Inteligência Artificial

Prof. Cristiano Roberto Franco

UNIASSELVI
2017

Elaboração:

Prof. Cristiano Roberto Franco

Revisão, Diagramação e Produção:

Centro Universitário Leonardo da Vinci – UNIASSELVI

Ficha catalográfica elaborada na fonte pela Biblioteca Dante Alighieri
UNIASSELVI – Indaial.

006.3
F825i

Franco; Cristiano Roberto
Inteligência artificial / Cristiano Roberto Franco: UNIASSELVI, 2017.

180 p. : il.

ISBN 978-85-515-0066-8

1.Inteligência Artificial.
I. Centro Universitário Leonardo Da Vinci.

Impresso por:

Apresentação

O campo da Inteligência Artificial (IA) sempre despertou fascínio no homem. Desde os primórdios tentamos criar uma máquina que pudesse demonstrar as mesmas características que nos definem como "inteligentes". Esta fascinação pode ser facilmente observada através da quantidade de livros e filmes que tratam sobre o assunto. Isaac Asimov, com seus livros e contos de ficção científica, e filmes como Blade Runner, Exterminador do Futuro, I. A. e Matrix trouxeram para os olhos do grande público uma visão do que é a Inteligência Artificial.

Ao mesmo tempo em que a ideia de que o homem seja capaz de construir uma máquina "inteligente" nos maravilha, ela nos assombra. Isaac Asimov, em seu livro (já transformado para filme) *Eu, Robô* idealiza três princípios básicos da robótica visando exclusivamente impedir que essas máquinas inteligentes possam ferir seres humanos ou mesmo outras máquinas inteligentes. Em Blade Runner e outros filmes citados acima, em algum momento a inteligência dessas máquinas iguala e até mesmo supera a dos homens e a partir daí somos subjugados por elas. Esse momento é o que chamamos de singularidade computacional.

A realidade, pelo menos por enquanto, é bem diferente. Os estudos científicos sobre a inteligência artificial hoje já sabem que existem diversos obstáculos a serem superados antes de conseguirmos algo semelhante à ficção científica. A representação do conhecimento, por exemplo, é um dos aspectos mais complexos para a resolução de problemas em inteligência artificial e a ciência ainda busca uma maneira eficiente de representar o conhecimento de forma mais genérica em sistemas computacionais. Em resumo, a singularidade computacional ainda é uma realidade relativamente distante. Entretanto, com o crescente número de computadores nos mais diversos setores da vida humana, com a utilização de sistemas computacionais interativos e com o aparecimento da internet, estamos cada vez mais envolvidos com essa tecnologia. Impulsionadas pelos gigantes da internet e seus praticamente "ilimitados" recursos, hoje as pesquisas em inteligência artificial avançam mais rápido do que nunca. Ao fazer uma busca no Google sobre determinado assunto ou ao pesquisar o preço de alguma mercadoria na Amazon, você está implicitamente utilizando alguma técnica de inteligência artificial que serve para retornar resultados mais relevantes, ou seja, para deixar sua busca mais "inteligente".

Consideramos este um dos nossos objetivos: fornecera você, acadêmico, o entendimento sobre de que forma a inteligência artificial pode ser inserida e auxiliar em nossas atividades diárias. Com este enfoque, ao longo deste livro e suas três unidades, ilustraremos os principais conceitos e técnicas utilizadas para a resolução de problemas no campo da inteligência artificial, bem como seus principais obstáculos.

Na primeira unidade trataremos de conceitos básicos de inteligência artificial, com luz aos principais fatos científicos e históricos que impulsionaram o seu desenvolvimento. Serão abordados também os conceitos de agente e ambiente, essenciais para o correto entendimento dos conteúdos das demais unidades. Ainda nesta unidade, abordaremos um dos aspectos mais complexos dentro do campo da inteligência artificial, a representação do conhecimento. Exemplificaremos modelos através dos quais o conhecimento pode ser representado, ilustrando os pontos fortes e fracos de cada um. Veremos ainda, através de exemplos, que diversos problemas da inteligência artificial podem ser resolvidos utilizando busca e técnicas de heurística. Faremos também uma introdução ao ramo específico da IA simbólica conhecido como Sistemas Baseados em Conhecimento (SBC).

Na segunda unidade detalharemos a lógica difusa, as redes bayesianas e a computação evolutiva, através dos algoritmos genéticos. Destacaremos ainda técnicas para fazer o tratamento da incerteza, ideais para determinados domínios do conhecimento e situações em que as informações são parciais ou incompletas. Durante esta unidade será ainda demonstrada a utilização de ferramentas computacionais que permitem a implementação prática dos conceitos estudados.

Por fim, na terceira unidade abordaremos o aprendizado de máquina, redes neurais artificiais e uma demonstração da ferramenta TensorFlow, que hoje é uma das mais completas existentes na área e que foi liberada para uso pela comunidade há pouco mais de um ano pelo Google.

Aproveitamos esse momento para destacar que os exercícios **NÃO SÃO OPCIONAIS**. O objetivo de cada exercício deste livro é a fixação de determinado conceito através da prática. É aí que reside a importância da realização de todos. Sugerimos fortemente que em caso de dúvida em algum exercício, você entre em contato com seu tutor externo ou com a tutoria da UNIASSELVI e que não passe para o exercício seguinte enquanto o atual não estiver completamente compreendido. Aprender inteligência artificial sem exercitar os conceitos de forma prática, somente com a leitura do livro, é equivalente a ir a um restaurante e tentar matar a fome lendo o cardápio.

Você precisará de muita determinação e força de vontade, pois o conteúdo abordado neste livro não é fácil e você levará mais do que um semestre para compreendê-lo totalmente, entretanto, aqui forneceremos um início sólido e consistente. Desta forma, passe por esse período de estudos com muita dedicação, pois você, que hoje é acadêmico, amanhã será um profissional de Tecnologia da Informação com o compromisso de construir uma nação melhor.

Lembre-se de que o encantamento com a programação e com os algoritmos de inteligência artificial deriva do seu entendimento; a beleza desta área do conhecimento é a compreensão da lógica envolvida na construção de programas. Por isso, bons programadores são apaixonados por linguagens de programação, novas tecnologias e ambientes de desenvolvimento e estão sempre buscando por novos conhecimentos. Como disse Drew Houston, criador do DropBox: "Programar é a coisa mais parecida que temos com superpoderes".

Bom aprendizado!

Cristiano Roberto Franco

Você já me conhece das outras disciplinas? Não? É calouro? Enfim, tanto para você que está chegando agora à UNIASSELVI quanto para você que já é veterano, há novidades em nosso material.

Na Educação a Distância, o livro impresso, entregue a todos os acadêmicos desde 2005, é o material base da disciplina. A partir de 2017, nossos livros estão de visual novo, com um formato mais prático, que cabe na bolsa e facilita a leitura.

O conteúdo continua na íntegra, mas a estrutura interna foi aperfeiçoada com nova diagramação no texto, aproveitando ao máximo o espaço da página, o que também contribui para diminuir a extração de árvores para produção de folhas de papel, por exemplo.

Assim, a UNIASSELVI, preocupando-se com o impacto de nossas ações sobre o ambiente, apresenta também este livro no formato digital. Assim, você, acadêmico, tem a possibilidade de estudá-lo com versatilidade nas telas do celular, *tablet* ou computador.

Eu mesmo, UNI, ganhei um novo *layout*, você me verá frequentemente e surgirei para apresentar dicas de vídeos e outras fontes de conhecimento que complementam o assunto em questão.

Todos esses ajustes foram pensados a partir de relatos que recebemos nas pesquisas institucionais sobre os materiais impressos, para que você, nossa maior prioridade, possa continuar seus estudos com um material de qualidade.

Aproveito o momento para convidá-lo para um bate-papo sobre o Exame Nacional de Desempenho de Estudantes – ENADE.

Bons estudos!

Olá acadêmico! Para melhorar a qualidade dos materiais ofertados a você e dinamizar ainda mais os seus estudos, a Uniasselvi disponibiliza materiais que possuem o código *QR Code*, que é um código que permite que você acesse um conteúdo interativo relacionado ao tema que você está estudando. Para utilizar essa ferramenta, acesse as lojas de aplicativos e baixe um leitor de *QR Code*. Depois, é só aproveitar mais essa facilidade para aprimorar seus estudos!

BATE PAPO SOBRE O ENADE!

Olá, acadêmico!

Você já ouviu falar sobre o **ENADE**?

Se ainda não ouviu falar nada sobre o ENADE, agora você receberá algumas informações sobre o tema.

Ouviu falar? Ótimo, este informativo reforçará o que você já sabe e poderá lhe trazer novidades.

Vamos lá!

Qual é o significado da expressão ENADE?

EXAME NACIONAL DE DESEMPENHO DOS ESTUDANTES

Em algum momento de sua vida acadêmica você precisará fazer a prova ENADE.

Que prova é essa?

É **obrigatória**, organizada pelo INEP – Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira.

Quem determina que esta prova é obrigatória... O **MEC – Ministério da Educação**.

O objetivo do MEC com esta prova é o de avaliar seu desempenho acadêmico assim como a qualidade do seu curso.

Fique atento! Quem não participa da prova fica impedido de se formar e não pode retirar o diploma de conclusão do curso até regularizar sua situação junto ao MEC.

Não se preocupe porque a partir de hoje nós estaremos auxiliando você nesta caminhada.

Você receberá outros informativos como este, complementando as orientações e esclarecendo suas dúvidas.

Você tem uma trilha de aprendizagem do ENADE, receberá e-mails, SMS, seu tutor e os profissionais do polo também estarão orientados.

Participará de webconferências entre outras tantas atividades para que esteja preparado para #mandar bem na prova ENADE.

Nós aqui no NEAD e também a equipe no polo estamos com você para vencermos este desafio.

Conte sempre com a gente, para juntos mandarmos bem no ENADE!

UNIASSELVI

SUMÁRIO

UNIDADE 1

INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

OBJETIVOS DE APRENDIZAGEM

Ao final desta unidade você será capaz de:

- identificar e descrever aspectos históricos do campo da inteligência artificial e ilustrar os principais conceitos que o envolvem;
- conhecer sistemas baseados em conhecimento e suas principais derivações.

PLANO DE ESTUDOS

Esta unidade de ensino está dividida em três tópicos. No final de cada um deles, você encontrará atividades que contribuirão para a apropriação dos conteúdos.

TÓPICO 1 – INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

TÓPICO 2 – REPRESENTAÇÃO DO CONHECIMENTO E RESOLUÇÃO DE PROBLEMAS ATRAVÉS DA BUSCA

TÓPICO 3 – SISTEMAS BASEADOS EM CONHECIMENTO (SBC)

TÓPICO 1

INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL

1 INTRODUÇÃO

Seja bem-vindo à Unidade 1 do Livro de Estudos de Inteligência Artificial. Nesta unidade fazemos a combinação entre teoria e prática com o objetivo de introduzir você a alguns dos principais conceitos dessa fascinante área da ciência da computação. A primeira questão que buscamos responder diz respeito à própria definição de inteligência. Como dizer que determinado sistema possui ou não características inteligentes se o significado de inteligência ainda é, por si só, bastante nebuloso?

Na Seção 1 elucidaremos o conceito de inteligência no contexto de sistemas computacionais de acordo com critérios de classificação utilizados por autores renomados da área. Diferenciaremos o pensar racional e o pensar inteligente, justificando o porquê de esta sutil diferença ser tão relevante para a área de inteligência artificial. Mostraremos ainda, nesta seção, os conceitos de agentes e ambientes de atuação.

Já na Seção 2 trataremos um aspecto extremamente relevante para a inteligência artificial: a representação do conhecimento. Entenderemos a relação intrínseca entre conhecimento e inteligência e o motivo de a escolha da técnica certa para representar o conhecimento ser uma das etapas mais importantes no projeto de sistemas de inteligência artificial.

Ainda com vistas à representação do conhecimento, demonstraremos que diversos problemas do campo da inteligência artificial são resolvidos através de busca em um espaço de soluções possíveis. Serão apresentadas algumas técnicas de busca e heurísticas utilizadas no sentido de otimizá-las.

Na Seção 3 apresentaremos os Sistemas Baseados em Conhecimento (SBC), em que discutiremos, principalmente, os sistemas especialistas, que aplicam regras e conhecimentos extraídos de especialistas humanos para resolver problemas.

Embarque conosco nesta jornada para explorar a inteligência artificial.

2 INTRODUÇÃO À INTELIGÊNCIA ARTIFICIAL, AGENTES E AMBIENTES

Desde o surgimento do termo inteligência artificial, ele tem gerado certa polêmica na comunidade científica. A primeira dificuldade vem da própria definição da inteligência como qualidade. O que define algo como inteligente? E de que forma a inteligência pode ser medida? O comportamento de um animal buscando alimento pode ser considerado inteligente? Um homem que joga xadrez é mais inteligente que um homem que cuida da lavoura? E uma máquina que reproduza um comportamento que, quando feito pelo homem este é considerado inteligente, pode ser considerada da mesma forma? Estas e outras perguntas ainda são controversas, cuja resposta depende muito da bibliografia e do ponto de vista utilizados.

Outro aspecto controverso é a área de conhecimento em que o campo inteligência artificial se enquadra. Filosofia? Matemática? Engenharia? Biologia? Ciência da Computação? Hoje em dia a resposta mais aceita é a de que o campo inteligência artificial é uma ciência multidisciplinar essencialmente parte da Ciência da Computação, uma vez que o computador foi escolhido como a máquina para se "implantar" inteligência. Trataremos desses temas quando abordarmos o histórico da inteligência artificial.

Na parte conceitual, ilustraremos, de forma geral, os sistemas de inteligência artificial utilizando o conceito amplamente aceito de agente, definido por Norvig e Russel (2004). Com vistas a esse enfoque, definiremos ainda os tipos de ambiente onde esses agentes atuam através de textos e exemplos práticos.

2.1 DEFINIÇÃO

O que hoje conhecemos como inteligência artificial é um ramo da ciência da computação ao mesmo tempo recente (o termo surgiu oficialmente em 1956) e antigo, cujos princípios baseiam-se em ideias filosóficas, científicas e tecnológicas herdadas de outras ciências, entre elas a lógica, que já existe há 23 séculos (BITTENCOURT, 2006).

A dificuldade em se definir a IA se transforma na dificuldade em definir o significado do termo inteligência propriamente dito. Todos somos capazes de identificar se determinado comportamento é ou não inteligente quando parte de um humano ou um animal, entretanto, tal observação é mais complexa quando se trata de programas de computador. O estudioso de Ciência da Computação Edsger Dijkstra cunhou uma frase polêmica a respeito da IA que dizia: "Perguntar se uma máquina pode pensar é o mesmo que perguntar se um submarino pode nadar". Ou seja, para Dijkstra, o objetivo final é o que importa, e se este foi atingido por uma máquina, pode-se considerar que esta máquina é inteligente.

Edsger Wybe Dijkstra foi um cientista da computação holandês que criou o conhecido algoritmo de Dijkstra, também conhecido como algoritmo do menor caminho, muito utilizado nos roteadores da internet e sistemas de GPS existentes hoje em dia.

Luger (2004) define IA como o ramo da ciência da computação que se ocupa da automação do comportamento inteligente. Fernandes (2008) esclarece que é a parte da ciência da computação voltada para o desenvolvimento de sistemas de computadores inteligentes, isto é, sistemas que exibem características que estão associadas à inteligência no comportamento humano, como compreensão da linguagem, aprendizado, raciocínio, resolução de problemas, entre outros.

As definições de IA, em geral, variam em duas características principais: o pensamento e raciocínio e o comportamento. O Quadro 1 mostra seis definições:

QUADRO 1 - DEFINIÇÕES DE IA

Sistemas que pensam como humanos	**Sistemas que pensam racionalmente**
O novo e interessante esforço para fazer os computadores pensarem... máquinas com mentes, no sentido total e literal (HAUGELAND, 1985 apud NORVIG e RUSSEL, 2004).	O estudo das faculdades mentais pelo uso de modelos computacionais (CHARNIAK e MCDERMOTT, 1985 apud NORVIG e RUSSEL, 2004).
Automatização de atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado (BELLMAN, 1978 apud NORVIG e RUSSEL, 2004).	O estudo das computações que tornam possível perceber, raciocinar e agir (WINSTON, 1992 apud NORVIG e RUSSEL, 2004).
Sistemas que atuam como humanos	**Sistemas que atuam racionalmente**
A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas (KURZWEIL, 1990 apud NORVIG e RUSSEL, 2004).	A Inteligência Computacional é o estudo do projeto de agentes inteligentes (POOLE et al., 1998 apud NORVIG e RUSSEL, 2004).

FONTE: Adaptado de Norvig e Russel (2004)

Norvig e Russel (2004) ainda afirmam que existe discussão entre essas abordagens, afinal, uma abordagem centrada nos seres humanos deve ser uma ciência empírica, que envolva hipóteses e confirmação experimental. Já uma abordagem racionalista envolve mais engenharia e matemática para confirmação.

Independentemente da abordagem adotada, a IA trata uma variedade ampla de problemas cujas características comuns são definidas por Luger (2004):

- Uso do computador para executar raciocínio, reconhecimento de padrões ou aprendizagem.
- Problemas que não podem ser resolvidos utilizando soluções algorítmicas tradicionais.
- Solução de problemas utilizando informação inexata, faltante ou pobremente definida.
- Respostas nem sempre exatas ou ótimas, mas suficientes em determinado contexto.
- Resolução de questões tratando significado semântico e sintático.
- Uso de grandes quantidades de conhecimento específico de um domínio particular.

2.2 HISTÓRICO

O primeiro trabalho agora reconhecido por tratar a IA foi um modelo de neurônios artificiais elaborado por Warren McCulloch e Walter Pitts, em 1943, que serviu como precursor da abordagem conexionista.

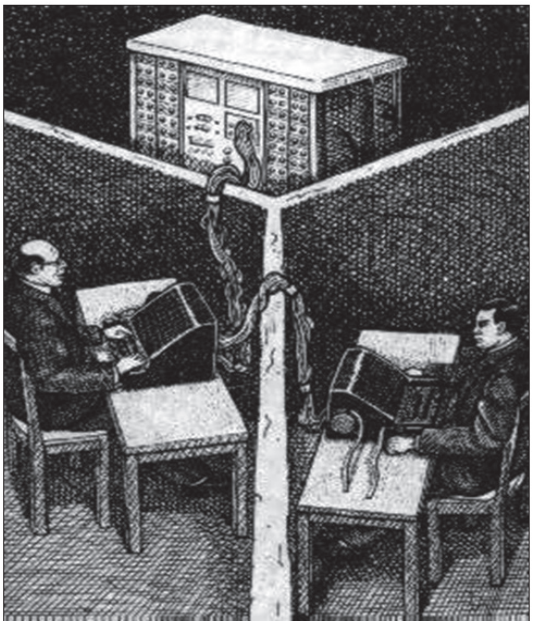
Um dos artigos mais conhecidos a tratar a questão da inteligência de máquinas computacionais foi escrito em 1950 pelo matemático britânico Alan Turing (Figura 1), conhecido por sua colaboração na decodificação da criptografia de mensagens do exército alemão na Segunda Guerra Mundial.

FIGURA 1 - MEMORIAL DE ALAN TURING

FONTE: Seebauer (2005)

O Teste de Turing ponderava sobre a possibilidade de uma máquina inteligente emular o comportamento de um ser humano. Turing propunha colocar um interlocutor conversando com duas entidades distintas sem a possibilidade de vê-las. Uma das entidades era uma pessoa e a outra uma máquina inteligente. O interlocutor deveria descobrir qual das duas entidades seria a máquina através de um interrogatório (Figura 2). Caso o interlocutor fosse ludibriado, dir-se-ia que a máquina seria inteligente. Existem diversas objeções ao Teste de Turing, ou seja, argumentos para refutar a hipótese de que o teste seria suficiente para determinar a inteligência de uma máquina. Maiores detalhes sobre as objeções feitas ao Teste de Turing podem ser obtidos em <http://www.ufrgs.br/alanturingbrasil2012/area5.html#2>.

FIGURA 2 - TESTE DE TURING



FONTE: Disponível em: <http://sti.br.inter.net/jferro/prec007.htm>. Acesso em: 17 jan. 2014.

Para conhecer mais detalhes sobre Alan Turing, importante figura da Ciência da Computação, recomendamos o filme Enigma. O filme trata sobre o processo de decodificação de mensagens criptografadas na Segunda Guerra Mundial e tem Alan Turing como personagem. Maiores detalhes sobre o filme podem ser obtidos em: <http://www.adorocinema.com/filmes/filme-26883/>.

O termo inteligência artificial foi cunhado por John McCarthy em um congresso sobre o tema realizado na Universidade de Dartmouth em 1956. O aspecto mais importante deste congresso foi o *networking*, ou seja, todos os grandes personagens que trabalhavam na área ficaram conhecendo seus pares e respectivos trabalhos.

Conforme Russel e Norvig (2004), os primeiros anos da IA foram repletos de sucessos. Considerando-se as ferramentas e computadores primitivos da época, o fato de os computadores serem capazes de efetuar atividades consideradas inteligentes e não somente operações aritméticas era surpreendente. Diversas atividades ditas como somente realizáveis por humanos por dependerem de certo grau de inteligência eram demonstradas uma a uma pelos pesquisadores da área.

Desde o início os pesquisadores de IA eram otimistas em seus prognósticos e previam a criação de um computador tão ou mais inteligente do que um humano em um período relativamente curto. O otimismo inicial logo cessou, na medida em que problemas mais complexos eram submetidos com fracasso aos sistemas de IA existentes. Um dos principais obstáculos era que as estruturas utilizadas para descrever o conhecimento nestes sistemas tinham poder de representação limitado, o que permitia somente a resolução de problemas que coubessem neste modelo de representação. A escassez de poder computacional da época também não ajudava os pesquisadores.

Russel e Norvig (2004) revelam que o quadro de resolução de problemas que surgiu durante as primeiras duas décadas de pesquisa em IA foi o de um mecanismo de busca de uso geral, que procurava reunir passos elementares de raciocínio para encontrar soluções completas. Estas abordagens foram chamadas de métodos fracos, pois embora gerais, não poderiam ter aumento de escala para instâncias de problemas grandes ou difíceis. Nesta época surgiram os sistemas especialistas e os sistemas de raciocínio baseados em casos, ambos caracterizados pela existência de um alto grau de conhecimento sobre determinado domínio. Surge então a abordagem simbólica.

Na década de 80, os sistemas especialistas começaram a se tornar viáveis para o mercado corporativo, sendo utilizados por diversas empresas, principalmente nos Estados Unidos e Japão. Em 1986, houve um retorno ao estudo das redes neurais por parte de diversos grupos de pesquisa que conseguiram recriar o algoritmo de aprendizagem por retropropagação, descoberto em 1969 por Bryson e Ho. Essa descoberta trouxe novamente à tona a abordagem conexionista, praticamente abandonada devido às dificuldades do final da década de 1960. Nesta década também surgiram os sistemas de inteligência artificial probabilística, chamados de redes bayesianas.

Informações adicionais sobre Thomas Bayes e sua teoria das probabilidades podem ser encontradas em: <http://www.icmc.usp.br/pessoas/francisco/SME0120/material/Thomas_Bayes_CH.pdf>.

Nos últimos anos, a IA procurou se utilizar das teorias existentes como base, ao invés de procurar por soluções completamente novas. O surgimento da internet, dos grandes volumes de dados e dos mecanismos de pesquisa deu um novo fôlego ao campo da IA. Atualmente, o conceito mais aceito é de um agente inteligente, em que as abordagens simbólicas e conexionistas podem trabalhar de forma colaborativa para a resolução de problemas através de um sistema computacional (um agente).

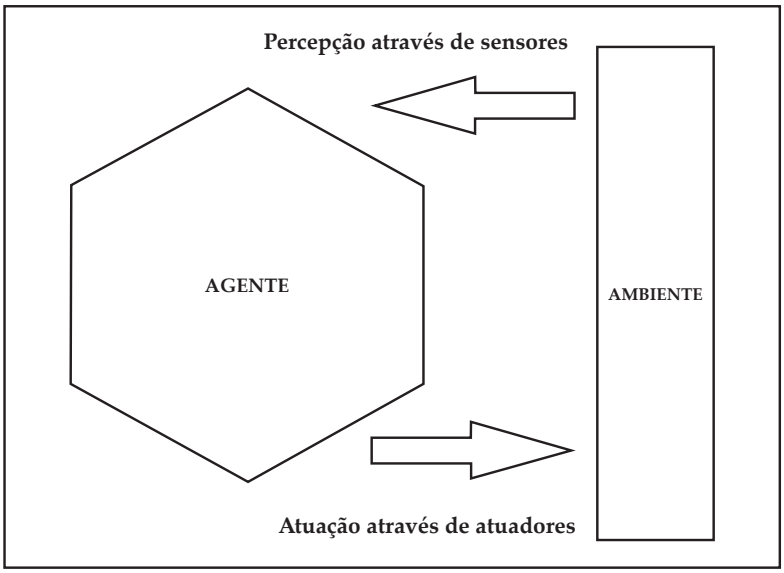
Mas o que a IA pode fazer hoje em dia? É difícil responder de forma concisa, porque existem atividades nas mais variadas áreas. A seguir são mostradas algumas das atividades-fim para as quais a IA vem sendo usada:

- Biometria – reconhecimento facial, impressão digital e de voz.
- Fiscalização de trânsito – reconhecimento de placas de veículos infratores.
- Jogos – *Deep Blue*, determinação do comportamento de avatares através de IA.
- Diagnóstico médico – sistemas especialistas diagnosticam doenças com base em regras.
- Controle autônomo – veículos que se dirigem sem interferência humana.
- Robótica – robôs que auxiliam cirurgiões em microcirurgias.
- Pesquisa – motores de busca na internet fazem uso de técnicas diversas de IA para aumentarem sua eficiência e grau de acerto.

2.3 AGENTES

O termo agente refere-se a algo que pode perceber seu ambiente por meio de sensores e de agir sobre este ambiente por intermédio de atuadores (RUSSEL; NORVIG, 2004). Wooldridge (1999, s.p.) define agente como: "Um agente é um sistema de computador que está situado em algum ambiente e que é capaz de executar ações autônomas de forma flexível neste ambiente, a fim de satisfazer seus objetivos de projeto". A Figura 3 ilustra um agente inteligente e sua interação com o ambiente.

FIGURA 3 - AGENTE INTELIGENTE

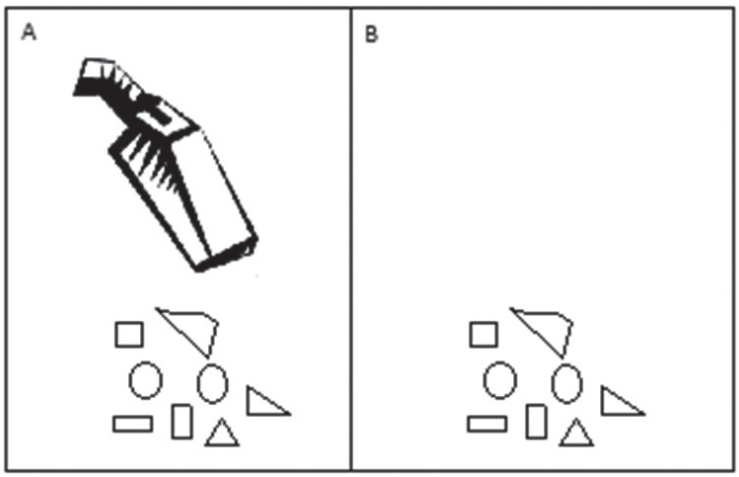


FONTE: O autor

Um agente humano, por exemplo, tem olhos e ouvidos como sensores e interage com o ambiente através de suas mãos, pernas, boca etc. Se considerarmos um agente computacional, como um sistema ou um robô, seus sensores seriam o teclado, o *mouse*, uma câmera ou microfone e este agiria sobre o ambiente através da exibição de resultados em algum dispositivo de saída, como monitor ou impressora, ou através de motores que o movimentariam no ambiente.

Para ilustrar essas ideias, Russel e Norvig (2004) descrevem um mundo fictício de um aspirador de pó com apenas dois locais: os quadrados A e B e apenas três ações possíveis: direita, esquerda, aspirar. O modelo pode ser visualizado na Figura 4 e as ações no Quadro 2.

FIGURA 4 - EXEMPLO DE ATUAÇÃO DE AGENTE



FONTE: O autor

QUADRO 2 - SEQUÊNCIA DE AÇÕES DO AGENTE

Sequência de Percepções	Ação
[A, limpo]	DIREITA
[A, sujo]	ASPIRAR
[B, limpo]	ESQUERDA
[B, sujo]	ASPIRAR

FONTE: Adaptado de Norvig e Russel (2004)

Neste exemplo, o agente inicialmente avalia o quadrado onde está e a condição do mesmo através de seus sensores (limpo ou sujo). Uma vez feita a avaliação, o agente toma a decisão e atua no ambiente através de seus atuadores, aspirando a sujeira ou movendo-se para a direita ou esquerda. Por exemplo, caso o agente entenda que o ambiente A esteja limpo, este desloca-se para a direita, buscando um ambiente que esteja sujo. Caso o agente esteja em B e perceba a existência de sujeira, a ação tomada é a de aspirar a sujeira.

Outra classificação comum dos agentes computacionais é de acordo com suas propriedades, conforme descrito no Quadro 3. Por exemplo, um agente é classificado como autônomo quando consegue tomar decisões para o cumprimento de seu objetivo sem a interferência do usuário final, e confiável quando suas informações são verídicas e suas ações em conformidade com o que foi predefinido.

QUADRO 3 - PROPRIEDADES DOS AGENTES INTELIGENTES

Propriedade	Descrição
Autonomia	Capacidade de tomar ações para o cumprimento de determinado objetivo sem a interferência do usuário final.
Aprendizagem	Capacidade de alterar seu comportamento com base em experiências prévias.
Comunicabilidade	Capacidade do agente de se comunicar com outros agentes.
Confiabilidade	Capacidade do agente em demonstrar veracidade nas informações mostradas e ações realizadas em nome do usuário.
Cooperatividade	Capacidade de mais de um agente agirem em conjunto buscando um objetivo comum.
Degradação	Capacidade de um agente completar a tarefa mesmo se alguma anomalia no sistema estiver ocorrendo.
Flexibilidade	Habilidade dos agentes em alterar o fluxo de execução em decorrência de algum evento do ambiente.
Inteligência	Capacidade de negociar com ambiguidades.
Persistência	Capacidade de o agente manter um estado interno conciso com o passar do tempo.

Proatividade	Não somente respondem ao ambiente, mas perseguem um objetivo.
Personalização	Capacidade de o agente aprender com o usuário e adaptar suas ações de acordo com ele.

FONTE: Adaptado de Faraco (1998)

2.4 AMBIENTES

Os ambientes de tarefas são basicamente os problemas para os quais os agentes devem encontrar as soluções. No exemplo do aspirador, nosso ambiente era composto por dois quadrados A e B, sendo que ele poderia estar sujo ou limpo. A variedade de ambientes de tarefas que podem surgir em IA é bastante variada e por esse motivo procuramos classificá-los por algumas propriedades específicas. Russel e Norvig (2004) definem os seguintes tipos de ambientes:

1) Completamente observável e completamente observável – se os sensores de um agente conseguem definir o estado completo do ambiente em cada instante, o ambiente da tarefa é considerado completamente observável. A vantagem deste tipo de ambiente é que o agente não precisa acessar informações anteriores do ambiente ou manter algum tipo de memória auxiliar para armazenar cada estado. Um exemplo de ambiente completamente observável seria um jogo de dominó (Figura 5). Em cada jogada o jogador consegue perceber quantas e quais pedras estão na mesa e tomar sua decisão com base nisso. A memória das jogadas anteriores não influencia o ambiente atual e, consequentemente, a atuação do agente no mesmo.

FIGURA 5 - AMBIENTE COMPLETAMENTE OBSERVÁVEL

FONTE: Microsoft Clipart (2010)

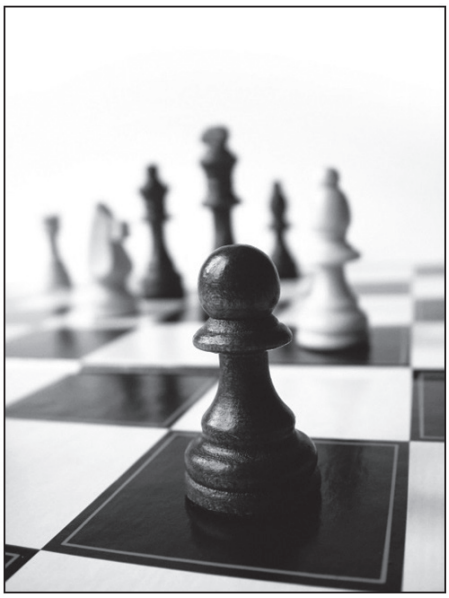
Um ambiente parcialmente observável é aquele em que os sensores definem somente o estado parcial do ambiente e a definição do estado completo depende de algum tipo de memória auxiliar. Podemos citar como exemplo um jogo de pôquer (Figura 6). O fato de determinado jogador ter pego cartas em um estado anterior do ambiente pode influenciar na decisão a ser tomada pelo agente, ou seja, as jogadas anteriores influenciam o estado atual do ambiente e, consequentemente, a atuação do agente sobre ele. Neste tipo de ambiente, o agente deve empregar algum tipo de memória auxiliar para armazenar os estados anteriores relevantes.

FIGURA 6 - AMBIENTE PARCIALMENTE OBSERVÁVEL

FONTE: Microsoft Clipart (2010)

2) Determinístico e estocástico – se o próximo estado do ambiente pode ser determinado pelo estado atual e pela ação executada pelo agente, dizemos que o ambiente é determinístico. Um exemplo seria um jogo de xadrez, onde caso um jogador mova uma peça do local a1 de um tabuleiro para o local a2 do mesmo tabuleiro n vezes, o resultado será sempre o mesmo (Figura 7). Em um ambiente estocástico, a mudança do estado atual para o próximo estado por ação do agente não pode ser prevista. Como exemplo podemos citar um jogo de dados, onde não é possível prever o resultado de cada jogada (Figura 8).

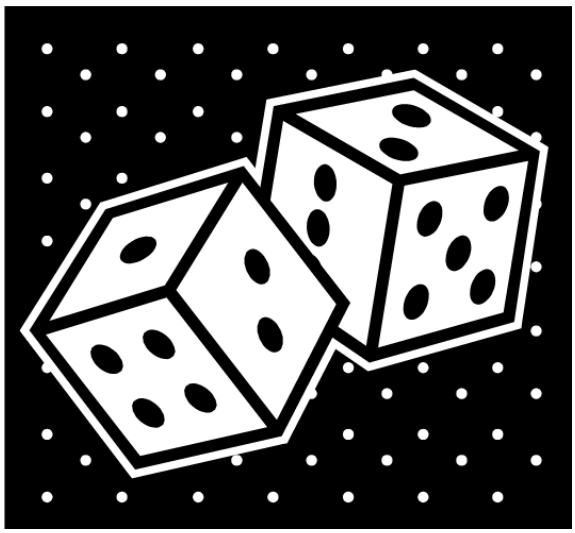
FIGURA 7 - AMBIENTE DETERMINÍSTICO



FONTE: Microsoft Clipart (2010)

Em 1997, a IBM lançou um desafio ao então campeão mundial de xadrez Garry Kasparov. O desafio consistia em enfrentar o supercomputador "inteligente" DEEP BLUE, elaborado com a finalidade de jogar xadrez. Para mais detalhes sobre os jogos entre Gary Kasparov e DEEP BLUE, acesse o *site*: <https://www.research.ibm.com/deepblue/>.

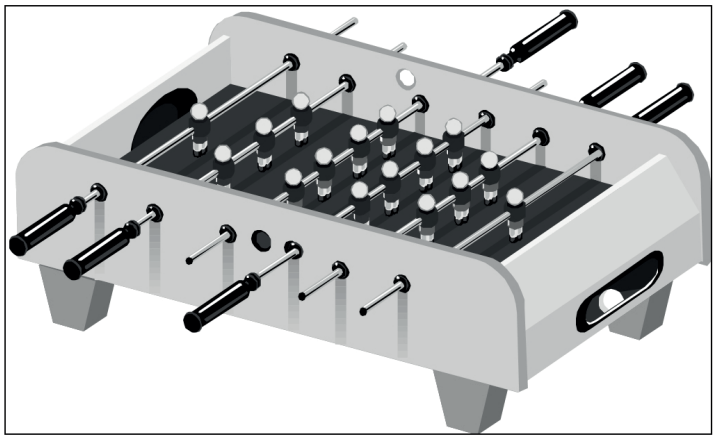
FIGURA 8 - AMBIENTE ESTOCÁSTICO



FONTE: Microsoft Clipart (2010)

3) Estático e dinâmico – se o ambiente se alterar durante a atuação do agente, dizemos que ele é dinâmico. Um agente que tentasse simular o comportamento de um motorista no trânsito estaria em um ambiente dinâmico, pois todos os demais elementos do ambiente se movimentam ao mesmo tempo e de forma independente. Outro exemplo seria uma mesa de pebolim, onde cada jogador deve observar continuamente o movimento do outro para decidir o que fazer, conforme ilustrado na Figura 9.

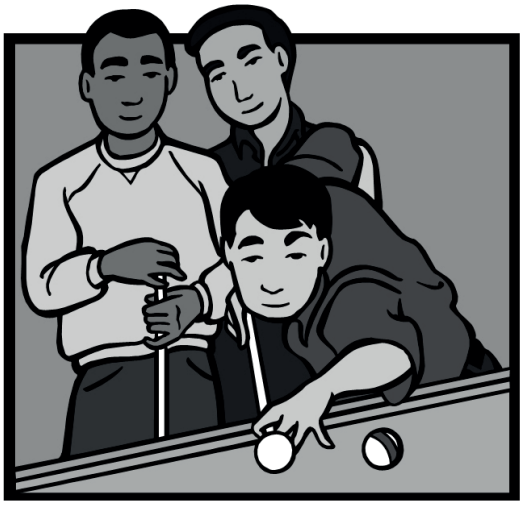
FIGURA 9 - AMBIENTE DINÂMICO



FONTE: Microsoft Clipart (2010)

Em um ambiente estático, cada elemento que o compõe não sofre alteração durante a atuação do agente. Nestes ambientes, a manipulação é mais fácil, pois o agente não precisa fazer observação contínua ou se preocupar com a passagem do tempo enquanto decide sua atuação. Como exemplo, podemos citar o jogo de sinuca onde um jogador joga de cada vez (Figura 10).

FIGURA 10 - AMBIENTE ESTÁTICO



FONTE: Microsoft Clipart (2010)

4) Discreto e contínuo – a distinção entre discreto e contínuo pode se aplicar ao estado do ambiente, ao modo como o tempo é tratado e às percepções e ações do agente. Um ambiente de estados é considerado discreto quando o número total de estados pode ser contado, por exemplo, um jogo de xadrez. Apesar de existirem diversas possibilidades, as peças em um jogo de xadrez possuem um número finito de posições disponíveis, o que torna o ambiente discreto. Por outro lado, um agente que simulasse o comportamento de um motorista no trânsito estaria em um ambiente contínuo, pois o número de possibilidades para a posição deste veículo em relação à velocidade, à rotação do volante e aos outros veículos é praticamente infinito.

5) Agente único e multiagente – um agente que procurasse resolver um jogo de palavras cruzadas estaria em um ambiente de agente único, enquanto um agente que jogasse pingue-pongue (Figura 11) estaria em um ambiente de dois agentes. Outro aspecto a ser considerado é a maneira como os agentes se comportam em relação um ao outro. No caso do jogo de pingue-pongue, cada agente tentaria derrotar o outro, o que o classifica como um ambiente multiagente competitivo.

FIGURA 11 - AMBIENTE MULTIAGENTE COMPETITIVO

FONTE: Microsoft Clipart (2010)

Podemos citar um time de futebol como exemplo de ambiente multiagente cooperativo (Figura 12). Neste caso, considerando somente um time, todos os agentes estão cooperando para um resultado comum. Ao aumentar o escopo e considerar os dois times, estaríamos em um ambiente multiagente competitivo.

FIGURA 12 - AMBIENTE MULTIAGENTE COOPERATIVO

FONTE: Creative Commons (2014)

A determinação do tipo de ambiente onde um agente se enquadra nem sempre é tão simples. A maioria das situações reais se enquadra em mais de uma categoria. O exemplo do motorista, que foi citado anteriormente, poderia ser classificado como parcialmente observável, estocástico, dinâmico, contínuo e multiagente competitivo. O Quadro 4 mostra alguns exemplos de ambientes e sua classificação de acordo com os critérios supracitados.

QUADRO 4 - EXEMPLOS DE AMBIENTES DE TAREFAS E SUAS CARACTERÍSTICAS

Ambiente	**Observável**	**Determinístico**	**Estático**	**Discreto**	**Agentes**
Palavras cruzadas.	Completamente	Determinístico	Estático	Discreto	Único
Xadrez com relógio.	Completamente	Estocástico	Estático	Discreto	Multi
Pôquer	Parcialmente	Determinístico	Estático	Discreto	Multi
Direção de táxi.	Parcialmente	Estocástico	Dinâmico	Contínuo	Multi
Diagnóstico médico.	Parcialmente	Estocástico	Dinâmico	Contínuo	Único
Análise de imagens.	Completamente	Determinístico	Estático	Contínuo	Único
Instrutor interativo de idiomas.	Parcialmente	Estocástico	Dinâmico	Contínuo	Multi/Único

FONTE: Adaptado de Norvig e Russel (2004)

RESUMO DO TÓPICO 1

Neste tópico você aprendeu que:

- A dificuldade em se definir a IA se transforma na dificuldade em definir o significado do termo inteligência propriamente dito.
- Uma das divisões mais aceitas da IA é a que trata de sistemas que tentam pensar e agir como humanos e sistemas que tentam pensar e agir racionalmente, sendo que racionalmente, neste caso, significa CORRETO.
- O primeiro trabalho que tratou de IA foi um modelo de neurônios desenvolvido por Warren McCulloch e Walter Pitts em 1943.
- O Teste de Turing é um dos testes mais conhecidos para determinar se um computador é ou não inteligente.
- Algumas das principais áreas em que a IA vem sendo utilizada atualmente são: biometria, fiscalização de trânsito, jogos, diagnóstico médico, controle autônomo, robótica e pesquisa na internet.
- O termo agente refere-se a algo que percebe o ambiente por seus sensores e atua sobre o mesmo, modificando o seu estado através de atuadores.
- Os ambientes de tarefas são basicamente os problemas que os agentes devem resolver.
- Os ambientes podem ser: completamente ou parcialmente observáveis, determinísticos ou estocásticos, estático ou dinâmico, discreto ou contínuo, agente único ou multiagente e cooperativo ou competitivo.

AUTOATIVIDADE

1 Considerando um agente que simulasse o comportamento de um jogador de dominó, avalie as seguintes afirmativas:

I – Este agente está em um ambiente completamente observável.
II – Este agente está em um ambiente estocástico.
III – Este agente está em um ambiente cooperativo.
IV – Este agente está em um ambiente estático.

São verdadeiras as afirmativas:

a) I, II e III.
b) I e III.
c) I, II e IV.
d) I e IV.

2 Com relação aos conceitos de inteligência artificial relacionados a agente e ambiente, avalie as afirmações a seguir:

I – Um agente é um elemento que percebe um ambiente através de seus sensores e atua neste ambiente com seus atuadores, após tomar uma decisão.
II – O ambiente é basicamente o problema a ser resolvido pelo agente.
III – Como exemplo de ambiente determinístico, discreto e competitivo podemos citar um jogo de tênis.
IV – Um ambiente pode ser cooperativo e competitivo ao mesmo tempo, dependendo do ponto de vista.

Agora assinale a alternativa que somente contém afirmações verdadeiras:

a) I, II, III e IV.
b) II, III e IV.
c) I, II e IV.
d) I e IV.

3 Nos últimos anos, a inteligência artificial tem buscado evoluir sobre as teorias existentes ao invés de buscar soluções completamente novas. Em parte, isso se deve ao fato de hoje possuirmos poder computacional e recursos suficientes para aplicar técnicas que antigamente eram impensáveis devido principalmente às restrições de *hardware*. Em relação à utilização da inteligência artificial hoje, avalie as afirmações a seguir:

I - A biometria é uma das áreas mais beneficiadas com a inteligência artificial.
II - Os sistemas de diagnóstico médico auxiliado por computador foram praticamente abandonados, especialmente devido ao grau de incerteza dos diagnósticos fornecidos.

III - Graças aos recursos praticamente infinitos das grandes empresas.com, a inteligência artificial tem avançado muito nos últimos anos.
IV - Uma das aplicações da inteligência artificial que efetivamente utilizamos no dia a dia são as pesquisas na internet.

Agora assinale a alternativa que somente possui afirmações CORRETAS:

a) I, II e IV.
b) I, II e III.
c) I, III e IV.
d) II, III e IV.

4 Nos primeiros anos de seu surgimento, a ciência conhecida como inteligência artificial gerou polêmica nos mais diversos campos, essencialmente pelo nome adotado para defini-la e por sua multidisciplinaridade característica, que tornou complexa a filiação em somente uma área do conhecimento. Com relação aos primeiros anos da inteligência artificial, assinale a alternativa CORRETA:

a) O Teste de Turing consistia em uma série de problemas que eram resolvidos pela máquina de Turing através de técnicas de inteligência artificial.
b) Os primeiros anos da inteligência artificial foram repletos de insucessos, principalmente pela dificuldade de resolver operações aritméticas complexas.
c) Na década de 80, os sistemas especialistas passaram a ser considerados viáveis para utilização no mercado empresarial.
d) Nos últimos anos, a inteligência artificial vem buscando teorias e técnicas completamente novas, de forma a abandonar as descobertas feitas no passado.

5 Para o cientista emérito da computação, Edsger Dijkstra, "Perguntar se uma máquina pode pensar é o mesmo que perguntar se um submarino pode nadar". Essa abordagem refere-se à classificação dos sistemas de inteligência artificial que considera o raciocínio, conhecida como "Sistemas que pensam racionalmente". Com relação ao exposto acima, avalie as afirmações abaixo, colocando V para as afirmações VERDADEIRAS e F para as afirmações FALSAS:

( ) A abordagem racionalista para classificação dos sistemas de inteligência artificial envolve hipóteses e confirmação através de experimentos empíricos.
( ) Problemas que não são passíveis de resolução através de algoritmos tradicionais são bons candidatos à aplicação de técnicas de inteligência artificial.
( ) A abordagem racionalista para classificação dos sistemas de inteligência artificial envolve engenharia e matemática para confirmação.
( ) A inteligência artificial é especialmente destinada para resolver problemas fortemente estruturados, ou seja, onde a informação sobre o problema está completamente definida.
( ) A abordagem centrada nos seres humanos para classificação dos sistemas de inteligência artificial envolve hipóteses e confirmação através de experimentos empíricos.

Agora assinale a alternativa que apresenta a sequência CORRETA:

F, V, V, F, V.
V, F, V, F, V.
F, F, V, V, F.
V, V, V, F, F.

TÓPICO 2

REPRESENTAÇÃO DO CONHECIMENTO E RESOLUÇÃO DE PROBLEMAS ATRAVÉS DA BUSCA

1 INTRODUÇÃO

Para que exista inteligência, pressupõe-se algum tipo de conhecimento. Você verá algumas das técnicas mais utilizadas para a representação do conhecimento nos sistemas ditos inteligentes. Demonstraremos essas técnicas através de exemplos práticos e exercícios, destacando os pontos fortes e fracos de cada uma. A aplicabilidade destas representações do conhecimento em técnicas de inteligência artificial será vista com mais detalhe nas unidades 2 e 3.

Mostraremos ainda, neste tópico, que diversos problemas da inteligência artificial podem ser resolvidos utilizando espaços de busca. Evidenciaremos a eficiência desta estratégia e a semelhança que possui com a estratégia que nós humanos adotamos para a resolução de problemas, mesmo que de forma inconsciente.

Para encerrar o tópico, abordaremos heurísticas e detalharemos os algoritmos de busca em profundidade e busca em largura com vistas às diferenças entre ambos.

2 REPRESENTAÇÃO DO CONHECIMENTO

Uma das principais dificuldades na resolução de problemas através de IA é a representação do conhecimento. Para Lugger (2002), a representação da informação para ser usada na solução inteligente de problemas oferece desafios importantes e difíceis que se encontram no âmago da IA. Fernandes (2003) afirma que para resolver os problemas mais complexos encontrados na IA, é necessária uma grande quantidade de conhecimentos representados pelo computador e certos mecanismos para manipulá-los.

Coppin (2010) complementa dizendo que para um computador estar apto a resolver um problema relacionado ao mundo real, ele primeiro precisa de um meio para representar aquela situação real internamente. Ao lidar com aquela representação interna, o computador torna-se capaz de solucionar problemas.

Rich (1993) coloca cinco características que o conhecimento representado para um problema de IA deve possuir:

1) As situações de generalização, onde elementos compartilham propriedades importantes, são agrupadas para evitar consumo excessivo de memória e atualização.
2) O conhecimento deve ser passível de compreensão pela pessoa que o fornece ou que avalia o resultado.
3) Deve ser facilmente modificado para corrigir erros e refletir mudanças do mundo e da visão do mundo que o usuário possui.
4) Deve ser usado em inúmeras situações, mesmo que não seja totalmente preciso, nem esteja completo.
5) Sirva de ajuda para superar seu próprio volume, auxiliando a limitar as várias possibilidades que em geral devem ser consideradas.

O modo como um computador representa um problema, as variáveis que ele usa e os operadores aplicados a essas variáveis podem fazer a diferença entre um algoritmo eficiente e um algoritmo que não funciona, especialmente para problemas que utilizam a busca (COPPIN, 2010). Concluindo, uma representação eficiente do conhecimento deve ser concisa, facilmente armazenável e recuperável, legível pelo especialista e passível de representação de restrições, de forma a facilitar o trabalho do motor de inferência. A seguir apresentamos algumas das principais formas através das quais o conhecimento é representado em sistemas de IA.

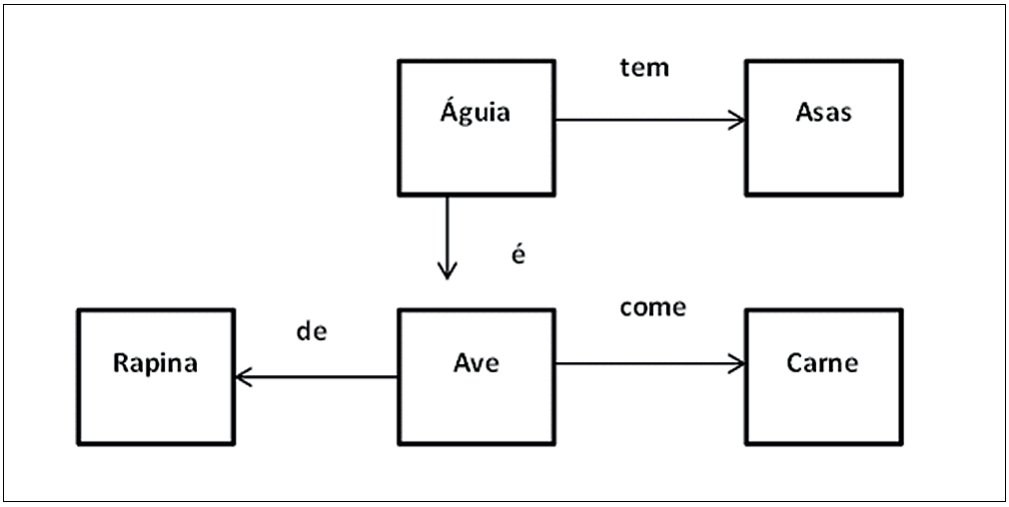
2.1 REDES SEMÂNTICAS

Uma representação de conhecimento bastante utilizada na IA são as redes semânticas. Uma rede semântica consiste em um conjunto de nodos conectados por arestas onde os nodos representam objetos e as arestas representam a relação entre eles. Uma rede semântica essencialmente é um **GRAFO**.

Grafos são estruturas utilizadas para as mais diversas finalidades dentro da computação. Como exemplo, podemos citar os sistemas de GPS. Para saber mais sobre os grafos e suas propriedades, acesse o *link*: <http://www.ime.usp.br/~pf/teoriadosgrafos/texto/TeoriaDosGrafos.pdf>.

Digamos que exista a necessidade de representar as características de um objeto águia, indicando que ela tem asas, é uma ave de rapina e come carne. A rede semântica para ilustrar essa situação está ilustrada na Figura 13.

FIGURA 13 - EXEMPLO DE REDE SEMÂNTICA



FONTE: O autor

O quadrado com a palavra Águia é o que chamamos de NODO. O relacionamento do nodo Águia com o nodo Asas é representado por uma aresta com a palavra TEM, ou seja, pode-se inferir, através da rede semântica, que a águia tem asas. O tipo de relacionamento pode variar, como podemos ver nas arestas que indicam uma ação da águia (come) e o que ela é (é).

Apesar de serem de fácil leitura e compreensão, especialmente pelo usuário final, as redes semânticas apresentam algumas desvantagens:

- Granularidade muito fina.
- Dificuldade de manutenção de consistência semântica em domínios complexos.
- Falta de um mecanismo de inferência naturalmente associado à representação do conhecimento.

2.2 ROTEIROS

Coppin (2010) define roteiro como uma estrutura de dados usada como uma representação estruturada para uma situação que pode ser desmembrada em uma sequência de eventos. Para Luger (2002), os roteiros são bastante usados em sistemas de compreensão de linguagem natural para organizar uma base de conhecimento em termos de situações que o sistema deve compreender. A utilização de roteiros em IA baseia-se na ideia de que uma dada situação usualmente possui um conjunto finito de conhecimento necessário para que se determine de que forma agir.

Um roteiro é geralmente composto por:

- Condições de entrada: precondições do mundo que devem ser verdadeiras para que o roteiro seja chamado.
- Resultados: fatos que devem ser verdadeiros ao final da execução do roteiro.
- Acessórios: elementos que compõem o ambiente a ser descrito.
- Papéis: determina as ações que os participantes realizam.
- Episódios: unidades temporais dos acontecimentos dos roteiros.

Através da combinação de entidades e papéis, um roteiro seria capaz de responder a perguntas sobre determinadas situações que não estejam explícitas na descrição do mesmo. Coppin (2010) utiliza a pequena história para exemplificar a atuação de um roteiro:

"Fred foi ao seu restaurante favorito. A comida não estava tão boa como de costume. No caminho para casa, percebeu que havia esquecido sua carteira".

O sistema de roteiro seria capaz de identificar que Fred é o cliente e que existem uma garçonete e um cozinheiro envolvidos na situação. Questões como "Fred comeu no restaurante?" seriam respondidas por associação.

É importante destacar que os roteiros são bastante específicos e que seu funcionamento depende muito da criação de episódios. No caso do restaurante mostrado acima, poderíamos ter episódios para as ações de entrar, fazer o pedido, pagar a conta, ir ao banheiro etc.

O Quadro 5 ilustra um exemplo de roteiro para a atividade de comer em um restaurante.

QUADRO 5 - EXEMPLO DE ROTEIRO

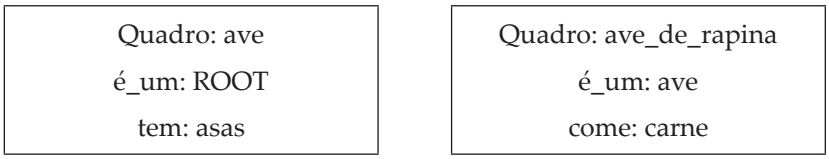
1. Roteiro: comer-em-restaurante 2. Apoio: (restaurante, dinheiro, alimento, menu, mesas, cadeiras) 3. Funções: (pessoas com fome, encontro de pessoas) 4. Ponto-de-vista: (pessoas com fome) 5. Tempo-de-ocorrência: (tempo-de-operação do restaurante) 6. Lugar-de-ocorrência: (localização do restaurante) 7. Sequência-de-eventos 8. Primeiro: Inicie o roteiro entrada-no-restaurante 9. Então: Se (há-convite-para-sentar ou reservas) então siga roteiro orientação-do-garçom 10. Então: siga roteiro aguarde-sentado 11. Então: siga roteiro solicite-comida 12. Então: siga roteiro comer, a menos que haja uma longa espera, caso em que seguirá o roteiro sai-do-restaurante-furioso.

FONTE: Adaptado de Silveira e Rosenberg (2007)

2.3 QUADROS

Os quadros (*frames*) foram introduzidos para permitir a expressão das estruturas internas dos objetos, mantendo a possibilidade de representar herança de propriedades (BITTENCOURT, 2006). São especialmente práticos para utilização em sistemas especialistas. Podemos dizer que essencialmente um quadro é uma estrutura que tem um nome e um conjunto de pares atributo-valor. A Figura 14 ilustra o mesmo exemplo da Águia visto anteriormente, mas desta vez representado por quadros.

FIGURA 14 - QUADROS



FONTE: O autor

O quadro à direita apresenta três propriedades: *nome*, *é um* e *tem*. Ele visa representar uma estrutura mais genérica para qualquer ave, informando que é o quadro raiz da relação e que tem asas. A utilização da herança, neste caso, se dá pela propriedade asas, uma vez que todas as aves têm asas. Todas as instâncias de um quadro herdam suas propriedades, valores e restrições. O quadro à esquerda é uma especialização do quadro ave, indicando que se trata de uma ave de rapina que come carne. As propriedades de um quadro em geral têm um tipo específico (*string*, booleano, inteiro, real etc.) no sentido de facilitar sua representação posterior no sistema. O relacionamento entre os quadros é definido pela propriedade *é_um.*

Para a construção de um sistema de quadros, definem-se as propriedades que caracterizam a categoria correspondente. As propriedades normalmente definidas para os quadros são:

- Nome.
- Classe em que está contido.
- Atributos que qualificam a categoria e todas as suas subclasses.
- Tipos de dados e intervalos que os atributos podem assumir.
- Relações entre o quadro e os demais do sistema.

A principal desvantagem da utilização dos quadros é que em geral as estruturas que comportam o processamento e as regras a serem aplicadas às propriedades não estão neles mesmos, aumentando a complexidade da solução.

2.4 REGRAS DE PRODUÇÃO

As regras de produção resultam da transformação de um problema computacional em um grafo de estados que possui um estado inicial e um ou n estados finais, sendo estes identificáveis quando atingidos.

Luger (2002) define as regras de produção como um sistema que fornece controle guiado por um processo de solução de problemas, consistindo em um conjunto de *regras*, uma *memória de trabalho* e um *ciclo de controle*.

Araújo (2004) afirma que as *regras* representam conhecimentos heurísticos tipicamente na forma. Se (condições) Então (conclusões). Muitos sistemas especialistas utilizam regras de produção aliadas a uma máquina de inferência para realizar a sua tarefa. Como exemplo de regras de produção, considere o Quadro 6, onde é mostrada a clássica representação de conhecimentos sobre alguns animais.

QUADRO 6 - EXEMPLO DE REGRAS DE PRODUÇÃO

1. Se (produz_leite∧tem_pelos) Então (mamífero)
2. Se (mamífero ∧come_carne) Então (carnívoro)
3. Se (mamífero ∧possui_presas∧possui_garras) Então (carnívoro)
4. Se (mamífero ∧possui_casco) Então (ungulado)
5. Se (carnívoro ∧ pardo ∧ pintado) Então (onça)
6. Se (carnívoro ∧ pardo ∧ listrado) Então (tigre)
7. Se (ungulado ∧ pardo ∧ pintado) Então (girafa)
8. Se (ungulado ∧ branco ∧ listrado) Então (zebra)

FONTE: Adaptado de Araújo (2004)

A regra Se (produz_leite∧tem_pelos) Então (mamífero) pode ser entendida como uma instanciação de x [(produz_leite(x) ∧t em_pelos(x)) → mamífero(x)]. A partir de fatos conhecidos sobre um dado animal A, do tipo {tem_pelos(A), produz_leite(A), come_carne(A), pardo(A), pintado(A)}, pode-se concluir que A é uma (onça). Quando se busca verificar qual a conclusão a ser obtida a partir dos dados, diz-se que o encadeamento é progressivo (ARAÚJO, 2004).

A memória de trabalho armazena o estado atual do mundo em um processo de raciocínio. Quando o elemento da condição de uma regra se encontra com o conteúdo da memória de trabalho, a ação associada à condição pode ser realizada (LUGER, 2002).

Finalmente, o *ciclo de controle* é definido como o agente que dispara as regras e modifica o conteúdo da memória de trabalho. Após o disparo das regras de produção, o ciclo de controle se repete usando o conteúdo modificado da memória de trabalho até que o conteúdo da memória de trabalho não se encontrar em nenhuma condição de regra.

2.5 REPRESENTAÇÃO PROCEDIMENTAL

Para utilizar uma representação procedimental, é preciso estender sua funcionalidade com um interpretador que siga as instruções fornecidas no conhecimento. O conhecimento procedimental passa por um conjunto de etapas sequenciais para encontrar soluções para problemas, sendo usualmente representado através de regras encadeadas ou árvores de decisão.

Rich (1993) destaca as principais vantagens da representação procedimental do conhecimento:

- A facilidade em representar o conhecimento de como fazer as coisas.
- A facilidade em representar o conhecimento que não se enquadra dentro de muitos esquemas declarativos simples, por exemplo, o raciocínio por omissão e o raciocínio probabilístico.
- A facilidade em representar o conhecimento heurístico de como fazer eficientemente as coisas.

No Quadro 7 apresentamos um exemplo de representação procedimental do conhecimento.

QUADRO 7 - REPRESENTAÇÃO PROCEDIMENTAL DO CONHECIMENTO

```
(José (é_um (pessoa))
                    (nascimento (dia 27)
                                (mês maio)
                                (ano 1979))
                    (altura 1.78)
                    (idade (calcular (agora - (ano de nascimento))))
```

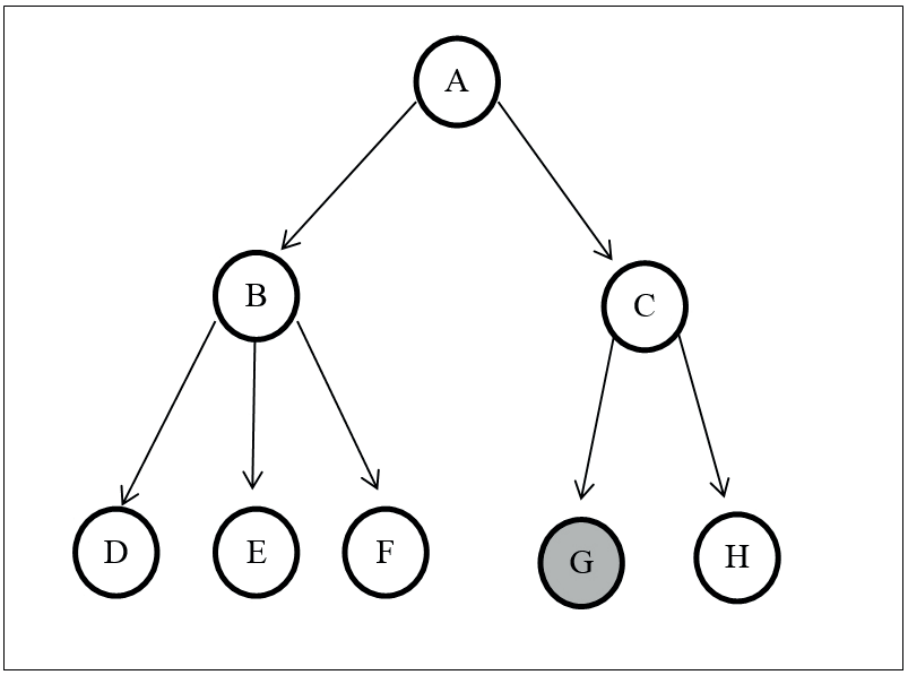
FONTE: O autor

3 REPRESENTAÇÃO DE PROBLEMAS ATRAVÉS DA BUSCA

Coppin (2010) destaca que muitos problemas em IA podem ser representados como um espaço de busca, onde este espaço é uma representação do conjunto de possíveis escolhas de um dado problema e uma ou mais destas escolhas é a

solução. Luger (2002) afirma que essa estratégia é apoiada por uma visão de senso comum de resolução de problemas por seres humanos, em que geralmente são consideradas várias alternativas diferentes sobre o modo de resolver um problema. Árvores são estruturas de dados geralmente utilizadas para essa representação.

FIGURA 15 - ESPAÇO DE BUSCA

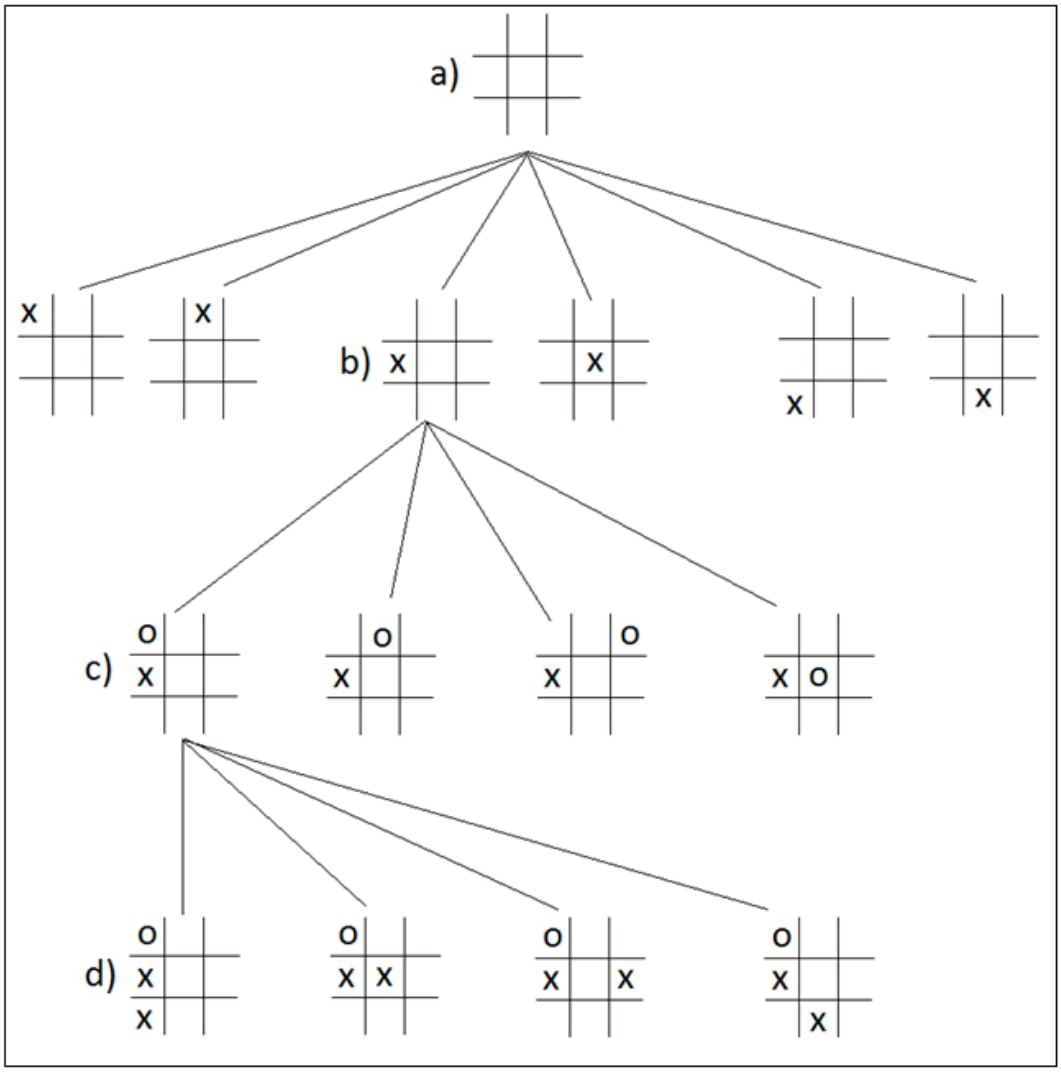


FONTE: O autor

Na Figura 15 está ilustrado um exemplo de espaço de busca em árvore. O estado inicial é representado pela letra *A*, e a partir deste estado pode-se optar pelos caminhos *B* ou *C*, que representam soluções parciais do problema. Se a solução adotada for representada por *B*, a partir daí existem as possibilidades *D*, *E* e *F*. Se a solução adotada for a representada por *C*, pode-se optar pelas soluções *G* ou *H*. Na figura temos a letra *G* destacada em cinza, identificando uma resposta para o espaço de busca. Neste caso, a solução do problema estaria em seguir o caminho *A*, *C* e finalmente *G*.

Para exemplificar melhor, consideremos como espaço o jogo da velha (Figura 16). Em qualquer configuração do tabuleiro existe um número finito de movimentos que um jogador pode fazer. Não foram colocadas todas as jogadas possíveis por motivo de simplificação.

FIGURA 16 - ESPAÇO DE BUSCA DE UM JOGO DA VELHA



FONTE: O autor

Na figura, temos destacado pela letra a o estado inicial da busca, onde nenhum jogador fez qualquer movimento ainda. Considerando que o primeiro jogador é o X, existem nove possibilidades distintas de jogada para ele. A letra b destaca a jogada do primeiro jogador. A partir deste momento, a busca reduz o número de possibilidades da próxima jogada, uma vez que todas as possibilidades devem agora partir daí e seguir este caminho no grafo. Existem somente oito possibilidades de jogada para o próximo jogador, uma vez que um quadrante do tabuleiro já foi preenchido. O segundo jogador, representado pelo O, faz sua jogada, destacada pela letra c. A partir daí somente existem sete possibilidades para o próximo jogador e estas devem respeitar as jogadas anteriores. Novamente o jogador representado pelo X faz sua jogada e o número de possibilidades reduz novamente. Essa representação nos permite tratar todos os jogos possíveis do jogo da velha como caminhos através deste grafo. Uma estratégia de jogo vencedora fará uma busca no grafo por caminhos que anteriormente conduziram ao número maior de vitórias e menor de derrotas. O caminho definido pela busca é a solução do problema.

Os seres humanos usam um processo de busca semelhante para a resolução de problemas. Por exemplo, um jogador de xadrez examina diversas possibilidades de jogada antes de efetivamente realizar seu movimento. Um médico analisa vários diagnósticos possíveis antes de decidir, um programador avalia diversos algoritmos e estruturas antes de efetivamente implementar a solução.

A principal diferença entre o mecanismo de busca mostrado no exemplo do jogo da velha, passível de implementação em um sistema computacional, e o utilizado pelos seres humanos é que nos seres humanos a resolução de problemas não considera todas as opções possíveis. Aparentemente, resolvemos problemas através de regras de julgamento que guiam nossa busca pelo espaço de estados considerando apenas as opções consideradas mais promissoras. Por exemplo, se você chegar a um mecânico dizendo que seu carro está com problema de aquecimento no motor, dificilmente ele avaliará o estado de seus pneus, apesar de isso ser parte do espaço de busca existente.

Essas regras são conhecidas como heurísticas (a palavra vem do grego “descobrir”) e constituem um dos tópicos centrais da pesquisa em IA. Uma heurística é uma estratégia para a busca seletiva de um espaço de problema. Ela guia nossa busca ao longo de linhas que têm alta probabilidade de sucesso, evitando esforço desnecessário (NORVIG; RUSSEL, 2004). Fernandes (2008) define heurística como um meio de solucionar problemas complexos, para os quais não há disponível uma abordagem mais direta. A seguir apresentamos algumas estratégias de busca utilizadas na IA.

3.1 BUSCA CEGA

A busca cega (também chamada de busca exaustiva ou busca de força bruta) é considerada a mais simples abordagem de busca, pois envolve simplesmente visitar cada nó do espaço de busca e testá-lo para verificar se é um nó objetivo. Se for, a busca teve sucesso e não precisa ser levada adiante (COPPIN, 2010).

Essa estratégia de busca é normalmente utilizada quando não existe conhecimento adicional sobre o problema, além de percorrer a árvore e identificar no objetivo. Um exemplo seria você precisar encontrar a casa de seu amigo em determinada rua que você não conhece. A única informação disponível é que a casa dele é a casa de número 241. Você buscaria casa por casa (nó) até encontrar a casa que seria identificada como a de seu amigo (nó objetivo).

Coppin (2010) ainda define que essa estratégia de busca deve possuir um gerador adequado que satisfaça a duas propriedades:

1) Ele deve ser completo e gerar (buscar) todas as soluções possíveis, caso contrário, uma solução adequada poderia não ser encontrada. No exemplo da casa de nosso amigo, isso seria o equivalente a pular algumas casas da rua sem observar o número dela.
2) Não se deve buscar a mesma solução mais de uma vez. Em geral isso é conseguido através da utilização de algum tipo de memória de trabalho, onde são armazenados os nós já gerados (visitados). No nosso exemplo, isso seria como verificar mais de uma vez a mesma casa procurando pelo número 241.

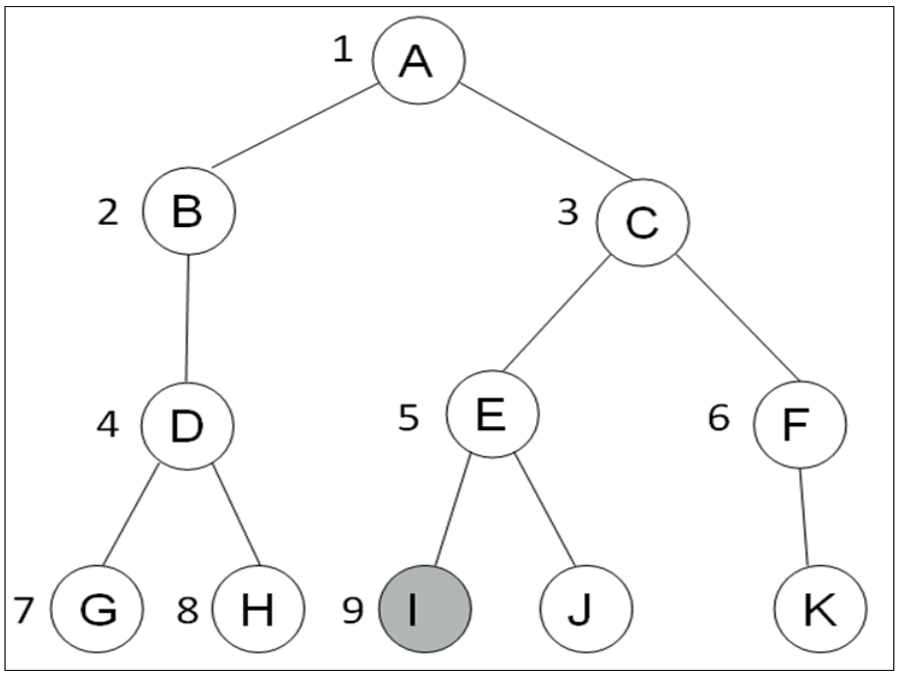
Ele deve restringir a busca ao espaço de busca pertinente ao problema informado. No exemplo citado acima, não buscaríamos a casa de nosso amigo em uma rua diferente da que nos foi informada, pois isso seria fugir do espaço de busca.

3.2 BUSCA EM LARGURA

Fernandes (2008) define a busca em largura como uma busca que utiliza o critério *First In First Out* (FIFO), onde todos os nós de certo nível da árvore são examinados antes do nível seguinte, sempre da esquerda para a direita. Coppin (2010) afirma que a busca em largura é mais adequada a problemas onde a árvore pode ter caminhos muito profundos, pois, nestes casos, o tempo para encontrar a solução seria menor. A Figura 17 ilustra o caminho a ser percorrido em uma árvore de quatro níveis sendo varrida com a busca em largura. Nesta árvore, o nó solução (representado pela letra I) está colorido em cinza. O nó A representa a raiz de nossa árvore e o nível 0 da mesma, por esse motivo é o primeiro a ser buscado. Como o algoritmo não encontrou a solução no nível 0, passa-se para o nível 1 da árvore, representado pelos nós B e C. Os nós B e C são buscados nessa

ordem, concluindo a busca desse nível da árvore também sem sucesso. O próximo nível da árvore é o nível 2, representado pelos nós D, E e F, que são buscados nessa sequência. O algoritmo ainda não encontrou a solução e passa para o nível 3 da árvore, representado pelos nós G, H, I, J e K. O algoritmo busca sequencialmente os nós G e H até finalmente encontrar o nó objetivo em I. Ao lado de cada nó está ilustrado o número de passos e a sequência do algoritmo de busca até encontrar a solução do problema. No exemplo em questão foi necessário varrer nove nós para encontrar a resposta.

FIGURA 17 - BUSCA EM LARGURA

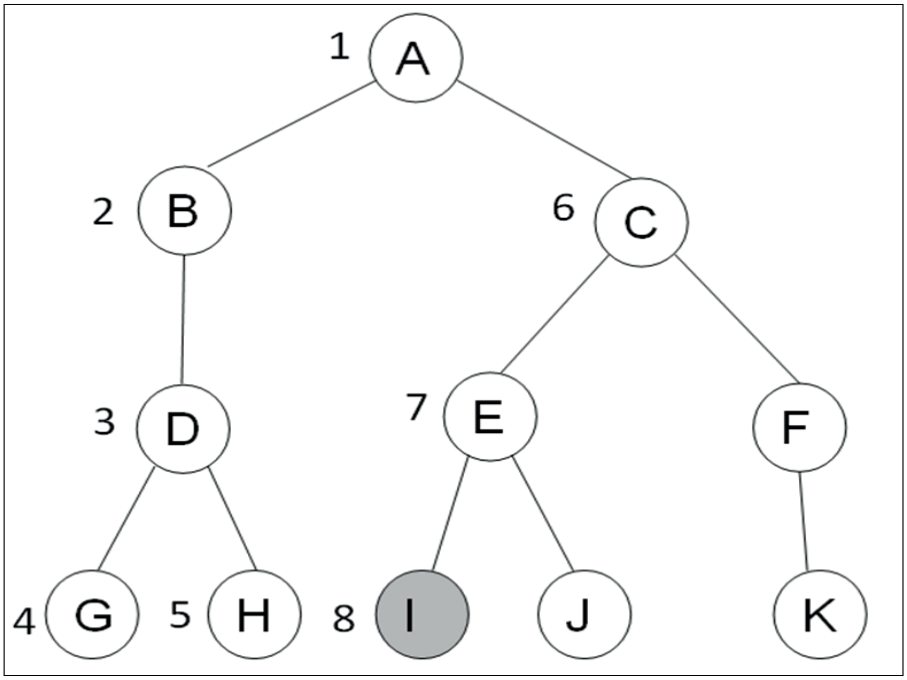


FONTE: O autor

3.3 BUSCA EM PROFUNDIDADE

A busca em profundidade é assim chamada, pois segue cada caminho até a maior profundidade possível antes de seguir para o próximo caminho. Luger (2002) afirma que na busca em profundidade, quando um estado é examinado, todos os seus filhos e os descendentes deles são examinados antes de qualquer um de seus irmãos, se aprofundando no espaço de estados sempre que possível. Na Figura 18 mostramos a árvore de estados do exemplo anterior sendo agora varrida por um algoritmo de busca em profundidade.

FIGURA 18 - BUSCA EM PROFUNDIDADE



FONTE: O autor

Neste exemplo, o primeiro nodo a seguir é o mesmo do exemplo anterior com a busca em largura, pois A representa a raiz da árvore. O próximo passo é verificar os filhos de A recursivamente até que não tenham mais descendentes. O primeiro filho de A é B, que passa a ser verificado. Como a solução ainda não foi encontrada em B, aplica-se a recursão e passa-se a buscar os seus descendentes. Como B somente tem um descendente, a busca passa para ele. A solução ainda não foi encontrada em D e passa-se para nós G e H. Um detalhe a ser salientado é que o algoritmo procura em D, passa para G, volta para D e daí sim procura em H. Essa característica do algoritmo exige que ele SEMPRE mantenha algum tipo de memória de trabalho que permita armazenar os pais de determinado nodo. Por exemplo: ao buscar B, o algoritmo deve manter o registro de A. Ao buscar o nó G, o algoritmo deve manter o registro de D, B e A.

Como a solução não foi encontrada neste ramo da árvore, o algoritmo volta e procura se B possui algum irmão. O nó C é irmão de B e passa então a ser varrido recursivamente pelo algoritmo utilizando o mesmo princípio usado no ramo anterior da árvore. A sequência de busca varre os nós C, E e finalmente I, encontrando o nó objetivo.

Um detalhe importante a ser destacado é que o segundo algoritmo foi mais eficiente para encontrar a resposta no exemplo em questão. Utilizando a busca em largura levamos nove passos para identificar o nó solução, enquanto a busca em profundidade levou oito. Quanto mais detalhes forem conhecidos sobre o problema, mais fácil é identificar qual é o algoritmo mais eficiente para encontrar a solução. No Quadro 8, Coppin (2010) faz uma comparação entre as buscas em profundidade e em largura.

QUADRO 8 - COMPARAÇÃO ENTRE ALGORITMOS DE BUSCA

Cenário	Profundidade	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)

As estruturas de árvore estão inseridas no dia a dia dos profissionais de TI, mesmo que seja de forma implícita. A maioria dos bancos de dados comerciais utiliza alguma estrutura de árvore em combinação com algoritmos de busca e ordenação para armazenar e buscar seus dados e índices de forma mais eficiente.

RESUMO DO TÓPICO 2

Neste tópico você aprendeu que:

- Muitos problemas em IA podem ser representados através de um espaço de busca de possibilidades em árvores.
- Essa estratégia é apoiada por uma visão de senso comum de resolução de problemas por seres humanos, onde geralmente são consideradas várias alternativas diferentes sobre o modo de resolver um problema.
- Uma heurística é uma estratégia para a busca seletiva de um espaço de problema.
- A busca cega é a abordagem mais simples para encontrar a solução de um problema em um espaço de busca, pois envolve simplesmente pesquisar todos os nós até encontrar uma solução para o problema.
- Na busca em largura todos os nós de certo nível da árvore são examinados em sequência para então se passar para o próximo nível.
- Na busca em profundidade segue-se cada caminho até a maior profundidade possível antes de seguir para o próximo caminho, examinando todos os descendentes de determinado nó antes de avaliar seu irmão.

AUTOATIVIDADE

1 Considerando um problema cujo espaço de busca de soluções é composto por uma árvore que possui um número muito grande de níveis e baixo grau de ramificação, assinale a alternativa CORRETA:

a) Para que a busca seja eficiente, deve ser aplicado um algoritmo de busca cega.
b) Neste caso específico, uma busca em profundidade seria mais eficiente que uma busca em largura.
c) Neste caso específico, uma busca em largura seria mais eficiente que uma busca em profundidade.
d) Neste caso poder-se-ia optar por qualquer estratégia de busca, pois o desempenho na busca da resposta seria idêntico.

2 Com relação à utilização de técnicas de busca na resolução de problemas de inteligência artificial, assinale a alternativa CORRETA:

( ) Uma heurística é uma técnica de busca onde todos os nós da árvore de solução são buscados sem critério de seleção até que uma solução seja encontrada.

( ) Na busca cega, inicialmente são verificados todos os descendentes de determinado nó, para então proceder à busca nos outros ramos das árvores.

( ) Na busca em profundidade, inicialmente são buscados todos os nós de determinado nível para só então se passar ao nível seguinte.

( ) As estratégias de IA que usam busca em uma árvore de possibilidades para encontrar a solução de um problema são baseadas nos procedimentos normalmente adotados pela inteligência humana para a resolução de problemas.

3 Visto que a existência de inteligência pressupõe a existência de conhecimento, considera-se que a representação do conhecimento é uma das etapas mais importantes na utilização de sistemas que apliquem técnicas de Inteligência artificial. Em relação à representação do conhecimento nos sistemas de Inteligência artificial, avalie as afirmações a seguir:

I - As redes semânticas baseiam-se na existência de regras.
II - Os roteiros representam conhecimento através do relacionamento entre nodos.
III - Uma das dificuldades de se trabalhar com quadros é o alto grau de granularidade.
IV - A representação procedimental é adequada para se trabalhar com heurísticas.
V - Os arcos estabelecem o tipo de relacionamento entre os nodos de uma rede semântica.

Agora assinale a alternativa que somente contém afirmações verdadeiras:

I, II, III e V.
I e V.
II, III e IV.
IV e V.

4 Um dos aspectos mais importantes na definição da técnica de inteligência artificial a ser aplicada é a maneira como a informação está representada dentro do sistema computacional. Entre as representações mais comuns, destacam-se as redes semânticas, os roteiros, os quadros e as regras de produção. Com relação às representações de conhecimento, avalie as afirmações a seguir:

I - As regras de produção podem ser representadas por um grafo de estados.
II - As redes semânticas permitem a reutilização de informações através da herança.
III - Os quadros permitem a representação do estado interno das entidades participantes do sistema.
IV - As redes semânticas podem ser transformadas em quadros, entretanto, o processo inverso não é possível.
V - Os roteiros podem ser utilizados como uma representação estruturada de uma sequência de eventos.

Agora assinale a alternativa que somente possui afirmações CORRETAS:

a) I, II e IV.
b) I, II e III.
c) II e IV.
d) I, III e V.

TÓPICO 3

SISTEMAS BASEADOS EM CONHECIMENTO (SBC)

1 INTRODUÇÃO

Existe um ramo da IA que é conhecido particularmente por classificar os sistemas como "sistemas que pensam como humanos". Mas o que significa pensar como humano? Basicamente, significa tentar reproduzir em um sistema computacional o raciocínio de um ser humano ao resolver problemas. Nessa linha da IA se enquadram os sistemas que fazem uso extensivo do conhecimento em nível de um especialista humano, chamados de Sistemas Baseados em Conhecimento (SBC).

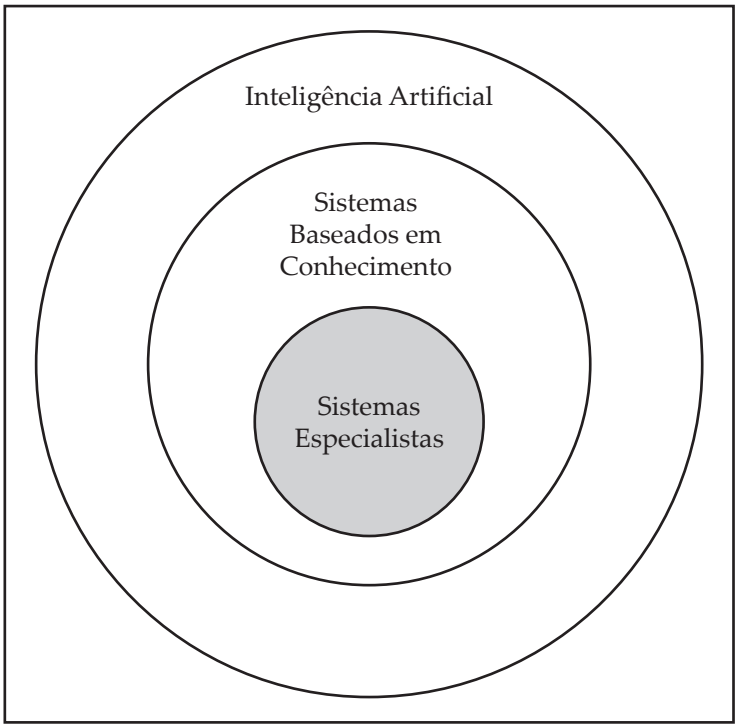
Esses sistemas aplicam regras e conhecimento extraídos de especialistas humanos para resolver problemas da maneira mais semelhante possível à desses especialistas. Tal estratégia mostra-se viável em uma larga gama de problemas, como diagnóstico médico, controle de tráfego e automação industrial.

Para facilitar seu entendimento, faremos a comparação entre o raciocínio de um especialista humano e um Sistema Especialista nos mais variados tipos de problemas. Nosso objetivo é preparar você para a engenharia do conhecimento e para a própria construção de sistemas especialistas, através da utilização das ferramentas sugeridas no texto.

2 SISTEMAS ESPECIALISTAS

Os Sistemas Especialistas (SE) se enquadram em uma categoria específica da IA chamada de SBC, caracterizada especialmente pela utilização de raciocínio sobre suas possíveis ações no mundo (Figura 19). Um SBC pode ser definido como um programa que se comporta como um ser humano em um domínio específico do conhecimento.

FIGURA 19 - ENQUADRAMENTO DOS SE DENTRO DOS SBC E IA



FONTE: O autor

O princípio básico por trás dos sistemas especialistas é sua habilidade em reproduzir o conhecimento de um especialista humano em determinado domínio do conhecimento. Luger (2002) afirma que os sistemas especialistas, assim como as pessoas experientes, tendem a ser especialistas, focando sobre um conjunto reduzido de problemas. Fernandes (2008) esclarece que um sistema especialista deve ser construído através de seu conhecimento e experiência adquiridos ao longo dos anos. Já Harmon e King (1988) definem um sistema especialista como um programa inteligente de computador que usa conhecimento e procedimentos inferenciais para resolver problemas complexos que geralmente requerem um especialista humano.

Ao contrário dos sistemas tradicionais, os sistemas especialistas apresentam algumas facilidades que aumentam sua flexibilidade e eficiência, como:

- possibilidade para construção de regras;
- tomada lógica de decisões sob imprecisão ou na ausência de informações.

Em um programa tradicional, o método de busca é baseado unicamente no conhecimento codificado (geralmente de forma estática) no sistema. Em caso de novos conhecimentos é preciso modificar o código-fonte. Os SE podem recuperar novos fatos e regras e usá-los sem modificar a estratégia de busca (FERNANDES, 2008).

Luger (2010) salienta que, por sua natureza heurística, os SE em geral:

1. suportam inspeção de seus processos de raciocínio, apresentando passos intermediários e respondendo a questões sobre o processo de solução. O SE deve ser aberto para inspeção, fornecendo informações sobre o estado de sua solução e explanação sobre seu processo decisório. Esse tipo de explanação é importante para que um perito humano, como um médico ou um engenheiro, aceite as recomendações feitas pelo computador;

2. permitem fácil modificação pela adição ou exclusão de conhecimentos da base. Por sua natureza exploratória e pelo dinamismo existente na maioria dos domínios de conhecimento, os SE devem ser facilmente implementados, testados e modificados;

3. raciocinam heuristicamente, usando conhecimento para obter soluções úteis. É importante salientar que nem todo o conhecimento útil para os SE vem de livros e manuais. Em geral, o conhecimento empírico, somente disponível através da experiência dos especialistas, oferece atalhos úteis para a solução de problemas.

O Quadro 9, adaptado de Harmon e King (1988), ilustra algumas das principais diferenças entre os SE e os sistemas convencionais.

QUADRO 9 - DIFERENÇAS ENTRE OS SE E OS SISTEMAS TRADICIONAIS

Sistemas especialistas	**Sistemas convencionais**
Heurísticas.	Algoritmos.
Conhecimento estruturado simbolicamente em uma memória de trabalho.	Bancos de dados numericamente endereçados.
Processamento simbólico.	Orientado por processamento numérico.
Processamento interativo.	Processamento sequencial.
Explicação do fluxo da execução fácil.	Explicação do fluxo de execução complexa, por meio da análise do código-fonte.

FONTE: Adaptado de Harmon e King (1988)

O processo de construção de um SE é chamado de "engenharia de conhecimento" e envolve normalmente uma forma especial de interação entre o construtor do sistema, chamado de "engenheiro do conhecimento", e um ou mais especialistas. O objetivo desse processo é extrair do especialista humano os procedimentos, estratégias e regras para a solução de problemas (FERNANDES, 2008).

O processo de aquisição do conhecimento envolve consulta a livros, manuais técnicos, manuais de procedimentos, normas e entrevista com especialistas

humanos, constituindo-se na etapa mais crítica na construção dos SE. Os principais obstáculos encontrados na extração do conhecimento dos especialistas humanos são a dificuldade em se extrair todo o conhecimento e esgotar o domínio, uma vez que parte do conhecimento é tácito e não explícito, e a dificuldade em conseguir a total cooperação do especialista, uma vez que usualmente existe o temor de ser "substituído" pelo sistema.

Um SE é composto por quatro elementos principais:

1. Base de conhecimento (regras) – consiste no conhecimento que foi extraído do especialista humano no domínio, sendo armazenada geralmente na forma de regras SE, ENTÃO. Por exemplo, um sistema de diagnóstico médico pode possuir a seguinte regra: SE febre > 38 graus ENTÃO diagnóstico = FEBRE_ ALTA;

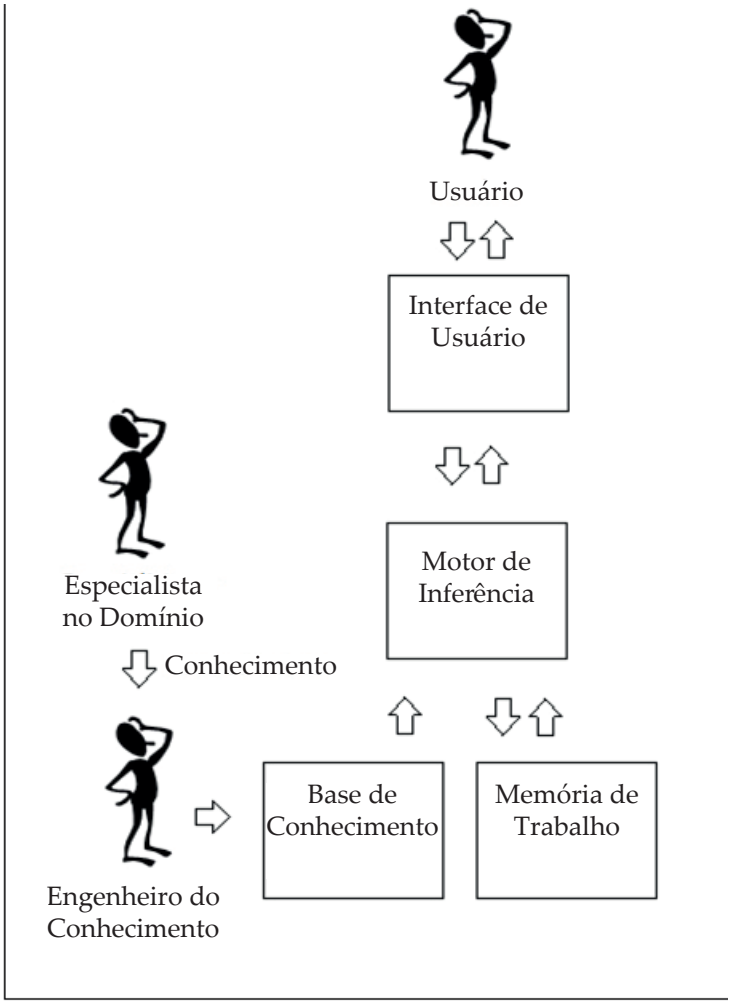
2. Memória de trabalho (base de fatos) - consiste no caso que está sendo apresentado para o SE trazer a resposta. Essa memória é de caráter transitório, pois novos fatos estão sendo acrescentados ou excluídos continuamente. Com base no exemplo anterior, colaríamos um valor numérico para a variável *febre* e aguardaríamos o diagnóstico do sistema;

3. Mecanismo de inferência – é quem faz a junção dos fatos com as regras, chegando a conclusões e gerando novos fatos. No exemplo anterior, digamos que *febre* contenha o valor 40 e que seja então submetido ao sistema. O mecanismo de inferência aplicaria o valor 40 (fato) à regra definida, chegando ao diagnóstico FEBRE_ALTA;

4. Interface do usuário – cuida da comunicação de forma mais amigável entre o usuário e o sistema, fornecendo também noção sobre o processo de resolução empregado.

A Figura 20 ilustra o processo de engenharia do conhecimento e as partes componentes de um SE:

FIGURA 20 - CONSTRUÇÃO E ARQUITETURA DE UM SE



FONTE: O autor

Para diferenciar o raciocínio de um especialista humano, em comparação com o raciocínio de um SE, observe o exemplo a seguir:

West é criminoso ou não?

"A lei americana diz que é proibido vender armas a uma nação hostil. Cuba possui alguns mísseis, e todos eles foram vendidos pelo Capitão West, que é americano". Para resolver esse problema, um especialista humano utilizaria três componentes:

1. **linguagem**: você entende o que está escrito em português;
2. **conhecimento**: você sabe um pouco de geopolítica e armas;
3. **inferência**: você é capaz de raciocinar usando esse conhecimento descrito em português.

Os Quadros 10, 11 e 12 ilustram o procedimento para resolver o problema através de linguagem, conhecimento e inferência:

QUADRO 10 - FATOS GERAIS

A) Todo americano que vende uma arma a uma nação hostil é criminoso.
B) Todo país em guerra com uma nação X é hostil a X.
C) Todo país inimigo político de uma nação X é hostil a X.
D) Todo míssil é uma arma.
E) Toda bomba é uma arma.
F) Cuba é uma nação.
G) EUA é uma nação.
H) Cuba é inimigo político dos EUA.

FONTE: Disponível em: <http://www.acso.uneb.br/marcosimoes/Disciplinas/UNEB/TEI3/Arquivos/Agentes-Baseados-em-Conhecimento_6-por-pagina.pdf>. Acesso em: 6 abr. 2014.

QUADRO 11 - FATOS ATUAIS (MEMÓRIA DE TRABALHO) – CONHECIMENTO

I) West é americano.
J) Existem mísseis em Cuba.
K) Os mísseis de Cuba foram vendidos por West.

FONTE: Disponível em: <http://www.acso.uneb.br/marcosimoes/Disciplinas/UNEB/TEI3/Arquivos/Agentes-Baseados-em-Conhecimento_6-por-pagina.pdf>. Acesso em: 6 abr. 2014.

QUADRO 12 - CONCLUSÕES PARCIAIS, NOVOS FATOS E CONCLUSÃO FINAL – INFERÊNCIA

L) Cuba possui um míssil M1	- de J
M) M1 é um míssil	- de J
N) M1 é uma arma	- de D e M
O) Cuba é hostil aos USA	- de F, G, H e C
P) M1 foi vendido a Cuba por West	- de K, L e M
R) West é criminoso	- de A, K, N, O e P

FONTE: Disponível em: <http://www.acso.uneb.br/marcosimoes/Disciplinas/UNEB/TEI3/Arquivos/Agentes-Baseados-em-Conhecimento_6-por-pagina.pdf>. Acesso em: 6 abr. 2014.

O Quadro 10 pode ser comparado à base de conhecimento, pois lá estão as regras e alguns fatos que devem ser considerados na resolução do problema. O Quadro 11 traz a situação atual e é análogo à memória de trabalho e, finalmente, o Quadro 12 seria o mecanismo de inferência, gerando novos fatos e a conclusão final, informando qual é o caminho tomado para chegar a cada conclusão.

O Quadro 13 demonstra o funcionamento de um SE para diagnóstico e tratamento de febre através de sua base de conhecimento e suas regras:

QUADRO 13 - BASE DE CONHECIMENTO DE UM SE

Regra FEBRE_LEVE Se a temperatura > 36,5 graus e a temperatura < 38 graus e a idade ADULTA então Tipo de febre = FEBRE_LEVE Procedimento: TOMAR_ASPIRINA
Regra FEBRE_ALTA Se temperatura >38 graus e idade ADULTA então Tipo de febre = FEBRE_ALTA Procedimento: CONSULTA_MEDICA
Regra FEBRE_LEVE_CRIANÇA Se temperatura > 36,5 graus e temperatura < 38 graus e idade CRIANÇA então Tipo de febre = FEBRE_LEVE Procedimento: OBSERVAÇÃO

FONTE: O autor

Neste exemplo alimentaríamos a memória de trabalho do SE com os fatos, informando a temperatura da pessoa e se ela está na idade adulta ou é criança. O mecanismo de inferência faria a junção dos fatos com as regras e nos daria o diagnóstico e o procedimento. Por exemplo, ao dizer que uma criança está com febre de 37º, o SE nos responderia, fundamentado por sua base de conhecimento, que se trata de uma febre leve e que devemos manter a criança em observação.

Podemos concluir, então, que os mecanismos de raciocínio adotados por especialistas humanos e pelos SE funcionam de formas semelhantes, o que justifica a eficiência desses sistemas nos mais variados domínios de conhecimento.

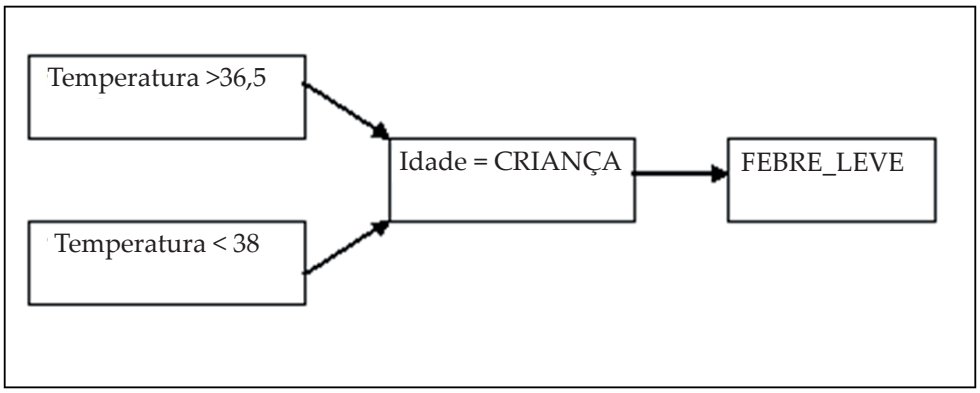
Os SE trabalham com dois tipos de raciocínio:

- encadeamento regressivo (*backward chaining*) – nesse caso iniciamos com uma hipótese e raciocinamos para trás na rede de inferência, buscando a confirmação. Normalmente é utilizado quando há a necessidade de verificar se uma hipótese é verdadeira, justificando o raciocínio utilizado para chegar nela. Em medicina, por exemplo, algumas observações iniciais do paciente levam o médico a gerar alguma hipótese inicial que deve ser confirmada ou rejeitada por evidências adicionais. Essas evidências podem ser obtidas usando encadeamento regressivo;

- encadeamento progressivo (*forward chaining*) – nesse caso iniciamos com evidências e raciocinamos para a frente na rede de inferência, buscando uma conclusão. Normalmente é utilizado quando existem muitas hipóteses disponíveis e não há razão para começar por qualquer uma delas. Em geral, o raciocínio progressivo é mais natural em tarefas de monitoramento nas quais os dados são adquiridos de forma contínua.

Vamos exemplificar o encadeamento progressivo e regressivo nas Figuras 21 e 22, respectivamente, utilizando a regra FEBRE_LEVE_CRIANÇA do Quadro 13.

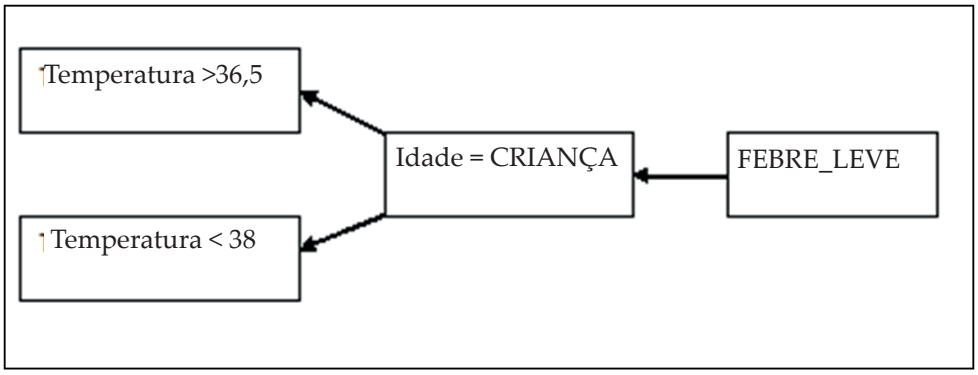
FIGURA 21 - ENCADEAMENTO PROGRESSIVO



FONTE: O autor

Na Figura 21, através do encadeamento progressivo, partiríamos de evidências como a temperatura e a idade da pessoa, para então chegarmos à conclusão de que ela está com FEBRE_LEVE.

FIGURA 22 - ENCADEAMENTO REGRESSIVO



FONTE: O autor

Na Figura 22, através do encadeamento regressivo, ao ouvir que a pessoa está com febre leve, voltaríamos buscando evidências de que a hipótese da febre é verdadeira, como a idade da pessoa e sua temperatura.

2.1 FATOR DE CONFIANÇA

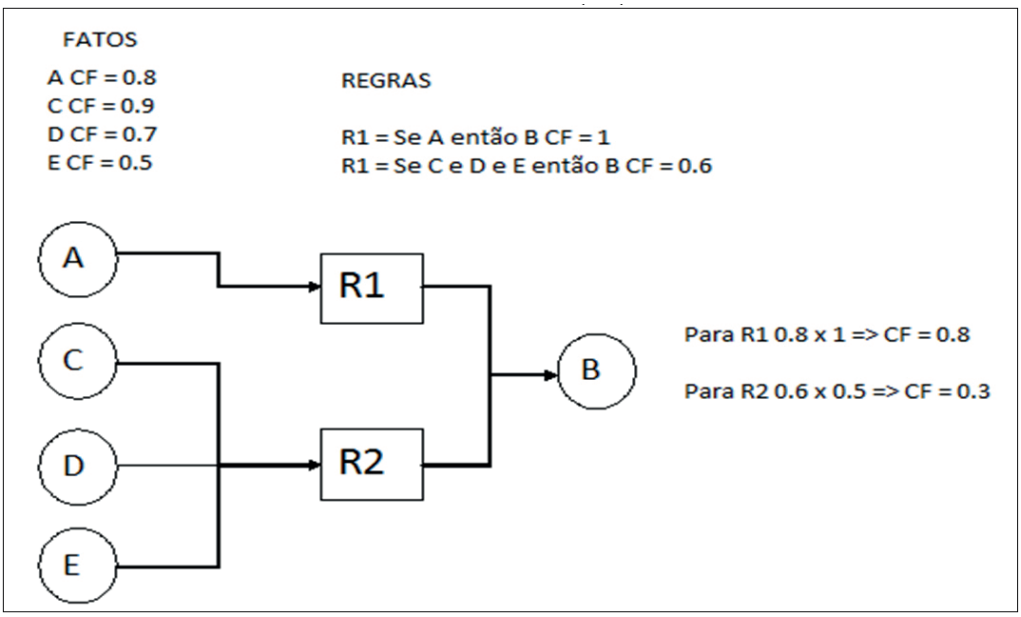
Em muitos problemas do mundo real, chega-se a uma solução sob a presença de incerteza. A incerteza num domínio de aplicação pode estar presente nos dados de entrada, na solução do problema ou em ambos. Por exemplo, no caso de diagnóstico médico, os sinais e sintomas que o médico coleta e trabalha apresentam vários problemas de incerteza, como: a inexatidão dos relatos do paciente e a percepção da intensidade de cada sintoma, entre outros. Além disso, o próprio raciocínio clínico não pode ser realizado com certeza, isto é, podem existir dois pacientes com dois conjuntos similares de sinais e sintomas e seus diagnósticos serem diferentes (NASSAR, 2012).

Para esse tipo de problemas, uma solução possível é a atribuição de valores que representam os fatores de confiança para as evidências. Esses valores variam entre 0,0 e 1, sendo que, quanto mais próximo do 1, mais forte é a certeza da regra ou do fato. Por exemplo, um fato que tenha fator de confiança igual a 0,4 significa que o sistema tem somente 40% de certeza da ocorrência daquele fato.

Outra solução típica para situações do mundo real onde existe necessidade de se tratar a incerteza são as **redes bayesianas e a lógica difusa**, objetos da segunda unidade.

Podemos observar na Figura 23 os Fatores de Confiança (CF) atribuídos aos fatos e às regras. Por exemplo, o fato A tem fator de confiança 0,8 enquanto o fato E tem fator de confiança 0,5. O mesmo princípio se aplica às regras, com fatores de confiança iguais a 1 na regra 1 (R1) e 0,6 na regra 2 (R2). Para definirmos o fator de confiança para a conclusão obtida, fazemos a multiplicação entre os fatores de confiança dos fatos e das regras. No caso de R1, basta multiplicar 0,8 por 1, obtendo um fator de confiança 0,8 para a conclusão B obtida. No caso de R2, como a conclusão depende da combinação de diversos fatos, escolhemos o fato que possui o menor fator de confiança e multiplicamos pelo fator de confiança da regra. Nesse caso, pegamos o valor 0,5, obtido de E, e multiplicamos por 0,6, obtido de R2, resultando em 0,3 como fator de confiança na conclusão B.

FIGURA 23 - EXEMPLO DE FATOR DE CERTEZA (CF)



FONTE: O autor

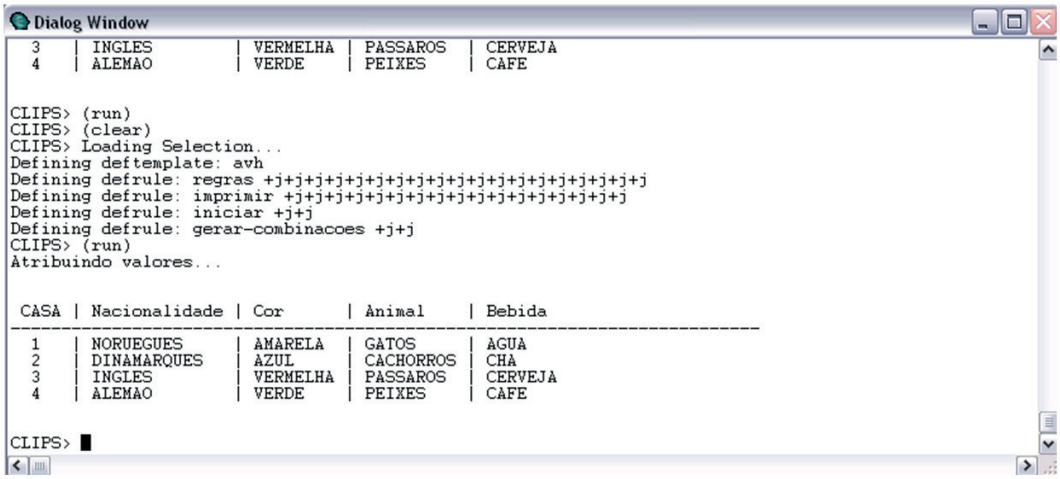
Caso haja mais de um fato ou de uma regra envolvidos na obtenção de uma conclusão, sempre utilizamos os menores fatores de confiança existentes. A utilização dessa estratégia assegura que a conclusão apareça da forma mais realista possível e, consequentemente, que não se tomem decisões erradas com base nos SE.

Existem diversas ferramentas que permitem a implementação de SEs no mercado. Para você experimentar na prática a criação e utilização destas ferramentas, recomendamos duas:

- CLIPS – disponível livremente em: <http://clipsrules.sourceforge.net/>;
- EXPERT SINTA – disponível livremente em: <http://www.lia.ufc.br/~bezerra/exsinta/>.

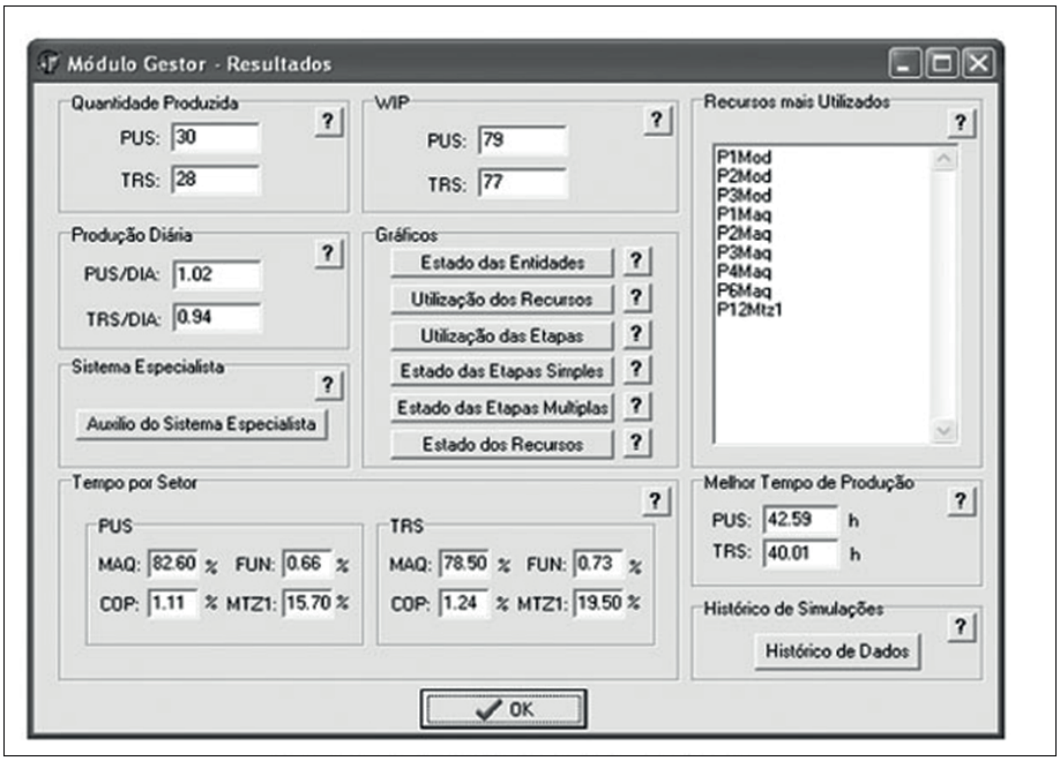
As Figuras 24 e 25 mostram implementações de sistemas especialistas nas ferramentas CLIPS e EXPERT SINTA, respectivamente.

FIGURA 24 - CLIPS DEMONSTRANDO A SOLUÇÃO DE UM PROBLEMA



FONTE: O autor

FIGURA 25 - EXPERT SINTA



FONTE: Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592006000100002>. Acesso em: 6 abr. 2014.

Os SE são especialmente úteis nos casos em que o conhecimento possuído pelo especialista humano é raro, difícil de transferir ou documentar, não previsível, perecível ou caro. Nestas situações, implementar um SE pode garantir a continuidade na resolução de problemas, pois o conhecimento armazenado torna-se então permanente, fácil de transferir ou documentar e de baixo custo. Por outro lado, ao contrário dos especialistas humanos, o SE possui um foco estreito na resolução de problemas e somente atua baseado no conhecimento técnico. A experiência de um especialista humano, em combinação com o bom senso, pode levar a uma conclusão diferente do que sua "base de conhecimento" afirma e chegar com isso a uma conclusão ótima. De forma geral, os problemas que são resolvidos mais eficientemente por um SE possuem as características a seguir:

- existem poucos indivíduos com o conhecimento necessário para a resolução do problema e estes passam muito tempo auxiliando outros indivíduos;
- problemas compostos por tarefas pequenas, realizadas por diversas pessoas, pois nenhum indivíduo sozinho sabe o suficiente;
- grande discrepância entre o pior e o melhor desempenho na resolução;
- quanto mais rápido for encontrada uma solução, melhor.

LEITURA COMPLEMENTAR

Preparado para a Era da Computação Cognitiva? – Adaptação livre do autor para o texto retirado de <http://www.mti.mt.gov.br/-/preparado-para-a-era-da-computacao-cognitiva->. Acesso em: 15 fev. 2017.

A "appficação" da sociedade já é um fato e só vai aumentar nos próximos anos. Em Cloud Computing não estamos mais discutindo se vamos ou não, mas tentando explicar porque ainda não fomos, muitas vezes agarrados a paradigmas que perderam sua essência. Big Data já é o cerne de negócios como Google, Facebook, Linkedin e em muitas empresas de varejo, transporte aéreo, bancos etc. Já existem, ao redor do mundo, muitos casos de sucesso na geração de *insights* e Inteligência a partir da exploração de dados, e qualquer sistema de *e-commerce* que se proponha a ser medianamente inteligente deve embutir algoritmos de recomendação em sua operação. Internet das Coisas já está na nossa vida. Basta ver como os aviões, navios e automóveis estão cada vez mais automatizados.

Os automóveis estão caminhando rápido para serem veículos autônomos ou quase autônomos e dirigir, na próxima década, provavelmente será uma tarefa opcional. Veículos autônomos, interagindo uns com os outros, vão eliminar sinais de tráfego, por exemplo. Estes se tornarão supérfluos. Talvez até mesmo os seguros de veículos como feitos hoje poderão ser considerados dispensáveis. A própria indústria automotiva passará por mudanças, de fabricante de automóveis a empresas de serviços de mobilidade pessoal, onde fabricar e vender o veículo deixará de ser o objetivo final do negócio, mas passará a ser o meio para a venda de serviços. Talvez hoje a principal barreira não seja a tecnologia embarcada, mas a

regulação de como tornar esta automação viável na nossa sociedade. A tecnologia embarcada neles, ainda cara, tende a baratear dramaticamente nos próximos anos. Vejam que a discussão já não está mais na questão de se é possível ou não um carro autônomo, mas de regulação. Este texto é bem interessante:

<http://spectrum.ieee.org/transportation/advanced-cars/plate-and-switch-googles-selfdriving-car-is-a-transformer-too>.

GOOGLE E SEU CARRO AUTÔNOMO

FONTE: Disponível em: <http://blog.caranddriver.com/driving-not-optional-google-shows-off-prototype-autonomous-car-with-no-steering-wheel-or-pedals/>.

E quanto à computação cognitiva? Lembro que em 2004 (dez anos...) li um livro que me chamou muito a atenção. O título é "The New Division of Labor: How Computers Are Creating The Next Job Market", dos economistas Frank Levy e Richard Murnane. Diante da acelerada evolução tecnológica, os autores argumentaram que os computadores assumiriam o lugar das atividades humanas em muitas tarefas, mas não poderiam operar em outras. As tarefas que envolvessem percepção sensorial, reconhecimento de padrões e conhecimento conceitual continuariam exclusivas dos seres humanos. Eles fizeram a distinção entre conhecimento tácito e explícito. O explícito ou declarativo poderia ser expresso via instruções orais ou escritas e, portanto, programáveis. Os computadores poderiam assumir todas as tarefas explícitas. Já o conhecimento tácito refere-se a tudo aquilo que fazemos, mas não conseguimos claramente definir como fazemos.

Aprendemos e internalizamos o conhecimento, como dirigir um veículo, por exemplo. Mesmo que consigamos explicar como fazer uma ultrapassagem, dificilmente alguém repetiria exatamente nossas ações. Não existe uma receita simples e declarativa para estas tarefas. Assim, o conhecimento tácito continuaria inerentemente humano. O carro do Google rompeu estas barreiras. E novos sistemas de interface e diálogo em linguagem natural, como o Siri, Cortana, Google Now e o Watson quebram mais uma vez a barreira entre o tácito e o explícito. As implicações serão significativas.

Os sistemas cognitivos são o resultado da convergência de avanços significativos em vários ramos da ciência da computação, como *hardware* (processadores e *storage* mais poderosos e baratos), processamento de linguagem natural, "*machine learning*, como redes neurais", reconhecimento de padrões etc. Recomendo a leitura de um livro muito interessante que aborda esta questão: "The Second Machine Age: work, progress and prosperity in a time of brilliant technologies", de Erik Brynjolfsson e Andrew McAfee.

Os sistemas cognitivos têm o potencial de criar rupturas nas empresas e na sociedade, mudando inclusive a natureza do trabalho. Não apenas as tarefas explícitas podem ser automatizadas, mas tarefas tácitas (um veículo autônomo, por exemplo, pode dispensar motorista). O vetor resultante cria um impacto potencial significativo na nossa sociedade. TI, por exemplo, deixará de ser apenas prestador de serviços de automação baseados em sistemas determinísticos, mas com aplicações "inteligentes" contribuirão diretamente para operações mais sofisticadas do negócio.

Atividades realizadas hoje por indivíduos, como o atendimento em *call center* e o suporte administrativo, podem ser inteiramente substituídos por estes sistemas. Não seria inimaginável pensar que diversas tarefas ligadas a setores como educação, direito e saúde também poderiam ser efetuadas por sistemas "inteligentes". Alguns exemplos de como isto está começando a se tornar realidade podem ser vistos no texto do Memorial Sloan-Kettering Cancer Center (http://www.mskcc.org/blog/msk-trains-ibm-watson-help-doctors-make-better-treatment-choices), que aborda como a computação pode ajudar na medicina – no caso, na oncologia – e nas empresas de serviços de saúde, na aprovação de exames especializados, como podemos ver em <http://www.fiercehealthpayer.com/story/wellpoint-watson-ibm-machine-learning-evidence-based-care/2014-06-11>. Ambos são casos de uso do Watson, da IBM.

IBM WATSON E A COMPETIÇÃO JEOPARDY

FONTE: Disponível em: <http://www.jenunderwood.com/2016/10/26/new-ibm-watson-data-platform-data-science-experience/>.

O exemplo da computação "inteligente" ajuda no diagnóstico médico e muda nossa maneira de ver as coisas. O sistema auxilia no diagnóstico acessando mais de 600 mil relatórios de evidência médica, dois milhões de páginas de texto de 42 publicações especializadas em câncer e 1,5 milhão de registros e exames de pacientes. O sistema compara cada sintoma de cada indivíduo, sinais vitais, histórico familiar, medicamentos já aplicados, genética e rotina diária como alimentação e exercícios para diagnosticar e propor um plano de tratamento específico. Muito difícil a qualquer médico conseguir analisar tal volume de informações para cada paciente. Na área do direito, um sistema cognitivo pode analisar milhões de casos precedentes para descobrir e recomendar uma linha de ação. Talvez não seja mais necessária uma legião de estagiários para fazer tal tarefa.

Claro, chegar lá não é simples, como as primeiras experiências realizadas por Watson estão demonstrando. Vejam em: <http://online.wsj.com/news/articles/SB10001424052702304887104579306881917668654>. Na verdade, um sistema complexo como Watson não é uma implementação *plug-and-play* como os prospectos comerciais tendem a mostrar. Demanda a preparação de um ecossistema que envolve atividades de pesquisa (coleta de novas informações), curadoria de conhecimento (filtrar o que é relevante para o domínio do conhecimento) e, naturalmente, análise e interação com o sistema. É um processo de evolução incremental, com constante aperfeiçoamento dos próprios algoritmos aplicados.

Mas há dez anos tal tarefa era considerada fora do escopo da computação e agora estamos discutindo qual seu grau de eficiência. Um avanço e tanto!

O desafio é que as mudanças acontecem em ritmos cada vez mais acelerados. Há uns 20 anos apenas 3% da população mundial tinha celulares e uma ínfima parcela de 1% acessava a internet. Há dez anos não existiam iPhone, iPads, Facebook, YouTube e Twitter. O fato do Watson não acertar tudo ou o Siri tropeçar nas respostas não significa que daqui a poucos anos sua margem de acerto não será imensamente maior.

Portanto, devemos estar preparados para as tecnologias futuras. Precisamos entender como elas mudarão a sociedade, a economia e a forma de atuação das nossas empresas. A destruição criativa, como disse Joseph Schumpeter, continua ativa. Desloca empresas e setores consolidados e cria aberturas para novos modelos de negócio. As tecnologias como a computação cognitiva já não estão mais no campo da ficção científica, mas na questão de quando e em que intensidade vão transformar nossas organizações.

Nesta e nas próximas décadas os executivos de negócio devem compreender e usar a tecnologia como força de ruptura nos seus negócios. O mundo evolui na velocidade da internet e tentar se segurar com a ilusão de que "meu negócio é estável e não vai mudar, pois não mudou nos últimos anos", provavelmente não o protegerá da inevitável transformação. A combinação do efeito de múltiplas tecnologias que evoluem rapidamente afetará todas as empresas e criará mudanças significativas na natureza do trabalho atual. Novas capacitações serão requeridas e novos modelos de negócio surgirão. Estamos nos preparando?

RESUMO DO TÓPICO 3

Neste tópico você aprendeu que:

- Os Sistemas Especialistas (SE) aplicam regras e conhecimento extraídos de especialistas humanos para resolver problemas da maneira mais semelhante possível destes especialistas.

- Os SE se enquadram em um ramo especial da IA, chamado de Sistemas Baseados em Conhecimento.

- O processo de construção de um SE envolve normalmente uma forma especial de interação entre o construtor do sistema, chamado de "engenheiro do conhecimento", e um ou mais especialistas.

- Um SE é composto por quatro elementos principais: base de conhecimento, memória de trabalho, motor de inferência e interface gráfica.

- Os SE trabalham basicamente com dois tipos de raciocínio: encadeamento progressivo ou encadeamento regressivo.

- No encadeamento regressivo iniciamos com uma hipótese e buscamos a confirmação da mesma verificando o raciocínio utilizado para chegar nela.

- No encadeamento progressivo iniciamos com evidências e raciocinamos para a frente na rede de inferência, buscando chegar a uma conclusão.

- Existem basicamente três maneiras de representar e tratar a incerteza em sistemas especialistas: fator de confiança (menos utilizado), redes bayesianas e lógica difusa.

AUTOATIVIDADE

1 Os Sistemas Especialistas (SE) essencialmente procuram reproduzir as respostas dadas por especialistas humanos na resolução de problemas. Para atingir isso, o conhecimento dos especialistas é extraído através de uma etapa chamada engenharia do conhecimento e armazenado em um repositório com características especiais chamado de base de conhecimento. Com relação aos Sistemas Especialistas, avalie as afirmativas a seguir:

I – A etapa mais crítica na criação dos SE é a implementação destes em alguma ferramenta computacional.
II – O motor de inferência é a parte responsável por unir as regras da base de conhecimento com os fatos da memória de trabalho.
III – A memória de trabalho é uma parte estática dentro dos SE.
IV – Uma das maneiras de implementar o tratamento da incerteza nos SE é através do fator de confiança.

Agora assinale a alternativa que somente contém afirmações corretas:

( ) I, II, III e IV.
( ) I, III e IV.
( ) I, II e III.
( ) II e IV.

2 Avalie as afirmações sobre os SE:

I. Os Sistemas Especialistas (SE) aplicam regras e conhecimento extraídos de especialistas humanos para resolver problemas da maneira mais semelhante possível destes especialistas, fazendo parte de uma subárea da IA chamada de SBC.
II. Um SE é composto por três elementos principais: base de conhecimento, motor de inferência e interface gráfica.
III. O raciocínio progressivo serve para determinar os sintomas de uma doença, a partir de um diagnóstico previamente obtido.
IV. O engenheiro do conhecimento é responsável por traduzir o conhecimento do especialista para o formato de regras e fatos.

Agora assinale a alternativa que somente possui afirmações VERDADEIRAS:

a) I, II e III.
b) I e IV.
c) I, II, IV.
d) I e II.

3 Os Sistemas Especialistas (SE) caracterizam-se, entre outras coisas, pela natureza heurística com que processam o conhecimento e suas regras, o que garante uma maior confiabilidade à conclusão obtida. Com relação à natureza heurística dos SE, avalie as afirmações a seguir:

I - Em um SE é possível avaliar os passos intermediários para se chegar a uma conclusão.
II - O raciocínio heurístico é especialmente útil quando o SE auxilia um especialista humano nos campos de engenharia e medicina.
III - A manutenção de um SE tem uma complexidade comparável à de um sistema tradicional, pois ocorre através da adição ou remoção de conhecimento da base.
IV - Todo o conhecimento de um SE é extraído diretamente de um especialista humano.

Agora assinale a alternativa que somente possui afirmações CORRETAS:

( ) I, II e IV.
( ) I, III e IV.
( ) I e II.
( ) III e IV.

4 Os sistemas especialistas (SE) caracterizam-se por possuírem uma arquitetura modular, que facilita a manutenção e alteração do conhecimento neles armazenado. Com relação à arquitetura modular dos SE, avalie as afirmações a seguir:

I - A base de conhecimento é onde são armazenadas as situações a serem processadas para cada problema, sendo considerada altamente volátil.
II - O mecanismo de inferência é a parte que combina os fatos e as regras, chegando a conclusões parciais e finais e gerando novos fatos.
III - A memória de trabalho é onde são armazenadas as situações a serem processadas para cada problema, sendo considerada altamente volátil.
IV - O conhecimento extraído do especialista é armazenado na base de fatos.

Agora assinale a alternativa que somente possui afirmações CORRETAS:

a) I, II e IV.
b) I, III e IV.
c) II e III.
d) III e IV.

5 O fator de confiança é uma maneira através da qual os sistemas especialistas (SE) fazem tratamento de incerteza, ou seja, quando não se tem certeza absoluta sobre determinada informação ou regra, aplica-se um grau de confiança na mesma, que influirá na conclusão final. Acerca da utilização do fator de confiança nos SE, assinale a alternativa CORRETA:

( ) O cálculo do fator de confiança para determinada conclusão em um SE depende unicamente do valor atribuído para os fatos.

( ) O fator de confiança em uma conclusão é obtido através da soma do fator de confiança dos fatos com o fator de confiança das regras.

( ) O cálculo do fator de confiança para determinada conclusão em um SE depende unicamente do valor atribuído para as regras.

( ) Caso exista mais de um fato envolvido em uma conclusão e estes fatos possuam valores distintos para o fator de confiança, pega-se sempre o menor valor.

UNIDADE 2

LÓGICA DIFUSA, REDES BAYESIANAS E COMPUTAÇÃO EVOLUTIVA

OBJETIVOS DE APRENDIZAGEM

Ao final desta unidade você será capaz de:

- conhecer os principais conceitos que envolvem o campo da inteligência artificial conhecido como computação evolutiva;
- especificar o funcionamento de algoritmos genéticos e programação genética;
- demonstrar de forma prática o funcionamento destas técnicas na resolução de problemas;
- comparar as técnicas de computação evolutiva com as vistas nas demais unidades deste livro de estudos.

PLANO DE ESTUDOS

Esta unidade de ensino está dividida em dois tópicos. No final de cada um deles você encontrará atividades que contribuirão para a apropriação dos conteúdos.

TÓPICO 1 – LÓGICA DIFUSA E REDES BAYESIANAS

TÓPICO 2 – COMPUTAÇÃO EVOLUTIVA

TÓPICO 1

LÓGICA DIFUSA E REDES BAYESIANAS

1 INTRODUÇÃO

Os sistemas computacionais, essencialmente por funcionarem através de bits e bytes, em geral, tendem a resolver problemas através de uma visão binária do mundo, onde todas as relações se restringem a "verdadeiro" ou "falso".

Qualquer observador atento perceberá que o mundo não funciona dessa maneira. As relações são em geral mais complexas e a certeza de que algo é 100% verdadeiro ou 100% falso não é fácil de alcançar. Nesse contexto é que entra o tratamento de incerteza.

Abordaremos, no Tópico 1, duas técnicas bastante utilizadas na Inteligência Artificial (IA) para esse tipo de situação: a lógica difusa e as redes bayesianas. Ilustraremos, nesta seção, a base teórica que sustenta cada uma dessas técnicas através de exemplos do mundo real e textos complementares. Para alavancar ainda mais seu aprendizado, demonstraremos a implementação de Sistemas Especialistas (SE) com lógica difusa e redes bayesianas em ferramentas computacionais, objetivando capacitar e encorajar você a fazer suas próprias implementações.

Contamos com você para vencermos esse desafio!

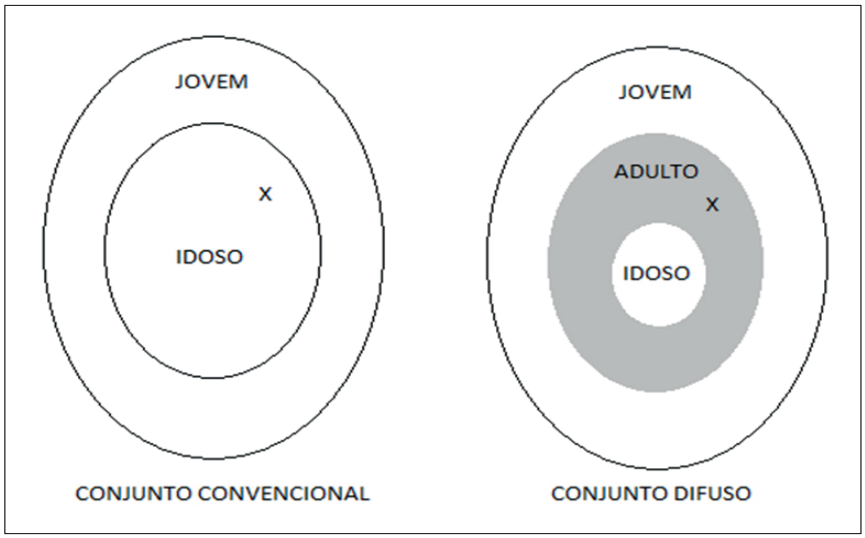
2 LÓGICA DIFUSA E REDES BAYESIANAS

Na lógica clássica, também conhecida como aristotélica ou bivalente, existem dois valores de verdade possíveis: verdadeiro ou falso. Em geral, nossas relações com o mundo real não funcionam dessa forma. Veja um exemplo: suponha que queremos modelar o conceito de "jovem", dizendo que se uma pessoa tiver menos de 25 anos, ela será considerada jovem. E as pessoas de 25 anos e um dia? Simplesmente não são mais consideradas jovens?

Esse tipo de conhecimento deve ser modelado, se quisermos uma representação mais apurada das relações que acontecem no mundo real. Para esse tipo de situação, utilizamos a lógica difusa, também conhecida como lógica nebulosa ou *fuzzy*, criada por Lotfi Zadeh, na Universidade da Califórnia, em 1956.

Conforme Bittencourt (2006), a lógica nebulosa é o modelo mais tradicional para o tratamento da informação vaga ou imprecisa. Já Norvig e Russel (2004) afirmam que a lógica difusa é um meio de especificar quanto um objeto satisfaz a uma descrição vaga através da utilização de variáveis linguísticas. Um detalhe interessante a ser observado é que a incerteza não é sobre o mundo exterior, pois estamos certos da idade da pessoa. A questão é que o termo "jovem" não trazia uma demarcação nítida entre os conjuntos "jovem" e não jovem. A Figura 26 ilustra um conjunto convencional em comparação com um conjunto difuso.

FIGURA 26 - CONJUNTO CONVENCIONAL E DIFUSO



FONTE: O autor

No conjunto convencional, uma pessoa X, colocada fora do conjunto jovem, seria considerada idosa, independentemente de sua localização. Enquanto que no conjunto difuso ela poderia ser considerada adulta, visto que há uma espécie de "suavização" entre jovem e idoso, representado pelo conjunto adulto.

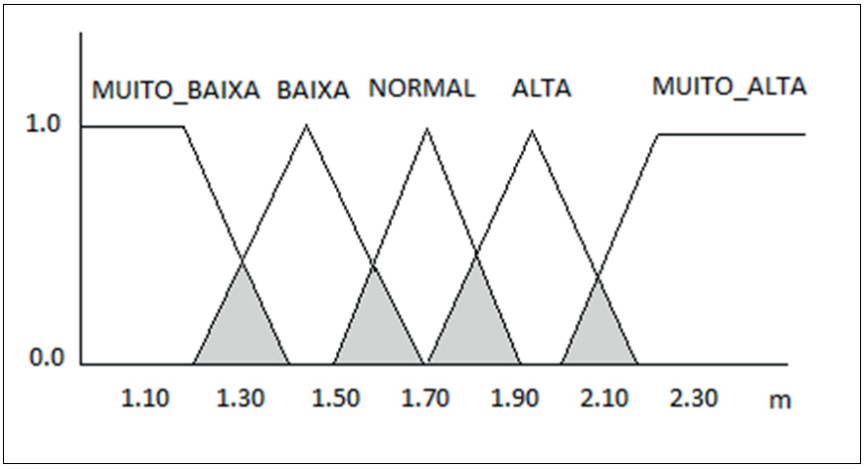
Na lógica difusa fazemos o uso extensivo das variáveis linguísticas e funções de pertinência. Uma variável linguística é um conceito como "altura", que pode ter um valor em uma faixa de valores nebulosos como "alto", "médio" e "baixo". Desta forma, a função de pertinência associa a cada elemento x pertencente a um conjunto A um número real no intervalo [0,1], que representa o grau de pertinência do elemento x a esse conjunto, isto é, o quanto o elemento X pertence ao conjunto A (COPPIN, 2010).

Percebemos a vantagem desse tipo de abordagem quando verificamos que a maior parte do conhecimento humano é imprecisa e, em muitos casos, a imprecisão pode ser interpretada mais facilmente. Por exemplo, nas sentenças "comece a frear 10 m antes do semáforo" e "comece a frear perto do semáforo",

a interpretação e a execução da segunda são mais fáceis do que a primeira, pois não temos como definir exatamente a distância de 10 m. Já o conceito de perto é impreciso, entretanto, nesse caso atende perfeitamente ao objetivo, que seria parar o automóvel sem cruzar o semáforo.

A Figura 27 ilustra o gráfico de uma variável linguística para representar a altura e suas funções de pertinência.

FIGURA 27 - VARIÁVEL LINGUÍSTICA PARA ALTURA

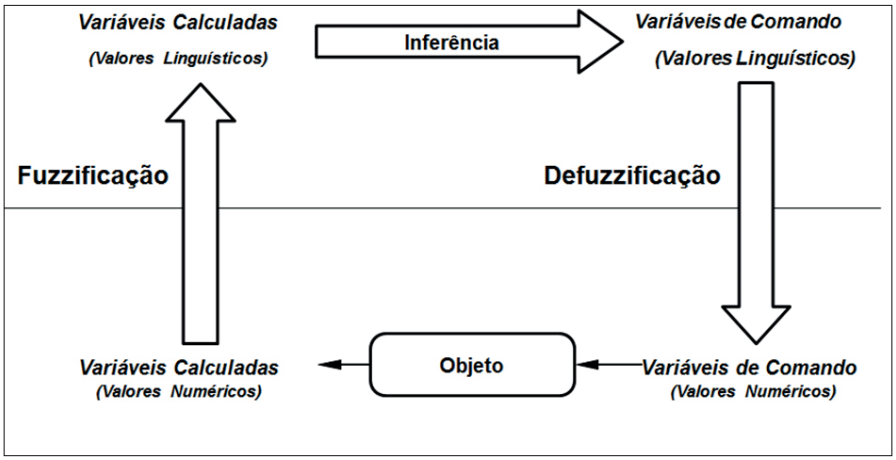


FONTE: O autor

Nessa figura podemos observar que foram criadas variáveis linguísticas cujo intervalo varia desde MUITO_BAIXA (de 0,0 m até aproximadamente 1,40 m) até MUITO_ALTA (de aproximadamente 2,0 m até 2,30 m, tendendo para o infinito). Cada variável linguística é representada por linhas dentro do gráfico, sendo que as linhas de MUITO_BAIXA e MUITO_ALTA formam um trapézio no início e no final do gráfico, respectivamente, e as linhas de BAIXA, NORMAL e ALTA formam três triângulos no centro do gráfico. Quanto mais próximo do valor 1,0 do eixo y, maior o grau de aderência ao conjunto. Quando um valor estiver contido nas áreas cinzas do gráfico, isso significa que ele está contido em dois conjuntos ao mesmo tempo. É principalmente nesses casos que podemos observar as vantagens da lógica difusa.

O funcionamento do raciocínio nos sistemas especialistas de lógica difusa e suas etapas podem ser vistos na Figura 28.

FIGURA 28 - ETAPAS DO RACIOCÍNIO DIFUSO



FONTE: O autor

Na fuzzificação, as variáveis linguísticas são definidas de forma subjetiva, bem como as funções de pertinência. Nesta etapa ainda é feita a análise do problema, a definição das variáveis, a definição das funções de pertinência e a criação das regiões. Em resumo, nessa etapa, os valores numéricos são transformados em valores linguísticos.

Na inferência Fuzzy, as regras são definidas e depois examinadas de forma paralela. Essa etapa encapsula a definição das proposições, a análise das regras e a criação da região resultante.

Finalmente, na defuzzificação, as regiões resultantes são convertidas de valores linguísticos para valores numéricos e então atribuídos para alguma variável de saída do sistema. Para essa etapa existem diversas técnicas, destacando-se a CENTROIDE e a FIRST-OF-MAXIMA.

Da mesma forma que nos sistemas especialistas convencionais, o conhecimento do sistema (neste caso representado pelas variáveis linguísticas, funções de pertinência, regras e proposições) é fornecido por especialistas ou retirado de análises numéricas.

As ferramentas para criação de sistemas especialistas de lógica difusa normalmente implementam de forma automática alguma das técnicas de defuzzificação. Para entender matematicamente de que forma essas técnicas ocorrem, acesse o *site*: <http://www.ufpi.br/subsiteFiles/rbritto/arquivos/files/aula03_Sistemas%20Fuzzy.pdf>.

Para finalizar o estudo dos sistemas especialistas de lógica difusa, demonstraremos parcialmente uma implementação através da ferramenta FuzzyCLIPS, disponível livremente em: <http://awesom.eu/~cygal/FuzzyCLIPS/fzclp610dWin.zip>.

A finalidade do sistema é controlar a distância de frenagem de um veículo com freios ABS, considerando como variáveis auxiliares a velocidade do mesmo e a pressão aplicada no pedal de frenagem. As variáveis linguísticas de entrada e saída e os valores atribuídos a elas são fictícios, servindo apenas para demonstração da implementação. Essas variáveis e as regras associadas a elas estão descritas no Quadro 14.

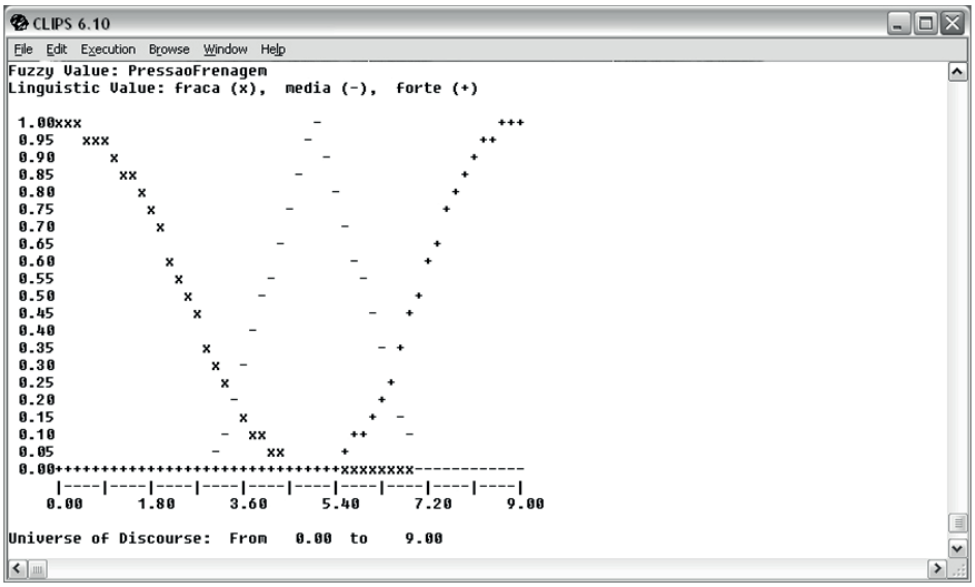
QUADRO 14 - VARIÁVEIS LINGUÍSTICAS PARA A DISTÂNCIA DE FRENAGEM DO ABS

	Velocidade		
Pressão da frenagem	BAIXA	MÉDIA	ALTA
FRACA	Muito longa	Longa	Média
MÉDIA	Longa	Média	Curta
FORTE	Média	Curta	Muito curta

FONTE: O autor

As regras do quadro funcionam da seguinte forma: se o automóvel estiver com velocidade ALTA e a pressão de frenagem for FORTE, isso significa que a distância de frenagem deve ser muito curta. Caso a pressão de frenagem seja FRACA e a velocidade BAIXA, a distância de frenagem deve ser muito longa. Cada uma dessas variáveis linguísticas deverá ter sua função de pertinência definida através de um intervalo de valores numéricos, caso contrário, o sistema perde sua utilidade. A pressão de frenagem é medida em Newtons, a distância de frenagem em metros e a velocidade em km/h. A Figura 29 ilustra os valores numéricos possíveis para a pressão na frenagem, considerando as variáveis linguísticas definidas no Quadro 14.

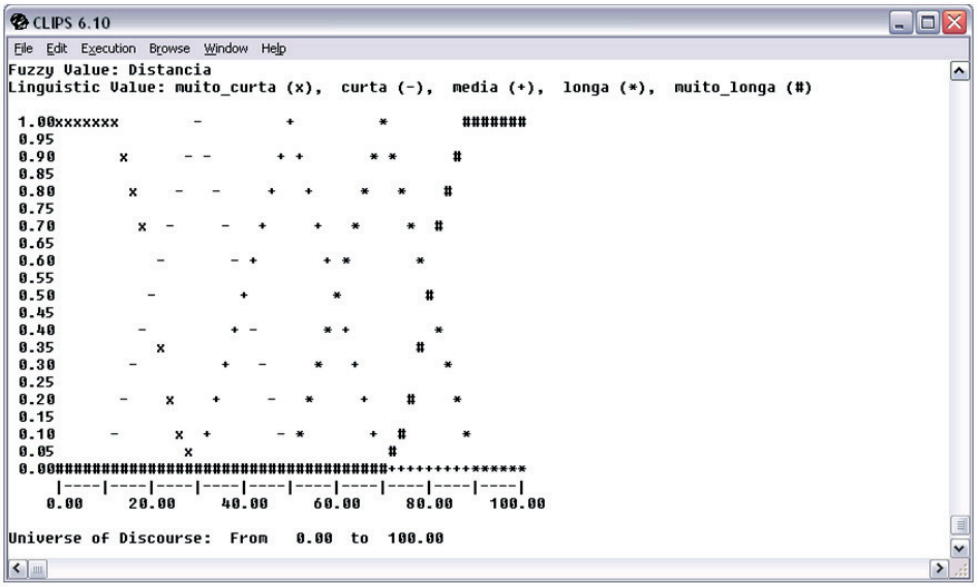
FIGURA 29 - GRÁFICO DOS VALORES NUMÉRICOS DA PRESSÃO DE FRENAGEM



FONTE: O autor

Nessa figura podemos perceber as funções de pertinência das variáveis linguísticas. A ferramenta desenha o gráfico de acordo com o símbolo definido na legenda da parte superior da tela. No nosso caso, a pressão de frenagem fraca é representada por asterisco (X) e varia de 0 a 5 Newtons. A pressão de frenagem média é definida por um hífen (-) e varia de 3 a 7 Newtons. Por fim, a pressão de frenagem forte é representada por um sinal de mais (+) e varia de 5 até 9 Newtons, tendendo para o infinito. As figuras 2.12 e 2.13 mostram as funções de pertinência para a distância de frenagem e velocidade, respectivamente.

FIGURA 30 - GRÁFICO DOS VALORES NUMÉRICOS DA DISTÂNCIA DE FRENAGEM

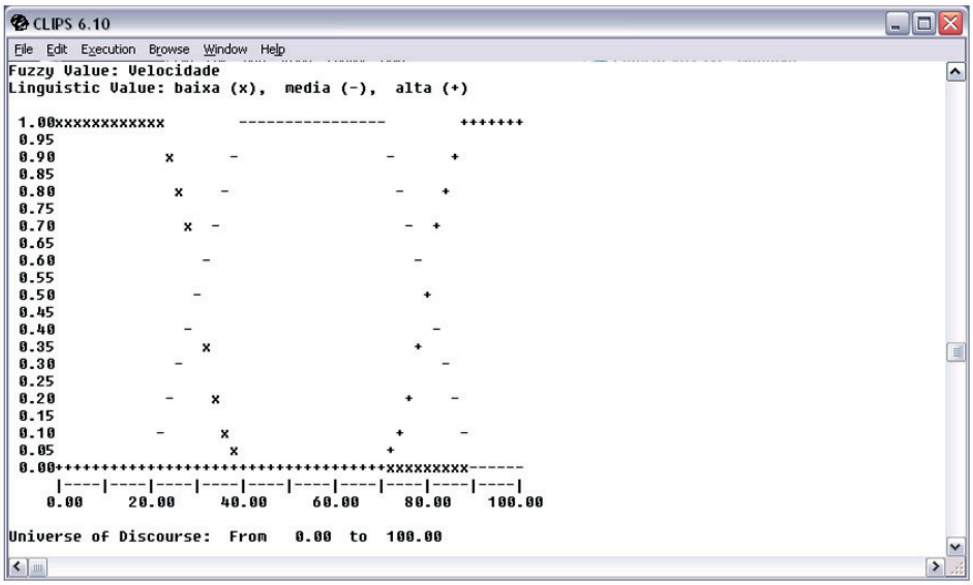


FONTE: O autor

Novamente podemos perceber as funções de pertinência das variáveis linguísticas para a distância de frenagem. Abaixo podem ser observadas as legendas e os valores para:

- **muito_curta**: é representada pela letra x (X) e varia de 0 até 30 metros;
- **curta**: é representada pelo hífen (-) e varia de 10 até 50 metros;
- **média**: é representada pelo mais (+) e varia de 30 até 70 metros;
- **longa**: é representada pelo asterisco (*) e varia de 50 até 80 metros e;
- **muito_longa**: é representada pelo sustenido (#) e varia de 80 até 100 metros, tendendo para o infinito.

FIGURA 31 - GRÁFICO DOS VALORES NUMÉRICOS DA VELOCIDADE



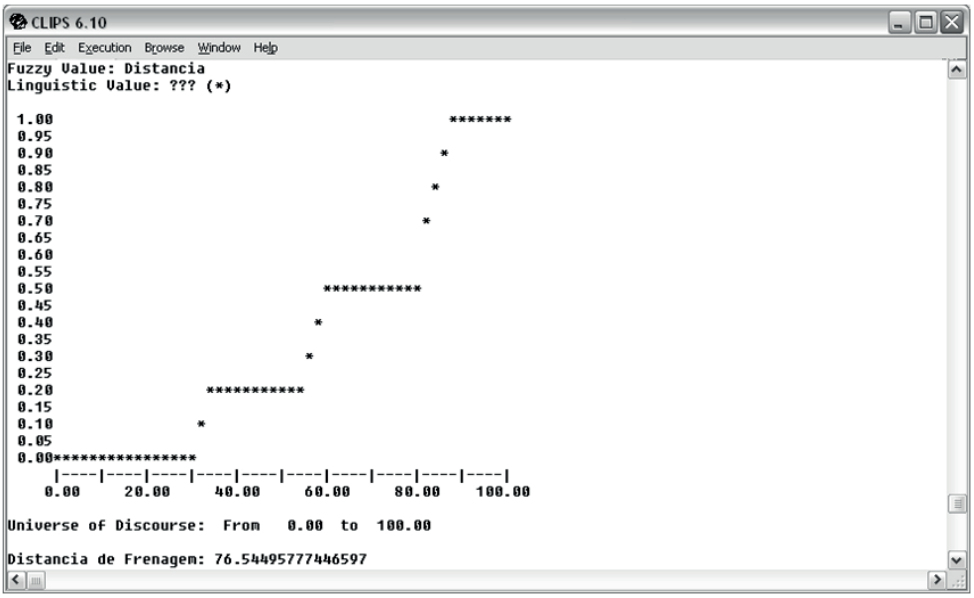
FONTE: O autor

A seguir mostramos as legendas e os valores para as variáveis linguísticas para a velocidade:

- **Baixa**: é representada pela letra x (X) e varia de 0 até 40 km/h;
- **Média**: é representada pelo hífen (-) e varia de 20 até 100 km/h;
- **Alta**: é representada pelo mais (+) e varia de 70 até 100 km/h, tendendo para o infinito.

Uma vez que todas as variáveis e regras foram definidas, podemos proceder com os valores de entrada e o consequente cálculo do valor otimizado de distância de frenagem pelo sistema. Na Figura 32, entramos com as informações de pressão de frenagem FRACA e velocidade BAIXA, resultando na distância de frenagem muito_longa como variável linguística e o valor aproximado de 76,54 m como variável numérica.

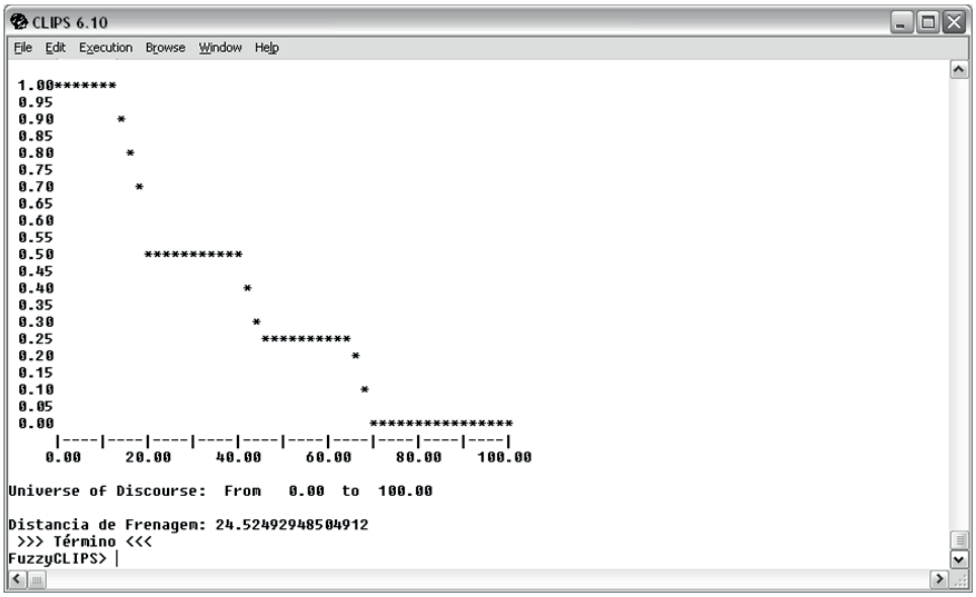
FIGURA 32 - VALOR RESULTANTE DE PRESSÃO DE FRENAGEM FRACA E VELOCIDADE BAIXA



FONTE: O autor

Na Figura 33, entramos com as informações de pressão de frenagem FORTE e velocidade ALTA, resultando na distância de frenagem muito_curta como variável linguística e o valor aproximado de 24,52 m como variável numérica.

FIGURA 33 - VALOR RESULTANTE DE PRESSÃO DE FRENAGEM FORTE E VELOCIDADE ALTA



FONTE: O autor

Podemos perceber que o sistema procedeu com a defuzzificação nas duas situações, utilizando para tal as regras, os valores linguísticos e os valores numéricos fornecidos na definição do mesmo. Vale salientar que o sistema calcularia quaisquer combinações de valores linguísticos disponíveis no Quadro 14 para velocidade e pressão de frenagem.

Essencialmente, o objetivo da lógica difusa é fazer com que as decisões tomadas pela máquina se aproximem cada vez mais das decisões humanas, principalmente ao trabalhar informações vagas e incertas. Entre as principais vantagens da lógica difusa, podemos citar:

- robustez, pois trabalha com entradas imprecisas;
- fácil manutenção, pois é baseada em regras;
- solução mais rápida e barata em alguns casos;
- implementável facilmente em microcontroladores;
- habilidade em codificar o conhecimento de uma forma próxima da linguagem usada pelos especialistas;
- mais fácil modelar sistemas envolvendo múltiplos especialistas.

Atualmente, algumas das grandes empresas de tecnologia fazem uso da lógica difusa em seus produtos. Como exemplo, podemos citar a Sony e suas câmeras fotográficas, a Nissan e seu sistema de transmissão automática automotiva e a Panasonic com suas câmeras de vídeo.

3 REDES BAYESIANAS

As redes bayesianas foram criadas por um reverendo presbiteriano chamado Thomas Bayes, que viveu na Inglaterra no início do século XVIII, conforme Figura 34. O processo de raciocínio idealizado por Thomas Bayes é tido hoje como uma nova forma de ver o mundo, como a base de uma verdadeira revolução em diferentes campos do conhecimento, da genética à teologia (PENA, 2006). Em resumo, a teoria de Thomas Bayes afirma que eventos passados alteram a probabilidade de ocorrência de eventos correlacionados no futuro.

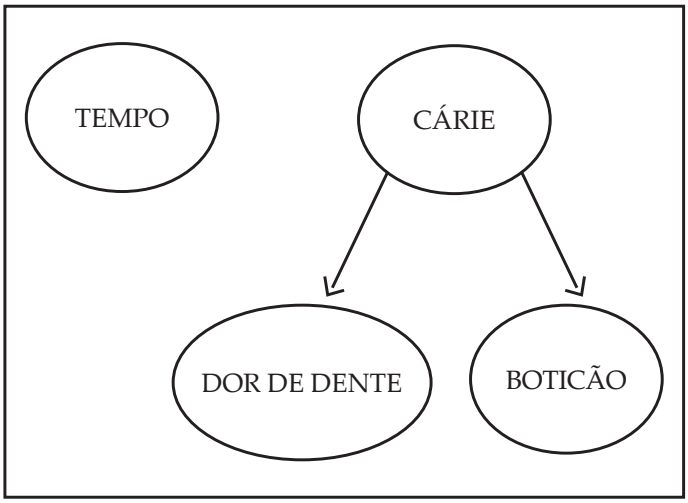
FIGURA 34 - THOMAS BAYES

FONTE: Disponível em: <http://commons.wikimedia.org/wiki/File:Thomas_Bayes.gif>. Acesso em: 5 mar. 2014.

Coppin (2010) coloca que essa teoria é amplamente usada nos dias atuais para lidar com situações em que não há certeza, servindo como base para os sistemas especialistas probabilísticos. As redes bayesianas permitem representar as dependências entre variáveis e fornecer uma especificação concisa de qualquer distribuição de probabilidade conjunta total. Elas consistem de grafos orientados, nos quais cada nó é identificado com informações de probabilidade quantitativa (NORVIG; RUSSEL, 2004).

A Figura 35 ilustra uma rede bayesiana com as variáveis tempo, cárie, dor de dente e boticão. A variável tempo é independente e não tem influência nas demais, enquanto as variáveis cárie e boticão dependem diretamente da ocorrência de cárie.

FIGURA 35 - REDE BAYESIANA SIMPLES



FONTE: Adaptada de: Norvig; Russel (2004)

Explicaremos o raciocínio bayesiano através de um exemplo médico retirado de Pena (2006). A partir dos 40 anos é recomendado que as mulheres façam mamografias anuais. Nessa faixa etária, 1% das mulheres são portadoras de um tumor assintomático de mama. Sabe-se que a mamografia apresenta resultados positivos em 80% das mulheres com câncer de mama e 9,6% das mulheres que não têm câncer de mama. Com base nessas informações, vamos calcular a probabilidade de uma mulher que fez a mamografia e obteve resultado positivo para câncer, esteja realmente doente.

Uma curiosidade sobre a teoria das probabilidades é ilustrada pelo problema de Monty Hall <http://pt.wikipedia.org/wiki/Problema_de_Monty_Hall>. Esse problema já foi mencionado também no cinema, no filme *Quebrando a banca* <http://www.adorocinema.com/filmes/filme-124755/>. O desafio aqui é você compreender a lógica por trás da solução do problema e fugir do senso comum.

Iniciamos fazendo cálculos de probabilidade *a priori* (antes de qualquer mamografia). Em média, 1% das mulheres com mais de 40 anos são portadoras de câncer, o que logicamente indica que 99% delas não são.

O próximo passo é a inclusão da probabilidade condicional no cálculo, nesse caso representada pela probabilidade de a mamografia estar certa no resultado positivo para câncer. Sabe-se que o exame tem grau de acerto de 80% nos casos positivos e grau de erro de 9,6%. Isso significa que 9,6% dos casos são falso-positivos, nos quais o exame mostrou positivo para câncer, mas a paciente não tem a doença. A compreensão fica mais fácil ao observarmos o Quadro 15, com as probabilidades já em formato decimal.

QUADRO 15 - PROBABILIDADE A PRIORI E CONDICIONAL

	TEM CÂNCER	NÃO TEM CÂNCER
Probabilidade *a priori*	0,01	0,99
Probabilidade condicional	0,8	0,096
Probabilidade conjunta	0,1 x 0,8 = 0,008	0,99 x 0,096 = 0,0095

FONTE: Adaptado de Pena (2006)

O cálculo da probabilidade conjunta é feito simplesmente pela multiplicação da probabilidade *a priori* pela probabilidade condicional. Podemos observar que a soma das probabilidades *a priori* é igual a 1, entretanto, isso não acontece com as probabilidades conjuntas. Para que isso aconteça é necessária uma etapa de normalização, na qual cada probabilidade conjunta é dividida pela soma de todas elas, como pode ser observado no Quadro 16.

QUADRO 16 - PROBABILIDADE A POSTERIORI

	TEM CÂNCER	NÃO TEM CÂNCER
Probabilidade *a priori*	0,01	0,99
Probabilidade condicional	0,8	0,096
Probabilidade conjunta	0,1 x 0,8 = 0,008	0,99 x 0,096 = 0,0095
Normalização	0,008 + 0,0095 = 0,0175	
Probabilidade a posteriori	0,008/0,0175 = 0,46	0,0095 / 0,0175 = 0,54

FONTE: Adaptado de Pena (2006)

Esse cálculo nos levará ao resultado da probabilidade *a posteriori*, ou seja, após fazer o teste. Através do raciocínio bayesiano, verificamos que a probabilidade de uma mulher que fez a mamografia e obteve resultado positivo para câncer estar realmente com a doença é de 0,46, ou 46%. Nesse exemplo, isso significa que a probabilidade de o exame estar incorreto e a mulher não ter a doença é maior, chegando a 54%.

Expressamos o raciocínio de Bayes matematicamente através da fórmula:

$$\Pr(A|B) = \frac{\Pr(B|A)\,\Pr(A)}{\Pr(B)}$$

onde Pr(A) e Pr(B) são as probabilidades *a priori* de A e B e Pr(B|A) e Pr(A|B) são as probabilidades *a posteriori* de B, sendo condicional a A e de A sendo condicional a B. Trazendo para o nosso exemplo, a probabilidade de alguém ter câncer de mama sabendo que a mamografia deu positivo para o teste depende não apenas da precisão da mamografia, mas também da probabilidade simples de uma mulher de mais de 40 anos ter câncer de mama.

Para exemplificarmos de que forma é possível sistematizar o raciocínio bayesiano, utilizaremos a ferramenta Genie para modelar o problema descrito a seguir:

O *software* Genie serve para fazer modelagem de redes bayesianas e simulação de cenários. Você pode fazer o *download* gratuito dessa ferramenta em: <http://genie.sis.pitt.edu/index.php/downloads>.

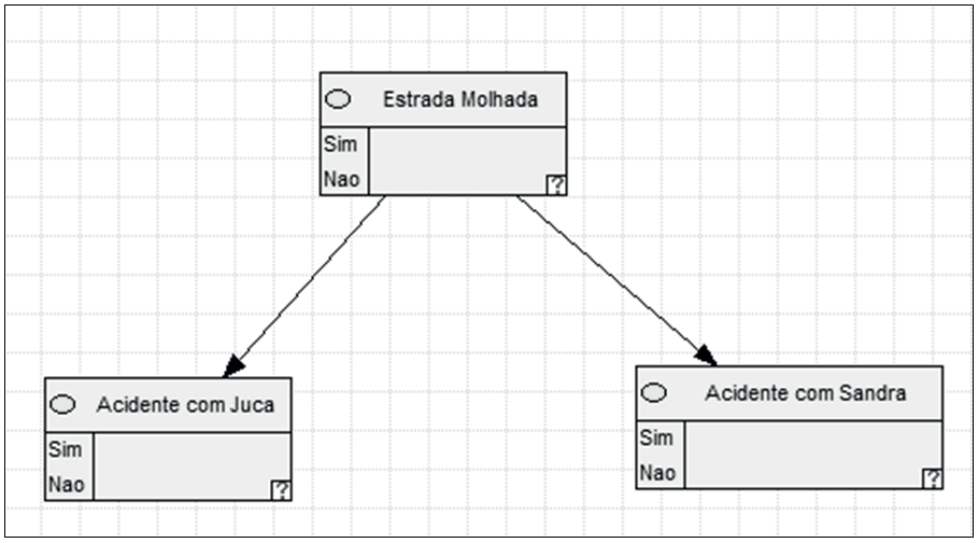
Juca e Sandra se dirigem para o aniversário de João, cada um em seu veículo. João os aguarda com certa preocupação, pois sabe que ambos são motoristas com pouca experiência. Além disso, João viu a previsão do tempo para a região indicando possibilidade de chuva, o que deixaria a estrada molhada e aumentaria a probabilidade de Juca e/ou Sandra sofrerem um acidente.

As seguintes probabilidades são dadas para os eventos citados anteriormente:

- Estrada molhada – 70% sim e 30% não.
- Juca sofrer acidente – se a estrada estiver molhada, 80% sim e 20% não. Se a estrada estiver seca, 90% não e 10% sim.
- Sandra sofrer acidente – se a estrada estiver molhada, 80% sim e 20% não. Se a estrada estiver seca, 90% não e 10% sim.

A primeira etapa na ferramenta é a modelagem da rede causal, conforme mostrado na Figura 36.

FIGURA 36 - REDE CAUSAL



FONTE: O autor

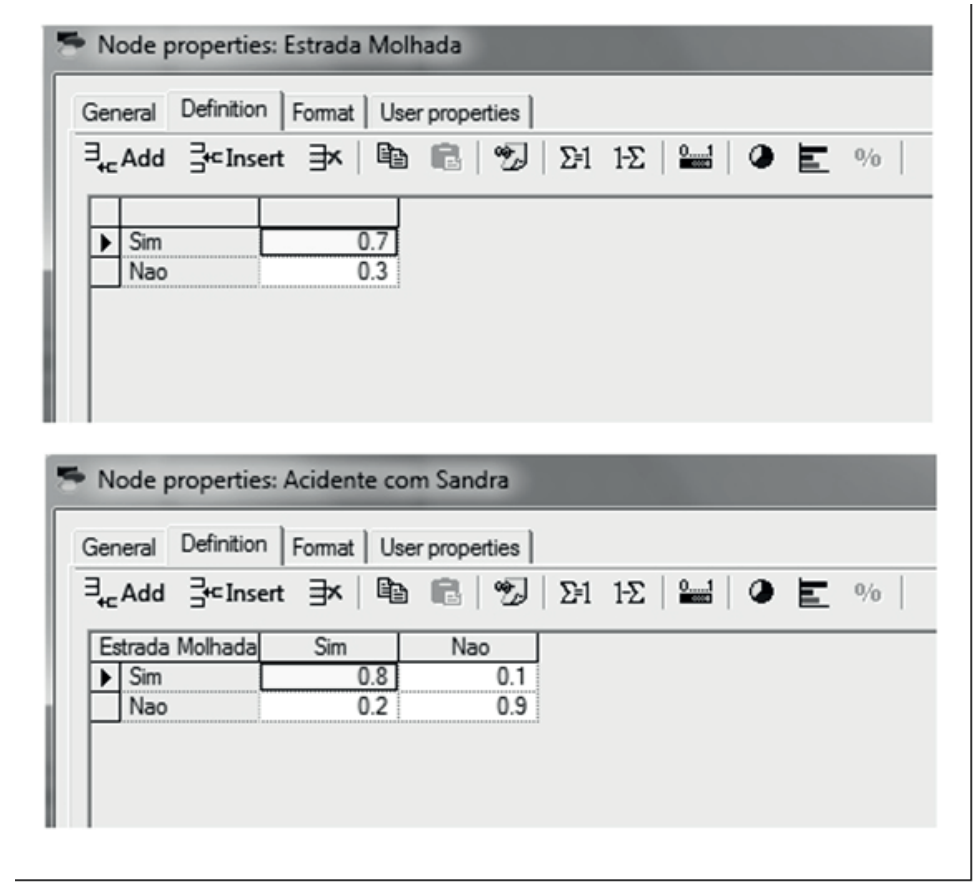
Nessa figura podemos ver os eventos e seus estados respectivos, neste caso, simplificados para sim ou não. O evento estrada molhada tem influência direta na ocorrência dos eventos Acidente com Juca e Acidente com Sandra, e essa relação condicional é representada na rede através de uma seta ligando o evento causa com o evento consequência.

A próxima etapa é atribuir as probabilidades para os estados dos eventos, conforme o enunciado de nosso problema. Na Figura 36 estão ilustradas as probabilidades para os eventos Estrada Molhada e Acidente com Sandra. Um detalhe a ser salientado é a configuração das probabilidades no evento Acidente com

Sandra. Como a probabilidade é condicional, esta deve ser atribuída considerando-se o evento anterior. Na figura podemos perceber que quando Estrada Molhada recebe sim, existe 80% de probabilidade de Sandra sofrer o acidente e 20% de não sofrer. Quando Estrada Molhada recebe não, existe somente 10% de probabilidade de que Sandra sofra um acidente e 90% de probabilidade que não. É sobre esses valores de probabilidade que o raciocínio bayesiano e consequentemente a ferramenta Genie operam. Os valores das probabilidades para o evento Acidente com Juca seriam exatamente iguais aos do evento Acidente com Sandra.

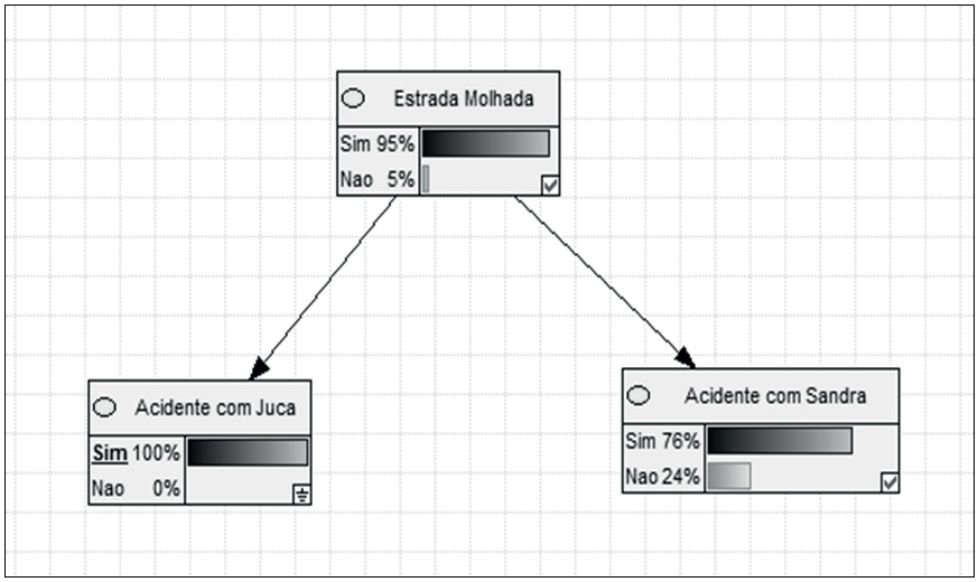
Para exemplificar de que forma os eventos influenciam uns aos outros, considere a seguinte situação: sabemos com 100% de certeza que Juca sofreu um acidente. Ao ter acesso a essa informação, poderíamos colocá-la na rede e atualizá-la, observando de que forma as demais probabilidades se comportam, conforme Figura 37.

FIGURA 37 - PROBABILIDADES DOS EVENTOS



FONTE: O autor

FIGURA 38 - ALTERAÇÃO DAS PROBABILIDADES DOS EVENTOS



FONTE: O autor

Nessa rede podemos observar, através do raciocínio bayesiano, que a certeza de que Juca sofreu um acidente faz com que a probabilidade de a estrada estar molhada aumente para 95%. Consequentemente, considerando essa alteração, agora temos 76% de certeza de que Sandra sofreu ou sofrerá um acidente, ilustrando a correlação entre causa e efeito.

Entre as principais aplicações para as quais as redes bayesianas vêm sendo utilizadas, podemos destacar:

a) diagnóstico de doenças: apneia do sono, narcolepsia, câncer de mama;
b) análise de risco de crédito: com base em variáveis como a duração do empréstimo, o histórico, o tempo de trabalho do solicitante e valor do empréstimo é possível determinar a probabilidade de o solicitante ser inadimplente;
c) classificação de superfícies: avalia a probabilidade de um pixel em uma foto tirada por satélite pertencer a uma determinada classe, tal como vegetação, solo exposto ou superfícies construídas.

E então? Após os estudos dessa seção e como Pena (2006) afirma em seu artigo, você concorda que Bayes "é o cara"?

RESUMO DO TÓPICO 1

- Existem basicamente três maneiras de representar e tratar a incerteza em sistemas especialistas: fator de confiança (menos utilizado), redes bayesianas e lógica difusa.

- Os sistemas especialistas de lógica difusa buscam representar os relacionamentos do mundo real de uma forma mais precisa, não se limitando aos conceitos de verdadeiro e falso.

- Na lógica difusa fazemos o uso extensivo das variáveis linguísticas e funções de pertinência. Uma variável linguística é um conceito como "altura", que pode ter um valor em uma faixa de valores nebulosos como "alto", "médio" e "baixo".

- As principais etapas de um SE de lógica difusa são a fuzzyficação, a defuzzyficação e a inferência fuzzy.

- Da mesma forma que nos sistemas especialistas convencionais, o conhecimento do sistema (neste caso representado pelas variáveis linguísticas, funções de pertinência, regras e proposições) é fornecido por especialistas ou retirado de análises numéricas.

- As redes bayesianas são usadas nos dias atuais para lidar com situações em que existem dados estatísticos para os eventos, servindo como base para os sistemas especialistas probabilísticos.

- As redes bayesianas atuam através de cálculos efetuados com a probabilidade de ocorrência de eventos passados e sua influência sobre eventos correlacionados no futuro.

- A probabilidade *a priori* é a probabilidade de ocorrência de determinado evento antes que ele aconteça.

- Entre as principais aplicações de redes bayesianas podemos destacar: diagnóstico de doenças, análise de risco de crédito e classificação de imagens.

AUTOATIVIDADE

1 Os Sistemas Especialistas Probabilísticos (SEP) têm por base o raciocínio proposto por Thomas Bayes, nos quais eventos passados exercem influência em eventos correlacionados no futuro. Com relação aos SEP, seu modo de operação e suas características, assinale a alternativa CORRETA:

( ) Os SEP são utilizados geralmente para situações onde as informações são corretas e precisas.
( ) A defuzzyficação é o mecanismo que faz o encadeamento das probabilidades nos SEP.
( ) A probabilidade a priori é a probabilidade de que um evento ocorra após a ocorrência de um evento causal relacionado.
( ) Uma rede causal é um grafo dirigido que modela os eventos e seus relacionamentos na representação de um problema passível de resolução através dos SEP.

2 Avalie as afirmações a seguir sobre os sistemas especialistas de lógica difusa:

I – Os sistemas especialistas de lógica difusa se baseiam na premissa de que praticamente todas as informações a serem utilizadas em um sistema podem ser representadas em função de verdadeiro e falso.
II – Uma variável linguística é um conceito como "altura", que pode ter um valor em uma faixa de valores nebulosos como "alto", "médio" e "baixo".
III – Da mesma forma que nos sistemas especialistas convencionais, o conhecimento do sistema é fornecido por especialistas ou retirado de análises numéricas.
IV – A defuzzificação é a etapa da transformação dos valores numéricos em variáveis linguísticas.

Agora assinale a alternativa que somente possui afirmações VERDADEIRAS:

a) I, II e III.
b) I e IV.
c) I, II, IV.
d) II e III.

3 Avalie as afirmações a seguir sobre os sistemas especialistas probabilísticos, colocando a letra V para afirmações VERDADEIRAS e a letra F para as FALSAS:

( ) As redes bayesianas são usadas principalmente para lidar com situações onde não existem informações prévias sobre os eventos.

( ) A probabilidade a posteriori é aquela probabilidade de ocorrência de determinado evento antes que ele aconteça.
( ) Entre as principais aplicações das redes bayesianas podemos citar: análise de risco de crédito e classificação de imagens.
( ) Uma rede causal é um grafo orientado onde percebe-se a relação entre eventos atuais e eventos futuros.

Agora assinale a alternativa que apresenta a sequência CORRETA:

a) F, V, V, V.
b) F, F, V, V.
c) F, V, F, V.
d) F, V, V, F.

4 Um conjunto difuso é um conjunto que se diferencia de um conjunto tradicional por existir uma suavização nos estados pertencer ou não pertencer a determinada condição. Por exemplo, em um conjunto tradicional, uma pessoa seria caracterizada como criança ou adulto, enquanto em um conjunto difuso, uma pessoa poderia ser ainda caracterizada como adolescente, um estado intermediário entre os dois anteriores. Com relação aos conjuntos difusos, avalie as afirmações a seguir:

I - As variáveis linguísticas são utilizadas frequentemente para determinar intervalos de valores dentro de um conjunto difuso.
II - A fuzzyficação é a etapa onde valores numéricos são convertidos para valores linguísticos.
III - Os conjuntos difusos são a base para o funcionamento dos sistemas especialistas de lógica difusa.
IV - Ao representarmos um conjunto difuso em um gráfico, o grau de aderência de um ponto ao conjunto é determinado pela proximidade com o valor 0 do eixo y, ou seja, quanto mais próximo do 0, mais aderente ao conjunto.

Agora assinale a alternativa que somente possui afirmações CORRETAS:

I, II e III.
I, III e IV.
II e III.
III e IV.

5 A lógica difusa faz uso de funções de pertinência e variáveis linguísticas para modelar problemas de uma forma mais similar à do mundo real, em um processo conhecido como tratamento de incerteza. Referente aos sistemas especialistas de lógica difusa, avalie as afirmações abaixo:

I – Na etapa conhecida como fuzzyficação, as variáveis linguísticas são associadas às funções de pertinência.
II – Na etapa conhecida como fuzzyficação, os valores numéricos são transformados em valores linguísticos.

III – Centroide é um dos algoritmos conhecidos para converter valores numéricos em valores linguísticos.
IV – Na inferência fuzzy, as regras e fatos são avaliados de forma paralela, buscando novos fatos e/ou conclusões.
V – Na defuzzyficação, as variáveis linguísticas são convertidas para valores numéricos e geralmente exibidas como conclusão.

Agora assinale a alternativa que somente possui afirmações CORRETAS:

I, II, III e IV.
I, II, IV e V.
I, II e IV.
I, II e III.

TÓPICO 2

COMPUTAÇÃO EVOLUTIVA

1 INTRODUÇÃO

O naturalista britânico Charles Darwin, em sua obra histórica de 1859, "*A origem das espécies*", propôs a ideia da evolução natural. Sua principal proposta é a de que, ao longo de diversas gerações, organismos biológicos evoluem baseados no princípio da seleção natural da "sobrevivência do mais apto". As formas perfeitas do albatroz, que maximizam sua eficiência no voo, ou a similaridade entre o golfinho e o tubarão são alguns dos melhores exemplos de realização da evolução aleatória (SIVANANDAM; DEEPA, 2008). Hoje, a evolução natural é reconhecida como fato e não mais como apenas teoria, tendo sido inúmeras vezes confirmada por verificações científicas. Não é difícil perceber porque seria tão interessante simular um método que funciona tão bem na natureza, na resolução de problemas concretos de busca e otimização.

É com base nestes fundamentos que opera a Computação Evolutiva. A computação evolutiva emprega algoritmos de otimização global, baseados nos mecanismos de seleção natural e da genética destacados por Darwin. Eles utilizam estratégias de busca aleatória, mas direcionadas, nas quais informações históricas servem de subsídio para encontrar novos pontos de busca em que se espera obter melhores desempenhos.

Para apresentar os principais conceitos da computação evolutiva, abordaremos os **algoritmos genéticos** e as **estratégias de evolução**, detalhando suas características e seu modo de operação. Com esse conhecimento, você entenderá o porquê de a computação evolutiva ser considerada uma alternativa extremamente interessante em processos de busca e otimização.

2 FUNDAMENTOS DA EVOLUÇÃO

Na natureza, um indivíduo pertencente a uma população compete com cada um de seus semelhantes por recursos como comida e abrigo. Ainda na mesma espécie, indivíduos competem entre si para atrair parceiros para reprodução. Devido a esta seleção, indivíduos que possuem desempenho inferior têm menos chance de sobreviver e os indivíduos mais aptos produzem uma prole relativamente maior.

A seleção natural, um dos mecanismos básicos da evolução, apesar de ser um conceito extremamente poderoso, é também surpreendentemente simples. No entanto, o funcionamento da seleção natural é frequentemente mal-entendido. Para compreender seu funcionamento, imagine uma população de besouros:

a) há na população uma variedade de traços: alguns besouros são verdes, e outros são marrons;

b) há reprodução diferenciada: uma vez que o ambiente não pode comportar um crescimento ilimitado da população, nem todos os indivíduos podem se reproduzir à vontade. Neste exemplo, os besouros verdes tendem a ser devorados por pássaros e sobrevivem para reproduzir menos frequentemente do que os besouros marrons;

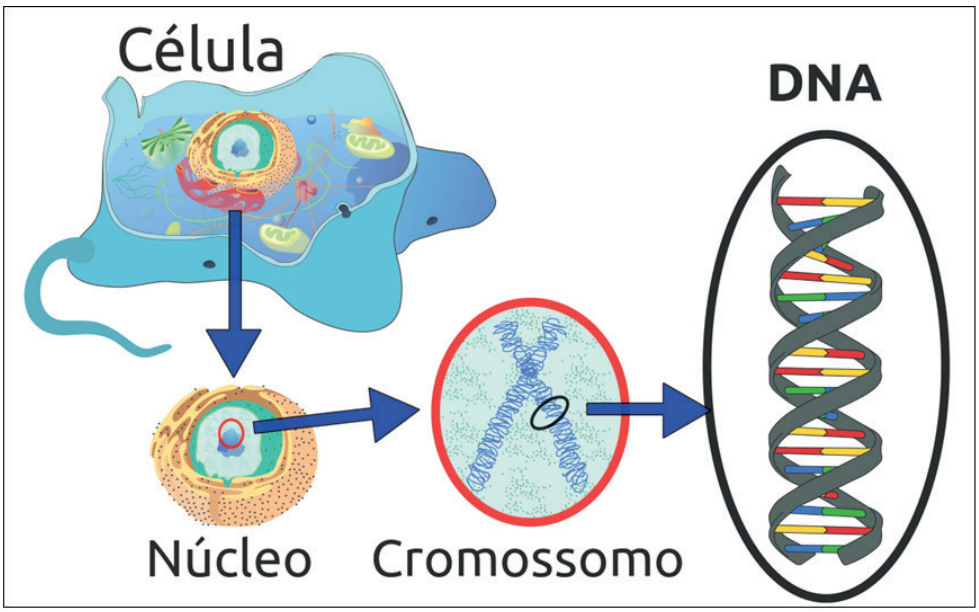
c) há hereditariedade: os besouros marrons sobreviventes têm filhos marrons, porque seus traços são hereditários;

d) resultado final: o traço mais vantajoso, a cor marrom, que permite que o besouro tenha mais filhos, se torna mais comum na população. Se este processo continuar, eventualmente, todos os indivíduos na população serão marrons (University of California Museum of Paleontology and National Center for Science Education, 2004).

Um aspecto importante a se observar é a recombinação genética proporcionada pelo acasalamento. Durante a reprodução, uma recombinação das boas características de cada ancestral pode produzir uma prole "mais apta", cuja aptidão é maior ainda do que de seus pais. Depois de algumas gerações, as espécies evoluem espontaneamente para se tornarem mais e mais adaptadas ao seu ambiente (SIVANANDAM; DEEPA, 2008).

A ciência que estuda os mecanismos responsáveis pelas semelhanças e diferenças entre os organismos de uma espécie é chamada de genética. A genética nos auxilia a compreender as similaridades e as variações da hereditariedade natural dos seres vivos. Os conceitos de algoritmos genéticos foram desenvolvidos diretamente baseados na evolução natural. É fundamental conhecer os conceitos principais da genética para poder entender os algoritmos genéticos.

FIGURA 39 - LOCALIZAÇÃO DA INFORMAÇÃO GENÉTICA NA CÉLULA



FONTE: Adaptado de: <http://en.wikipedia.org/wiki/File:Eukaryote_DNA-en.svg>. Acesso em: 22 mar. 2017.

Toda célula animal é um complexo de "pequenas fábricas" que trabalham juntas, tendo como seu centro o núcleo celular, como pode ser visualizado na Figura 39. O núcleo celular é a estrutura que contém a informação genética da célula. No núcleo, toda a informação genética da célula é armazenada nos cromossomos. Os cromossomos são cadeias de ácido desoxirribonucleico (DNA). Os cromossomos são divididos em diversas partes chamadas genes. São os genes que codificam ou "contêm" as características de um indivíduo, bem como de sua espécie. Existe, por exemplo, um gene para a cor dos olhos e este gene pode definir os olhos negros, castanhos, azuis ou verdes de uma pessoa. Cada uma destas possibilidades é chamada de alelo.

Ao conjunto de todos os possíveis alelos presentes na população de uma espécie se dá o nome de conjunto de genes. Este conjunto de genes pode determinar todas as diferentes variações possíveis nas futuras gerações de uma espécie. O conjunto de todos os genes de uma espécie é chamado de genoma.

A combinação de genes presente em um indivíduo é chamada genótipo, enquanto o aspecto físico da "implementação" de um genótipo é chamado de fenótipo. Um aspecto interessante da evolução é que a seleção natural dos organismos é feita pelo fenótipo, enquanto que a reprodução recombina o seu genótipo e assim o desenvolvimento da forma de cada organismo representa um papel-chave na seleção e evolução.

Nos seres vivos superiores, os cromossomos contêm dois conjuntos de genes. A estes seres vivos se dá o nome de diploides. Durante a reprodução destes seres, a informação genética de um indivíduo é recombinada, pois o indivíduo recebe um conjunto de genes do pai e outro da mãe. Em caso de conflitos entre dois valores do mesmo par de genes, o gene dominante irá determinar o fenótipo, enquanto o outro, chamado recessivo, ainda estará presente e pode ser transmitido aos descendentes. Como já mencionado, esta recombinação é muito importante para a diversidade genética, pois permite uma maior diversidade de alelos, o que serve como um mecanismo de memória em um ambiente que muda com frequência.

Por outro lado, os organismos que possuem somente um conjunto de genes são chamados de haploides. A maioria dos algoritmos genéticos se foca em cromossomos haploides, porque eles são muito mais simples de se construir. Nos haploides, como somente um conjunto de cada gene é armazenado, o processo de determinar qual alelo deve ser dominante e qual deve ser recessivo é evitado.

Informações adicionais sobre o princípio da evolução podem ser encontradas em: <http://www.ib.usp.br/evosite/evohome.html>.
Informações adicionais sobre a genética podem ser encontradas em: <http://pt.wikipedia.org/wiki/Introdu%C3%A7%C3%A3o_%C3%A0_gen%C3%A9tica>.

3 COMPUTAÇÃO EVOLUTIVA

Por mais estranho que possa parecer, a ideia de aplicar princípios darwinianos à solução automatizada de problemas surgiu nos anos 1940, antes mesmo da revolução dos computadores. As primeiras referências que se tem a respeito desta ideia são do próprio Alan Turing, que em 1948 propôs uma "busca genética ou evolucionária". Durante os anos 1960, três diferentes tentativas de se implementar o conceito central foram feitas em lugares diferentes: nos Estados Unidos, L. J. Fogel, A. J. Owens e M. J. Walsh introduziram a "Programação Evolutiva", e J. H. Holland chamou seu método de "Algoritmo Genético". Enquanto isso, na Alemanha, I. Rechenberg e H.-P. Schwefel desenvolveram "Estratégias de Evolução". Durante muito tempo estas áreas evoluíram independentemente, até que no início dos anos 1990 elas começaram a ser consideradas diferentes "dialetos" de uma única tecnologia que passou a ser chamada Computação Evolutiva. Desde então, a terminologia denota os algoritmos envolvidos na computação evolutiva de algoritmos evolucionários, e considera programação evolutiva, estratégias de evolução, algoritmos genéticos e programação genética como subáreas desta (EIBEN; SMITH, 2003).

O que todas estas técnicas possuem em comum são quatro conceitos fundamentais: todas envolvem reprodução, variação aleatória, competição e seleção de indivíduos em uma população. Estes conceitos formam a essência da evolução e quando estes quatro processos são colocados em funcionamento, a evolução é o resultado inevitável, seja na natureza ou em um computador (BÄCK; FOGEL; MICHALEWICZ, 2000).

Evolução é um processo de **otimização**. Apesar de otimização não ser igual à perfeição, a evolução pode chegar a soluções altamente precisas e funcionais. Desta forma, é muito comum a utilização de métodos evolucionários que possam resolver problemas de otimização difíceis. A evolução provê a inspiração para a computação de soluções para problemas que anteriormente eram insolúveis.

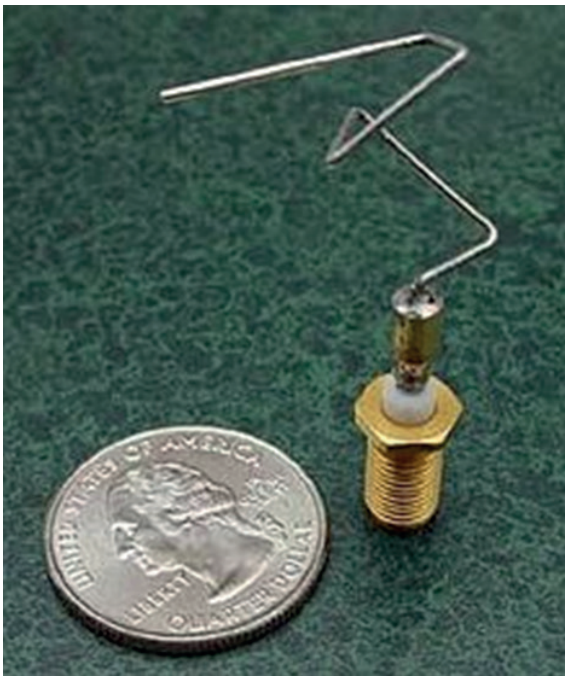
O mundo real nunca é estático e alguns problemas são muito desafiadores, pois requerem estratégias de comportamento que mudam com o tempo. Por exemplo: na otimização de um problema dinâmico, a escolha de uma estratégia para melhorar o resultado pode depender dos resultados das estratégias utilizadas até o momento. Neste aspecto, a Computação Evolutiva pode explorar o potencial para recombinar fragmentos bem-sucedidos de estratégias em novas abordagens para se adaptar melhor aos problemas. O resultado é uma capacidade robusta de adaptação, capaz de ajustar a performance com base nos resultados obtidos no ambiente.

Inteligência pode ser definida como a capacidade de um sistema de adaptar seu comportamento de acordo com os seus objetivos. A evolução criou na natureza seres de inteligência cada vez maior ao longo do tempo. A Computação Evolutiva é uma alternativa à tentativa de criar inteligência de máquina replicando os humanos, seja através de regras ou de suas conexões neurais.

Em muitos casos, sistemas especialistas e outras tentativas de imitar a inteligência humana são frágeis: eles não são robustos a mudanças no domínio da aplicação e são incapazes de predizer corretamente circunstâncias futuras, bem como tomar a decisão apropriada. Por outro lado, com a utilização correta do fator aleatório, as possibilidades são muito grandes para os sistemas evolucionários (BÄCK; FOGEL; MICHALEWICZ, 2000).

Um exemplo de aplicação prática recente de algoritmos evolucionários é a criação de antenas para finalidades específicas. A Figura 40 mostra uma antena criada para a espaçonave ST5 da NASA. O formato da antena foi encontrado por um programa de desenho evolutivo, de forma a criar o melhor padrão de radiação possível. O programa começa com alguns formatos simples, e então vai adicionando ou modificando elementos, de maneira a criar novos formatos de antenas candidatas. Estes novos formatos são então avaliados para determinar o quão bem eles atendem aos requisitos iniciais. Em seguida, o algoritmo elimina os formatos que tiveram os piores resultados, para então começar a gerar novas alternativas a partir dos formatos que não foram eliminados (HORNBY et al., 2006).

FIGURA 40 - ANTENA DA ESPAÇONAVE ST5 DA NASA



FONTE: Hornby et al. (2006)

A ideia de encontrar uma solução desejada em meio a uma coleção de possíveis soluções é tão comum na Ciência da Computação que chega a ter um termo próprio: pesquisa em um "espaço de busca". Um exemplo prático deste tipo de problema é a busca de proteínas, em bioengenharia computacional (MITCHELL, 1999). Você quer usar o computador para encontrar uma sequência de aminoácidos (uma proteína) que possui uma determinada forma tridimensional que pode ser usada, digamos, para combater um vírus específico. O espaço de busca é o conjunto de todas as sequências possíveis. Como este espaço de busca seria infinito, você restringe a busca a todas as sequências possíveis com comprimento de 100 ou menos aminoácidos. Este ainda é um espaço de busca enorme, porque há 20 possíveis aminoácidos em cada posição na sequência.

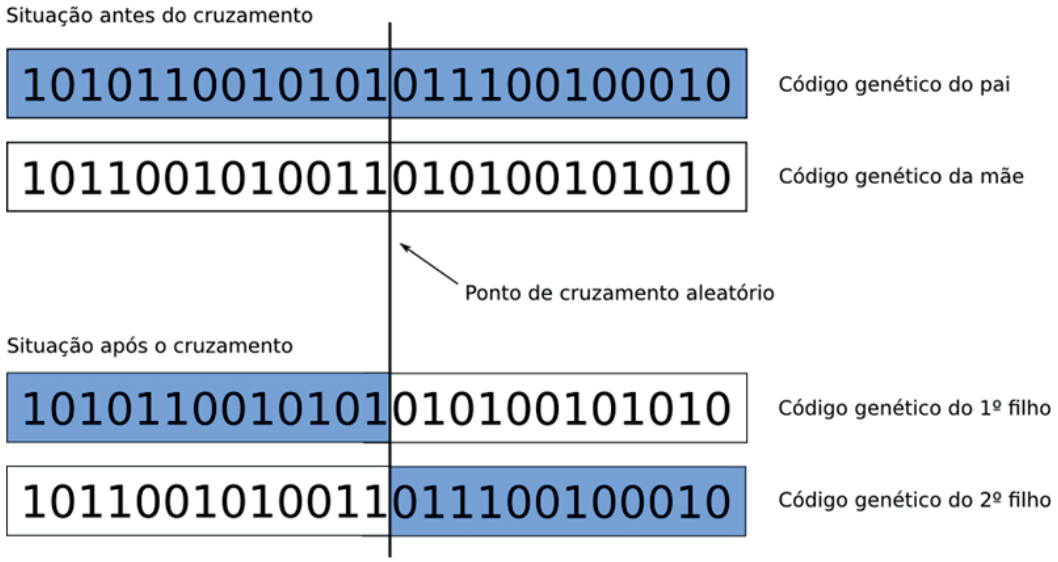
Como já mencionado, quatro paradigmas históricos servem como base para muito do que se produz na área da computação evolutiva (SIVANANDAM; DEEPA, 2008): algoritmos genéticos, programação evolutiva, estratégias evolutivas e programação genética. A partir de agora, este material vai delinear as características de cada uma, diferenciando-as umas das outras.

4 ALGORITMOS GENÉTICOS

Algoritmos genéticos são a técnica mais popular na área da computação evolutiva. Nesta técnica, a representação utilizada é uma *cadeia de bits* de tamanho fixo, onde cada *bit* representa uma característica particular de um indivíduo. Esta cadeia é interpretada como uma coleção de características estruturais de uma solução possível para um problema. Esta é uma analogia direta aos genes dos organismos biológicos.

O principal operador de reprodução utilizado nesta técnica é o cruzamento, no qual duas cadeias são usadas como pais e novos indivíduos são formados misturando-se subsequências entre as duas cadeias, conforme pode ser visto na Figura 41. Em geral, os indivíduos selecionados para o cruzamento são escolhidos probabilisticamente com base em seus níveis de adaptação.

FIGURA 41 - CÓDIGO GENÉTICO DOS PAIS E DE SUA PROLE ANTES/APÓS O CRUZAMENTO



FONTE: adaptado de: <http://www.ro.feri.uni-mb.si/predmeti/int_reg/Predavanja/Eng/3.Genetic%20algorithm/images/pic005.png>.

O Quadro 17, a seguir, apresenta um resumo das principais propriedades características dos Algoritmos genéticos, que os diferencia das demais técnicas dentro da Computação Evolutiva.

QUADRO 17 - PROPRIEDADES DOS ALGORITMOS GENÉTICOS

Representação	Cadeias binárias
Recombinação	Cruzamento simples
Mutação	Troca de bits
Seleção dos pais	Adaptação proporcional
Seleção dos sobreviventes	A cada geração

FONTE: Adaptado de Eiben; Smith (2003)

Os algoritmos genéticos são uma classe de técnicas de computação evolucionária que foi proposta e analisada por John Holland e seus colegas na Universidade de Michigan, por volta de 1975 (BÄCK; FOGEL; MICHALEWICZ, 2000).

Os principais objetivos de sua pesquisa eram:

a) abstrair e explicar rigorosamente os processos adaptativos dos sistemas naturais;
b) produzir *software* para sistemas artificiais que fosse capaz de reproduzir os mecanismos importantes dos sistemas naturais.

O tema central da pesquisa em algoritmos genéticos foi a robustez e o equilíbrio entre eficiência e eficácia. As implicações destes parâmetros em sistemas artificiais são muitas, por exemplo:

a) se um sistema for robusto, ele precisará de menos ajustes custosos;
b) se maiores níveis de adaptação forem atingidos, o sistema pode rodar melhor e por mais tempo (GOLDBERG, 1999).

Em um algoritmo genético, uma população de possíveis soluções para um problema de otimização é evoluída para melhores soluções. Estas possíveis soluções são chamadas de indivíduos ou fenótipos. Cada indivíduo tem um conjunto de propriedades, o seu genótipo, que pode sofrer mutações e ser alterado. É bastante comum representar estas soluções em cadeias binárias de 0 ou 1, mas outras formas de codificação também podem ser usadas.

Normalmente, a evolução se inicia com uma população de indivíduos gerados aleatoriamente. O processo de evolução é iterativo, e a população criada a cada iteração é chamada de geração. Depois que uma nova geração é produzida, o nível de adaptação de cada indivíduo daquela população é medido. Quanto mais "bem adaptado" o organismo estiver, mais próximo ele estará do objetivo da otimização. Os indivíduos mais bem adaptados são então selecionados, e o seu genótipo é modificado para produzir uma nova geração. Esta modificação dos genótipos pode ser feita através de recombinação e/ou mutação aleatória, com o objetivo de simular a reprodução dos organismos biológicos. Esta nova geração de indivíduos é então utilizada para a próxima iteração do algoritmo. O algoritmo genético normalmente termina quando:

a) um número máximo de gerações foi produzido; ou
b) um nível satisfatório de adaptação foi atingido pela população.

Para que você possa compreender os detalhes do funcionamento de um algoritmo genético, vamos descrever o seu uso em uma aplicação cotidiana (GOLDBERG, 1999): suponha que temos uma função matemática $f(x) = x^2$, onde x possa ter valores entre 0 e 31. Tudo o que a função faz é elevar o valor de entrada que é fornecido ao quadrado. O que desejamos fazer é encontrar o máximo desta função. É óbvio que a solução é o valor $x = 31$, mas o que queremos na realidade é entender o funcionamento do algoritmo genético.

Em geral, o algoritmo genético irá iniciar com uma população aleatoriamente escolhida, composta de uma série binária. Uma função de critério nos ajudará a determinar a qualidade dos indivíduos desta população, e com base nesta informação, o número de cópias que cada indivíduo terá ao produzir a próxima geração.

QUADRO 18 - CÁLCULO DAS CARACTERÍSTICAS DA POPULAÇÃO INICIAL

Indivíduo	População inicial (aleatória)	Valor da variável x	Resultado da função $f(x) = x^2$ f_i	Taxa de adaptação $\frac{f_i}{f_{avg}}$	Número de filhos
1	01101	13	169	0,58	1
2	11000	24	576	1,97	2
3	01000	8	64	0,22	0
4	10011	19	361	1,23	1
Soma			**1170**	**4,00**	**4**
Média f_{avg}			**239**	**1,00**	**1**
Máximo			**576**	**1,97**	**2**

FONTE: O autor

O Quadro 18 mostra como é feito o cálculo das características da população inicial, que foi gerada aleatoriamente. A partir da sequência binária aleatória de cada indivíduo é gerado o valor da variável x (13 para o indivíduo 1, 24 para o indivíduo 2, e assim por diante). Na coluna seguinte é calculado o resultado da função para cada indivíduo. Chamaremos este resultado de f_i.

A seguir, calculamos a taxa de adaptação, que é também chamada de **função de adaptação**, porque mede o quão próximo um indivíduo está de nosso objetivo. Neste caso, o que queremos buscar é o valor de x que gera o maior valor quando aplicado à função $f(x)$. Por isso, calculamos a média dos valores gerados por cada indivíduo, a que chamamos de f_{avg}. Em seguida, dividimos o valor produzido por cada indivíduo pela média de todos, o que produz um número que chamamos de

taxa de adaptação. Quanto maior esta taxa de adaptação, mais próximo estamos de nosso objetivo. Com base na taxa de adaptação, decidimos então quantos filhos cada indivíduo terá ao produzir a próxima geração. Perceba que o indivíduo 3, que possui o menor valor de *x*, não produzirá nenhum filho, pois é o valor que menos se aproxima de nosso objetivo. Enquanto isso, o indivíduo 2, que possui o maior valor de *x*, terá um maior número de filhos.

Neste momento se dá o processo de produção da próxima geração. Para gerar os novos indivíduos, os cromossomos dos organismos são recombinados, sendo que parte do código genético vem do pai e a outra parte vem da mãe. O Quadro 19 mostra um exemplo de como pode acontecer o processo de cruzamento.

Desta forma, temos no Quadro 19 a descrição do processo que cria a segunda geração de organismos. O indivíduo 1 é selecionado para cruzamento com o indivíduo 2, e o ponto de cruzamento é a posição 4. Isto significa que na criação do novo indivíduo, as quatro primeiras posições do seu cromossomo serão herdadas do indivíduo 1, e a quinta posição é herdada do indivíduo 2. As posições do cromossomo que serão copiadas do parceiro de cruzamento estão marcadas em vermelho.

QUADRO 19 - CRUZAMENTO E CÁLCULO DAS CARACTERÍSTICAS DA PRIMEIRA GERAÇÃO

Primeira geração	Parceiro de cruzamento aleatório	Ponto de cruzamento (aleatório)	Novo indivíduo	Valor da variável x	$f(x) = x^2$ $\frac{f_i}{f_{avg}}$
01101	2	4	01100	12	144 (0,32)
11000	1	4	11001	25	625 (1,42)
11000	4	2	11011	27	729 (1,66)
10011	3	2	10000	16	256 (0,58)
Soma					**1754**
Média f_{avg}					**439**
Máximo					**729**

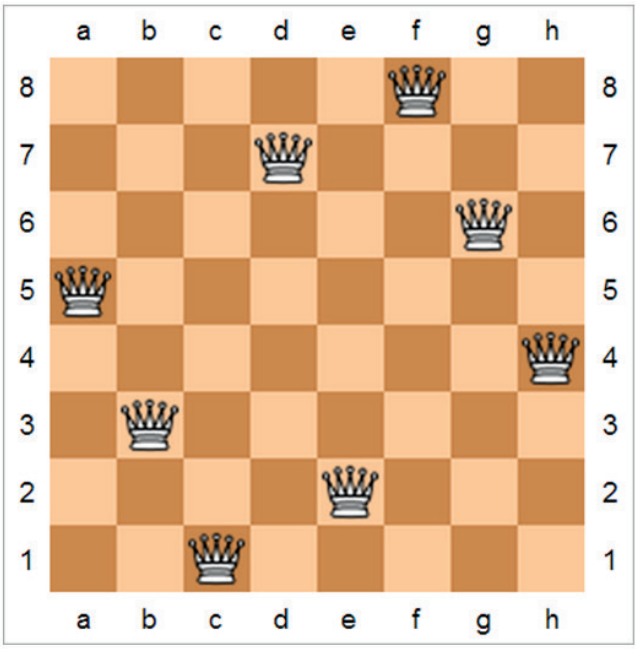
FONTE: O autor

É fácil perceber, nesta nova geração, como a seleção de indivíduos e o cruzamento destes levaram a um melhoramento da população, no sentido do objetivo desejado (observe as colunas 4 a 6 do Quadro 19). Claramente, a média de nossa função objetivo f_{avg} subiu de 239 na geração inicial para 439 na nova geração. O valor máximo de adaptação dos indivíduos subiu de 576 para 729 neste mesmo período. É fácil imaginar como a repetição deste processo pode levar rapidamente a uma aproximação do resultado ótimo para a função.

Na sequência serão mostrados alguns problemas da inteligência artificial que podem ser resolvidos com a aplicação de algoritmos genéticos:

O problema das oito rainhas: este é um problema clássico da área de inteligência artificial. Em um tabuleiro normal de xadrez (8 por 8 casas), oito rainhas devem ser distribuídas de forma que duas rainhas não possam ameaçar uma a outra (Figura 42). Este problema pode ser também generalizado para N rainhas, em tabuleiros de tamanho N x N.

FIGURA 42 - A ÚNICA SOLUÇÃO SIMÉTRICA PARA O PROBLEMA DAS 8 RAINHAS



FONTE: Adaptado de: <http://en.wikipedia.org/wiki/Eight_queens_puzzle>. Acesso em: 22 mar. 2017.

Muitas abordagens de inteligência artificial para este problema procuram resolvê-lo de maneira incremental, ou seja: começar posicionando uma rainha, e em seguida tenta-se colocar a próxima rainha em uma posição que não seja ameaçada por aquela que já está no tabuleiro, e assim por diante. A abordagem evolucionária para este problema é totalmente diferente, pelo fato de não ser incremental. As soluções candidatas para o problema são completas, ao invés de parciais.

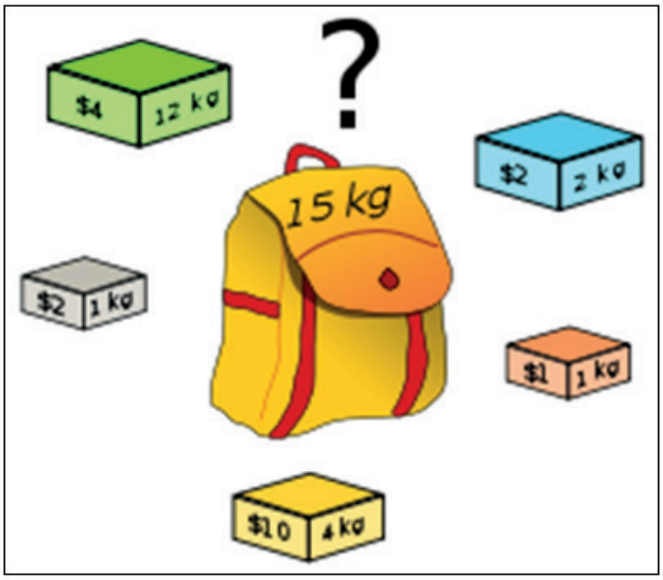
O espaço de busca para este problema é o conjunto de todas as possíveis disposições para as oito rainhas em um tabuleiro. Obviamente, a maior parte destas soluções candidatas não é aceitável, pois violam a condição básica de posicionamento das rainhas.

A função de adaptação neste caso é simplesmente o número de pares de rainhas em situação de ameaça. Quanto mais baixa esta medida, mais adaptada é a solução candidata, e um valor zero significa uma solução válida.

Este problema não é uma otimização, e sim uma busca. Resultados só serão válidos quando o número de pares de rainhas for igual a 0. Desta forma, quando se utilizam algoritmos genéticos para resolver este tipo de problema, o critério para interromper a busca é a descoberta de uma solução válida. Até que esta não seja encontrada, o algoritmo continuará a criar novas gerações, selecionar os indivíduos mais aptos e repetir o processo.

O problema da mochila: outro clássico da inteligência artificial, o problema da mochila é uma generalização para inúmeros problemas industriais, na área de otimização combinatória. Podemos descrevê-lo da seguinte forma: você precisa encher uma mochila com objetos de diferentes pesos e valores. O objetivo é preencher a mochila com o maior valor total possível, sem ultrapassar o peso máximo que a mochila comporta (Figura 43).

FIGURA 43 - PROBLEMA DA MOCHILA



FONTE: Disponível em: <http://upload.wikimedia.org/wikipedia/commons/thumb/f/fd/Knapsack.svg/250px-Knapsack.svg.png>.

Em um algoritmo genético para resolver este problema, seria natural representar as soluções candidatas para o problema como cadeias binárias de comprimento *n*, onde 1 em uma posição indica que o item é incluído na mochila, e 0 o item é excluído.

Para este caso, a função de adaptação é a soma do valor de todos os itens inclusos na mochila. Quanto mais a soma dos valores incluídos na mochila se aproximar da capacidade máxima dela sem a ultrapassar, melhor a adaptação da solução. Assim como na solução do problema das oito rainhas, a abordagem evolutiva para este problema normalmente não é incremental. Uma população inicial de soluções candidatas é gerada aleatoriamente para dar início à evolução.

Em um problema como este, nem sempre o resultado da otimização será igual ao ideal (capacidade máxima da mochila). Por isso, é normal que se defina um limite máximo de iterações para o algoritmo. Este então repete o processo de reprodução e seleção até que o limite de iterações seja atingido, após o qual a solução é considerada otimizada.

SmartRockets: no *site* <http://www.blprnt.com/smartrockets/> é possível visualizar o potencial enorme dos algoritmos genéticos para resolução de problemas, e inclusive observar o comportamento da população à medida que se alteram alguns parâmetros, como a taxa de mutação ou o tamanho da população.

Na tela há um alvo, e atingi-lo é o objetivo dos foguetes. Cada espaçonave possui cinco foguetes, que podem ser distribuídos em qualquer posição ou ângulo na espaçonave. Seu poder de propulsão também é variável, então a direção da nave depende do padrão de posicionamento dos foguetes.

FIGURA 44 - DEMONSTRAÇÃO DA UTILIZAÇÃO DE ALGORITMOS GENÉTICOS PARA CÁLCULO DA TRAJETÓRIA DE FOGUETES

FONTE: Disponível em: <http://www.blprnt.com/smartrockets/>. Acesso em: 23 mar. 2017.

A Figura 44 mostra uma situação na qual é possível visualizar as duas principais rotas encontradas pelos foguetes para atingir o alvo. Como cada foguete registra sua trajetória, é possível acompanhar o avanço da evolução.

5 ESTRATÉGIAS DE EVOLUÇÃO

Estratégias de Evolução, ou EE, é um algoritmo de otimização global, pertencente à área de Computação Evolutiva. É uma técnica similar aos Algoritmos Genéticos, Programação Genética e Programação Evolutiva. Assim como em suas técnicas "irmãs", as EE são inspiradas pela evolução através da seleção natural.

O que difere estas técnicas de outras mencionadas é que as EE são inspiradas pelo processo de evolução em nível de espécie, ou macroevolução. Neste nível, o foco é mantido no fenótipo, na hereditariedade e na variação, enquanto as outras técnicas se concentram nos mecanismos genéticos da evolução (genoma, cromossomos, genes e alelos) (BROWNLEE, 2011).

O principal objetivo das EE é maximizar a adequação de uma coleção de soluções candidatas no contexto de um problema. Em suas abordagens mais modernas, os parâmetros de mutação e seleção são adaptados entre si, de modo a controlar a variação das soluções candidatas. O Quadro 20 apresenta um resumo das propriedades principais das Estratégias de Evolução.

QUADRO 20 - PROPRIEDADES DAS ESTRATÉGIAS DE EVOLUÇÃO

Representação	Números reais
Recombinação	Discreta ou intermediária
Mutação	Perturbação Gaussiana (aleatória)
Seleção dos pais	Uniforme aleatória
Seleção dos sobreviventes	(μ, λ) ou $(\mu + \lambda)$
Especialidade	Autoadaptação do tamanho dos passos de mutação

FONTE: Adaptado de Eiben; Smith (2003)

Em 1964, os estudantes Peter Bienert, Ingo Rechenberg e Hans-Paul Schwefel se encontraram no Instituto Hermann Föttinger da Universidade Técnica de Berlim, na Alemanha. Os três desenvolviam seus estudos na área da aerodinâmica, mas também possuíam um grande interesse por cibernética. Eles tiveram a ideia de resolver o problema aparentemente intratável de aperfeiçoar um formato aerodinâmico com a ajuda de algum tipo de robô. O robô deveria realizar os experimentos necessários, repetidamente manipulando um modelo flexível posicionado em um túnel de vento (BÄCK; FOGEL; MICHALEWICZ, 2000).

De início, o experimento foi realizado manualmente, utilizando uma chapa dobrável. Em um processo repetitivo, diferentes formatos e ângulos de dobra iam sendo testados no túnel de vento, até se encontrar um formato com um mínimo de arrasto aerodinâmico. O resultado esperado era obter uma chapa plana ao final do processo iterativo, pois seria a forma com menor arrasto. O experimento não teve sucesso, pois a estratégia escolhida foi a mudança de uma variável por vez, e

a variação era realizada com uma escala fixa. O resultado positivo foi conseguido quando Ingo Rechenberg teve a ideia de utilizar entre cada iteração pequenas modificações aleatórias, que somente seriam aceitas caso tivessem efeito positivo.

Em sua primeira "versão", as EE possuíam não mais do que duas regras (BEYER; SCHWEFEL, 2002):

1. Alterar as variáveis ao mesmo tempo, leve e aleatoriamente.
2. Se o novo conjunto de variáveis não diminui a qualidade do resultado final, mantenha-as, caso contrário, retorne ao estado anterior.

A primeira regra lembra as mutações aleatórias ocorridas na natureza, enquanto a segunda modela a "sobrevivência dos mais aptos", correspondente ao princípio da seleção natural de Darwin. Neste modelo havia apenas um objeto pai, e apenas um objeto filho por geração, e por isso essa forma foi chamada de **(1 + 1) – ES**. Observe que ES é a sigla inglesa para Estratégias de Evolução, ou *Evolution Strategies*.

Como muitos outros métodos, o desempenho das Estratégias Evolutivas depende bastante do ajuste de parâmetros internos. À medida que a técnica foi amadurecendo, algumas versões modificadas acabaram surgindo.

Na **(μ + λ) – ES**, na qual mais de um indivíduo filho é criado por vez, e para manter o tamanho da população constante, os indivíduos menos aptos são descartados. Já na **(μ, λ) – ES**, a seleção é feita somente entre os indivíduos da nova geração, e os pais são descartados, não importando quão bem adaptados eles sejam. Neste tipo de notação, **μ** (a letra *mu* do alfabeto grego) corresponde ao número de soluções candidatas existentes na geração pai, enquanto **λ** (a letra grega *lambda* do alfabeto grego) é o número de soluções candidatas geradas pela geração pai.

Algumas características que são particulares às Estratégias de Evolução (BROWNLEE, 2011):

- Estratégias de evolução geralmente utilizam representações de dados específicas do problema a que são aplicadas, por exemplo, números reais para problemas de otimização contínua.

- O algoritmo é comumente configurado de forma que $1 \leq \mu \leq \lambda$, isto é, um número inicial de soluções candidatas maior do que 1, e um número ainda maior de soluções candidatas na próxima geração.

- A proporção de **μ** para **λ** regula a pressão exercida pelo algoritmo.

- Em algumas versões, pode-se configurar o número de pais que irão contribuir para cada solução candidata da próxima geração.

- Uma regra clássica para controlar o grau de mutação é a regra de 1/5: isso significa que 1/5 das mutações devem ser bem-sucedidas. Se o número for maior, a variação é aumentada, e se for menor, a variação é diminuída.

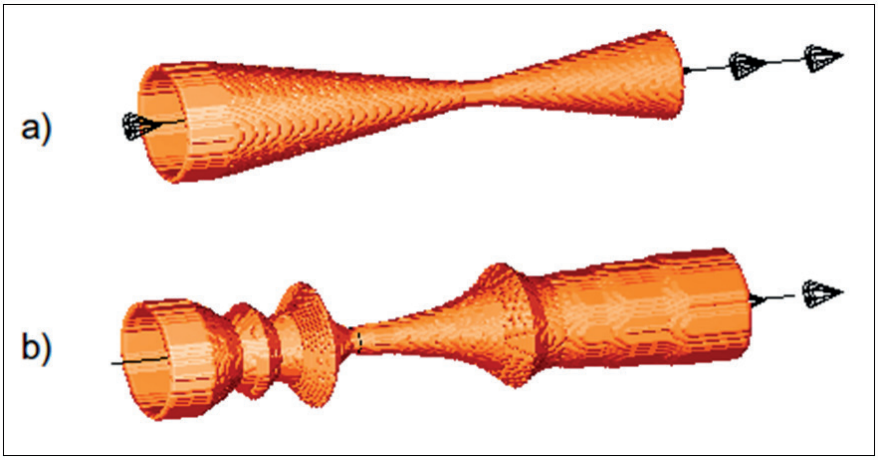
- A versão **(μ, λ) – ES** do algoritmo é recomendada para aplicação em problemas dinâmicos, onde não se tem ideia do resultado que se deseja alcançar.

- A versão **(μ + λ) – ES** é indicada para problemas de refinamento e convergência, onde se tem uma noção da solução, mas é necessário encontrar um valor ótimo.

Na sequência serão mostrados alguns problemas da inteligência artificial que podem ser resolvidos com a aplicação de estratégias de evolução:

Otimização de formato: um dos exemplos mais clássicos da utilização de EE é a otimização de um bocal de jato, em que um formato inicial é conhecido e utilizado como ponto de partida, e o uso das EE proporciona a aplicação de mutações aleatórias na forma e seleção de novas formas baseadas em seu resultado. A Figura 45 mostra um formato inicial (a), e o formato otimizado obtido após a aplicação de Estratégias de Evolução (b).

FIGURA 45 - A) FORMATO INICIAL DO BOCAL; B) FORMATO FINAL OBTIDO ATRAVÉS DA UTILIZAÇÃO DE EE



FONTE: Adaptado de: <http://ls11-www.cs.uni-dortmund.de/people/schwefel/EADemos/>. Acesso em: 23 mar. 2017.

No processo de otimização do formato do bocal, uma mutação aleatória é aplicada ao formato inicial, que é chamada solução candidata. Em seguida, a solução candidata é testada, para saber se houve evolução do formato original. Caso tenha havido uma melhoria, a solução candidata é mantida, do contrário, é descartada. Este processo é repetido até que se encontre um formato ideal, ou

um número limite de iterações tenha sido atingido. Um vídeo demonstrando o processo de evolução pode ser encontrado em <http://ls11-www.cs.uni-dortmund.de/people/schwefel/EADemos/Duese.mpg>.

Mistura de cores: em um experimento realizado por Michael Herdy (HERDY, 1996), um grupo de estudantes foi utilizado para atuar com a função de avaliação para uma EE. O objetivo da evolução era determinar as quantidades apropriadas de água e corantes vermelho, amarelo e azul, de forma que, quando misturados em um tubo de ensaio, formariam 30 ml de líquido com a mesma cor de uma marca conhecida de licor de cereja.

Para o experimento, a representação utilizada foi um vetor de quatro posições de números reais, cada componente representando a quantidade de cada ingrediente utilizado. Os estudantes foram utilizados como o fator de seleção, escolhendo manualmente a cada geração a solução candidata que mais se aproximasse do objetivo.

O experimento foi conduzido da seguinte forma (EIBEN; SMITH, 2003):

a) começando a partir de uma mistura azul de proporções conhecidas, oito novas receitas são criadas utilizando mutação;
b) cada estudante mistura os componentes em um tubo de ensaio de acordo com as especificações de uma das receitas;
c) as oito misturas são colocadas frente a uma fonte de luz, e os estudantes decidem qual delas é a que mais se aproxima da mistura desejada;
d) o processo é repetido até que os estudantes estejam satisfeitos com a coloração obtida.

Neste trabalho, a cor desejada foi encontrada em menos de 20 gerações. É um resultado bastante impressionante da robustez da abordagem da EE, ainda mais se for levada em consideração a imprecisão humana na preparação manual das misturas.

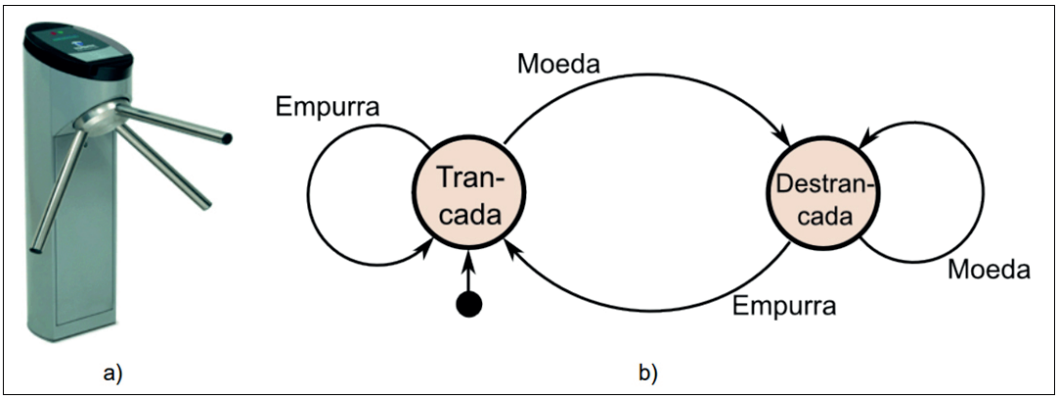
Para saber mais sobre Estratégias de Evolução, leia o relatório técnico disponível neste endereço: <http://pt.scribd.com/doc/68515615/Levantamento-bibliogra%EF%AC%81co-sobre-a-ComputacaoEvolucionaria>.

6 PROGRAMAÇÃO EVOLUTIVA E PROGRAMAÇÃO GENÉTICA

O surgimento da PE foi motivado pelo desejo de se criar uma alternativa à inteligência artificial tradicional. Em 1962, Lawrence Fogel imaginou uma forma de utilizar uma simulação da evolução para desenvolver inteligência artificial, sem baseá-la em regras, e sim em organismos com "intelecto" evolutivo. Mais tarde, em 1964, ele observou que o comportamento inteligente requer que um organismo tenha capacidade de fazer previsões corretas dentro de seu ambiente, e além disso, ser capaz de transformar estas previsões em respostas adequadas a um determinado objetivo.

A maneira encontrada por Fogel para colocar sua ideia em prática foi criar e evoluir uma população de autômatos finitos. Um autômato finito (Figura 46b) é um dispositivo capaz de operar sobre um conjunto de símbolos de entrada, e com base em um número de estados internos, produzir uma saída em resposta à sua entrada (BÄCK, FOGEL, & MICHALEWICZ, 2000). De uma maneira bastante geral, é uma forma muito rudimentar de um computador, com um poder de processamento muito limitado.

FIGURA 46 - A) CATRACA DE CONTROLE DE ACESSO; B) REPRESENTAÇÃO DA CATRATA COMO UM AUTÔMATO FINITO



FONTE: Adaptado de: <http://en.wikipedia.org/wiki/Finite-state_machine>. Acesso em: 23 mar. 2017.

Um exemplo simples de um objeto que pode ser modelado por um autômato finito é uma catraca de controle de acesso, conforme Figura 46a. O estado inicial da catraca é *trancada*. Ao se inserir uma moeda na catraca, ela passa ao estado *Destrancada*, permitindo a passagem de uma pessoa. Neste estado, se a barra for empurrada a catraca retorna ao estado inicial, exigindo agora uma outra moeda para liberar a passagem. Quando a barra é empurrada e a catraca se encontra trancada, não há troca de estado. O mesmo acontece se a catraca estiver destrancada e se inserir outra moeda. Neste exemplo, os símbolos de entrada são "Moeda" e "Empurra", pois eles é que desencadeiam a troca de estados no autômato. A saída do autômato neste caso é a liberação da passagem.

Nos experimentos de Fogel e seus colegas na década de 1960, o problema foi definido da seguinte maneira: evoluir um algoritmo que opere em uma sequência predefinida de símbolos de entrada, de maneira que sua saída possa prever o próximo símbolo de entrada. Assim, uma população de autômatos finitos é exposta ao ambiente (o conjunto de símbolos de entrada). À medida que cada símbolo de entrada é alimentado a um autômato, a saída é observada para verificar quão correta está a predição do próximo símbolo, feita por este autômato.

Na sequência, uma nova geração de autômatos é criada através de mutação aleatória dos indivíduos, e o processo é repetido: a mesma sequência de símbolos é apresentada a cada autômato, e suas saídas são analisadas quanto à capacidade de acerto dos próximos símbolos. Neste ponto, a metade dos autômatos existentes (pais ou filhos) é eliminada, mantendo-se os que tiveram melhor taxa de acerto de símbolos (BÄCK; FOGEL; MICHALEWICZ, 2000).

As principais características que podem ser encontradas em uma implementação de PE são (BROWNLEE, 2011):

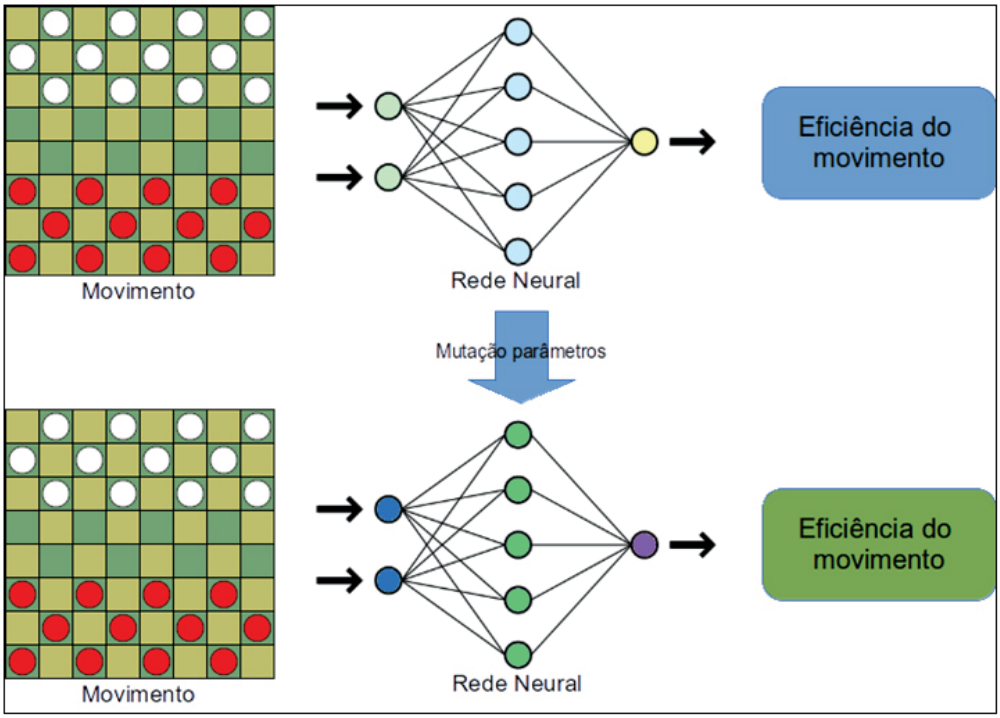
a) a representação das soluções candidatas é específica para o domínio da aplicação;
b) PE tradicionalmente usa apenas mutação para criar novos indivíduos. O cruzamento não é utilizado;
c) preocupa-se com a relação entre soluções candidatas pais e filhos e sua evolução, mas não existe tentativa de imitar os mecanismos genéticos;
d) uma das principais aplicações para PE é a otimização de funções contínuas;
e) os parâmetros da mutação usados na aplicação podem ser adaptados de acordo com as soluções candidatas.

6.1 EXEMPLOS DE UTILIZAÇÃO DE PROGRAMAÇÃO EVOLUTIVA

Evolução de jogadores de damas: uma das aplicações em que David Fogel e seus colegas experimentaram a PE foi o desenvolvimento de um programa capaz de jogar damas. O jogo de damas é um jogo de tabuleiro (8x8), onde cada jogador recebe um número fixo de peças, que se movem diagonalmente no tabuleiro. Uma peça pode capturar o oponente se este estiver adjacente a ela, e a peça puder saltar sobre o oponente em uma casa vazia. Se uma peça atingir o lado do adversário no tabuleiro, ela se torna uma "Dama" que pode se mover para frente e para trás no tabuleiro.

Para que o programa possa jogar damas, ele tem que avaliar o valor futuro dos possíveis movimentos. Isto é feito calculando o estado do tabuleiro se aquele movimento for feito. Para cada possível estado do tabuleiro, uma Rede Neural Artificial (RNA) atribui um valor, que é usado para medir a eficiência do movimento. Desta forma, a RNA define uma "estratégia" para jogar, e é esta estratégia que é evoluída através da Programação Evolutiva. Os indivíduos a evoluir são os pesos e parâmetros da RNA (EIBEN; SMITH, 2003).

FIGURA 47 - EVOLUÇÃO DA EFICIÊNCIA DOS MOVIMENTOS DO JOGO DE DAMAS



FONTE: Adaptado de: <http://upload.wikimedia.org/wikipedia/commons/b/bc/Draughts.png> e <http://upload.wikimedia.org/wikipedia/commons/3/3c/Neuralnetwork.png>. Acesso em: 23 mar. 2017.

A Figura 47 exemplifica como o fluxo é realizado. Para cada possível movimento no tabuleiro, diferentes configurações de RNA são testadas, o que gera uma medida da eficiência do movimento realizado. Periodicamente, as melhores configurações para a RNA são selecionadas, e o processo continua se repetindo até que se consiga uma otimização das configurações da RNA para o jogo de damas.

Usando esta metodologia, os pesquisadores fizeram as redes neurais competirem uma contra a outra por 840 gerações antes de selecionar a melhor estratégia evoluída, e testá-la contra jogadores humanos na internet. Os resultados foram impressionantes: de acordo com as normas do torneio, o programa foi classificado como "expert", tendo tido resultados melhores do que 99,61% de todos os outros jogadores daquele *site*. O programa que conseguiu estes resultados ficou conhecido como "Blondie24" (faça uma busca pela internet para obter mais informações).

O mais importante a se observar deste exemplo de utilização da PE é como ela permite uma grande flexibilidade de implementação. Neste caso, uma estratégia totalmente diferente (RNA) foi utilizada, e seus parâmetros foram otimizados através da Programação Evolutiva.

Diversos exemplos práticos, notas de aula e *links* para ferramentas podem ser encontradas na página da disciplina de Computação Evolucionária da PUC do Rio de Janeiro <http://www.ica.ele.puc-rio.br/disciplines/view.rails?idDiscipline=45&name=Computa%C3%A7%C3%A3o+Evolucion%C3%A1ria+(ELE2395>.

6.2 PROGRAMAÇÃO GENÉTICA

A última categoria de técnicas de Computação Evolutiva que será abordada neste material é a Programação Genética (PG). Os algoritmos da PG são "irmãos" de outros algoritmos evolucionários já mencionados, como os Algoritmos Genéticos, Estratégias de Evolução e Programação Evolutiva. A Programação Genética é o mais "jovem" dos métodos da Computação Evolutiva, tendo sido proposta no final da década de 1980 por John R. Koza.

Os algoritmos de Programação Genética são inspirados pela genética de uma população e pela evolução em nível da população, bem como fazem uso das estruturas (cromossomos, genes e alelos) e mecanismos (recombinação e mutação) mendelianos de seleção (BROWNLEE, 2011). O Quadro 21 mostra um resumo das propriedades mais importantes de um algoritmo de PG.

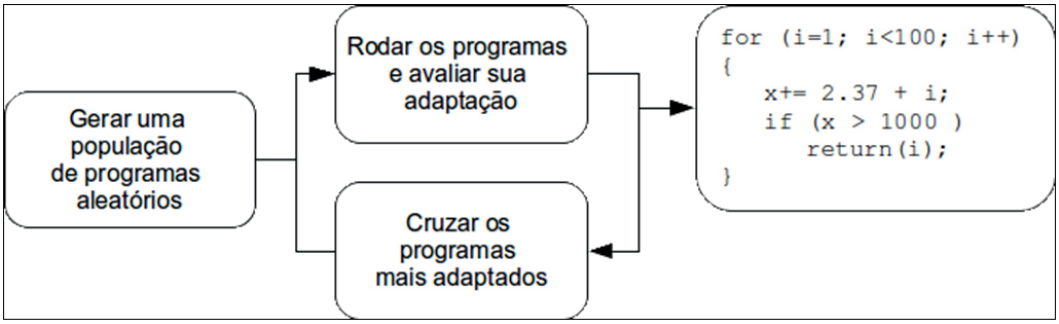
QUADRO 21 - PROPRIEDADES DOS ALGORITMOS DE PROGRAMAÇÃO GENÉTICA

Representação	Estruturas de árvore
Seleção dos pais	Adaptação proporcional
Recombinação	Troca de subárvores
Mutação	Mudança aleatória nas árvores
Seleção dos sobreviventes	Substituição da geração

FONTE: Adaptado de: Eiben e Smith (2003)

A PG pode ser considerada como uma especialização dos algoritmos genéticos, em que cada indivíduo é um programa de computador. Esta técnica pode ser utilizada como uma forma de aprendizado de máquina, para otimizar um conjunto de programas (SIVANANDAM; DEEPA, 2008). A Figura 48 mostra um diagrama do funcionamento dos algoritmos de PG.

FIGURA 48 - O CICLO PRINCIPAL DA PROGRAMAÇÃO GENÉTICA



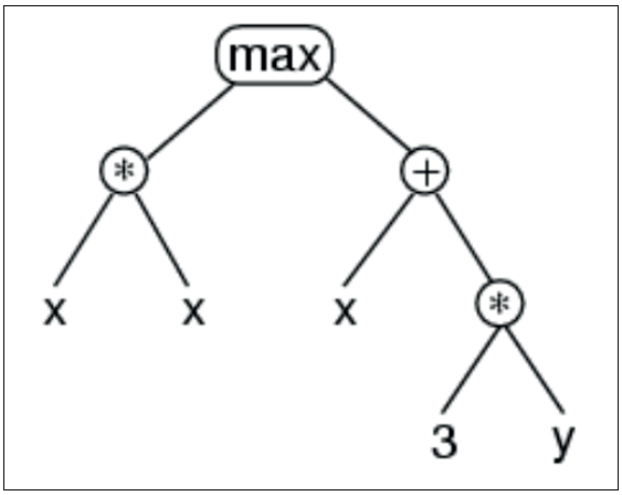
Fonte: Adaptado de: Koza e Poli (2005)

A PG é um método sistemático para fazer com que os computadores possam resolver um problema automaticamente, a partir de uma descrição do que precisa ser feito. Neste método, uma população de programas de computador é evoluída para resolver um problema. A PG iterativamente transforma uma população de programas em uma nova geração de programas, aplicando operadores genéticos de cruzamento e mutação (KOZA; POLI, 2005).

Como a programação genética evolui programas de computador, tradicionalmente estes são representados em memória como estruturas em árvore. Por isso, normalmente são utilizadas linguagens que favorecem a utilização de estruturas de árvore, por exemplo, Lisp (SIVANANDAM; DEEPA, 2008).

Assim, uma expressão simples como max (x*x,x+3*y) é representada conforme mostrado na Figura 49. A árvore aqui mostrada é composta de nós e arcos. Os nós indicam as instruções a executar, enquanto os arcos indicam os argumentos para cada instrução.

FIGURA 49 - REPRESENTAÇÃO DE PROGRAMA USADA EM PROGRAMAÇÃO GENÉTICA



FONTE: Adaptado de: Koza; Poli (2005)

Para poder entender melhor como funciona a PG, podemos imaginar um banco que concede empréstimos, e acompanha a forma como seus clientes pagam seus financiamentos. Esta informação sobre os clientes pode ser usada para criar um modelo que descreve bons e maus clientes. Este modelo pode então ser utilizado para prever o comportamento dos clientes, e assim auxiliar na avaliação de potenciais empréstimos.

A combinação dos dados pessoais dos clientes com seu histórico de pagamento pode ser usada para a criação de regras para a avaliação dos clientes. Por exemplo, o salário e o número de filhos podem ser usados como entrada, e uma avaliação do potencial cliente será a saída, como a seguir:

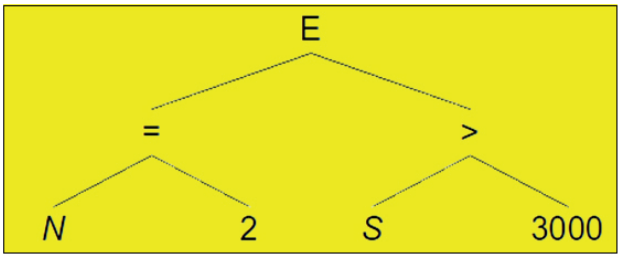
SE (*N° de Filhos* = 2) E (*Salário*> 3000) ENTÃO *Bom* SENÃO *Ruim*

Esta expressão pode ser simplificada como:

SE *fórmula* ENTÃO *Bom* SENÃO *Ruim*

Perceba que a *fórmula* é a única parte não conhecida da regra, e todos os demais elementos são fixos. Nosso objetivo é encontrar a *fórmula* ideal para uma regra de classificação dos clientes. Veja agora como esta regra pode ser representada através de uma árvore sintática, conforme pode ser observado na Figura 50.

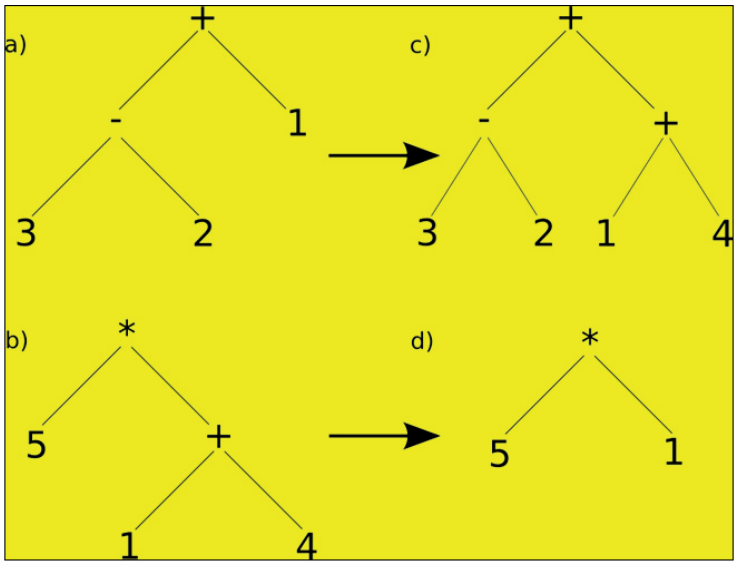
FIGURA 50 - ÁRVORE SINTÁTICA PARA REPRESENTAÇÃO DE REGRA



FONTE: Adaptado de Eiben e Smith (2003)

O operador de reprodução normalmente utilizado é o cruzamento de subárvores, em que uma subárvore inteira é trocada entre dois programas pai, conforme pode ser visualizado na Figura 51. Como em um programa genético todos os valores e funções retornam um mesmo tipo de dados, o cruzamento sempre produz novos indivíduos válidos (SIVANANDAM; DEEPA, 2008).

FIGURA 51 - CRUZAMENTO DE SUBÁRVORES DOS PAIS (A) E (B) PARA FORMAR OS FILHOS (C) E (D)



FONTE: Adaptado de: Sivanandam; Deepa (2008)

Durante a evolução de nossa solução para classificação de clientes, a população de fórmulas iniciais (Figura 50) é evoluída através de cruzamentos e mutações até que uma fórmula adequada seja encontrada (Figura 51).

Em uma solução de Programação Genética, é comum encontrar algumas das seguintes características (BROWNLEE, 2011):

a) apesar de se utilizar tradicionalmente linguagens interpretadas como Lisp, a abordagem também pode ser usada com linguagens de programação compiladas;

b) todas as funções usadas no conjunto de nós função devem retornar um resultado útil (por exemplo, a função de divisão deve retornar um valor como zero ou um, mesmo quando uma divisão por zero ocorre);

c) o algoritmo geralmente é configurado com uma alta probabilidade de cruzamento (>90%) e uma baixa probabilidade de mutação (<1%);

d) o algoritmo de PG pode ser realizado em um autômato de pilha (uma abstração mais simples do que um computador de propósito geral);

e) as funções evoluídas através da PG podem fazer uso de funções automaticamente definidas, que são subárvores, que podem ser criadas aleatoriamente, e podem evoluir juntamente com os programas que pertencem à solução desejada;

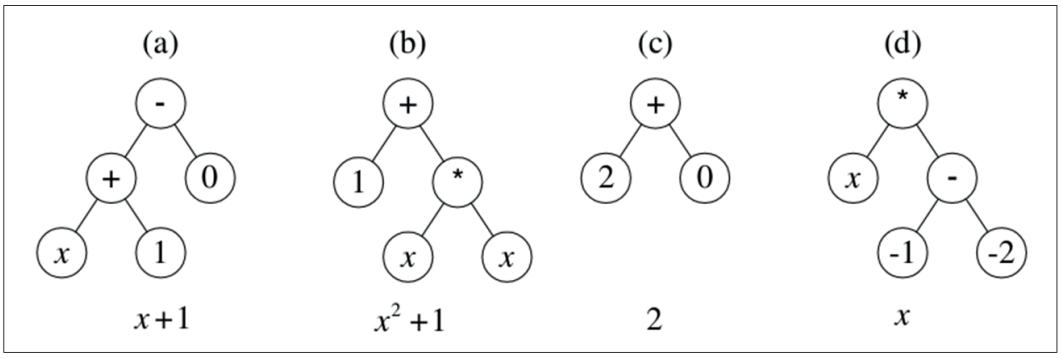
f) os operadores genéticos empregados durante a reprodução podem ser considerados programas de transformação para soluções candidatas, e podem evoluir para melhorar os resultados do processo.

6.3 EXEMPLOS DE UTILIZAÇÃO DE PROGRAMAÇÃO GENÉTICA

Regressão simbólica: para explicar o funcionamento da Programação Genética de uma forma simples. Koza e Poli (2005) propuseram a seguinte aplicação: criar um programa de computador cuja saída é igual aos valores do polinômio quadrático $x^2 + x + 1$ dentro da faixa de variação -1 até +1.

Desta forma, na implementação proposta, uma população inicial de quatro indivíduos criados aleatoriamente é descrita na Figura 52.

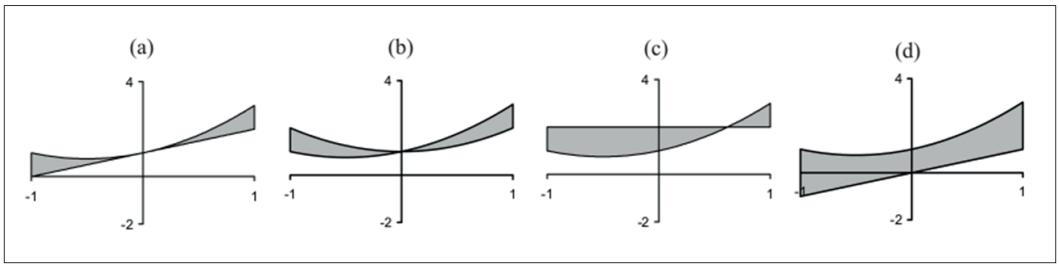
FIGURA 52 - POPULAÇÃO INICIAL DE QUATRO INDIVÍDUOS ALEATÓRIOS



FONTE: Adaptado de Koza; Poli (2005)

A adequação desses quatro indivíduos é medida através da área entre a curva gerada pelo indivíduo e a curva correspondente ao objetivo. Quanto menor a área, mais adaptado está o indivíduo. Esta comparação é mostrada na Figura 53, e através dela é fácil perceber a melhor adequação dos indivíduos (a) e (b).

FIGURA 53 - ADEQUAÇÃO DE CADA UM DOS QUATRO INDIVÍDUOS, REPRESENTADA PELA ÁREA ENTRE AS CURVAS

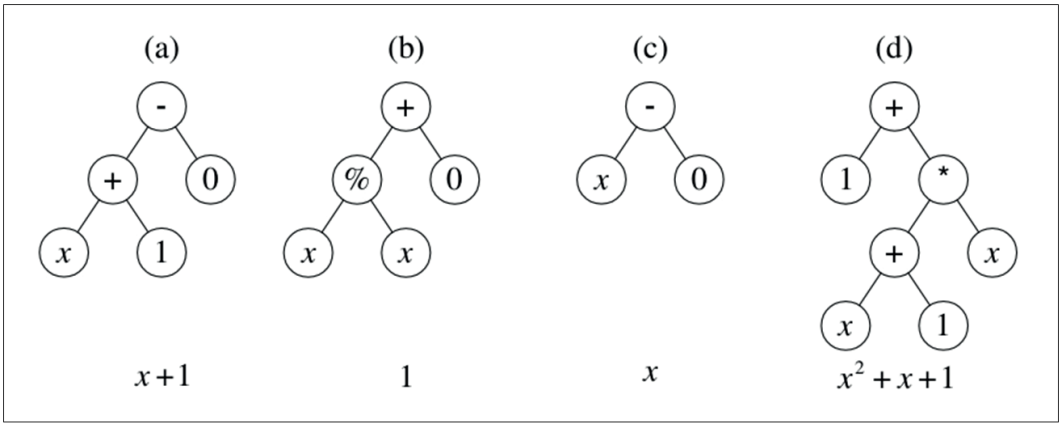


FONTE: Adaptado de Koza; Poli (2005)

Depois de avaliar a adequação de cada um dos indivíduos, a PG então seleciona os programas probabilisticamente mais adequados e é realizado o processo de reprodução, criando uma nova geração (Figura 54).

Na nova geração, o indivíduo (d) é o melhor adaptado, pois ele expressa o próprio polinômio quadrático que era o objetivo da evolução. Este indivíduo foi gerado a partir da combinação dos indivíduos (a) e (b) da primeira geração. É interessante observar que no indivíduo (d), um traço positivo (o termo quadrático x^2) veio do seu progenitor (b), enquanto outros dois traços positivos (o termo linear x e o termo constante 1) vieram de seu progenitor (a). A operação de cruzamento produziu uma solução para o problema recombinando os traços positivos de dois pais relativamente adaptados na geração anterior, em um filho de qualidade superior.

FIGURA 54 - POPULAÇÃO DA GERAÇÃO 1



FONTE: Adaptado de Koza; Poli (2005)

Assim concluímos o estudo desta segunda unidade. Aqui você aprendeu que a área da Inteligência Artificial, conhecida como Computação Evolutiva, utiliza como premissa para a resolução de problemas a reprodução do comportamento evolutivo encontrado na natureza e evidenciado por Charles Darwin. Aprendeu também que estas técnicas possuem características que as qualificam como alternativas interessantes para problemas que envolvam busca e alocação, como determinar a melhor distribuição de horários e turmas de professores em uma universidade, respeitando critérios específicos como limitações de tamanho das salas em relação ao número de alunos e restrições de horários dos professores.

Demonstramos ao longo do texto que, embora otimização seja diferente de perfeição, a computação evolutiva pode chegar a soluções altamente otimizadas, precisas e funcionais, sendo comum sua aplicação na computação de soluções para problemas que anteriormente eram insolúveis.

Mas o aprendizado não encerra aqui. Para aprofundar seus estudos na área de Algoritmos Genéticos, recomendamos a leitura do material disponível em <http://www.uniesp.edu.br/revista/revista6/pdf/13.pdf>, que faz um apanhado geral sobre o tema.

Ainda com vistas aos Algoritmos Genéticos, indicamos a leitura de <http://www.icmc.usp.br/CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_ND_75.pdf>, onde são ilustradas diferentes estratégias de implementação.

LEITURA COMPLEMENTAR

APLICAÇÕES DE ALGORITMOS GENÉTICOS

Os algoritmos genéticos são técnicas de busca que utilizam procedimentos iterativos que simulam o processo de evolução de uma população de possíveis soluções de um determinado problema. O processo de evolução é aleatório, porém, guiado por um mecanismo de seleção baseado na adaptação de estruturas individuais. A cada iteração do algoritmo (uma geração), um novo conjunto de estruturas é criado através da troca de informações (bits ou blocos) entre estruturas bem adaptadas selecionadas da geração anterior. Novas estruturas são geradas aleatoriamente com uma dada probabilidade e incluídas na população. O resultado tende a ser um aumento da adaptação de indivíduos ao meio, podendo acarretar também em um aumento global da aptidão da população a cada nova geração. Neste caso, a população evolui a cada geração, se aproximando de uma solução ótima (SOBRINHO, 2003).

Os Algoritmos Genéticos (AG) podem ser aplicados em diversas áreas de desenvolvimento. Neste texto, destacamos as seguintes: Setor de Petróleo e Gás, Musical, Telecomunicações e Saúde.

Petróleo e Gás – Inversão Sísmica: De acordo com Linden (2008), o problema da inversão sísmica, que é extremamente importante no campo da geologia, consiste na determinação da estrutura dos dados de subsolo a partir da prospecção geológica, tendo como objetivo primário obter uma seção geológica ou um modelo 3D. Este problema é extremamente suscetível à aplicação de algoritmos genéticos, pois sua função objetivo é extremamente irregular, sendo altamente não linear, possuindo muitos mínimos e máximos locais e podendo apresentar descontinuidades.

Uma abordagem bastante tradicional (MONTESINOS, 2005), usando apenas algoritmos genéticos, foi usada para a inversão de dados de anomalias gravitacionais correspondendo a contrastes de densidade fixos a priori para a obtenção da distribuição tridimensional de fontes subsuperficiais. Apesar do uso exclusivo de algoritmos genéticos, o trabalho não utiliza um modelo pronto, mas

sim escolhe cuidadosamente sua representação (baseada em cromossomos reais), seus operadores de mutação e crossover e sua função de avaliação, que embute um grande conhecimento sobre a área.

Ademais, as soluções obtidas são corrigidas usando informação conhecida a priori, de forma a evitar desvios de mínimos locais e para produzir modelos com uma transição geométrica mais suave, o que demonstra a necessidade do uso do máximo de informação sobre um problema que estamos tentando resolver. Podemos concluir que o uso de AGs torna-se bastante útil no campo de Petróleo e Gás, principalmente para o caso acima, pois o mesmo apresenta dados totalmente inconsistentes e incertos. A partir de sua implementação e com estudos detalhados, podemos obter informações valiosas para a solução desses problemas.

Telecomunicações – Segundo Blanchard (1994), existem soluções promissoras a situações reais utilizando Algoritmos Genéticos. Blanchard mostrou o caso da US West, uma companhia regional de telecomunicações do Estado do Colorado, que vem usando um sistema baseado em AGs que possibilita projetar, em duas horas, redes óticas especializadas, trabalho que levaria seis meses utilizando especialistas humanos. O sistema produz resultados ainda 10% (dez por cento) melhores que os realizados pelo homem. A companhia estimava, naquele momento, que o sistema possibilitará uma economia de 100 milhões de dólares até o final do século.

Saúde – No Hospital universitário da UFSC, em Florianópolis, os Algoritmos Genéticos foram utilizados para auxiliar na elaboração de uma escala de trabalho dos médicos plantonistas neonatalogistas da maternidade. O objetivo pretendido foi o de auxiliar na solução da escala de trabalho dos médicos, em como diminuir o esforço e o desgaste humanos para a confecção do plantão. O problema resumia-se na disponibilidade de 12 (doze) médicos e na necessidade de atendimento 24 (vinte e quatro) horas por dia, tendo-se como variáveis envolvidas o número de médicos contratados e o turno com número adaptável de horas.

O questionamento apresentava ainda todo o conjunto de restrições de trabalho, como cargas horárias, turnos de trabalho, plantões noturnos e diurnos, finais de semana e feriados, número máximo de horas de trabalho consecutivas, períodos específicos de possibilidade de trabalho, horários fixos para determinados médicos e cargas horárias variáveis entre os médicos, podendo inclusive haver mudança nas variáveis todos os meses. A representação do conhecimento se refletiu da seguinte maneira: a. cromossomo – representa a escala de plantão (Ng = 8*S + 3*F); b. gene – representa o turno de trabalho variando entre 4, 8 e 12 horas; c. alelo – representa um médico.

O grupo de pesquisa observou em um primeiro momento que a função aptidão (fitness) precisava ser refinada, pois os resultados obtidos não eram satisfatórios. Na análise seguinte, após o refinamento dos dados obteve-se grande melhoria, com 11 (onze) médicos satisfeitos e 1 (um) insatisfeito, com 20 (vinte) horas a mais, resultando num melhor aproveitamento no que diz respeito à função funcionário/horas trabalhadas.

Conclusões – como podemos observar com os textos e exemplos acima, Algoritmos Genéticos (AGs) possuem uma aplicabilidade variada, merecendo uma maior atenção com o decorrer do tempo, principalmente pela rápida evolução dos computadores, que tornarão as aplicações da técnica cada vez mais viáveis. Os AGs são apropriados para problemas de otimização complexos, que envolvem muitas variáveis e um espaço de soluções de dimensão elevada. Caso trabalhe com problemas específicos, é aconselhável a utilização de algoritmos híbridos, que misturam as técnicas dos AGs com os métodos de otimização tradicionais. Os Algoritmos Genéticos buscam soluções para problemas de otimização, de forma análoga ao processo de evolução natural. Sua resolução se dá a partir da busca em uma população inicial, que efetuando o processo de evolução da mesma, obtém uma nova população que apresenta melhores soluções para o problema em questão.

FONTE: – Adaptação livre do autor para o texto retirado de: <http://www.fsma.edu.br/si/edicao3/aplicacoes_de_alg_geneticos.pdf>. Acesso em: 23 mar. 2017.

Referências

BLANCHARD, D. It's a fuzzy world out there. WCCI'94. **Orlando: Expert Systems**. 1994. p. 179-181.

LINDEN, R. **Algoritmos genéticos**. Ed. Brasport, Brasil, 2006.

MONTESINOS, F. G; ARNOSO, J.; VIEIRA, R., 2005, **Using a genetic algorithm for 3-D inversion of gravity data in Fuerteventura (Canary Islands)**, Int J Earth Sci (Geol Rundsch) n. 94, 2005. p. 301–316.

SOBRINHO, C. **Uma análise das aplicações dos algoritmos genéticos em sistemas de acesso à informação personalizada**, 2003.

RESUMO DO TÓPICO 2

- O princípio da evolução natural, proposto por Charles Darwin, é a base por trás da ideia de Computação Evolutiva.
- Na natureza, os organismos mais adequados ao seu ambiente conseguem viver mais e ter mais filhos, desta forma, acabam sendo "selecionados" ao longo do tempo.
- O acasalamento dos seres vivos é uma importante oportunidade de recombinação genética, que pode criar filhos geneticamente superiores.
- Termos da biologia e genética que você deve conhecer: *informação genética*, *cromossomo*, *gene*, *alelo*, *conjunto de genes*, *genoma*, *genótipo*, *fenótipo*.
- As primeiras ideias de usar evolução associada à computação surgiram ainda nos anos 1940.
- Três propostas independentes de Computação Evolutiva surgiram nos anos 1960: Programação Evolutiva, Algoritmos Genéticos e Estratégias de Evolução; somente 30 anos depois surgiu uma variação chamada Programação Genética.
- Todas estas propostas são subáreas da Computação Evolutiva, e envolvem: reprodução, variação aleatória, competição e seleção de indivíduos.
- Computação Evolutiva pode ser utilizada para resolver problemas de otimização ou de busca exaustiva.
- Algoritmos Genéticos são a técnica mais popular de Computação Evolutiva.
- AGs se caracterizam pela utilização de cruzamento e pelo fato de utilizarem cadeias binárias como forma de representação dos indivíduos.
- A função de adaptação é crucial para a evolução, pois ela determina o quão bem adaptado um indivíduo está com relação à solução do problema.
- Na seleção, os indivíduos mais aptos são escolhidos com base na função de adaptação, e os demais são descartados.
- AGs podem ser utilizados em diversos problemas de otimização, como o da mochila, e também de busca, como o problema das oito rainhas.
- Estratégias de Evolução se diferenciam de outras técnicas de Computação Evolutiva, porque não se focam nos detalhes dos mecanismos da evolução, somente na macroevolução da população.

- EE utilizam em geral vetores de números reais para representar os indivíduos.
- Em EE, os parâmetros de mutação e seleção podem ser ajustados durante a evolução para melhorar os resultados.
- A seleção dos sobreviventes pode ser sobre toda a população ($(\mu + \lambda) - ES$), ou somente sobre a nova geração ($(\mu, \lambda) - ES$).
- A Programação Evolutiva se assemelha bastante às EE, mas suas principais diferenças são que: a) na PE não se utiliza cruzamento, apenas mutação, e b) a representação dos indivíduos pode variar de acordo com o problema.
- A primeira proposta de PE representava indivíduos como autômatos finitos.
- Uma das principais aplicações da PE é otimizar funções contínuas.
- Um programa para jogar damas baseado em Redes Neurais Artificiais e otimizado com Programação Evolutiva foi classificado como superior a 99,61% dos jogadores humanos em um *site* especializado.

AUTOATIVIDADE

1 Analise as seguintes afirmações a respeito da evolução dos seres vivos através da seleção natural:

I. A evolução natural ainda é uma teoria não comprovada.
II. Indivíduos mais aptos deixam uma prole maior, e com isso tendem a se sobressair em seu ambiente.
III. A recombinação genética por meio da reprodução sexuada é um fator sem muito impacto na evolução natural.
IV. Todas as informações sobre a constituição de um ser vivo estão contidas em seus alelos, no núcleo de suas células.

Agora assinale a opção correta:

(a) Todas as afirmações são verdadeiras.
(b) Apenas as afirmações I e II são verdadeiras.
(c) Apenas as afirmações I e III são verdadeiras.
(d) Apenas as afirmações II e IV são verdadeiras.

2 A computação evolutiva (CE) e os algoritmos genéticos (AG) compartilham sua fundamentação na biologia, mais especificamente na seleção natural, onde gerações sucessivas de descendentes selecionados aleatoriamente acabam gerando indivíduos mais adaptados. Acerca da CE e dos AG, avalie as afirmações a seguir:

I - Os três conceitos fundamentais destas técnicas de Inteligência Artificial são: seleção de indivíduos, competição e variação aleatória.
II - A otimização, no caso dos CE e AG, refere-se à busca por uma solução mais adequada a determinado problema.
III - A principal vantagem dos CE e AG frente às demais técnicas de inteligência artificial é a capacidade de adaptação.
IV - A reprodução é um fator essencial para a evolução.

Agora assinale a alternativa que contém somente afirmações CORRETAS:

I, II, III e IV.
I, II e IV.
III e IV.
II, III e IV.

3 Os algoritmos genéticos são a técnica mais popular na área da população evolutiva. Eles chegam a soluções otimizadas para problemas através de mutações sucessivas e aleatórias em suas propriedades, o que acaba causando a evolução propriamente dita. No que se refere a algoritmos genéticos, avalie as afirmações a seguir:

I – A representação das propriedades que sofrem mutações é feita através de uma cadeia de *bits.*
II – A cada nova geração produzida o nível de adaptação é medido, no sentido de verificar se houve otimização.
III – O critério para encerramento dos cruzamentos e das mutações é o número máximo de gerações permitidas para o algoritmo genético.
IV – A população inicial é geralmente escolhida para o cruzamento após um processo complexo de escolha com base em seu genótipo.

Agora assinale a alternativa que contém somente afirmações CORRETAS:

a) I, II, III e IV.
b) I, II e IV.
c) III e IV.
d) I e II.

UNIDADE 3

APRENDIZADO DE MÁQUINA, REDES NEURAIS ARTIFICIAIS E *TENSOR FLOW*

OBJETIVOS DE APRENDIZAGEM

Ao final desta unidade você será capaz de:

- conhecer os principais conceitos que envolvem o campo da Inteligência Artificial conhecido como aprendizado de máquina;
- especificar o funcionamento de Redes Neurais Artificiais (RNA);
- demonstrar de forma prática o funcionamento destas técnicas na resolução de problemas;
- montar um ambiente que permita a implementação de RNA para a resolução de problemas através do *TensorFlow*.

PLANO DE ESTUDOS

Esta unidade de ensino está dividida em três tópicos. No final de cada um deles, você encontrará atividades que contribuirão para a apropriação dos conteúdos.

TÓPICO 1 – APRENDIZADO DE MÁQUINA

TÓPICO 2 – REDES NEURAIS ARTIFICIAIS

TÓPICO 3 – *TENSOR FLOW*

TÓPICO 1

APRENDIZADO DE MÁQUINA

1 INTRODUÇÃO

A maneira como o cérebro humano trabalha sempre exerceu especial fascínio sobre a ciência. Somos o mais eficiente processador de informação conhecido, sendo natural que se queira reproduzir a maneira como processamos informação em máquinas que comportem essa possibilidade.

Em razão da utilização da Tecnologia da Informação (TI) em praticamente todos os setores da sociedade, cria-se diariamente um volume muito grande de informação. Num ambiente em que a velocidade das mudanças e a necessidade de adequação a estas é crescente, a possibilidade de analisar grandes volumes de informação de forma mais ágil e confiável pode representar um diferencial competitivo significativo.

Um dos grandes desafios enfrentados atualmente pelas organizações é conseguir estruturar essa grande quantidade de informações de forma que seus gestores possam utilizá-las como subsídio para a tomada de decisões estratégicas. Esse é um dos diversos campos em que atua o aprendizado de máquina. Através do texto e dos exemplos desta seção, mostraremos de que forma a IA, mais especificamente o aprendizado de máquina, vem auxiliando a atividade de mineração de dados em empresas de todo o mundo.

Abordaremos inicialmente os conceitos e algoritmos de aprendizado de máquina supervisionado, nos quais a participação humana é essencial para a obtenção de um bom resultado. Além disso, mostraremos de que forma o aprendizado não supervisionado ocorre, ilustrando aplicações práticas dessa técnica. Por fim, destacaremos uma ferramenta gratuita para que você possa implementar seus próprios testes de aprendizado de máquina.

Conto com você para vencermos este desafio!

2 APRENDIZADO DE MÁQUINA

Na maioria dos casos, quando se deseja resolver um problema com o auxílio de um computador, nós precisamos de um algoritmo. Um algoritmo é uma sequência de instruções com as quais podemos transformar a entrada na saída desejada. Em uma simples soma, a entrada são os dois valores que desejamos somar, enquanto a saída é a soma desses valores.

Em muitos casos, no entanto, não há um algoritmo para realizar esse processamento. Um exemplo bastante típico é a tarefa de separar *e-mails* legítimos de mensagens *spam*. A entrada desse problema é, normalmente, um arquivo de texto e a saída deve ser uma resposta do tipo sim/não, indicando se a mensagem é um *spam* ou não. O problema nesse caso é: como transformar a entrada em uma saída (ALPAYDIN, 2010).

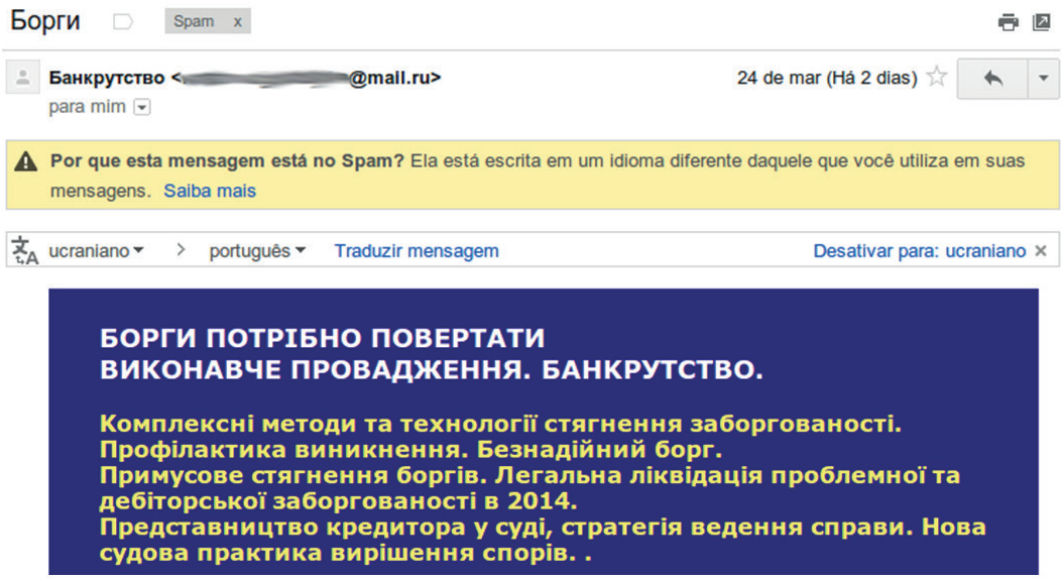
A definição de mensagens *spam* é bastante difícil de se formalizar. Em geral, são consideradas mensagens não desejadas, mas as mensagens podem ser indesejadas para uma pessoa, mas interessantes para outra. Em muitos casos, mensagens enviadas para muitos destinatários são *spam*, mas nem todas as mensagens com múltiplos destinatários são indesejadas. Estes são só dois exemplos das dificuldades enfrentadas nesse tipo de problema.

Se, por um lado, não temos uma definição clara do que caracteriza uma mensagem indesejada, temos uma quantidade mais do que suficiente de exemplos em nossa conta de *e-mail*. À medida que uma pessoa utiliza seus *e-mails*, ela acaba classificando-os como *spam*/não *spam*. As pessoas costumam guardar as mensagens que têm relevância e descartar as que não lhes interessam. Aí é que entra em cena o aprendizado de máquina: um programa pode extrair automaticamente características das mensagens descartadas e "aprender" a descartá-las automaticamente quando elas se encaixarem nessas características aprendidas.

Daumé III (2012) define, em um nível muito básico, aprendizado de máquina como a tarefa de predizer o futuro com base em fatos do passado. É exatamente isso o que acontece quando a sua conta do Gmail move dezenas de mensagens que você receberia diretamente para a caixa de *spam*. Com base no que o programa (nesse caso, o Gmail) aprendeu sobre as mensagens que você leu ou descartou no passado, ele é capaz de tomar uma decisão sobre o conteúdo das novas mensagens que chegarem em sua caixa de entrada.

Você pode aprender um pouco sobre o "aprendizado" do filtro de *spam* do Gmail se for até à caixa de *spam* e abrir uma mensagem que foi colocada lá. Na Figura 55 é mostrada uma mensagem que foi rotulada como indesejada, porque o filtro "percebeu" que normalmente aquele idioma não é utilizado nas mensagens daquela conta de *e-mail*.

FIGURA 55 - EXEMPLO DO FUNCIONAMENTO CLASSIFICADOR DE SPAM DO GMAIL



FONTE: O autor

Apesar de ser fácil entender o que é aprendizado de máquina a partir de exemplos, é mais difícil encontrar uma definição do que é aprendizado de máquina. Em parte, devido à dificuldade em definir o significado da própria palavra aprendizado. Definições do dicionário incluem "adquirir conhecimento ou entendimento, adquirir habilidade através de estudo, instrução ou experiência", e assim por diante (NILSSON, 1998). Já Grus (2015) assume que aprendizado de máquina se refere à construção de modelos, que são inferidos a partir de dados, não sendo programados diretamente no computador.

No entanto, essas definições são vagas no que diz respeito a como isso acontece, e também porque são focadas no aprendizado humano. Se pensarmos no aprendizado como uma habilidade que pode ser adquirida por animais ou máquinas, essas definições são ainda mais imprecisas. Os autores preferem apresentar o aprendizado de máquina como um ramo da IA a criar uma definição formal para o termo, mas algumas conceituações gerais podem ser encontradas, e são bastante úteis.

Enquanto Daumé III (2012) define aprendizado de máquina como uma predição do futuro com base em fatos do passado, Nilsson (1998) faz uma definição um pouco mais específica que pode nos ajudar a entender melhor a questão, afirmando que, com respeito às máquinas, pode-se dizer que elas aprendem quando há mudança em sua estrutura, em seu programa ou em seus dados, com base em suas entradas, de maneira que seu comportamento futuro seja melhorado. Domingues (2015) comenta que aprendizado de máquina é uma subárea da IA, mas cujo objetivo é ensinar os computadores a "aprenderem" de forma automática.

Em alguns casos, esse processo pode ser totalmente mecânico e não se enquadrar como "aprendizado", por exemplo, quando um registro novo é inserido em um banco de dados. Porém, quando um filtro de *spam* é melhorado após o usuário classificar manualmente algumas mensagens que ele considera indesejadas, nos sentimos inclinados a aceitar que o programa "aprendeu" com os novos dados.

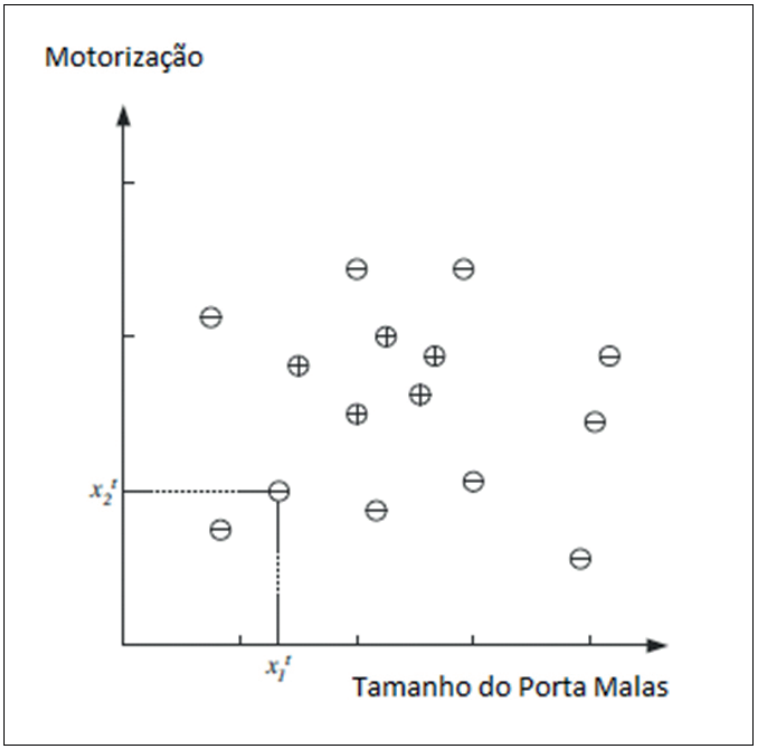
2.1 APRENDIZADO SUPERVISIONADO

Vamos discutir o aprendizado supervisionado da forma mais simples possível, em que o aprendizado de uma determinada classe ocorre de seus casos positivos e negativos. Digamos que estamos tentando definir uma classe "C" que represente um carro que sirva para toda a família. A definição dos carros adequados acontece a partir de uma pesquisa realizada com diversas famílias avaliando diversos carros. As famílias observam os carros e, de acordo com suas características, os rotulam como carros "adequados para a família" ou "não adequados para a família". Os carros considerados adequados são os exemplos positivos e os não adequados são os exemplos negativos.

O aprendizado dessa classe de carros consiste em determinar características compartilhadas por todos os exemplos positivos e por nenhum dos exemplos negativos. Uma vez que isso foi feito, podemos fazer o seguinte: dado um determinado carro que nunca foi avaliado antes, com base nas suas características e no aprendizado que tivemos, podemos determinar se ele se enquadra ou não na classe dos carros "adequados para a família" (ALPAYDIN, 2010).

Para efeito de simplificação, vamos considerar duas características dos carros como determinantes na classificação de carro "adequado para a família" ou não: o tamanho do porta-malas e a motorização. Ao considerar essas duas características (entradas), optamos por ignorar as demais, ação que deve ser cuidadosamente planejada em situações do mundo real. A Figura 56 mostra os exemplos que se enquadram ou não dentro da classe positiva.

FIGURA 56 - CONJUNTO DE TREINAMENTO PARA "CARRO PARA A FAMÍLIA"

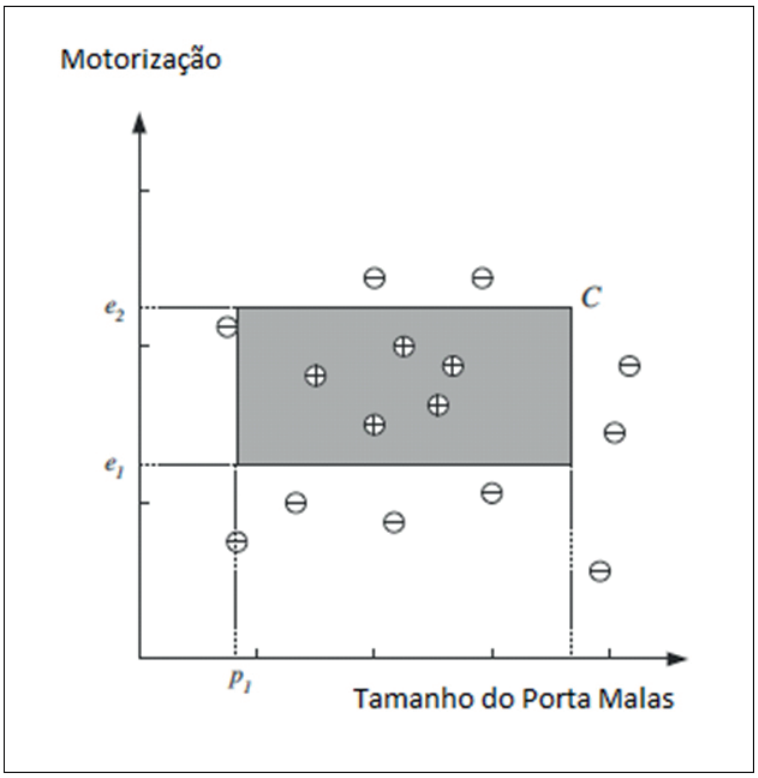


FONTE: Adaptado de Alpaydin (2010)

Nessa figura, cada ponto representa um carro da amostra, e a coordenada dele em Motorização indica a potência do motor em cilindradas, enquanto a coordenada dele em Tamanho do Porta Malas indica o volume do mesmo em litros. Todos os pontos marcados com um + são exemplos positivos dentro de nosso universo, enquanto todos os pontos marcados com um – são os exemplos negativos.

A partir daí, podemos criar nossa classe de hipótese, onde qualquer carro lá dentro se enquadre como um exemplo positivo (Figura 57).

FIGURA 57 - CLASSE DE HIPÓTESE



FONTE: Adaptado de Alpaydin, 2010

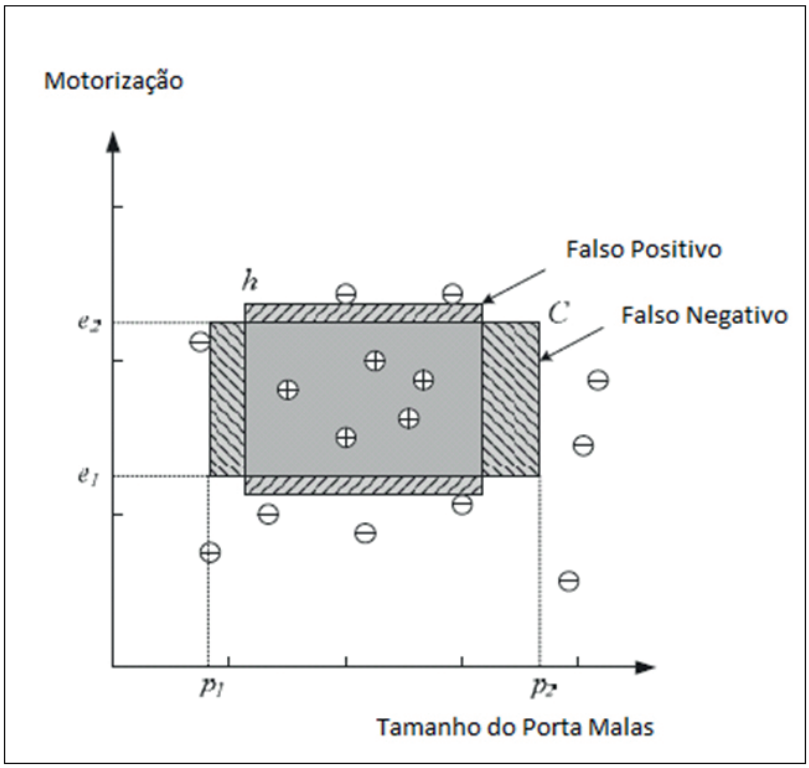
Diante disso, nossos dados de treinamento podem ser marcados em um espaço bidimensional, marcado por ex e px, onde x é a coordenada inicial ou final. Depois de analisar os dados, podemos chegar à seguinte conclusão:

$$(p1 \leq tamanho \leq p2)\boldsymbol{E}\ (e1 \leq motor \leq e2),$$

onde qualquer valor válido de px e ex que se enquadre dentro da faixa de valores válidos torna o carro um exemplo positivo.

Dispomos apenas de um pequeno subconjunto de todos os valores possíveis em nosso conjunto de treinamento, o que ocasiona um problema no aprendizado supervisionado, frequentemente chamado de generalização, ou seja, quão precisa será a definição de carros futuros que não se enquadrem em nosso conjunto de treinamento. A Figura 58 ilustra situações particulares em que ocorrem erros de generalização.

FIGURA 58 - EXEMPLOS DE ERRO DE GENERALIZAÇÃO



FONTE: Adaptado de Alpaydin (2010)

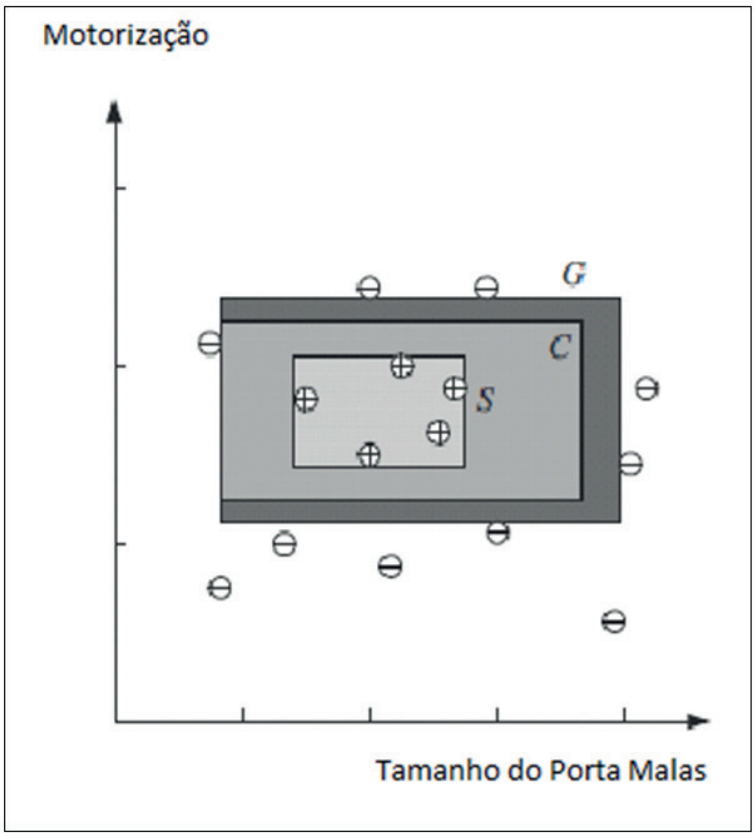
Podemos observar situações nas áreas hachuradas do gráfico, onde determinados carros não estão completamente dentro da faixa de valores que os classificariam como positivos. A classificação desses carros ocorre da seguinte forma:

- falso-positivos: exemplos em que o tamanho do porta-malas se enquadra na faixa de valores que o classificaria como positivo, mas a motorização não;

- falso-negativos: exemplos em que a motorização se enquadra na faixa de valores que o classificaria como positivo, mas o tamanho do porta-malas não;

Uma alternativa para resolver esse problema é encontrar a hipótese mais específica, isto é, o menor retângulo que inclui todos os exemplos positivos e nenhum dos negativos. Já a hipótese mais genérica consiste em encontrar o maior retângulo que inclui todos os exemplos positivos e nenhum dos negativos. Ambas as alternativas podem ser visualizadas na Figura 59.

FIGURA 59 - POSSÍVEIS SOLUÇÕES PARA A GENERALIZAÇÃO



FONTE: Adaptada de Alpaydin (2010)

Na Figura 59, S representa a hipótese mais específica e G representa a hipótese mais genérica. A partir disso, qualquer valor entre S e G é uma hipótese válida, o que minimiza o problema da generalização.

2.2 ÁRVORES DE DECISÃO

Artero (2009) afirma que a árvore de decisão é um dos métodos de aprendizado mais utilizados na prática, sendo recomendada para aplicações de mineração de dados. Uma árvore de decisão utiliza a estratégia de divisão e conquista, em que um problema mais complexo é dividido em problemas mais simples, que são resolvidos de forma recursiva.

O Quadro 22 apresenta um conjunto parcial de dados de treinamento chamado PlayTennis, que objetiva verificar se é possível disputar uma partida de tênis nas condições climáticas. Os primeiros quatro atributos correspondem às condições climáticas, enquanto a última coluna indica se é possível ou não disputar a partida através de duas classes (SIM e NÃO).

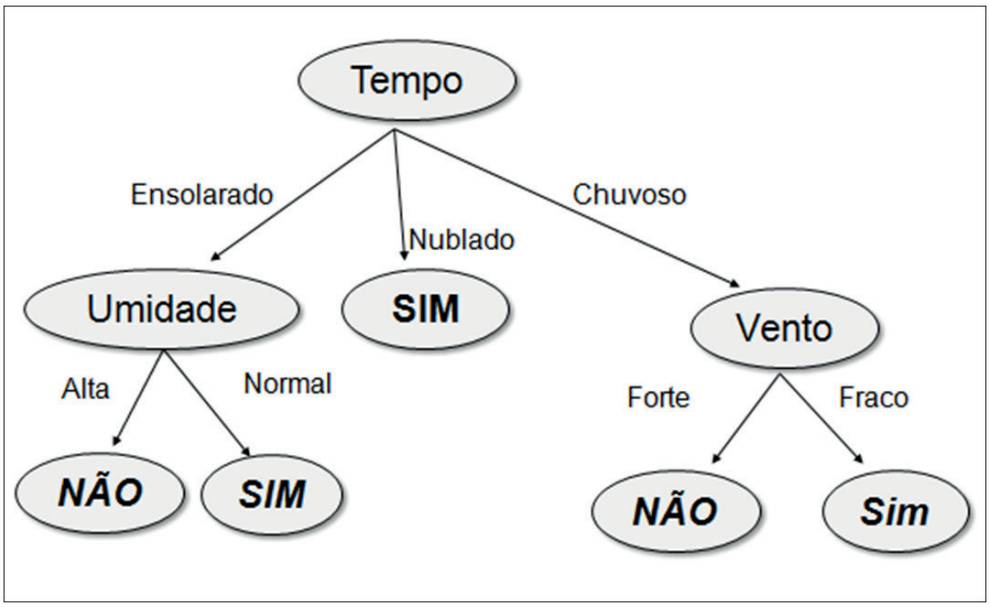
QUADRO 22 - CONJUNTO DE DADOS PLAYTENNIS

Dia	Clima	Temperatura	Umidade	Vento	PlayTennis
D1	Sol	Quente	Alta	Fraco	Não
D2	Sol	Quente	Alta	Forte	Não
D3	Nublado	Quente	Alta	Fraco	Sim
D4	Chuva	Médio	Alta	Fraco	Sim
D5	Chuva	Frio	Normal	Fraco	Sim
D6	Chuva	Frio	Normal	Forte	Não
D7	Nublado	Frio	Normal	Forte	Sim
D8	Sol	Médio	Alta	Fraco	Não

FONTE: Adaptada de Artero (2008)

Com base nesses dados de entrada tem-se a árvore de decisão, mostrada na Figura 60. Os atributos são avaliados do nó raiz para os inferiores, o que nos leva a concluir que a maior dificuldade na construção da árvore é a determinação na ordem adequada para avaliação dos atributos.

FIGURA 60 - ÁRVORE DE DECISÃO PARA PLAYTENNIS



FONTE: Adaptada de Artero (2008)

Vamos supor um exemplo em que o atributo Aspecto tem o valor Sol e a Umidade tem o valor Elevada, onde cada percurso na árvore representa uma regra de classificação. O exemplo entra na classe NÃO, ou seja, quando havia Sol e umidade elevada não se jogou tênis. Os atributos Temperatura e Vento não são considerados, pois são desnecessários para classificar esse exemplo. A partir das regras definidas na árvore, busca-se inferir resultados para dias não existentes na base de treinamento, mas que compartilham os mesmos atributos.

Outros exemplos de aplicações de árvores de decisão, bem como o detalhamento dos cálculos utilizados para a determinação da ordem dos atributos, podem ser encontrados em: <http://www.ppgia.pucpr.br/~alekoe/AM/2012/4-ArvoresdeDecisao-AM-2012.pdf>.

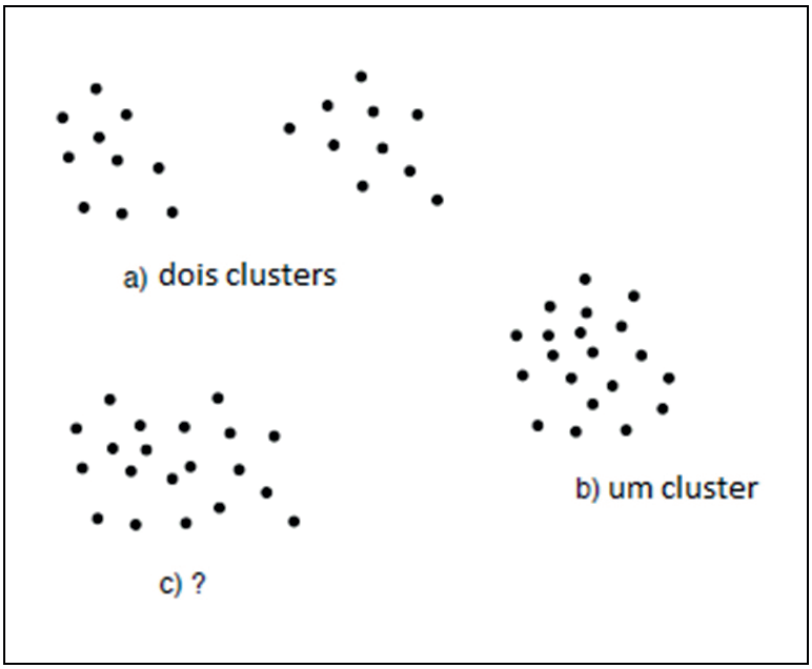
2.3 APRENDIZADO NÃO SUPERVISIONADO

Se você tem acesso a dados de treinamento já identificados, é como se existisse um professor que dissesse a você o que fazer. Infelizmente, encontrar esse professor para qualquer massa de dados é difícil e custoso. O problema é que quando se usam dados não identificados para aprendizado de máquina, esses dados podem ter centenas de dimensões, o que dificulta a análise (ALPAYDIN, 2010).

Considere uma máquina ou organismo vivo que receba sequências de entradas de informações. Essas informações, chamadas de dados, poderiam ser uma imagem na retina, os pixels em uma câmera, uma onda sonora ou mesmo palavras em uma página na internet ou itens em uma cesta de compras. No aprendizado supervisionado, para cada sequência de entrada é dada uma saída desejada, sendo que o objetivo da máquina é aprender a produzir a saída correta para entradas novas, que não fizeram parte de seu conjunto de treinamento. Já no aprendizado não supervisionado, a máquina recebe um conjunto de entradas, mas nenhum conjunto de saídas correspondente. Nesse caso, cabe à máquina encontrar padrões de semelhanças e diferenças entre os dados e com esses padrões gerar novas saídas corretas (GHAHRAMANI, 2004).

O principal interesse do aprendizado não supervisionado é desvendar a organização dos padrões existentes nos dados através de agrupamentos (*clusters*) consistentes, permitindo a descoberta de semelhanças e diferenças entre esses padrões, derivando conclusões úteis a respeito deles. A Figura 61 ilustra a identificação de *clusters* dentro de conjuntos de dados.

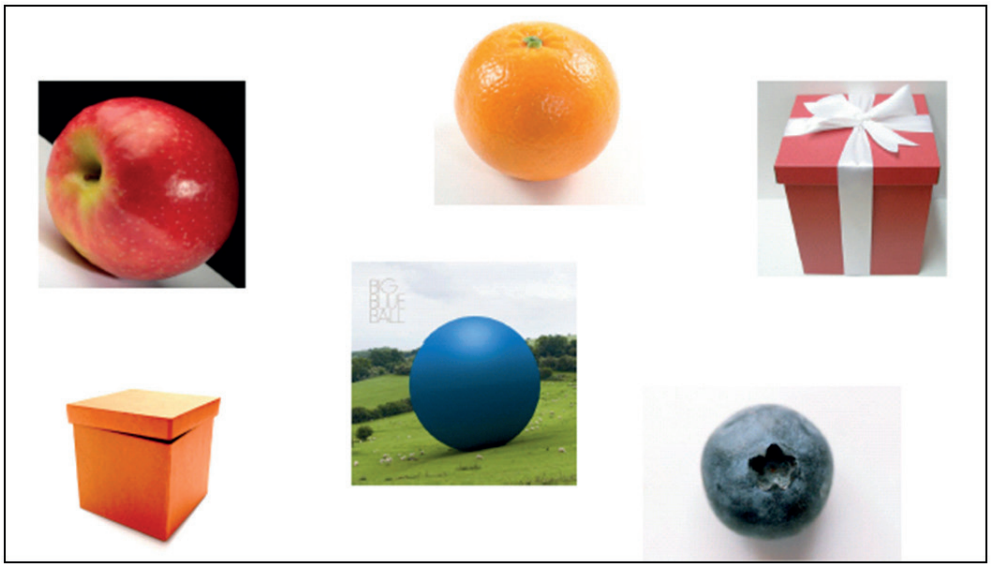
FIGURA 61 - *CLUSTERS*



FONTE: Adaptado de Nilsson (1998)

Independentemente do número de objetos que se enquadrem nesses padrões, a clusterização pode ser representativa em dados sobre os quais não se tem informação prévia. A Figura 62 traz um exemplo de dados aleatórios sobre os quais será aplicada a clusterização. Observe que os dados são simplesmente mostrados sem nenhum critério específico de agrupamento ou classificação.

FIGURA 62 - DADOS ALEATÓRIOS

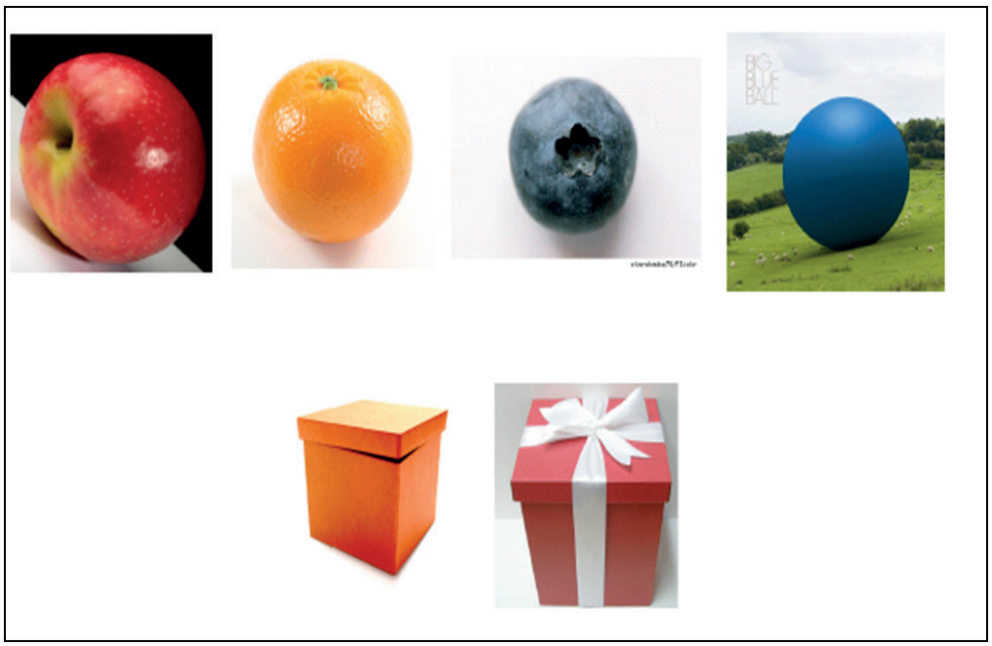


FONTE: Disponível em: <http://www.astro.caltech.edu/~george/aybi199/Donalek_Classif.pdf>. Acesso em: 18 abr. 2014.

A clusterização ocorre através de iterações, ou seja, uma sequência de passos em que se busca agrupar os dados de forma que esses agrupamentos, chamados *clusters*, façam sentido. A Figura 63 mostra a primeira iteração de uma operação de clusterização nos dados aleatórios. Nessa primeira iteração, a operação de clusterização identificou basicamente uma característica entre os objetos da amostra: o formato. Uma vez que essa característica foi identificada, procurou-se agrupar esses objetos em *clusters*, de acordo com os padrões (esférico e não esférico).

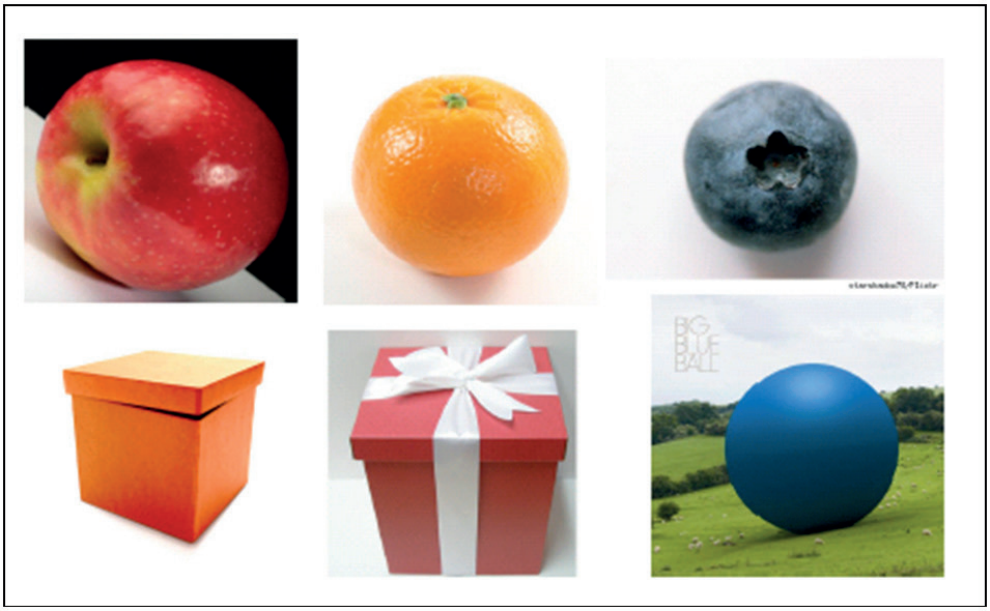
Caso buscássemos mais padrões, faríamos uma nova iteração nos dados já organizados, conforme a Figura 63.

FIGURA 63 - PRIMEIRA ITERAÇÃO DA CLUSTERIZAÇÃO



FONTE: Disponível em: <http://www.astro.caltech.edu/~george/aybi199/Donalek_Classif.pdf>. Acesso em: 18 abr. 2014.

FIGURA 64 - SEGUNDA ITERAÇÃO DA CLUSTERIZAÇÃO



FONTE: Disponível em: <http://www.astro.caltech.edu/~george/aybi199/Donalek_Classif.pdf>. Acesso em: 18 abr. 2014.

Essa segunda iteração organizou os dados em dois *clusters* distintos, de acordo com a natureza dos objetos: orgânicos e não orgânicos. Nosso exemplo é bastante simples e não permite um número elevado de iterações, mas em um conjunto de dados com um número de amostras grande, o número de iterações pode facilmente chegar a 100.000.

Outra tarefa bastante comum da clusterização é a identificação de *outliers*. Um *outlier* é um objeto que difere completamente dos demais objetos da amostra, não interferindo na obtenção de padrões e, consequentemente, na classificação. Na Figura 65 podemos ver um exemplo de *outlier* em que este foi identificado com uma moldura vermelha e separado da amostra.

É importante salientar que nesse tipo de aprendizado a máquina "aprende" os padrões sozinha, sem interferência de um ser humano. Algumas das principais aplicações do aprendizado de máquina, seja ele supervisionado ou não, podem ser encontradas no Quadro 23.

FIGURA 65 - EXEMPLO DE *OUTLIER*

FONTE: Disponível em: <http://www.astro.caltech.edu/~george/aybi199/Donalek_Classif.pdf>. Acesso em: 18 abr. 2014.

QUADRO 23 - APLICAÇÕES DE APRENDIZADO DE MÁQUINA

Área	**Aplicação**
Diagnóstico médico	Classificação de imagens mamográficas.
Segurança de redes	Identificação de tráfego potencialmente malicioso.
Genética	Classificação e identificação de genes.
Reconhecimento de padrões	Reconhecimento de voz, imagens (visão de máquina) e assinaturas.

FONTE: O autor

Você sabia que existe uma subárea da Inteligência Artificial chamada visão de máquina? O objetivo da visão de máquina (também chamada de visão computacional) é extrair informações a partir de imagens digitais, por exemplo, identificar através de uma imagem se o motorista está utilizando cinto de segurança. Uma aplicação prática pode ser encontrada em: <http://www.informedevalor.com.br/pesquisadores-criam-sistema-para-fiscalizar-uso-do-cinto-de-seguranca/>.

A grande vantagem das técnicas de aprendizado de máquina não supervisionado é que não é necessário conhecer a classe de saída dos itens de dados e, em alguns casos, nem mesmo a quantidade de classes de saída.

Artero (2009) afirma que um dos algoritmos mais conhecidos para realizar agrupamentos de dados é o algoritmo conhecido como k–médias, principalmente por causa de sua simplicidade. Para um conjunto de dados contendo m registros com n atributos, o algoritmo consiste em três passos:

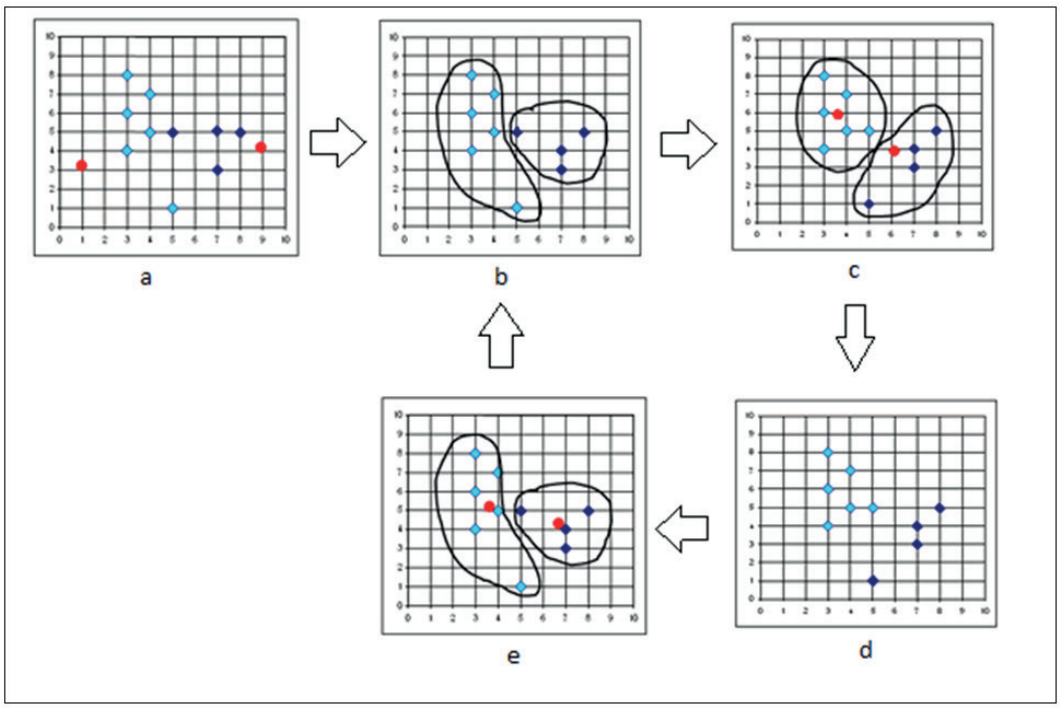
1) escolher k centros aleatórios para os prováveis centros dos grupos;

2) calcular os centros dos grupos, usando a posição média entre os marcadores mais próximos de cada centro;

3) repetir o passo 2 até que os centros dos grupos não se movimentem mais.

Na Figura 66 podemos ver o funcionamento do algoritmo. Na Figura 66a, são escolhidos aleatoriamente k objetos para fazer parte dos centros dos grupos (neste caso, k = 2). Na Figura 66b, os objetos são determinados como pertencentes ao centro que estiver mais perto. Na Figura 66c, os centros dos grupos são atualizados. Na Figura 66d os objetos são atribuídos novamente aos novos centros e, consequentemente, aos novos grupos. A partir da Figura 66e, o processo se repete até que praticamente não haja mais movimento nos objetos.

FIGURA 66 - ILUSTRAÇÃO DO ALGORITMO K-MÉDIAS



FONTE: Disponível em: <http://kdbio.inesc-id.pt/~atf/bioinformatics.ath.cx/bioinformatics.ath.cx/index651a.html?id=147>. Acesso em: 21 abr. 2014.

Os principais pontos positivos e negativos do algoritmo K-médias são mostrados no Quadro 24, a seguir:

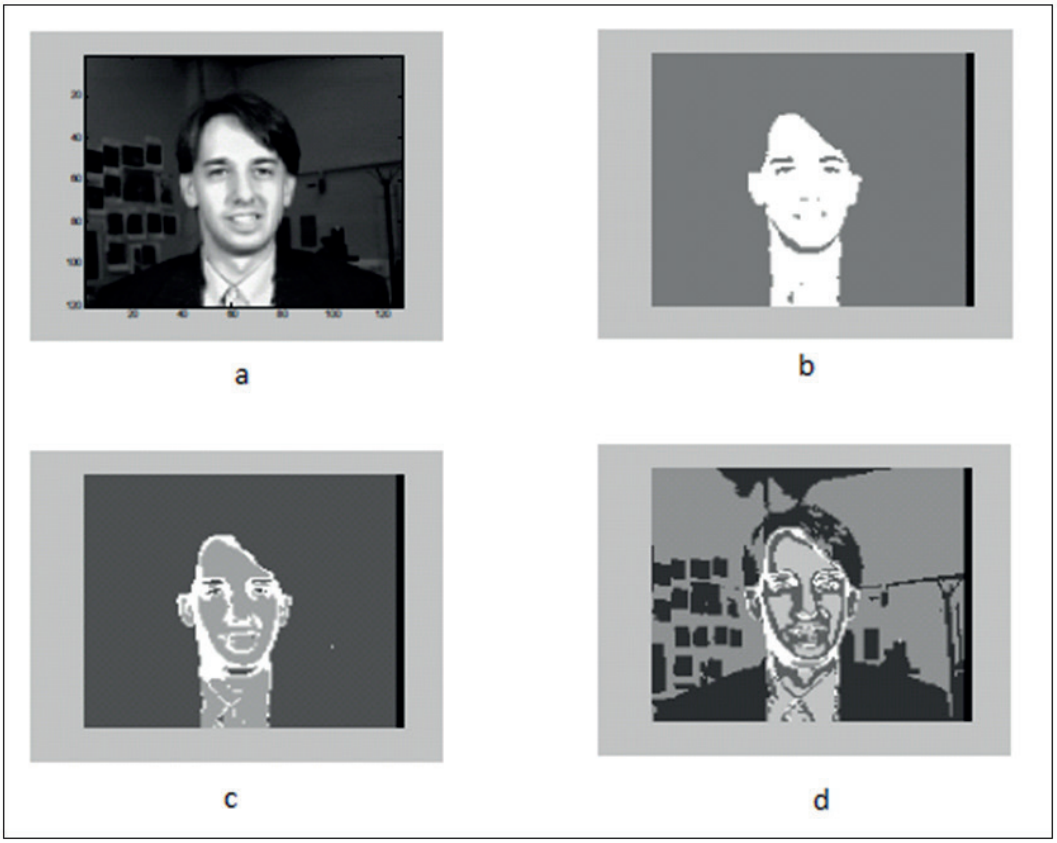
QUADRO 24 - PONTOS POSITIVOS E NEGATIVOS DO K-MÉDIAS

PONTOS POSITIVOS	PONTOS NEGATIVOS
Todos os objetos avaliados são automaticamente atribuídos a um grupo.	O número de grupos deve ser determinado antes de iniciar o algoritmo.
A localização inicial do ponto definido como centro de cada grupo pode variar e, consequentemente, determinar condições iniciais de agrupamento.	Todos os objetos avaliados devem pertencer a um grupo.

FONTE: O autor

O algoritmo k-médias pode ser utilizado para a segmentação de imagens em regiões distintas, pois funciona de forma mais automática do que uma operação de limiarização. A segmentação através do k-médias ocorre através do agrupamento dos pixels da imagem através da intensidade dos pixels (Figura 67).

FIGURA 67 - SEGMENTAÇÃO DE IMAGEM ATRAVÉS DO ALGORITMO K-MÉDIAS



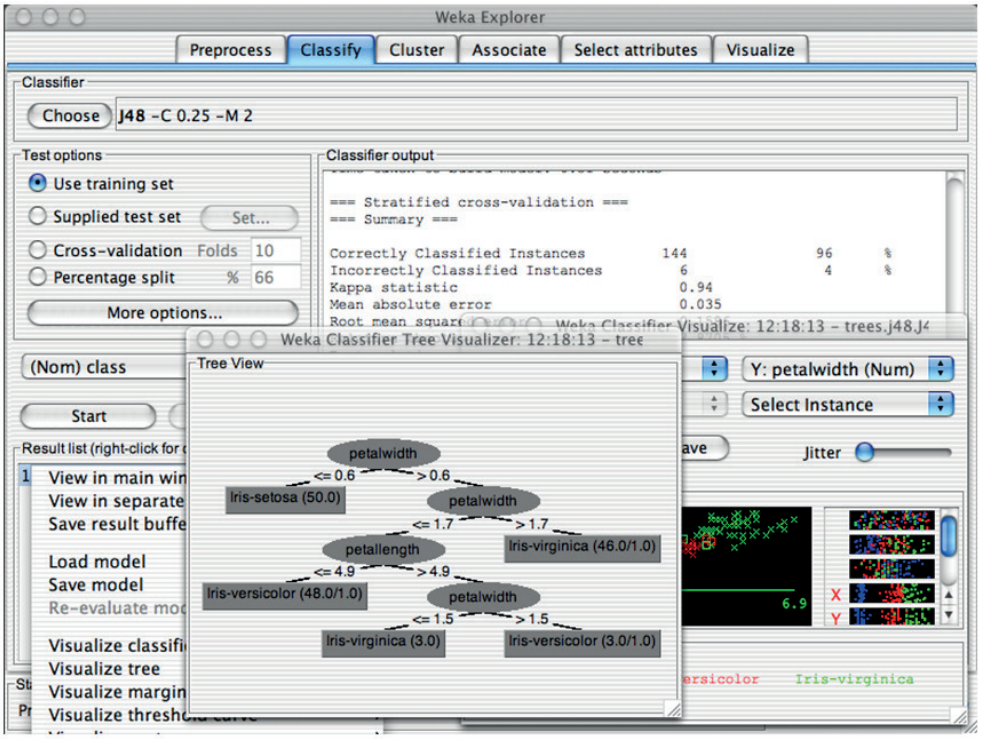
FONTE: Disponível em: <http://www.ppgia.pucpr.br/~alekoe/AM/2007/6a-AlgoritmokMeans-ApreMaq-2007.pdf>. Acesso em: 17 abr. 2014.

Na Figura 67a temos a imagem original, seguida pela 67b, em que k foi definido arbitrariamente como 2. Nas Figuras 67c e 67d, percebemos uma melhoria significativa na segmentação, na medida em que aumentamos os valores de k (3 e 5, respectivamente).

A operação chamada de limiarização ou *threshold* é comumente utilizada para segmentar imagens em regiões significativas. A técnica consiste na definição de um valor limiar que determina se os pixels da imagem valerão 1 ou 0. Mais detalhes e exemplos da limiarização podem ser obtidos em: <http://www.pessoal.utfpr.edu.br/janeczko/index_files/pdi/aula06_PDI_Limiarizacao.pdf>.

Para complementar seus estudos, sugerimos uma visita ao *site* de Waikato, da Nova Zelândia <http://www.cs.waikato.ac.nz/ml/weka/>, que é responsável pelo desenvolvimento da ferramenta Weca. Essa ferramenta disponibiliza os principais algoritmos de mineração de dados, incluindo alguns algoritmos de agrupamento, classificação e regras de associação (Figura 68).

FIGURA 68 - INTERFACE DA FERRAMENTA WEKA



FONTE: Disponível em: <http://www.cs.waikato.ac.nz/~ml/weka/gui_explorer.html>. Acesso em: 30 abr. 2014.

RESUMO DO TÓPICO 1

- As técnicas de aprendizado supervisionado e não supervisionado de máquina são especialmente úteis no processamento de grandes volumes de informação.

- Um dos grandes desafios enfrentados atualmente pelas organizações é conseguir estruturar essa grande quantidade de informações de forma que seus gestores possam utilizá-las como subsídio para a tomada de decisões estratégicas.

- Aprendizado de máquina pode ser definido como uma predição do futuro com base em fatos do passado.

- A árvore de decisão é um dos métodos de aprendizado mais utilizados na prática, sendo recomendada para aplicações de mineração de dados.

- O principal interesse do aprendizado não supervisionado é desvendar a organização dos padrões existentes nos dados através de agrupamentos (*clusters*) consistentes, permitindo a descoberta de semelhanças e diferenças entre esses padrões.

- A clusterização ocorre através de iterações, ou seja, uma sequência de passos em que se busca agrupar os dados de forma que esses agrupamentos, chamados *clusters*, façam sentido.

- A grande vantagem das técnicas de aprendizado de máquina não supervisionado é que não é necessário conhecer a classe de saída dos itens de dados e, em alguns casos, nem mesmo a quantidade de classes de saída.

- O algoritmo k-médias pode ser utilizado para a segmentação de imagens em regiões distintas, pois funciona de forma mais automática do que uma operação de limiarização.

AUTOATIVIDADE

1 No que se refere ao aprendizado de máquina, assinale a alternativa CORRETA:

a) No aprendizado supervisionado, os dados são processados através de iterações em que se procuram padrões ocultos de semelhança.
b) No aprendizado não supervisionado, os dados são agrupados em *clusters*, de acordo com suas diferenças.
c) A utilização de dados de treinamento é característica de sistemas que utilizam aprendizado de máquina não supervisionado.
d) A utilização de dados de treinamento é característica de sistemas que utilizam aprendizado de máquina supervisionado.

2 Avalie as afirmações a seguir, colocando V para as afirmações verdadeiras e F para as falsas.

( ) As árvores de decisão auxiliam os métodos de aprendizado de máquina não supervisionado.
( ) O algoritmo k-média se baseia na definição de centros e atualização da posição dos dados em relação a estes centros.
( ) Um *outlier* pode ser definido como uma informação que se diferencia das demais do grupo e, portanto, não tem influência no comportamento geral dele.
( ) Um dos objetivos do aprendizado supervisionado é permitir a classificação de informações que não fizeram parte do aprendizado de acordo com as classes de saída definidas nessa etapa.

Agora assinale a alternativa que apresenta a sequência CORRETA:

a) V, F, F, V.
b) V, F, V, V.
c) F, F, F, V.
d) F, V, V, V.

TÓPICO 2

REDES NEURAIS ARTIFICIAIS

1 INTRODUÇÃO

Nos últimos anos, as Redes Neurais Artificiais (RNA) têm sido aplicadas com sucesso nas mais diversas áreas. Sua utilização vem aumentando progressivamente, uma vez que um número cada vez maior de problemas se mostra adequado à resolução através dessa técnica e as soluções demonstradas apresentam um melhor desempenho do que os obtidos com outras técnicas computacionais. Esta seção tem por objetivo fazer uma apresentação das RNA utilizando uma abordagem que combina teoria, prática e pesquisa através dos materiais complementares.

Inicialmente é discutida a influência dos aspectos biológicos na criação e evolução dessa técnica, com destaque para as similaridades existentes entre o processamento da informação no cérebro humano e nas RNA.

Ao longo do texto demonstramos a arquitetura e as características de uma RNA do tipo Perceptron, juntamente com as fases envolvidas no projeto dessa rede. Damos especial atenção à fase de treinamento, em que ocorre efetivamente o "aprendizado", detalhando todo o processo e os cálculos envolvidos. Com a compreensão do aprendizado, você poderá projetar e treinar suas próprias RNAs.

Por fim, indicamos algumas das principais aplicações para as quais as RNAs são apropriadas e sugerimos materiais complementares para você aprofundar seus conhecimentos.

2 REDES NEURAIS ARTIFICIAIS

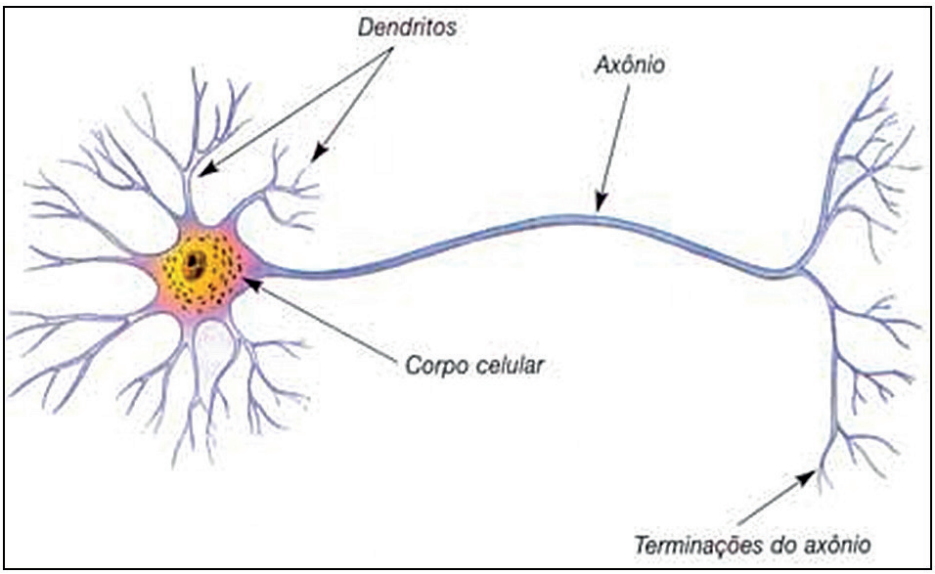
O cérebro humano possui mais de 10 bilhões de neurônios, cada qual conectado, através de sinapses, a milhares de outros neurônios. Seu funcionamento ocorre de forma completamente diferente de qualquer computador digital, sendo considerado um processador de informação altamente complexo e que atua a maior parte do tempo em paralelo (BARRETO, 2009).

Considerando o conceito numérico binário, se um cérebro humano fizesse apenas uma sinapse, seria capaz de registrar apenas dois estados (1 e 0). Se esse mesmo cérebro tivesse o dobro de sinapses, poderia realizar até quatro estados (22 = 4). Esse processo se estende a n sinapses. Com apenas dez sinapses, por exemplo, chegaríamos a impressionantes 1.024 estados (210 = 1024) (TAFNER; FILHO; XEREZ, 1996). Como existem aproximadamente 10 bilhões de neurônios (1010) e cada neurônio é capaz de criar até 10 mil sinapses (104) com neurônios adjacentes, podemos concluir de modo rudimentar que o número de estados que o cérebro humano pode encapsular é igual a:

$$10^{10} \times 10^{4} = 10^{14} = 100.000.000.000.000$$

Os neurônios são elementos de processamentos muito simples, compostos por uma soma, que é seu corpo, um axônio e vários dendritos (Figura 69). Os dendritos são responsáveis por receber os sinais de entrada dos demais neurônios. Quando a intensidade desses sinais ultrapassa determinado limiar, o neurônio é ativado, enviando um pulso elétrico ao axônio (saída) em direção às sinapses que o conectam aos demais neurônios (COPPIN, 2010).

FIGURA 69 - UM NEURÔNIO DO CÉREBRO HUMANO



FONTE: Disponível em: <http://www.sogab.com.br/anato.mia/sistemanervosojonas.htm>. Acesso em: 2 abr. 2014.

Coppin (2010) salienta que, embora cada neurônio isoladamente seja muito simples, a enorme e complexa rede de neurônios que temos em nosso cérebro é capaz de processar informações de extrema complexidade a uma velocidade impressionante. Para o processamento de situações específicas (reconhecimento de padrões, controle motor), o cérebro humano excede, de longe, a velocidade de qualquer dispositivo computacional existente hoje em dia.

Mas como o cérebro consegue isso? O cérebro tem a habilidade de estruturar seus neurônios e, consequentemente, o aprendizado através de um conjunto de regras que conhecemos mais comumente como experiência. A maior parte do desenvolvimento fisiológico dos neurônios e suas conexões (*hardware*) ocorre até os dois anos de idade, mas todos sabem que o aprendizado vai muito além dessa idade (HAYKIN, 1999).

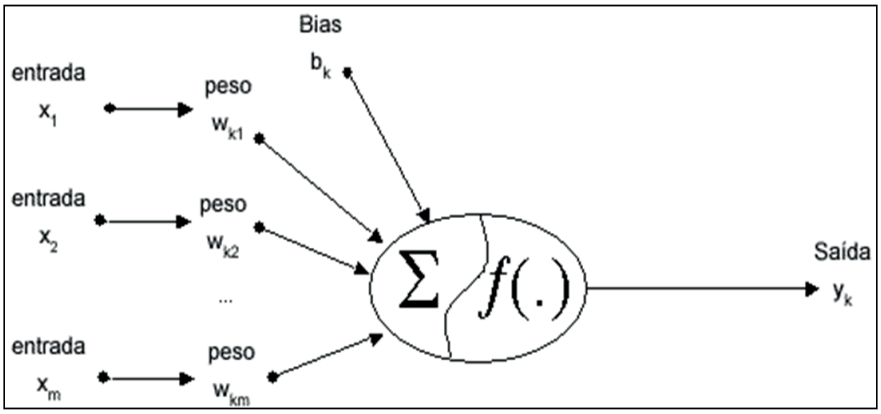
O cérebro humano tem uma propriedade chamada plasticidade, o que significa que um determinado número de neurônios pode se reorganizar em resposta a eventos que ocorram, o que ocasiona o aprendizado. Plasticidade neuronal é o nome dado a essa capacidade que os neurônios têm de formar novas conexões a cada momento. Por isso, crianças que sofreram acidentes, às vezes gravíssimos, com perda de massa encefálica, déficits motores, visuais, de fala e audição, vão se recuperando gradativamente e podem chegar à idade adulta sem sequelas, iguais às crianças que nenhum dano sofreram (SANTOS, 2014).

O cérebro utiliza uma forma de atribuição de crédito que o induz a reforçar conexões entre neurônios que levem a soluções corretas para problemas e enfraquecer conexões que levem a soluções incorretas. A intensidade de uma sinapse determina qual será sua influência nos neurônios dessa conexão, o que implica que, se uma conexão for enfraquecida, seu papel será atenuado progressivamente nas computações seguintes (COPPIN, 2010).

As redes artificiais de neurônios são modeladas de forma análoga ao cérebro humano, sendo compostas por vários neurônios artificiais. O número de neurônios e de conexões entre estes tende a ser significativamente menor em termos de números do que o cérebro humano, embora possam chegar a milhares (COPPIN, 2010).

Com o objetivo de simular o comportamento do neurônio biológico, McCulloch e Pitts (1943) propuseram em seu artigo o modelo de neurônio artificial ilustrado na Figura 70.

FIGURA 70 - NEURÔNIO ARTIFICIAL



FONTE: Disponível em: <http://www.lncc.br/~labinfo/tutorialRN/frm1_neuronio.htm>. Acesso em: 2 abr. 2014.

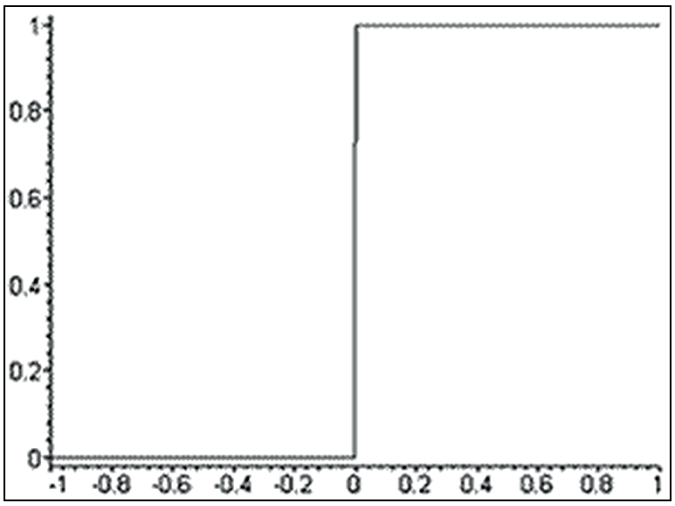
Nessa figura, o neurônio é representado através de suas entradas (x), seus pesos para cada entrada (w), seu corpo e sua saída (y). O corpo é dividido em duas partes, sendo a primeira responsável por somar o produto das entradas por seus respectivos pesos, e a segunda responsável por controlar o valor da saída (y) através de uma função de ativação ou transferência. Além das entradas regulares (x), é prevista uma entrada extra chamada de bias (b), que embora não esteja presente nos neurônios biológicos, pode ser útil em várias situações, pois aumenta os graus de liberdade e a adaptação da rede ao conhecimento a ela fornecido. A fórmula da primeira parte do processamento pode ser representada por:

$$v_j = \sum_{i=1}^{m} w_{ji} x_i + b$$, onde:

- v é o valor de ativação do neurônio k;
- w são os pesos das conexões do neurônio k;
- x é o valor de cada um dos m estímulos que chegam ao neurônio k;
- b é o valor do bias que será somado ao valor do combinador linear para compor o valor de ativação.

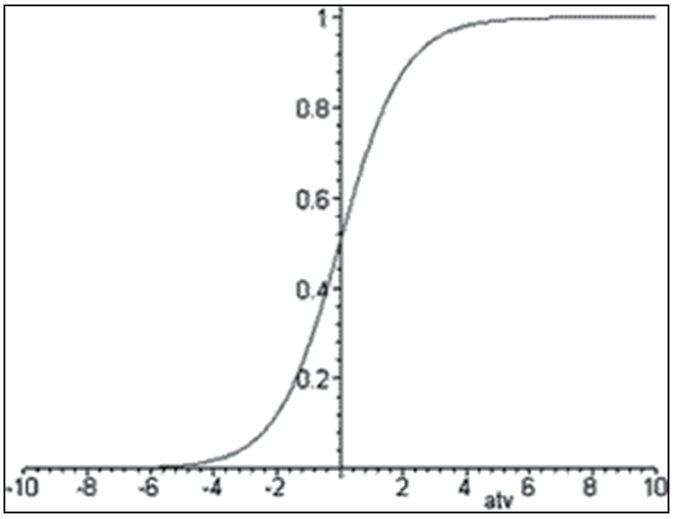
As Figuras 71, 72 e 73 ilustram, respectivamente, as funções de ativação **Degrau**, **Sigmoide** e **Linear**, que estão entre as mais utilizadas nas aplicações práticas de redes neurais artificiais. Destacamos que o eixo x de cada gráfico representa o valor de entrada para o neurônio e o eixo y representa a saída ou o nível de ativação do mesmo.

FIGURA 71 - FUNÇÃO DEGRAU OU LIMIAR



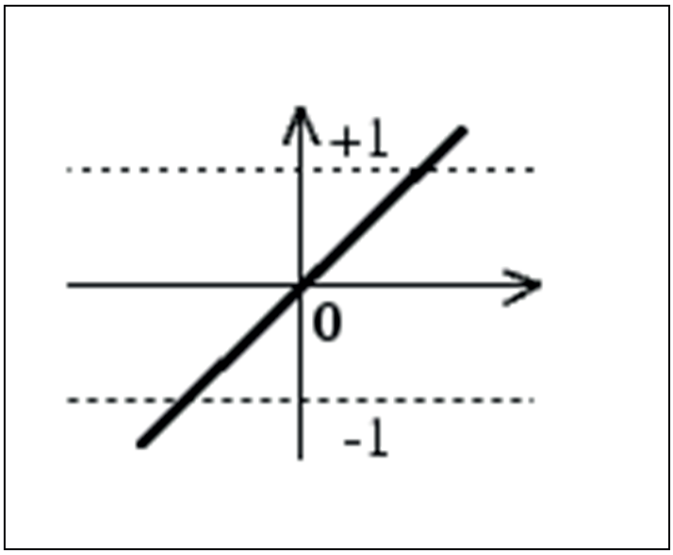
FONTE: Disponível em: <http://www.lncc.br/~labinfo/tutorialRN/frm1_neuronio.htm>. Acesso em: 2 abr. 2014.

FIGURA 72 - FUNÇÃO SIGMOIDE



FONTE: Disponível em: <http://www.lncc.br/~labinfo/tutorialRN/frm1_neuronio.htm>. Acesso em: 2 abr. 2014.

FIGURA 73 - FUNÇÃO LINEAR

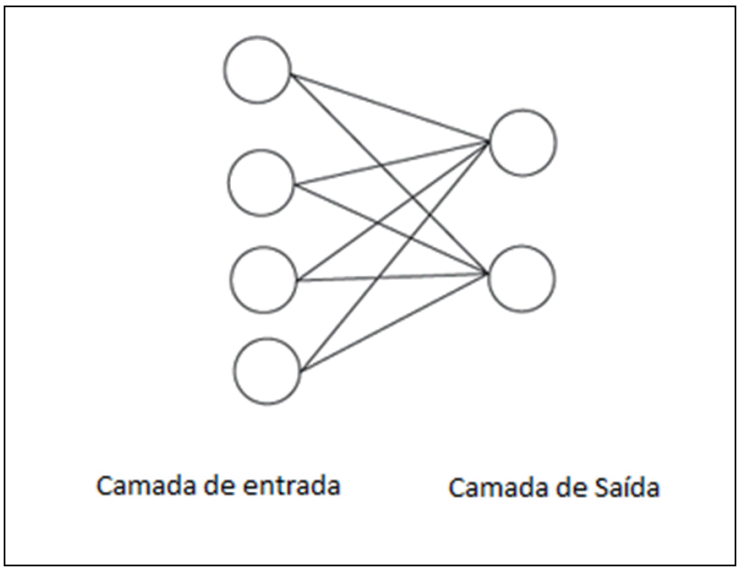


FONTE: Disponível em: <http://radio.feld.cvut.cz/matlab/toolbox/nnet/backpr52.html>. Acesso em: 21 abr. 2014.

2.1 CLASSIFICAÇÃO DAS REDES NEURAIS

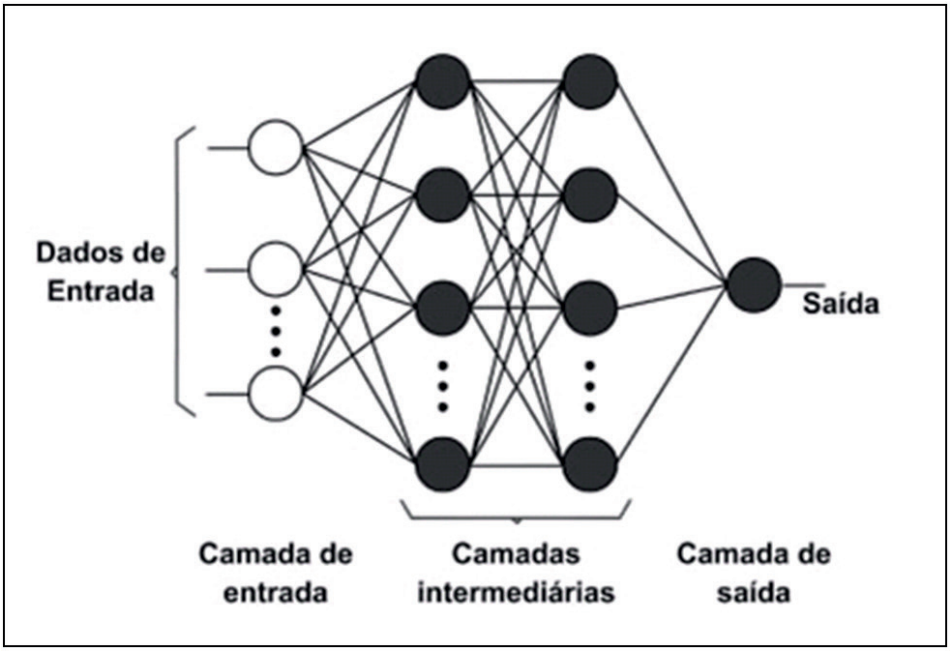
Conforme Artero (2009) e Rashid (2016), as Redes Neurais Artificiais (RNA) podem ser classificadas de acordo com diferentes critérios, por exemplo, quanto ao número de camadas de neurônios, quando é possível ter redes de duas ou múltiplas camadas. As Figuras 74 e 75 mostram duas RNAs, que possuem, respectivamente, duas e quatro camadas.

FIGURA 74 - RNA COM DUAS CAMADAS



FONTE: O autor

FIGURA 75 - RNA COM QUATRO CAMADAS

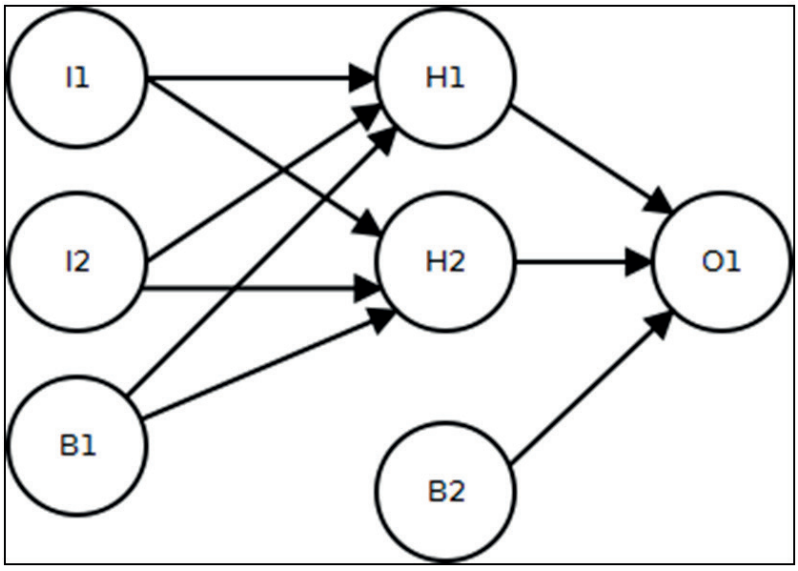


FONTE: Disponível em: <http://www.scielo.br/scielo.php?pid=S1806-11172011000100009&script=sci_arttext>. Acesso em: 1 maio 2014.

A primeira camada é a camada de entrada, em que cada nó recebe um sinal de entrada único. Esses neurônios não processam informações, apenas as repassam para as próximas camadas, que são chamadas de camadas ocultas ou intermediárias. Coppin (2010) afirma que uma rede pode ter uma ou mais camadas que contenham os neurônios que realmente fazem o trabalho. Podemos observar, na Figura 75, que cada sinal de entrada é passado a cada nó na camada oculta e assim sucessivamente, até que atinjam efetivamente a camada de saída. A diferença principal entre uma rede com duas camadas e uma rede com muitas camadas é que cada neurônio tem pesos associados a suas entradas e há muito mais pesos a ser ajustados quando um erro é cometido para um fragmento de dados de treinamento.

Uma segunda alternativa para a classificação das RNAs é a conectividade delas. Uma RNA pode ser parcialmente conectada ou totalmente conectada. Em uma rede completamente conectada, todos os neurônios de uma camada estão conectados a todos os neurônios da camada seguinte (Figura 75). Já em uma rede parcialmente conectada isso não necessariamente ocorre, conforme demonstrado na Figura 76, em que podemos observar que o neurônio representado por B1 está conectado aos neurônios da camada seguinte, H1 e H2, entretanto não apresenta conexão ao neurônio B2.

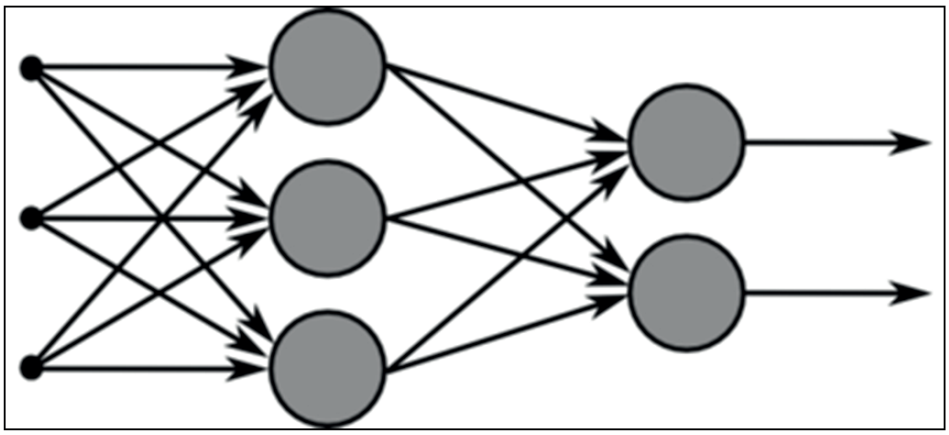
FIGURA 76 - RNA PARCIALMENTE CONECTADA



FONTE: Disponível em: <http://www.heatonresearch.com/wiki/Partial_Connections>. Acesso em: 1 maio 2014.

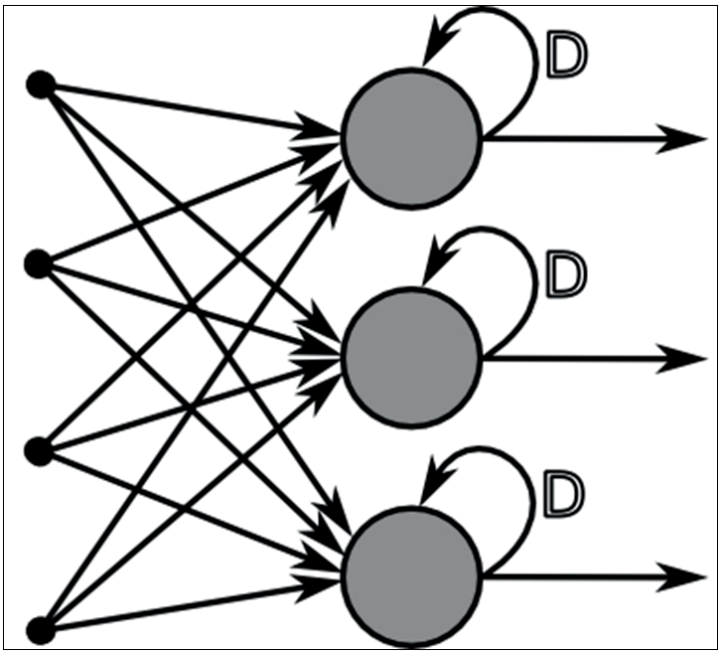
Uma terceira possibilidade de classificação é o fluxo dos sinais processados, que pode ser sempre da entrada para a saída, no caso das redes de alimentação adiante (*feed-forward*) ou também usando o conceito de retroalimentação (*feed-backward*), quando a alimentação também vem de neurônios de camadas posteriores (ARTERO, 2009). As RNAs *feed-forward* e *feed-backward* são mostradas nas Figuras 77 e 78, respectivamente.

FIGURA 77 - RNA COM ALIMENTAÇÃO ADIANTE



FONTE: Disponível em: <http://en.wikibooks.org/wiki/Artificial_Neural_Networks/Feed-Forward_Networks>. Acesso em: 2 maio 2014.

FIGURA 78 - RNA COM RETROALIMENTAÇÃO



FONTE: Disponível em: <http://en.wikibooks.org/wiki/Artificial_Neural_Networks/Print_Version>. Acesso em: 2 maio 2014.

2.2 PROJETO DE REDES NEURAIS ARTIFICIAIS

Existem várias formas de se organizar a arquitetura de uma RNA. A quantidade de camadas, a quantidade de neurônios em cada camada e a forma de conexão entre os neurônios devem ser definidas antes do treinamento e dependem do problema a ser resolvido (ARTERO, 2009).

Na camada de entrada deve-se usar um neurônio para cada atributo contínuo e, no caso de atributos categóricos, um neurônio para cada valor assumido pelo atributo. No caso do atributo Tempo, assumindo os valores *sol*, *nublado e chuva*, recomenda-se a utilização de três neurônios, conforme demonstrado no Quadro 25:

QUADRO 25 - EXEMPLO DE NEURÔNIOS DE ENTRADA

Condição do tempo	Valores	Neurônios
sol	(1, 0, 0)	{N1 = 1, N2 = 0, N3 = 0}
nublado	(0, 1, 0)	{N1 = 0, N2 = 1, N3 = 0}
chuva	(0, 0, 1)	{N1 = 0, N2 = 1, N3 = 0}

FONTE: Adaptado de Artero (2009)

Artero (2009) coloca que, para a camada de saída, quando existirem apenas duas classes, é possível usar apenas um neurônio com duas saídas, mas quando houver mais de duas classes, recomenda-se utilizar um neurônio para cada uma. Nesses casos, apenas o neurônio correspondente ao número da classe deverá apresentar saída 1, enquanto os demais devem apresentar saída 0.

Já para a camada oculta não existe regra universal, e o projeto consiste geralmente em tentativa e erro. Na verdade, quando uma rede não apresenta bom desempenho, uma prática comum é refazer todo seu projeto alterando seus parâmetros (camadas, neurônios, conexões etc.) com base em tentativa e erro.

Para demonstrar o projeto de RNAs, utilizaremos um exemplo cujo objetivo é aprender a distinguir laranjas de bananas com base nos atributos largura e comprimento (ARTERO, 2009). O Quadro 26 traz os dados que devem ser utilizados para o treinamento, com base nos atributos das laranjas e bananas.

QUADRO 26 - EXEMPLO DE PROJETO DE RNA

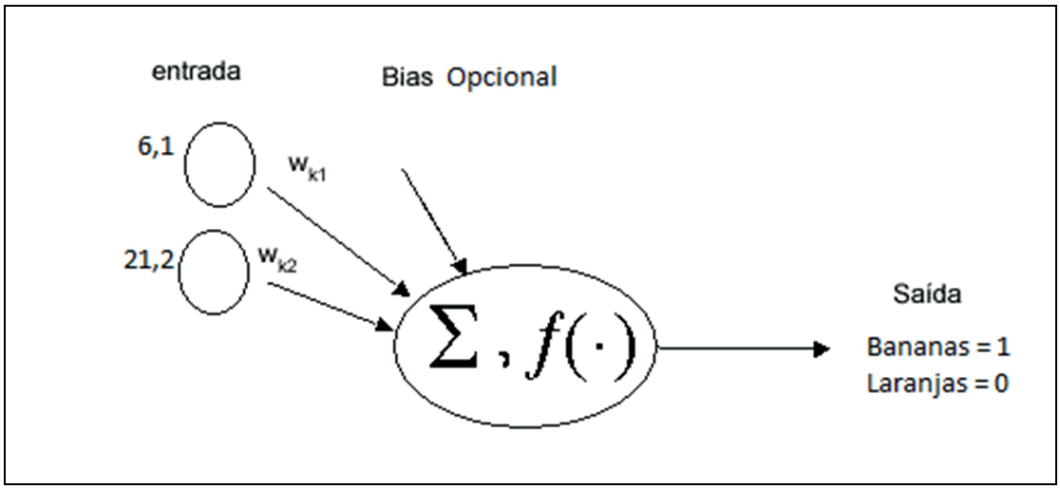
Fruta	Largura	Comprimento	Classe
Banana	6,1	21,2	1
Banana	6,3	20,9	1
Laranja	12,1	13,0	0
Laranja	11,9	12,9	0

FONTE: Artero (2009)

A Figura 79 mostra o resultado do projeto, no qual se tem uma RNA com duas camadas, sendo treinada com os atributos da primeira fruta do conjunto de dados (banana – largura = 6,1; comprimento = 21,2; classe = 1). Nesse caso o treinamento seria supervisionado, pois fornecemos os valores de entrada e a saída correspondente.

Os conceitos de aprendizado supervisionado e não supervisionado aplicados às redes neurais artificiais são semelhantes aos que estudamos na seção de aprendizado de máquina.

FIGURA 79 - EXEMPLO DE PROJETO PARA RNA



FONTE: Adaptada de Artero (2009)

2.3 TREINAMENTO

O processo de treinamento da rede é bastante delicado, pois ela não deve ser treinada 100% para cada fato a ser reconhecido, mas com o conjunto inteiro, sendo apresentadas as entradas várias vezes iterativamente. Caso uma rede seja treinada 100% para um fato de cada vez, isto afetará negativamente o conhecimento de todos os fatos previamente aprendidos (TAFNER; FILHO; XEREZ, 1996). Esse processo também é conhecido como *overfitting* (RASHID, 2016).

Quando um conjunto de entrada é apresentado à rede, esta pode acertar em uma iteração e errar na próxima, pois quando a rede erra durante o treinamento, ela tenta adaptar os pesos para produzir a resposta correta na próxima iteração. Essa compensação poderá interferir no conhecimento adquirido em iterações anteriores. Em outras palavras, o treinamento é um processo extremamente iterativo, em que o conjunto de entrada é apresentado à rede continuamente, até que acerte todos ou pelo menos um percentual razoável deles (TAFNER; FILHO; XEREZ, 1996). O Quadro 27 traz um algoritmo para o treinamento iterativo de uma RNA.

QUADRO 27 - ALGORITMO DE TREINAMENTO PARA UMA RNA

1) Apresentar um fato à rede.
2) Conferir o resultado e ajustar os pesos se necessário.
3) Se não for o final do conjunto de treinamento, voltar ao passo 1 apresentando um novo fato.
4) Se terminado o conjunto de treinamento e a rede ainda não apresentar resultados satisfatórios, reiniciar o processo até que a rede seja considerada treinada.

FONTE: Adaptado de Tafner, Filho e Xerez (1996)

Com base nessas informações, podemos concluir que insistir no treinamento de uma rede pode não ser uma solução inteligente, visto que esta pode já ter atingido seu ponto otimizado. Para exemplificar o treinamento de uma RNA, utilizaremos uma rede Perceptron que identifica se determinado atleta pertence ao esporte motovelocidade ou basquetebol, adaptado de Tafner, Filho e Xerez (1996).

Você sabia que existe um tipo particular de RNA chamado de Perceptron, em que cada entrada e saída devem ser representadas por números binários? Para saber mais sobre esse tipo de RNA, sugerimos o material disponível em: <http://wiki.icmc.usp.br/images/c/cd/SCC5809Cap4.pdf>.

Treinaremos a rede para reconhecer dois grandes jogadores de basquete (Oscar e Michael Jordan) e pilotos de motovelocidade (Valentino Rossi e Alexandre Barros), conforme mostrado no Quadro 28.

QUADRO 28 - CARACTERÍSTICAS PARA TREINAMENTO DA RNA

	Basquetebol	Motovelocidade
Oscar	x	
Michael Jordan	x	
Valentino Rossi		x
Alexandre Barros		x

FONTE: Adaptado de Tafner, Filho e Xerez (1996)

A representação de Oscar foi dada pelos valores 00, o que pode prejudicar o treinamento da rede, visto que a multiplicação de qualquer valor por 0 é igual a 0. Acrescentando o valor de Bias ao conjunto de entrada, podemos resolver esse problema e as informações ficarão da forma que aparece no Quadro 29.

QUADRO 29 - INTRODUÇÃO DO VALOR DE BIAS

Oscar – **1**00
Michael Jordan – **1**01
Valentino Rossi – **1**10
Alexandre Barros – **1**11

FONTE: Adaptado de Tafner; Filho; Xerez (1996)

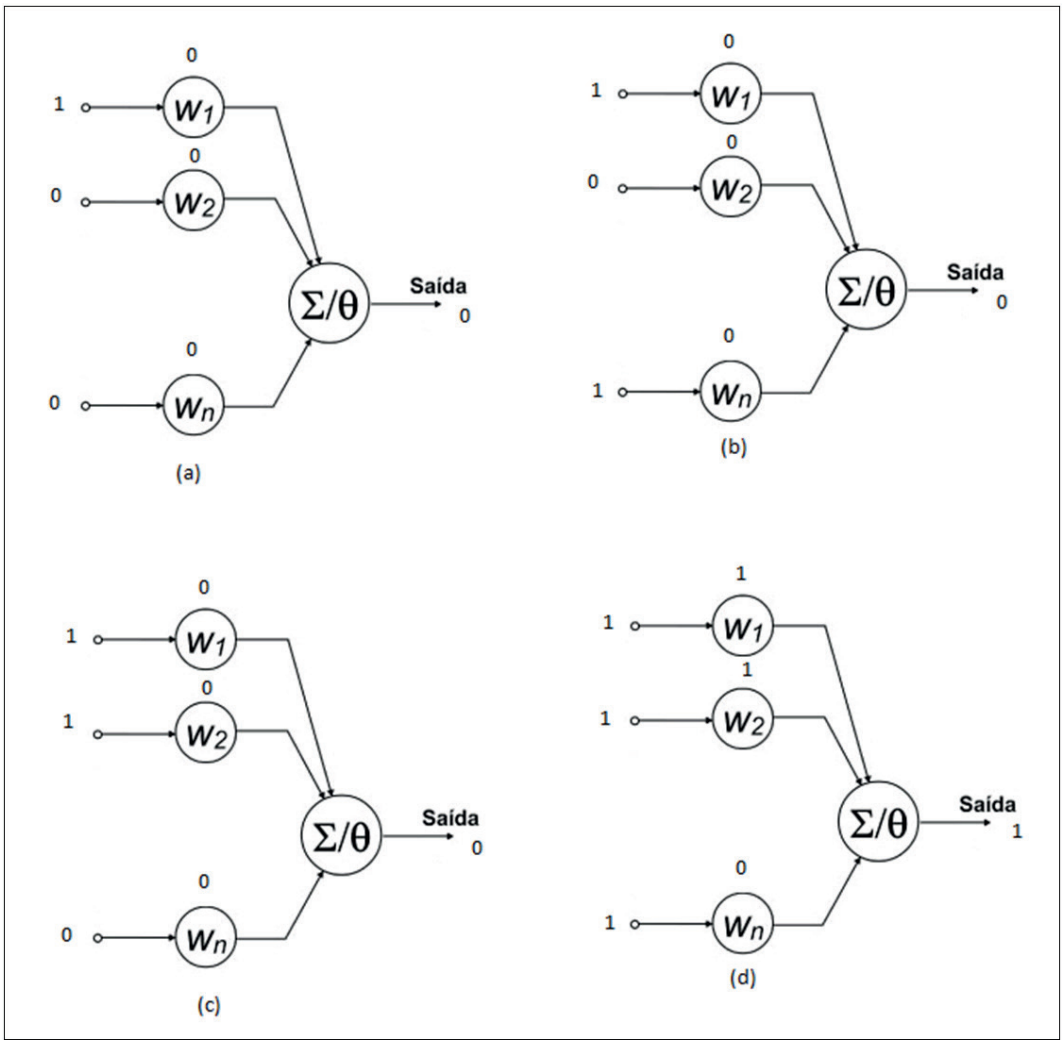
O desenho da rede neural deve ser feito com base na tabela de treinamento, na qual podemos notar que a saída é constituída apenas por duas posições, 0 ou 1. Isso viabiliza a utilização de um neurônio de saída que identifica se Oscar – 100, Michael Jordan – 101, Valentino Rossi – 110 ou Alexandre Barros – 111 são atletas do basquetebol (0) ou da motovelocidade (1).

Iniciando os pesos do neurônio com zero, temos o seguinte conjunto de pesos: w = {w0, w1, w2} = { 0, 0, 0 } e a função de ativação/transferência do neurônio igual a:

Se soma ponderada > 0, então saída = 1; se não, saída = 0

A primeira iteração dos dados de treinamento na rede é mostrada na Figura 80.

FIGURA 80 - TREINAMENTO DA REDE NEURAL



FONTE: O autor

O Quadro 30 traz os cálculos efetuados pela rede, conforme as entradas mostradas na Figura 26.

QUADRO 30 - DEMONSTRAÇÃO DO TREINAMENTO DA RNA

Calculando a saída para a entrada mostrada em 25.a (Oscar), através da soma ponderada das entradas pelos respectivos pesos, temos: Soma = 1*1+0*0+0*0 = 0 => Saída desejada = 0, logo a saída está **correta**. Agora calculando a saída para a entrada mostrada em 25.b (Michael Jordan): Soma = 1*0+0*0+1*0 = 0 => Saída desejada = 0, logo a saída está **correta**. Calculando a saída para a entrada mostrada em 25.c (Valentino Rossi): Soma = 1*0+1*0+0*0 = 0 => Saída desejada = 1, logo a saída está **incorreta**. Neste caso, adicionamos a cada peso a entrada correspondente onde: W1 = 0 + 1, W2 = 0 + 1 e Wn = 0 + 0 A alteração nos pesos é utilizada para o próximo valor de entrada. Calculando a saída para a entrada mostrada em 25.d (Alexandre Barros): Soma = 1*1+1*1+1*0 = 0 => Saída desejada = 1, logo a saída está **correta**.

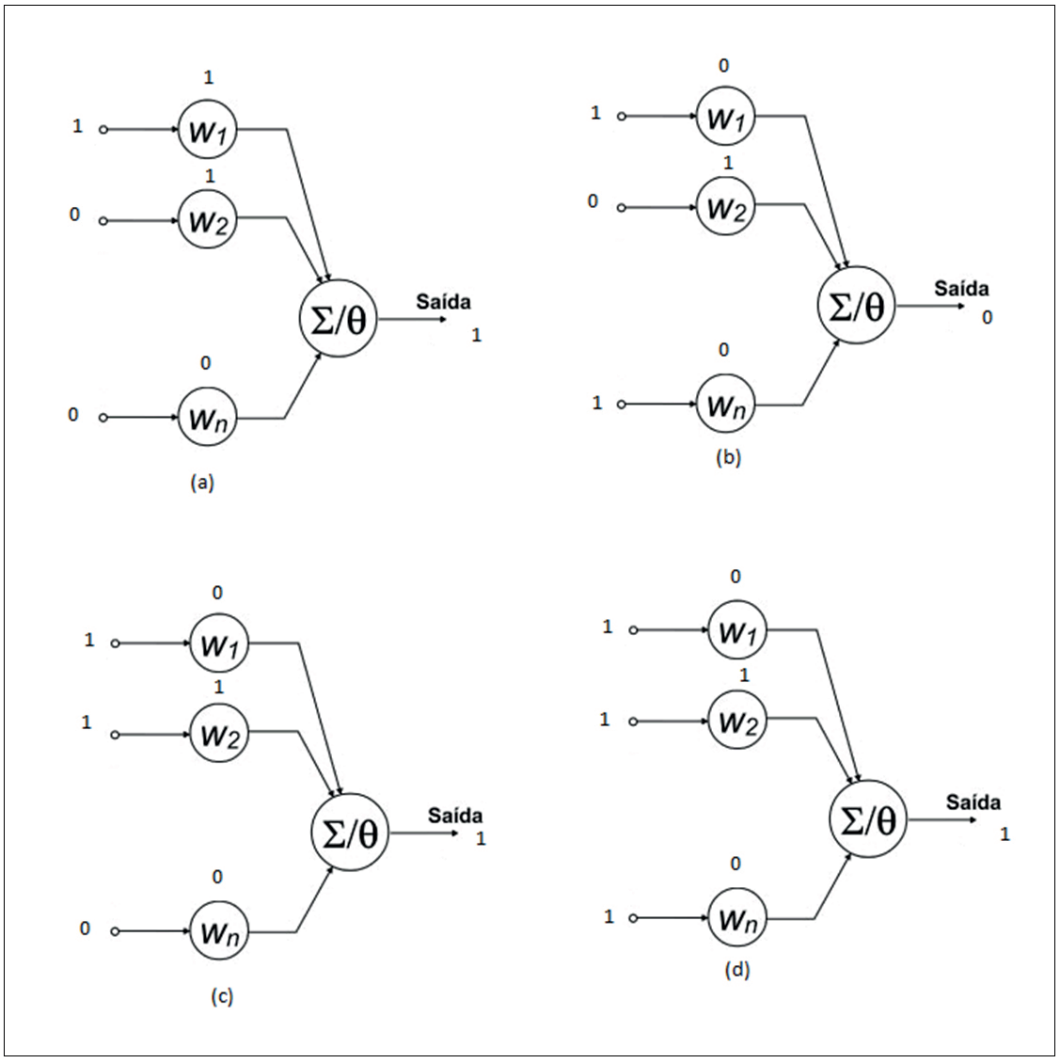
FONTE: O autor

Uma vez que todas as entradas do conjunto de treinamento foram apresentadas, deve-se fazer uma avaliação do treinamento. Essa avaliação permite verificar se o treinamento foi eficiente ou não, ou seja, se ela reconhece todos ou pelo menos um percentual significativo do conjunto apresentado. Uma forma de avaliar é fazer o teste total, no qual todo o conjunto de treinamento é apresentado e, caso uma entrada não seja reconhecida, interrompem-se os testes e ajustam-se os pesos para depois dar continuidade ao treinamento. Em alguns casos, quando o conjunto de treinamento for muito grande, é mais interessante realizar por amostragem, dividindo o conjunto em percentuais de 10% ou 20% para então verificar o sucesso (TAFNER; FILHO; XEREZ, 1996).

Outro ponto importante que deve ser levado em consideração é o grau de aceitação definido para a precisão da rede. Dependendo do conjunto de treinamento da rede, ela pode não reconhecer 100% dos casos apresentados, e sim determinado percentual. O percentual de acerto mínimo para a RNA pode ser definido em tempo de treinamento.

Em nosso caso, como o conjunto de treinamento é pequeno, procederemos com as iterações até que a rede atinja 100% de reconhecimento. Uma vez que houve reajuste de pesos na iteração anterior, devemos proceder por todas as entradas para verificar se o reconhecimento continua com o grau de acerto da iteração anterior (Figura 81 e Quadro 31).

FIGURA 81 - TREINAMENTO DA REDE NEURAL – ITERAÇÃO 2



FONTE: O autor

QUADRO 31 - DEMONSTRAÇÃO DO TREINAMENTO DA RNA

Calculando a saída para a entrada mostrada em 25.a (Oscar), através da soma ponderada das entradas pelos respectivos pesos, temos: Soma = 1*1+0*1+0*0 = 1 => Saída desejada = 0, logo a saída está **incorreta**. Neste caso, subtraímos de cada peso a entrada correspondente onde: W1 = 1- 1, W2 = 1-0 e Wn = 0 -0 A alteração nos pesos é utilizada para o próximo valor de entrada. Agora calculando a saída para a entrada mostrada em 25.b (Michael Jordan): Soma = 1*0+0*1+1*0 = 0 => Saída desejada = 0, logo a saída está **correta**. Calculando a saída para a entrada mostrada em 25.c (Valentino Rossi): Soma = 1*0+1*1+0*0 = 1 => Saída desejada = 1, logo a saída está **correta**. Calculando a saída para a entrada mostrada em 25.d (Alexandre Barros): Soma = 1*0+1*1+1*0 = 0 => Saída desejada = 1, logo a saída está **correta**.

FONTE: O autor

A rede sofreu novamente um ajuste de pesos, mas desta vez não há a necessidade de verificar todas as saídas, e sim somente a primeira.

Representando agora somente a Entrada 1 (Oscar) com os novos pesos na rede:

Soma = 1*0+1*1+0*0 = 1 => Saída desejada = 1, logo a saída está correta.

Desta vez não há a necessidade de se passar o conjunto inteiro de dados pelo treinamento, pois as demais entradas já foram submetidas ao treinamento com esses pesos e foram reconhecidas corretamente. Em nosso exemplo, atingimos o objetivo de 100% de reconhecimento e, portanto, podemos parar com o treinamento da RNA.

É importante salientar que nosso exemplo é bastante simples e tem a finalidade tão somente de demonstrar como é feito o ajuste de uma RNA buscando atingir o grau de precisão definido como objetivo. Em redes mais elaboradas, com mais camadas e mais atributos de entrada, o processo de treinamento é em geral proporcionalmente mais complexo. Em contrapartida, essas redes são capazes de identificar e classificar entradas inéditas com base no conhecimento adquirido na etapa de treinamento. Em nosso exemplo, é como se fornecêssemos o nome Marc Marquez como entrada para a rede e esta, com base no que aprendeu, pudesse classificá-lo como atleta de motovelocidade. Essa capacidade de generalização e aprendizado é o real objetivo da utilização das RNAs, e é aí que reside o grande potencial delas.

Demonstramos nesta unidade um tipo particular de RNA, cuja arquitetura é denominada de Perceptron, entretanto, existem diversos tipos de arquitetura e diferentes algoritmos para treinamento. Como exemplo, podemos citar as redes de Hopfield e os mapas auto-organizáveis de Kohonen. O artigo disponível em: <http://www.ppgia.pucpr.br/~euclidesfjr/Metodos_Inteligentes/0206/RNA-XIIERI.pdf> traz uma relação de arquiteturas e algoritmos de treinamento para RNA.

2.4 APLICAÇÕES PRÁTICAS DAS RNA

Atualmente, as RNAs têm sido amplamente utilizadas na tarefa de reconhecimento de padrões em diversas fontes de dados. Com esse processo, procura-se apresentar à rede um conjunto de dados conhecidos de modo que, através de um processo de treinamento, ela determine agrupamentos de dados com características similares, os padrões, e seja capaz de reconhecê-los em uma nova entrada apresentada à rede, de acordo com um dos padrões previamente fixados. A listagem a seguir apresenta diversas aplicações práticas que utilizam RNAs para melhorar seu desempenho:

- reconhecimento de voz;
- reconhecimento de grafia (assinaturas e caligrafia);
- reconhecimento facial através de imagens;
- economia e finanças: a utilização das RNAs nessa área vem recebendo bastante atenção nos últimos anos, pelo fato de serem capazes de melhorar o desempenho dos métodos estatísticos tradicionais;
- medicina: diagnóstico de pacientes e auxílio à tomada de decisão;
- jogos: criação de oponentes controlados pelo computador que reajam de forma mais "inteligente" às ações dos jogadores humanos.

RESUMO DO TÓPICO 2

- Cada neurônio isoladamente é muito simples, mas a enorme e complexa rede de neurônios que temos em nosso cérebro é capaz de processar informações de extrema complexidade a uma velocidade impressionante.

- O cérebro humano tem uma propriedade chamada plasticidade, o que significa que um determinado número de neurônios pode se reorganizar em resposta a eventos que ocorram, o que ocasiona o aprendizado.

- As redes artificiais de neurônios são modeladas de forma análoga ao cérebro humano, sendo compostas por vários neurônios artificiais.

- As RNAs podem ser classificadas de acordo com diferentes critérios, por exemplo, quanto ao número de camadas de neurônios, quanto às conexões entre esses neurônios e quanto ao fluxo de informação entre as camadas.

- O processo de treinamento da rede é bastante delicado, pois a mesma não deve ser treinada 100% para cada fato a ser reconhecido, mas treinada com o conjunto inteiro, sendo apresentadas as entradas várias vezes iterativamente.

- Caso uma rede seja treinada 100% para um fato de cada vez, isso afetará negativamente o conhecimento de todos os fatos previamente aprendidos.

- Entre as principais aplicações de redes neurais artificiais, podemos citar: reconhecimento de voz, reconhecimento de faces, previsão de cotações na bolsa e diagnóstico de doenças.

AUTOATIVIDADE

1 Com relação às redes neurais artificiais, avalie as afirmações a seguir:

I – A camada de entrada realiza o primeiro processamento das informações.
II – Os pesos servem para fazer uma média aritmética dos valores das entradas.
III – A função de ativação é a função que determina o valor que o neurônio artificial envia em sua saída.
IV – Uma rede Perceptron pode representar seus valores somente no formato binário.

Agora assinale a alternativa que somente possui afirmações CORRETAS:

a) I, II, III e IV.
b) I, II e IV.
c) I e II.
d) III e IV.

2 Disserte sobre as principais diferenças entre as técnicas de aprendizado de máquina supervisionado e não supervisionado.

3 Por que se diz que as técnicas de clusterização são extremamente úteis para as organizações hoje em dia?

4 Com relação às RNA, avalie as afirmações a seguir:

I – As redes artificiais de neurônios são modeladas de forma análoga ao cérebro humano, sendo compostas por vários neurônios artificiais.
II – Uma RNA deve ser treinada com 100% de precisão para cada fato de entrada, garantindo desta forma um desempenho ótimo.
III – O cérebro humano é mais eficiente do que o computador no que se refere a reconhecimento de padrões.
IV – O número de camadas de uma RNA deve ser definido antes de se saber quais são os dados de entrada para o treinamento.

Agora assinale a alternativa que somente contém afirmações CORRETAS:

a) I, II, III, IV.
b) I, II, IV.
c) I, III.
d) III, IV.

5 Com relação às RNA do tipo Perceptron, assinale a alternativa CORRETA:

a) Os Perceptron representam somente as informações de entrada em formato binário, pois na saída não existe essa restrição.
b) É através da função de ativação que se calcula o valor pelo qual as entradas são multiplicadas para ativar ou não o neurônio artificial.
c) Uma vez que as informações são representadas em formato binário, o número máximo de camadas de uma RNA Perceptron é igual a dois.
d) Os neurônios de entrada não realizam efetivamente nenhum processamento, servindo apenas para distribuir a informação para as demais camadas.

TÓPICO 3

TENSORFLOW

1 INTRODUÇÃO

O TensorFlow é uma biblioteca de *software open source* para computação numérica que usa grafos de fluxos de dados. Cada nodo do grafo representa operações matemáticas, enquanto os vértices representam as matrizes multidimensionais de dados (conhecidos como *tensors*). A arquitetura flexível da biblioteca permite que se implemente soluções que executam em uma ou mais *Computer Processing Unit* (CPU) ou *Graphic Processing Unit* (GPU) em um computador *desktop*, servidor ou mobile utilizando a mesma *Application Program Interface* (API) (TENSORFLOW, 2017).

A ferramenta foi originalmente desenvolvida por pesquisadores e engenheiros trabalhando no time conhecido como Google Brain, dentro do departamento de *Machine Learning* da companhia e liberado para a comunidade no final de 2015 (TENSORFLOW, 2017). Foi desenvolvida especialmente para resolver problemas de *deep learning*, um tipo especial de redes neurais artificiais, mas pode também resolver problemas utilizando redes neurais *feed forward* (demonstradas no Tópico 2 desta unidade) e regressão linear.

Mais informações sobre a teoria dos grafos podem ser encontradas em: <https://www.ime.usp.br/~pf/teoriadosgrafos/texto/TeoriaDosGrafos.pdf>>

Você provavelmente já tirou proveito de subprodutos do *TensorFlow* indiretamente, visto que os mecanismos de resposta automática de *e-mail*, tradução, classificação de imagens e a própria pesquisa do Google utilizam a biblioteca.

Durante esta unidade demonstraremos como fazer a instalação do *TensorFlow* e treinaremos nossa primeira rede neural artificial na ferramenta, de modo a tentar prever os passageiros que sobreviveram ao naufrágio do Titanic, utilizando dados reais.

2 INSTALAÇÃO

O *TensorFlow* possui suporte para Linux, windows e macOS, dependendo das suas respectivas versões. Para fins de estudo veremos a sua instalação na distribuição Linux conhecida como Ubuntu. Você pode optar por criar uma máquina virtual em um *software* de virtualização que execute no Windows, por exemplo, o VirtualBox, distribuído gratuitamente em <https://www.virtualbox.org/wiki/Downloads>. Caso prefira, você pode fazer uma instalação do Ubuntu em paralelo com seu sistema operacional atual e selecionar o sistema a ser executado na inicialização do computador. É possível instalar o *TensorFlow* no sistema operacional Windows através do *container Docker* ou mesmo do *Linux Bash*, sendo necessário ainda instalar o *Python*. Como esses tópicos ultrapassam o escopo deste material, optamos por utilizar o *Linux*.

Uma das vantagens de se optar pelo Ubuntu é que a linguagem de programação *Python*, atualmente a que possui a api mais completa para desenvolvimento no *TensorFlow*, já vem nativamente instalada no sistema operacional (TENSORFLOW 2017).

Mais informações sobre a distribuição Linux Ubuntu podem ser encontradas em: <https://www.ubuntu.com/>.

Demonstraremos a instalação em um ambiente com suporte a CPU e virtual environment no python 2.7. Os passos a serem executados no terminal estão demonstrados no Quadro 32.

QUADRO 32 - INSTALAÇÃO DO TENSORFLOW

1) Instale o pip e o virtualenv através do seguinte comando:
   sudo apt-get install python-pip python-dev python-virtualen
2) Crie um virtual environment para execução do tensorflow dentro do python sem interferir na instalação do sistema operacional:
   virtualenv --system-site-packages ~/tensorflow
3) Ative o virtual environment:
   source ~/tensorflow/bin/activate
4) O cursor do terminal deve mudar para o seguinte: **(tensorflow)$**
5) Instale o tensorflow propriamente dito com o comando:
   pip install --upgrade tensorflow
6) Sempre que for utilizar o tensorflow, o virtual environment deve ser ativado através do comando:
   source ~/tensorflow/bin/activate
7) E para desativá-lo:
   deactivate

FONTE: Adaptado de Tensorflow (2017)

Para validar a instalação, escreveremos um pequeno programa em Python dentro do shell do mesmo. Os passos estão descritos no Quadro 33.

QUADRO 33 - VALIDANDO A INSTALAÇÃO DO TENSORFLOW

1) Chame o shell do python digitando o seguinte comando no terminal:
   python
2) Digite os comandos abaixo, pressionando ENTER ao encerrar cada um deles:
   >>> **import tensorflow as tf**
   >>> **hello = tf.constant('Hello, TensorFlow!')**
   >>> **sess = tf.Session()**
   >>> **print(sess.run(hello))**
3) Ao final, deve aparecer escrita no terminal a mensagem **Hello TensorFlow!**

FONTE: Adaptado de Tensorflow (2017)

Se você conseguiu executar os passos acima, PARABÉNS! Você instalou o *TensorFlow* com sucesso e agora podemos "brincar" com a biblioteca.

Caso você tenha encontrado problemas durante a instalação do *TensorFlow*, existe uma seção para a resolução de problemas no *site* da biblioteca <http://www.tensorflow.org>. Você também pode entrar em contato com a nossa tutoria através dos diversos canais de comunicação disponibilizados.

3 IMPLEMENTAÇÃO DE UM MODELO NO *TENSORFLOW*

Demonstraremos a utilização do *TensorFlow* na previsão de sobreviventes do naufrágio do Titanic. Os dados reais dos passageiros podem ser obtidos na seguinte URL: <https://www.kaggle.com/c/titanic/data>. Junto com os dados existe uma descrição deles, de forma a facilitar o entendimento. Quanto maior seu entendimento, melhor será seu resultado final. Os dados são disponibilizados no formato *Comma Separated Values* (CSV) e podem ser visualizados em qualquer *software* de planilha ou mesmo editores de texto puro. A Figura 82 mostra parte dos dados que utilizaremos para treinar nossa primeira rede neural artificial.

O código-fonte que faz o treinamento da rede neural será descrito a seguir, mas antes de tentar interpretá-lo, cabem algumas considerações. Inicialmente o modelo é treinado com algumas colunas de dados dos passageiros que podem ou não ter influência em sua sobrevivência ou morte. Por exemplo: o nome de um passageiro não tem influência em sua sobrevivência. Isso seria como dizer que os passageiros chamados João têm mais probabilidade de sobreviver que os passageiros chamados Paulo. Por outro lado, o sexo e a idade definitivamente têm relação com a sobrevivência, visto que mulheres e crianças embarcaram primeiro no navio. O trabalho do cientista de dados envolve inicialmente conhecer os dados e definir quais são ou não relevantes para o trabalho a ser realizado. Muitas vezes envolve tentativa e erro.

Uma vez que os dados foram analisados, parte-se para a definição do algoritmo/método que pode fazer a melhor interpretação dos dados. Nesse exemplo, utilizaremos regressão logística.

FIGURA 82 - DATASET DO TITANIC

```
PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
8,0,3,"Palsson, Master. Gosta Leonard",male,2,3,1,349909,21.075,,S
9,1,3,"Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)",female,27,0,2,347742,11.1333,,S
10,1,2,"Nasser, Mrs. Nicholas (Adele Achem)",female,14,1,0,237736,30.0708,,C
11,1,3,"Sandstrom, Miss. Marguerite Rut",female,4,1,1,PP 9549,16.7,G6,S
12,1,1,"Bonnell, Miss. Elizabeth",female,58,0,0,113783,26.55,C103,S
13,0,3,"Saundercock, Mr. William Henry",male,20,0,0,A/5. 2151,8.05,,S
14,0,3,"Andersson, Mr. Anders Johan",male,39,1,5,347082,31.275,,S
15,0,3,"Vestrom, Miss. Hulda Amanda Adolfina",female,14,0,0,350406,7.8542,,S
16,1,2,"Hewlett, Mrs. (Mary D Kingcome) ",female,55,0,0,248706,16,,S
17,0,3,"Rice, Master. Eugene",male,2,4,1,382652,29.125,,Q
18,1,2,"Williams, Mr. Charles Eugene",male,,0,0,244373,13,,S
19,0,3,"Vander Planke, Mrs. Julius (Emelia Maria Vandemoortele)",female,31,1,0,345763,18,,S
20,1,3,"Masselmani, Mrs. Fatima",female,,0,0,2649,7.225,,C
21,0,2,"Fynney, Mr. Joseph J",male,35,0,0,239865,26,,S
22,1,2,"Beesley, Mr. Lawrence",male,34,0,0,248698,13,D56,S
23,1,3,"McGowan, Miss. Anna ""Annie""",female,15,0,0,330923,8.0292,,Q
24,1,1,"Sloper, Mr. William Thompson",male,28,0,0,113788,35.5,A6,S
25,0,3,"Palsson, Miss. Torborg Danira",female,8,3,1,349909,21.075,,S
26,1,3,"Asplund, Mrs. Carl Oscar (Selma Augusta Emilia Johansson)",female,38,1,5,347077,31.3875,,S
27,0,3,"Emir, Mr. Farred Chehab",male,,0,0,2631,7.225,,C
28,0,1,"Fortune, Mr. Charles Alexander",male,19,3,2,19950,263,C23 C25 C27,S
29,1,3,"O'Dwyer, Miss. Ellen ""Nellie""",female,,0,0,330959,7.8792,,Q
30,0,3,"Todoroff, Mr. Lalio",male,,0,0,349216,7.8958,,S
31,0,1,"Uruchurtu, Don. Manuel E",male,40,0,0,PC 17601,27.7208,,C
32,1,1,"Spencer, Mrs. William Augustus (Marie Eugenie)",female,,1,0,PC 17569,146.5208,B78,C
33,1,3,"Glynn, Miss. Mary Agatha",female,,0,0,335677,7.75,,Q
34,0,2,"Wheadon, Mr. Edward H",male,66,0,0,C.A. 24579,10.5,,S
35,0,1,"Meyer, Mr. Edgar Joseph",male,28,1,0,PC 17604,82.1708,,C
36,0,1,"Holverson, Mr. Alexander Oskar",male,42,1,0,113789,52,,S
```

FONTE: Disponível em: <https://www.kaggle.com/c/titanic/data>. Acesso em: 18 abr. 2017.

Iniciamos nosso desenvolvimento com as importações necessárias do *tensorflow* e da biblioteca os, necessária para a leitura do arquivo csv. Nas linhas 4 e 5, declaramos duas variáveis do *tensorflow* para conter um vetor de pesos e o bias, respectivamente (Quadro 34).

QUADRO 34 - INÍCIO DA IMPLEMENTAÇÃO

```
1 import tensorflow as tf
2 import os
3
4 W = tf.Variable(tf.zeros([5, 1]), name="weights")
5 b = tf.Variable(0., name="bias")
```

FONTE: O autor

Os pesos têm cinco linhas e somente uma coluna, pois os utilizaremos para cada uma das características que considerarmos para treinar a rede neural. No Quadro 35 continuamos com a implementação.

QUADRO 35 - FUNÇÕES DA IMPLEMENTAÇÃO

```
7  # Combina as entradas lidas com os pesos e o bias
8  def combine_inputs(X):
9      return tf.matmul(X, W) + b
10
11 # o novo valor calculado é obtido aplicando uma função sigmoide nos dados de entrada
12 def inference(X):
13     return tf.sigmoid(combine_inputs(X))
14
15 # cálculo da perda
16 def loss(X, Y):
17     return tf.reduce_mean(tf.nn.sigmoid_cross_entropy_with_logits(combine_inputs(X), Y))
```

FONTE: O autor

Na linha 8 da Quadro 35 começa a função que combina as entradas do arquivo de treinamento, com os pesos e o bias, que consiste basicamente de uma multiplicação de matrizes. No Tópico 2 desta unidade tratamos de redes neurais com valores escalares, a diferença é que agora os valores estão em matrizes multidimensionais. O cáculo dos novos valores para os pesos e o bias é então submetido a uma função sigmoide de ativação (linha 13). Por fim, na linha 13 é calculada a perda, que representa efetivamente o aprendizado da rede neural. A perda é calculada comparando o resultado que a rede trouxe na iteração de treinamento atual com o resultado esperado. Se o resultado foi menor, a rede se ajusta aumentando o valor dos pesos e bias gradativamente, de modo a se aproximar do resultado esperado e reduzir a perda conforme o treinamento avança. Se o resultado for maior, a rede se ajusta diminuindo o valor dos pesos e bias, buscando também se aproximar do resultado esperado com a diminuição da perda.

Se você tiver dificuldades com a linguagem de programação *python*, sugerimos o material disponível no *site*: <http://turing.com.br/pydoc/2.7/tutorial/>.

A função mostrada no Quadro 36 (linha 20) lida com as entradas de dados para a rede. A linha 20 carrega um conjunto de variáveis com informações do arquivo csv. Nesse caso, todas as informações são utilizadas, sendo que são carregadas 100 de cada vez. O procedimento das linhas 25 a 27 é a transformação de valores não numéricos em valores numéricos, no caso, a classe do passageiro. Visto que a rede neural é treinada com números, obrigatoriamente todas as informações devem ser transformadas. Não é suficiente dizer que a primeira classe corresponde ao 1, a segunda classe a 2 e assim sucessivamente. Dessa forma, a rede poderia inferir que a terceira classe é igual a 3 vezes a primeira, o que não faz sentido. Para este tipo

de situação, utiliza-se uma representação do tipo perceptron, também conhecida como one-hot. Neste caso, com três classes, cria-se uma representação onde sempre existem três valores. Se for primeira classe, a variável primeira classe recebe 1, enquanto a segunda e a terceira recebem 0. Se for segunda classe, a primeira recebe 0, a segunda recebe 1 e a terceira recebe 0. A segunda transformação é com o sexo. Na linha 29, fazemos uma comparação entre o valor do sexo e a string "female". Se forem iguais, a expressão retorna true, que convertido para float no Python retorna 1. Se forem diferentes, a expressão retorna 0. Por fim, nas linhas 33 e 34 são montadas as matrizes com os valores que serão efetivamente considerados para o treinamento da rede: primeira, segunda ou terceira classe, gênero e idade.

QUADRO 36 - TRATAMENTO DA ENTRADA DE DADOS

```
18
19  def inputs():
20      passenger_id, survived, pclass, name, sex, age, sibsp, parch, ticket, fare, cabin, embarked = \
21          read_csv(100, 'train.csv', [[0.0], [0.0], [0], [""], [""], [0.0], [0.0],
22                                     [0.0], [""], [0.0], [""], [""]])
23
24      # Converte dados não numéricos
25      is_first_class = tf.to_float(tf.equal(pclass, [1]))
26      is_second_class = tf.to_float(tf.equal(pclass, [2]))
27      is_third_class = tf.to_float(tf.equal(pclass, [3]))
28
29      gender = tf.to_float(tf.equal(sex, ["female"]))
30
31      # Matriz com as características que são utilizadas para treinamento e matriz somente com a
32      # informação de sobrevivência
33      features = tf.transpose(tf.pack([is_first_class, is_second_class, is_third_class, gender, age]))
34      survived = tf.reshape(survived, [100, 1])
35
36      return features, survived
```

FONTE: O autor

No Quadro 37 é mostrado o procedimento de treinamento e avaliação de nosso modelo. Para o treinamento, definimos uma taxa de aprendizado de 0,01 e utilizamos a função *Gradient Descent do TensorFlow* (linhas 39 e 40). Para avaliar os resultados, estes são comparados com um subconjunto aleatório de dados do *dataset*, onde os acertos em relação à sobrevivência são contados.

Toda resposta onde o resultado é maior que 0,5 é computada como um SIM, indicando que o passageiro sobreviveu. Finalmente, a função *reduce_mean* é utilizada para contar todas as respostas corretas e dividir pelo número total de dados do *subset*, o que nos dá o percentual de respostas corretas. Como os dados são escolhidos de forma aleatória, rodar duas vezes o mesmo modelo pode gerar resultados finais diferentes.

QUADRO 37 - TREINAMENTO E AVALIAÇÃO DO MODELO

```
38 def train(total_loss):
39     learning_rate = 0.01
40     return tf.train.GradientDescentOptimizer(learning_rate).minimize(total_loss)
41
42 def evaluate(sess, X, Y):
43     predicted = tf.cast(inference(X) > 0.5, tf.float32)
44     print (sess.run(tf.reduce_mean(tf.cast(tf.equal(predicted, Y), tf.float32))))
```

FONTE: O autor

O Quadro 38 traz código-fonte referente à leitura dos dados do arquivo CSV, o que é feito quase que inteiramente pelo *Python*, extrapolando o escopo deste material, e por isso não será explicado. Entretanto, o código deve ser digitado juntamente com as demais funções, caso contrário o *script* não conseguirá ler os dados do CSV e, consequentemente, não funcionará.

QUADRO 38 - LEITURA DO ARQUIVO CSV

```
47 def read_csv(batch_size, file_name, record_defaults):
48
49         filename_queue = tf.train.string_input_producer([os.path.dirname(__file__)
50                                                          + "/" + file_name])
51
52         reader = tf.TextLineReader(skip_header_lines=1)
53         key, value = reader.read(filename_queue)
54
55         decoded = tf.decode_csv(value, record_defaults=record_defaults)
56
57         return tf.train.shuffle_batch(decoded,
58                                       batch_size=batch_size,
59                                       capacity=batch_size * 50,
60                                       min_after_dequeue=batch_size)
```

FONTE: O autor

QUADRO 39 - EXECUÇÃO DO TREINAMENTO NO TENSORFLOW

```
64 with tf.Session() as sess:
65
66     tf.initialize_all_variables().run()
67     caminho = os.path.dirname(__file__) + "/train.csv"
68     X, Y = inputs()
69     total_loss = loss(X, Y)
70     train_op = train(total_loss)
71     coord = tf.train.Coordinator()
72     threads = tf.train.start_queue_runners(sess=sess, coord=coord)
73     training_steps = 1000
74     for step in range(training_steps):
75         sess.run([train_op])
76         # para fins de debug
77         if step % 10 == 0:
78             print ("loss: ", sess.run([total_loss]))
79
80     evaluate(sess, X, Y)
81
82     coord.request_stop()
83     coord.join(threads)
84     sess.close()
```

FONTE: O autor

No Quadro 39 é ilustrada a rotina principal de nosso *script*. Na linha 64 é instanciado um objeto *Session* do *TensorFlow* e daí por diante podemos chamar suas funções. Perceba que as funções que definimos nos quadros anteriores, como *inputs*, *loss* e *train*, são chamadas dentro desse trecho do código. Definimos ainda 1000 passos para treinamento, o que pode não ser suficiente para atingir um resultado satisfatório, mas sem dúvida é o suficiente para demonstrar o aprendizado da rede. A cada 100 passos é feita a impressão da perda, para fins de *debug* e verificação da diminuição do valor, conforme ilustrado no Quadro 32. Se sua perda aumenta, algo está errado e sua rede não conseguirá "aprender" o conteúdo dos dados de treinamento.

Na Figura 83 também é possível perceber que a perda aumenta em uma iteração e diminui em outra, caracterizando a tentativa de atingir a perda mínima. Se o valor obtido para determinada iteração no treinamento é maior que o esperado, os pesos e bias são ajustados de forma a diminuir a saída e, consequentemente, a perda, na próxima iteração. Se o valor obtido for menor, os pesos e bias são ajustados de forma a aumentar a saída e a perda na próxima iteração. Por fim, é impresso o valor da acurácia obtido no método *evaluate*. Em geral, os dados são separados em dois subgrupos:

- **Treinamento**: dados efetivamente utilizados para o treinamento da rede, em geral composto por 80% do total.
- **Avaliação**: dados para teste da rede uma vez treinada. O procedimento consiste em aplicar o algoritmo nos 20% restantes e verificar a acurácia.

Nesse caso, obtivemos uma acurácia de 76%, o que é bastante positivo para uma rede neural tão simples, com somente 1000 passos de treinamento. Seus resultados podem variar, uma vez que os dados de treinamento são enviados para a rede neural em batches de tamanho variável. Isso implica inclusive que, se você rodar duas vezes o mesmo algoritmo, pode obter resultados diferentes, pois não há garantia de que exatamente os mesmos dados serão utilizados. Na verdade, a probabilidade disso acontecer é praticamente nula.

FIGURA 83 - RESULTADO DO TREINAMENTO DA REDE NEURAL

```
(tensorflow3.5) crfranco@CRFranco-VirtualBox:~$ python titanic2.py
loss:  [0.74723101]
loss:  [0.65298927]
loss:  [0.66269392]
loss:  [0.62551087]
loss:  [0.60454142]
loss:  [0.66782427]
loss:  [0.61239713]
loss:  [0.62798876]
loss:  [0.62444943]
loss:  [0.61509848]
loss:  [0.58837926]
loss:  [0.538885]
loss:  [0.55904174]
loss:  [0.56768054]
loss:  [0.57575476]
loss:  [0.69834918]
loss:  [0.55603254]
loss:  [0.59083045]
loss:  [0.59374547]
loss:  [0.6203441]
0.76
```

FONTE: O autor

Assim, encerramos nossa Unidade 3. Esperamos que você tenha aproveitado a experiência de montar um modelo, treinar uma rede neural e avaliar sua acurácia utilizando o *TensorFlow*. A simplicidade e a praticidade da ferramenta tornam possível sua utilização por qualquer pessoa com um mínimo de noção de programação, abstraindo a complexidade matemática que em geral envolve a utilização das RNA.

Em caso de dúvidas, não hesite em entrar em contato conosco através dos canais disponíveis e não se esqueça de realizar as autoatividades, elas são essenciais para consolidar o conhecimento que você recém-adquiriu.

Bons estudos e até a próxima!

LEITURA COMPLEMENTAR

Cognifying: A força inevitável. Adaptação livre do autor para o texto retirado de <http://ofuturodascoisas.com/cognifying-a-forca-inevitavel/>.

O que é uma força inevitável?

Tivemos diversas evoluções no decorrer da história. O homem criou o fogo, que permitiu criar a luz, depois a eletricidade, até chegarmos na superconectividade através da internet, que segue uma linha cada vez mais exponencial e inevitável.

Uma definição do que é inevitável são as forças que movem as tecnologias ao longo do tempo, ou seja, o processo científico por trás delas. Dentro desta análise, olhando para o nosso tempo, as redes interconectadas por todo o mundo (a.k.a internet) é que ditam o ritmo das tecnologias que temos hoje.

Isso implica que não poderíamos cravar a existência do Google, o Uber, o Iphone etc., porém, seria possível observar as forças que motivaram a criação deles, como o sistema de busca universal de informações, a ideia do acesso ao transporte particular compartilhado ou um sistema de comunicação a longa distância. Portanto, a compreensão do processo científico pode multiplicar o número de inovações.

Quando está chovendo, conseguimos saber com absoluta certeza qual a direção das gotas de chuva (para baixo), mas é difícil saber o caminho delas quando estão na superfície. Assim são as forças inevitáveis: conseguimos enxergar padrões e direções, mas, dificilmente o caminho que elas vão percorrer.

Força inevitável: cognificação

A primeira Revolução Industrial foi potencializada por uma força inevitável que revolucionou toda a sociedade. Através da eletricidade tivemos acesso a carros, geladeiras e até mesmo ao computador. Ela foi fundamental para sustentar o paradigma industrial.

Estamos no princípio de um novo paradigma que pode moldar uma nova etapa na evolução da humanidade. A cognificação pode significar uma mudança da evolução pela seleção natural para uma outra, orientada pela inteligência. O verbo "cognificar", que significa adicionar inteligência a um organismo, representa bem essa nova era.

O que caracteriza a cognificação como inevitável é o avanço exponencial do poder computacional, a abundância de dados *on-line*, e também todos conectados a uma rede comum, além de tudo, estar mais acessível. Significa dizer que temos o cenário ideal para criar novas formas de inteligências não limitadas pela biologia, as chamadas inteligências artificiais (AIs).

Já temos AIs no nosso cotidiano em forma de recomendação do Netflix e Spotify, nas *tags* automáticas do Facebook e Instagram, ou até como assistentes virtuais (Siri, Cortana, Allo). Elas agem como um regulador de hábitos, influenciando sem você perceber o seu próprio comportamento.

Estamos diante da cognificação de qualquer coisa, ou seja, qualquer objeto, processo ou serviço pode tornar-se inteligente ao se conectar com uma rede de AI.

Dentro deste novo paradigma, o que antes era moldado pela eletricidade será pela cognificação.

Vamos imaginar um cenário de cognificação no trânsito:

Temos um sinal de trânsito com AI que permite regular os seus comandos com base na conexão com toda a via de tráfego que também possui elementos de AI, assim como os carros autônomos que se comunicam com outros carros, evitando possíveis congestionamentos ou até batidas. Com esta colaboração das Ais, toda a mobilidade urbana tende a se transformar e ficar mais eficiente e inteligente.

Imagine o impacto que uma cognificação em massa pode ter em processos industriais e empresariais, nas cirurgias clínicas, no avanço no ensino, na distribuição de recursos, e na gestão pública, por exemplo. Tudo isso pode ser feito de forma mais eficiente e rápida por uma AI.

Apenas lembrando que a AI já vem sendo desenvolvida há um bom tempo. O principal fator do seu avanço é que, ao invés de ser toda programada por um humano, agora ela está sendo ensinada como aprender por conta própria.

Isso se deve bastante a pelo menos quatro elementos que eu considero essenciais para compreender a cognificação:

1- Big Data: A torrente de dados que nutrem o aprendizado da AI.
2- Redes Neurais: Desenvolvidas tendo como base o funcionamento do cérebro humano.
3- Deep Learning: Uma forma de treinamento que acelera bastante o aprendizado da AI.
4- Cloud Computing: Redes de AIs que aprendem instantaneamente de acordo com a experiência de cada AI.

Vamos conhecer mais a fundo cada um e o que já está sendo cognificado através deles.

Big Data

Uma vez que grande parte dos serviços passa a ser digital, todo tipo de dados, desde relacionamentos, estilo musical, preferência política, classe social, sentimentos, passa a ser coletado por uma imensa base de dados, o Big Data. Isso

ajuda a fomentar um sistema que pode personalizar vários serviços para você. Porém, há uma contrapartida: o acesso às nossas informações.

Podemos ter uma assistente virtual como a Siri da Apple ou até os chatbots que estão sendo utilizados na comunicação de algumas empresas. Munidos de dados *on-line* e sendo ensinados a aprender padrões de linguagem natural, para que possam auxiliar na compra de um produto, indicar desde conteúdos ou até locais legais para conhecer, e ser uma companhia para as pessoas.

O Instituto Gartner aponta que em 2020 os chatbots possam substituir os apps, ou seja, ao invés de precisar de vários apps, um chatbot nutrido de todos os serviços (Waze, Doc, etc.) que você utiliza poderia otimizar o seu dia.

Outra aplicação interessante com a união do Big Data e a Internet das Coisas serão dispositivos que possam ajudar a monitorar a nossa saúde para prevenir potenciais doenças.

O Scanadu é uma ferramenta de medicina personalizada, inspirado na série Star Trek que opera através de um "médico" AI (Watson) dando um diagnóstico instantâneo.

"Com o ritmo de crescimento da AI, é bem provável que uma criança nascida hoje não vá precisar ir a um médico quando ela for adulta no futuro" – Alan Green, fundador do Scanadu.

Portanto, somos professores e os algoritmos aprendem através da nossa montanha de informações, porém eles estão se tornando cada vez mais independentes.

Redes Neurais

Imagine que você esteja caminhando em uma praça. Neste momento o seu cérebro está trabalhando paralelamente em uma série de coisas, registrando, categorizando e interpretando imagens e sons de todas as coisas que estão ao seu redor, além de tentar prever o que pode acontecer depois. Um processo que realizamos a qualquer momento e que é operado através da rede neural biológica.

Andrew Ng, um dos precursores de algoritmos que funcionam como o cérebro humano, percebeu, através de um chip (GPU) utilizado nos jogos de videogames mais complexos, que poderia rodar redes neurais em paralelo. Assim, seria possível realizar de forma independente diversas ações diferentes, com a vantagem de ser simulado por um *software*, ou seja, seria digital e exponencial, sem limites biológicos. Podemos programar a rede neural para aprender qualquer coisa.

As redes neurais estão sendo ensinadas a fazer de tudo, desde criar o roteiro de um filme, compor uma canção, aprender a jogar um jogo ou até ser um tradutor tão eficiente quanto um humano. Há uma gama incrível de possibilidades

para se aproveitar neste campo. Há inclusive um exemplo brasileiro no uso de redes neurais para criar uma nova composição para o cantor de rapper Sabotage, morto em 2003.

A música Neuron, gravada pelo grupo RZO e com a participação do Sabotage via uma rede neural criada pela Kunumi, que faz parte do grupo Flag.cx em conjunto com o Spotify (onde foi lançada), é o primeiro rap no mundo criado com o uso da AI. A rede neural foi treinada com as produções intelectuais do rapper que eram avaliadas por um "braintrust" composto pelos realizadores do projeto, que validava cada frase criada.

O resultado de uma produção musical através de uma AI demonstra o que pode estar por vir dentro deste campo. O chip (GPU) para operar as redes neurais hoje está bastante acessível, o que aumenta exponencialmente o número de aplicações e testes.

Estamos diante de um sistema com o processamento de um computador e a versatilidade do cérebro humano.

Deep Learning

Da mesma forma que temos um algoritmo que funciona como um cérebro humano, há uma forma de treinamento da AI que é crucial para intensificar a interpretação e compreensão de imagens, palavras e sons. Este método é o que permite uma maior eficiência em como a AI vê ouve, interpreta e escreve as coisas. A cognificação dos sentidos humanos.

Para exemplificar, no modelo mais comum de aprendizado máquina temos milhares de vídeos sobre cães correndo na praia. A cada momento que você assiste a estes vídeos está treinando o algoritmo do Youtube para entender o que é um "cão correndo na praia".

No entanto, com o advento do *deep learning* (ou "aprendizado profundo") é possível não só aprender com o estímulo humano, mas o próprio algoritmo do Youtube vai sendo treinado a analisar diversas camadas (fragmentos do vídeo) daquela situação para interpretar por "conta própria" e afirmar que é um "cão correndo na praia" e não um gato, cavalo, etc.

Tivemos diversos avanços no aprendizado via deep learning neste ano de 2016, apenas para citar alguns:

– A AlphaGo AI do Google venceu o campeão mundial de GO, um jogo hipercomplexo até mesmo para humanos.

– AI do MIT interpreta apenas pela imagem quais sons estão sendo realizados e criar sons em vídeos, o que pode ser utilizado para criar trilhas sonoras para filmes e séries no futuro.

– IA Watson detectou um tipo raro de leucemia em uma mulher no Japão em 10 minutos, um processo que se feito por humanos levaria duas semanas.

O Google, uma das empresas que mais investem em AI (foram 11 aquisições até agora), possui a infraestrutura ideal para desenvolver AI, um amplo banco de dados e uma estrutura computacional de dar inveja. Para se ter uma ideia a empresa tem dentro do seu Lab o criador do deep learning, Geoff Hinton, e Ray Kurzweil, que, além de ser um dos mais renomados futuristas, possui diversas invenções no campo de AI.

Mas, apesar da otimização de como a AI aprende, faltava uma plataforma *open-source* onde pudesse ser utilizada como um campo de treinamento ou escola para treinar essas AIs.

O Universe, projeto criado através da Open AI (iniciativa do Elon Musk), veio para preencher um pouco esse *gap*. Ele utiliza vários conteúdos usados por nós e, principalmente, *games* para treinar a AI em ambientes próximos à realidade, ou seja, é uma espécie de escola aberta, onde a AI pode aprender sobre várias tarefas, em uma técnica de tentativa e erro chamada "reinforcement learning".

Os carros autônomos estão sendo treinados utilizando o jogo GTA, que por ter uma narrativa que se passa em uma cidade, vem ajudando bastante o algoritmo a entender diversas circunstâncias possíveis na vida real. Outro fato interessante é que, por se tratar de ambientes dinâmicos, otimiza a percepção das AIs.

Ainda sobre carros autônomos, o MIT elaborou um teste para entender como a AI vai reagir em situações conflitantes de cunho ético, como em uma colisão inevitável em que você precisa escolher salvar uma criança e o seu cachorro ou um casal de idosos.

Cloud Computing

Para entender porque a cloud computing (computação na nuvem) é tão relevante para a cognificação, vou fazer um paralelo como seria a nossa forma de aprender se tivéssemos todos conectados a uma única rede.

Imagina que você acabou de aprender algo sobre física quântica. Ao mesmo tempo que você aprendeu o assunto, todos que estão conectados à sua rede, instantaneamente, aprendem a mesma coisa. Em resumo, nós seríamos organismos conectados a uma inteligência unificada.

Apesar disso ainda não ser possível para gente, na AI acontece assim: qualquer coisa que uma AI aprenda, todas as outras da rede também vão aprender.

Isso é o que permite a AI Watson encontrar diagnósticos mais rápido do que qualquer especialista em câncer no mundo. Supondo que uma mutação rara surgiu no Brasil e o Watson nunca se deparou com tal circunstância, somente

pelo fato do Watson estar conectado a uma rede de AI's, ele vai conseguir dar um diagnóstico preciso.

O que esperar da cognificação no futuro?

Uma visão possível é que a cognificação pode ser aquilo que alguns futuristas chamam de Singularidade. A inteligência passa a ser um fator comum na evolução de qualquer organismo. Penso em uma escala da cognificação que começa por:

objetos inertes → processos → serviços → empregos → organizações → seres biológicos.

Vale citar a visão do Kevin Kelly (fundador da revista Wired Magazine – http://www.wired.com) quando ele diz que não estamos no caminho para criar uma AI consciente, mas sim diversas espécies diferentes com propósitos e especialidades distintas. São organismos que pensam totalmente diferente dos humanos e por isso eles são tão essenciais. Para ele, a mesma AI que vai fazer um diagnóstico da sua doença é diferente da que dirige um carro autônomo: "cada um no seu quadrado".

Queremos que o nosso carro autônomo esteja focado na estrada, e não com um problema com a garagem. Também não faz sentido o Watson "médico" ter distrações durante uma consulta. Estamos criando tipos de inteligências que serão completamente diferentes da nossa". – Kevin Kelly

Portanto, o grande ponto da cognificação é hackear uma falha biológica que não permite expandir a nossa inteligência. Com a integração e a interação com organismos inteligentes será possível encontrar respostas para grandes questões que afligem a humanidade há séculos, como os segredos do universo, e até mesmo entender a nossa mente.

Apenas lembrando que citamos o impacto que tem com a convergência com outras tecnologias, por exemplo, com a biotecnologia e a nanotecnologia. O que causa a verdadeira disruptura são todas elas nutridas por cognificação a serviço da humanidade.

Finalizamos este texto com uma mensagem do próprio Kevin Kelly:

"Daqui a milhares de anos, quando os historiadores revisitarem o passado, nosso tempo hoje será visto como o início de um período extraordinário. O momento em que os habitantes deste planeta começam a se conectar a um grande organismo unificado. As pessoas do futuro irão nos invejar e desejar ter presenciado o nascimento disso tudo".

Por essa razão somos os arquitetos de um futuro que ainda não existe, porém as nossas escolhas e a nossa própria consciência irão decidir qual o futuro que vamos deixar para as próximas gerações.

RESUMO DO TÓPICO 3

- Cada neurônio isoladamente é muito simples, mas a enorme e complexa rede de neurônios que temos em nosso cérebro é capaz de processar informações de extrema complexidade a uma velocidade impressionante.

- O TensorFlow é uma biblioteca de *software open source* para computação numérica que usa grafos de fluxos de dados.

- Cada nodo do grafo representa operações matemáticas, enquanto os vértices representam as matrizes multidimensionais de dados (conhecidos como tensors).

- O *deep learning* é uma modalidade especial de treinamento de redes neurais artificiais, onde um número expressivo de camadas ocultas é utilizado.

- O Google utiliza o *deep learning* em diversos de seus produtos, como a busca, a tradução e o reconhecimento de voz.

- A api mais rica do tensor *flow* atualmente é desenvolvida para a linguagem de programação *open source Python*.

- O tensor *flow* facilita as operações com matrizes multidimensionais, conhecidas como *tensors*.

- Uma das etapas mais importantes do trabalho do cientista de dados é a avaliação dos dados que serão utilizados para o treinamento da rede neural artificial.

AUTOATIVIDADE

1 Considerando a implementação demonstrada ao longo do tópico 3, altere o número de passos que o algoritmo utiliza para treinamento. Faça experimentos com 10000, 100000 e 1000000 passos. A perda continua diminuindo na mesma razão? O que acontece com a acurácia? Você encontrou *overfitting*?

2 *TensorFlow* é uma biblioteca *open source* que contém diversas implementações de algoritmos para aprendizado de máquina e redes neurais. Sua flexibilidade e praticidade tornam a implementação de RNA praticamente agnóstica, eliminando a necessidade de profundos conhecimentos matemáticos. Sobre o *tensorflow* e as RNA, assinale a alternativa CORRETA:

a. ( ) A perda e a acurácia são diretamente proporcionais em um treinamento não supervisionado de redes neurais artificiais.
b. ( ) Em um treinamento supervisionado, quanto maior o valor da perda, maior a acurácia da rede neural.
c. ( ) Até um certo limite, a perda e a acurácia são diretamente proporcionais em um treinamento supervisionado de redes neurais artificiais.
d. ( ) Até um certo limite, a perda e a acurácia são inversamente proporcionais em um treinamento supervisionado de redes neurais artificiais.

3 Considerando a implementação demonstrada ao longo do tópico 3, faça alterações nas variáveis que foram utilizadas como relevantes para a determinação da sobrevivência ou não do passageiro. Por exemplo, na implementação foram consideradas somente a classe, o gênero e a idade. Será que existem mais variáveis que podem interferir no resultado? Lembre-se de que boa parte do trabalho na ciência de dados consiste em conhecer seus dados e experimentar novas alternativas. Com isso em mente, dependendo de suas alterações, a acurácia pode até diminuir, mas o importante é experimentar e relatar seus resultados.

REFERÊNCIAS

ALPAYDIN, E. **Introduction to Machine Learning**. London: The MIT Press, 2010.

ARAÚJO, Fábio Meneghetti. **Controle inteligente**. 2004. Disponível em: <http://www.dca.ufrn.br/~meneghet/FTP/IA/contrint.pdf>. Acesso em: 3 mar. 2014.

ARTERO, A. O. **Inteligência artificial**: teoria e prática. Livraria da Física: São Paulo, 2009.

BÄCK, T.; FOGEL, D. B.; MICHALEWICZ, Z. **Evolutionary computation 1**: basic algorithms and operators. Bristol and Philadelphia: Institute of Physics Publishing, 2000.

BARRETO, J. M. **Inteligência artificial no limiar do século XXI**. 3. ed. Florianópolis: Duplic, 2009.

BEYER, H. G.; SCHWEFEL, H. P. Evolution strategies: a comprehensive introduction. **Natural Computing**, 3-52, 2002.

BITTENCOURT, Guilherme. **Inteligência artificial**: ferramentas e teorias. 3. ed. Florianópolis: Editora da UFSC, 2006.

BROWNLEE, J. **Clever algorithms**: nature-inspired programming recipes. Raleigh: Lulu.com, 2011.

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

CREATIVE COMMONS. **Goleiro Carlos seleção brasileira 1986**. Disponível em: <http://commons.wikimedia.org/wiki/File:Goleiro_carlos_sele%C3%A7%C3%A3o_brasileira_1986.jpg?uselang=pt-br#filehistory>. Acesso em: 27 fev. 2014.

DAUMÉ III, H. **A course in machine learning**, 2012. Disponível em: <http://ciml.info/>. Acesso em: 5 abr. 2017.

DOMINGUES, P. **The master algorithm**. New York: Basic Books, 2015.

EIBEN, A.; SMITH, J. **Introduction to evolutionary computing**. New York: Springer, 2003.

FARACO, Rafael Avila. **Uma arquitetura de agentes para negociação dentro do domínio de comércio eletrônico**. Florianópolis, 1998, 100p. Dissertação (Mestrado) – Programa de Pós-Graduação em Engenharia da Produção, Universidade Federal de Santa Catarina.

FERNANDES, Anita Maria da Rocha. **Inteligência artificial**: noções gerais. Florianópolis: Editora da UFSC, 2008.

GHAHRAMANI, Z. **Unsupervised learning**, 2004. Disponível em: <http://mlg.eng.cam.ac.uk/zoubin/papers/ul.pdf>. Acesso em: 15 abr. 2014.

GOLDBERG, D. E. **Genetic algorithms in search, optimization, and machine learning**. Reading: Addison Wesley, 1999.

GRUS, J. **Data Science from Scratch**. Sebastopol: O'Reilly Media Inc, 2015.

HARMON, Paul; KING, David. **Sistemas especialistas**: a inteligência artificial chega ao mercado. Rio de Janeiro: Campus, 1988.

HAYKIN, S. **Neural networks**: a comprehensive foundation. London: Prentice Hall International, 1999.

HERDY, M. Evolution strategies with subjective selection. **Parallel Problem Solving from Nature – PPSN**, v. IV, p. 22-31, 1996.

HORNBY, G.; GLOBUS, A.; LINDEN, D. S.; LOHN, J. D. Automated antenna design with evolutionary algorithms. **AIAA Space**, 19-21, 2006. Disponível em: <http://alglobus.net/NASAwork/papers/Space2006Antenna.pdf>. Acesso em: 1 jul. 2014.

KOZA, J. R.; POLI, R. Genetic programming. In: **Search methodologies**: introductory tutorials in optimization and decision support techniques. New York: Springer, 2005. p. 127-164.

LUGER, George F. **Inteligência artificial**: estruturas e estratégias para a resolução de problemas complexos. 4. ed. Porto Alegre: Bookman, 2002.

MICROSOFT CLIPART (2010): Figura disponível no software Microsoft Word.

MITCHELL, M. **An introduction to genetic algorithms**. Cambridge: The MIT Press, 1999.

NASSAR, Sílvia M. **Sistemas especialistas probabilísticos**. 2012. Disponível em: <http://www.inf.ufsc.br/~silvia/disciplinas/sep/material_didatico/MaterialDidatico.pdf>. Acesso em: 1 mar. 2014.

NILSSON, N. J. **Introduction to machine learning**, 1998.

NORVIG, P.; RUSSEL, S. J. **Artificial intelligence**: a modern approach. London: Prentice Hall International, 1995.

NORVIG, Peter; RUSSEL, Stuart**. Inteligência artificial**. São Paulo: Elsevier, 2004.

PENA, Sérgio D. Thomas Bayes: O cara. **Revista Ciência Hoje**, v. 38, n. 228, jul. 2006. Disponível em: <http://cienciahoje.uol.com.br/banco-de-imagens/lg/protected/ch/228/bayes.pdf/at_download/file>. Acesso em: 8 fev. 2014.

RASHID, T. **Make your own neural network**. Lavergne: CreateSpace Independent Publishing Platform, 2016.

RICH, E; KNIGHT, K. **Artificial intelligence**. Nova Iorque: McGrawHill inc., 1993.

SANTOS, C. G. **Plasticidade neuronal**. Disponível em: <http://drauziovarella.com.br/envelhecimento/plasticidade-neuronal/>. Acesso em: 19 abr. 2014.

SEEBAUER, Kurt. **Alan Turing Memorial**. 2005. Disponível em: <http://upload.wikimedia.org/wikipedia/commons/b/b8/Alan_Turing_Memorial_Closer.jpg>. Acesso em: 20 fev. 2014.

SILVEIRA Ricardo; ROSENBERG Mauro. **Representação do conhecimento**. 2007. Disponível em: <http://joaopizani.hopto.org/graduacao/disciplinas/ine5430/4_Aula2.pdf>. Acesso em: 3 mar. 2014.

SIVANANDAM, S. N.; DEEPA, S. N. **Introduction to genetic algorithms**. Berlin Heidelberg: Springer, 2008.

TAFNER, M.; FILHO, I.; XEREZ, M. **Redes neurais artificiais**: introdução e princípios de neurocomputação. Blumenau: FURB, 1996.

TENSORFLOW. **An open source software library for machine intelligence**. Disponível em: <http://www.tensorflow.org>. Acesso em: 2 de mar. 2017.

UNIVERSITY OF CALIFORNIA MUSEUM OF PALEONTOLOGY AND NATIONAL CENTER FOR SCIENCE EDUCATION. Natural Selection. **Understanding Evolution**, 2004. Disponível em: <http://evolution.berkeley.edu/evolibrary/article/evo_25>. Acesso em: 1 jul. 2014.

WOOLDRIDGE, M. **Intelligent agents** In: G. Weiss, editor: Multiagent Systems. The MIT Press. Abril, 1999.

WOOLDRIDGE, M. J. **An introduction to multiagent systems**. West Sussex: John Wiley & Sons, 2002.

ANOTAÇÕES