



Representação de conhecimento, raciocínio automático e aprendizado

Apresentação de conceitos e aplicações da representação de conhecimento para construção de algoritmos especialistas, bem como de conceitos sobre redes neurais artificiais, com exemplos práticos empregando a linguagem de programação Prolog.

Prof. Sérgio Assunção Monteiro

Propósito

Compreender as formas de representação do conhecimento, raciocínio automático e aprendizado permite a criação de sistemas especialistas para desenvolver aplicações que tenham escalabilidade pela representação do conhecimento.

Preparação

Para executar nossos exemplos, você precisará fazer o download do software SWI-Prolog.

Objetivos

- Descrever conceitos de lógica para inteligência artificial, inferência e resolução na solução de problemas
- Reconhecer regras e sistemas especialistas.
- Empregar conceitos de redes neurais artificiais.
- Analisar algoritmos de treinamento de redes neurais artificiais.

Introdução

Os sistemas estão presentes nas diversas atividades que precisamos realizar, tais como transações financeiras, assistir a conteúdo on-line e fazer consultas na internet, por exemplo. Uma das categorias de sistemas são os chamados **sistemas especialistas**.

Os sistemas especialistas são muito úteis para apoiar na tomada de decisão com a proposta de recomendações a partir de uma base de conhecimento. Por isso, vamos aprender alguns dos principais aspectos para representar o conhecimento e, assim, desenvolver os alicerces necessários para arquitetar sistemas especialistas.

Conceitos fundamentais: conhecimento, representação e raciocínio

O conhecimento está relacionado ao conjunto de informações sobre um domínio — como mercado financeiro, medicina, meteorologia, entre outros — que pode ser usado para resolver problemas naquele próprio domínio. Alguns exemplos sobre conhecimento são:

Diagnósticos

de doenças a partir de exames laboratoriais.

Estimativas

sobre o movimento de ações do mercado financeiro a partir da análise histórica.

Previsões

de vendas considerando fatores históricos.

Dentro do contexto da inteligência artificial, a nossa preocupação é representar o conhecimento de modo a utilizá-lo em programas, ou seja, precisamos de uma forma objetiva de armazenar o conhecimento, acessá-lo e, em muitos casos, atualizá-lo com a inclusão de informações ou alterações das que já existem. Portanto, em primeiro lugar, precisamos identificar o que deve ser representado. Entre os principais itens que precisamos representar estão:

Objeto

Refere-se a um item que pode ser identificado explicitamente dentro de um contexto com características que, posteriormente, serão úteis para referenciá-lo. Por exemplo, os carros têm quatro pneus, o violão tem seis cordas, o computador tem tela e teclado.

Eventos

Estão relacionados a ocorrências dentro de determinado domínio. As oscilações de uma ação ao longo do tempo, a evolução de uma doença e expressões faciais após uma vitória são alguns exemplos de eventos.

Desempenho

É a descrição do processo para realizar uma tarefa, ou seja, o comportamento que é resultado do uso do conhecimento sobre como fazer as coisas. Por exemplo, tocar violão é um comportamento que envolve conhecimento sobre como fazer as coisas.

Metaconhecimento

É uma descrição sobre o conhecimento em si. De outra forma, é o conhecimento do que sabemos.

Fatos

É o conhecimento explícito sobre verdades do mundo real.

Base de conhecimento

Refere-se a um grupo de informações sobre qualquer domínio. Por exemplo, uma base de conhecimento sobre tocar violão.

Precisamos representar na inteligência artificial os seguintes tipos de conhecimento (Russel, 2004):

Metaconhecimento

Define o conhecimento sobre outros tipos de conhecimento.

Conhecimento processual

Também conhecido como conhecimento imperativo, trata da habilidade de saber fazer algo e inclui regras, estratégias, procedimentos etc.

Conhecimento estrutural

É um conhecimento aplicado na resolução de problemas que descreve a relação entre conceitos e objetos.

Conhecimento declarativo

É a área do conhecimento responsável por armazenar informações de fatos, tais como eventos ou processos. É descrição de saber algo sobre alguém ou alguma coisa (do inglês know-what).

Conhecimento heurístico

Representa algum conhecimento especializado. Vale a pena destacar a importância deste conhecimento para aplicações de inteligência artificial, uma vez que incorpora experiências e técnicas aplicadas com sucesso na prática em métodos que são muito úteis para auxiliar na resolução de problemas semelhantes. Normalmente, é utilizado em sistemas especialistas.

Após abordar aspectos sobre o conhecimento, precisamos tratar do uso dele para resolver tarefas. É aqui que precisamos tratar do raciocínio.

Então vem a pergunta: o que é raciocínio?

No campo da inteligência artificial, de modo geral, o raciocínio é a manipulação formal dos símbolos que representam um conjunto de proposições verdadeiras para produzir representações de novas proposições. Vamos detalhar um pouco mais sobre isso.

Primeiro, definimos o que é uma proposição. Trata-se de uma declaração que só pode ser verdadeira ou falsa. Nunca pode ser duas coisas ao mesmo tempo ou ter uma terceira opção. É uma técnica da lógica matemática para fazer a representação do conhecimento.



Exemplo

Proposições: O Natal é comemorado no dia 25 de dezembro (proposição verdadeira). $3 + 3 = 7$ (proposição falsa). 5 é um número primo (proposição verdadeira). Cada uma dessas afirmações é uma proposição, pois é verdadeira ou falsa. Agora, vamos observar um exemplo de raciocínio: Pedro estudou com Maria. Maria viajou. Após manipular as sentenças, produzimos a frase: Alguém com quem Pedro estudou viajou.

A **inferência lógica** tem esse nome, pois a sentença final representa uma conclusão lógica das proposições iniciais, como vamos tratar com mais detalhes um pouco mais adiante.

Do que vimos até agora, percebemos como é essencial termos uma forma de fazer representação do conhecimento e estruturar o raciocínio para que um computador possa compreender as informações que são fornecidas e, em seguida, utilizar esse conhecimento para resolver problemas, que podem ser complexos, como o processamento de linguagem natural.

Portanto, a representação do conhecimento em inteligência artificial não é apenas o armazenamento de informações em um banco de dados, mas o desenvolvimento da habilidade da máquina em incorporar esse conhecimento e se comportar de maneira inteligente, que é o que chamamos de **aprendizado de máquina**.

Assim, concluímos que uma representação tem alguns objetivos, entre os quais destacamos:

- Disponibilizar recursos para expressar o conhecimento necessário e resolver um problema.

- Facilitar o desenvolvimento de programas através de uma sintaxe próxima da definição dos problemas.
- Utilizar técnicas para otimizar a execução dos programas sob os pontos de vista de gerenciamento dos recursos computacionais, tais como tempo e memória.
- Possibilitar a exploração das características específicas do problema, de modo a tornar a execução mais eficiente.

Conceitos de lógica para IA

Uma das formas de representar o conhecimento em inteligência artificial é por meio da **representação lógica**. Basicamente, é uma linguagem caracterizada por possuir regras para manipular proposições e que não tem ambiguidade na sua representação.

A representação lógica obtém uma conclusão com base em várias condições e consiste em regras sintáticas e semânticas, para que seja possível obter uma conclusão. Portanto, temos que sintaxe e semântica têm as seguintes características:

Sintaxe

São regras para construção de sentenças válidas na lógica, ou seja, definem o que é uma estrutura sentencial válida. Além disso, a sintaxe é responsável por definir:

- **Símbolos lógicos:** Verdade e falso
- **Símbolos proposicionais:** Por exemplo, as letras minúsculas “p” e “q”
- **Símbolo de igualdade:** “=”
- **Conectivos proposicionais:** \sim (NÃO), \wedge (E), \rightarrow (Implica ou “Se, Então”), \leftrightarrow (equivalência ou “se e somente se”)
- **Quantificadores:** \exists (existencial ou “existe pelo menos um”), \forall (universal ou “para todo”)
- **Símbolos de pontuação:** “(” (abre parênteses) e “)” (fecha parênteses)

Semântica

São as regras pelas quais podemos interpretar a frase na lógica. Por exemplo, seja a sentença proposicional: $p = \sim q$. Se o valor da proposição “q” for falso, então, o valor da proposição “p” será verdadeiro.

A representação lógica pode ser categorizada principalmente de duas formas:

Lógica proposicional

Também chamada de lógica sentencial. Faz a representação dos fatos do mundo real por meio de sentenças chamadas de proposições. Apresenta como vantagens que a sua representação nos permite desenvolver raciocínio lógico e é a base para as linguagens de programação. Como desvantagens, tem algumas restrições sobre algumas representações lógicas e que são difíceis de trabalhar. Além disso, a inferência, em alguns casos, não é eficiente.



Lógica de predicado

Também conhecida como lógica de primeira ordem. Permite o uso de variáveis e de quantificadores. Tem como vantagens a representação de objetos e das relações entre eles através de funções e predicados.

Inferência e resolução na solução de problemas de lógica proposicional

Também chamada de lógica booleana e lógica sentencial, ela opera com proposições que podem assumir os valores **verdade** ou **falso**. Usamos variáveis simbólicas, normalmente minúsculas, tais como as letras **p**, **q** e **r** para fazer a representação das sentenças lógicas. Ela possui três princípios básicos:

Princípio da identidade

Uma proposição que é verdadeira é verdadeira; uma proposição que tem valor falso é falsa.

Princípio da não contradição

Nenhuma proposição pode ser verdadeira e falsa ao mesmo tempo.

Princípio do terceiro excluído

Uma proposição só tem exclusivamente uma possibilidade: ou é verdadeira, ou é falsa.

A lógica proposicional relaciona as proposições através de conectivos lógicos que também são chamados de operadores lógicos. Os conectivos lógicos são:

- E
- OU
- NEGAÇÃO
- CONDICIONAL
- BICONDICIONAL

1

E

Chamado de conjunção, normalmente, é representado pelo símbolo \wedge . É um operador binário, ou seja, é aplicado sobre duas proposições. Ele só retorna o valor verdade quando os valores lógicos das suas proposições de entrada são verdadeiros.

2 Ou

Chamado de disjunção, normalmente, é representado pelo símbolo \vee . É um operador binário. Só retorna o valor falso quando os valores lógicos das suas proposições de entrada são ambos falsos. Para os demais casos, retorna verdade.

3

Negação

Normalmente, é representado pelo símbolo \sim . É um operador unitário que significa que é aplicado para uma entrada. Inverte o valor lógico da proposição, ou seja, se o valor de proposição for verdade, retorna falso; se o valor lógico for falso, retorna verdade.

4

Condicional

Normalmente, é representado pelo símbolo \rightarrow . É um operador binário.

Por exemplo, seja a proposição $p \rightarrow q$. Ele só retorna o valor falso se o valor lógico da proposição p for verdade e o valor lógico da proposição q for falso. Para os demais casos, retorna verdade.

5

Bicondicional

Negação

Chamado de disjunção, normalmente, é representado pelo símbolo \leftrightarrow . É um operador binário.

Por exemplo, seja a proposição $p \leftrightarrow q$. Ele só retorna o valor **verdade** se os valores lógicos das proposições p e q forem verdade. Para os demais casos, ele retorna falso.

Um instrumento muito útil para visualizar os valores lógicos das proposições quando fazemos a aplicações dos operadores lógicos é a **tabela verdade**. Na tabela a seguir, podemos ver um resumo de como os operadores lógicos funcionam na tabela verdade.

p	q	$p \wedge q$	$p \vee q$	$\sim p$	$p \rightarrow q$	$p \leftrightarrow q$
F	F	F	F	V	V	V
F	V	F	V	V	V	F
V	F	F	V	F	F	F
V	V	V	V	F	V	V

Tabela: Tabela verdade dos operadores lógicos
Elaborada por: Sérgio Assunção Monteiro

Como os valores que as proposições podem assumir são verdade ou falso, podemos calcular exatamente a quantidade de linhas da tabela verdade que é dado por 2^n , onde n representa a quantidade de proposições simples de uma sentença.

Outro ponto importante a respeito dos operadores é a prioridade deles. Na próxima tabela, mostramos a ordem de precedência dos operadores.

Precedência	Operador
1	Parênteses
2	Negação (\sim)
3	E (\wedge)
4	OU (\vee)
5	Condicional (\rightarrow)
6	Bicondicional (\leftrightarrow)

Tabela: Ordem de precedências dos operadores
Elaborada por: Sérgio Assunção Monteiro

Agora, vamos resolver um problema-exemplo.

Dada a sentença $p \wedge q \rightarrow \sim q$, vamos resolvê-la pela tabela verdade.

Para encontrar a solução, nosso primeiro passo é entender as prioridades da sentença. Analisando, vemos que primeiro, precisamos resolver $\sim q$, em seguida, temos que resolver $p \wedge q$ e, por fim, temos que resolver $p \wedge q \rightarrow \sim q$. Então, vamos construir a tabela verdade seguindo essa sequência.

Na tabela a seguir, temos a solução.

P	q	$\sim q$	$p \wedge q$	$p \wedge q \rightarrow \sim q$
F	F	V	F	V
F	V	F	F	V
V	F	V	F	V
V	V	F	V	F

Tabela: Solução para o exemplo
Elaborada por: Sérgio Assunção Monteiro

Interpretando o resultado, vemos que o valor lógico da sentença proposicional $p \wedge q \rightarrow \sim q$ é **verdade**, quando o valor lógico de p é verdade e o valor lógico de q é falso, por exemplo.

Quando o valor lógico de uma sentença é verdadeiro para quaisquer valores lógicos das proposições de entrada, ela é chamada **tautologia**.

Veja um exemplo.

Vamos resolver pela tabela verdade a sentença $p \wedge q \rightarrow q \wedge p$.

p	q	$p \wedge q$	$q \wedge p$	$p \wedge q \rightarrow q \wedge p$
F	F	F	F	V
F	V	F	F	V
V	F	F	F	V
V	V	V	V	V

Tabela: Tabela verdade dos operadores lógicos
Elaborada por: Sérgio Assunção Monteiro

Quando o valor lógico de uma sentença proposicional é sempre falso, independentemente dos valores lógicos das proposições de entrada, então ela é chamada de **contradição**.

Veja um exemplo.

Vamos resolver pela tabela verdade a sentença $(p \wedge q) \wedge \sim (p \wedge q)$.

p	q	$p \wedge q$	$\sim (p \wedge q)$	$(p \wedge q) \wedge \sim (p \wedge q)$
F	F	F	V	F
F	V	F	V	F
V	F	F	V	F
V	V	V	F	F

Tabela: Solução do exemplo de contradição
Elaborada por: Sérgio Assunção Monteiro

E quando os valores lógicos de uma sentença não são homogêneos, ela é chamada de **contingência**.

Veja um exemplo.

Vamos resolver pela tabela verdade a sentença $p \vee q \rightarrow p \wedge q$.

p	q	$p \vee q$	$p \wedge q$	$p \vee q \rightarrow p \wedge q$
F	F	F	F	V
F	V	V	F	F
V	F	V	F	F

V	V	V	V	V
---	---	---	---	---

Tabela: Solução do exemplo de contingência
Elaborada por: Sérgio Assunção Monteiro



Atenção

A tabela verdade é um recurso muito útil para resolvermos problemas com poucas proposições, mas é inadequada para resolver problemas de um modo geral. Para resolver problemas mais complexos, utilizamos as regras de inferência.

Inferência é aplicação de um conjunto de regras que já sabemos que são verdadeiras para afirmar se uma sentença proposicional é ou não verdadeira.

Na próxima tabela, apresentaremos algumas das principais regras de inferência.

De	Podemos inferir	Nome da regra
$p, p \rightarrow q$	q	Modus ponens (MP)
$\sim q, p \rightarrow q$	$\sim p$	Modus tollens (MT)
$\sim q, p \vee q$	p	Silogismo disjuntivo (SD)
$p \rightarrow q, q \rightarrow r$	$p \rightarrow r$	Silogismo hipotético (SH)
$p \wedge q$	p	Simplificação conjuntiva (SC)
p	$p \vee q$	Amplificação disjuntiva (AD)

Tabela: Solução do exemplo de contingência
Elaborada por: Sérgio Assunção Monteiro

Teoria e prática

Vamos resolver um exemplo para mostrar como funciona o processo de inferência em problemas da lógica proposicional.

Dada a sentença proposicional $p, p \rightarrow q, p \rightarrow s \vdash q \wedge s$, vamos resolvê-la pelo método da inferência.

Chave de resposta

O símbolo \vdash significa: “concluimos que” e as vírgulas são usadas como pontuação. Ou seja, temos como hipóteses para a nossa sentença as proposições $p, p \rightarrow q, p \rightarrow s$ e a conclusão — que precisamos demonstrar — é $q \wedge s$. Então, passamos a resolver o problema em uma sequência de passos:

1. p , (Hipótese)

2. $p \rightarrow q$ (Hipótese)

3. $p \rightarrow s$, (Hipótese)

4. q , (1 e 2 MP)

5. s , (1, 3 MP)

6. $q \wedge s$, (4 e 5)

7. cqd

Detalhando um pouco mais a nossa solução: nos três primeiros passos, usamos as hipóteses fornecidas pelo problema. No passo 4, deduzimos “ q ” usando os passos 1 e 2 por Modus ponens. No passo 5, usamos os passos 1 e 3 por Modus ponens e obtivemos s . No passo 6, usamos o fato que tínhamos à disposição “ q ” e “ s ”. Por fim, escrevemos cqd que significa “como queríamos demonstrar”.

Inferência e resolução na solução de problemas de lógica de predicados

A lógica de primeira ordem, ou dos predicados, possibilita que possamos representar sentenças mais complexas do que na lógica proposicional. Na lógica de primeira ordem, as declarações são divididas em duas partes:

Objeto

É a parte principal da declaração.

Predicado

É uma relação entre objetos no mesmo contexto ou domínio.

A seguir, mostramos alguns exemplos de predicados:

Maria é irmã de João. Podemos representar por: $Irmã(Maria, João)$.

Toda casa é um objeto físico. Na lógica de primeira ordem, escrevemos: $\forall x (casa(x) \rightarrow objeto_fisico(x))$.

Alguns objetos físicos são casas. Na lógica de primeira ordem, escrevemos: $\exists x (\text{objeto_fisico}(x) \wedge \text{casa}(x))$.

No segundo exemplo, utilizamos o quantificador universal, representado pelo símbolo \forall , e uma variável x , que nos dá mais flexibilidade de representação. Portanto, a frase pode ser entendida como: "Para qualquer coisa que seja casa, então, ela é também um objeto físico."

No terceiro exemplo, usamos o quantificador existencial, que é representado pelo símbolo \exists . A frase pode ser entendida como: "Existem algumas coisas que são casa e objeto físico."

Em resumo, temos as seguintes considerações sobre os quantificadores:

Quantificador existencial

Vamos considerar que $p(x)$ é uma proposição sobre o universo U . Então, quando escrevemos $\exists x p(x)$, devemos entender como: "Existe pelo menos um valor no universo da variável x tal que $p(x)$ é verdadeiro."

Existem várias maneiras equivalentes de escrever uma proposição, com um quantificador existencial. Por exemplo, todas estas formas são equivalentes: $(\exists x \in A) p(x)$, ou $\exists x \in A$ tal que $p(x)$, ou $(\exists x) p(x)$, ou $p(x)$ é verdadeiro para algum $x \in A$.

Quantificador universal

Vamos considerar que $p(x)$ é uma proposição sobre o universo U . Então, quando escrevemos $\forall x, p(x)$, devemos entender como: "Para todo $x \in U$, $p(x)$ é verdadeiro."

Semelhantemente ao quantificador existencial, podemos escrever uma proposição de muitas maneiras equivalentes. Por exemplo, todas as seguintes formas são equivalentes: $\forall x \in A, p(x)$, ou $p(x)$, $\forall x \in A$, ou $\forall x, p(x)$, ou $p(x)$ é verdadeiro para todo $x \in A$.

Na tabela a seguir, apresentamos uma lista de axiomas válidos para lógica de primeira ordem, necessários para a resolução de problemas (Gersting, 2001).

1	$P \rightarrow (Q \rightarrow P)$
2	$[P \rightarrow (Q \rightarrow R)] \rightarrow [(P \rightarrow Q) \rightarrow (P \rightarrow R)]$
3	$(\sim Q \rightarrow \sim P) \rightarrow (P \rightarrow Q)$
4	$(\forall x)[P(x) \rightarrow Q(x)] \rightarrow [(\forall x)P(x) \rightarrow (\forall x)Q(x)]$
5	$(\forall x)P(x) \rightarrow P(a)$, onde a é uma constante
6	$(\exists x)P(x) \rightarrow P(t)$, onde t é uma constante
7	$P(a) \rightarrow (\exists x)P(x)$, onde a é uma constante e x não ocorre em $P(a)$
8	$\sim(\exists x)P(x) \rightarrow (\forall x)\sim P(x)$

Tabela: Axiomas para lógica de predicados.

Adaptada de: Gersting (2001, p. 27).

As regras de inferência para lógica dos predicados são dadas por:

Modus ponens

Q pode ser inferida de P e $P \rightarrow Q$.

Generalização

$(\forall x)Q$ pode ser inferida de Q desde que **não** ocorra uma das seguintes situações:

- Q tenha sido deduzida de qualquer hipótese na qual x seja uma variável livre;
- Q tenha sido deduzida pelo uso do axioma 6 de uma fórmula bem formada da forma na qual x seja uma variável livre.

Como exemplo de solução de um problema, vamos demonstrar a validade do teorema $(\forall x)[P(x) \wedge Q(x)] \rightarrow (\forall x)P(x)$.

1. $(\forall x)[P(x) \wedge Q(x)]$, (Hipótese)
2. $P(x) \wedge Q(x)$, (1, axioma 5, Modus ponens)
3. $P(x)$, (2, simplificação conjuntiva)
4. $(\forall x)P(x)$, (3, generalização)
5. **cqd**

Vamos detalhar um pouco mais a solução:

Passo 1

Utilizamos a hipótese dada no problema, ou seja, $(\forall x)[P(x) \wedge Q(x)]$.

Passo 2

Agora, precisamos manipular a hipótese para chegarmos à conclusão do teorema. No caso queremos obter $(\forall x)P(x)$. Para isso, aplicamos o axioma 5, que trata da especialização de um resultado e aplicamos Modus ponens.

Passo 3

Aplicamos uma simplificação conjuntiva em $P(x) \wedge Q(x)$ e obtivemos $P(x)$.

Passo 4

Aplicamos a generalização de $P(x)$, pois ela satisfaz as regras de inferência.

Para finalizar

Terminamos a demonstração e escrevemos **cqd**.

Conceitos de lógica para IA, inferência e resolução na solução de problemas

No vídeo a seguir, o especialista fala sobre conceitos e práticas da lógica de primeira ordem.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Verificando o aprendizado

1. A inteligência artificial tem diversas aplicações práticas. Entre os seus objetivos, visa resolver problemas, a partir da execução de algoritmos sobre bases de conhecimento. Selecione, a seguir, a alternativa correta sobre a representação do conhecimento.

A

Quando é necessário fazer uma autorreferência à base de conhecimento, trata-se da representação de fatos.

B

A descrição de como realizar uma ação é chamada de desempenho e é um dos itens necessários de serem representados.

C

Os elementos sobre os quais as ações são realizadas e, eles mesmos, realizam tarefas são chamados de fatos.

D

Uma das principais vantagens da representação do conhecimento é consolidar uma base de regras estáveis que raramente sofrerão mudanças.

E

A representação de eventos é uma das etapas necessárias para que seja possível descrever a própria base de conhecimento.



A alternativa B está correta.

A representação do conhecimento precisa descrever explicitamente todos os elementos necessários para realizar uma ação. É necessário destacar objetos, o modo como esses objetos se relacionam entre si estaticamente e ao longo do tempo, e os procedimentos para realizar uma ação, sendo que estes últimos são chamados de desempenho.

2. A inteligência artificial possui diversos conceitos relacionados à representação do conhecimento. Devido à objetividade desses conceitos, é possível identificar se o desenvolvimento de uma solução está ou não correta. Selecione, a seguir, a alternativa correta sobre os conceitos de inteligência artificial e de representação do conhecimento.

A

A análise sintática está relacionada ao significado das sentenças lógicas.

B

Símbolos, como parênteses e vírgulas, aplicados em sentenças lógicas são verificados pela análise semântica.

C

A análise semântica cobre aspectos como se os operadores lógicos estivessem escritos corretamente.

D

O significado de uma expressão lógica abrange uma interpretação subjetiva.

E

Quando a nomenclatura de determinado símbolo não segue as regras da representação lógica, trata-se de um erro sintático.



A alternativa E está correta.

Os conceitos de sintaxe e semântica ajudam na construção de linguagens lógicas e, também, de linguagens de programação. A análise sintática é responsável por verificar, entre outras coisas, se as estruturas de determinada entrada, como sentenças lógicas, obedecem às regras da linguagem. Dessa forma, verificar se os símbolos obedecem ou não às regras de nomeação de uma linguagem é atribuição da análise sintática.

Sistemas especialistas

A aquisição de conhecimento não é uma tarefa simples. O exato momento em que você está lendo este texto, fazendo interpretações e analogias, é um exemplo concreto de como esse exercício exige tempo e dedicação.

Quando alguém adquire conhecimento suficiente sobre um assunto de modo que suas opiniões e conclusões sejam úteis para resolver problemas de maneira eficaz, essa pessoa é chamada de **especialista**. Existem muitas áreas em que o papel do especialista é fundamental. Por exemplo:

Diagnóstico de doenças

a partir de análises clínica e de laboratório.

Recomendações de carteiras

de investimentos financeiros com a análise do histórico e do contexto econômico-político.

Constatar a veracidade

de assinaturas e autorias de obras de arte etc.

A lista de exemplos de aplicações em que o papel do especialista é muito importante é difícil de ser limitada, pois, nas mais diversas áreas da sociedade, existem aplicações em que o conhecimento sobre determinados assuntos é essencial. Como podemos perceber, formar um especialista não é algo simples.

Além disso, existe outra questão: é muito difícil proporcionar escala para uma solução que dependa de um especialista, pois, apesar da profundidade do conhecimento, a amplitude de atuação dele é limitada.

Como o especialista tem um papel importante para garantir a qualidade de uma solução, mas sua atuação é limitada pelo volume de tarefas que pode realizar, foram desenvolvidos os sistemas especialistas que incorporam o conhecimento em determinada área, por meio de métodos que estão organizados em algoritmos e bancos de dados. A ideia é que um sistema especialista tenha um comportamento semelhante ao de um especialista humano para resolver problemas para os quais possui domínio.

Características e componentes de sistemas especialistas

Um sistema especialista normalmente é acionado quando um usuário fornece dados sobre determinadas ocorrências em relação a um objeto de análise.

Por exemplo, vamos tratar de um sistema especialista em análise preditiva de equipamentos automotores. Um usuário do sistema pode fornecer dados de variação de temperatura e pressão no motor de determinado equipamento, obtidos após uma inspeção. O sistema, então, fará a análise desses dados, que podem ser sintomas de um problema, e, então, oferecerá sugestões ou recomendações, conforme sua base de conhecimento. Além disso, o sistema pode ser preparado para justificar como chegou às suas sugestões.

Portanto, um sistema especialista deve ter características que aumentem a confiança nas suas sugestões, entre as quais, destacamos as seguintes (Russel, 2004):

Alto nível de especialização

Significa que o sistema precisa incorporar a base de conhecimento disponível, de modo a tratar de um assunto com profundidade, ou seja, mapear objetos, relações, processos, exceções e todas as outras informações sobre o que o sistema trata.

Reação no momento oportuno

Logo que receber os dados, um sistema especialista deve processá-los e fornecer resultados em um tempo que suas sugestões sejam úteis para ser analisadas e aplicadas.

Alta confiabilidade

Significa que o sistema não deve falhar ou pelo menos que o grau de confiança de suas sugestões é aceitável.

Flexibilidade

O conhecimento é dinâmico e, portanto, sua evolução é algo esperado. Quando o sistema faz sugestões, é possível que elas possam ser aperfeiçoadas por um especialista. Após um processo bem-estruturado, esses aperfeiçoamentos podem ser incorporados à base de conhecimento do sistema.

Explicação dos resultados

Quando o sistema propõe sugestões, elas são resultantes de um processo de análise que, muitas vezes, precisam ser acessíveis para o usuário do sistema.

Resolução de problemas

É o principal objetivo dos sistemas especialistas. Devem produzir sugestões e recomendações utilizando a base de conhecimento quando o usuário fornece os dados para serem analisados. Em muitos contextos, esses dados são sintomas de um problema.

Além dessas características, o sistema especialista deve ser composto pelos seguintes componentes:

Interface de usuário

É por meio desse componente que o usuário interage com o sistema a partir da solicitação de consultas e análise das sugestões propostas. A interface de usuário direciona como as consultas devem ser feitas, de modo a serem transformadas para um formato que o mecanismo de inferência possa compreender e processar. Ao final do processamento, as sugestões do sistema são exibidas para o usuário para que possa interagir com elas e dar prosseguimento à sua análise. Preocupações com facilidade de uso, ser acessível em diversos ambientes — redes locais e na internet — e segurança da informação devem ser tratadas explicitamente.

Motor de inferência

É o mecanismo que contém as regras para resolver os problemas que fazem parte do escopo do sistema. Ele analisa a consulta do usuário e, com base nela, seleciona as regras que devem ser aplicadas para resolver o problema. Portanto, o raciocínio sobre o uso das informações da base de conhecimento para resolver problemas é realizado por esse componente.

Base de conhecimento

É repositório de fatos, ou seja, é nele que estão todos os aspectos conhecidos e armazenados a respeito da solução dos problemas que fazem parte do escopo do sistema. Ele pode ser formado por diversas fontes de informação, como as sugestões de vários especialistas, manuais técnicos e retornos fornecidos pelos usuários do sistema, a respeito da efetividade da aplicação das sugestões.

Como construir um sistema especialista

Agora que entendemos as características e os componentes de um sistema especialista, precisamos descrever o processo de criação do sistema.

Determinar características do problema

O primeiro aspecto que devemos tratar é determinar as características dos problemas que são objeto de análise do sistema, ou seja, qual é o escopo para o qual o sistema é projetado. Deixar essa etapa clara é fundamental, pois ela é a base para fazer testes e validações de resultados.

Criar fundamentos do sistema

Na segunda etapa, é necessário criar os fundamentos do sistema elaborados em um trabalho conjunto pelo engenheiro do conhecimento e o especialista no domínio. O engenheiro do conhecimento é responsável, entre outras atividades, por especificar qual o formato da entrada de dados para os usuários realizarem consultas, como o conhecimento deve ser estruturado na base de dados do sistema e projetar um mecanismo de inferência que possa acessar o conhecimento adequado para resolver um problema.

Mapear sintomas

Em seguida, entra o papel do especialista do conhecimento. Ele fará o mapeamento de sintomas com recomendações. Esse processo é bastante complexo, pois transferir a experiência adquirida para um sistema precisa de um direcionamento claro e de um processo que possa ser validado posteriormente, uma vez que esse conhecimento, normalmente, tem aspectos de incerteza.

A base de conhecimento de um sistema especialista é caracterizada pela experiência e, nesse sentido, ela é intrinsecamente sujeita a falhas. Em especial, porque a experiência é obtida em determinados contextos que podem influenciar as análises das sugestões, mesmo que estejam na mesma área de domínio.

Por exemplo, um sistema pode ser construído para tratar do diagnóstico de equipamentos automotores aplicados à mineração. A experiência do especialista, nesse exemplo, é contextualizada em um ambiente de temperatura que varia entre 25 e 35 graus centígrados, aproximadamente. Agora, a empresa deseja aplicar o sistema para analisar os mesmos equipamentos, mas em um ambiente de temperatura que varia de 0 a 5 graus centígrados.

Nesse caso, o sistema de recomendações ainda pode ser aplicado?

A resposta depende de vários fatores, mas, de forma objetiva, os resultados do sistema precisam ser analisados constantemente. Esse exemplo é uma demonstração de como o conhecimento é dinâmico e, por isso, é necessário haver uma forma de adquirir e atualizar o conhecimento do sistema.

Os sistemas especialistas podem ser aplicados para diversas áreas, tais como gestão de informação, diagnósticos médicos, avaliação de desempenho dos funcionários, análise de empréstimo, detecção de vírus de computadores, entre tantas outras possibilidades. Ainda assim, há limitações que um sistema especialista pode ter:

- Ser incapaz de produzir uma resposta criativa em uma situação para a qual não foi projetado.
- Conduzir a decisões erradas devido a erros na base de conhecimento.
- Manter um sistema especialista é um processo complexo e, normalmente, caro.
- Resolver teoremas por meio de processos dedutivos, pois não se trata apenas de aplicar regras, mas, em muitos casos, de analisar características que só podem ser exploradas por um processo criativo.

Alguns sistemas especialistas utilizam um método de raciocínio que se assemelha ao raciocínio humano, chamado de **lógica fuzzy**, também chamado de **lógica difusa**. Essa lógica permite que o sistema faça uso de inferências aproximadas e de dados incompletos ou ambíguos (**dados fuzzy**) para tratar com a incerteza em alguns sistemas especialistas.

Os sistemas especialistas que utilizam lógica fuzzy são mais flexíveis e, até certo ponto, criativos para resolver problemas. Alguns dos contextos em que esses sistemas são utilizados são os controles de processos de fabricação.

Regras e sistemas especialistas

As regras são uma das formas mais utilizadas e simples para representar o conhecimento em um sistema especialista. Inclusive esses sistemas são chamados, não por acaso, de **sistemas baseados em regras**.

O conhecimento, portanto, é especificado por meio de fatos e regras que têm o formato **SE-ENTÃO**. De posse desse conhecimento, o motor de inferência do sistema aplica o Modus ponens para obter novas conclusões do conhecimento existente.

A estrutura SE-ENTÃO de uma regra é chamada de regra de produção, em que a parte SE é chamada de **condição** ou **premissa** e a parte do ENTÃO é chamada de **ação** ou **conclusão**.

Regras de produção são capazes de representar o conhecimento sobre uma área específica por meio de uma codificação direta, e a parte da regra que se refere ao SE faz a especificação de uma condição e a parte do ENTÃO trata da ação, ou ações, para tratar o problema.

Vamos ver um exemplo simples para verificar se um animal é um cachorro, gato ou um pato:

Regra 1

SE o animal tiver pelos E latir, ENTÃO o animal é um cachorro.

Regra 2

SE o animal tiver pelos E miar, ENTÃO o animal é um gato.

Regra 3

SE o animal tiver penas E grasnar, ENTÃO o animal é um pato.

Para o nosso exemplo, se receber uma consulta em que o animal tem pelos e late, o sistema vai responder que o animal é um cachorro, pois a consulta se encaixa perfeitamente na regra 1.

Podemos implementar sistemas especialistas em muitas linguagens de programação. Uma linguagem bastante utilizada para essa finalidade é a **Prolog**, acrônimo de Programming in Logic, ou seja, traduzindo, **Programação em Lógica**.

Vamos implementar o exemplo de classificar um animal em Prolog. O código é dado a seguir:

```
prolog

animal(cachorro) :- eh_fato_que('tem pelo'), eh_fato_que('late').
animal(gato) :- eh_fato_que('tem pelo'), eh_fato_que('mia').
animal(pato) :- eh_fato_que('tem penas'), eh_fato_que('grasna').

eh_fato_que(Q) :-
    format("~w?\n", [Q]),
    read(sim).
```

Para poder executar o código anterior, utilizamos o programa **SWI-Prolog**, que pode ser baixado no site do SWI-Prolog. As primeiras três linhas do código correspondem às regras 1, 2 e 3, respectivamente. O símbolo :- significa o condicional **SE**. Por exemplo, a linha:

```
prolog

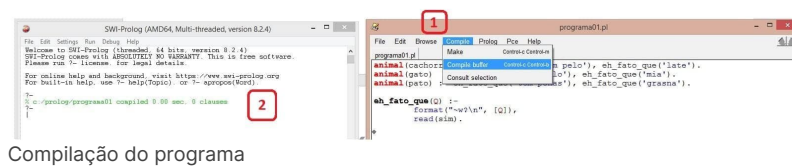
animal(cachorro) :- eh_fato_que('tem pelo'), eh_fato_que('late').
```

Pode ser lida como: “SE é um fato que tem pelo E late, ENTÃO o animal é um cachorro.” Já a sequência:

```
prolog
eh_fato_que(Q) :-
    format("~w?\n", [Q]),
    read(sim).
```

Recebe uma variável — no caso, chamada de **Q** — como entrada e a transforma para uma String com uma interrogação no final.

Para rodar esse programa, precisamos salvar o arquivo. Vamos chamá-lo de **programa01.pl**. Em seguida, precisamos compilar o programa, conforme a imagem a seguir.



Após compilar o programa, podemos fazer a consulta ao sistema, escrevendo na linha de comando a instrução: **animal(X)**.

Em seguida, o sistema vai nos perguntar sobre os fatos que sabemos a respeito do animal X. Na imagem a seguir, mostramos uma sequência de interações com o sistema.



Vamos detalhar a saída:

Passo 1

Primeiro, escrevemos “animal(X).” — sem as aspas — e pressionamos a tecla Enter. Não devemos esquecer de digitar o “.”, pois faz parte da sintaxe do Prolog.



Passo 2

Depois, o motor de inferência começa a nos consultar sobre o animal. A primeira pergunta que ele nos faz é: “tem pelo?” No exemplo, escrevemos que “sim”. Então, ele continua a fazer as perguntas até concluir que o animal se trata de um cachorro.

Nesse exemplo, mostramos como o mecanismo de inferência faz o percurso através das regras que fornecemos até chegar a uma conclusão. Existem dois tipos de mecanismo de inferência:

Encadeamento progressivo

Do inglês **forward chaining**. É uma estratégia baseada em dados. Basicamente, os fatos são fornecidos para o motor de inferência e ele os combina para chegar a uma conclusão. No exemplo da classificação de animais que apresentamos, adotamos essa estratégia, pois tivemos que informar os dados a respeito do animal para que o sistema chegasse a uma conclusão.



Encadeamento reverso

Do inglês **backward chaining**. O mecanismo de inferência tem como ponto de partida a conclusão assumida e, em seguida, retorna até chegar aos fatos que dão base para a conclusão. Novamente, no caso do nosso código de exemplo, a aplicação dessa técnica ocorre informando que o animal é um cachorro. Agora, o motor de inferência precisa retroagir para obter os fatos que justificam essa conclusão.

Regras e sistemas especialistas

No vídeo a seguir, o especialista fala sobre o conhecimento baseado em regras e implementados em Prolog.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Verificando o aprendizado

1. Os sistemas especialistas são ferramentas úteis em diversas áreas. Isso ocorre porque suas soluções automatizam etapas do processo de análise. Nesse sentido, selecione a opção correta que trata dos componentes de um sistema especialista.

A

Basicamente, um sistema especialista possui quatro componentes, sendo que o mais importante deles é a interface com o usuário.

B

O motor de inferência é a base de conhecimento do sistema especialista.

C

Deve ter preocupação com a interação com o usuário, uma vez que é nela que ocorrem as consultas ao sistema.

D

A utilização de sistemas de bancos de dados é essencial para que as sugestões do sistema especialista sejam confiáveis.

E

A base de conhecimento do sistema especialista é baseada na interação com o usuário, uma vez que ela está em constante evolução.



A alternativa C está correta.

A comunicação do usuário com o sistema especialista é feita por meio da interface gráfica. Portanto, é necessário investir na construção de um ambiente que facilite que o usuário faça consultas para o sistema e que possa interagir facilmente com as sugestões propostas.

2. Entre as aplicações dos sistemas especialistas está o suporte para diagnósticos a partir da análise de sintomas. O ideal é que as sugestões do sistema sejam confiáveis de modo que os usuários possam utilizá-las sem o receio de precisar fazer correções constantemente. Nesse sentido, selecione a opção correta que trata sobre a construção de um sistema especialista.

A

A base de conhecimento confiável é composta pelas recomendações que estão disponíveis nos manuais técnicos.

B

O especialista técnico sempre deve ter o principal papel na construção da base de conhecimento e não deve ser questionado.

C

A construção do conhecimento está associada à experiência do especialista técnico ao resolver problemas práticos, portanto, é importante não o direcionar, pois isso reduziria a sua colaboração para formar a base de conhecimento.

D

Para poder extrair o conhecimento do especialista técnico é necessário que o engenheiro do conhecimento conduza o processo.

E

Deve haver um esforço no início de um projeto para construção da base de conhecimento mais genérica possível, para que, ao longo do tempo, ela evolua naturalmente para sugestões mais específicas.



A alternativa D está correta.

Ambos especialistas, técnico e engenheiro do conhecimento têm papéis fundamentais na construção do sistema especialista, pois transferir conhecimento não é um processo trivial. Então, é necessário que haja um esforço para facilitar esse processo de transferência, ao mesmo tempo que também deve haver o desenvolvimento de metodologias de validação da efetividade das sugestões que fazem parte da base de conhecimento.

Redes neurais artificiais

São inspiradas nas redes neurais biológicas que constroem a estrutura do cérebro. Trata-se de modelos computacionais utilizados na inteligência artificial para lidar com problemas de regressão e de classificação (Haykin, 2008).

Os dendritos da rede neural biológica representam entradas em redes neurais artificiais; o núcleo da célula representa os nós; a **sinapse** representa os pesos; e o axônio representa a saída.

De forma resumida, a relação entre as duas redes é dada na tabela a seguir.

sinapse

Estímulo das células nervosas.

Rede neural biológica	Rede neural artificial
Dendritos	Entradas
Núcleo da célula	Nós
Sinapse	Pesos
Axônio	Saída

Tabela: Relação entre redes neurais biológicas e artificiais.
Elaborada por: Sérgio Assunção Monteiro.

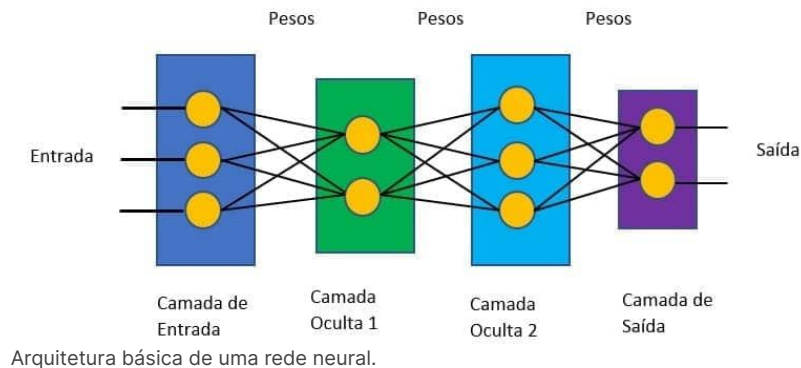
No campo da inteligência artificial, uma rede neural artificial tenta imitar a rede de neurônios que forma o cérebro humano, a fim de que os computadores possam ter um comportamento semelhante aos humanos para entender as coisas e tomar decisões. Por meio da programação de computadores, as redes neurais artificiais são projetadas para se comportar como células cerebrais interconectadas.

Arquitetura básica das redes neurais

As redes neurais artificiais são compostas por camadas de nós, contendo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída. Cada nó ou, ainda, neurônio artificial pode se conectar a outros nós, sendo que essa conexão também possui um peso e uma função de ativação associados.

Se a saída de qualquer nó individual polarizar a função de ativação, ou seja, se estiver acima do valor limite especificado, de modo a ativar a função, o respectivo nó também será ativado e enviará dados para a próxima camada da rede. Caso o nó não seja ativado, nenhum dado será passado para a próxima camada da rede.

Na imagem a seguir, apresentamos a arquitetura básica de uma rede neural.



Uma rede neural artificial é composta por seis componentes básicos, como veremos a seguir:

Camada de entrada

Os sinais de entrada — normalmente, representados pelas variáveis X_1, X_2, \dots, X_m , sendo que o índice da variável indica a que entrada a variável se refere — são enviados para a camada de entrada e transmitidos para a próxima camada, sem que haja processamento estatístico. Portanto, se os dados precisam ser **limpos**, o que significa que demandam um pré-processamento, isso deve ser feito antes de serem enviados para a rede neural, caso contrário isso pode prejudicar o processo de aprendizado da rede. A única função que se espera dessa camada é que ela transmita os dados para a próxima camada. Os dados dos sinais de entrada precisam ser quantificáveis, ou seja, devem ser numéricos. No entanto, as redes neurais podem ser aplicadas para diversas áreas, como reconhecimento de imagens e sons, por exemplo. Nesse caso, é necessário converter esses dados para números. Muitas bibliotecas disponíveis, como **Keras**, facilitam esse trabalho para o desenvolvedor.

Pesos sinápticos

Ou pesos, normalmente, são representados pelas variáveis w_{ji}^l , sendo que o índice l se refere à camada origem do nó, i e j são o índice do nó da camada $l + 1$ que relaciona os nós i e j . Os pesos indicam a importância dos dados de entrada para a camada oculta e seu efeito no aprendizado da rede. São coeficientes estatísticos obtidos por um processo de treinamento da rede em que ela **aprende** o padrão dos dados. São, portanto, uma forma de armazenar o conhecimento adquirido.

Camada oculta

São camadas intermediárias que possibilitam que haja troca de informações entre as camadas de entrada e saída. As redes neurais artificiais podem ter mais de uma camada oculta. A decisão de quantas camadas ocultas deve ter uma rede depende do contexto do problema, pois, em alguns casos, se não houver camadas ocultas suficientes, a rede não vai aprender, enquanto em outros casos, se houver mais camadas ocultas do que o necessário, a rede pode se tornar especialista nos dados de treinamento e não será útil para ser aplicada em outras situações. Além disso, a quantidade de camadas ocultas afeta o desempenho computacional de treinamento da rede.

Junção somadora

Essa junção, também chamada de **mesclagem e de função de propagação**, faz a combinação linear dos pesos dos respectivos valores de entrada, incluindo o bias, para cada valor de entrada. Sua fórmula é dada por:

$$s_j = \sum_{i=1}^M x_i w_{ij} + \text{bias}_j$$

Sendo que M representa a quantidade de entradas em determinada camada.

Função de ativação

A função de ativação, também chamada de função de transferência, determina a saída que um nó produz em resposta à entrada, por meio do processamento do resultado da função de adição.

Existem diversos tipos de funções de ativação, como:

- **Função linear:** Sua fórmula é dada por $f(s) = \alpha s$,

sendo que s é o valor da função de soma e α é um escalar (um número).

- **Função sigmoide:** Também chamada de função logística, cuja fórmula é dada por:

$$\sigma(s) = \frac{1}{1+e^{-s}}$$

Sendo que e representa a constante de Euler.

- **Função Tanh (tangente hiperbólica):** Sua fórmula é dada por:

$$\tanh(s) = 2\sigma(2s) - 1$$

Sendo que σ representa a função sigmoide.

- **Função ReLU (unidade linear retificada):** Sua fórmula é dada por:

$$\text{ReLU}(s) = \max\{0, s\}$$

Camada de saída

Transmite o resultado da rede neural artificial para o mundo externo. Consiste em uma única camada.

As redes neurais precisam ser treinadas por meio de dados pré-classificados. Esses dados também são chamados de **rotulados**. À medida que treinamos o modelo, precisamos avaliar sua precisão, usando uma função de custo (ou perda). A função de perda é dada por:

$$(y) = \frac{1}{2M} \sum_{j=1}^M (y_j - \hat{y}_j)^2$$

Sendo que:

"

Representa o valor de referência, ou seja, o dado pré-classificado.

"y"

É a saída produzida pela rede neural.

M

É a quantidade de amostras (ou observações).



Atenção

A função de perda é chamada de erro quadrático médio.

O objetivo do algoritmo de treinamento da rede neural é minimizar a função de perda para garantir a exatidão do ajuste para as observações. Desse modo, à medida que ajusta seus pesos, o modelo usa a função de perda para atingir o ponto de convergência ou o mínimo local.

A seguir, apresentaremos um exemplo de como o algoritmo de treinamento trabalha para encontrar o ponto em que a função de perda é mínima.

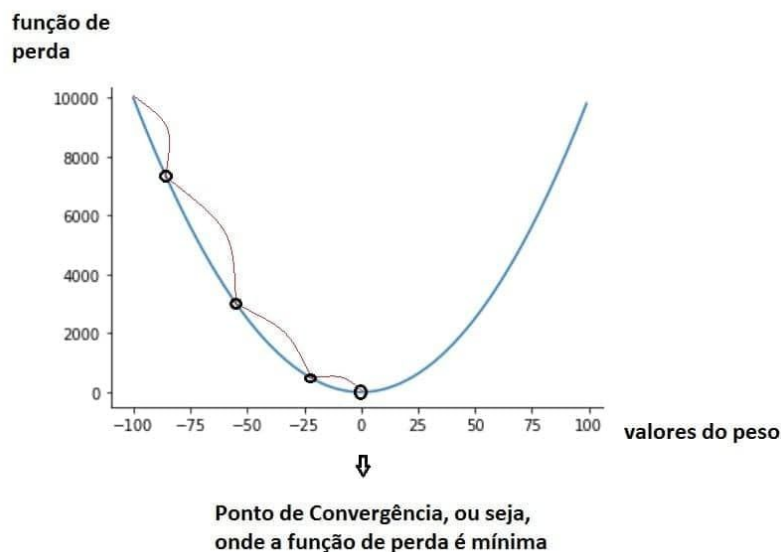


Gráfico: Minimização da função de perda.

Tipos de redes neurais artificiais

As redes neurais podem ser classificadas em diferentes categorias. Cada uma delas é mais adequada para determinados tipos de aplicação. Vamos analisar alguns dos principais tipos de redes neurais.

Perceptron

É a rede neural mais antiga, criada por Frank Rosenblatt, em 1958. Veremos a seguir como funciona sua estrutura:

Entrada

É um vetor de valores reais.

Combinação linear

Os valores de entrada são multiplicados com os respectivos pesos e somados (combinação linear), exatamente como vimos na definição da função de junção somadora.

Função de ativação

A função de ativação retorna +1, se o resultado da função de junção for superior a um determinado limiar, ou -1, caso contrário.

Esse tipo de rede é aplicado para problemas linearmente separáveis.

Redes neurais feedforward

Também chamadas de **redes de propagação direta**, são formadas por uma camada de entrada e uma camada de saída. Em relação à presença de camada intermediária, podem ser:

Single layer

Quando não possuem camada intermediária.



Multi layer

Quando possuem uma ou mais camadas ocultas.

Sua principal característica é que as conexões entre os nós não formam ciclos. São usadas em processamento de linguagem natural e outras redes neurais.

Redes neurais convolucionais (CNNs)

A sua estrutura é parecida com as redes feedforward. São aplicadas, principalmente, para problemas de reconhecimento de imagem e de visão computacional.



Dica

CNNs aplicam operações da álgebra linear para identificar padrões em uma imagem.

Redes neurais recorrentes (RNNs)

São caracterizadas, principalmente, por possuir ciclos direcionados, o que significa que o processamento do algoritmo de treinamento pode retornar para um nó que já foi visitado.



Dica

RNNs são especialmente interessantes para aplicações de dados de séries temporais para fazer estimativas sobre resultados futuros, como previsões do mercado de ações ou previsões de demanda do mercado consumidor de determinado segmento.

Vantagens e desvantagens das redes neurais artificiais

As redes neurais são aplicadas para resolver problemas em diversas áreas. No entanto, a mesma flexibilidade que as tornam bastante interessantes de serem utilizadas, devido à facilidade de adaptação, também é um desafio, pois é necessário configurar muitos parâmetros para que as redes funcionem corretamente, como a quantidade de camadas intermediárias, a quantidade de nós por camada, além de diversos aspectos do algoritmo de treinamento. Nesse sentido, vamos tratar um pouco sobre as vantagens e desvantagens das redes neurais.

Vantagens

Entre as principais vantagens de trabalhar com redes neurais, temos:

Capacidade de processamento paralelo

A estrutura das redes neurais artificiais pode ter muitas subestruturas que viabilizam o processamento simultâneo de mais de uma tarefa.

Capacidade de trabalhar com conhecimento incompleto

A rede neural não precisa de todos os dados dos pesos para continuar trabalhando, ou seja, mesmo com a ausência de alguns deles, pode produzir saídas. No entanto, seu desempenho será influenciado pela importância dos dados ausentes.

Armazenamento de dados em toda a rede

Os dados são os pesos das conexões entre os nós da rede. Eles, de fato, são a representação do conhecimento da rede.

Ter uma distribuição de memória

A ideia do bom uso da rede neural artificial é que ela tenha a capacidade de generalizar sua análise. Portanto, se o conjunto de treinamento e as regras de validação forem bem-ajustadas, a rede poderá extrair as características necessárias para generalizar comportamentos e produzir saídas com qualidade, tanto para fazer classificações como regressões.

Ter tolerância a falhas

Caso sejam perdidos dados sobre alguns pesos da rede, ou algumas ligações entre os nós, ou seja, perdido algum nó, ainda assim a rede pode funcionar. É claro que é necessário verificar o quanto essas perdas influenciaram a qualidade dos resultados, por meio de testes.

Desvantagens

Entre as principais desvantagens de trabalhar com redes neurais, temos:

Garantia de estrutura de rede adequada

Escolher a estrutura adequada da rede não é trivial. Já existem arquiteturas que funcionam muito bem para algumas aplicações, como as redes convolucionais para reconhecimento de imagens e as redes recorrentes para tratamento de dados sequenciais, o que as torna úteis para problemas de previsão de vendas e processamento de linguagem natural, por exemplo. Mas, de um modo geral, a estrutura de rede adequada é obtida por meio de experiência, tentativa e erro.

Comportamento não reconhecido da rede

É o problema mais significativo da rede neural artificial. Quando produz uma solução de teste, a rede neural artificial não fornece uma visão sobre o porquê e como. Isso diminui a confiança na rede.

Dependência de hardware

Redes neurais artificiais precisam de processadores com poder de processamento paralelo, de acordo com sua estrutura. Portanto, a realização do equipamento é dependente.

Dificuldade de explicar o conhecimento obtido pela rede

O conhecimento obtido pela rede está relacionado aos pesos e à estrutura de como os nós se relacionam. Portanto, não é uma forma simples de explicar como a rede chegou a determinado resultado. A confiança no resultado obtido está nos testes que foram feitos na rede anteriormente, mostrando sua qualidade.

A duração de treinamento da rede é desconhecida

O algoritmo de treinamento da rede é iterativo, mas não há uma quantidade ideal de iterações que o algoritmo deve executar para considerarmos que a rede já foi treinada o suficiente. O que é feito, normalmente, é estabelecer uma quantidade máxima de iterações, ou uma tolerância de aproximação para a função de perda que é a que vai ser minimizada. Essas duas soluções, inclusive, podem ser combinadas de modo que o algoritmo de treinamento encerra se atingir uma das duas situações primeiro: determinada tolerância ou uma quantidade máxima de iterações.

Conceitos de redes neurais artificiais

Neste vídeo o especialista aborda os conceitos de redes neurais artificiais.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Verificando o aprendizado

1. As redes neurais artificiais são aplicadas para resolver muitos problemas práticos. Um dos motivos para seu sucesso está na flexibilidade da sua arquitetura. Selecione, a seguir, a alternativa correta sobre os componentes de uma rede neural.

A

O processo de memorização de uma rede neural ocorre com a ativação da função somatório.

B

Uma rede neural pode ser aplicada para problemas de classificação, desde que tenha camadas escondidas.

C

Os pesos da rede indicam a importância dos dados de entrada para a rede.

D

As funções de ativação fazem a soma dos dados de entrada através de um processo sináptico.

E

Não é possível armazenar uma base de conhecimento em uma rede neural.



A alternativa C está correta.

A rede neural artificial armazena o conhecimento por meio dos seus pesos que, por sua vez, indicam a importância dos dados de entrada.

2. As redes neurais artificiais são modelos flexíveis. Elas podem ser adaptadas para serem utilizadas em diferentes contextos. Selecione, a seguir, a alternativa correta que trata dos tipos de redes neurais.

A

Problemas como estimativa de vendas para o próximo período podem ser modelados por redes neurais recorrentes.

B

Redes convolucionais devem ser aplicadas para problemas de análise de regressão.

C

Redes neurais perceptrons devem ser aplicadas para problemas de análise temporal.

D

Problemas de visão computacional devem ser tratados por redes neurais recorrentes.

E

Redes neurais com retroalimentação são representadas pelas redes convolucionais.



A alternativa A está correta.

Redes neurais artificiais recorrentes são adequadas para tratar de problemas que envolvam o estudo de séries temporais.

Regras de aprendizado para redes neurais artificiais

A qualidade de um modelo de rede neural é medida por sua taxa de acertos. Para que o modelo extraia as características dos dados de treinamento de modo a ser capaz de fazer generalizações, ou seja, aprender com os dados, é necessário aplicar métodos que ajustem os pesos sinápticos das conexões entre os nós. Chamamos esses métodos de **regras de aprendizagem**.

Entre as principais regras de aprendizagem (Wilamowski; Irwin, 2011) para as redes neurais estão as de:

- Hebbian
- Correlação
- Perceptron
- Delta Widrow-Hoff Rule
- Instar
- Outstar

A seguir, vamos analisar brevemente cada uma dessas regras.

Regra de aprendizagem Hebbian

Recebe o nome do seu criador, Donald Hebb. Ela é aplicada para fazer classificações. Trata-se de uma rede neural que tem uma camada de entrada e uma camada de saída — ou seja, de camada única. A camada de entrada pode ter diversas unidades de entrada; mas a camada de saída possui apenas uma unidade. O algoritmo faz a atualização dos pesos entre os neurônios na rede neural para cada amostra de treinamento. A formulação matemática para atualização dos pesos sinápticos é dada por:

$$w_{ji}^{t+1} = w_{ji}^t + \eta x_i y_j .$$

Sendo que:

Representa o peso sináptico na iteração $t+1$.

Representa o peso sináptico na iteração t , ou seja, é o peso para a conexão (j,i) na iteração anterior à iteração $t + 1$.

A taxa de aprendizagem positiva e constante.

O valor de entrada da unidade i .

A saída do neurônio.



Atenção

A regra de aprendizado Hebbian é um processo não supervisionado.

Regra de aprendizagem de correlação

É um processo de aprendizado supervisionado, ou seja, precisamos fornecer um valor esperado para a saída que vamos representar por d_j . Os pesos entre neurônios são positivos quando eles respondem simultaneamente, e negativos, caso contrário. A atualização dos pesos é dada por:

$$w^{t+1}_{ji} = w^t_{ji} + \Delta w_{ji} ,$$

Onde ΔW_{ji} — chamado de passo — é definido por:

$$\Delta w_{ji} = \eta x_i d_j .$$

Os pesos do início do treinamento, normalmente, recebem valores iguais a zero e são atualizados ao longo das iterações.

Regra de aprendizagem do Perceptron

É um método de aprendizado supervisionado, que utiliza o cálculo da diferença entre a saída-alvo e a real — o que chamamos de **erro** — para atualizar os pesos sinápticos. O erro é formalizado pela fórmula:

$$e_j = d_j - y_j .$$

Sendo que d_j é a saída desejada e y_j é o resultado de saída obtido. A fórmula de atualização dos pesos é dada por:

$$\Delta w_{ji} = \eta x_i e_j .$$

Regra de aprendizagem Delta Widrow-Hoff

Recebe o nome dos seus criadores Bernard Widrow e Marcian Hoff. Ela também é chamada de método dos mínimos quadrados médio (least mean square — LMS). É um algoritmo de aprendizado supervisionado. A ideia do algoritmo é minimizar o quadrado da diferença entre o valor de saída-alvo e o valor de saída obtido. O erro para o j -ésimo neurônio é dado por:

$$e_j = \sum_{p=1}^P (s_{jp} - d_{jp})^2 .$$

Sendo que:

P

É a quantidade de padrões de entrada, ou seja, o total de dados de treinamento.

É a saída alvo para o j -ésimo neurônio aplicado ao p -ésimo padrão.

É a saída obtida para o j -ésimo neurônio aplicado ao p -ésimo padrão.

Também s_{jp} é dada por:

$$s = \sum_{i=1}^n w_i x_i$$

Agora, precisamos calcular o gradiente, obtido através da derivação de e_j em relação a w_{ij} . O gradiente, então, é dado por:

$$\frac{\partial e_j}{\partial w_{ij}} = 2x_{ij} \sum_{p=1}^P (d_{jp} - s_{jp}) .$$

Por fim, obtemos a fórmula de atualização dos pesos que é dada por:

$$\Delta w_{ji} = \eta x_{ij} \sum_{p=1}^P (d_{jp} - s_{jp})$$

Regra de aprendizagem Instar (o vencedor da regra de aprendizagem competitiva leva tudo)

É um método de aprendizado não supervisionado. A ideia é que os nós de saída compitam entre si para representar o padrão de entrada. Isso ocorre durante o treinamento, em que será declarada vencedora a

unidade de saída que tiver a maior ativação para um determinado padrão de entrada. Essa regra também foi desenvolvida por Grossberg. A ideia básica é: quando os valores das entradas e dos pesos forem apenas binários bipolares, ou seja, -1 ou +1, então, o somatório dos produtos entre os respectivos pares de peso e entrada terá o maior valor positivo quando os pesos e os sinais de entrada forem iguais, ou seja:

$$s = \sum_{i=1}^n w_i x_i ,$$

Portanto, os pesos devem ser alterados apenas se forem diferentes dos sinais. A fórmula de atualização dos pesos é dada por:

$$\Delta w_j = \eta (x_j - w_j) .$$

Aqui precisamos observar que as informações necessárias para atualização dos pesos são obtidas apenas dos sinais de entrada.

Regra de aprendizagem Outstar

É um método de aprendizado supervisionado, pois os resultados alvos são conhecidos. Também é chamado de aprendizagem de Grossberg, como referência ao seu criador. É aplicado para neurônios de uma camada. A fórmula para atualização dos pesos é dada por:

$$\Delta w_j = \eta (d - w_j) .$$

Sendo que η é a taxa de aprendizado e d é a saída-alvo.

Treinamento de redes neurais artificiais

O principal método para treinar uma rede neural artificial de múltiplas camadas é o backpropagation. Devemos aplicá-lo para ajustar os pesos sinápticos da rede, de modo que ela atinja seu objetivo, que é extrair as características do conjunto de dados de treinamento e ter uma taxa de acerto aceitável. Esse método consiste em duas fases:

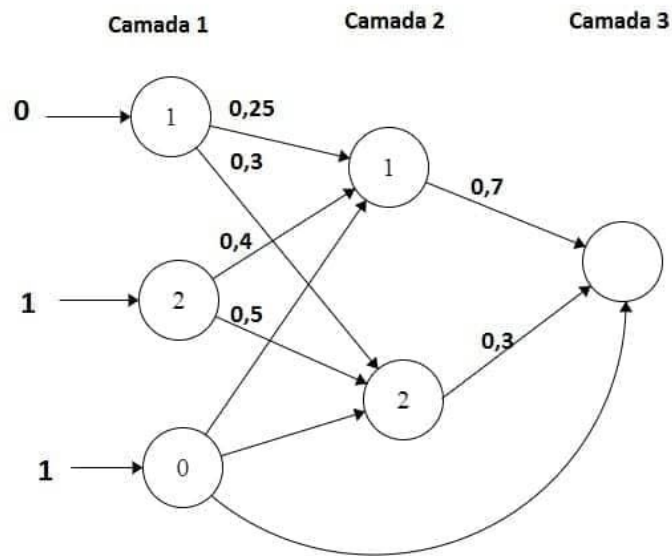
Ida

Chamada de **fase forward**. Consiste no cálculo de variáveis intermediárias que vai do sentido da camada de entrada até a saída da rede.

Volta

Chamada de **fase backward**. Consiste no ajuste dos pesos através do cálculo dos erros que são combinados e aplicados com a regra **delta** de aprendizagem. O sentido do processamento vai da saída até a entrada dos dados.

Vamos analisar um exemplo para entendermos melhor. Suponha a seguinte situação: temos as entradas binárias 0 e 1 e a sua saída-alvo é 1. Desse modo, vamos montar a rede neural representada na imagem a seguir.



Exemplo de rede neural multicamadas.

Os nós 1 e 2 correspondem à entrada da rede. O nó 0 é o bias. Os valores que estão nas ligações entre os nós são os pesos sinápticos. Eles podem ser gerados de diversas formas, inclusive, como em nosso exemplo, aleatoriamente. Os pesos associados ao nó bias são iguais a 1. Precisamos fazer o processamento do algoritmo, porém, antes disso, vamos estabelecer o valor do parâmetro taxa de aprendizado (η). Vamos utilizar $\eta = 0,7$.

Agora, vamos calcular a fase forward com os seguintes passos:

Passo 1

Calcular a soma do produto dos pesos com suas respectivas entradas para os nós 1 e 2, respectivamente, da camada 2. Assim, obtemos:

$$s_1 = 0,25 \cdot (0) + 0,4 \cdot (1) + 1,0 \cdot (1) = 1,40$$

$$s_2 = 0,3 \cdot (0) + 0,5 \cdot (1) + 1 = 1,50$$

Passo 2

Calcular a função de ativação dos nós 1 e 2:

$$\text{amp: } a_1 = f(s_1) = \frac{1}{1 + e^{-1.4}} = 0,80$$

$$\text{amp: } a_2 = f(s_2) = \frac{1}{1 + e^{-1.5}} = 0,82$$

Passo 3

Calcular a soma do produto dos pesos da camada 3 com seus respectivos valores a_1 e a_2 . Desse modo, obtemos:

$$s = 0,80 \cdot (0,7) + 0,82 \cdot (0,3) + 1,0 \cdot (1) = 1,81$$

Passo 4

Calcular a função de ativação:

$$y = f(s) = \frac{1}{1 + e^{-1,81}} = 0,86$$

Devemos lembrar que o valor-alvo é $d = 1$. Então, temos que calcular o módulo da diferença entre os valores obtidos e o alvo, que dá: $|y - d| = 0,14$. O modelo estará mais bem-ajustado quanto mais próximo de zero for o erro no nó de saída. Então, passamos a fazer os ajustes dos pesos. Essa é a fase backward, que tem a sequência de passos dada por:

Passo 5

Calcular o delta do nó de saída:

$$\delta = (d - y)f(s)(1 - f(s)) = 0,0169$$

Passo 6

Calcular os ajustes dos pesos das conexões entre os nós da camada 2 e 3:

$$\begin{aligned} <lr> \Delta w_{13} = \eta \delta a_1 = 0,7 \cdot (0,0169) \cdot (0,80) = 0,0095 \\ <lr> <lr> \Delta w_{23} = \eta \delta a_2 = 0,7 \cdot (0,0169) \cdot (0,82) = 0,0097 <lr> \\ <lr> \Delta w_{33} = \eta \delta = 0,7 \cdot (0,0169) = -0,0118 <lr> \end{aligned}$$

Passo 7

Ajustar os pesos das conexões entre os nós da camada 2 e 3:

$$\begin{aligned} <lr> <lr> w_{13} = w_{13} + \Delta w_{13} = 0,7095 \\ <lr> <lr> w_{23} = w_{23} + \Delta w_{23} = 0,3097 <lr> \\ <lr> w_{33} = w_{33} + \Delta w_{33} = 1,0118 <lr> \end{aligned}$$

Passo 8

Calcular os valores de delta para os nós da camada 2:

$$\begin{aligned} amp; \delta_1 &= \delta f(s_1)(1 - f(s_1)) = 0,0027 \\ amp; \delta_2 &= \delta f(s_2)(1 - f(s_2)) = 0,0024 \end{aligned}$$

Passo 9

Calcular os ajustes dos pesos das conexões entre as conexões 1 e 2:

$$\begin{aligned} \Delta w_{12}^{(1)} &= -\delta_{12} \cdot x_1 \\ \Delta w_{12}^{(2)} &= -\delta_{12} \cdot x_2 \\ \Delta w_{12}^{(3)} &= -\delta_{12} \cdot x_3 \\ \Delta w_{12}^{(4)} &= -\delta_{12} \cdot x_4 \\ \Delta w_{12}^{(5)} &= -\delta_{12} \cdot x_5 \end{aligned}$$

Passo 10

Fazer os ajustes dos pesos:

$$\begin{aligned} \Delta w_{12}^{(1)} &= -\delta_{12} \cdot x_1 \\ \Delta w_{12}^{(2)} &= -\delta_{12} \cdot x_2 \\ \Delta w_{12}^{(3)} &= -\delta_{12} \cdot x_3 \\ \Delta w_{12}^{(4)} &= -\delta_{12} \cdot x_4 \\ \Delta w_{12}^{(5)} &= -\delta_{12} \cdot x_5 \end{aligned}$$

$$\begin{aligned} \Delta w_{12}^{(1)} &= -\delta_{12} \cdot x_1 \\ \Delta w_{12}^{(2)} &= -\delta_{12} \cdot x_2 \\ \Delta w_{12}^{(3)} &= -\delta_{12} \cdot x_3 \\ \Delta w_{12}^{(4)} &= -\delta_{12} \cdot x_4 \\ \Delta w_{12}^{(5)} &= -\delta_{12} \cdot x_5 \end{aligned}$$

O método backpropagation é relativamente simples de implementar e bastante adaptável. Suas desvantagens são a convergência, que pode ser lenta, e o desempenho de tempo depender dos dados de treinamento e dos parâmetros iniciais.

Treinamento de redes neurais artificiais

No vídeo a seguir, o especialista fala sobre as regras de aprendizado de redes neurais e algoritmo de treinamento, além dos aspectos fundamentais para que as redes neurais produzam resultados que possam ser utilizados na prática.



Conteúdo interativo

Acesse a versão digital para assistir ao vídeo.

Verificando o aprendizado

1. As redes neurais artificiais têm a capacidade de extrair características dos dados. Essa capacidade é muito útil para fazer classificações. A respeito das regras de aprendizado das redes neurais, selecione a seguir a alternativa correta.

A

O processo de ajuste dos pesos sinápticos influencia a capacidade de a rede generalizar suas classificações. Por isso, deve-se aplicar o somatório dos pesos para cada nó.

B

A função de ativação é o principal aspecto a ser considerado, para que uma rede neural aumente sua taxa de acerto.

C

O ajuste dos pesos é um processo probabilístico e, portanto, não é um aspecto que deve ser levado em consideração para aumentar a qualidade da classificação de uma rede.

D

O processo de aprendizado das redes neurais artificiais é baseado na redução dos ruídos. Por isso, sempre que possível, deve-se aumentar os valores dos pesos ao longo das iterações.

E

A atualização dos pesos sinápticos e a estrutura da rede influenciam a capacidade de generalização da rede neural.



A alternativa E está correta.

As redes neurais artificiais são modelos probabilísticos muito úteis para resolver problemas de classificação. Para que uma rede extraia as características do conjunto de treinamento é necessário tratar dois aspectos: a arquitetura da rede, que é a forma como os nós são conectados entre si através de camadas; e um método de ajustes dos pesos sinápticos.

2. Gerenciar o processo de aprendizado de uma rede neural assegura que o modelo possa ser utilizado em aplicações práticas. Esse processo, no entanto, não é simples, pois muitos pontos devem ser considerados. Selecione, a seguir, a alternativa correta que trata do processo de treinamento das redes neurais.

A

O processo de aprendizado backpropagation garante o ajuste ótimo dos pesos de um modelo com apenas uma passagem de ida e uma de volta pela rede neural.

B

O treinamento de uma rede neural pode ser feito através do método backpropagation, que tem como vantagem a independência dos dados de treinamento.

C

Uma das etapas do ajuste dos pesos sinápticos de uma rede neural é a aplicação de uma regra de aprendizado.

D

O parâmetro de aprendizado ajuda a aumentar a taxa de acerto do modelo de rede neural, por isso ele deve ser maior que 1.

E

Durante o cálculo dos pesos sinápticos de uma rede, é preciso ter atenção com possíveis erros de arredondamento, pois o modelo utiliza esses valores de forma exata.



A alternativa C está correta.

O método backpropagation é um dos métodos mais utilizados para treinamento de redes neurais artificiais. Ele tem duas etapas bem distintas: o processamento da camada de entrada para a camada de saída e da saída para a entrada. O ajuste dos pesos é feito na segunda etapa chamada de backward, com a aplicação de uma regra de aprendizagem. Normalmente, a regra usada é a delta.

Considerações finais

Estudamos os aspectos que devem ser tratados para a criação da base de conhecimento, o alicerce dos sistemas especialistas. Esses sistemas podem ser aplicados para suporte à decisão em diversas situações práticas. A ideia é que trabalhem como um apoio ao analista.

Um dos pontos fundamentais dos sistemas especialistas é que tenham a capacidade de extrair características dos dados de treinamento e, a partir disso, façam generalizações. Isso é importante para reduzir a dependência do especialista humano e dar escalabilidade para as soluções.

As redes neurais artificiais são modelos computacionais aplicados para desenvolver sistemas especialistas. A ideia básica desses modelos é simular o comportamento de aprendizado e classificação das redes neurais biológicas. Como vimos, montar a arquitetura da rede e ajustar os pesos sinápticos estão entre os desafios para desenvolver redes que produzam resultados úteis na prática, ou seja, com alta taxa de acerto.

Podcast

Ouçá sobre os conceitos de lógica para inteligência artificial, o que são e para que servem os sistemas especialistas, e sobre as redes neurais artificiais.



Conteúdo interativo

Acesse a versão digital para ouvir o áudio.

Explore+

Pesquise o site oficial do **pacote scikit-learn** e aprenda mais sobre aplicações práticas de redes neurais artificiais.

Referências

GERSTING, J. L. **Fundamentos matemáticos para a ciência da computação**. São Paulo: LTC, 2001.

GROSSBERG, S. **Adaptive pattern classification and universal recoding**: i. parallel development and coding of neural feature detectors. *Biological Cybernetics*, v. 23, n. 3, p. 121-34, set. 1976.

HAYKIN, S. **Neural networks and learning machines**. Nova Jersey: Prentice Hall, 2008.

KERAS. Consultado na internet em: 29 jun. 2021.

ROSENBLATT F. **The perceptron**: a probabilistic model for information storage and organization in the brain. *Psychological Review*, v. 65, n. 6, p. 386-408, nov. 1958.

RUSSEL, S.; NORVIG, P. **Inteligência artificial**. 2. ed. Rio de Janeiro: Campus, 2004.

WILAMOWSKI, B. M.; IRWIN, J. D. **Intelligent systems**. 2. ed. Boca Raton: Taylor and Francis Group, 2011.

SWI-Prolog. Consultado na internet em: 29 jun. 2021.