

EAD **UNISANTA**

INTELIGÊNCIA ARTIFICIAL

Me. Luiz Antonio Ferraro Mathias

GUIA DA DISCIPLINA

Objetivo:

A disciplina de Inteligência Artificial (IA) objetiva proporcionar ao aluno um conhecimento básico de inteligência artificial, apresentando de forma teórico-prática as informações necessárias para aplicação dos conceitos que norteiam esta área de pesquisa. Foca ainda na capacitação do aluno para distinguir tipos de conhecimento sendo utilizados em uma aplicação dita inteligente e criação de modelos que utilizem técnicas e algoritmos baseados em IA para resolução de problemas que não possam ser resolvidos pelo uso de técnicas convencionais

Introdução:

A inteligência artificial (IA) tornou-se a mais nova etapa desta evolução tecnológica, sucedendo aos processadores que eram capazes de efetuar cálculos e que, depois, evoluíram para a utilização de sistemas programáveis (como conhecemos hoje). No modelo de IA, ocorre o processamento de informações baseadas em aprendizado de experiências anteriores, algo similar ao que ocorre em relação ao funcionamento cérebro humano, uma vez que este recebe informações e vai processando-as para adquirir aprendizados. O aluno participante da disciplina de Inteligência Artificial, na modalidade de Ensino à Distância (EaD) da UNISANTA, compreenderá os fundamentos da Inteligência Artificial; as Redes Neurais Artificiais; as Redes de Funções de Base Radial (RBF), os Sistemas Nebulosos (*Fuzzy*); os Algoritmos Genéticos; a Computação Cognitiva (Visão Computacional e Processamento de Linguagem Natural) as Redes Bayesianas, o Aprendizado de máquina e os Agentes Inteligentes

1. FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

1.1 Inteligência Humana

As características vinculadas ao conceito de inteligência, como a capacidade de resolver problemas, de raciocinar, de se adaptar ao meio ambiente, foram e continuam sendo piamente reconhecidas ao longo dos anos. Desde muito tempo até os dias de hoje, pensa-se que esse conjunto de características diferencia positivamente as pessoas, dando a elas um lugar especial na sociedade. Isso foi apontado antes mesmo de o conceito de inteligência e sua medição serem práticas de estudos científicos. Hoje já se sabe que a inteligência de fato existe em todas as pessoas em maior ou menor grau.

Etimologicamente, a palavra "inteligência" se originou a partir do latim *intelligentia*, oriundo de *intelligere*, em que o prefixo *inter* significa "entre", e *legere* quer dizer "escolha". Assim sendo, o significado original deste termo faz referência a capacidade de escolha de um indivíduo entre as várias possibilidades ou opções que lhe são apresentadas.

De acordo com Richardson (1999 apud DALGALARRONDO, 2008, p. 277), “A inteligência é um conceito fundamental da psicologia moderna que todos utilizam; entretanto, quase ninguém consegue defini-la de modo definitivo ou pelo menos amplamente convincente.”.

Funções psicológicas ou conjuntos de funções graças às quais o organismo se adapta ao seu meio produzindo combinações originais de condutas, adquire e explora conhecimentos novos e, eventualmente, raciocina e resolve os problemas de uma maneira conforme as regras destacadas pelas formalizações da lógica. (DORON & PAROT, 2001).

Em outras palavras, a inteligência humana é a consequência de um conjunto de funções que o ser humano naturalmente detém, podendo ajudar na adaptação ao meio que vive, e que ao longo do tempo vão se desenvolvendo e crescendo, como por exemplo, as tomadas de decisões e as escolhas feitas quando confrontado aos obstáculos. Outros elementos que constituem a inteligência do ser humano, como o raciocínio, a memória, linguagem verbal e/ou corporal, e as emoções, são peças fundamentais para o próprio

desenvolvimento, nos quais sem elas o indivíduo não seria capaz de se tornar independente e preparado para sobreviver.

Para Sobral (2013), a inteligência humana é a capacidade de enfrentar novas situações, resolvendo problemas utilizando conceitos concretos e abstratos. O mesmo, por sua conceitua que: “[...] a inteligência é um “potencial inato” a ser desenvolvido de acordo com a personalidade, o meio ambiente e o contexto socioeconômico e histórico-cultural do sujeito.

Uma avaliação detalhada de todos os aspectos relativos à aquisição e desenvolvimento da inteligência nos permite um melhor entendimento daquilo que podemos chamar de “ação inteligente”. O conhecimento é o que faz que seja possível o encadeamento e desenvolvimento da inteligência. Diante da dificuldade de definirmos claramente o que vem a ser a inteligência, podemos descrever algumas das características identificadas nos seres ditos como inteligentes (figura 1).



Figura 1: Características identificadas nos seres ditos como inteligentes (Carvalho, 2005).

1.2 Inteligência Artificial

Com o passar do tempo, o ser humano ambicionou a criação de um ser semelhante ao seu, um ser artificial com capacidade exponencial de inteligência para poder talvez substituir o humano em suas atividades e ações cotidianas, para isso, desde os anos 40 vem sendo desenvolvida e aprimorada, a chamada “Inteligência Artificial” que, ao decorrer dos anos, vem se mostrando uma área de estudo com potencial para ajudar e complementar a Inteligência Humana.

É a ciência e engenharia de criar máquinas inteligentes, especialmente programas de computador inteligentes. Está relacionada à tarefa semelhante de utilizar computadores para entender a inteligência humana, porém a Inteligência Artificial não necessariamente deve se confinar aos métodos que são biologicamente observáveis. (JOHN MCCARTHY, 2007, p. 2, tradução nossa).

Desde o começo da utilização dos computadores pelos seres humanos existe uma tentativa de usufruir da capacidade de processamento dessas máquinas para a simulação do comportamento humano. Com o passar dos anos, diversas pesquisas foram sendo desenvolvidas com o foco nesse sentido.

Nascida no âmbito da Ciência da Computação, a Inteligência Artificial também conhecida popularmente como IA foi criada em 1940. Sua ideia originou-se da junção de várias áreas de conhecimento ao longo da história, que contribuíram de maneira significativa para o seu desenvolvimento, como: Filosofia, Matemática, Economia, Neurociência, Psicologia e Engenharia de Computadores. Pensado de forma a dar mais uma utilidade para o computador, com o intuito de emular a forma que o humano pensa, a IA possui algumas categorias como: máquinas que pensam como seres humanos; máquinas que se comportam como seres humanos; máquinas que pensam racionalmente e máquinas que se comportam de forma racional. É difícil impor uma definição assertiva para o que é inteligência artificial, mas a ideia é tentar por meio dessa emulação do raciocínio humano, gerar soluções para problemas complexos. Segundo Coppin (2010, p.4) “Inteligência Artificial envolve utilizar métodos baseados no comportamento inteligente de humanos e outros animais para solucionar problemas complexos”.

A IA abrange uma vasta área de conhecimento que consiste em técnicas como: Lógica Fuzzy, Redes Bayesianas, Redes Neurais Artificiais, Aprendizado de Máquina,

dentre outras. Um exemplo de aplicação que faz uso de inteligência artificial é tradutor do Google, por exemplo. Este famoso tradutor artificial é um dos vários sistemas de inteligência artificial que lidam com o processamento de linguagem natural. No início, era movido a uma tecnologia chamada SMT (*Statistical Machine Translation* – Tradução automática estatística), onde todas as palavras eram traduzidas para o inglês através de referências cruzadas nas bases de dados da própria Google. Com o avanço das tecnologias, este mecanismo sofrera mudanças. A partir de 2016, as traduções passaram a ser processadas por IA, capazes de processar e traduzir frases inteiras. Em essência, ele usa um método de aprendizado relacionado a muitos casos relacionados, isto é, não se baseia no aprendizado ou na aplicação das regras gramaticais de cada idioma, mas sim baseada no contexto de cada sentença.

A partir da inteligência humana, várias e diversas inovações tecnológicas que empregam técnicas de IA foram desenvolvidas, e suportadas por algoritmos inteligentes ou algoritmos de aprendizado que, entre muitos outros propósitos, são usados para identificar tendências econômicas, prever crimes, diagnosticar doenças, prever nossos comportamentos digitais etc.

De acordo com Barreto (2001) e Baranauskas (2011), é possível dividir a história da Inteligência Artificial entre as seguintes fases:

1.1.1 *Época Pré-histórica*

Nesta época, informações sobre mecanismos empregados pela mente humana eram desconhecidos. Teve como objetivo criar seres e mecanismos que apresentassem um comportamento inteligente, estendendo-se até 1875, ano em que o médico e histologista italiano Camilo Golgi consegue visualizar células nervosas do bulbo olfatório. A época obteve como sucesso mecanismos baseados em autômatos, porém foi limitada pela dificuldade que estes apresentavam para serem utilizados.

1.1.2 *Época Antiga (1875 - 1943)*

Surge a lógica formal (Russell e Gödel); O cérebro humano é reconhecido como o órgão responsável pela inteligência; David Hilbert, matemático alemão, acredita que tudo pode ser “axiomatizado” (reduzido) à lógica, porém este é refutado por Gödel com o teorema de incompletude da aritmética; Bases da Inteligência Artificial Simbólica e

Conexionista são propostas juntamente ao modelo de neurônio proposto por McCulloch e Pitts. O objetivo desta época foi compreender melhor a inteligência humana.

1.1.3 A gestão de Inteligência Artificial (1943 - 1956)

Programas de xadrez são escritos por Shannon e Turing para máquinas com a arquitetura de von Neumann; Primeiro computador baseado em redes neurais é desenvolvido por Minsky e Edmonds em 1951. O objetivo desta época foi simular a inteligência humana em situações previamente determinadas, porém foi limitada pela capacidade de processamento computacional.

1.1.4 Época Barroca: o entusiasmo da IA (1956 - 1969)

Newell e Simon desenvolvem o “*General Problem Solver*” (1957), programa projetado a fim de simular o comportamento humano na resolução de problemas; John McCarthy desenvolve a linguagem de programação LISP (1958); J. Alan Robinson propõe o “princípio da resolução”, algoritmo de provas e teoremas para a lógica de primeira ordem (1963); Pesquisas de redes neurais passam a se popularizar, como o livro “*Perceptrons*” publicado por Minsky e Papert em 1969. Esta época teve como objetivo, expandir as aplicações de inteligência artificial quanto ao uso de abordagens simbólicas e conexionistas.

1.1.5 Época das “Trevas” (1969 - 1981)

Falta de verbas forçaram a paralização de pesquisas; Edward Feigenbaum (e outros) em Stanford passam a investigar novas metodologias para sistemas especialistas; primeiras aplicações da lógica fuzzy ou lógica nebulosa passam a surgir, como o controle inteligente de Mamdani. O objetivo desta época foi encontrar soluções práticas para a inteligência artificial.

1.1.6 O renascimento (1981 - 1987)

Popularização de sistemas especialistas; Pesquisas feitas por John Hopfield em 1982 sobre redes neurais artificiais ganham credibilidade; Computadores de “quinta geração”, fomentado no Japão em 1981, fazendo uso da linguagem de programação PROLOG. O objetivo desta época foi incentivar novamente, a utilização das abordagens simbólicas e conexionistas.

1.1.7 *Época Contemporânea (1987 - atualmente)*

Redes neurais passam a serem fortemente empregadas no campo da física; Dúvidas sobre as possíveis aplicações de sistemas especialistas passam a surgir; Pesquisas envolvendo redes de crença neural começam a ser realizadas; Buscadores inteligentes aplicados à web; Emprego da inteligência artificial em: reconhecimento de voz, mineração de dados e casas inteligentes.

A inteligência artificial busca, através de símbolos computacionais, construir mecanismos e/ou dispositivos que simulem a capacidade do ser humano de pensar, resolver problemas, ou seja, de ser inteligente. O estudo e desenvolvimento desse ramo de pesquisa, conforme exposto anteriormente, tiveram início na Segunda Guerra Mundial. Os principais idealizadores foram os seguintes cientistas: Hebert Simon, Allen Newell, Jonh McCarthy e vários outros, que com objetivos em comum tinham a intenção de criar um “ser” que simulasse a vida do ser humano.

Os problemas alvo da IA envolvem: grande complexidade (número, variedade e natureza das tarefas); não há “solução algorítmica”, mas existe conhecimento; e modelagem do comportamento de um ser inteligente (conhecimento, aprendizagem, iniciativa etc.).

1.3 **Histórico da Inteligência Artificial**

As pesquisas em IA remontam da década de 40 com destaques para:

1940

Desenvolvimento de um ramo de pesquisa interessado na realização da representação das células nervosas do ser humano no computador. Esta linha de pesquisa motivou o desenvolvimento de uma formalização matemática para o neurônio e concepções sobre a forma de aprendizado deste, ou seja, como os neurônios armazenam informações.

1950

Nesta década iniciou-se o estudo na linha de pesquisa psicológica com o emprego da programação através de comandos de lógica de predicados aprimorando assim as aplicações que utilizassem esquemas de raciocínio. Desta forma se tornou possível o aperfeiçoamento de jogos, aplicações matemáticas e simuladores. Esta década foi marcada

ainda pela implementação do primeiro simulador de redes neurais artificiais e do primeiro neurocomputador.

1960

Foram identificados avanços nas pesquisas relacionadas à linha biológica, mais precisamente aos conceitos das redes neurais artificiais. Ainda nesta linha de pesquisa, a década representou a descoberta da Inteligência Artificial. Acreditava-se ser possível o desenvolvimento de ferramentas que realizassem tarefas humanas como pensamento e compreensão da linguagem, ou seja, a representação do raciocínio apenas presente nos seres humanos. Em 1969, dois grandes pesquisadores (Marvin Minski e Sigmour Papert) publicaram um livro (*Perceptrons*) no qual defendiam a tese de que os modelos de redes neurais artificiais não possuíam sustentação matemática para terem credibilidade. Este documento gerou um ostracismo entre os pesquisadores da linha biológica.

1970

Década negra para a linha biológica, mesmo com a existência de novas pesquisas envolvendo concepções de redes neurais. Tais pesquisas levaram ao desenvolvimento de modelos baseados nas potencialidades de redes de neurônios interagindo entre si, representadas na forma de mapas cerebrais, ambiente no qual havia o compartilhamento de recursos. Na linha psicológica as pesquisas culminaram com a demonstração do óbvio, ou seja, a impossibilidade de representação numa máquina dos estados mentais humanos. A Empresa Rand Corporation cria os *Sistemas Especialistas*, ampliando a Inteligência Artificial tradicional.

1980

As redes neurais artificiais recuperam sua credibilidade através dos trabalhos desenvolvidos pelo físico John Hopfield que provou em 1982 ser possível a simulação de um sistema físico através de um modelo baseado na teoria das RNA's. Ele construiu uma rede neural com a capacidade de simular a memória associativa humana, de acesso pelo conteúdo, e não pelo endereço como nos computadores digitais. Na IA tradicional ampliaram-se as técnicas e aplicações baseadas em Sistemas Especialistas.

1990

Um volume expressivo de aplicações foi implementado com as técnicas de RNA's, consolidando as redes neurais como parte integrante do estudo da Inteligência Artificial.

Por considerar que os paradigmas biológico e psicológico são complementares e necessários para o desenvolvimento de sistemas mais sofisticados e evoluídos, começam a ser concebidos os Sistemas Híbridos, os quais possuem como foco de abrangência, uma forma mais completa da representação do comportamento humano.

1.4 Teste de Turing

O matemático britânico Alan Turing , personalidade bastante influente no desenvolvimento da ciência da computação formulou em 1950 um teste (*jogo da imitação*) que segundo ele, definiria o ponto em que a máquina computacional estaria “*pensando*”.

O teste consistia em uma conversa que envolvia dois humanos e um computador, todos os três tentando parecer humanos e colocados em ambientes isolados de tal forma que um não pudesse visualizar o outro. Uma das pessoas participantes era um árbitro e se não conseguisse identificar claramente qual dos outros integrantes era o computador, então se dizia que este último passou no teste com sucesso.

Ou seja, o computador era “inteligente” se o árbitro que participou do teste não fosse capaz de dizer se foi o computador ou o ser humano que respondeu às suas perguntas.

Existem alguns programas inteligentes que “conversam em português”, criados com o objetivo de passar no Teste de Turing, conversando com os usuários como se fossem pessoas de verdade como a Sete Zoom (<http://www.inbot.com.br/sete/>).



Saiba mais

Alan Turing (1912-1954) foi um matemático britânico, pioneiro da computação e considerado o pai da ciência computacional e da inteligência artificial.

Depois de formado, empreendeu estudos para criar uma máquina automatizada, que materializasse a lógica humana e solucionasse qualquer cálculo representado no formato de um algoritmo, que seriam exibidos no formato de instruções a serem processadas de forma mecânica, dentro da própria máquina. A “Máquina de Turing” se tornou um protótipo dos computadores modernos. Instruções sobre como programar um simulador da Máquina de Turing podem ser obtidas no site <http://ironphoenix.org/tm/> (acessado em 19/06/2021).

Trabalhou como funcionário do *Governmental Code and Cypher School* e entre 1940 e 1941, e desenvolveu uma máquina capaz de decifrar o “Enigma”, código utilizado pelos nazistas, durante a Segunda Guerra Mundial, dando assim aos aliados uma vantagem que permitiu derrotar mais depressa a Alemanha.

Depois da guerra, trabalhou no Laboratório Nacional de Física do Reino Unido onde pesquisou e trabalhou no projeto para o programa de armazenamento de dados, o ACE. Criou o Manchester 1, o primeiro computador com as diretrizes parecidas com as de hoje. Interessou-se também por química, quando passou um período trabalhando nos laboratórios da Bell, nos Estados Unidos.

1.5 Homem x Máquina

É uma resposta prática do quanto a área de IA avançou veio em 1997, quando a máquina derrotou o homem em um jogo de xadrez. O campeão soviético Garry Kasparov foi derrotado em uma das rodadas pelo computador Deep Blue (figura 2), da IBM, em partidas que repercutiram ao redor do mundo. O Deep Blue adotava um método de cálculo via força bruta que analisava possibilidades, previa respostas e sugeria o melhor movimento.



Figura 2: Computador Deep Blue da IBM (Google Imagens).

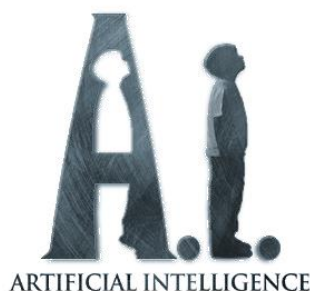
1.6 Inteligência Artificial retratada em filmes de Hollywood

A inteligência artificial pauta projetos de grandes empresas de tecnologia, debates entre cientistas e, também, uma forte leva de produções de Hollywood. O tema é tão inspirador que Hollywood nunca deixou de retratar o tema em seus filmes de ficção científica. Desde *Metrópolis*, filme mudo de 1927, temos produções com robôs, computadores e programas que agem para o nosso bem ou em busca da nossa destruição. A vontade da máquina em se tornar humana, de poder manifestar sentimentos até então humanos, foi retratada em filmes como “2001: Uma Odisseia no Espaço”, “O Homem Bicentenário”, *Blade Runner: O Caçador de Androides*, “A.I. (Inteligência Artificial)” e “Eu Robô” (figura 3).

Homem Bicentenário (1999)



**Artificial Intelligence
(2001)**



Eu Robô (2004)



Figura 3: Filmes de Hollywood que retratam a Inteligência Artificial (Google Imagens).

1.7 Paradigmas de pesquisa em IA

A Inteligência Artificial aborda atualmente um conjunto de paradigmas de pesquisa (algo que está relacionado a determinadas crenças e pressupostos que temos sobre a realidade, sobre como as coisas são – ontologia - e sobre a forma como acreditamos que o conhecimento humano é construído - epistemologia). Estes paradigmas de IA envolvem (figura 4):

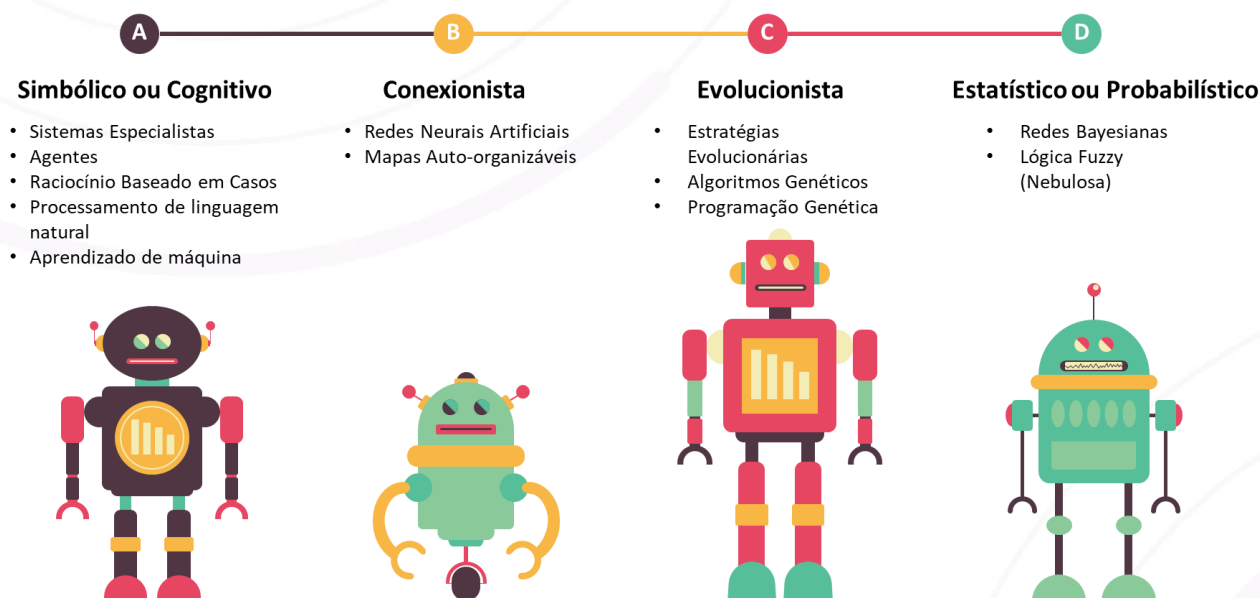


Figura 4: Paradigmas de Inteligência Artificial.

Saiba mais

imagine a seguinte cena:

Ao fim de um exaustivo dia de trabalho, uma executiva ativa um aplicativo no celular que avisa ao computador de bordo de seu carro que está de saída. Quando ela deixa o escritório, o automóvel a espera na entrada do prédio e, automaticamente, abre a porta para a dona, que é reconhecida pela máquina assim que se aproxima. Dentro do veículo, uma voz computadorizada pergunta: “Para casa?”. Depois da confirmação, o carro é guiado pela inteligência artificial (IA) de um software interno, enquanto a executiva responde a e-mails sem seu tablete, desatenta ao tráfego, intenso àquela hora da tarde. Já próximo da residência, o computador de bordo contata outra IA, a do assistente pessoal virtual que administra a rotina doméstica. “A geladeira está vazia. Quer que eu peça uma pizza?”, pergunta a máquina caseira. Após ouvir um “sim”, a IA providencia a demanda, já sabendo qual é a cobertura preferida de quem confirmou o pedido, do marido e dos filhos. Ao chegar em casa, nossa protagonista é surpreendida: acabou a ração do cachorro. Ela, então, se volta para a IA: “Pode encomendar para chegar hoje?”. A resposta: “Já fiz isso pela manhã. O pacote deve ser entregue por um drone em dez minutos”.

Todas as tecnologias descritas acima existem e muitas delas possuem versões comerciais.



Referências

COPPIN, B. Inteligência Artificial. 1ª Edição. São Paulo: LTC, 2010.

MCCARTHY, John. What is AI? Stanford University, Stanford, Estados Unidos, 12 nov. 2007.

SOBRAL, Osvaldo J. Inteligência Humana: Concepções e Possibilidades. Revista Científica Facmais. Inhumas, volume III, número I, 1º Semestre 2013, ISSN 2238-8427.

2. GENTES INTELIGENTES

2.1 Agentes

Um agente é tudo o que pode ser considerado capaz de perceber seu ambiente por meio de sensores (câmeras, microfone, teclado, finger etc.) e de agir sobre esse ambiente por intermédio de atuadores (vídeo, alto-falante, impressora, braços, ftp etc.) conforme exibido na figura 5 abaixo.

Um agente humano tem olhos, ouvidos e outros órgãos como sensores, e tem mãos, pernas, boca e outras partes do corpo que servem como atuadores. Um agente robótico poderia ter câmeras e detectores da faixa de infravermelho funcionando como sensores e vários motores como atuadores.

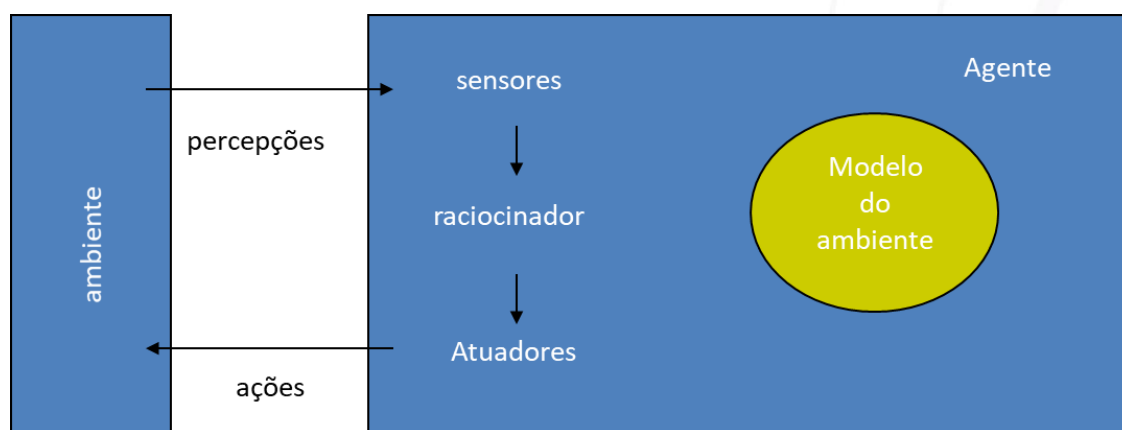


Figura 5: Arquitetura de um Agente.

Wooldridge & Jennings afirmam que agentes são sistemas que apresentam um comportamento determinado por um processo de raciocínio baseado na representação de suas atitudes, tais como crenças, comprometimentos e desejos.

Segundo Adja F. Andrade, um sistema de agentes é semelhante a uma sociedade de seres, que resolvem problemas e trocam informações. Trata-se de um sistema aberto, onde qualquer agente pode entrar ou deixar a sociedade.

Para GIESE (1998), as partes componentes, assim como o funcionamento da arquitetura básica de um agente autônomo é assim descrita (figura 6):

Setas: simbolizam o fluxo de dados;

Sensores: recebem informações do ambiente e providenciam dados para o mecanismo de inferência;

Mecanismo de inferência: é o cérebro do agente inteligente. Quando notificado de algum evento, o mecanismo de inferência opera sobre os conjuntos de regras e execução de raciocínio simbólico complexo para determinar como reagir ao evento e qual a ação executar;

Base de conhecimento: é o local onde o agente armazena seu conhecimento;

Atuadores: são responsáveis pela execução das ações do agente sobre o ambiente.

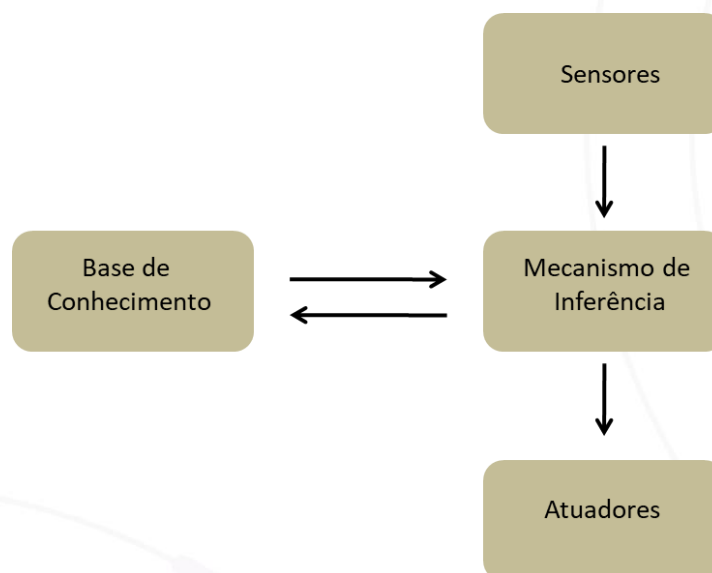


Figura 6: Componentes de um Agente.

O ambiente dos agentes pode ser: físico: robôs; software: softbots (robôs de software).

Realidade virtual (simulação do ambiente físico); softbots e avatares.

2.1.1 Propriedades dos agentes

Acessível: quando os sensores do agente podem obter o estado completo do ambiente. Também dizemos que o ambiente é efetivamente acessível se todas as informações necessárias para se tomar uma decisão ótima estão disponíveis aos sensores do agente;

Determinista: o próximo estado do ambiente pode ser completamente determinado pelo estado atual e as ações selecionadas pelo agente;

Estático: o ambiente não muda enquanto o agente está definindo a próxima ação a ser realizada;

Episódico: a experiência do agente é dividida em episódios. Cada episódio consiste na percepção e ação por parte do agente. Cada episódio não depende das ações que ocorreram em episódios prévios;

Discreto: quando existe um número distinto e claramente definido de percepções e ações em cada episódio;

Contínuo: quando não existe um número distinto e claramente definido de percepções e ações em cada episódio.

2.1.2 Características dos agentes

Autonomia: capacidade de agir sem a necessidade de intervenção humana, balizados em seus próprios princípios;

Mobilidade: capacidade de se mover através de uma rede de computadores, manter-se estáticos ou residentes na máquina do cliente temporariamente;

Cooperação: capacidade dos agentes de trabalharem de forma colaborativa e em conjunto com o objetivo de concluírem tarefas em comum (ambiente multiagente);

Comunicabilidade (habilidade social): capacidade do agente de se comunicar com outros agentes ou pessoas;

Aprendizagem: capacidade de aprender, avaliando as condições ambientes e tomando decisões quando a ação mais correta;

Reatividade: habilidade de reação a mudanças no ambiente do agente;

Proatividade: capacidade de apresentar um comportamento independente, ou seja, mostrar iniciativa em relação ao cumprimento de determinados objetivos;

2.1.3 *Nível de inteligência dos agentes*

De acordo com River (apud SOUZA, 1996), as aplicações com agentes apresentam diferentes níveis de inteligência e podem ser classificados como:

Baixo: neste nível, os softwares agentes desempenham tarefas rotineiras, disparadas por eventos externos. Estes agentes não executam redes de regras complexas, não se adaptam a mudanças, ou seja, não alteram suas regras;

Médio: estes agentes utilizam uma base de conhecimento para desenvolver raciocínio em eventos monitorados. Podem adaptar-se a mudanças de condições na base de conhecimento e manipular as novas condições, porém, normalmente não desenvolvem novas regras para tratar as mudanças;

Alto: neste nível de inteligência, os softwares de agentes utilizam tanto aprendizado quando raciocínio na base de conhecimento. Aprendem com o comportamento do usuário, adquirem novos conhecimentos e desenvolvem novas regras, com o tempo podem adaptar-se a mudanças de condições.

2.1.4 *Tarefas executadas pelos agentes*

Para Wooldrige & Jennings (apud SOUZA, 1998), existem três classes distintas de agentes segundo a tarefa que executam, são elas:

Gopher: são considerados agentes que executam as tarefas mais simples, baseando-se em suposição e regras pré-especificadas. Por exemplo, o agente pode avisar o usuário que ele possui uma reunião marcada para sexta-feira às 14h;

Prestadores de serviço: são agentes que executam tarefas bem definidas, quando requisitas pelo usuário. Estes agentes podem organizar uma reunião (negociar datas e horários com os participantes);

Proativo/preditivo: são agentes que desempenham as tarefas mais complexas, apresentando serviços ou informações voluntariamente para o usuário, sem ser explicitamente solicitado, sempre que julgar apropriado. Como por exemplo, um agente pode monitorar novos grupos na Internet e retornar discussões que ele acredita serem de interesse do usuário.

2.1.5 *Aquisição da inteligência*

De acordo com Nwama (apud GIESE, 1998), os agentes podem ser classificados quanto à aquisição de inteligência em:

Reflexivos/reativos: são agentes simples, possuem um mapeamento de situações e respostas associadas. Quando um estado ambiente ocorre, o agente executa a ação correspondente. Eles atuam utilizando um tipo de comportamento estímulo/resposta, respondendo para o estado presente do ambiente no qual eles estão inseridos.

Deliberativos: derivam do paradigma do pensamento, onde os agentes possuem um modelo simbólico e de raciocínio interno e dedicam-se a planejar e negociar tarefas para alcançar coordenação com outros agentes. Eles preveem que ações serão necessárias para alcançar o objetivo, interpretando o modelo para, então, executá-las.

2.1.6 *Tipologia dos agentes*

Com base nas classes de agentes (Gopher, Prestadores de Serviço e Preditivos) podemos identificar alguns tipos de agentes:

Colaborativos: enfatizam a autonomia e cooperação (com outros agentes) para executar suas próprias tarefas. Eles podem aprender, mas não é dada muita ênfase a esta característica em suas operações. Geralmente, agentes colaborativos podem negociar;

De interface: enfatizam a autonomia e aprendizado para executar suas tarefas. Interagem com o usuário, recebendo especificações e fornecendo resultados;

Móveis: são programas escritos especificamente em uma linguagem *script*, o qual pode ser despachado de um computador cliente e transportado para um computador servidor remoto para execução;

De Informação: exercitam o papel de gerenciar, manipular e colecionar informações de diversas fontes distribuídas. Não existe um padrão para suas operações, podendo ter características variadas: estáticos ou móveis, não-cooperativos ou sociais e aprender ou não;

Híbridos: são constituídos pela combinação de duas ou mais filosofias de agentes. Adotados para tentar maximizar a potencialidade e minimizar as deficiências das várias filosofias;

Heterogêneos: conjunto de dois ou mais agentes, os quais pertencem a duas ou mais classes de agentes inteligentes diferentes. Um sistema de agente heterogêneo pode também conter um ou mais agentes híbridos;

Inteligentes: aspiração dos pesquisadores ao invés da realidade.

2.1.7 Sistemas multiagentes

Um sistema multiagente (SMA) é composto por um conjunto de agentes que interagem em um ambiente comum. Exemplos: Jogos de computador, Interpretação de linguagem natural, Simulação, Assistentes pessoais, Comércio eletrônico, Tutores inteligentes, Controle de redes de energia elétrica.

De acordo com a sua complexidade a arquitetura de um SMA pode ser classificada em três grupos:

- a) Arquitetura simples: quando é composta por um único e simples agente;

- b) Arquitetura moderada: quando é composta por agentes que realizam as mesmas tarefas, mas possuem diferentes usuários e podem residir em máquinas diferentes (Fig. 1);
- c) Arquitetura complexa: quando é composta por diferentes tipos de agentes (Fig. 2), cada um com certa autonomia, podendo cooperar e estar em diferentes plataformas (KNAPIK; JOHNSON, 1998).

Em um SMA uma política de cooperação é necessária, uma vez que Agentes expressam as suas necessidades a outros Agentes a fim de realizar uma determinada tarefa. O processo de cooperação pode ocorrer de duas formas: partilhada de tarefas, Agentes auxiliam outros Agentes em uma determinada tarefa; partilha de resultados, os Agentes disponibilizam suas informações para a comunidade (GIRARDI, 2004). De acordo com a sua cooperação a arquitetura de um SMA por ser classificada em três grupos:

- a) Arquitetura Quadro-Negro: os Agentes não se comunicam de maneira direta, mas através de um quadro-negro (Fig. 3). O quadro-negro é uma estrutura de dados dividida em regiões ou níveis para facilitar a busca de informações. Em SMA funciona como um repositório de mensagens de perguntas e respostas, onde um Agente deposita uma mensagem de pergunta e espera que outro Agente colete esta mensagem, processe e deposite uma mensagem de resposta. Estas mensagens podem ser interpretadas como mensagens em que Agentes solicitam recursos de outros Agentes. Não é uma arquitetura viável para sistemas de tempo real;
- b) Arquitetura de troca de mensagens: os Agentes se comunicam diretamente por troca de mensagens assíncronas. Nesta arquitetura os Agentes precisam saber todos os nomes e endereços de todos os Agentes presentes no sistema. Estas mensagens precisam ser protocoladas para que possam ser encaminhadas e compreendidas pelos Agentes;
- c) Arquitetura Federativa: quando o número de Agentes no sistema é grande, mensagens de broadcasting levarão muito tempo para serem processadas. Por isto, os Agentes são estruturados em grupos ou federações onde existem Agentes facilitadores em cada grupo para receber e encaminhar mensagens entre Agentes remetentes e destinatários.

A Coordenação entre Agentes é referente à maneira como estarão organizados, para alcançar seus objetivos de sistema. Existem dois mecanismos de coordenação:

Mecanismo mestre-escravo: neste mecanismo existem duas classes de Agentes. Agentes mestres, ou gerentes, que distribuem tarefas para Agentes escravos, ou trabalhadores, e ficam à espera de resultados. Os Agentes escravos executam tarefas. Podem existir Agentes facilitadores se, os demais Agentes estiverem distribuídos em grupos;

Mecanismo de mercado: neste mecanismo os Agentes estão organizados em um mesmo nível e todos conhecem as tarefas que todos podem desempenhar. A vantagem é a diminuição da quantidade de mensagens trocadas, pois todos os Agentes se conhecem (PARAISO, 1997).

A comunicação é uma importante característica que os Agentes inteligentes possuem para atingir seus objetivos. Assim, foi definida uma Linguagem de Comunicação entre Agentes (ACL - Agent Communication Language) (FININ; LABROU; MAYFIELD, 1993) que foi dividida em:

- a) Vocabulário: consiste em um dicionário de todos os conceitos utilizados no domínio dos Agentes. Também conhecido como ontologia;
- b) Linguagem Interna: trata-se de linguagem de programação baseada em lógica de primeira ordem, para codificação de dados simples, regras, restrições e expressões. Também conhecida como linguagem KIF (*Knowledge Interchange Format*);
- c) Linguagem Externa: é uma camada linguística que encapsula estruturas KIF, para comunicação mais eficiente. Também conhecida como KQML (*Knowledge Query and Manipulation Language*).

2.2 Limitação dos Agentes

De acordo com Wooldridge & Jennings (apud YEPES, 2002), o enfoque em agentes cria uma quantidade de problemas comuns a todas as aplicações baseadas em agentes:

Nenhum controlador total do sistema: uma solução baseada em agentes pode não ser apropriada para os domínios em que os vínculos globais têm que ser mantidos, nos domínios onde uma resposta tempo real deve ser garantida, ou nos domínios em que os becos sem saída ou os travamentos devem ser evitados;

Nenhuma perspectiva global: as ações dos agentes, por definição, são determinadas pelo estado local dos agentes. Entretanto, desde que em quase nenhum sistema realista do agente, o conhecimento global completo é uma possibilidade, isto pode significar que os agentes fazem decisões globais sub-ótimas;

Confiança e delegação: para que os usuários estejam de acordo com a ideia de delegar tarefas aos agentes, devem primeiramente confiar neles. Os usuários têm que adquirir confiança nos agentes que trabalham em seu interesse, e este processo pode levar bastante tempo. Deve-se atentar para que o agente não exceda a sua autoridade com relação ao usuário ou organização; um agente deve saber suas limitações.

2.3 Aplicações

Diversas são as aplicações para agentes conforme quadro abaixo:

Tipo de agente	Medida de Desempenho	Ambiente	Sensores	Atuadores
Sistemas de diagnóstico médico	Paciente saudável, minimizar custos, processos judiciais	Paciente, hospital, equipe	Entrada pelo teclado para sintomas, descobertas, respostas do paciente	Exibir perguntas, testes, diagnósticos, tratamentos, indicações
Sistema de análise de imagens de satélite	Definição correta da categoria da imagem	Link de transmissão de satélite em órbita	Arrays de pixels em cores	Exibir a categorização da cena
Robô de seleção de peças	Porcentagem de peças em bandejas corretas	Correia transportadora com peças; bandejas	Câmera, sensores angulares articulados	Braço e mão articulados
Controlador de refinaria	Maximizar pureza, rendimento, segurança	Refinaria, operadores	Sensores de temperatura, pressão, produtos químicos	Válvulas, bombas, aquecedores, mostradores
Instrutor de inglês interativo	Maximizar nota de aluno em teste	Conjunto de alunos, testes de agência	Entrada pelo teclado	Exibir exercícios, sugestões, correções

As aplicações com agentes podem envolver ainda:

Sistemas de Administração de Rede: podem ajudar a filtrar e a executar ações automáticas a um nível mais alto de abstração, e podem até mesmo serem usados para descobrir e reagir a padrões de comportamento de sistema;

Acesso/Administração móvel: podem processar dados pela sua fonte e enviar somente respostas compactadas para o usuário, em vez de sobrecarregar a rede com grandes quantias de dados não processados ;

Mail e Messaging: podem facilitar funções permitindo que regras de manuseio de correio sejam especificadas previamente, e deixando para os agentes operarem em nome do usuário de acordo com essas regras;

Comércio eletrônico: podem "ir às compras" para um usuário, levar especificações e devolver com recomendações de compras aqueles que alcançarem essas especificações. Eles podem agir como "ajudantes de compras" para vendedores provendo produto ou serviço de conselho de vendas, e eles podem ajudar a localizar e reparar os problemas de clientes.

2.4 Estudo de caso

Todo ambiente de Datawarehouse enfrenta o obstáculo da sobrecarga de informações. Portanto, a essência do sucesso no ambiente de DW consiste na capacidade de monitorar e gerenciar o fluxo de dados. Segundo A. L. Gonçalves, citando W. H. Inmon, o monitoramento dos dados no ambiente de um DW envolve:

- a. Número de registro que existem no DW;
- b. Calcular e armazenar os padrões de crescimento dos registros;
- c. Mensurar perfis de dados;
- d. Calcular e armazenar os padrões de crescimento desses perfis de dados;
- e. Verificar a validade dos dados que são carregados, garantindo muita qualidade nos dados do DW.

Para efetuar eficientemente o monitoramento e o gerenciamento do fluxo de dados no DW, podem ser utilizados agentes autônomos responsáveis pela carga, sintetização, envelhecimento e particionamento dos dados.



Saiba mais

Um datawarehouse é um tipo de sistema de gerenciamento de dados projetado para fornecer suporte às atividades de business intelligence (BI), especialmente a análise avançada. Os datawarehouses destinam-se exclusivamente a realizar consultas e análises avançadas e geralmente contêm grandes quantidades de dados históricos. Os dados em um datawarehouse geralmente são derivados de uma ampla variedade de fontes, como arquivos de log de aplicativos e aplicativos de transações. Este banco de dados centraliza e consolida grandes quantidades de dados de várias fontes. Seus recursos analíticos permitem que as organizações obtenham informações de negócios úteis de seus dados para melhorar a tomada de decisões. Com o tempo, cria-se um registro histórico que pode ser inestimável para cientistas de dados e analistas de negócios.

2.5 Linguagens de programação para Agentes

Várias são as linguagens de programação que podem ser empregadas para desenvolvimento de agentes de software:

Java: linguagem de programação similar em sintaxe ao C++ (W3C, 1996). Conta com o Java Agent Template para a construção de agentes de software que podem se comunicar em uma rede P2P distribuídos pela Internet;

LALO (Langage d'Agents Logiciel Objet): linguagem de programação orientada a agentes e um ambiente para desenvolvimento de sistemas multiagentes. Um programa escrito em LALO é traduzido em código fonte C++, para depois ser compilado (FININ, 1997);

Obliq: é uma linguagem interpretada e orientada a objetos com suporte para computação móvel, tal como agente móvel (ITA, 1997a);

Phantom: é uma linguagem interpretada projetada para aplicações distribuídas, interativas e em larga escala, tais como sistemas de conferência distribuída, jogos em rede, e ferramentas de trabalho colaborativo (FININ, 1997);

3APL: (*Artificial Autonomous Agents Programming Language*): Ferramenta experimental de linguagem de programação para o desenvolvimento, implementação, monitoramento e depuração de agentes.

2.6 JADE (Java Agent Development Framework)

JADE (Java Agent DEvelopment Framework) é uma estrutura de software totalmente implementada na linguagem Java. Ele simplifica a implementação de sistemas multiagentes por meio de um middleware que atende às especificações do FIPA (*Foundation for Intelligent Physical Agent*) e por meio de um conjunto de ferramentas gráficas que suportam as fases de depuração e implantação. Um sistema baseado em JADE pode ser distribuído entre máquinas (que nem precisam compartilhar o mesmo sistema operacional) e a configuração pode ser controlada por meio de uma GUI remota. A configuração pode ser alterada até mesmo em tempo de execução, movendo os agentes de uma máquina para outra, como e quando necessário. O JADE é totalmente implementado na linguagem Java e o requisito mínimo do sistema é a versão 5 do JAVA (o ambiente de tempo de execução ou JDK).

Além da abstração do agente, JADE fornece um modelo de composição e execução de tarefa simples, mas poderoso, comunicação de um agente ponto a ponto com base no paradigma de passagem de mensagem assíncrona, um serviço de páginas amarelas com suporte para mecanismo de descoberta de assinatura de publicação e muitos outros recursos avançados que facilitam o desenvolvimento de um sistema distribuído.

A documentação técnica desta plataforma pode ser obtida em <https://jade.tilab.com/>.

3. SISTEMAS ESPECIALISTAS

3.1 Conceito de SE

Os Sistemas Especialistas (SE) podem ser caracterizados como sistemas que reproduzem o conhecimento de um especialista adquirido ao longo dos anos de trabalho (KANDELL, 1992). São sistemas que solucionam problemas que são resolvíveis apenas por pessoas especialistas (que acumularam conhecimento exigido) na resolução destes problemas (FEI, 1977).

Também são identificados como programas de computador que tentam resolver problemas que os seres humanos resolveriam emulando o raciocínio de um especialista, aplicando conhecimentos específicos e inferências são ditos Sistemas Especialistas (ROL, 1988).

O desenvolvimento de um SE incorpora, para além de uma vertente técnica, uma vertente humana complexa, visto haver a necessidade do estabelecimento de um relacionamento de confiança entre quem especifica e desenvolve o sistema e quem possui o conhecimento.

Um Sistema Especialista é aquele que:

- a) É projetado e desenvolvido para atender a uma aplicação determinada e limitada do conhecimento humano;
- b) É capaz de emitir uma decisão, apoiado em conhecimento justificado, a partir de uma base de informações, tal qual um especialista de determinada área do conhecimento humano;
- c) Além de inferir conclusões, deve ter capacidade de aprender novos conhecimentos (melhorando o seu desempenho de raciocínio, e a qualidade de suas decisões).

Um Sistema Especialista é apropriado quando:

- a) Existe escassez de especialistas;

- b) A informação disponível é pobre, parcial ou incompleta, ou quando o problema é incompletamente definido;
- c) O conhecimento é baseado em regras que somente podem ser aprendidas através da experiência;
- d) O problema está sujeito a rápidas mudanças de regras e códigos.

3.2 Atributos dos sistemas especialistas

Atributos dos Sistemas Especialistas segundo CASTILHO, 1998.

- a) O sistema tem separado o conhecimento específico do especialista e a metodologia de solução de problemas;
- b) A transferência interativa de conhecimento pode minimizar o tempo necessário para transferir o conhecimento do especialista para uma base de conhecimento;
- c) A estratégia de controle pode ser, simples e transparente, ao usuário, isto é, o usuário capaz de compreender e prever os efeitos de adição, alteração e deleção de itens na base de conhecimento.

3.3 Sistemas especialistas x conhecimento

Os Sistemas Especialistas são construídos a partir da experiência e do conhecimento de um especialista humano (adquirido ao longo dos anos), o qual fornecerá a base das informações necessárias.

Os especialistas têm a capacidade de resolver problemas difíceis, explicar os resultados obtidos, aprender, reestruturar o conhecimento e determinar as suas características relevantes, porém muitas vezes têm dificuldade em explicar o seu modo de raciocínio de uma maneira analítica (FERNANDES, 1996).

3.4 Sistemas especialistas x sistemas convencionais

A diferença básica entre um Sistema Convencional (SC) e um Sistema Especialista (SE) reside no fato de que o primeiro é baseado em um algoritmo, processa um conjunto de dados e instruções de forma repetitiva para emitir determinados resultados enquanto um

SE trabalha com situações e problemas para os quais não existe uma solução convencional.

Um sistema convencional, fundamentalmente, é baseado em um algoritmo, emite um resultado correto ao final do seu processamento, e processa um volume de dados de maneira repetitiva. Já um sistema especialista, é baseado em uma busca heurística, trabalha com problemas para os quais não existe uma solução convencional organizada de forma algorítmica disponível ou é muito demorada.

Os Sistemas Especialistas possuem algumas facilidades em relação aos Sistemas Convencionais:

- a) Possibilidade para construção de regras
- b) Tomada lógica de decisões sob imprecisão ou na ausência de informações
- c) Nas aplicações (programas) tradicionais, o método de busca é baseado no conhecimento e nas regras codificadas previamente, havendo a necessidade de reescrita do código no caso do surgimento de novos conhecimentos. Já os SE's podem recuperar novos fatos e regras e usá-los sem modificar a estratégia de busca (SABBATINI, 1993).

3.5 Engenharia do conhecimento

O processo de construção de um Sistema Especialista é chamado de Engenharia do Conhecimento e envolve a interação entre:

Especialista humano: toma decisões sobre determinado assunto a partir de fatos que encontra e hipóteses que formula, buscando em sua memória um conhecimento prévio armazenado durante anos, no período de sua formação e no decorrer de sua vida profissional, sobre esses fatos e hipóteses. E o faz de acordo com sua experiência, isto é, com seu conhecimento acumulado sobre o assunto e, com esses fatos e hipóteses, emite a decisão (SCHWABE e CARVALHO, 1987).

Para tomar uma decisão sobre um determinado assunto, um especialista humano o faz a partir de fatos que encontra; de hipóteses que formula, buscando em sua memória um conhecimento prévio armazenado durante anos, no período de sua formação e no

decorrer de sua vida profissional, sobre esses fatos e hipóteses; e o faz de acordo com a sua experiência e o seu conhecimento acumulado sobre o assunto e, com esses fatos e hipóteses.

Durante o processo de raciocínio, o especialista humano vai verificando qual a importância dos fatos que encontra, comparando-os com as informações já contidas no seu conhecimento acumulado sobre esses fatos e hipóteses. Com esse processo de raciocínio, o especialista humano pode não chegar a uma decisão se os fatos de que dispõe para aplicar o seu conhecimento prévio não forem suficientes.

O construtor do Sistema: chamado também de Engenheiro do Conhecimento, cujo papel é o de “extrair” procedimentos, estratégias e de um especialista humano para a solução de determinado problema (WATERMAN, 1986).

Um Engenheiro do conhecimento entrevista especialistas em um determinado domínio e tenta incorporar seu conhecimento em um programa de computador para realizar alguma tarefa. O quão bem isso funciona depende se os mecanismos intelectuais necessários para a tarefa estão dentro do estado atual da IA. Quando isso não aconteceu, houve muitos resultados decepcionantes. Um dos primeiros sistemas especialistas foi o MYCIN em 1974, que diagnosticou infecções bacterianas do sangue e sugeriu tratamentos. Ele se saiu melhor do que estudantes de medicina ou médicos praticantes, desde que suas limitações fossem observadas. Ou seja, sua ontologia incluía bactérias, sintomas e tratamentos e não incluía pacientes, médicos, hospitais, morte, recuperação e eventos que ocorriam no tempo. Suas interações dependiam da consideração de um único paciente. Como os especialistas consultados pelos engenheiros do conhecimento sabiam sobre pacientes, médicos, morte, recuperação etc., é claro que os engenheiros do conhecimento forçaram o que os especialistas disseram a eles em uma estrutura predeterminada. No estado atual da IA, isso tem que ser verdade.

O Engenheiro do conhecimento é responsável pelo processo de aquisição do conhecimento; poderá ser o implementador do Sistema; deve ser capaz de lidar com incertezas sobre o conhecimento em causa; deve conhecer os métodos de representação e estruturação do conhecimento; deve ser alguém com capacidades humanas e sociais adequadas à motivação do especialista, e deve saber quais as necessidades dos usuários finais do sistema.

Usuário: é a pessoa que vai determinar as principais restrições de projeto. As habilidades e necessidades do usuário devem ser consideradas ao longo de todo o ciclo do projeto, pois se o usuário não ficar satisfeito, o esforço de desenvolvimento será um desperdício.

3.6 Estrutura de um sistema especialista

Os Sistemas Especialistas apresentam 5 componentes básicos (figura 7):

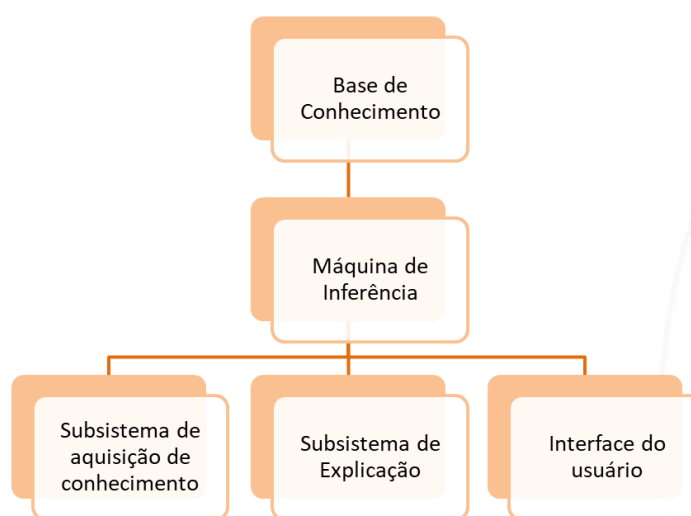


Figura 7: Estrutura básica de um Sistema Especialista.

3.6.1 Base de conhecimento

A base de Conhecimento é o local onde estão armazenados os fatos e as regras. Alguns Sistemas Especialistas utilizam regras como base para seu processamento e operação e por isso são chamados de Sistemas Baseados em Regras, outros representam o conhecimento através de Redes Semânticas ou frames. O conhecimento armazenado nesta Base pode facilmente ser modificado e quando isto ocorre basta a inserção, modificação ou exclusão de regras antigas. Uma das formas de representação do conhecimento em um Sistema Especialista é através regras de produção, ou simplesmente regras que são definidas pelo par: condição – ação, na forma de regras do tipo “IF – THEN”. O resultado do teste da condição depende do estado atual da Base de Conhecimento.

O emprego de regras traz consigo um conjunto de vantagens: fácil de entender (forma natural do conhecimento); fácil de derivar inferências e explicações; fácil modificar e manter; fácil combinar com incertezas; são frequentemente independentes.



Exemplo de Regras de produção

REGRA 1

SE amanhã pode chover = Não
E tenho dinheiro suficiente = Sim
E tenho tempo suficiente = Sim
ENTÃO devo ir à praia = Sim

REGRA 2

SE o serviço de meteorologia disse que vai chover amanhã = Sim
ENTÃO amanhã pode chover = Não

REGRA 3

SE não vou sair hoje = Sim
E nenhuma emergência ocorrer = Sim
ENTÃO tenho dinheiro suficiente = Sim

REGRA 4

SE meu orientador passar trabalho extra = Sim
ENTÃO tenho tempo suficiente = Não

As regras podem ser validadas através do:

- a) **Forward Chaining** (Encadeamento para Frente), onde se partindo de um ponto inicial, chega-se a uma conclusão.

- b) **Backward Chaining** (Encadeamento para Trás), que se inicia com uma previsão (hipótese) e procura valores para confirmá-la.

Exemplo de “encadeamento para trás” (figura 8).

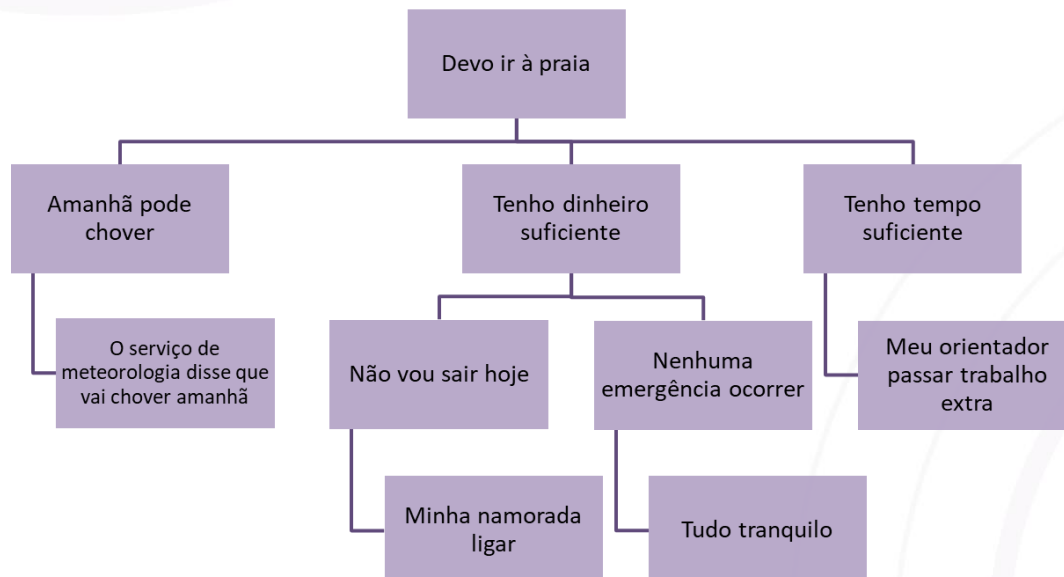


Figura 8: Exemplo de Sistema Especialista que emprega regra de “encadeamento para trás”.

3.6.1.1 Memória de trabalho

A memória de trabalho é parte de um sistema especialista que contém os fatos do problema que são descobertos durante a sessão de consulta e contém todas as informações sobre o problema que são fornecidas pelo usuário ou inferidas pelo sistema. Toda informação obtida durante uma consulta é frequentemente chamada de contexto da sessão.

3.6.2 Máquina de inferência

Também conhecido como mecanismo de inferência, este processo procura as respostas na Base de Conhecimento, encontrando as regras necessárias a serem avaliadas e ordena-as de forma lógica. É o processador em um sistema especialista que confronta os fatos contidos na memória de trabalho com os conhecimentos de domínio contidos na base de conhecimento para tirar conclusões sobre o problema.

Uma máquina de inferência opera como um “supervisor”, tomando decisões e julgamentos baseados em dados simbólicos contidos na Base de Conhecimento. Uma vez

iniciado o Sistema, cabe a Máquina de Inferência buscar na Base de Conhecimento, fatos e regras que serão comparados com as informações fornecidas pelos usuários e buscando “combinações”.

A máquina de inferência, a partir da análise de uma determinada regra, trabalha com um mecanismo e uma estratégia de inferência:

- a) **Se** as premissas estão contidas na memória de trabalho;
- b) **Então** aplica-se a regra, adicionando as conclusões memória de trabalho;
- c) **Senão** passa para a próxima regra;
- d) **Quando** se detecta que um objetivo foi atingido ou que mais nenhuma regra se aplica, o processo de raciocínio é encerrado.

Exemplo:

Os pais de Carlos são Paulo e Rose. Os pais de Sandra são Paulo e Rose.

SE uma pessoa do sexo masculino e uma pessoa do sexo feminino têm os mesmos pais, **ENTÃO** eles são irmãos. Esta dedução é conseguida através e um mecanismo chamado PODA que concentra uma regra guardada em nosso cérebro.

A parte da inteligência que nos permitiu deduzir que Carlos e Sandra são irmãos (novo fato) é chamada de mecanismo de inferência. Este é fundamental em nossa capacidade de aprender com a experiência porque nos permite gerar novos fatos a partir dos já existentes e, também, colabora para a detecção de falhas em nosso pensamento.

3.6.3 *Subsistema de aquisição do conhecimento*

Através deste mecanismo podemos inserir, alterar ou excluir conhecimentos armazenados na Base de Conhecimento.

3.6.4 *Subsistema de explicações*

Empregado para explicar aos usuários a linha de raciocínio que o Sistema Especialista empregou para alcançar determinada conclusão. Tais subsistemas são importantes para o “*debugging*” do sistema durante o seu desenvolvimento.

Um sistema especialista emite resposta utilizando três modos de técnicas diferentes:

- a) O sistema determina um universo onde a resposta poderá ser encontrada. Por exemplo: pode determinar, de maneira geral, onde o petróleo poderá ser encontrado, sem, no entanto, afirmar de maneira precisa em que regiões se encontrará o petróleo;
- b) O sistema é mais preciso e determina um resultado;
- c) O sistema não emite resultado nenhum, apenas interage com o profissional que o está usando.

3.6.5 Interface com usuário

Estabelece um meio de comunicação entre o usuário e o Sistema, podendo utilizar menus, perguntas e representações gráficas.

3.7 Benefícios da utilização de sistemas especialistas

- Velocidade na determinação dos problemas;
- A decisão está fundamentada em uma base de conhecimento;
- Segurança;
- Exige pequeno número de pessoas para interagir com o sistema;
- Estabilidade;
- Dependência decrescente de pessoal específico;
- Flexibilidade;
- Integração de ferramentas;
- Evita interpretação humana de regras operacionais.

3.8 Problemas enfrentados por sistemas especialistas

- **Fragilidade** - Como os Sistemas Especialistas somente têm acesso a conhecimento altamente específicos do seu domínio não possuem conhecimentos mais genéricos quando a necessidade surge;
- **Falta de metac conhecimento** - Geralmente não possuem conhecimentos sofisticados sobre sua própria operação, portanto não conseguem raciocinar sobre seu próprio escopo e restrições (dificuldade em explicitar o conhecimento

que utiliza). A aquisição do conhecimento continua sendo um dos maiores obstáculos a aplicação de tecnologia dos Sistemas Especialistas a novos domínios.

- **Validação** - A medição do desempenho de Sistemas Especialistas é muito difícil porque não sabemos quantificar o uso de conhecimento.

3.9 Tipos genéricos de emprego de sistemas especialistas

Os tipos genéricos de emprego de sistemas especialistas consideram as características do funcionamento destes:

- a) **Interpretação:** inferem descrições de situações a partir da observação de fatos, fazem a análise de dados e procuram determinar as relações e seus significados. Envolve sistemas que vão desde a compreensão de voz, compreensão de texto e compreensão de imagem;
- b) **Diagnósticos:** inferem mal funcionamentos de sistemas a partir de observações, prescrevem “remédios” para o mal funcionamento. São frequentemente utilizados por sistemas que também tratam a Interpretação.
- c) **Monitoramento:** aparecem muitas vezes associados ao diagnóstico. Trata-se, no fundo, de acompanhar a evolução de um sistema através de medições de grandezas que possam indicar o modo como o sistema evolui;
- d) **Previsão:** a partir da modelagem de dados do passado e do presente, este sistema permite uma determinada previsão do futuro;
- e) **Planejamento:** desenvolvem-se sistemas capazes de responder à questão de como gerar um plano para resolver um dado problema;
- f) **Projeto:** possui características parecidas com as do planejamento, sendo capaz de justificar a alternativa tomada para o projeto final, e de fazer uso desta para alternativas futuras;

- g) Depuração: possui mecanismos para fornecerem soluções para o mau funcionamento provocado por distorções de dados;
- h) Reparo: desenvolve e executa planos para administrar os reparos verificados na etapa de diagnóstico;
- i) Instrução: trata-se de proporcionar uma ferramenta de ensino e treino por computador;
- j) Controle: governa o comportamento geral de outros sistemas com o objetivo de “dar ordens” a um conjunto de entradas de modo a que um dispositivo ou sistema consiga cumprir adequadamente as suas funções.

3.10 Exemplos de sucesso de sistemas especialistas periciais

- a) MYCIN - Sistema Especialista mais conhecido com conhecimento sobre parâmetros de análises sanguíneas para diagnosticar automaticamente problemas relacionados com infecções bacterianas. A base de conhecimento do MYCIN desenvolvida ao longo de vários anos com conhecimento oriundo de vários especialistas (450 regras);
- b) DENDRAL - Primeiro Sistema Especialista que tratava espectros de massa e respostas magnéticas ao nível nuclear de modo a fornecer informação sobre a estrutura molecular de compostos desconhecidos, sendo desenvolvido em meados dos anos 60 e melhorado por vários investigadores;
- c) INTERNIST - Sistema Especialista aplicado à Medicina Interna, com foco no diagnóstico de problemas individuais. Capaz de combinar problemas separados que ocorram simultaneamente num mesmo paciente. Um dos sistemas especialistas mais completos no âmbito da medicina

3.11 Expectativas em relação a um sistema especialista

- a) Resolução de problemas para o domínio para o qual foi concebido;
- b) Facilidade de manutenção incremental da sua base de conhecimento;

- c) Conhecimento apresentado de forma atraente e legível (alguns geradores de sistemas especialistas permitem a inclusão do conhecimento numa língua quase natural);
- d) Desempenho semelhante a um especialista (capacidade de explicar como chegou a uma dada conclusão, porque qual razão não foi possível chegar a uma dada conclusão ou porque está a pôr uma dada questão ao usuário);
- e) Mecanismo de raciocínio eficiente em domínios nos quais a quantidade de conhecimento seja elevada;
- f) Interface que se adapte ao tipo de utilizador (especialista ou novato) e à situação em causa (normal ou crítica);
- g) Capacidade de efetuar raciocínios considerando o raciocínio com incertezas;
- h) Assistência nas fases de aquisição, estruturação e transferência de conhecimento, ou nas fases de verificação e validação do conhecimento.

3.12 Vantagens dos sistemas especialistas

- a) Explicação;
- b) Rapidez de resposta;
- c) Completude das respostas (estáveis, completas e não emocionais);
- d) Tutor inteligente;
- e) Base de dados inteligente;
- f) Boa confiabilidade;
- g) Flexibilidade;
- h) Capacidade de treinamento (estagiários em uma atividade, Pessoas sobre uma organização).

4. ALGORITMOS GENÉTICOS

4.1 Teoria da Evolução das Espécies

A Teoria da Evolução das Espécies é creditada ao naturalista britânico Charles Robert Darwin. Uma expedição científica a bordo do navio Beagle seria o despertar de suas ideias sobre a evolução. Essa expedição levaria cinco anos no qual ele navegou pela costa do Pacífico e pela América do Sul. Em todo lugar que parava, recolhia vários tipos de espécies de plantas e animais. Ao retornar à Inglaterra, Darwin iniciou seu estudo sobre a origem e evolução das espécies. Começou a surgir então a doutrina darwinista, ou teoria da seleção natural.

Em 1859 publica o livro "A Origem das espécies por meio da seleção natural" no qual expõe sua ideia sobre a evolução. Sua teoria dizia que as espécies geram descendentes a partir da reprodução. Tais descendentes gerados possuem variações genéticas, ou seja, possuem traços decorridos da hereditariedade, porém não idênticos aos pais. Num ambiente de recursos limitados, os organismos menos adaptados têm menos chance de sobreviver do que os mais adaptados. Sendo assim os mais adaptados acabam gerando mais descendentes e estes então são selecionados para aquele ambiente.

Na teoria Darwinista são identificados cinco pilares no qual ela foi elaborada. Esses cinco pilares são: Evolução dos seres vivos, Ancestral comum, Multiplicação das espécies, Gradualismo e Seleção natural. Nesta mesma teoria:

- a) Indivíduos com uma melhor adequação do seu fenótipo ao meio ambiente (fitness melhor) reproduzem mais;
- b) Ao reproduzirem mais, têm mais chances de passar seus genes para a próxima geração;
- c) Entretanto, graças aos operadores genéticos (recombinação e mutação) os cromossomos dos filhos não são exatamente iguais aos dos pais;
- d) Assim, eles podem evoluir e se adaptar cada vez mais ao meio ambiente que os cerca.

4.2 Conceitos básicos sobre Genética

Segundo o dicionário Aurélio, Genética é a Ciência que estuda a transmissão, de pais a filhos, dos caracteres de hereditariedade.

Segundo Darwin, indivíduos mais adaptados acabam gerando mais descendentes e estes então são selecionados para aquele ambiente. O que Darwin não sabia era que essa tal adaptação era originada da genética, ou seja, transmissões de informações entre pais e filhos.

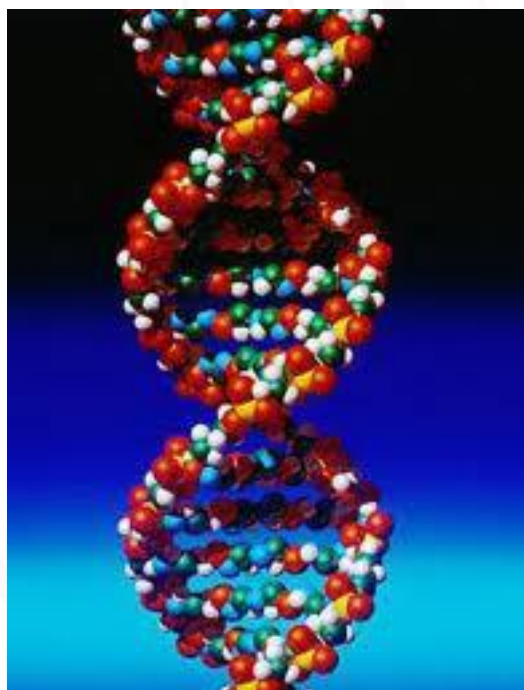
Foi a partir do século XX, que a genética conheceu o padre Gregor Mendel. Ele descobriu que o conceito de herança vem através de partículas (genes) e não pela mistura do sangue. Essa descoberta não só contribuiu para entender a genética quanto a evolução das espécies.



4.3 DNA

Cada indivíduo de uma espécie é formado por uma ou mais células. Cada célula possui informações que caracterizam cada indivíduo. Essas informações genéticas ficam armazenadas nas moléculas de DNA ou AND (ácido desoxirribonucléico). O DNA tem uma forma de duas faixas espiraladas, entre as quais existem as moléculas denominadas bases nitrogenadas que as ligam.

A molécula de DNA fica organizada e armazenada em estruturas denominadas de cromossomos que ficam no núcleo das células.



Os cromossomos estão sempre em pares, o que acaba variando para cada espécie. O cromossomo é constituído de genes, ou seja, segmento do DNA. Cada posição

específica do gene no DNA é denominada de **lócus** que controla algumas características específicas referente à hereditariedade.

As alterações ocorridas nos cromossomos, tanto em estrutura como em quantidade, podem ocasionar doenças como são os casos da síndrome de Down, síndrome de Turner e síndrome de Klinefelter.

Quando falamos na composição genética que determina as características de um indivíduo estamos falando de genótipo. O genótipo é uma característica fixa, ou seja, que não é alterada por fatores do ambiente.

Enquanto o genótipo refere-se à composição genética, o fenótipo refere-se às características físicas de um indivíduo. Por exemplo, duas pessoas com a mesma pigmentação de pele, uma delas se expõem a algumas horas de sol, logo, ao comparar a cor da pele das duas pessoas iremos notar que a que tomou sol está com pigmentação diferente. Na verdade, não é porque ela possui esse tom de pele e sim ela obteve este devido à exposição do sol.

Resumindo, de forma bem simplificada, o fenótipo de um indivíduo nada mais é que a composição do genótipo mais as interações do meio ambiente.

4.4 Hereditariedade e reprodução

Ao olhar para uma criança que possui olhos verdes, e falar esses olhos são do pai ou da mãe. Na verdade, estamos falando de hereditariedade, a característica olho claro que o pai ou a mãe possui foi passado ao seu filho por meio do material genético no momento da fecundação. Simplificando, hereditariedade é a transmissão das informações genéticas de pais para filhos durante a reprodução. (FONSECA, 2008).

Pode-se resumir reprodução como a função que faz com que os indivíduos produzem seus descendentes. Existem dois tipos de reprodução: Assexuada e Sexuada.

A reprodução assexuada é aquela que ocorre sem ser por gametas (material genético), ou seja, não ocorre a fecundação. Essa reprodução geralmente é encontrada em bactérias e plantas.

A reprodução sexuada é aquela que ocorre a fecundação, dois indivíduos geralmente de sexo diferentes, trocam seus materiais genéticos. Essa troca de material genético ocorre pela fusão dos gametas, ou o *crossover*. Essa reprodução geralmente é encontrada nos seres humanos (LINDEN, 2008).

4.5 Operadores genéticos - *crossover*

Como vimos anteriormente na reprodução sexuada, há troca de material genético entre dois indivíduos podendo assim gerar o *crossover*. Nesse processo inicialmente há a duplicação dos cromossomos, um pedaço de cada cromossomo é trocado com outro pedaço de outro cromossomo. São gerados assim quatro novos cromossomos que serão separados e enviados aos gametas (figura 9).

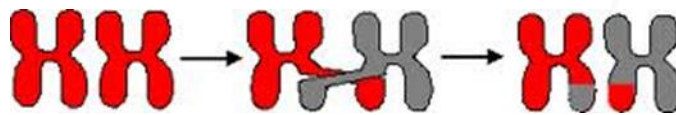


Figura 9: Exemplo de emprego do operador de “crossover”.

4.5.1 Operadores genéticos - *mutação*

No processo de reprodução, devido à sua complexidade, podem ocorrer alguns erros, denominados de mutação que ocorrem quando há um dano na sequência de nucleotídeo do DNA. E esses danos são irreparáveis e assim repassando na sua replicação. A outra forma de ocorrência da mutação acontecer é quando há modificação dos genes e os cromossomos sofrem alterações em seus números ou sua estrutura.



4.6 Algoritmos Evolucionários

Os Algoritmos evolucionários usam modelos computacionais dos processos naturais de evolução como uma ferramenta para resolver problemas. Há uma grande variedade de modelos computacionais. Em comum:

- a) Conceito de simulação da evolução das espécies
- b) Uso de operadores de seleção, mutação e reprodução
- c) Todos os processos dependem do "desempenho" dos indivíduos desta espécie dentro do "ambiente";
- d) Mantêm uma população de estruturas, denominadas indivíduos ou cromossomos
- e) Comportam-se de forma semelhante à evolução das espécies;
- f) A estas estruturas são aplicados os chamados operadores genéticos, como recombinação e mutação, entre outros;
- g) Cada indivíduo recebe uma avaliação que é uma quantificação da sua qualidade como solução do problema em questão;
- h) Baseado nesta avaliação serão aplicados os operadores genéticos de forma a simular a sobrevivência do mais apto;
- i) São extremamente dependentes de fatores estocásticos (probabilísticos), tanto na fase de inicialização da população quanto na fase de evolução (durante a seleção dos pais, principalmente);
- j) Seus resultados raramente sejam perfeitamente reprodutíveis;
- k) São heurísticas que não asseguram a obtenção do melhor resultado possível em todas as suas execuções.

4.7 Os Algoritmos Genéticos no contexto da IA

Os algoritmos genéticos são um ramo da computação evolucionária e fazem parte do universo das técnicas de busca conforme exposto na figura 12 abaixo.

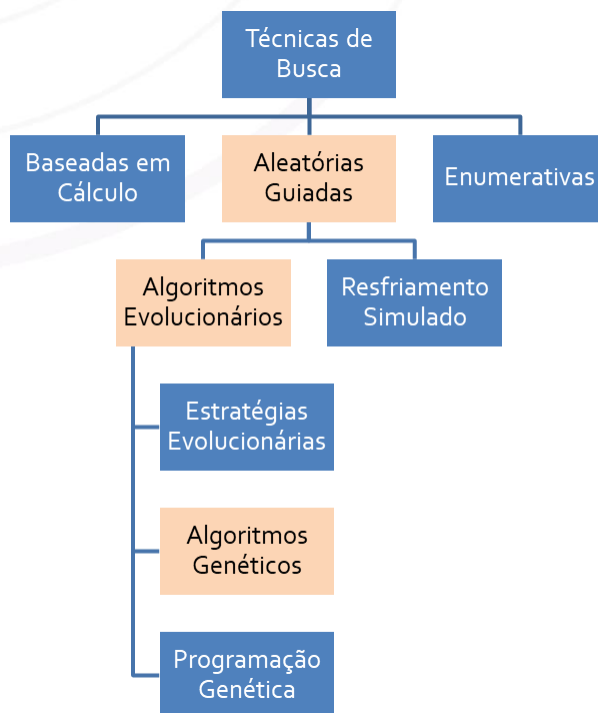


Figura 12: Os Algoritmos Genéticos no contexto da IA.

4.8 História dos Algoritmos Genéticos

A história dos algoritmos genéticos teve seu início a partir da década de 40 com cientistas que utilizavam a natureza para criar um ramo na inteligência artificial. Várias foram às contribuições para a história dos algoritmos genéticos, dentre eles podemos citar: Bledsoe, Bremmerman, Rechenberg. Porém em 1975 foi que o professor John Holland propõe a ideia de algoritmos genéticos.



Ele define como princípio os processos da evolução das espécies juntamente com o algoritmo computacional a fim de solucionar problemas que possuam um alto índice de dificuldade. A partir dos anos 80 a ideia sobre algoritmos genéticos começou a ser propagada dentro da comunidade científica.

4.9 Conceitos sobre Algoritmos Genéticos

Os algoritmos genéticos formam um ramo da Inteligência Artificial chamada de Computação Evolutiva. Inspirados no processo genético e evolutivo de organismos vivos,

simulam processos naturais e aplicando-os à solução de problemas reais, com base na ideia darwiniana de seleção.

O AG é indicado para a solução de problemas de otimização complexos que envolve um número elevado de variáveis e, conseqüentemente, espaços de soluções de dimensões elevadas. Além disso, em muitos casos em que outras estratégias de otimização falham na busca de uma solução, os AG's convergem. Os AG's são numericamente robustos, ou seja, não são sensíveis a erros de arredondamento no que se refere aos seus resultados. Trabalham com uma população de indivíduos, cada qual representando uma solução de determinado problema.

A cada indivíduo se associa a um grau de aptidão, o que determina sua capacidade de competir com os demais membros da população. Quanto maior a sua aptidão, maior a probabilidade do mesmo ser selecionado para se reproduzir, cruzando seu material genético com o de outro indivíduo selecionado de igual forma. Este cruzamento produzirá novos indivíduos – descendentes dos anteriores – os quais compartilham algumas características de seus pais.

Quanto menor for o grau de aptidão, menor será sua probabilidade de ser selecionado para reprodução e, conseqüentemente, que o material genético se propague por sucessivas gerações.

Dessa forma, se produz uma nova população de possíveis soluções, a qual substitui a anterior por possuir características mais promissoras em relação à população ótima.

Assim, ao longo das gerações, as boas características se propagam, favorecendo desta forma, a exploração das áreas mais promissoras do espaço de busca. Sendo os AGs bem projetados, as chances de se obter o ótimo global para o problema em questão são excelentes.

O poder dos AGs provém do fato de se tratar de uma técnica robusta, podendo manipular com êxito uma grande variedade de problemas provenientes das mais diferentes áreas, incluindo aqueles que os outros métodos encontram dificuldades para resolver.

Apesar de não garantir que os AGs encontrem a solução ótima do problema, existem evidências empíricas que respostas aceitáveis podem ser obtidas em um tempo real bastante razoável.

Basicamente, os AGs tratam problemas de otimização como um processo iterativo de busca da melhor solução dentro do espaço de possíveis respostas para o problema.

4.9.1 População

Quando trabalhamos com os AGs, o primeiro passo que devemos realizar é a definição de uma população inicial. Ela é geralmente formada por indivíduos que são escolhidos de forma aleatória. Um fato importante que devemos levar em consideração é o tamanho desta população. Populações muito pequenas impossibilitam a variedade genética. Populações muito grandes ocasionam um algoritmo lento.

Segundo Linden, não há uma definição da melhor população. Ele sugere adotar uma população cujo tamanho varie com o tempo, que haja variedade genética e esteja de acordo com a idade. Dentre a definição da população inicial existem técnicas que auxiliam na geração de novas gerações de indivíduos melhores. Como por exemplo, o Elitismo e *Steady state*. O Elitismo consiste na permanência dos melhores indivíduos para uma próxima geração. O conceito *Steady state* é baseado na forma que nós utilizamos para geração de novas gerações, ou seja, não é gerada uma nova população de uma só vez e sim os filhos são gerados um de cada vez substituindo assim os piores pais pelos novos filhos aos poucos.

4.9.2 Função de avaliação

Também conhecida como Função de Custo, a função de avaliação é aplicada após a identificação da população inicial. Seu objetivo é avaliar a qualidade em que um indivíduo da população age sobre o problema relacionado. Quanto melhor é a solução que um indivíduo utiliza, melhor será a sua avaliação.

A função de avaliação deve ser definida com os objetivos pelo qual o algoritmo genético está sendo aplicado de forma bem clara. Assim como as restrições e as punições para os indivíduos que as desrespeitam.

4.9.3 Tipos de Seleção

O método de seleção é aquele utilizado para selecionar os pais mais aptos para que possam gerar os seus filhos. Um dos métodos mais utilizados é o da roleta viciada.

A roleta viciada funciona da seguinte maneira: é criada uma roleta virtual, dividida em pedaços. Cada pedaço é de acordo com a avaliação dos indivíduos. A roleta é rodada e o indivíduo selecionado será aquele em que a roleta sobre o qual para no respectivo pedaço.

Existem outros tipos de seleção como: método de Torneio, Método de Seleção por ranking e Seleção truncada.

O método de seleção por torneio é realizado da seguinte forma: é formado um grupo com determinados indivíduos que utilizam o seu grau de avaliação para competirem entre si.

No método de Seleção por Ranking, ocorre a ordenação os indivíduos de uma população pelo grau de aptidão. Cada indivíduo recebe um novo valor referente à sua posição no ranking. É feito então a seleção dos melhores indivíduos conforme sua posição no ranking.

No método de seleção Truncada são ordenados os indivíduos de forme decrescente de acordo com o grau de aptidão. Os indivíduos selecionados serão aqueles que estiverem entre 1 e o ponto de corte.

4.9.4 Operadores Genéticos

Os operadores genéticos são operadores que possuem como objetivo a transformação de uma população durante suas gerações. Ou seja, gerar uma nova população diferente da que já existia.

Há uma grande variação de operadores genéticos os mais comuns são: Cruzamento (crossover), Mutação, Inversão e Elitismo.

Crossover simples - Após a seleção dos pais pelo módulo de seleção é definido um ponto de corte, ou seja, posição entre dois genes do cromossomo e feita a separação dos pais a partir do ponto de corte. Feito isso é feita a junção dessas partes, ou seja, temos 2 pais divididos entre 2 partes (figura 13).

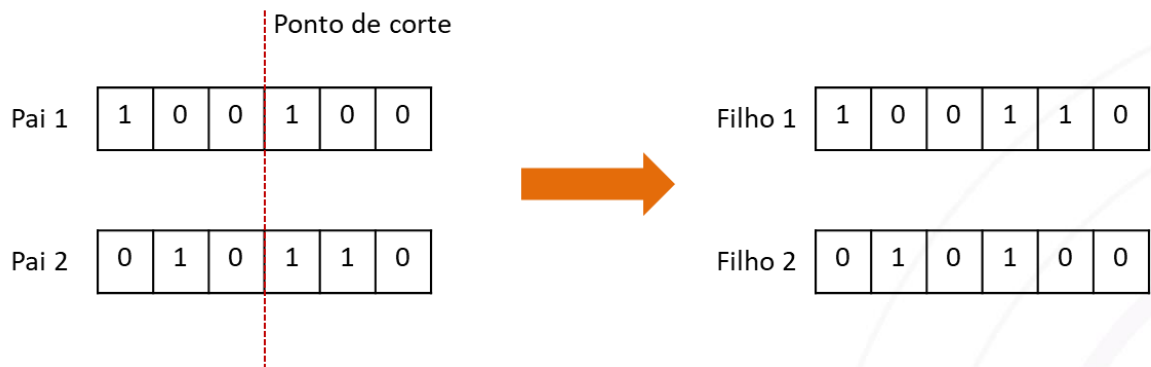


Figura 13: Exemplo de emprego do operador de “crossover”.

Crossover de dois pontos - o crossover de 2 (dois) pontos tem a seguinte regra de funcionamento (figura 10):

- Sortearmos dois pontos de corte;
- O primeiro filho será então formado pela parte do primeiro pai fora dos pontos de corte e pela parte do segundo pai entre os pontos de corte;
- O segundo filho será formado pelas partes restantes.

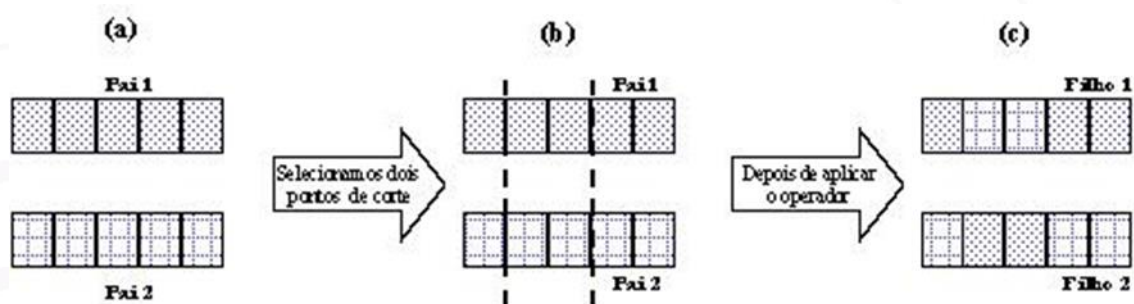


Figura 10: Exemplo de operador crossover de dois pontos.

Exemplo:

- Suponha que temos dois pais de tamanho 10, dados respectivamente pelas strings 0101010101 e 1111000011;

- b) Sorteamos os pontos de corte 4 e 8;
- c) Primeiro filho será dado, então, por:
 - I. A parte do primeiro pai até o ponto de corte 4 (0101);
 - II. A parte do segundo pai entre o ponto de corte 4 e o ponto de corte 8 (0000);
 - III. A parte do primeiro pai localizada após o ponto de corte 8 (01);

No final, o valor deste filho será **0101000001**.

Crossover uniforme - o crossover uniforme tem a seguinte regra de funcionamento (figura 11):

- a) Para cada gene é sorteado um número zero ou um;
- b) Se sortearmos um, primeiro filho recebe gene da posição corrente do primeiro pai e o outro, o do segundo pai;
- c) Se o valor sorteado for zero, as atribuições serão invertidas.

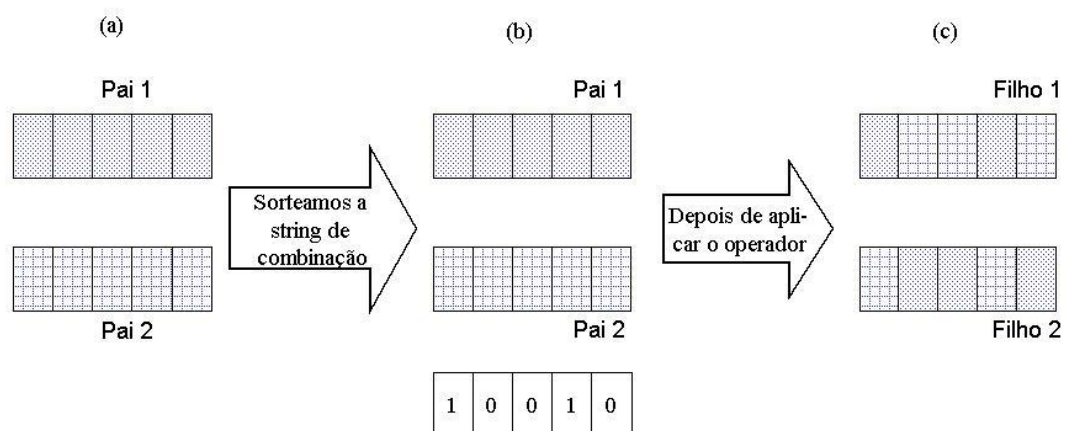
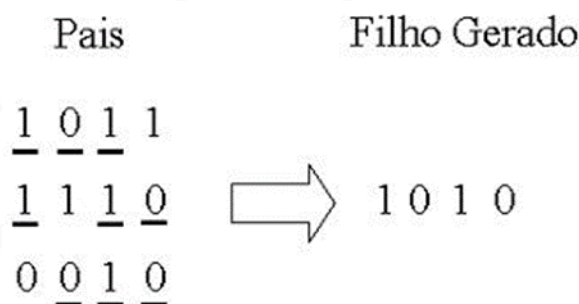


Figura 11: Exemplo de operador crossover uniforme.

Crossover baseado em maioria - não muito usado pois tende a fazer com que a convergência genética ocorra rapidamente. Operação básica: sortear “n” pais; cada bit do filho seja igual ao valor da maioria dos pais selecionados.



Crossover de mutação - O operador de mutação é definido, geralmente, depois da geração dos filhos. É definido uma probabilidade, por exemplo, de 0,5%. É sorteado então para cada gene um número entre 0 e 1. Caso o valor sorteado seja menor que a probabilidade, ou seja, 0,5% ele atua sobre o gene, o alterando. Abaixo segue uma representação de mutação (figura 14).

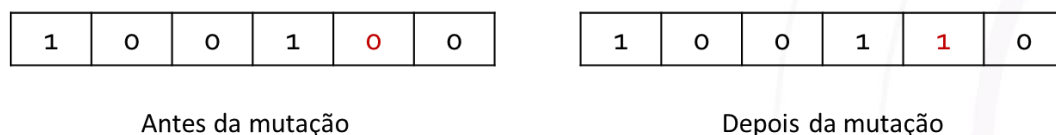


Figura 14: Exemplo de emprego do operador de “mutação”.

Em resumo, as etapas do processo compreender as seguintes atividades (figura 15):

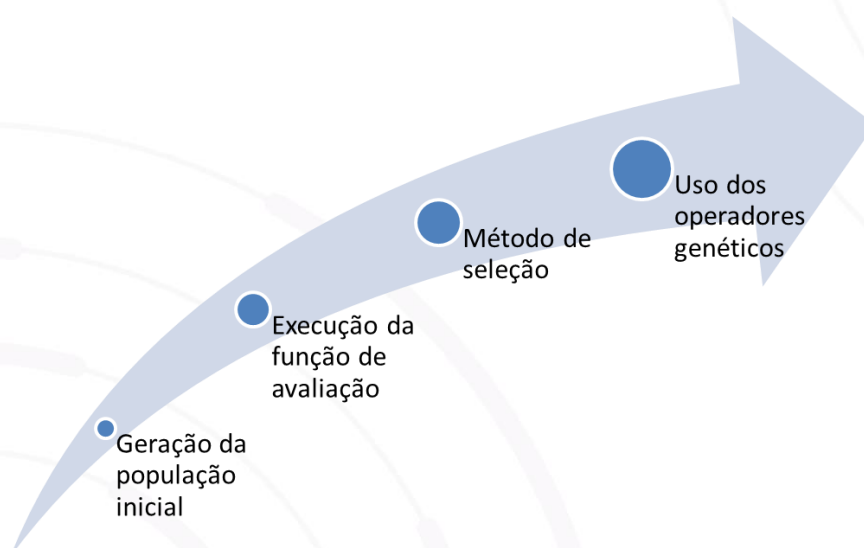


Figura 15: Etapas do processo que envolve os algoritmos genéticos.

Na figura 16 abaixo é possível verificar o esquema de um algoritmo genético:

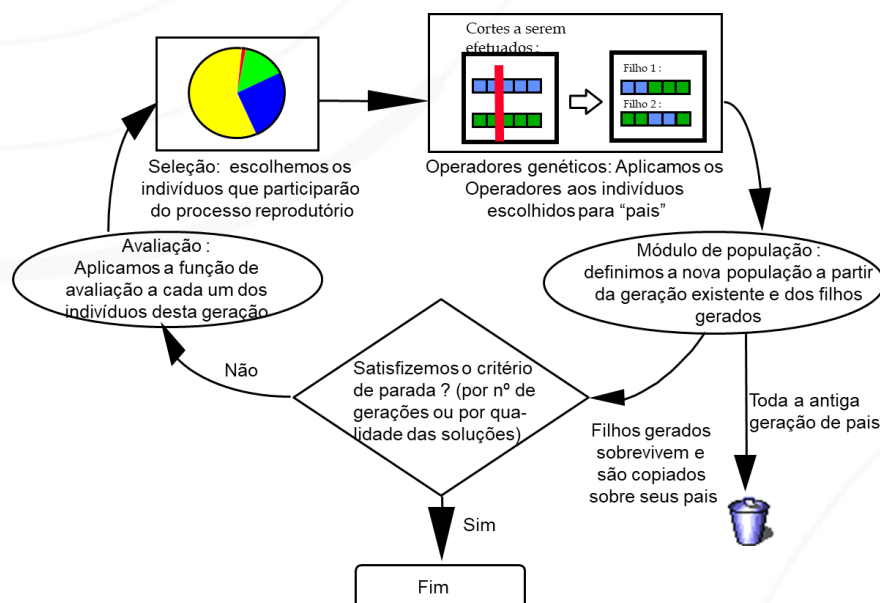


Figura 16: esquema de um algoritmo genético (Linden, 2008).

Basicamente, o esquema pode ser resumido, algoritmicamente, através dos seguintes passos:

- Inicialize a população de cromossomos;
- Avalie cada cromossomo na população;
- Selecione os pais para gerar novos cromossomos;
- Aplique os operadores de recombinação e mutação a estes pais de forma a gerar os indivíduos da nova geração;
- Apague os velhos membros da população;
- Avalie todos os novos cromossomos e insira-os na população;
- Se o tempo acabou, ou o melhor cromossomo satisfaz os requerimentos e desempenho, retorne-o, caso contrário volte para o passo c).

4.9.5 Exemplo de um Algoritmo Genético

Problema: Encontrar o valor máximo para $F(x) = x^2$, sendo que $0 \leq x \leq 60$.

Representação da solução: são selecionados alguns indivíduos de forma aleatória: 36, 18, 22 e 1. Para a representação destes, transformamos o número decimal para número binário:

Decimal	Binário
36	100100
18	010010
22	010110
1	000001

Após definir nossa população, devemos aplicar a função de aptidão que no nosso exemplo é x^2 . Abaixo segue uma tabela indicando a função de aptidão correspondente a cada indivíduo em questão.

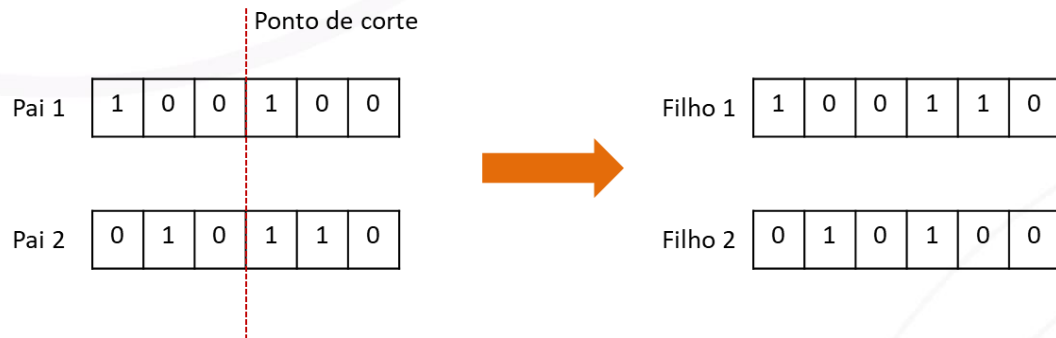
Decimal	Binário	Aptidão (x^2)
36	100100	1296
18	010010	324
22	010110	482
1	000001	1

Após a aplicação da função de aptidão, selecionados os pais para darmos continuação na evolução da população. No nosso caso o método de seleção que utilizaremos será o de **Seleção por Ranking**. Vamos atribuir um novo valor para cada indivíduo da população de acordo com o seu grau de aptidão conforme podemos verificar na tabela abaixo:

Decimal	Binário	Aptidão (x^2)	Ranking
36	100100	1296	1
18	010010	324	3
22	010110	482	2
1	000001	1	4

Após listarmos por ordem crescente os melhores indivíduos da população, vamos selecionar os dois melhores do ranking (100100 e 010110) para serem os pais que geram novos indivíduos. Para isto, serão aplicados os operadores genéticos (nesse caso

utilizaremos o *crossover* simples e mutação). Abaixo podemos identificar a representação desse cruzamento.



Após realizarmos o *crossover* simples iremos representar a mutação que ocorrerá em um dos filhos gerados, no caso vamos utilizar o filho 1.



Acima representamos o processo de evolução de uma população, o qual se repetirá até chegarmos a uma solução ótima para o problema relacionado, no caso o valor máximo que a função $f(x) = x^2$ pode ter.

4.10 Aplicações com Algoritmos Genéticos

Síntese de circuitos analógicos: para uma certa entrada e uma saída desejada, por exemplo tensão, o AG gera a topologia, o tipo e o valor dos componentes do circuito;

Síntese de protocolos: determinação de quais funções do protocolo devem ser implementadas em hardware e quais devem ser implementadas em software para que um certo desempenho seja alcançado;

Programação Genética: gera a listagem de um programa, numa determinada linguagem especificada, para que um determinado conjunto de dados de entrada forneça uma saída desejada;

Gerenciamento de redes: supervisão do tráfego nos links e das filas nos "buffers" de roteadores para descobrir rotas ótimas e para reconfigurar as rotas existentes no caso de falha de algum link;

Computação Evolutiva: gera programas que se adaptam a mudanças no sistema ao longo do tempo;

Otimização evolutiva multicritério: otimização de funções com múltiplos objetivos que sejam conflitantes;

Problemas de otimização complexos: problemas com muitas variáveis e espaços de soluções de dimensões elevadas. Ex: problema do caixeiro viajante, gerenciamento de carteiras de fundos de investimento;

Ciências biológicas: modela processos biológicos para o entendimento do comportamento de estruturas genéticas;

Autômatos auto programáveis

Engenharia em Construções: os AG's têm ganhado aceitação em um elevado número de problemas de engenharia . Uma aplicação é na otimização discreta de estruturas;

Compressão de Dados: a compressão de dados em geral, e a compressão de imagens sólidas em particular. Esta aplicação consiste em encontrar um método que utiliza os AG's para encontrar um sistema de funções locais iteradas (LIFS) para a codificação de imagens. Produzindo como resultado uma imagem com qualidade similar a utilização do método convencional de compressão fractal, com um tempo 30% menor;

Redes Neurais: alguns autores têm buscado encontrar uma relação entre RNA's e AG's. De alguma maneira eles tentam criar um relacionamento com o processo de aprendizagem em Redes Neurais. Eles têm estudado problemas de otimização relacionados com a busca de funções lineares discriminantes em problemas de classificação.

5. APRENDIZADO DE MÁQUINA

5.1 Aprendizagem

O aprendizado pode ser definido como: o ato ou efeito de aprender um ofício, uma arte ou ciência; o tempo despendido nesse processo de aquisição de conhecimentos; o exercício ou desempenho inicial do que se aprendeu; a denominação comum a mudanças permanentes de comportamento em decorrência de treino ou experiência anterior (MICHAELIS, 2020).

Dessa maneira é possível observar que o processo de aprendizado está intimamente relacionado com a influência de conhecimentos e novas observações que serão encarregados de tornar possível a execução de novos conhecimentos.

Essas influências estão associadas a métodos de inferência lógica. Tais métodos de influência lógica, poderão ser acoplados em três classes distintas, sendo elas a dedução, indução e abdução (PRATI, 2006).

Na dedução, a conclusão é obtida através de dados e informações já existentes, obtendo dessa forma um conhecimento verdadeiro. De modo geral, esse tipo de aprendizagem engloba qualquer conhecimento adquirido, sendo resultante de transformações sobre um conhecimento já existente, mantendo assim a verdade.

No aprendizado por indução é definido como uma forma de raciocínio que tem como origem um conceito particular e generalizado, devido o aprendizado por indução permitir sobre-exceder as premissas obtendo conclusões genéricas originadas de exemplos específicos. Dessa maneira a indução procede da parte para o todo, do particular para o universal.

No aprendizado por abdução o objetivo é deduzir um conhecimento específico por meio de observações e informações já conhecidas. Algumas inferências de aprendizagem por abdução são criadas a partir de um grupo de predicados particulares não observados. Diante disso esses tipos de inferências utilizam conhecimentos incompletos além de agregar conhecimentos sobre um caso específico acerca de determinado assunto.

Um processo de aprendizagem inclui a aquisição de novas formas de conhecimento: o desenvolvimento motor e a habilidade cognitiva (através de instruções ou prática), a organização do novo conhecimento (representações efetivas) e as descobertas de novos fatos e teorias através da observação e experimentação. Desde o início da era dos computadores, tem sido realizadas pesquisas para implantar algumas destas capacidades em computadores.

5.2 Aprendizado indutivo

O objetivo do aprendizado indutivo é encontrar uma hipótese (h) que concorde com um conjunto de exemplos. Uma hipótese é considerada satisfatória quando possui uma boa capacidade de generalização.

Exemplo 1: detecção de bons clientes para um cartão de crédito.

- a) Tarefa (T): classificar potenciais novos clientes como bons ou maus pagadores;
- b) Medida de Desempenho (P): porcentagem de clientes classificados corretamente;
- c) Experiência de Treinamento (E): uma base de dados histórica em que os clientes já conhecidos são previamente classificados como bons ou maus pagadores.

Exemplo 2: diagnóstico de gravidez de risco

- a) Tarefa (T): classificar novas gestantes com potenciais riscos na gravidez;
- b) Medida de Desempenho (P): porcentagem de pacientes classificadas corretamente;
- c) Experiência de Treinamento (E): base de dados histórica contendo exemplos de gestantes com ou sem gravidez de risco.

5.3 Aprendizado de máquina

As técnicas que utilizavam a IA solucionavam um encadeamento de problemas práticos utilizando para isso um conhecimento de determinado âmbito como, por exemplo, o campo de medicina sendo possível criar inferências por meio de regras, descobertas de diagnósticos e detecção de infecções. Os sistemas desenvolvidos para tal finalidade

ficaram denominados de Sistemas Especialistas ou então de Sistemas Baseados em Conhecimento.

Para que esses sistemas fossem capazes de exercer sua função, era necessário a aquisição do conhecimento de um especialista do ramo. O processo realizado para isso era submetido a falta de cooperação, intuição do especialista, informações instáveis, entre outros fatores críticos para formar as características das informações apuradas.

Com o progresso da utilização desses sistemas, os problemas tornaram – se mais complexos além da existência de uma grande quantidade de dados. Diante dessa situação tornou – se inevitável a criação de sistemas mais sofisticados que fossem capazes de resolver problemas sem necessitar de intervenções humanas. Os sistemas cobiçados para essas funções deveriam possuir como base para isso as experiências ocorridas anteriormente e construir inferências, para assim serem capazes de resolverem determinados problemas.

A técnica para a obtenção de conclusões genéricas com base em um agrupamento de exemplos é denominada de indução, que é um processo de inferência lógica, assim como detalhado anteriormente.

O Aprendizado de Máquina (AM) - do inglês, *Machine Learning* - é um subcampo da área de Inteligência Artificial, tem como intuito de criar sistemas que a partir do fornecimento de dados sejam capazes de tornar os mesmos aptos a aprender (PRATI, 2006 apud MONARD, 2003). O AM utiliza de algoritmos que são habilitados a executar a tomada de decisões. Dessa maneira, o AM irá desenvolver sistemas que serão treinados a partir do emprego de um grande volume de dados. Estes sistemas também são responsáveis por prever resultados, tendências e padrões de forma que, quanto maior for o volume de dados mais preciso os algoritmos (BARNES, 2015).

O aprendizado é adquirido através de experiências, que através do progresso das tarefas, o sistema aprende a melhor forma de resolver os problemas (SILVA, 2005 apud SANTOS, 2005 apud MITCHELL, 1997). Dessa forma o Aprendizado de Máquina é definido como “Diz – se que um programa de computador aprende a partir da experiência E com respeito a algumas classes de tarefas T e uma dada medida de desempenho P se o seu

desempenho nas tarefas T, medidas por P, melhorarem com a experiência E.” (SILVA, 2005 apud MITCHELL, 1997, p. 2).

5.1.1 Características gerais do Aprendizado de máquina

As técnicas que utilizavam a IA solucionavam um encadeamento de problemas práticos utilizando para isso um conhecimento de determinado âmbito como, por exemplo, o campo de medicina sendo possível criar inferências por meio de regras, descobertas de diagnósticos e detecção de infecções. Os sistemas desenvolvidos para tal finalidade ficaram denominados de Sistemas Especialistas ou então de Sistemas Baseados em Conhecimento.

Os sistemas de Machine Learning possuem características próprias que permitem uma classificação não exclusiva deles em função da linguagem de descrição, tipo de aprendizado, paradigma de aprendizado, formas e tarefas de aprendizado (PRATI, 2006).

A tabela 1 exibe de forma resumida as características de um sistema de AM.

Tipos de Aprendizado	Técnicas	Paradigmas	Linguagens de Descrição	Formas de Aprendizado	Tarefas de Aprendizado
Supervisionado	K-NN	Simbólico	Lógica de Ordem Zero	Incremental	Classificação
Não Supervisionado	Redes Neurais Artificiais	Conexionista	Lógica de Atributos	Não Incremental	Regressão
Semi-Supervisionado	Árvores de Decisão	Estatístico	Lógica de Primeira Ordem		Ordenação
Por Reforço	Naive Bayes	Genético			
		Baseado em Exemplos			

Tabela 1: Técnicas e paradigmas de Aprendizado de Máquina.

5.1.2 Tipos de Aprendizado de máquina

Em relação ao modo como os algoritmos aprendem, os sistemas de Machine Learning são divididos em: supervisionado, não supervisionado, semi-supervisionado e por reforço (PRATI, 2006).

No aprendizado supervisionado o sistema irá receber um conjunto de dados rotulados, onde cada um desses dados é representado por um conjunto de atributos e pelo valor da classe à qual o dado pertence. O aprendizado supervisionado tem como objetivo classificar de maneira correta a classe de novos dados que ainda não foram rotulados.

No aprendizado não supervisionado, por outro lado, os algoritmos não conhecem a classe à qual o conjunto de dados estão relacionados, com isso o sistema tenta classificar esses conjuntos agrupando os semelhantes em determinadas classes. Esse agrupamento também é conhecido como Clusterização – do inglês, Clustering –um processo que otimizará os dados tornando os resultados da classificação mais eficiente. Após a realização dos agrupamentos, é realizada uma análise para determinar o significado de cada cluster na situação do problema sendo analisado.

O aprendizado semi-supervisionado se baseia entre o aprendizado supervisionado e o aprendizado não supervisionado. Nesse aprendizado são utilizados dados rotulados e não rotulados. A técnica é utilizada, normalmente, em situações em que os conjuntos de dados rotulados são difíceis de serem adquiridos, enquanto os dados não rotulados estão presentes de forma abundante no conjunto e são facilmente apurados. Dessa forma, o objetivo desse aprendizado é classificar os dados obtidos para a criação de melhores classificadores.

5.1.3 Paradigmas de Aprendizado de máquina

A representação do conhecimento é obtida a partir de dados, podendo ser feita de várias maneiras, a partir de técnicas de aprendizado de máquina, como K - Vizinhos mais Próximos, Redes Neurais Artificiais, Árvores de Decisão ou pelo Naive Bayes, como visto na seção anterior (PRATI, 2006).

Essas distintas formas de representação do conhecimento são provenientes dos paradigmas de aprendizado. Nesta seção são detalhados alguns destes, tais como o paradigma simbólico, conexionista, evolucionista, estatístico e baseado em exemplos.

O paradigma simbólico o processo de aprendizado de máquina se dá através de representações simbólicas, ou seja, por meio de números, letras, palavras e símbolos e expressa o seu aprendizado através de algoritmos de AM, como por exemplo, expressões lógicas, árvores de decisão, regras de produção ou redes semânticas.

O paradigma conexionista tem como principal representante as RNA's, que são esquemas matemáticos para representar o funcionamento do neurônio biológico como visto no tópico anterior. Tal paradigma recebe o nome de conexionismo pelo fato de os sistemas funcionarem através de unidades neurais interconectadas.

A ideia principal do paradigma estatístico baseia-se em utilizar modelos estatísticos a fim de buscar possibilidade de ter uma aproximação do conceito de ser induzido. Grande parte desses modelos são paramétricos e para isso apresentam-se de modelos pré-estabelecidos, de modo que os valores são ajustados para os parâmetros do modelo a partir dos dados.

O paradigma genético é proveniente do paradigma evolucionista de aprendizagem. Esse paradigma utiliza como analogia a teoria darwinista ou teoria da seleção natural que estabelece que as espécies mais adaptadas ao meio têm mais chances de sobreviver do que as espécies menos adaptadas, enquanto no modelo genético de aprendizado, um conjunto de exemplos disputam entre si para fazer um prognóstico de novos exemplos, no qual os elementos que possuem menor desempenho são desprezados, à medida que os mais fortes se proliferam (PRATI, 2006 apud HOLLAND, 1975).

O paradigma baseado em exemplos tem como conceito classificar casos desconhecidos, tendo como base para isso os casos similares já classificados anteriormente no qual as classes são conhecidas. Um exemplo de técnica de aprendizado de máquina que utiliza esse paradigma são os K-vizinhos mais próximos (K-NN), onde o sistema irá armazenar todos os exemplos de treinamentos e a classificação será composta pela maior quantidade de exemplos vizinhos de uma determinada classe.

5.1.4 Aprendizado profundo de máquina

O Aprendizado Profundo de Máquina ou *Deep Learning* “[...]é uma subárea de Aprendizagem de Máquina que investiga técnicas para simular o comportamento do cérebro humano em tarefas como reconhecimento visual, reconhecimento de fala e processamento de linguagem natural.” (BEZERRA, 2016, p. 30). Em outras palavras, criar uma máquina que possa “fingir” e se passar por qualquer ser humano, não importa a nacionalidade, sem ser percebida como tal.

“O Aprendizado Profundo permite que modelos computacionais compostos por várias camadas de processamento aprendam representações de dados com vários níveis de abstração.” (LECUN et al., 2015). O objetivo é fazer com que a máquina reconheça e separe os dados e aloque-os de acordo com a sua similaridade, assim percebendo variações que os seres humanos demorariam para perceber, fazendo uso do algoritmo de retropropagação (LECUN et al., 2015 apud RUMELHART; HINTON; WILLIAMS, 1986) para treinamento das redes, afim de indicar como uma máquina deve alterar seus parâmetros usando a representação em cada camada a partir do representação na camada anterior, para assim realizar seus cálculos (LECUN et al., 2015).

O Aprendizado Profundo de Máquina é muito utilizado em diversas aplicações como o Google Translate ou Tradutor, para o reconhecimento e tradução de textos, pelos assistentes pessoais como o Cortana (Microsoft), programas de reconhecimento facial que são muito utilizados em redes sociais (Facebook), pelos sistemas de segurança (Facewatch), com a finalidade de identificar pessoas através de imagens. Este tipo de aprendizado é utilizado para classificação de doenças, principalmente na área oftalmológica, para verificação por meio de imagens da retina ocular se o indivíduo possui glaucoma ou qualquer outra doença no âmbito oftalmológico (DeepMind). Esse tipo de aprendizado também é usado para redução de erro de diagnósticos cancerígenos por meio de imagens de linfonodos (Google). Programas de carros autônomos também fazem uso para a verificação de rotas mais curtas, lugares mais visitados, entre outros (Uber).

6. REDES BAYESIANAS

6.1 Conceitos básicos de Probabilidade

Para elucidar as Redes Bayesianas, primeiramente é necessária a compreensão da probabilidade, visto que esta é fundamentada nesses conceitos. Existem situações em que não é possível determinar o resultado de um evento, pois este apresenta variabilidade em sua ocorrência. Logo, para lidar com esses casos, é preciso atribuir uma medida que lide com a incerteza, a probabilidade. Segundo Faria R. (2014), há diversos conceitos possíveis para probabilidade como: a frequentista, clássica e bayesiana.

A teoria frequentista define que a probabilidade de ocorrência de um evento é a sua frequência relativa, ou seja, a quantidade de vezes que um evento aconteceu, repetido por diversas vezes até formar uma amostra, colocado sob condições similares. Com isso, é possível perceber que o resultado se aproxima de um número finito de repetições. Tomando como exemplo, um dado que é lançado 50 vezes, e observando que em 28 dos lançamentos resultaram no número 6, pode-se dizer que para esse dado, o número 6 ocorreu 0.56. No entanto, essa teoria possui alguns problemas como: a dificuldade de saber quantas vezes um experimento deve ser repetido para conseguir uma amostra significativa; definição de condições similares, pois isso poderia tirar o nível de incerteza do processo e em situações que não podem ser replicadas, não é possível obter a frequência relativa.

Já a teoria clássica estabelece que todos os possíveis resultados têm a mesma chance de ocorrência, ou seja, equiprováveis. Para isso, o cálculo é feito da seguinte forma: o número de elementos do evento sobre o número de resultados possíveis. Dando como exemplo, uma moeda e desejando-se saber qual a probabilidade de resultar em cara, sabendo que o número de caras é igual a 1 e o número de total de resultados é igual a 2 (cara ou coroa), logo cara e coroa tem 0.5 de chance de ocorrer. Essa teoria também possui uma restrição, pois não existe método para casos que não sejam equiprováveis.

Por fim, a teoria Bayesiana ou Subjetivista, inclusive o foco deste trabalho, é baseada na crença que uma pessoa tem sobre a ocorrência de um evento, o nível de crença é medido no intervalo de $[0,1]$, no qual 0 representa a incapacidade de ocorrência e 1 o indicativo de que certamente a crença ocorrerá. Nessa teoria, não existe uma probabilidade certa, mas sim a probabilidade adquirida pelo conhecimento de um indivíduo sobre uma

ocorrência. Para ilustrar esta teoria, suponha que o conhecimento de uma pessoa A sobre a chance de ocorrência de um dado seja o número $6 = 0.5$ e, que outro indivíduo B que não viu o lançamento anterior, e a partir de seus conhecimentos diga que o número 6 tem a chance de ocorrência de 1. De acordo com Faria R. (2014), a teoria Bayesiana é criticada por diversos autores por alegarem que uma pessoa não está livre de preconceitos ao atribuir uma probabilidade e pelo fato de apresentar dificuldades quando conduzidas por mais de um pesquisador. No entanto são capazes de lidar com a incerteza e, também, podem atuar em conjunto com a frequência relativa e clássica.

Em resumo pode-se dizer que segundo Faria A. (2009), a Probabilidade é um campo que estuda a ocorrência de experimentos aleatórios. Experimentos aleatórios são acontecimentos que não se pode prever com certeza.

Para melhorar a compreensão dos assuntos que serão abordados, alguns conceitos de probabilidade são definidos a seguir:

- a) Espaço Amostral é o conjunto de todos os resultados possíveis de um experimento aleatório. Sendo definido pela letra Ω ou S.
- b) Evento é um subconjunto do espaço amostral. Sendo os eventos representados por letras maiúsculas e os elementos por minúsculas.
- c) Intersecção entre eventos são os elementos que estes têm em comum.
- d) Eventos disjuntos não possuem nenhum elemento em comum.
- e) Eventos são mutuamente exclusivos, se não podem ocorrer simultaneamente.

6.1.1 Axiomas da Probabilidade

Axiomas são premissas que se admite como plenamente verdadeiras sem a necessidade de constatação. São as propriedades que dão a base para a probabilidade, portanto uma probabilidade P que associa a um evento A do Ω em um número real $P(A)$, deve satisfazer os três axiomas:

I. Axioma: $P(A) \geq 0$: Todas as probabilidades devem ser positivas, ou seja, iguais ou maiores que 0.

II. Axioma: $P(\Omega) = 1$: A probabilidade do espaço amostral será sempre 1, isto é, a soma de todos os resultados possíveis, sempre deve resultar em 1.

III. Axioma: $A \cap B = \emptyset \rightarrow P(A \cup B) = P(A) + P(B)$: Se A intersecção com B resultar no conjunto vazio, ou seja, nenhum elemento em comum, isto quer dizer que a probabilidade de A união com B é igual a probabilidade de A mais a probabilidade de B.

6.1.2 Probabilidade Condicional

Axiomas são premissas que se admite como plenamente verdadeiras sem a necessidade de constatação. São as propriedades que dão a base para a probabilidade, portanto uma probabilidade P que associa a um evento A do Ω em um número real $P(A)$, deve satisfazer os três axiomas:

A probabilidade condicional define que um evento depende de outro para ocorrer. É representada por $P(B|A)$, que pode ser lida como: “a probabilidade do evento B acontecer dada a ocorrência do evento A”. Logo B será calculado em função do espaço amostral reduzido de A.

$$P(B|A) = \frac{P(B \cap A)}{P(A)}$$

A fórmula acima é lida da seguinte maneira: a probabilidade dos eventos em comum de A e B em relação ao espaço amostral, sobre o espaço de A. Uma importante consequência da probabilidade condicional é o teorema da multiplicação (HAZZAN; IEZZI, 2004 apud GONÇALVES, 2008). Em situações, em que a probabilidade condicional é fácil de se calcular, é possível resolver a ocorrência simultânea de eventos pelo teorema da multiplicação. O teorema define que a probabilidade de dois eventos ocorrerem simultaneamente é a probabilidade do primeiro multiplicado pela probabilidade do segundo em relação ao primeiro. É representado pela fórmula a seguir:

$$P(A \cap B) = P(B|A) \cdot P(A)$$

Considere n eventos $B_1 \dots B_n \in \Omega$ e que são disjuntos, mas cuja união resultam no espaço amostral. Pode-se dizer que esses eventos formam uma partição.

Para calcular eventos diferentes que possuem partições, utilizamos o teorema da probabilidade total. Então para todo evento $A \in \Omega$:

$$P(A) = \sum_{j=1}^n P(B_j) \cdot P(A | B_j)$$

6.1.3 Independência

Um evento A independe de B, se a ocorrência de B não afeta a probabilidade de A.

$$P(A \cap B) = P(A) \cdot P(B)$$

Um evento X é condicionalmente independente de Y dado Z, se a distribuição que rege Z é independente de Y dado o valor de Z, que pode ser representado por:

$$P(X | Y \cap Z) = P(X|Z)$$

6.1.4 Variáveis Aleatórias

Conforme Silva (2016, p.13), uma variável aleatória (v.a) pode ser considerada um experimento e cada possível saída para um experimento é chamada de estado. Variáveis aleatórias normalmente são representadas por letras maiúsculas (A, B) e os elementos ou estados associados a ela são denotados por letras minúsculas (a, b). Além disso, Norvig e Russel (2004 apud Valentim, 2007) ainda as classificam em três categorias: variáveis aleatórias booleanas, variáveis aleatórias discretas e variáveis aleatórias contínuas.

As variáveis aleatórias booleanas só contêm dois valores, como por exemplo: a variável *TemGaragem* pode possuir os estados {*verdadeiro* ou *falso*}. Já as variáveis aleatórias discretas, ao contrário das booleanas, permitem múltiplos valores finitos, por exemplo, a variável *EstadoCivil* pode conter os valores {*Solteiro*, *Casado*, *Divorciado*, *Viúvo*}. Em relação, as variáveis aleatórias contínuas, estas podem classificar seus valores em um intervalo de [0,1] ou na linha real inteira.

6.1.5 Distribuição da Probabilidade Conjunta

Segundo Silva (2016), a distribuição da probabilidade conjunta é a probabilidade de ocorrência de duas ou mais variáveis aleatórias x_1, x_2, \dots, x_n ocorrerem juntas. Uma função de

probabilidade conjunta necessariamente precisa satisfazer a seguinte propriedade, de acordo com Valentim (2016):

$$\sum P(X_1, X_1, \dots, X_n) = 1$$

Na equação 1.6 diz que os somatórios de todos os valores das variáveis devem resultar em 1. Russel (2004 apud Valentim, 2007) ainda as classificam em três categorias: variáveis aleatórias booleanas, variáveis aleatórias discretas e variáveis aleatórias contínuas.

6.1.6 Teorema de Bayes

O Teorema de Bayes é uma extensão do teorema da probabilidade total e da probabilidade condicional. Enquanto o segundo permite calcular um resultado dentre múltiplos eventos de cada partição a priori, ou seja, antes do evento acontecer, o Bayes permite esses mesmos cálculos a partir de uma condição a posteriori, ou seja, depois que o evento a priori aconteceu. A definição formal do Teorema de Bayes definida por (LIPSCHUTZ, 1993 apud GONÇALVES, 2008):

Teorema: Suponha $A_1, A_2, A_3 \dots A_n$ ser uma partição de S e B, um evento qualquer. Então para qualquer i

$$P(A_i|B) = \frac{P(B|A_i) \cdot P(A_i)}{P(B)} = \frac{P(B|A_i) \cdot P(A_i)}{\sum_i P(B|A_i) \cdot P(A_i)}$$

Uma interpretação do Teorema de Bayes, **consiste em considerar os eventos A_i como “causas” do evento B**, sendo atribuído probabilidades deste evento atuar na ocorrência de B. Esta probabilidade é calculada antes da realização do experimento, sendo designada como a probabilidade *a priori* de A_i . Após a realização do experimento, é conhecido que o evento B ocorreu, então a probabilidade *a priori* é revista por meio da fórmula de Bayes e então passa a atribuir aos eventos A_i , $i = 1, 2 \dots n$ as probabilidades *a posteriori* $P(A|B_i)$, $i = 1, 2 \dots n$ (CRAMÉR, 1955 apud GONÇALVES, 2008)(PAULINO; TURKMAN; MURTEIRA, 2003 apud GONÇALVES, 2008).

6.2 Teoria dos Grafos

A teoria dos grafos é utilizada no estudo de Redes Bayesianas para representar visualmente as relações de causa e consequência de variáveis e probabilidades. Conforme Faria, R. (2014), um grafo é um conteúdo matemático, que tem como principal capacidade a representação de conhecimento de maneira visual, e assim é amplamente utilizado em diversas aplicações.

Um grafo pode ser definido como um conjunto de elementos que possuem conexões entre si. Os elementos são denominados vértices, nós ou variáveis e suas conexões são chamadas de arestas ou arcos, tal estrutura pode ser compreendida na Figura 17. A sua representação matemática é definida como $G = (V, A)$, onde V são os vértices que podem ser representados por letras ou nomes dentro de um círculo, aresta ou retângulo e as arestas que são representadas por setas. A junção de nós e arcos apresentam a relação de dependência probabilística.



Figura 17: Elementos básicos da Teoria dos Grafos (Souza, 2011).

6.3 Conceito de Redes Bayesianas

As Redes Bayesianas (RB's) são uma representação qualitativa (teoria dos grafos) e quantitativa (probabilidade) de cenários que envolvam o processo decisório, prognóstico ou analítico, tendo em vista condições de incerteza, complexidade e falta de informações. De acordo com Charniak (1991), uma maneira de entender a aplicação de Redes Bayesianas é quando se tem conhecimento prévio da causa de um evento, mas ainda é insuficiente para a compreensão total do processo, e dessa forma é preciso modelar o problema probabilisticamente. Para melhor compreensão das Redes Bayesianas considere o exemplo de Yudkowsky (2003, apud Valentim, 2007, p.13):

1% das mulheres com mais de 40 anos que participam de exames de rotina são portadoras de câncer de mama. 80% das mulheres com câncer terão resultados positivos de mamografias. 9,6% das mulheres sem a doença também terão resultado positivo nas mamografias. Uma mulher dessa idade se depara com um

resultado positivo de mamografia, qual a probabilidade dela portar câncer de mama?

Para a construção de uma Rede Bayesiana com o exemplo dado acima, primeiramente é preciso identificar as variáveis de interesse. Como a situação foca na influência que o câncer tem sobre o resultado da mamografia, é notável que as variáveis são: *Câncer* e *Mamografia*. O próximo passo é definir as relações de causa e consequência do domínio em questão. A causa é o câncer e a consequência é a mamografia, uma vez que o fato de possuir ou não a doença é o fator que determina o resultado da mamografia, exceto para os casos em que há erros. Depois de identificado as variáveis de causa e consequência é necessário representá-las na rede como mostra a Figura 18.



Figura 18: Representação da estrutura da Rede Bayesiana com o exemplo citado.

Na Figura 18 nota-se que a estrutura de uma Rede Bayesiana é a mesma de um grafo acíclico dirigido (DAG), já que os arcos possuem direção e o nó inicial (*Câncer*) é diferente do nó final (*Mamografia*). Em relação a causalidade, esta é representada por um nó *Câncer* com um arco direcionado para o nó *Mamografia*, também sendo equivalente a dizer que *Câncer* é “pai” de *Mamografia*. Também há os estados dos nós *Câncer* e *Mamografia*, sendo respectivamente {presente, ausente} e {positiva e negativa}, no qual indica a probabilidade de ocorrência.

Quando a estrutura da rede é construída, a próxima etapa é a definição das probabilidades *a priori* e as probabilidades condicionais para as variáveis. As probabilidades das variáveis aleatórias *a priori* contêm os valores antes de qualquer ocorrência de outra variável, enquanto as condicionais apresentam a probabilidade de uma variável acontecer dado que outra ocorreu. A Figura 19 mostra as probabilidades *a priori* das variáveis *Câncer*.



Figura 19: Representação da estrutura e das probabilidades *a priori* da Rede Bayesiana do exemplo citado.

Na Figura 19 pode-se interpretar assim como o exemplo, que a probabilidade *a priori* de uma mulher com mais de quarenta anos ter câncer é 1.0 e consequentemente, a probabilidade dela não ter é 99.0. A probabilidade *a priori* dos estados de mamografia é considerada 50.0 para resultado positivo da doença, tanto para resultado negativo, já que o exemplo não definiu uma probabilidade específica. É importante lembrar que, as probabilidades *a priori* e as condicionais, normalmente são definidas por especialistas na área, como o caso do domínio em questão que foram definidas por médicos, mas também podem ser definidas pela frequência relativa.

Assim que é finalizado a parametrização das probabilidades *a priori*, é necessário definir as probabilidades condicionais. Nesse caso, os estados da *Mamografia* {positiva, negativa}, dado que o estado de *Câncer* está presente ou ausente. A Figura 20 mostra o estado de *Câncer* estando presente e a Figura 20 apresenta a ausência da ocorrência de *Câncer*.



Figura 20: Representação da condicional Mamografia dado que Câncer está presente.

Na Figura 20, pode-se notar que o nó *Câncer* contendo o estado presente influencia o estado do nó da *Mamografia* resultar em positivo em 80.0 dos casos e negativo em 20.0, coincidindo com as informações dadas no exemplo.

Na Figura 21 (abaixo), é perceptível que o estado da variável *Câncer* estar ausente influencia no estado da mamografia dar um resultado positivo de 9.60 e negativo de 90.4, equivalentes ao exemplo dado acima.



Figura 21: Representação da condicional Mamografia dado que Câncer está ausente.

De forma geral, uma Rede Bayesiana é uma combinação da Teoria dos Grafos, ou seja, a representação visual da Figura 1 e dos conceitos de Probabilidade e Estatística. Além disso, as redes bayesianas também podem ser chamadas de Rede de Crença, Rede Probabilística ou Causal. O objetivo da rede é apresentar as relações de causa e efeito das variáveis, juntamente com a disposição da probabilidade conjunta e, a partir disso possibilitar o procedimento de inferência. A rede também é capaz de por meio das relações de independência condicional, probabilidade condicional e independências incondicionais aliado com a representação visual das variáveis providas pelos grafos, compreender como um cenário atua e até mesmo, prever as ações de uma intervenção, a partir do cálculo de probabilidades *a priori*, ou seja, antes do evento acontecer e *a posteriori*, sendo depois do evento acontecer.

6.4 Construção de uma Rede Bayesiana

Em primeiro lugar, deve-se selecionar as variáveis que são importantes no processo. Para isso, deve-se seguir alguns critérios Kjærulff e Madsen (2013 apud FARIA R, 2014): as variáveis devem conter valores mutuamente exclusivos, ou seja, que não podem ocorrer ao mesmo tempo; devem representar um conjunto único de eventos, sem a ocorrência de redundâncias; e devem ser muito bem definidas. Com as variáveis do modelo devidamente selecionadas, agora o problema é construir a estrutura da Rede Bayesiana, que é identificar entre quais variáveis terão arcos e qual direcionamento deve ser dado a cada um deles.

A construção pode ser feita através das relações de causa efeito identificadas pelo especialista da área a ser estudada, porém, caso não esteja disponível um especialista para auxiliar na construção da rede ou ele precise de um guia para ajudá-lo, é útil manter em mente os tipos de variáveis que podem surgir:

Variáveis problema: São variáveis de interesse, das quais deseja-se calcular distribuições *a posteriori* dado os valores das variáveis de informação.

Variáveis de *background*: São informações disponíveis antes da ocorrência do problema, com influência causal sobre as variáveis problema e variáveis de sintomas. São geralmente as variáveis ‘raízes’ das redes bayesianas.

Variáveis de sintomas: São as consequências do problema, sendo disponíveis somente após sua ocorrência, ou seja, variáveis de sintomas são causadas pelas variáveis problema. Geralmente, as variáveis problema e de background são pais das variáveis de sintomas.

Variáveis mediadoras: São variáveis não observáveis cujas probabilidades *a posteriori* não são de interesse imediato, mas que ajudam a manter as relações de independência corretas na rede. Estas variáveis geralmente são filhas de variáveis problema e de background e pais de variáveis de sintomas.

Os tipos de variáveis acima, propostos por Kjærulff e Madsen (2013 apud ARA-SOUZA, p.41), são uma forma de auxiliar a construção de Redes Bayesianas feitas manualmente, pois, após a identificação do tipo de uma variável, é muito mais fácil conectá-la às outras, já que geralmente o fluxo causal apresentado na Figura 22 é satisfeito.

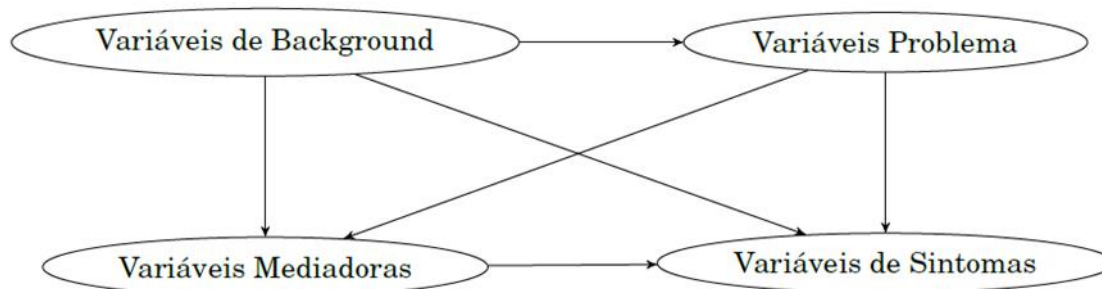


Figura 22: Fluxo casual mais comum em Redes Bayesianas (Faria, 2014).

6.5 Um exemplo prático – seguro de automóveis.

O seguro de automóveis tem como função a responsabilização pelos riscos e, também, pela indenização do veículo para o cliente no caso de colisão; acidentes; roubos e entre outros, em troca de um pagamento, denominado prêmio. Para que isso seja possível, a seguradora precisa estar regulamentada pela autarquia da Superintendência de

Seguros Privados (SUSEP), que é responsável pelo controle e fiscalização do mercado de seguros no Brasil.

A identificação do grupo com riscos semelhantes é realizada por um questionário de avaliação de risco do segurado. Esta avaliação de risco é feita por intermédio do corretor e se resumem basicamente em perguntas referentes a idade; tempo de habilitação; sexo; região de circulação do veículo; obtenção de garagem ou estacionamento fechado para o automóvel; utilização do veículo e dispositivos de segurança. A partir disso e de dados estatísticos, a seguradora é capaz de identificar o perfil do segurado e os possíveis riscos que irá assumir, como por exemplo, um grupo com idade entre dezoito e vinte cinco anos, do sexo masculino que apresenta muitos sinistros, ou seja, qualquer dano ao veículo, terá o valor do prêmio mais alto para todos com essas características.

De modo geral, existem diversos fatores que contribuem para a precificação do prêmio de seguro de automóveis: cobertura por colisão, idade do veículo, franquia, bônus e grau de periculosidade da cidade etc. Segue abaixo modelos de redes bayesianas aplicadas ao “case” citado.

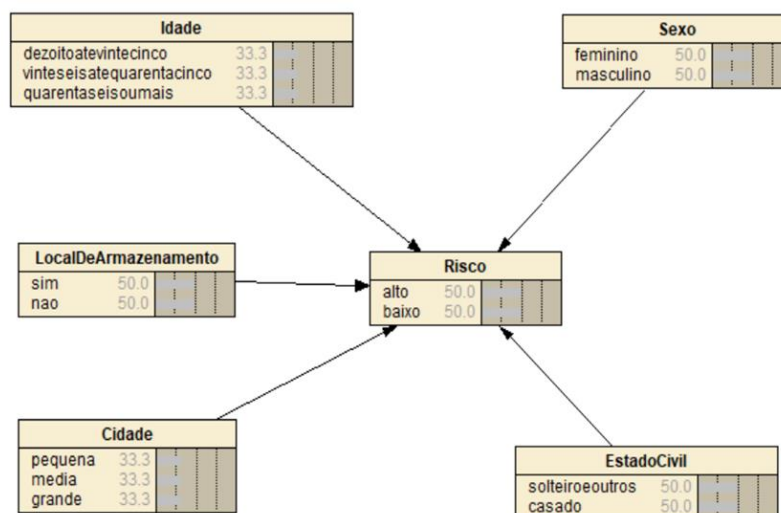


Figura 23: Redes Bayesianas com as probabilidades.

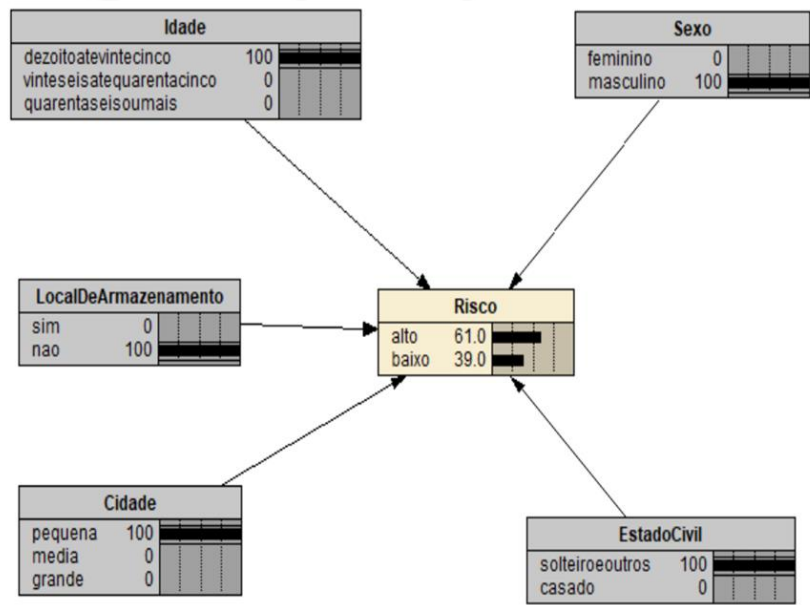


Figura 24: Redes Bayesianas com a inserção de evidências em todas as variáveis.

7. SISTEMAS NEBULOSOS

7.1 Conceitos de Lógica Difusa (Fuzzy)

A Lógica difusa (fuzzy) é uma lógica multivalorada capaz de capturar informações vagas, em geral descritas em uma linguagem natural e convertê-las para um formato numérico, de fácil manipulação pelos computadores de hoje em dia. A representação depende não apenas do conceito, mas também do contexto em que está sendo usada. Esta lógica pode ainda ser definida como a lógica que suporta os modos de raciocínio que são aproximados, ao invés de exatos, como estamos acostumados a trabalhar.

qA Lógica Difusa foi desenvolvida por Lofti A. Zadeh da Universidade da Califórnia em Berkeley na década de 60 e combina lógica multivalorada, teoria probabilística, IA e RNA para poder representar o pensamento humano, ou seja, ligar a linguística e a inteligência humana, pois muitos conceitos são, melhor definidos, por palavras do que pela matemática.



7.2 Objetivo da Lógica Difusa

A lógica difusa objetiva fazer com que as decisões tomadas pela máquina se aproximem cada vez mais das decisões humanas, principalmente ao trabalhar com uma grande variedade de informações vagas e incertas, as quais podem ser traduzidas por expressões do tipo: a maioria, mais ou menos, talvez etc. Antes do surgimento da lógica fuzzy essas informações não tinham como ser processadas. Ela vem sendo aplicada nas seguintes áreas: Análise de dados, Construção de sistemas especialistas, Controle e otimização, Reconhecimento de padrões etc.

Ela baseia-se em palavras e não em números, ou seja, os valores verdadeiros são expressos linguisticamente. Por exemplo: baixo, médio, alto, e outros usados para definir

estados de uma variável. Possui vários modificadores de predicado como por exemplo: muito, mais ou menos, pouco, bastante, médio etc.

Possui também um amplo conjunto de quantificadores, como por exemplo: poucos, vários, em torno de, usualmente. Faz usos das probabilidades linguísticas, como por exemplo: provável, improvável, que são interpretados como números fuzzy e manipulados pela sua aritmética; e manuseia todos os valores entre 0 e 1, tomando estes, como um limite apenas.

É baseada em teoria dos conjuntos Fuzzy (este conjunto é formado por valores de 0 até 1), demonstrando o quanto um elemento pertence a um determinado conjunto através de seu grau de pertinência. Este grau pode ser obtido através de um cálculo, no qual são retornados valores entre 0 (não pertence) e 1 (pertence totalmente). A diferença da teoria do conjunto fuzzy para a teoria dos conjuntos clássica é justamente que para o fuzzy seus conjuntos de pertinência possuem valor de 0 a 1, já na clássica o valor só pode ser 0 ou 1.

A Lógica Fuzzy apresenta as seguintes características em relação a outras técnicas de controle:

- a) Robusta porque não requer entradas precisas;
- b) Modificada facilmente pois é baseada em regras;
- c) Controle de sistemas não-lineares sem modelo matemático;
- d) Solução mais rápida e barata em alguns casos.
- e) Implementável facilmente em microprocessadores.

7.3 Variáveis linguísticas

As variáveis linguísticas expressam os valores estipulados aos conjuntos Fuzzy, como por exemplo, na altura os valores linguísticos são: alto, médio e baixo. Esses valores podem ser formados a partir de uma sentença de uma linguagem específica como “muito baixa”, neste caso os valores seriam somados a partir do “muito” e “baixa” formando assim um valor composto. As categorias de termos linguísticos envolvem:

- a) **Termos primários:** nomes do conjunto universo como alto, médio e baixo;

- b) **Conectivos Lógicos:** São operações entre conjuntos utilizando “Não”, “E” e “OU”, “mas”, “porém”. Exemplo: ele é leve, mas não alto, equivale a: **peso leve E altura média**;
- c) **Modificadores:** Serve para unir dois termos linguísticos como, por exemplo: muito alto junta-se o conjunto muito com o alto;
- d) **Delimitadores:** ficam entre parênteses.

7.4 Raciocínio Fuzzy

Todos os conjuntos de valores e regras utilizadas na lógica Fuzzy são elaborados pelo especialista humano, que determina todos os limites dos conjuntos a serem utilizados. O raciocínio Fuzzy é baseado em 3 etapas: Fuzzificação, Inferência e Desfuzzificação, que serão explicados a seguir.

7.5 Fuzzificação

O papel da fuzzificação é o de transformar um dado numérico em uma variável linguística através de uma função de pertinência, determinando assim em qual conjunto Fuzzy o valor “fuzzificado” mais se enquadra. Vamos nos utilizar de um exemplo de peso e altura que ao final da lógica fuzzy irá nos retornar se o indivíduo deverá emagrecer, engordar, ou não fazer nada e o valor, para chegar no peso ideal. Para o peso temos 3 conjuntos Fuzzy: leve, médio e pesado; para a altura teremos 3 conjuntos Fuzzy: alto, médio e baixo. Supondo que o valor de entrada seja para peso seja 43 kg e altura 1,70 cm. Abaixo podemos verificar a tabela de limites de cada conjunto de peso e altura.

LIMITES DOS CONJUNTOS				LIMITES DOS CONJUNTOS ALTURA			
	Leve	Medio	Pesado		Baixo	Medio	Alto
MIN	20	40	80	MIN	1,4	1,6	1,8
MED	40	80	120	MED	1,5	1,7	1,9
MAX	60	100	140	MAX	1,7	1,8	2

Tabela de limites dos conjuntos fuzzy.

Na tabela abaixo temos a tabela de peso e altura e suas respectivas pertinências, o dado de entrada que utilizamos como exemplo estão em destaque, podendo observar que o valor de peso 43 kg possui pertinência 0,85 para leve e 0,075 para médio. Já a altura 1,70 possui grau de pertinência 1 para médio.

Tabela Peso				Tabela Altura			
LIMITES	leve	medio	pesado	LIMITES	baixo	medio	alto
20	0	0	0	1,4	0	0	0
40	1	0	0	1,5	1	0	0
60	0	0,5	0	1,6	0,5	0	0
80	0	1	0	1,7	0	1	0
100	0	0	0,5	1,8	0	0	0
120	0	0	1	1,9	0	0	1
140	0	0	0	2	0	0	0
43	0,85	0,075	0	1,7	0	1	0

Tabela pertinência de peso e altura.

Iremos encontrar o valor de pertinência para cada conjunto através da fórmula abaixo, esta é utilizada quando se trabalha com a função de pertinência triangular (figura 25):

$$\mu(x) = \max(\min(X - LI / M - LI, LM - X / LM - M), 0)$$

Inserindo o valor de exemplo na fórmula citada acima:

$$\mu_{\text{Leve}}(43) = \max(\min(43 - 20 / 40 - 20, 60 - 43 / 60 - 40), 0) = 0,85$$

Onde X será o valor de entrada no caso do exemplo acima 43, LI será o Limite inferior do grupo que, no caso deste exemplo, observando a figura 1, podemos observar que o limite inferior do conjunto leve é 20, M será o valor de pertinência 1 neste caso será 40, LM será o limite máximo do grupo neste exemplo será 60. E assim fazemos para todos os grupos.

$$\mu_{\text{Médio}}(43) = \max(\min(43 - 40 / 80 - 40, 100 - 43 / 100 - 80), 0) = 0,075$$

$$\mu_{\text{Pesado}}(43) = \max(\min(43 - 80 / 120 - 80, 140 - 43 / 140 - 120), 0) = 0$$

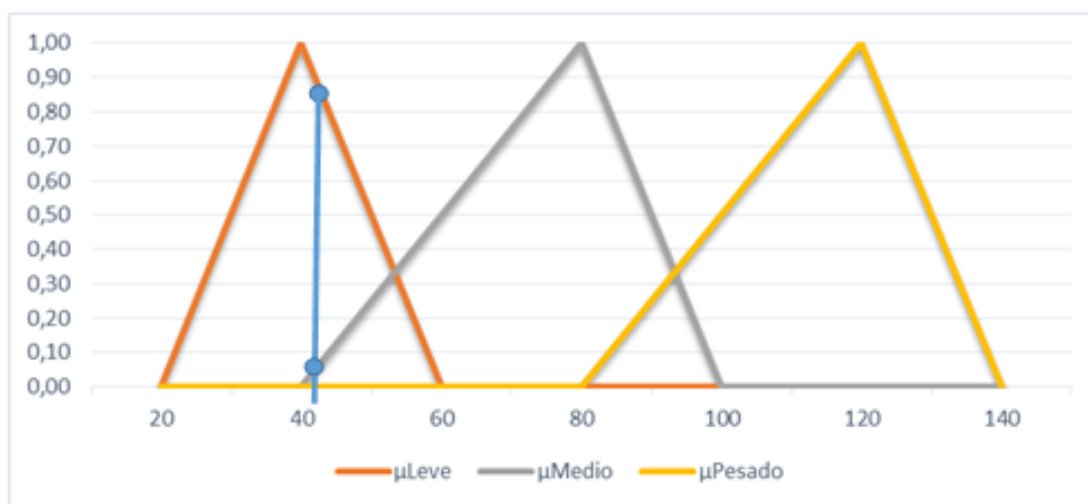


Figura 25: Resultado da fuzzificação do peso visualizado no gráfico, onde o eixo x representa os valores de peso e y representa o grau de pertinência de 0 a 1.

Da mesma forma que fizemos os cálculos acima para o peso será feito na altura, buscando no gráfico os valores dos limites de cada conjunto (figura 26).

$$\mu_{\text{Baixo}}(1,70) = \max(\min(1,70 - 1,40 / 1,50 - 1,40, 1,70 - 1,70 / 1,70 - 1,50), 0) = 0$$

$$\mu_{\text{Médio}}(1,70) = \max(\min(1,70 - 1,60 / 1,70 - 1,60, 1,90 - 1,70 / 1,90 - 1,70), 0) = 1$$

$$\mu_{\text{Alto}}(1,70) = \max(\min(1,70 - 1,80 / 1,90 - 1,80, 2,00 - 1,70 / 2,00 - 1,90), 0) = 0$$

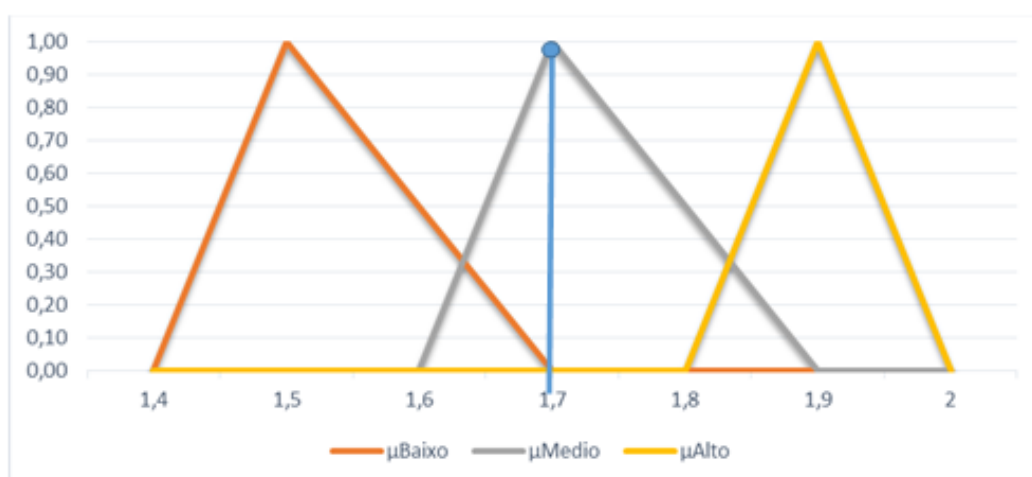


Figura 26 – Resultado da fuzzificação da altura visualizado no gráfico.

7.6 Função de pertinência triangular

Uma função de pertinência triangular depende de 3 parâmetros escalares $[a,b,c]$, onde $[a]$ e $[c]$ possuem grau de pertinência igual a 0 (zero) e $[b]$ possui grau de pertinência igual a 1 (um), significando que a e c não pertencem ao grupo e b pertence totalmente. Com esses 3 valores $[a,b,c]$ formamos um triângulo, conforme podemos ver na figura 27 abaixo.

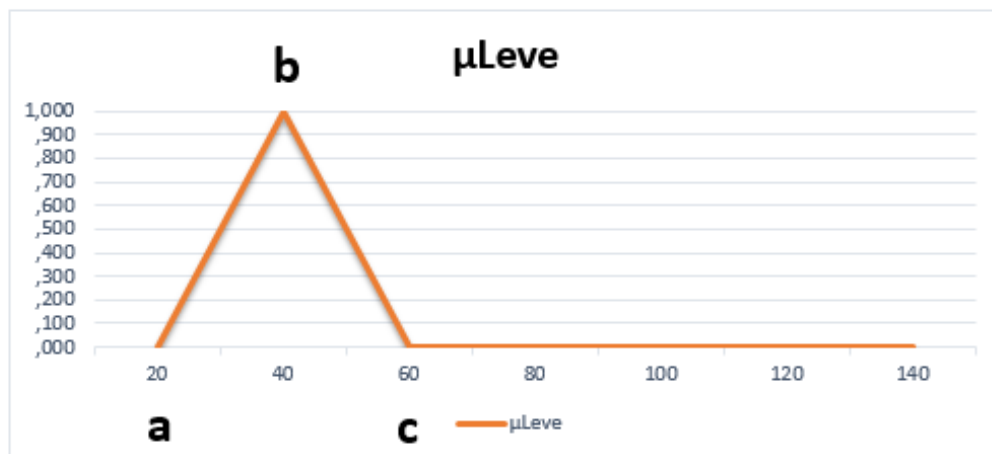


Figura 27: Gráfico de função de pertinência triangular, Observando pelo eixo y, vemos que a e c possuem valor 0 e b possui valor 1.

Nesta etapa tivemos como resultado os valores de pertinência para peso leve = 0,85 e peso médio = 0,075, o resultado para altura média = 1. Estes dados obtidos são chamados de valores “fuzzificados” que serão analisados na próxima etapa.

7.7 Inferência (Método de Inferência de Mandani min)

O processo de inferência (derivação de conclusão lógica de premissa conhecida ou verdadeira) recebe o dado “fuzzificado” que será processado de acordo com as regras especificadas pelo especialista humano. Esta etapa pode ser considerada como o coração da lógica Fuzzy. As etapas de “fuzzificação” e “defuzzificação” consistem em: a primeira recebe o valor de entrada e o transformam em um número “fuzzificado” como já verificamos acima; a segunda transforma esses valores “fuzzificados” em resultados de saída para que a pessoa que irá receber o retorno da informação obtida após todo processo da lógica possa entender.

Existem dois procedimentos para realização da inferência, Modus Ponens Generalizado (MPG) e Modus Tollens Generalizado (MTG).

- a) O MPG tem a regra: se x é A então y é B , ela permite implicar valor se x é A então y é B .
- b) O MTG tem a regra: se x é A então y é B , ela permite implicar valor se y é B então x é A . Abaixo seguem dois exemplos para melhor entendimento.

Exemplos:

MPG:

Se não tiver dinheiro, então não saio.

Não tenho dinheiro.

Então não saio

MTG:

Se o tempo não está bom, então está chovendo.

Está chovendo.

Então o tempo não está bom

Ambos os procedimentos possuem estruturas lógicas similares, porém o MPG permite encontrar o valor do antecedente já o MTG encontra o procedente. Desta forma escolhemos para nosso trabalho o procedimento MPG.

Utilizamos a mesma fórmula de pertinência triangular da etapa de fuzzificação para achar a função de pertinência de B' : $\mu(x) = \max (\min (X - LI / M - LI , LM - X / LM - M), 0)$.

A relação $\mu(x,y)$ é determinada por uma operação de implicação fuzzy, para isso utilizaremos Mandami $\min (\min (\mu A(x) , \mu B(y)))$. Cada regra estipulada pelo especialista humano possui uma estrutura: “se; então; condição”, a condição será chamada de valor de saída e para isso precisamos de um novo conjunto de valores para cada condição. Utilizando do exemplo para melhor entendimento podemos ter a seguinte regra: se peso é leve e altura é baixa então condição é normal.

O especialista humano irá definir os valores desses conjuntos de condições (também chamado de estado). A figura abaixo 28 demonstra os valores de saída (estado) para nosso exemplo.

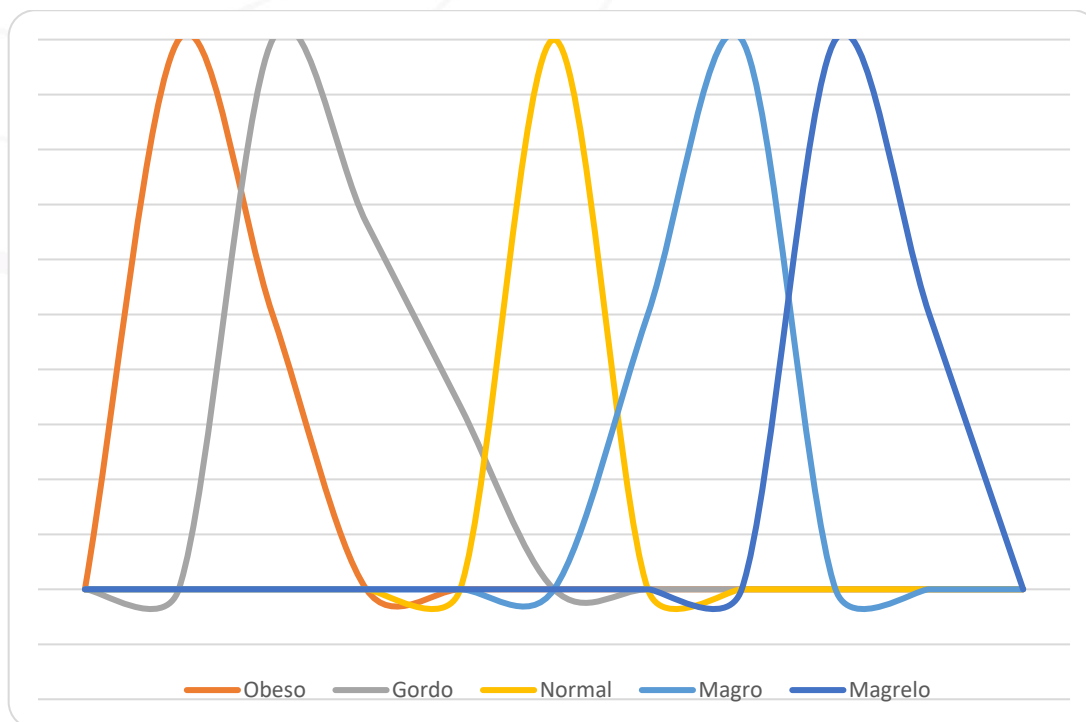


Figura 28: Gráfico de função de pertinência de saída.

Após determinar os conjuntos de saída podemos ver na tabela 1 abaixo um conjunto de regras do estilo “se-então”.

REGRAS	REGRA DISPARADA	MINIMO
se peso é leve e altura é baixo entao condicao é normal SENAO;	0	0
se peso é leve e altura é mediano entao condicao é magro SENAO;	2	0,85
se peso é leve e altura é alto entao condição é magrelo senao;	0	0
se peso é médio e altura é baixo entao condicao é gordo senao	0	0
se peso é medio e altura é mediano entao condicao é normal senao;	5	0,075
se peso é médio e altura é alto entao condicao é magro senao	0	0
se peso é pesado e altura é baixo entao condicao é obeso senao;	0	0
se peso é pesado e altura é mediano entao condicao é gordo senao;	0	0
se peso é pesado e altura é alto entao condição é normal	0	0

Podemos observar pela tabela acima que após os dados fuzzificados foram disparadas as regras 2 e 5. Para melhor entendimento vamos analisar agora somente a regra 2, utilizando o MPG teremos:

Se x é A1 e h é A2 Então y é B

Onde x representa o peso, h a altura, y estado.

Para A1 conforme a regra 2 o valor será "leve" com grau de pertinência 0,85. A2 será "médio" com grau de pertinência 1. Lembrando que o valor do grau de pertinência de A1 foi retirado do cálculo de pertinência fuzzificação para o peso e A2 da altura. B representa "magro", mas ainda não temos o seu grau de pertinência, que será obtido posteriormente. Ou seja:

Se peso (**X**) é leve(**A1**) e altura (**h**) é médio(**A2**) Então y é Magro (**B**)

Como foi utilizado no MPG o "E", será utilizado o menor valor, pois ele representa o mínimo. Neste caso utilizaremos o valor 0,85. Abaixo vamos inserir o valor na fórmula para melhor entendimento de como o valor 0,85 foi extraído:

Se peso (**X**) é leve (**A1**)(cujo valor é: **0,85**) e altura (**h**)(cujo valor é: **1**) é médio(**A2**) Então y é Magro (**B**). Logo o valor de peso "leve" é inferior ao de altura "médio", sendo assim ele será utilizado.

Utilizaremos a fórmula de Mandami min para determinar a relação de implicação, com a seguinte fórmula:

$$\text{Min} (\mu_A(X), \mu_B(Y))$$

Podemos observar na figura 29 abaixo que este conjunto representa os limites dos conjuntos identificados na inferência: Magro e Normal, somando os dois grupos temos como menor valor o -5 e maior valor 15, vamos então pegar os números que estão dentro deste limite (iremos implementar de 1 em 1 o intervalo para este conjunto, porém, quanto menor o intervalo mais preciso fica o resultado). Com isso conseguimos encontrar o conjunto universo [-5,-4,-3,-2,-1,0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15].

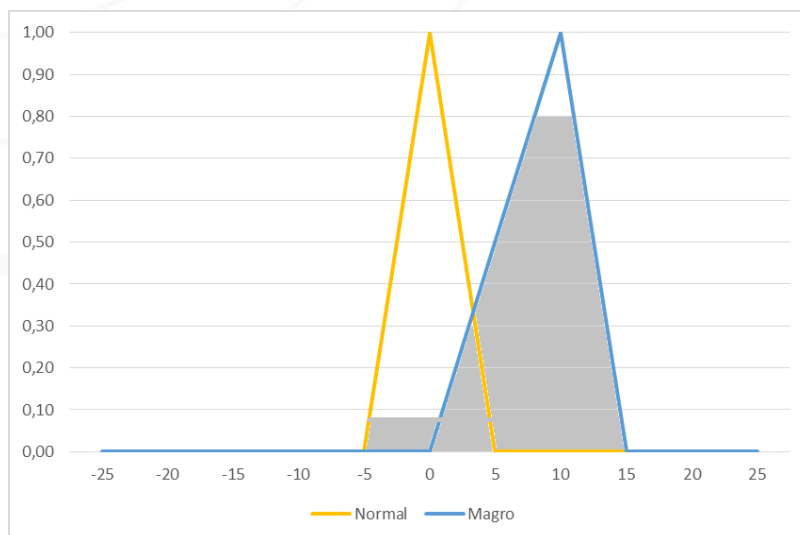


Figura 29: Conjunto universo obtido na inferência.

7.8 Defuzzificação (método da média dos máximos)

Esta etapa processará a variável da inferência transformando-a em uma variável linguística. O método da média dos máximos retorna o ponto que possui maior grau de pertinência. Por existir mais de um ponto de pertinência de valor máximo, para não pegar um valor aleatório e somado o conjunto é tirada a média desse conjunto. Utiliza-se a fórmula:

$$u^* = \sum_{m=1}^M \frac{u_m}{M}$$

A partir do conjunto universo na inferência $[-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]$, primeiramente calculamos a média deste conjunto, soma-se todos os valores do conjunto obtido e extrai a média (no caso deste exemplo será 5), após encontrarmos o valor da média pega-se os próximos 4 valores que estão posicionados acima do valor da média, (conforme nosso exemplo pegaremos o 5 por ser a média e mais 3 valores que estão acima dele no conjunto obtido que será o 6, 7 e 8). Após isto pegaremos estes 4 valores que encontramos (5, 6, 7, 8) e inserimos na fórmula, como podemos ver no exemplo abaixo:

$$\frac{5 + 6 + 7 + 8}{4} = 6,5$$

Assim finalizamos o raciocínio Fuzzy. Teremos então como saída para o usuário que o indivíduo deverá engordar 6,5kg.

8. PROCESSAMENTO DE LINGUAGEM NATURAL

8.1 Linguagem Natural

O *Homo Sapiens* é separado de outras espécies pela capacidade da linguagem (RUSSEL e NORVIG, 2013). A linguagem natural é utilizada como principal meio de comunicação entre seres humanos. Diferentemente das linguagens de programação (Java, C#, JavaScript), que se pode classificar como linguagens formais – que são projetadas para que não aconteça ambiguidade – a linguagem natural, seja ela escrita ou falada, pode conter ambiguidade, isto é, uma única sentença pode conter mais de um significado, que muitas vezes são difíceis de interpretar até por humanos, quanto mais por um computador. As linguagens naturais (e.g. português, inglês, espanhol) são difíceis de lidar, pois estão em constante evolução e mutação e, além disso, possuem gírias, dialetos e abreviações, além da já citada ambiguidade, o que se torna desafiador para o desenvolvimento de ferramentas que precisam compreendê-las. Para tanto, dentro do campo de estudo da IA, existe um conjunto de técnicas que são usadas no processamento destas linguagens.

8.2 O processamento da Linguagem Natural

O processamento de linguagem natural, ou PLN, é compreendido como uma área da computação que tem como objetivo extrair representações e significados mais completos em linguagem natural, de maneira inteligente. Segundo Malta (2019, apud ARANHA, 2007) é uma ciência que abrange um conjunto de técnicas e métodos que facilitam a análise textual por um computador. Antes, cabe recapitular que linguagem natural se compreende pelas tecnologias utilizadas por seres humanos para comunicação, expressão ou interação com outro ser humano.

De acordo com Barbosa (2007), a PLN utiliza-se conceitos linguísticos como classes de palavras, como verbo, substantivo, adjetivo, estruturas e classes gramaticais, conhecidas como Part-Of-Speech, bem como de situações mais complexas, como sinônimos, metáforas, anáforas, catáforas, semânticas, léxicas e sintática. A PLN é utilizada não apenas para textos digitados via documentos digitais, mas também para textos escritos em papéis e a língua falada, utilizando-se também a tradução automática de texto entre as linguagens, considerando, em alguns casos, os sentimentos e opiniões transmitidas pela linguagem.

Tal ferramenta é usada não apenas no meio científico, mas também por empresas e outros segmentos, independentemente de seu ramo de atuação, ganhando importância no mercado. Dentro da indústria, esta é aproveitada pelos profissionais na análise e pesquisa de informações, interação humano-computador e desenvolvimento de aplicações, e, quando combinada com ferramentas de mineração de dados ou inteligência artificial se torna uma ferramenta poderosa.

Todavia, nem sempre a linguagem humana, seja ela escrita ou falada, não é usada corretamente. Torna-se difícil a sua legítima interpretação, sua consideração ou seu real sentido, sendo um problema difícil para computadores. Descobrir quem fez o que, a quem, quando, onde, como e o porquê não basta. O ser humano expressa-se por seus sentidos e impressões em sua comunicação, sendo necessária também sua interpretação pela ferramenta. Scarpa (2017) relata que alguns problemas comuns são o uso de sinônimos, que não entendem o real sentido da palavra em determinada oração. Outro problema apresentado pela autora é a polissemia, que é o múltiplo significado de uma palavra, como é o caso da palavra manga, papel etc.

8.2.1 Pré-processamento

Primeiramente, para realização da análise léxica do documento, se faz necessário um pré-processamento dos textos recebidos, processo esse que analisa as entradas de cada palavra em uma oração, retirando todas as palavras que aparecem com determinada frequência, em um processo chamado stopwords (Scarpa, 2017). Para exemplificar, essa etapa é responsável por receber as palavras, separando-as em grupos, retirar palavras idênticas e separar em grupos menores. Após isso, é retirado seus prefixos e sufixos das palavras e, o seu resultado são armazenados em um banco de dados.

Malta (2019) ressalta que, para começar o pré-processamento, é necessário fazer um processo de stemming (stemização), em morfologia linguística é o processo de análise de uma palavra reduzindo-a a sua forma primitiva, como um radical, removendo seus prefixos (exemplo: “gato” para “gat”, “meninas” para “menin”. Como resultado, são gerados tokens (são cadeias de caracteres que possuem significado. Em linguagens de programação podem ser palavras reservadas, identificados, números, sinais, operadores etc. No caso apresentado refere-se ao resultado da extração dos radicais).

Após essa análise léxica, é necessário fazer a análise sintática da oração, sendo próprio deste procedimento durante a extração de palavras chaves. Ribeiro (2017) diz que é “a tarefa de identificação automática de um conjunto de termos que descreva satisfatoriamente o assunto de um documento”, sendo de suma importância para o estudo da PNL. Há algoritmos disponíveis para a realização desta tarefa, separados em duas categorias: os que utilizam de métodos estatísticos ou métodos estáticos. Para métodos estáticos, é realizado o processo *gazetter4*, que, segundo Ribeiro (2017) acontece “quando realiza a extração através do uso de uma lista pré-definida de palavras” e armazenada em um banco de dados.

8.2.2 Modelos vetoriais

Na etapa seguinte ao pré-processamento, é criado um formato interpretável pelos modelos gerados, uma vez que se trata de modelos matemáticos. Scarpa (2017) diz que o modelo mais utilizado são modelos vetoriais, qual considera como entrada cada termo apresentado. Neste modelo, é considerado apenas a sequência das palavras em formato matricial. Com a criação da matriz, as palavras são inseridas em vetores, de acordo com as suas entradas. Todavia, dependendo da quantidade de palavras inseridas em uma matriz pode alcançar dimensões muito grandes, tendo um custo de processamento muito elevado ao computador. Para solucionar tal problema, este modelo considera apenas uma pequena parcela de palavras por vez.

Para que o real sentido semântico das palavras seja mantido em uma frase, são utilizados vários algoritmos, como Deep Learning (Aprendizagem profundo de Máquinas) e de Machine Learning (Aprendizagem de Máquina), que contém técnicas para recuperação destes “corpus” (mistura aleatória de tópicos latentes, em uma estrutura não supervisionada) de dados, como a LSI (Latent Semantic Indexing ou Indexação Semântica Latente), LDA (Latent Dirichlet Allocation ou Alocação Latente de Dirichlet) etc.



Saiba mais

LSI: algoritmo de recuperação dos “corpus” em modelo vetorial, de acordo com a relevância e inter-relações de sua representação no documento, assumindo-se assim que o documento apresentado possui alguma estrutura subjacente ou latente no uso padrão das palavras. O LSI se utiliza de técnicas estatísticas para considerar a estrutura latente do conteúdo semântico.

LDA: algoritmo de modelo probabilístico generativo utilizada para modelagem de tópicos, apresentado em um artigo na revista Journal of Machine Learning Research por Blei, Ng e Jordan. Co. O Algoritmo analisa a probabilidade de uso das palavras distribuídas aleatoriamente no documento, atribuindo tópicos (ou coleções) para palavras associadas ao assunto. Utiliza o Modelo estatístico de distribuição do Matemático alemão Johann Peter Gustav Dirichlet.

A LSI é um conjunto de procedimentos estatísticos automatizados para medir quantitativamente a semelhança de significado entre duas palavras ou grupos de palavras. Assim, analisa-se qual a estrutura subjacente ou latente no padrão de uso das palavras. Já o LDA, é um algoritmo de abordagem bayesiana que utiliza o modelo matemático criado por Thomas Bayes, conhecido como teorema de Bayes, para cálculos probabilísticos. Em comparação prática, Souza (2019) diz que a LSI “tem seu processamento mais rápido apresentando maior variedade entre os termos contidos em cada tópico, todavia, o autor ressalta que em alguns casos, seus respectivos pesos possuem características distintas”. Em contrapartida a LDA possui seu processamento muito mais lento que o anterior, mas apresentou resultados mais promissores de similaridade que o anterior. Com a evolução e avanço de todos estes métodos, foi possível aplicá-los, tornando-se cada vez mais viável que uma máquina interaja mais eficazmente com um humano através de robôs de conversação que simulem a linguagem humana.

8.3 Chatbots

Robôs são dispositivos ou máquinas capazes de executar funções manuais e trabalhos, de maneira autônoma ou pré-programada. O robô só age conforme a resposta dos usuários, tomando este ou aquele caminho. Atualmente são utilizados na medicina, nas indústrias, em projetos tecnológicos e em diversas aplicações do conhecimento.

Chatbot, chatterbot ou simplesmente bot são robôs de conversação, ou seja, ferramentas de comunicação que simulam a fala humana e são capazes de dialogar com usuários. Segundo Gallert (2010) um “Chatbot é um sistema que utiliza linguagem natural para se comunicar com o usuário, respondendo como um humano.”.

Sua principal função é automatizar o atendimento ao cliente através de uma interação com um agente de entendimento humano. Possuem entrada e saída de texto, fazendo com que haja uma comunicação em linguagem natural, que é a forma como os humanos se expressam, com um sistema.

Para alguns autores, esses softwares podem ser classificados como: searchbots, que auxiliam a filtrar e procurar sites na internet, mailbots, utilizados para classificar e responder perguntas via e-mail, modbots, responsáveis por moderar fóruns de discussão online e, por fim, os chatbots, que funcionam como simuladores de conversação.

Levando em conta essa necessidade, o emprego de robôs de conversação na interatividade com clientes, tende, além da melhora do atendimento, automatização deste e reduções significativas nos custos das organizações.

Dado a compreensão dos conceitos e derivados que permeiam a criação e utilização de um Chatbot, no cenário atual muitos clientes buscam qualidade e praticidade quando se trata de atendimento, e em virtude disso, muitas empresas querem entregar essa melhoria de experiência, algo que viabiliza a fidelização do cliente com a empresa que dispõe destes serviços.

8.4 O serviço cognitivo LUIS (*Language Understanding*)

O chatbot necessita de um serviço inteligente, que consiga reconhecer padrões em frases e identificar a qual intenção a frase pertence. No mercado atual, existem algumas ferramentas que suprem essa necessidade, como o Watson, sistema cognitivo da IBM que oferece uma API de PLN para ser consumido por outras aplicações, bem como o LUIS que envolve uma série de serviços cognitivos oferecidos pela Microsoft™.

O serviço LUIS usa três aspectos principais para entender idiomas:

- a) **Enunciados:** um enunciado é uma entrada do usuário que seu aplicativo precisa interpretar;
- b) **Intenções:** uma intenção representa uma tarefa ou ação que o usuário deseja executar. É uma finalidade ou uma meta expressa no enunciado de um usuário;
- c) **Entidades:** uma entidade representa uma palavra ou frase dentro do enunciado que você deseja extrair.

O serviço LUIS, permite a definição de um conjunto de intenções provido pelo serviço, relacionadas a frases acrescentadas neste pelos autores da pesquisa. Essa ferramenta foi empregada com o propósito de permitir que o chatbot pudesse compreender a intenção da pergunta do usuário relacionada com a LGPD, verificando se esta intenção, que o usuário digitou por meio de plataformas de mensagens instantâneas, corresponde com alguma intenção que foi estabelecida dentro do serviço, possibilitando responder de forma mais natural.

No Bot LGPD ao receber um enunciado enviado pelo usuário, o serviço aciona um algoritmo baseado em técnicas de inteligência artificial que irá calcular e analisar as semelhanças entre o enunciado que foi digitado pelo usuário da solução através de uma interface, com as outras já existentes no serviço, identificando a qual intenção aquela frase mais se assemelha através de um sistema de pontuação.

O resultado dessa pontuação é melhorado com um processo de “treinamento”, permitindo ao serviço LUIS uma melhor identificação das particularidades de cada intenção. Um outro recurso importante desta plataforma de PLN são as Entidades, representadas por dados essenciais que foram digitados pelo usuário e que devem ser colhidas da frase para que o bot possa fazer a busca da informação desejada. No serviço LUIS existe algumas categorias de entidades:

- a) **Máquina aprendida:** Entidades aprendidas à máquina são aprendidas a partir do contexto. Use uma entidade de ML para identificar dados que nem sempre estão bem formatados, mas têm o mesmo significado. Uma entidade de ML pode ser composta por subentidades menores, cada uma das quais pode ter suas próprias propriedades.
- b) **Lista:** As entidades de lista representam um conjunto fixo e fechado de palavras relacionadas, juntamente com seus sinônimos. Uma entidade de lista não é

aprendida por máquina. É uma correspondência exata de texto. LUIS marca qualquer correspondência com um item em qualquer lista como uma entidade na resposta.

- c) **Regex**: extraem uma entidade com base em um padrão de expressão regular que você fornece. As entidades Regex são uma boa opção para os dados que são formatados consistentemente com qualquer variação que também seja consistente.
- d) **Pattern.any**: são entidades são espaços reservados de tamanho variável usados apenas na declaração do modelo de um padrão para marcar onde as entidades começam e terminam.

8.5 Microsoft Bot Framework

O Microsoft Bot Framework é uma ferramenta para a criação, testes, implantação e gerenciamento de chatbots. É um recurso do Serviço de Bot da plataforma em nuvem Azure, da Microsoft™, e nesta plataforma os serviços que normalmente seriam armazenados em servidores locais, agora poderão ser abrigados na nuvem.

Conjuntos de ferramentas como o Microsoft Bot Framework são chamados de SDK's (Software Development Kits – Kits para desenvolvimento de Software). Através da estrutura provenientes do SDK do Bot Framework e em conjunto com serviços de IA, neste contexto, torna-se possível a criação de robôs de conversação capazes de compreender a linguagem natural e interagir com pessoas através de perguntas e repostas. Dentre as funcionalidades do Microsoft Bot Framework, juntamente com o serviço de Bot do Azure, destacam-se:

- a) SDK para o desenvolvimento de robôs de conversação;
- b) Ferramentas que dão suporte completo para uma aplicação de chatbot;
- c) Serviço de envio e recebimento de mensagens e eventos do Bot Framework entre bots e determinados canais a serem utilizados;
- d) Serviços da plataforma Azure, como exemplo, serviços cognitivos da própria plataforma, bem como armazenamento em nuvem para soluções necessárias.

9. REDES NEURAIS ARTIFICIAIS

9.2 Neurônios biológicos

O neurônio (figura 30) é a célula do sistema nervoso responsável pela condução do impulso nervoso. O neurônio é constituído pelas seguintes partes: corpo celular (onde se encontra o núcleo celular), dendritos, axônio e telodendro.

Dendritos: são prolongamentos numerosos dos neurônios especializados na recepção de estímulos nervosos, que podem ser do meio ambiente ou de outros neurônios.

Axônio: é o um prolongamento dos neurônios responsável pela condução dos impulsos elétricos que partem do corpo celular, até outro local mais distante, como um músculo ou outro neurônio. Alguns axônios de neurônios de um humano adulto podem chegar a mais de um metro de comprimento.

Telodendro: é uma ramificação terminal do axônio, onde o impulso passa de um neurônio para o outro, ou para outro órgão.

O espaço entre o dendrito de um neurônio e o axônio de outro é o que se chama uma sinapse: os sinais são transportados através das sinapses por uma variedade de substâncias químicas chamadas neurotransmissores.

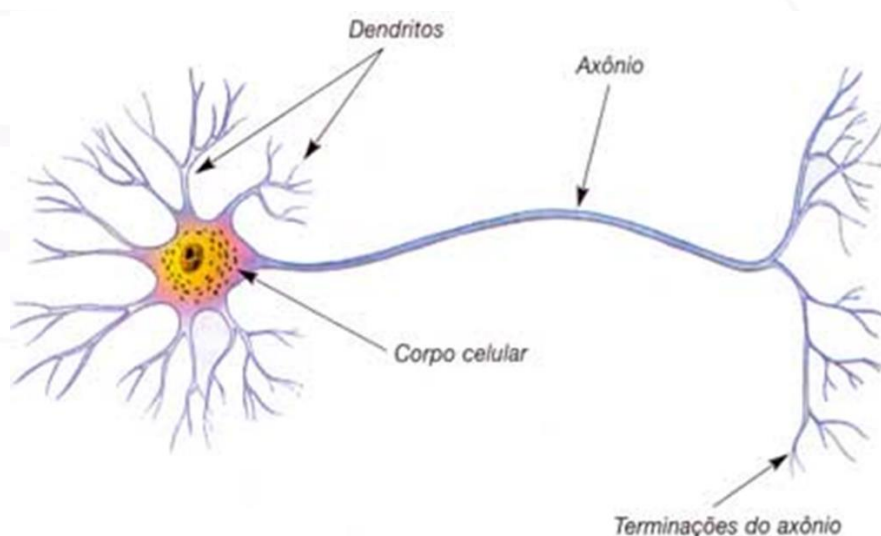


Figura 30: Neurônio biológico.

No quadro abaixo, visualizamos um quadro comparativo entre a capacidade de processamento do cérebro humano, formado por células neurais e um computador.

Requisito	Computador	Cérebro humano
Velocidade	Nanosegundo	Milisegundo
Tipo de processamento	Sequencial	Paralelo
Número de unidades de armazenamento	10e9 bits	10e14 sinapses
Número de unidades de processamento	+ - 1024	10e11
Velocidade de processamento	nanossegundos	milissegundos
Controle de processamento	Controle autocrático, centralizado	controle anárquico, distribuído
Armazenamento do conhecimento	estritamente relocável	adaptativo
Material	Metal e plástico	orgânico
Tolerância a falhas	Mínima, quando existe	boa

Os neurônios biológicos são de cinco a seis ordens de grandeza mais lentos que as portas lógicas dos microprocessadores.

9.3 Conceitos de Redes Neurais Artificiais

Segundo Haykin (2001), uma rede neural é um processador maciçamente e paralelamente distribuído, constituído de unidades de processamento simples, que têm a propensão natural de armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos: o conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem; forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

O histórico das redes neurais artificiais é dividido da seguinte forma:

- 1943 - McCulloch e Pitts apresentam um estudo sobre o comportamento do neurônio biológico com o objetivo de criar um modelo matemático para este;
- 1949 – Donald O. Hebb definiu o conceito de atualização de pesos sinápticos;

- c) 1951 – Marvin Minski constrói o 1º neuro-computador chamado “Snark”;
- d) 1957 – Surge o 1º neuro-computador a obter sucesso (Mark I Perceptron), desenvolvido por Frank Rosenblatt, Charles Wighman e outros;
- e) 1958 – Frank Rosenblatt implementa o 1º modelo de rede neural: o Perceptron (1 camada de entrada e 1 camada de saída). Este modelo não podia simular o comportamento de uma simples função XOR (ou-exclusivo);
- f) 1972 – Teuvo Kohonen definiu um novo modelo de rede neural, conhecido como “mapa auto-reorganizável de características”, com isto, introduziu um novo paradigma no estudo das redes neurais, o aprendizado não-supervisionado;
- g) 1982 – Surge o “modelo de Hopfield” desenvolvido pelo físico John Hopfield. Este modelo é caracterizado por ser do tipo “feedback”, isto é, uma conexão das entradas com as saídas. A saída será sempre a mesma para um mesmo padrão de entrada;
- h) 1986 – David Rumelhart e James McClelland publicam “Parallel Distributed Processing” estimulando o campo de pesquisa;
- i) 1987 – Ocorre a 1ª conferência de Redes Neurais Artificiais em tempos modernos, a IEEE (The Institute of Electrical and Electronics Engineers) International Conference on Neural Networks. Nasce a International Neural Networks Society (INSS).

A discussão em torno da Redes Neurais Artificiais é norteadada pela análise dos seguintes aspectos:

- a) Neurônio: unidade computacional básica da rede;
- b) Arquitetura: estrutura topológica de como os neurônios são conectados;
- c) Aprendizagem: processo que adapta a rede de modo a computar uma função desejada ou realizar uma tarefa.

A arquitetura de uma RNA é distribuída em camadas (figura 31).

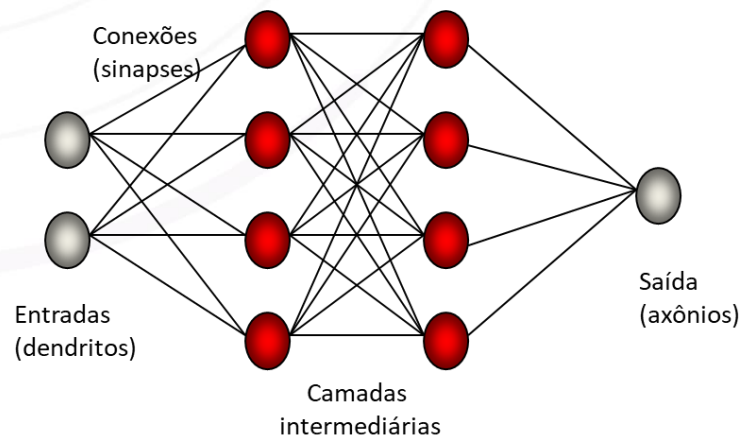


Figura 31: Arquitetura de uma RNA.

Uma RNA é composta por várias unidades de processamento, que geralmente são conectadas por canais de comunicação que estão associados a determinado peso. O comportamento inteligente de uma RNA vem das interações entre as unidades de processamento de dados.

Os elementos de processamento são os neurônios da rede, local onde é realizado todo o processamento. Tais elementos possuem duas qualidades importantes:

- Necessitam de informações locais. A saída do elemento de processamento é uma função dos pesos e das entradas;
- Elementos de processamento produzem apenas um valor de saída que é propagado através das conexões de elemento para outro (emissor para receptor), ou para fora da rede quando for um elemento da camada de saída.

9.4 Operação de uma RNA

A operação de uma unidade de processamento pode ser resumida da seguinte forma (figura 32):

- a) Sinais são apresentados à entrada;
- b) Cada sinal é multiplicado por um número (peso) que indica a sua influência na saída da unidade;
- c) É feita a soma ponderada dos sinais que produz um nível de atividade:

$$y = f\left(\sum_{i=1}^n x_i \cdot w_i\right)$$

- d) Se este nível de atividade exceder um certo limite, a unidade produz uma determinada resposta de saída.

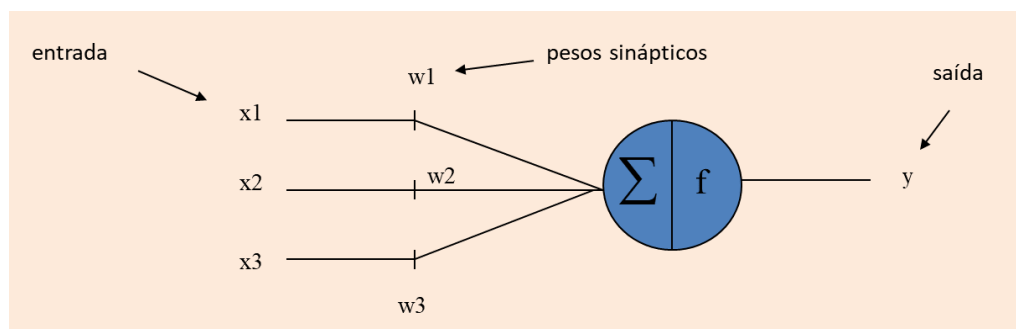


Figura 32: Operação de uma RNA.

9.5 Função de ativação

Um neurônio soma todos os pesos das entradas e passa o resultado para uma função de ativação (função de transferência) que será responsável por determinar a forma e a intensidade de alterações dos valores transmitidos de um neurônio a outro.

Exemplos de funções de ativação:

Função Degrau

$$f(x) = \begin{cases} 1 & \text{se } x \geq 0 \\ 0 & \text{se } x < 0 \end{cases}$$

Função Linear

$$f(x) = \begin{cases} 1, & x \geq \frac{1}{2} \\ x, & -\frac{1}{2} < x < \frac{1}{2} \\ 0, & x \leq -\frac{1}{2} \end{cases}$$

Função Sigmoidal

$$\int(x) = \frac{1}{1 + \exp(-\alpha x)}$$

9.6 Aprendizado de RNAs

As RNA's podem ser distinguidas com base na direção na qual o sinal flui:

- **Feedforward:** sinais se propagam em uma direção a partir da unidade de entrada, passando pela camada intermediária até a saída;
- **Feedback:** os sinais de entrada podem propagar da saída de qualquer neurônio para a entrada em um outro neurônio.

A habilidade de aprender de uma RNA ocorre através de um processo iterativo de ajustes aplicado a seus pesos, o treinamento. O aprendizado ocorre quando a RNA atinge uma solução generalizada para uma classe de problemas.

Algoritmo de aprendizado

A maioria dos modelos de Redes Neurais possui alguma regra de treinamento, onde os pesos das conexões são ajustados de acordo com os padrões apresentados, ou seja, “aprendem” através de exemplos. Desta forma uma RNA é capaz de extrair regras básicas a partir de dados reais.

Paradigmas de aprendizado:

- **Supervisionado:** utiliza-se um agente externo que indica à rede a resposta desejada para o padrão de entrada;
- **Não-supervisionado:** não existe um agente externo.

Existem diversos algoritmos de aprendizado específicos para determinados modelos de RNA's, diferenciando-se entre si pelo modo como os pesos são modificados.

O algoritmo de Backpropagation

Durante o treinamento com este algoritmo, a rede opera em uma sequência de dois passos. Primeiro, um padrão é apresentado à camada de entrada da rede. A atividade resultante flui através da rede, camada por camada, até que a resposta seja produzida pela camada de saída. No 2º passo, a saída obtida é comparada à saída desejada para esse padrão particular.

Se não estiver correta, o erro é calculado e propagado a partir da camada de saída até a camada de entrada, e os pesos das conexões das unidades das camadas internas vão sendo modificados conforme o erro é retropropagado. Com isso o erro vai sendo progressivamente diminuído.

Etapas para o desenvolvimento de aplicações de RNA

1- Coleta de dados

Os dados devem ser significativos e cobrir amplamente o domínio do problema e também as exceções.

2- Separação em conjuntos

Dados de treinamento e dados de teste (verificar performance sob condições reais de utilização).

3- Configuração da rede

- Seleção do paradigma neural
- Definição da topologia da rede
- Determinação de parâmetros do algoritmo de treinamento e funções de ativação

4- Treinamento

- Emprego do algoritmo de treinamento
- Ajuste do peso das conexões

5- Teste

Determinar a performance da rede, medida nesta fase, é uma boa indicação de sua performance real.

9.7 Aplicações com RNAs

Diversas são as aplicações com redes neurais artificiais:

- Análise, processamento e tratamento de sinais
- Controle de processos
- Classificação de dados
- Reconhecimento de formas e padrões
- Análise de imagens
- Avaliação de crédito e risco
- Visão, fala
- Previsão e modelagem
- Auxílio à decisão
- Robótica
- Mineração de dados

9.8 Exemplo de aplicação para resolução de problema de PREVISÃO

O órgão gestor de uma cidade está estudando a possibilidade de construir um dique de proteção da cidade contra eventuais inundações decorrentes de chuvas. Para estabelecer a altura do dique em função de uma análise custo x benefício, o órgão precisa dispor de uma função que forneça a probabilidade para cada vazão possível de ocorrer. Para desenvolver esta função vazão versus probabilidade, o órgão elaborou uma tabela com os dados de sete valores de vazões com suas probabilidades acumuladas (probabilidade de ocorrer um evento um valor menor ou igual).

Exemplo	Vazão (q) (m ³ /s)	Probabilidade (x100) (valor ≤ q)
1	1,50	0,02
2	2,00	0,08

3	2,50	0,18
4	3,00	0,50
5	3,50	0,65
6	4,00	0,88
7	4,50	0,96

O objetivo é o desenvolvimento de uma rede PERCEPTRON Multicamadas para realizar a aproximação da função probabilidade tendo como entrada a vazão. Considerar 2 (dois) neurônios na camada escondida e utilizar a função SIGMOIDAL como função de saída (função de ativação).

Quando se apresentam os sete exemplos de treino, tem-se um ciclo (época). Para treinar a rede são necessários muitos ciclos. Após cada ciclo se pode calcular o valor do erro médio quadrático de tal maneira que se fizermos um gráfico onde na horizontal se colocam os ciclos e na vertical o erro médio quadrático, tem-se a evolução do erro ao longo dos ciclos durante o processo de treinamento. Esta curva é construída é denominada de curva de aprendizagem.

Os pesos sinápticos das conexões da rede (determinados aleatoriamente no intervalo de $[-1,1]$ são (figura 31):

$$w_{10}^1 = 0,1$$

$$w_{20}^1 = -0,7$$

$$w_{10}^2 = -0,6$$

$$w_{12}^2 = -0,8$$

$$w_{11}^1 = -0,3$$

$$w_{21}^1 = 0,4$$

$$w_{11}^2 = 0,1$$

Onde:

w_{ij} ,

“i” é o índice do neurônio que recebe o sinal

“j” é o índice do neurônio que envia o sinal

α (ALFA): taxa de aprendizagem

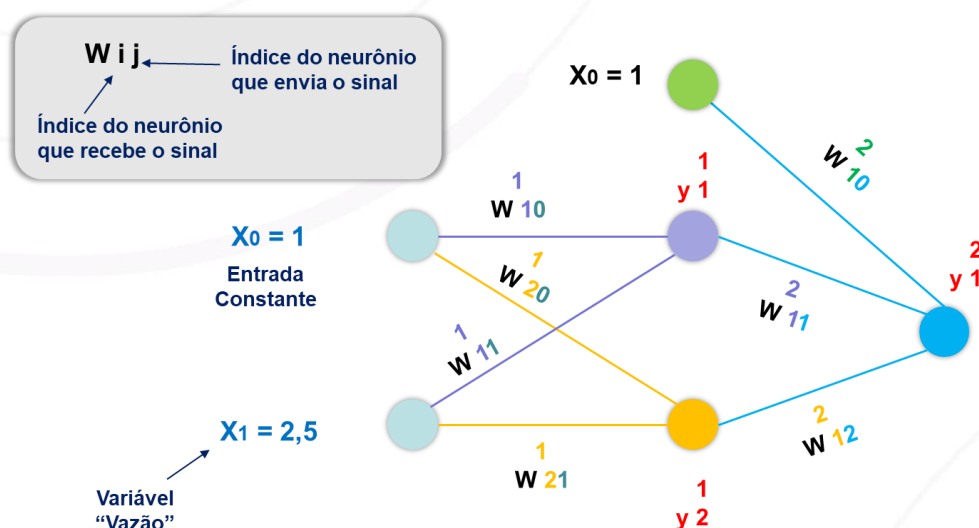


Figura 33: Arquitetura de RNA do exemplo.

Boa Prática de Arquitetura na Construção de uma RNA

Quando há uma Rede Neural com uma entrada constante, 1 entrada só, é uma boa prática criar sinteticamente uma segunda entrada com valor inicial constante igual a 1.

Além disso, pesquisas também indicam que é importante, na camada escondida (ou intermediária) acrescentar um neurônio a mais, com valor constante igual a 1. Esse neurônio é tecnicamente chamado de BIAS.

9.8.1 1ª Etapa – Propagação do sinal

Fase 1: Cálculo da entrada líquida para os neurônios da camada escondida (intermediária).

$$net_1^1 = \sum_{j=0}^1 w_{ij} \cdot x_j = w_{10}^1 \cdot x_0 + w_{11}^1 \cdot x_1$$

$$net_1^1 = (0,1) \cdot 1 + (-0,3) \cdot 2,5 = -0,65$$

$$net_2^1 = \sum_{j=0}^1 w_{ij} \cdot x_j = w_{20}^1 \cdot x_0 + w_{21}^1 \cdot x_1$$

$$net_2^1 = (-0,7). 1 + (0,4). 2,5 = \mathbf{0,30}$$

Fase 2: Cálculo da função de saída para os neurônios da camada escondida (intermediária)

$$y_1^1 = f^1(net_1^1) = \frac{1}{1 + e^{-(net_1^1)}} = \frac{1}{1 + e^{-(-0,65)}} = \mathbf{0,34299}$$

$$y_2^1 = f^1(net_2^1) = \frac{1}{1 + e^{-(net_2^1)}} = \frac{1}{1 + e^{-(0,30)}} = \mathbf{0,57444}$$

O número de Euler (e) é, aproximadamente: 2,718.

Fase 3: Cálculo da entrada líquida para o neurônio da camada de saída.

$$net_1^2 = \sum_{j=0}^2 w_{ij}. f^1(net_j^1) = w_{10}^2. f^1(net_0^1) + w_{11}^2. f^1(net_1^1) + w_{12}^2. f^1(net_2^1)$$

$$net_1^2 = (-0,6). 1 + (0,1). 0,34299 + (-0,8). 0,57444 = \mathbf{-1,02526}$$

Fase 4: Cálculo da função de saída para o neurônio da camada de saída.

$$y_1^2 = f^2(net_1^2) = \frac{1}{1 + e^{-(net_1^2)}} = \frac{1}{1 + e^{-(-1,02526)}} = \mathbf{0,26401}$$

Fase 5: Cálculo do ERRO para o neurônio da camada de saída.

$$\epsilon_i = d_i - y_1$$

$$\epsilon_i = d_1 - y_1^2 = (0,18) - (0,26401) = \mathbf{-0,08401}$$

O valor 0,18 (saída prevista) foi estabelecida por um especialista humano com domínio do problema.

9.8.2. 2ª Etapa – Retropropagação do erro da camada de saída para a camada de entrada.

Fase 1: Cálculo da sensibilidade para o neurônio da camada de saída.

Observação: As sensibilidades necessárias para ajuste dos pesos são calculadas da última para a 1ª camada. A sensibilidade de cada neurônio de saída (δ_i^2) é calculada pelo produto da derivada da função de ativação ($f^2(net_i^2)$) pelo respectivo erro de saída (e_i).

$$\delta_1^2 = y_1^2 \cdot (1 - y_1^2) \cdot e_1$$

$$\delta_1^2 = 0,26401 \cdot (1 - 0,26401) \cdot (-0,08401) = -0,01632$$

Fase 2: Cálculo da sensibilidade para os neurônios da camada escondida.

$$\delta_1^1 = f^1(net_1^1) \cdot w_{11}^2 \cdot \delta_1^2$$

$$\delta_1^1 = 0,34299 \cdot (1 - 0,34299) \cdot 0,1 \cdot (-0,01632) = -0,00037$$

$$\delta_2^1 = f^1(net_2^1) \cdot w_{12}^2 \cdot \delta_1^2$$

$$\delta_2^1 = 0,57444 \cdot (1 - 0,57444) \cdot (-0,8) \cdot (-0,01632) = -0,00319$$

Fase 3: Ajustes dos pesos que ligam a camada de saída a camada escondida.

α (ALFA): taxa de aprendizagem: o valor da taxa de aprendizagem da RNA pode variar entre 0 e 1. Quanto maior a constante, maior a mudança dos pesos. Quanto menor a constante, mais tênue será a mudança.

$$w_{ij}^2(novo) = w_{ij}^2(antigo) + \alpha \cdot \delta_i^2 \cdot f^1(net_j^1)$$

$$w_{10}^2(novo) = w_{10}^2(antigo) + \alpha \cdot \delta_1^2 \cdot f^1(net_1^1)$$

$$w_{10}^2(novo) = -0,6 + 0,7 \cdot (-0,01632) \cdot 1 = -0,61143$$

$$w_{11}^2(novo) = 0,1 + 0,7.(-0,01632).0,34299 = 0,09608$$

$$w_{12}^2(novo) = -0,8 + 0,7.(-0,01632).0,57444 = -0,80656$$

Fase 4: Ajustes dos pesos que ligam a camada escondida a camada de entrada.

$$w_{10}^1(novo) = 0,1 + 0,7.(-0,00037).1 = 0,09974$$

$$w_{11}^1(novo) = -0,3 + 0,7.(-0,00037).2,5 = -0,30064$$

$$w_{20}^1(novo) = -0,7 + 0,7.(0,00319).1 = -0,69777$$

$$w_{21}^1(novo) = 0,4 + 0,7.(0,00319).2,5 = 0,40559$$

Quando se apresentam os 7 EXEMPLOS de treino, tem-se um ciclo (época). Para treinar a rede são necessários muitos ciclos. Após cada ciclo se pode calcular o valor do ERRO MÉDIO quadrático verificando a evolução de erro ao longo dos ciclos durante o processo de treinamento. Esta curva é denominada CURVA DE APRENDIZAGEM.