visa adotar modelos estatísticos com o objetivo de encontrar boa aproximação dos conceitos induzidos. Dentre os métodos estatísticos existentes, o aprendizado Bayesiano é um dos mais usados, pois utiliza um modelo probabilístico baseado no conhecimento prévio de determinado problema, combinando com exemplos de treinamentos que possam identificar a probabilidade final de uma respectiva hipótese.

Os sistemas que precisam manter os exemplos na memória, para que então possam classificar novos exemplos, são conhecidos por fazerem uso de um **aprendizado baseado em exemplos**. Em outras palavras, uma forma de lembrar um determinado exemplo é por meio da similaridade cuja classe é conhecida. É assumir que o novo exemplo terá a mesma classe, ou seja, sistemas baseados em exemplos classificam novos exemplos nunca vistos, considerando exemplos similares já conhecidos.

Por fim, o **aprendizado evolutivo** é derivado de um modelo biológico de aprendizado, ou seja, um classificador evolutivo é composto de uma população de elementos de classificação que competem para fazer a predição.

O Quadro 1 apresenta um resumo dos tipos de aprendizados existentes, possibilitando maior entendimento em relação às formas de representação.

 **Quadro 1**
Tipos de aprendizado

Aprendizado	Descrição	Representação
Simbólico	Adota a construção de exemplos e contraexemplos no processo de aprendizado.	Regras de decisão. Árvores de decisão. Redes semânticas.
Conexionista	Construção de um modelo matemático que imita o funcionamento de uma rede neural natural.	Camadas de entradas, intermediários e de saída para os neurônios da RN.
Estatístico	Todo aprendizado é construído por modelos estatísticos e/ou probabilístico para conseguir identificar/classificar determinada hipótese.	RN.
Baseado em exemplos	Todos os exemplos e aprendizados precisam ser mantidos na memória para que possam ser usados na identificação de um novo exemplo.	Regras. K vizinhos mais próximos. Raciocínio baseado em casos.
Evolutivo	Caracterizado por representar o modelo biológico do aprendizado.	Amostra de população.

Fonte: Elaborado pela autora.

Diante do exposto, podemos concluir que independentemente do paradigma de aprendizado de máquina usado, é necessário que se entenda quais são as principais características e como é feita a sua representação. Pois, para determinados problemas, por exemplo, a abordagem baseada em exemplos não é tão robusta e produtiva em relação ao custo computacional do que o aprendizado por computação evolutiva.

4.1.1 Conceitos e definições

1

Pode ser definido como suposições utilizadas pelos diversos algoritmos na generalização dos dados de treinamento.

Para trabalhar com aprendizado de máquina, é necessário ter entendimento de determinados conceitos, como: exemplo, atributo, classe, conjunto de exemplos, classificador ou hipótese, conhecimento do domínio, bias¹, modo de aprendizado, erro e precisão, espaço de descrição, poda, completude e consistência, entre outros. Pois, somente entendendo cada um desses conceitos, torna-se mais fácil a tarefa de

saber qual modelo de aprendizado de máquina usar para resolver as demandas de determinada aplicação.

Como exemplos, que também podem ser chamados de *dado*, *registro* ou *caso*, dependendo do autor adotado, refere-se a uma tupla² de valores de atributos. Ou seja, um paciente pode ser um exemplo e esse paciente possui uma tupla de valores, como nome e número da carteira do plano de saúde.

Por outro lado, os atributos descrevem uma determinada característica e podem ser de dois tipos: nominal, não existe uma ordem entre os valores, exemplo, atributo cor – vermelho, verde, azul amarelo –, isto é, a ordem que os valores aparecem não interfere na característica do atributo; e contínuo, que necessita que a ordem seja respeitada em relação aos seus valores.

No entanto, independentemente do tipo de atributo, geralmente existe um símbolo usado que se chama *desconhecido* e que representa a ausência de um valor para aquele atributo.

Em relação à classe, quando se adota um aprendizado caracterizado como supervisionado, ela precisa ser definida e é usada como um rótulo, ou seja, para os determinados exemplos de entrada deseja-se saber a qual classe/rótulo eles pertencem.

Já o conjunto de exemplos é composto por exemplos que contêm os valores dos atributos e as classes a eles associadas. O Quadro 2 demonstra como é o formato padrão normalmente adotado para um conjunto de exemplos segundo Rezende (2003).



Quadro 2

Conjunto de exemplos no formato atributo-valor

X1	X2	...	Xm	Y
x11	x12	...	x1m	y1
x21	x22	...	x2m	y2
...
xn1	xn2	...	xnm	yn

Fonte: Adaptado de Rezende, 2003.

No exemplo da tabela, podemos observar que uma tupla T é composta por x_1, x_2, \dots, x_n e tem como resultado o y_1 , isto é, matematicamente a notação para a tupla seria $T = (x_i, y_i)$, sendo que o x_i seria um vetor contendo todos os valores de cada uma das colunas da primeira linha do quadro.

2

Tuplas são formadas por dois valores, sendo que fica subentendido que um desses valores pertence ou não ao segundo valor que chamamos de classe.



Atenção

Quando se trata de um atributo desconhecido isso não é igual a um atribuído ter valor zero (bastante usado em números). Da mesma forma que um atributo pode apresentar um valor que não se aplica. Esse último pode ser usado, por exemplo, para representar o número de gestações, caso a pessoa não tenha tido nenhuma gestação esse atributo pode ser usado.

Além das tuplas, outros conceitos também são importantes de serem conhecidos quando se trata da definição do conjunto de treinamento. Esse conjunto de treinamento pode ser entendido como a saída de um classificador, permitindo que a cada exemplo inserido como entrada um indutor (classificador) possa especificar a qual classe o exemplo pertence.

Conhecimento de domínio inclui informação a respeito dos valores válidos dos atributos, um critério de preferência para a escolha entre possíveis atributos ou mesmo hipóteses. Em relação ao uso do *bias* em aprendizado de máquina, é usado sempre que determinada hipótese tem preferência sobre outra.

O espaço de descrição corresponde a uma coordenada no espaço m -dimensional, em que estão a quantidade de exemplos estudados nesse espaço. Quando desejamos avaliar o desempenho de um determinado modelo ou algoritmo, usamos o erro e a precisão: o erro irá informar a taxa de erro do classificador. Em geral, o erro calculado sobre o conjunto de exemplos de treinamento é menor que o erro calculado sobre o conjunto de exemplos de teste.

No caso da distribuição de classes, se existir um conjunto de exemplos denominado por T , podemos calcular sua distribuição de classes no caso de classificação, por exemplo.

De posse desses conceitos passamos, então, a compreender os tipos de aprendizado: simbólico, conexionista e probabilístico.

Leitura

Com o objetivo de compreender mais sobre os fundamentos de aprendizado de máquina, leia o Capítulo 4 do livro de Rezende (2003).

Disponível em: <http://dcm.ffclrp.usp.br/~augusto/publications/2003-sistemas-inteligentes-cap4.pdf>. Acesso em: 3 fev. 2021.

4.2 Aprendizado simbólico

Vídeo



Todo sistema desenvolvido e que, em seus objetivos, visa possuir algum tipo de aprendizado pode ser classificado como caixa preta e orientado a conhecimento (REZENDE, 2003). Quando nos referimos a sistemas de aprendizados orientados a conhecimento, esses apresentam como principal interesse obter descrições simbólicas de fácil compreensão e são usados por meio de modelos mentais.

As descrições simbólicas podem, por exemplo, apresentar em seus componentes regras ou nós em uma árvore de decisão, além de possuírem expressões contendo menos de cinco condições em uma conjunção ou poucas condições em uma disjunção; no máximo um nível e parênteses, no máximo uma implicação etc.

Nesse sentido, quando nos referimos a aprendizado de máquina simbólico, este pode apresentar vários tipos de linguagens de representação entre as quais se destacam: lógica proposicional, lógica de atributos, lógica de primeira ordem e lógica de segunda ordem.

Em cálculo proposicional, os itens a serem representados são descritos por conjunções, disjunções e negações de constantes booleanas que representam os atributos individuais, conforme exemplo, descrito por Rezende (2003, p. 117):

“Fêmea \wedge adulta \rightarrow pode_ter_filhos”.

No entanto, analisando a linguagem e a maneira como ela é representada, percebemos que possui um baixo poder descritivo, não sendo capaz de descrever objetos sobre os quais as relações são observadas.

Enquanto a lógica de atributos é responsável por representar itens, vários indutores que adotam uma linguagem baseada em atributos, ou seja, a lógica de atributos é equivalente ao cálculo proposicional, porém sempre emprega uma notação mais poderosa e flexível. Notamos que grande parte dos indutores adota a lógica de atributos para descrever exemplos e hipóteses, dessa forma muitos objetos estruturados não podem ser representados devido à baixa capacidade de expressão da lógica de atributos.

A lógica de primeira ordem surgiu para suprir as limitações encontradas na lógica de atributos e permite descrever e raciocinar sobre objetos e predicados que possam especificar propriedades de objetos ou relacionamentos entre objetos de determinado domínio D. As cláusulas de Horn³ são consideradas subconjuntos que são muito utilizados na lógica de primeira ordem.

Normalmente as cláusulas de Horn são usadas e representadas pela linguagem PROLOG, que por sua vez é composta por regras, predicados e fatos. Como extensão da lógica de primeira ordem tem-se a lógica de segunda ordem, que possibilita que os predicados sejam considerados como variáveis, conforme pode ser visto no exemplo apresentado por Rezende (2003, p. 119):

$P_1(X, Y) :- P_2(X), P_3(X, Z), P_4(Y, Z).$

Onde P_1, P_2, P_3, P_4 são as variáveis que representam os predicados e X, Y, Z são as variáveis que representam os objetos.

Considerando as explicações acima, além das linguagens de representação, o aprendizado simbólico também se utiliza de outras variantes e técnicas chamadas de *indução de árvores de decisão*, *indução de regras ordenadas*, entre outras.

3

Cláusula de Horn é uma regra cuja cabeça contém um único predicado e um corpo com zero, um ou mais predicados.

4.2.1 Indução de árvores de decisão

A indução de árvores de decisão é usada em grande parte de aplicações que adotam como forma de resolução de problemas o aprendizado simbólico. Nesse sentido, usamos algoritmos que induzem árvores de decisão.

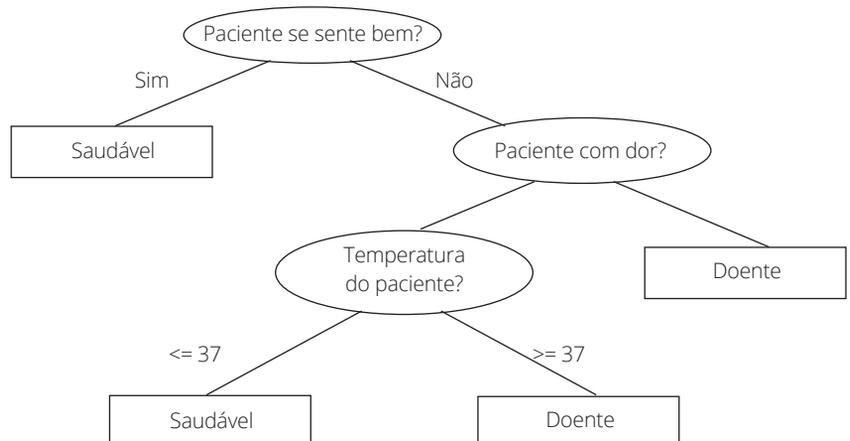
Para Rezende (2003), uma árvore de decisão (AD) é composta de uma estrutura de dados definida recursivamente, contendo:

- a. Um nó folha que corresponde a uma classe.
- b. Um nó de decisão que contém um teste sobre algum atributo, isto é, para cada resultado do teste existe uma aresta para uma subárvore. E cada subárvore possui a mesma estrutura de uma árvore.

Um exemplo de árvore de decisão pode ser visualizado na imagem da Figura 1, em que as elipses representam um teste em um determinado atributo para um conjunto de dados de pacientes. E cada retângulo é uma classe que consiste no diagnóstico desse paciente. Assim, para entendermos melhor o funcionamento de uma árvore de decisão e, principalmente, possibilitar o diagnóstico de um determinado paciente, é necessário que o processo se inicie pela raiz da árvore, seguindo cada teste até que uma folha seja alcançada.

Outra forma de representação de uma árvore de decisão é por meio de regras. Cada regra tem seu início na raiz da árvore e caminha até uma de suas folhas, como mostra a Figura 2, em que é possível entender a leitura referente a imagem da Figura 1.

 **Figura 1**
Árvore de decisão para diagnóstico de paciente



Fonte: Adaptado de Rezende, 2003, p. 120.



Figura 2

Leitura da árvore de decisão (Figura 1) por meio de regras.

```
Se paciente se sente bem = sim então
    Classe = saudável

Senão
    Se paciente tem dor = não então
        se temperatura do paciente <= 37 então
            classe = saudável
        senão {temperatura do paciente >= 37}
            classe = doente
    fimse
senão {paciente tem dor = sim}
    classe = doente
fimse
fimse
```

Fonte: Adaptado de Rezende, 2003, p. 120.

Para a construção de uma árvore de decisão baseando-se em determinado conjunto de treinamento T , devemos seguir os seguintes passos (REZENDE, 2003, p. 121):

- 1 O conjunto de treinamento T contém um ou mais exemplos, sendo que esses devem ser pertencentes à mesma classe.
- 2 T não contém exemplos. Isto é, a árvore é uma folha, mas a classe associada à folha deve ser determinada a partir da informação além de T .
- 3 T contém exemplos que pertencem a diversas classes. Assim, a ideia é refinar T em um subconjunto de exemplos que são conjuntos de exemplos pertencentes a uma única classe.
- 4 Os primeiros 3 passos são aplicados recursivamente para cada subconjunto de exemplos de treinamento, de maneira que em cada nó as arestas levem para as subárvores construídas a partir do subconjunto de exemplos.
- 5 Após construção da AD, pode ser realizada para melhorar a capacidade de generalização da AD.



Vídeo

O vídeo *Árvores de Decisão - Exemplo completo com construção e cálculo*, do canal Aprendendo Gestão, é um ótimo aliado para que você possa entender melhor como é o processo de construção de uma árvore de decisão.

Disponível em: <https://www.youtube.com/watch?v=PCTEn3hWDxA>. Acesso em: 3 fev. 2021.

Outra questão que vale a pena ressaltar ao construir uma árvore de decisão está relacionada à escolha do melhor atributo a ser usado, ou seja, a chave de sucesso de um algoritmo está diretamente relacionada ao atributo escolhido, que possibilita a participação de um conjunto de exemplos em cada uma das iterações. Rezende (2003) descreve algumas possibilidades que devem ser consideradas na escolha do atributo, a saber:

- Aleatoriedade: possibilidade de selecionar qualquer atributo.
- Menos valores: selecionar o atributo que contém o menor número de valores possíveis.
- Mais valores: selecionar o atributo que contém o maior número de valores possíveis.
- Ganho máximo: seleciona o atributo que apresenta o maior ganho de informação possível em relação ao tamanho da subárvore.
- Índice Gini e razão de ganho.

Sempre após a criação de uma árvore de decisão também é possível que o classificador induzido seja muito específico para o conjunto de treinamento. Isso geralmente acontece quando o classificador faz um super ajuste dos dados de treinamento. Como forma de solucionar esse problema, alguns indutores acabam adotando a poda de uma AD ao seu final.

O processo de poda faz com que o número de nós (testes internos) seja reduzido, de modo a reduzir a complexidade da árvore e melhorando o desempenho do que a árvore original.

4.2.2 Indução de regras ordenadas

A indução de regras ordenadas permite que a divisão do conjunto de treinamento seja dividida de forma direta, diferentemente da forma recursiva usada pelas AD.

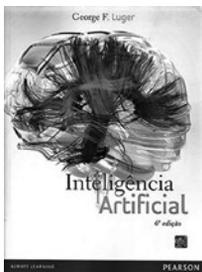
Dessa forma, entendemos que essa indução trabalha de forma iterativa, e cada regra cobre um subconjunto de exemplos que pertencem a uma classe específica.

Para classificar novos exemplos por meio de regras ordenadas o classificador tenta classificar cada regra em ordem até encontrar uma, onde as condições sejam satisfeitas pelo novo exemplo.

Leitura

Para entender sobre o índice Gini e razão de ganho de informação leia o post *Árvores de decisão*. Disponível em: <https://medium.com/machine-learning-beyond-deep-learning/%C3%A1rvores-de-decis%C3%A3o-3f52f6420b69>. Acesso em: 3 fev. 2021.

Livro



O livro *Inteligência Artificial* apresenta elementos que auxiliam no processo sobre o funcionamento de um aprendizado de máquina simbólico. O livro como um todo apresenta ótimos recursos e exemplos, contudo indicamos em específico a leitura do Capítulo 10.

LUGER, G. F. São Paulo: Pearson Education do Brasil, 2013.

As regras ordenadas podem ser visualizadas como uma árvore binária degenerada, pois, para classificar cada novo exemplo, cada uma das regras é testada até que uma dispare. Podemos dizer que esse processo é bastante semelhante e equivalente ao comando *if-then-else*, sendo que o espaço de descrição pode ser considerado como particionado em regiões que não se sobrepõem, assim como ocorre para árvores de decisão (REZENDE, 2003).

4.3 Aprendizado conexionista

▶ Vídeo



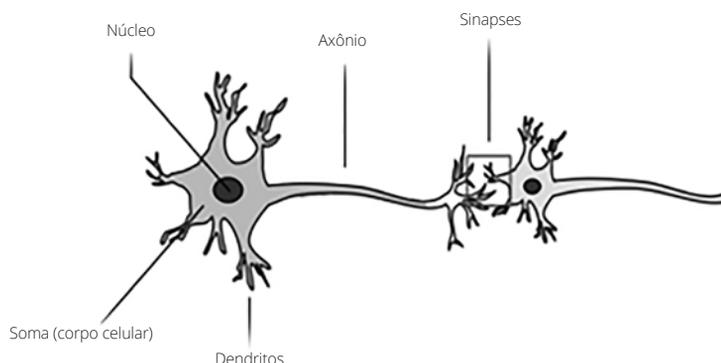
Para entender os conceitos envolvidos no aprendizado de máquina, inicialmente é preciso conhecer o funcionamento dos neurônios biológicos contidos no cérebro humano. Sabemos que um cérebro humano possui mais de dez bilhões de neurônios conectados a outros neurônios, sendo essas conexões chamadas de *sinapses*.

Dessa forma, podemos dizer que os neurônios são compostos por uma soma, contendo o corpo do neurônio, um axônio e vários dendritos, conforme ilustrado na Figura 3.



Figura 3

Neurônio no cérebro humano



Fonte: Adaptado de Coppin, 2010,.

Conforme ilustrado na figura anterior, as sinapses ou saídas dos axônios realizam a ligação entre os neurônios. Já os dendritos são responsáveis por captar os estímulos recebidos em um determinado período e os transmitirem ao corpo do neurônio (corpo celular), onde são processados. Quando tais estímulos atingirem determinado limite, o

corpo da célula envia um novo impulso que se propaga pelo axônio e é transmitido às células vizinhas por meio de sinapses.

Observando a estrutura de um neurônio, notamos que individualmente ela é bastante simples, porém, quando se tem um conjunto de milhares de neurônios, transforma-se em uma complexa rede de neurônios capaz de processar informações de alta complexidade a uma grande taxa.

Diante do exposto, o aprendizado conexionista se refere a modelos de redes neurais artificiais (RNAs) que se baseiam em modelos matemáticos similares às estruturas biológicas neurais, visando, por meio computacional, apresentar uma capacidade de aprendizado e generalização. Para Rezende (2003, p. 142), o aprendizado, quando aplicado em uma RNA, é “associado à capacidade de adaptarem os seus parâmetros como consequência da sua interação com o meio externo”. Dessa forma, o processo de aprendizado pode ser dito como iterativo, e a RNA sempre busca melhorar o seu desempenho toda vez que exerce uma interação com o meio.

Para Luger (2013), os algoritmos e as arquiteturas que implementam o aprendizado conexionista são normalmente treinados ou condicionados em vez de serem programados. Dentre as aplicações ou tarefas que a abordagem conexionista pode exercer adequadamente, encontram-se:

- Classificação: decide a categoria, ou grupo, a qual pertence um valor de entrada.
- Reconhecimento de padrões: identifica a estrutura ou determinados padrões nos dados.
- Predição: possibilita a identificação de doenças, por exemplo, a partir de determinados sintomas, causas baseadas em efeitos.
- Otimização: encontrara melhor organização de restrições.
- Filtragem de ruído: permite separar o sinal de ruído de fundo, retirando os componentes irrelevantes de um sinal.

4.3.1 Evolução das redes neurais artificiais

As RNAs vêm sendo estudadas ao longo das décadas, uma vez que os pesquisadores buscam um modelo computacional que simule o funcionamento das células do cérebro humano e, principalmente, que seja viável de ser aplicado. As primeiras pesquisas surgiram na década de 1940, com o trabalho de MacCulloch e Pitts, em 1943, no qual os

? Curiosidade

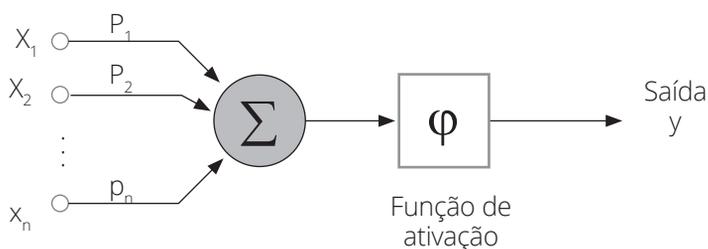
Os neurônios em redes neurais artificiais tendem a ter menos conexões se comparadas as redes neurais biológicas, além de apresentarem tamanhos menores em quantidade de neurônios. Isso é decorrente do processo computacional para treinar essas redes, pois uma rede com milhares de neurônios torna-se inviável de ser construída computacionalmente.

neurônios artificiais ganharam os nomes de MacCulloch e Pitts. Posteriormente, em 1958, Rosenblatt propôs um método inovador de aprendizagem para as redes neurais artificiais denominado *perceptron* e vários trabalhos foram desenvolvidos utilizando esse modelo. Ao final da década de 1960, mais especificamente em 1969, Minsky e Pappert publicaram um livro com importantes explicações relacionadas às limitações encontradas quando adotado o modelo perceptron.

Diante do acontecido, somente na década de 1980, as pesquisas e trabalhos relacionados à área de RNA começaram, em consequência dos avanços tecnológicos e do aumento dos recursos computacionais.

Na Figura 4, o modelo matemático proposto por MacCulloch e Pitts pode ser visualizado. Nesse modelo, observamos um conjunto de sinais de entrada, representado por um vetor $X = [x_1, x_2, x_3, \dots, x_n]$ de dimensão n .

 **Figura 4**
Modelo matemático de uma rede neural



Fonte: Haykin, 2001, p. 36.

Além do conjunto de entradas, o modelo apresentado também possui um conjunto de pesos com valor real, representado por p_i ou w_i (a nomenclatura de representação depende do autor que está sendo referenciado). Os pesos são responsáveis por descreverem as forças de conexão.

Um nível de ativação $\sum p_i x_i$ é determinado pela força cumulativa de seus sinais de entrada, sendo que cada sinal de entrada é escalado pelo peso da conexão. Por fim, uma função de limiar (f) calcula o estado final ou de saída do neurônio, determinando o quanto o nível de ativação do neurônio está abaixo ou acima de um valor de limiar. Em outras palavras, a função de limiar visa produzir o estado de ligado/desligado dos neurônios reais.

Dessa forma, podemos dizer que, além das propriedades individuais de um neurônio, a RNA também é caracterizada por propriedades globais. Conforme descreve Luger (2013):

! **Atenção**

Os sinais de entrada, em uma RNA, podem vir do ambiente ou da ativação de outros neurônios. E o valor do intervalo de entrada é variável e depende do modelo de rede utilizado. Porém, normalmente, as entradas são discretas, ou seja, do conjunto $\{0, 1\}$; $\{-1, 1\}$, ou número reais.

a) Topologia da rede: determina os padrões de conexões entre os neurônios individuais.

b) Algoritmo de aprendizado usado: tipo de algoritmo que será usado no processo de aprendizado da rede.

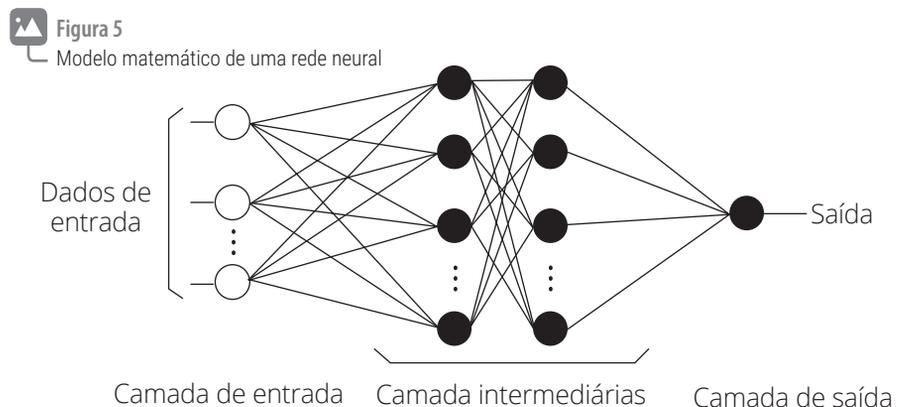
c) Esquema de codificação: interpretação dada aos dados fornecidos para a rede e o resultado do seu processamento.

4.3.2 Redes neurais multicamadas

Ao se reproduzir computacionalmente um problema do mundo real por meio de redes neurais artificiais, percebe-se que esses não são linearmente separáveis, dificultando o uso de um modelo como o *perceptron*. Pois, conforme já descrito na Seção anterior, as RNAs são compostas por um número de neurônios conectados e normalmente organizados em camadas.

Quando nos referimos a um único *perceptron*, esse é considerado como de única camada. Porém, é necessário perceptrons de multicamadas, pois por meio deles é que se torna possível modelar funções mais complexas, incluindo aquelas que não são linearmente separáveis como, por exemplo, a função OU exclusivo (COPPIN, 2010).

Uma rede neural multicamada, ou Multi Layer Perceptron (MLP), apresenta em sua arquitetura uma quantidade de neurônios na camada de entrada, outra quantidade na camada intermediária e por fim na camada de saída, como pode ser visualizado na Figura 5.



Fonte: Adaptada de Coppin, 2010.

A primeira camada, conhecida por *camada de entrada*, é responsável pelos dados de entrada da rede, recebe um sinal de entrada único. Vale ressaltar que geralmente nessa camada os nós não são neurônios,

Artigo

Com a leitura do artigo *Redes Neurais Artificiais: Algoritmo Backpropagation*, publicado no site DEVMEDIA, é possível entender melhor o conceito de problemas linearmente separáveis, quando aplicados a RNAs.

Disponível em <https://www.devmedia.com.br/redes-neurais-artificiais-algoritmo-backpropagation/28559>. Acesso em: 3 fev. 2021.

porém atuam na passagem de sinais de entrada para os nós da camada seguinte, chamada de *camada intermediária* ou *oculta*.

Uma rede neural pode ter uma ou mais camadas intermediárias, a figura anterior ilustra uma RNA contendo duas camadas intermediárias. Outro ponto importante a ser considerado é que cada sinal de entrada é passado para cada um dos nós da camada intermediária e para a saída de cada nó e passado para cada nó na camada final ou de saída. Sendo assim, a camada de saída é responsável por operar a última etapa de processamento e enviar os sinais de saída.

Considerando a forma como os dados e sinais são passados pela rede, essa pode ser definida com uma alimentação à frente, ou seja, os dados são alimentados a partir dos nós de entrada e vão em direção aos nós de saída.

Outro ponto importante a ser estudado quando se trabalha com redes neurais é a forma como aprendem. Em redes multicamadas, o aprendizado é igual ao de uma rede *perceptron*. Porém, tem como diferença que cada neurônio possui pesos associados às suas entradas e, nesse caso, existem muitos pesos a serem ajustados quando determinado erro é cometido por um fragmento de dados de treinamento.

Um método que é bastante usado para atribuir responsabilidades aos diferentes pesos é a retropropagação. Em outras palavras, a retropropagação fornece um algoritmo para atribuir aos neurônios a sua parcela de culpa pelo erro da rede e ajustar os pesos de forma correspondente (LUGER, 2013).

4.3.3 Modelos de redes neurais artificiais

As redes neurais artificiais apresentam vários modelos que normalmente são classificados de acordo com o tipo de aprendizado usado, ou seja, supervisionado ou não supervisionado. Modelos de redes neurais que usam aprendizado supervisionado se caracterizam pela existência de um supervisor externo a rede que tem a função de monitorar a resposta de saída para cada um dos vetores de entrada. Já o aprendizado não supervisionado não tem existência de qualquer saída desejada para as respectivas entradas, isto é, não existe um supervisor externo e os ajustes de pesos são realizados apenas com base nos valores de entrada (REZENDE, 2003).

? Curiosidade

Uma rede neural com alimentação adiante é composta por uma camada de entrada, uma ou duas camadas intermediárias, e uma camada de saída. A quantidade de neurônios em cada camada pode variar entre 10 e 1000.

📖 Leitura

Para saber mais sobre o funcionamento de uma rede neural que adota o aprendizado por retropropagação, leia a Seção 11.3, intitulada por *Aprendizado por retropropagação*, do livro *do Inteligência Artificial*, de Luger (2013).

▶ Vídeo

O vídeo *Backpropagation Descomplicado* irá te auxiliar a compreender sobre o funcionamento do algoritmo Backpropagation usado em uma MLP.

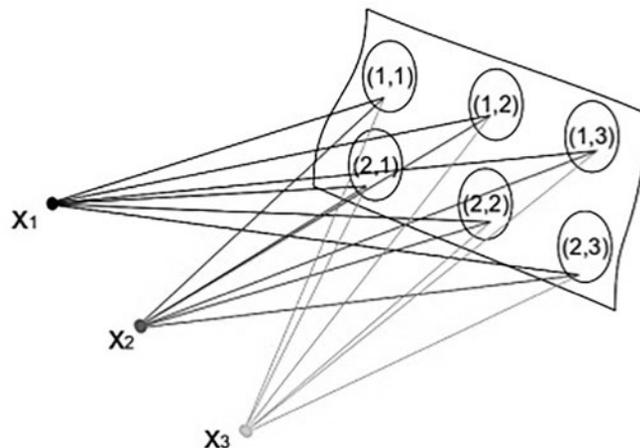
Disponível em: <https://www.youtube.com/watch?v=L3lbrdCTK9w>. Acesso em: 3 fev. 2021.

Geralmente quando adotamos um treinamento de dados usando um aprendizado supervisionado é porque desejamos obter um mapeamento entre os padrões de entrada e saída existentes. Os modelos de RNA que utilizam o aprendizado não supervisionado são as redes de Kohonen.

As redes de Kohonen foram desenvolvidas por Teuvo Kohonen e adotam um modelo de mapas auto-organizáveis, com alimentação para frente. Seus neurônios são dispostos como nós de uma grade uni ou bidimensional. Esse modelo tem como característica o aprendizado competitivo, em outras palavras, os neurônios competem entre si, sempre que uma determinada entrada à rede for apresentada, e o vencedor ajusta seus pesos para que os estímulos possam ser impulsionados para a saída.

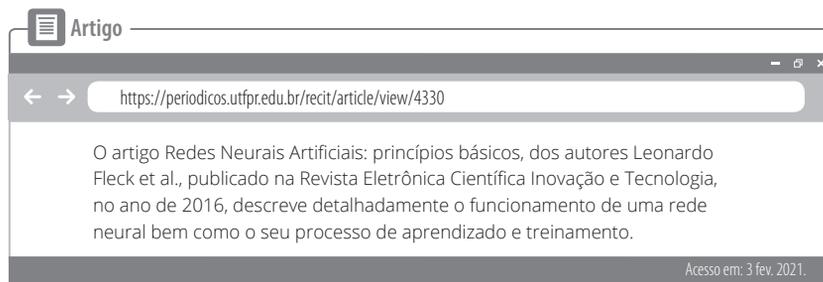
As redes de Kohonen se caracterizam por apresentar um aprendizado competitivo, ou seja, sempre que um conjunto de neurônios ou apenas um neurônio for apresentado à rede, ele irá competir entre si. O neurônio vencedor terá seus pesos ajustados e balanceados para que possam gerar os próximos estímulos.

 **Figura 6**
Rede de Kohonen



Fonte: Adaptado de Ludwig Jr., Costa, 2007.

Nessa rede, percebemos que existem três nós na camada de entrada e seis neurônios dispostos.



Outro modelo de RNA para aprendizado não supervisionado é o desenvolvido, em 1949, por Donald Hebb, chamado de *aprendizado de Hebb*. Esse modelo se baseia no princípio de que, se dois neurônios em uma rede neural estiverem conectados e forem ativados ao mesmo tempo, quando ocorrer uma entrada particular na rede, a conexão entre esses dois neurônios deverá ser reforçada.

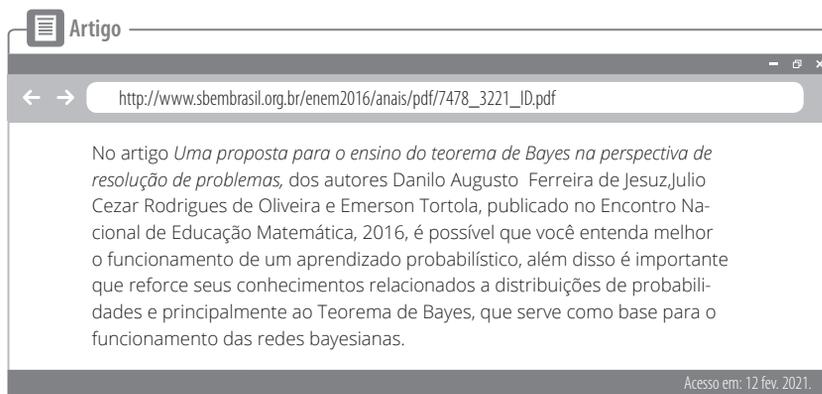
4.4 Aprendizado probabilístico

▶ Vídeo



O aprendizado probabilístico se utiliza da mesma teoria para discutir eventos, categorias e hipóteses, sendo que sobre qualquer dessas não se tem 100% de certeza.

Um modelo de aprendizado probabilístico pode ser construído usando os conceitos das regras de Bayes. Sempre que se deseja utilizar a teoria da probabilidade em aprendizado, inicialmente é necessário entender sobre a probabilidade de uma determinada hipótese ser verdadeira, dado um conjunto de evidências (COPPIN, 2010).



Dado o exemplo, adotando as notações de probabilidade:

$P(H|E)$

Diante dessa notação, podemos determinar em qual conjunto de hipóteses acreditar, ao identificar a probabilidade *a posteriori* de cada uma delas. Vamos supor que existem n hipóteses possíveis H_1, \dots, H_n , onde para cada H_i temos

$$P(H_i | E) = \frac{P(E | H_i) \cdot P(H_i)}{P(E)}$$

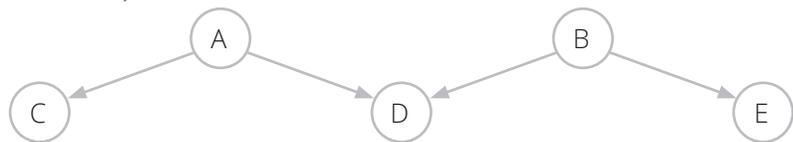
Utilizando-se dessa fórmula de probabilidade, o algoritmo poderá calcular $P(H_i | E)$ para cada uma das possíveis hipóteses e selecionar aquela que tiver maior probabilidade.

Ainda em relação à fórmula anterior, podemos sintetizá-la como $P(E)$ é **independente** de H_i , ou seja, ela terá o mesmo valor para cada uma das hipóteses existentes. Esse tipo de aprendizado por meio de probabilidade pode ser usado em aplicações que exigem que novas ações sejam realizadas. No caso de um robô, por exemplo, as ações serão executadas considerando os passos anteriores e a probabilidade de se atingir o objetivo final mais rapidamente.

Conforme já descrito, o conceito de dependência é fundamental quando se aplica a teoria da probabilidade. Ou seja, dois eventos A e B, serão independentes se a probabilidade de ocorrência de A for inteiramente desvinculada da ocorrência ou não de B.

Essas relações podem ser expressas de forma extremamente sucinta em redes de crenças, conforme mostra a Figura 7, que apresenta uma rede bayesiana de crença em forma de grafo orientado acíclico, ou seja, onde os nós no grafo representam as evidências ou hipóteses e onde um arco que conecte dois nós representa a dependência entre esses nós.

 **Figura 7**
Rede de crença



Fonte: Adaptada de Coppin, 2010, p. 295.

Artigo

O artigo *Um modelo para recomendação de artigos acadêmicos baseado em filtragem colaborativa aplicado à ambientes móveis*, dos autores Silvio Cazella, Irismar Córrea das Chagas e Jorge Barbosa, publicado na revista *Novas Tecnologias na Educação*, de 2008, é ótimo para entender como funciona a aplicação da filtragem colaborativa em um determinado problema.

Disponível em: <https://www.seer.ufg.br/renote/article/viewFile/14458/8380>. Acesso em: 3 fev. 2021.

Podemos observar que essa rede apresenta cinco nós que representam elementos da evidência (A e B) e três hipóteses (C, D, E). Sendo que os arcos entre os nós se referem à interdependência das hipóteses. Analisando o gráfico, podemos perceber que C e D são ambos dependentes de A; e D e E são ambos dependentes de B.

Já quando dois nós que não possuem um arco entre eles são independentes um do outro, ou seja, podemos dizer que b é independente de A. Normalmente a lista de probabilidade quando combinada com o diagrama da Figura 5 representa uma rede bayesiana completa.

Outro uso bastante prático para uma rede ou raciocínio bayesiano é a filtragem colaborativa, que é uma técnica usada pelas lojas virtuais que tem como objetivo oferecer sugestões plausíveis aos clientes, sempre levando em consideração as aquisições anteriormente realizadas.

Várias são as formas de implementação da filtragem colaborativa e, dentre as existentes, a inferência bayesiana tem demonstrado bastante sucesso. Em outras palavras, significa trabalhar com as probabilidades *a posteriori*.



CONSIDERAÇÕES FINAIS

Aprendizado de máquina é uma subárea da Inteligência Artificial que possibilita a criação de sistemas que buscam, por meio do aprendizado, tomar decisões de forma mais inteligente e automática, por meio de algoritmos aplicados a determinadas bases de dados.

O processo de aprendizado pode ser classificado como supervisionado e não supervisionado, ou seja, no primeiro caso todo exemplo de um conjunto de treinamento possui uma classe predeterminada. Assim, sempre que um novo exemplo seja inserido, ele será atribuído a uma determinada classe. Já o aprendizado não supervisionado não apresenta uma classe predeterminada associada aos exemplos.

Além disso, quando nos referimos a aprendizado de máquina, ele pode ser de modo simbólico, conexionista e probabilístico. Aprendizados simbólicos são normalmente usados por meio de árvores de decisão, que podem ser representadas por meio de linguagens como lógica de primeira e segunda ordem, lógica de atributos, entre outras.

Já para o aprendizado conexionista, as redes neurais artificiais são o modelo mais usado, e são uma representação computacional e seme-

lhante ao funcionamento do cérebro humano. As RNAs normalmente são multicamadas, isto é, compostas por um, ou mais, neurônio na camada intermediária da rede.

Por fim, o aprendizado probabilístico utiliza da matemática para resolver problemas estocásticos e determinísticos, sendo que muitas vezes isso também ocorre de forma probabilística. Dentre os principais modelos de aprendizado probabilístico existentes, estão as redes beasyanas, cadeias de Markov, entre outros.



ATIVIDADES

1. O que é aprendizado de máquina?
2. Defina o que são atributos em aprendizado de máquina.
3. O que compõe uma rede neural multicamadas?
4. O que é aprendizado de máquina probabilístico?



REFERÊNCIAS

COPPIN, B. *Inteligência artificial*. Rio de Janeiro, LTC, 2010.

FLECK, L. *et al.* Redes Neurais Artificiais: princípios básicos. *Revista Eletrônica Científica Inovação e Tecnologia*, v. 1, n. 13, p. 47-57, 2016.

HAYKIN, S. *Redes Neurais: princípios e prática*. Selo Bookman, 2001.

LUDWIG JR., O.; COSTA, E. M. M. *Redes Neurais Artificiais: fundamentos e aplicações com programas em C*. Rio de Janeiro: Ciência Moderna, 2007.

LUGER, G. F. *Inteligência artificial*. 6. ed. São Paulo: Pearson Education do Brasil, 2013.

REZENDE, S. O. *Sistemas Inteligentes: fundamentos e aplicações*. São Paulo: Manole, 2003.

RUSSELL, S.; NORVIG, P. *Inteligência artificial*. Rio de Janeiro: Campus, 2010.

5

Computação evolutiva

A computação evolutiva pode ser aplicada a uma variedade de disciplinas que vão da biologia à engenharia da computação. Sua principal característica está relacionada à resolução de problemas e fundamentada em técnicas ou em modelos que imitam o processo da evolução natural.

Comportamentos inspirados na teoria evolutiva são modelados matematicamente e implementados por meio de computadores e, então, aplicados para solucionar determinados problemas. Diante do exposto, podemos dizer que a computação evolutiva engloba uma família de algoritmos inspirados na teoria evolutiva de Darwin.

Dentre os principais modelos e técnicas encontram-se os algoritmos genéticos – muito aplicados na resolução de problemas nas mais diversas áreas de atuação – e a programação genética com os conceitos computacionais para criação automática de programas, embasando-se na descrição em alto nível de um problema a ser resolvido. E, por fim, um conjunto de algoritmos evolutivos que são capazes de imitar o comportamento de vários insetos, mamíferos, entre outros, e atuar em um determinado problema.

5.1 Fundamentos da computação evolucionária –

▶ Vídeo



Os sistemas com base na computação evolucionária tiveram início pelos pesquisadores na década de 1950, quando buscavam sistemas que resolvessem problemas por meio de modelos computacionais fundamentados na teoria da evolução natural.

Essa teoria, desenvolvida por Darwin em 1859, é a principal ideia na área biológica, pois foi por meio dela que se teve uma força propulsora

que permitiu distinguir as diferenças existentes entre os sistemas biológicos e os demais sistemas físicos e químicos.

Nesse sentido, podemos dizer que a teoria da seleção natural não prevê apenas a ocorrência de possíveis variedades de indivíduos que pertencem a uma determinada espécie, ela permite identificar as possíveis variações que essa espécie poderia aderir para se adequar ao meio em que vive.

A teoria evolutiva vem sendo adotada há algumas décadas e assim como outros modelos, como a lógica *fuzzy* e as redes neurais, não foi aceita e reconhecida de imediato. Com os avanços tecnológicos e a necessidade de resolução de problemas complexos com base em simulações computacionais, os algoritmos evolutivos passaram a ser reconhecidos.

Atualmente, a computação evolutiva é considerada uma alternativa para problemas com complexidade alta, pois por meio de técnicas e/ou modelos que imitam o comportamento de determinados animais, torna-se mais fácil a obtenção de uma solução satisfatória para um conjunto de amostras pertencentes a um espaço de busca.

Independentemente do comportamento da evolução natural adotado pelos algoritmos evolutivos, estes precisam corresponder a uma sequência de passos até que se encontre o resultado desejado. Resumidamente, a computação evolutiva pode ser entendida como um conjunto de técnicas ou de procedimentos capazes de se adaptar ao meio.

Assim, a teoria evolucionária e a computação foram criadas em épocas similares, sendo que aquela combina as ideias de Darwin com a genética sobre a seleção natural por meio do princípio básico da genética populacional (REZENDE, 2003).

Com o objetivo de melhor entendermos o processo de seleção natural proposta por Darwin, é interessante levarmos em consideração as seguintes hipóteses propostas por Linden (2008):

- a) Os filhos tendem a ser em maior número que os pais.
- b) O número de indivíduos de uma espécie permanece aproximadamente constante.
- c) De (a) e (b) se conclui que haverá uma luta pela sobrevivência.
- d) Dentro de uma mesma espécie, os indivíduos apresentam pequenas diferenças, sendo que a maioria delas também está presente nos respectivos pais.

Dessa forma, podemos entender que o princípio da seleção natural indica que os indivíduos com as variações melhor adaptadas ao ambiente possuem maior probabilidade de sobreviver e de se reproduzir no meio em que se encontram inseridos. Em outras palavras, essa evolução é uma competição entre sistemas de reprodução de informação, operando no interior de uma arena finita em um universo.

É importante destacarmos que quando se adota uma técnica com base na computação evolucionária é necessário que determinados conceitos sejam entendidos, pois são bastante aplicados nas fases de funcionamento de seus processos, conforme demonstra o Quadro 1.



Quadro 1

Analogia em relação aos sistemas naturais

Natureza	Descrição no modelo evolucionário
Cromossomo	Palavra binária, vetor etc.
Gene	Característica do problema
Alelo	Valor da característica
Loco	Posição na palavra, vetor
Genótipo	Estrutura
Fenótipo	Estrutura submetida ao problema
Indivíduo	Solução
Geração	Ciclo

Fonte: Adaptado de Linden, 2008.

Como podemos verificar, todos os conceitos representados na natureza por meio das espécies podem ser aplicados ao processo computacional. O que difere o uso dos conceitos é o modelo a ser adotado, por exemplo, em algoritmos genéticos grande parte deles são aplicados, assim como na programação genética.

5.2 Algoritmos genéticos

Vídeo



Algoritmos genéticos (AG) fazem parte de um ramo dos algoritmos evolucionários e são fundamentados em técnicas de busca que imitam o processo biológico da evolução natural (LINDEN, 2008).

Outra definição para AG é embasada no algoritmo de busca considerando os mecanismos de seleção natural e genética, ou seja, trata-se da combinação e da sobrevivência dos melhores indivíduos como forma estruturada da troca de informação genética entre dois indivíduos.

Vídeo

Para entender mais sobre a teoria da evolução natural, assista ao vídeo *Introdução à Teoria da evolução e à seleção natural* em que é possível visualizar o processo de maneira ilustrada e compreender as variações que ocorrem durante o processo.

Disponível em: <https://pt.khanacademy.org/science/biology/her/evolution-and-natural-selection/v/introduction-to-evolution-and-natural-selection>. Acesso em: 2 fev. 2021.

Artigo

No artigo *Algoritmos genéticos e variantes na solução de problemas de configuração de redes de distribuição*, dos autores Bento e Kagan, publicado em 2008, é possível perceber que algoritmos genéticos podem ser aplicados à resolução de problemas das mais diversas áreas de atuação, e a aplicação da técnica está relacionada à otimização das redes de distribuição visando à minimização de perdas elétricas.

Disponível em: https://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592008000300006. Acesso em: 2 fev. 2021.

Dentre as características dos AG, Linden (2008) apresenta algumas que destacamos a seguir:

- São técnicas probabilísticas.
- Considerados problemas simples e que necessitam de informações locais ao ponto avaliado.
- Aplicáveis a problemas do mundo real.
- Trabalham com uma grande população de pontos.

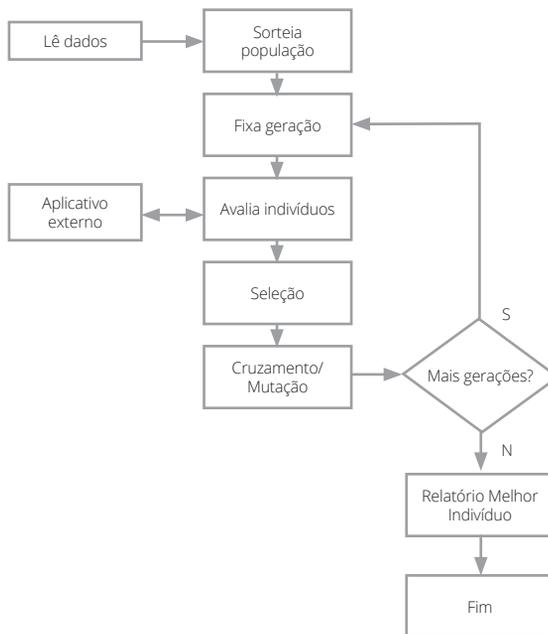
O processo de um algoritmo genético, normalmente, é iniciado com uma população de indivíduos gerados aleatoriamente. Sendo que cada indivíduo (cromossomo) faz a codificação de uma única solução possível para o problema em questão (COPPIN, 2010).

Os indivíduos que tiverem melhor aptidão considerando o resultado da função de aptidão (*fitness*), é avaliado para cada membro da população. Todos os indivíduos que apresentarem maior valor da aptidão representam uma melhor solução para o problema do que os indivíduos que apresentam valores inferiores aos escolhidos. A Figura 1 ilustra o funcionamento básico de um algoritmo genético.



Figura 1

Passo a passo de um algoritmo genético



Fonte: Adaptado de Linden, 2008.

Com relação à população inicial, é feito um sorteio que permite a realização da geração dos número aleatórios de probabilidade uniforme. Dessa forma, a população inicial é realizada de maneira simples por meio de uma escolha aleatória independente para cada indivíduo. Ou seja, escolher n indivíduos dentro do espaço de pesquisa.

Já a avaliação dos indivíduos de cada geração é feita pela função que é definida inicialmente. Toda vez que se avalia qualquer processo de seleção, é necessário que essa função seja normalizada, assim, o resultado dela pode ser associado a uma probabilidade de ser selecionado para o cruzamento na sequência.

A função de avaliação é a parte fundamental dos algoritmos genéticos, pois é por meio dela que se avalia o quão próximo determinado indivíduo está da solução desejada ou não.

Outra fase, chamada *seleção de indivíduos*, sempre visa simular o mecanismo semelhante à seleção natural das espécies, em que os indivíduos mais fortes e adaptados ao ambiente são os que sobrevivem, conforme descrito anteriormente. Mesmo os indivíduos menos aptos são capazes de gerar filhos, possibilitando maior diversidade entre os indivíduos bons e ruins. Isso acontece porque, se apenas os melhores indivíduos podem reproduzir, a população tende a ser mais similar a cada nova reprodução.

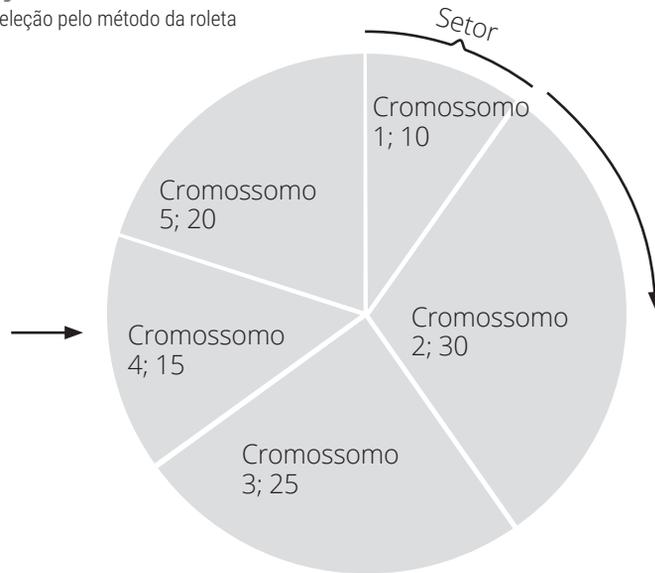
No algoritmo genético, a solução para a seleção dos indivíduos pode ser realizada pelo método da roleta, o qual consiste na associação de cada cromossomo a um pedaço da roleta, sendo que esta tenha tamanho proporcional a sua função de avaliação. Os passos para a seleção do indivíduo por meio da roleta são apresentados a seguir e podem ser visualizados na Figura 2.

1. Sortear um número entre o valor 0 e a soma das funções de avaliação.
2. Somar os valores dos setores alocados a cada um dos cromossomos.
3. O setor que tiver uma soma que ultrapassar o número sorteado será o escolhido.

Atenção

Outros métodos, como o do torneio e o da amostragem universal estatística, também podem ser usados para a seleção dos indivíduos em uma determinada população. O método da roleta é considerado o mais simples de ser implementado computacionalmente e como consequência o mais utilizado quando adotado o algoritmo genético na resolução de problemas.

Figura 2
Seleção pelo método da roleta



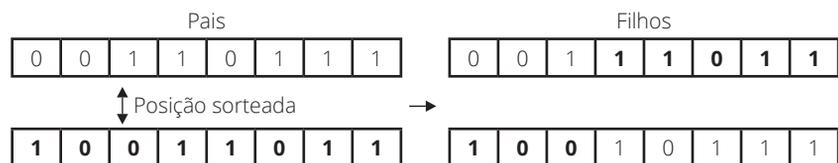
Fonte: Adaptado de Linden, 2008.

Durante esse processo pode ocorrer a perda do melhor indivíduo de determinada geração, ou seja, o indivíduo mais apto pode não ser escolhido para formar a próxima geração.

O cruzamento é a etapa após a seleção. No primeiro momento, um número é sorteado entre 0 a 1, se o valor sorteado for menor que a taxa de cruzamento, segue o processo, caso contrário, os cromossomos dos pais são repetidos para a próxima geração.

O processo de cruzamento pode ser realizado para apenas um ou mais de um ponto. Quando escolhido o cruzamento de um ponto, o sorteio é feito entre o número 1 e o tamanho máximo da *string*, de modo que os filhos tenham os genes dentro desse intervalo de um dos pais e outro número +1 até o final do outro. A figura a seguir ilustra melhor o processo de cruzamento realizado entre os pais.

Figura 3
Cruzamento de um ponto

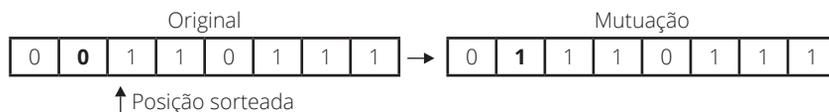


Fonte: Adaptado de Linden, 2008.

Na Figura 3 é possível visualizarmos a escolha da posição sorteada, no caso, o índice 3 do vetor de um dos pais. Dessa maneira, podemos ver no primeiro filho que seus genes são, inicialmente, iguais aos do pai e os restantes são iguais ao do segundo pai e vice-versa. Pois, dessa forma, sempre é realizada uma combinação entre os genes.

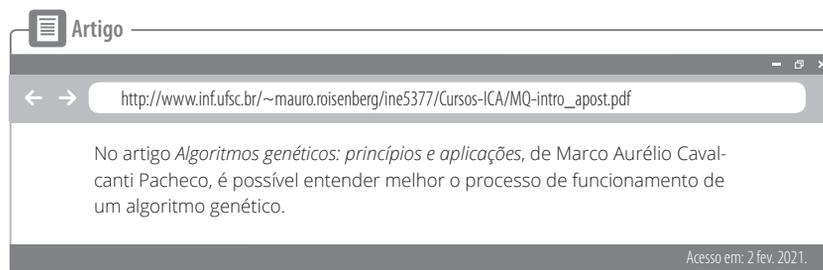
O processo de mutação é outra fase primordial nos algoritmos genéticos, visto que esse processo permite que saltos qualitativos na população possam ser analisados (LINDEN, 2008). Para que isso ocorra, cada gene de toda a população sorteia um número entre 0 e 1, sendo o número sorteado menor do que a taxa de mutação definida, é realizada a troca do valor daquele *bit*, isto é, se o valor era 0 passa a ser 1 e vice-versa, conforme ilustrado na Figura 4.

 **Figura 4**
Processo de mutação



Fonte: Linden, 2008, p. 78.

Notamos na imagem que por meio do salto foi possível verificar a posição 2 do vetor e, então, foi realizada a mutação que é a troca do valor do *bit*. No exemplo, o vetor original, nessa posição, tinha o valor zero e no processo de mutação passa a ter o valor 1 nessa posição, sendo que as demais permanecem iguais.



Considerando as etapas expostas e o principal funcionamento de um algoritmo genético, concluímos que as operações de *crossover* e mutação são muito usadas onde os indivíduos da população produzem seus descendentes e para estes são atribuídas pontuações de *fitness*, ou seja, pontuações geradas com base na função de avaliação.

5.3 Programação genética

▶ Vídeo



Como podemos ver na ficção científica, durante muitos anos cientistas vêm elaborando filmes em que computadores podem se autoprogramar e, assim, tomar decisões e executar tarefas da mesma forma que seres humanos, por exemplo, nos filmes *Guerra nas estrelas*, *Matrix*, entre outros.

A programação genética (PG) é um ramo da computação evolucionária que visa tentar evoluir programas para que eles possam resolver um determinado problema em questão (LINDEN, 2008). Porém, quando trabalhamos com a programação genética, nos deparamos com o problema que está relacionado à sequência dos dados de entrada e de saída, de modo que determinado programa possa fazer um melhor mapeamento entre eles.

Nesse sentido, entendemos que a PG é um parente próximo dos algoritmos genéticos, pois como existem programas e expressões ao invés de apenas conjunto de valores, a avaliação de cada cromossomo torna-se possível por meio da execução de cada programa representado para todos os conjuntos de dados existentes, possibilitando determinar o quão boa é a saída do programa que representa a saída desejada (LINDEN, 2008).

O funcionamento da programação genética é igual ao funcionamento dos algoritmos genéticos, no qual é preciso encontrar um forma de codificar os programas e, então, aplicar operadores de *crossover* e mutação, de maneira a evolui-los a ponto de encontrar a melhor solução para o problema em questão.

Independentemente da PG ser parecida com os AG, é importante destacarmos determinados motivos que precisam de atenção, os quais Linden (2008) descreve da seguinte maneira:

- Programação genética, rotineiramente, produz inteligência de máquina em nível humano, não apenas em problemas de teste, que normalmente são pouco significativos em termos de aplicabilidade.
- A PG é uma máquina automática de invenções, além disso, podemos dizer que não se prende a crenças e a axiomas existentes, ela normalmente descobre soluções que nunca seriam imagina-

das por cientistas ou por engenheiros que sempre se baseiam em conceitos científicos vigentes.

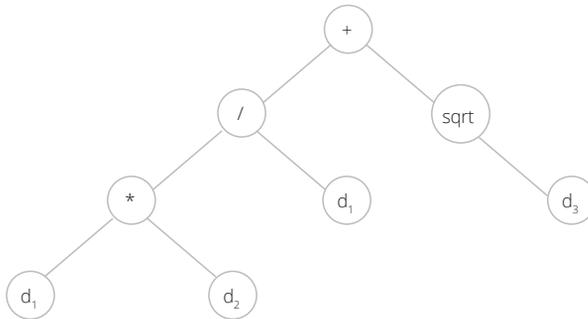
- Possibilita criar, facilmente, uma solução geral para um problema com uma solução parametrizada de maneira automática. Em outras palavras, a PG pode criar uma solução para um problema por meio de uma estrutura genérica.
- Os resultados obtidos pela PG melhoram em termos qualitativos cinco ordens de grandeza mais rápido do que o tempo de computação gasto.

Grande parte dos trabalhos que adotam a programação genética como forma de resolução de determinado problema usa as árvores na qualificação de representação dos cromossomos, conforme ilustrado na Figura 5.



Figura 5

Representação de um indivíduo em programação genética



Fonte: Adaptado de Linden, 2008.

Nesse exemplo, podemos verificar que os nós folhas são os terminais e são representados por d_1 , d_2 e d_3 . Enquanto a função (nó não folhas) é representada pelos operadores $+$, $-$, $/$, $*$, sqrt e \log . Posteriormente, temos os demais componentes como a função de avaliação/adequação (*fitness*), reprodução, mutação e *crossover*. E conforme mostra a Figura 6, os passos para o funcionamento da programação genética são: iniciar população, avaliar indivíduo e aplicar evolução de modo a compreender quem são os indivíduos candidatos na escolha do melhor indivíduo.

www. Site

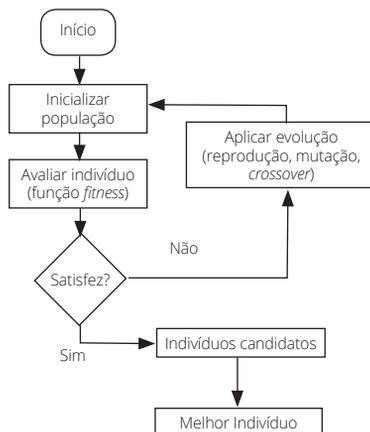
O site *Genetic-programming* afirmou há 15 anos que já existiam problemas e que a programação genética havia produzido um resultado que era competitivo com o desempenho humano. Imagine atualmente como isso ocorre. Acesse o site e veja o que é publicado em relação a essa área da computação evolutiva.

Disponível em: <http://www.genetic-programming.org>. Acesso em: 2 fev. 2021.

! Atenção

Árvore é um tipo de dado abstrato que possibilita a representação de seus elementos de modo hierárquico. Sendo os elementos compostos por raiz e nós terminais e intermediários.

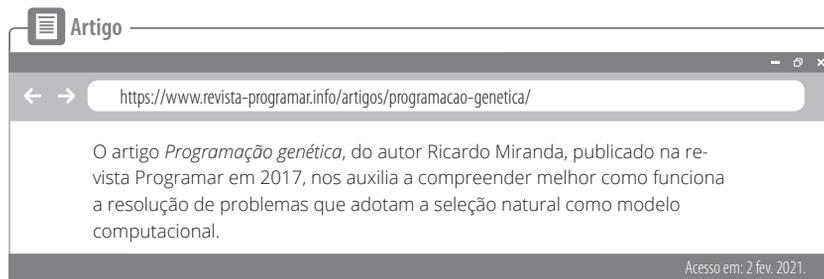
 **Figura 6**
Funcionamento da PG



Fonte: Adaptado de Brito; Salgado; Beijo, 2016.

Assim, a Figura 6 pode ser traduzida para a sequência de passos que compõe a programação genética, como:

- criar aleatoriamente uma população de programas;
- executar os seguintes passos até que um Critério de Término seja satisfeito;
- avaliar cada programa por meio de uma função heurística (*fitness*), que expressa quão próximo cada programa está da solução ideal;
- selecionar os melhores programas de acordo com o *fitness*;
- aplicar a esses programas os operadores genéticos (reprodução, cruzamento e mutação);
- retornar com o melhor programa encontrado.



Em outras palavras, cada execução desse laço representa uma nova geração de programas. Tradicionalmente, o Critério de Término é estabelecido como: encontrar uma solução satisfatória ou atingir um número máximo de gerações.

5.4 Algoritmos evolutivos

Video



Na computação evolucionária existem diversos algoritmos que são usados e denominados *evolutivos*, dos quais podemos destacar: Otimização de Enxames de Partículas, Evolução Diferencial e Colônia de Formigas.

A Otimização de Enxames de Partículas (PSO) surgiu em 1995, proposta por Kennedy e Elberhart, com o intuito de solucionar problemas contínuos. Existem aspectos dessa técnica muito semelhantes a algoritmos evolutivos, como a geração randômica da população inicial, entretanto, a Otimização por Enxame de Partículas associa velocidades para cada partícula e as permite “navegar” por um espaço de possíveis soluções definidas.

Essa teoria baseia-se em que cada indivíduo de uma população possui experiência própria e é capaz de medir a qualidade desta. Sendo assim, com a sociabilidade dos indivíduos, estes apresentam conhecimentos sobre o comportamento de seus vizinhos. Assim, essas duas formas de compreensão correspondem à aprendizagem individual e à transmissão cultural, respectivamente.

Por sua vez, a Evolução Diferencial (DE) foi desenvolvida por Storn e Price, em 1995, e surgiu de tentativas de resolver o problema de ajuste polinomial de Chebychev. O processo da DE pode ser descrito em 4 etapas principais:

- a. Inicialização – as variáveis de controle e o vetor população são inicializados.
- b. Mutação – ocorre a mutação de soluções do vetor populacional do algoritmo.
- c. Combinação – ocorre a combinação das soluções que sofreram mutações com outros presentes no vetor populacional.
- d. Seleção – as soluções são comparadas e a menos apta é eliminada.

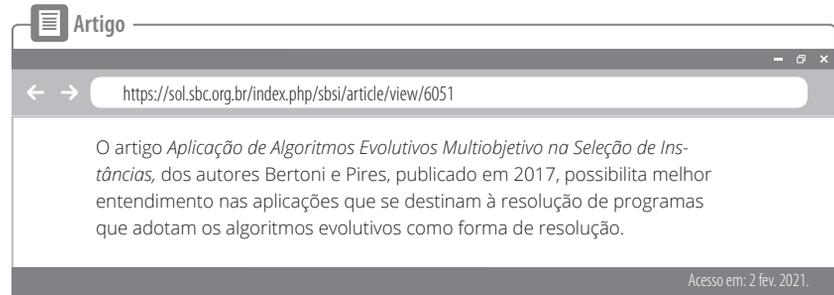
Os passos do algoritmo são compostos pela inicialização aleatória de uma população, de modo que esta possa abranger todo o espaço de busca do problema abordado. Diante disso, torna-se possível que novos indivíduos sejam criados, em consequência do processo de mutação e de combinação entre os parâmetros estabelecidos no algoritmo.

Filme

Com o filme *Transcendence: A Revolução* entenderemos mais sobre a computação evolutiva, seus algoritmos e a forma desejada para a construção de máquinas que tenham pensamentos e ações similares aos seres humanos e demais espécies. O filme aborda muitos conceitos da inteligência artificial e do aprendizado de máquina.

Direção: Wally Pfister. EUA; Reino Unido: Diamond Films, 2014.

Os algoritmos conhecidos como *Colônia de Formigas* (ACO) baseiam-se no comportamento das formigas, que mesmo sendo insetos praticamente cegos conseguem estabelecer o menor caminho entre o formigueiro e a fonte de alimento e retornar da mesma forma. Uma formiga em movimento excreta uma certa quantidade de feromônio no solo, criando assim uma trilha dessa substância biologicamente muito ativa.



Enquanto uma formiga isolada se move de uma maneira absolutamente aleatória, uma formiga encontrando uma trilha já percorrida por outra detecta seu rastro e decide com alta probabilidade seguir o mesmo caminho, dessa forma, reforçando a trilha escolhida com seu próprio feromônio. Esse processo é, então, caracterizado por uma re-atoalimentação (feedback) positiva, uma vez que a probabilidade de uma formiga escolher um determinado caminho é um tanto maior quanto maior for o número de formigas que já o tenham percorrido.



CONSIDERAÇÕES FINAIS

A computação evolutiva surgiu e vem sendo adotada durante muitos anos e foi na década de 1970 que passou a ser mais utilizada em pesquisas e publicações de vários autores, principalmente, para trabalhos que envolvessem a otimização de processos de determinado problema.

Em síntese, os passos necessários para a aplicação da computação evolutiva são fundamentados em indivíduos que sofrem mutações a cada geração, sendo que os indivíduos mais adaptados ao meio são capazes de se reproduzir gerando filhos e ampliando a seleção natural para as novas gerações.

Considerando essas explicações, a computação evolutiva busca por meio de modelos, de técnicas e de algoritmos representar o comportamento dos seres vivos de modo que tenham maior desempenho e rapidez na

resolução de problemas. Para que isso seja possível, modelos como algoritmos genéticos, programação genética e algoritmo da formiga são usados em estudos e em aplicações nas mais variadas áreas de atuação.



ATIVIDADES

1. O que é a computação evolutiva?
2. Qual a principal finalidade dos algoritmos genéticos?
3. O que é a mutação em algoritmos evolucionários?
4. Cite dois algoritmos evolutivos.



REFERÊNCIAS

BRITO, O. B; SALGADO, R. M. ; BEIJO, L. A. Intelligent Modeling for Streamflow Forecasting. *IEEE Latin America Transactions*, v. 14, n. 8, p. 3669-3677, ago. 2016.

COPPIN, B. *Inteligência artificial*. Rio de Janeiro: LTC, 2010.

LINDEN, R. *Algoritmos Genéticos: uma importante ferramenta da inteligência computacional*. 2. ed. Rio de Janeiro: Brasport, 2008.

REZENDE, S. O. *Sistemas Inteligentes: fundamentos e aplicações*. São Paulo: Manole, 2003.

GABARITO

1 Introdução à inteligência artificial

1. É um ramo de pesquisa da Ciência da Computação que se ocupa de desenvolver mecanismos e dispositivos tecnológicos que simulam o raciocínio humano.
2. O Teste de Turing, criado por Alan Turing em 1950, tem como objetivo testar a capacidade de uma máquina exibir determinado comportamento inteligente e equivalente a um ser humano.
3. Várias são as áreas que recebem aplicações que adotam recursos de IA, mas dentre elas destaca-se a área de processamento de linguagem natural, a qual visa construir sistemas computacionais que conseguem compreender a linguagem oral e escrita. Exemplo: programas de reconhecimento e de interpretação da fala, sistemas capazes de gerar voz a partir de texto etc.
4. Um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores e de agir sobre ele por intermédio de atuadores.

2 Resolução de Problemas

1. A busca por meio de grafos ou árvores pode ser usada na resolução de problemas, pois permite a representação computacional de situações que demandam determinado tipo de inteligência no desenvolvimento das aplicações. São várias as aplicações que adotam o método de resolução de problemas em busca, como problemas de roteamento de rotas, navegação de robôs, problemas do caixeiro viajante, entre outros.
2. A busca em largura é uma técnica que permite percorrer uma árvore de busca examinando todos os nós um nível abaixo da raiz. Se um estado objetivo for encontrado, recebe-se uma informação de sucesso; caso contrário, a busca prossegue pela expansão de caminhos, partindo de todos os nós do nível corrente em direção ao próximo nível.
3. Uma das técnicas da busca heurística é a subida da encosta da montanha. Essa abordagem é um método de busca que envolve continuamente o ato de ir de uma solução em potencial para outra melhor até que não se tenha solução melhor.

4. Os algoritmos que implementam técnicas de resolução de problemas em busca necessitam apresentar as seguintes propriedades: complexidade, completude, admissibilidade, irrevogabilidade e quanto a ser ótimo.

3 Sistemas baseados em conhecimento

1. Um sistema baseado em conhecimento é um programa de computador que utiliza o conhecimento representado para resolver problemas, isto é, são desenvolvidos para serem utilizados em problemas que requerem uma quantidade considerável de conhecimento humano e de perícia.
2. Sistemas especialistas são programas de computador que manipulam conhecimento e informações de maneira inteligente, ou seja, são desenvolvidos para resolver problemas que requerem uma grande quantidade de conhecimento humano e de especialização.
3. O núcleo do sistema baseado em conhecimento é constituído de um módulo coletor de dados (MCD) que faz a interação com o usuário e obtém as informações a respeito do problema em questão; um motor de inferência (MI) responsável pelo raciocínio baseado nas informações do MCD; e o módulo de explicações (ME), responsável pelas explicações geradas com base nas perguntas realizadas pelo SBC.
4. Aquisição do conhecimento pode ser definida como o processo de compreender e organizar o conhecimento de várias fontes, o qual deverá ser codificado e armazenado em uma base de conhecimento para posterior resgate por um sistema especialista. As fases do processo de aquisição do conhecimento são: identificação, conceituação, formalização, implementação e testes.

4 Aprendizado de máquina

1. Aprendizado de máquina é um subcampo da ciência da computação que sofreu evolução da parte que estuda o reconhecimento de padrões e da teoria do aprendizado computacional da área de inteligência artificial.
2. Atributos em aprendizado de máquina são usados para representar uma característica ou um aspecto de determinado exemplo, sendo que esses podem ser nominais ou contínuos.
3. Uma rede neural multicamadas é composta por um conjunto de neurônios na camada de entrada, uma ou mais camadas intermediárias

ou camadas ocultas, que são interligadas as camadas de entrada e de saída por meio de pesos, e uma camada de saída.

4. O aprendizado de máquina probabilístico é quando esse usa de conceitos matemáticos para resolver problemas determinísticos.

5 Computação evolutiva

1. A computação evolutiva vem sendo usada desde muitos anos atrás, no entanto, somente com os avanços tecnológicos é que passou a ser utilizada em muitas aplicações. Nesse sentido, a computação evolucionária visa reproduzir o comportamento de determinada espécie matematicamente e computacionalmente para a resolução de um problema de modo mais rápido e com maior desempenho.
2. Os algoritmos genéticos visam encontrar determinadas soluções quando o problema tem um espaço de busca grande, em que soluções não ótimas podem ser encontradas em menor espaço de tempo.
3. A mutação apresenta como principal ideia criar uma variabilidade extra na população, mas sem destruir o progresso já obtido com a busca.
4. Vários são os algoritmos evolutivos existentes, mas dentre eles podemos destacar os algoritmos da evolução diferencial e o algoritmo de enxame de partículas. Aqui vale ressaltar que, como os algoritmos evolucionários tentam imitar o comportamento de determinada espécie, sempre novos comportamentos são criados e modelos matematicamente são implementados, de modo a serem testados em determinado problema.



Código Logístico

59688

Fundação Biblioteca Nacional
ISBN 978-85-387-6711-4

9 788538 767114