



INTELIGÊNCIA ARTIFICIAL

Unidade 03 - Redes
Neurais Artificiais

Marco Antonio Sartori



Fonte: unsplash

INTELIGÊNCIA ARTIFICIAL

Unidade 03 - Redes
Neurais Artificiais

Marco Antonio Sartori

Copyright Centro Universitário Fundação Assis Gurgacz, Núcleo de Educação a Distância.

Todos os direitos são reservados. É proibida a reprodução, distribuição, comercialização ou cessão sem autorização dos detentores dos direitos. Esse livro foi publicado no Ambiente Virtual de Aprendizagem do Centro Universitário Fundação Assis Gurgacz, para leitura exclusiva online pelos alunos matriculados no Ensino a Distância. Os leitores poderão imprimir as páginas para leitura pessoal. Os direitos dessa obra não foram cedidos.

Autor: **Marco Antonio Sartori**

Título: **Inteligência Artificial**

Direção geral: Cleber Fagundes Ramos

Gerente de produção: Vanessa Aparecida Anderle Zaiatz

Design Instrucional: Anderson Julio de Souza

Projeto gráfico: Bruno Vinícius dos Reis

Diagramação: Júlia Tonin Bordin

1ª Edição

fag.edu.br/ead

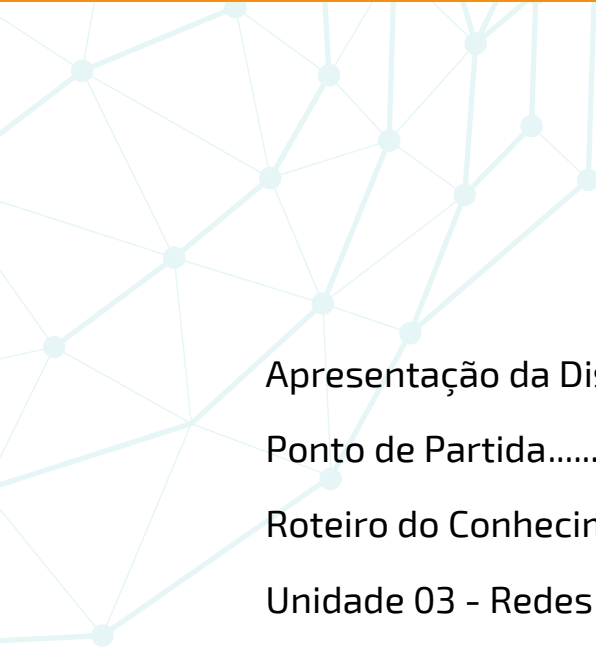
Av. das Torres, 500, Bloco 03

Fone: (045) 3321-3422

Bairro Santo Inácio | CEP: 85.806-095

Cascavel | Paraná | Brasil

SUMÁRIO



| | |
|--|----|
| Apresentação da Disciplina..... | 10 |
| Ponto de Partida..... | 11 |
| Roteiro do Conhecimento..... | 12 |
| Unidade 03 - Redes Neurais Artificiais | 14 |
| 3.1 - Introdução às Redes Neurais | 15 |
| 3.2 - Algoritmos Baseados em Redes Neurais | 28 |
| 3.2.1 - Back-Propagation..... | 32 |
| 3.2.2 - Mapas de Kohonen..... | 35 |
| 3.3 - Aplicação das Redes Neurais | 40 |
| 3.4 - Outros Métodos Bioinspirados..... | 48 |
| 3.4.1 - Algoritmos Genéticos..... | 49 |
| 3.4.2 - Lógica Nebulosa..... | 55 |
| O Que Aprendemos..... | 59 |
| Referências | 60 |

APRESENTAÇÃO DA DISCIPLINA

Caro (a) estudante, seja muito bem-vindo (a) a mais uma etapa do seu processo de formação profissional! É com muito prazer que apresento o e-book da disciplina de Inteligência Artificial! No decorrer deste e-book você irá conhecer as principais características, conceitos e aplicações da Inteligência Artificial. O principal objetivo deste material é permitir que você obtenha os conhecimentos básicos necessários para que você possa não apenas conhecer os principais conceitos da IA, mas também ter uma ideia de que formas poderá aplicá-la em sua carreira enquanto profissional da área de tecnologia. O mercado tem utilizado cada vez mais a IA, nos mais diversos segmentos e para as mais diversas finalidades. Por isso é tão importante que você conheça o tema e busque aprofundar os seus conhecimentos sobre ele.

Ao longo da disciplina, trataremos dos seguintes assuntos:

Unidade 01 – Introdução à Inteligência Artificial.

Unidade 02 – Métodos de resolução de problemas.

Unidade 03 – Redes neurais artificiais.

Unidade 04 – Métodos de aprendizagem.

PONTO DE PARTIDA

Caro (a) aluno (a), iniciaremos agora a terceira unidade do nosso e-book. Nesta unidade iremos falar sobre as Redes Neurais Artificiais, seus objetivos, aplicações, além de suas principais características.

Ao final desta unidade, você deve apresentar os seguintes aprendizados:

- Conhecer os principais conceitos ligados às redes neurais (RN);
- Conhecer alguns dos principais algoritmos que implementam os conceitos de RN;
- Conhecer algumas das principais formas de aplicação das redes neurais;
- Conhecer outros algoritmos baseados em métodos bioinspirados.

ROTEIRO DO CONHECIMENTO

No decorrer desta unidade, você conhecerá mais detalhadamente os principais conceitos ligados às Redes Neurais, e compreenderá por que elas são tão importantes dentro do contexto da Inteligência Artificial.

A seguir, você pode acompanhar todas as seções que serão trabalhadas ao longo da unidade:

- 3.1 - Introdução às Redes Neurais.
- 3.2 - Algoritmos baseados em Redes Neurais.
- 3.3 - Aplicação das Redes Neurais.
- 3.4 - Outros métodos bioinspirados.



UNIDADE 03 - REDES NEURAIS ARTIFICIAIS

3.1 - INTRODUÇÃO ÀS REDES NEURAIS

Antes de iniciarmos a nossa conversa sobre as Redes Neurais, é importante tratarmos do conceito de métodos bioinspirados, afinal, as Redes Neurais são um exemplo desse tipo de método.

Os métodos bioinspirados têm como principal característica o fato de que são baseados em padrões complexos encontrados na natureza. Esses padrões são estudados e utilizados para auxiliar na resolução de problemas, desenvolvimento de novas tecnologias ou evolução de tecnologias e sistemas já existentes.

Um exemplo de como isso acontece é a inspiração na forma como as formigas se comportam. Elas são capazes, por meio de uma varredura do ambiente, de definir o melhor caminho entre uma fonte de alimentos e o seu ninho. Esse comportamento serviu de inspiração para o desenvolvimento de algoritmos que permitem realizar o roteamento de veículos, por exemplo. Em decorrência do dinamismo inerente ao comportamento das formigas, o algoritmo é capaz de se ajustar a eventuais obstáculos no caminho, tendo como base informações fornecidas pelos usuários (FERRARI, 2011).

Há, atualmente, uma série de algoritmos bioinspirados desenvolvidos por diversas áreas de pesquisa que incluem as Redes Neurais, Algoritmos Genéticos, Sistemas Imunológicos Artificiais, Algoritmos de Enxames, Inteligência Coletiva, Sistemas Nebulosos, Química Artificial, dentre outras (FERRARI, 2011). A tabela 3.1 apresenta alguns exemplos de tecnologias baseadas na natureza.

| TECNOLOGIA | INSPIRAÇÃO |
|---------------------------------|--|
| Computação gráfica | Movimentação de exames ou bandos (peixes, insetos, pássaros), desenhos de formas naturais como filhas, nuvens, montanhas. |
| Computação quântica e molecular | Utilização de materiais naturais para superar limitações tecnológicas dos processadores e outros dispositivos utilizados atualmente. |

Tabela 3.1 - Tecnologias baseadas em elementos naturais
Fonte: Ferrari (2011)

De maneira resumida, os objetivos da computação natural são o desenvolvimento de técnicas para a solução de problemas complexos, o uso de computadores para sintetizar formas, comportamentos e padrões que sejam semelhantes àqueles encontrados na natureza, além da utilização de novos materiais naturais para fazer computação. Dentre as principais áreas de aplicação, podem ser destacadas a robótica, classificação de dados, predição de séries temporais, além do reconhecimento de padrões em grandes quantidades de dados.

Falando especificamente sobre as Redes Neurais (RN), podemos dizer que elas são uma técnica de Inteligência Artificial baseada na simulação do funcionamento do cérebro por estruturas de dados computacionais. As RN são recomendadas para situações em que as tarefas que precisam ser executadas demandam tolerância a falhas, flexibilidade, lógica imprecisa e paralelismo (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

Para compreendermos melhor como funcionam as redes neurais, é importante falarmos um pouco sobre o cérebro humano e o funcionamento de seus neurônios.

Muito já se sabe sobre o cérebro e seu funcionamento. Além disso, é de conhecimento de todos que o elemento principal do nosso cérebro é o neurônio, e que a capacidade de formação de redes de neurônios que trocam sinais por meio das sinapses é o fator responsável pela característica de aprendizado dos seres humanos. É por meio do aprendizado que os seres humanos podem lidar com informações imprecisas e formar generalizações sobre elas. A simulação do funcionamento dessas unidades de processamento – chamadas de neurônios – por meio de um computador permite, dentre outras coisas, um melhor entendimento dos mistérios do cérebro. (SILVA; PERES; BOSCARIOLI, 2016).

A simulação de um neurônio se dá por meio de um modelo baseado na fisiologia básica de um neurônio biológico, conforme apresentado na Figura 3.1 (a). De acordo com a figura, um neurônio é dividido em três partes: o corpo celular, os dendritos e o axônio. A aproximação entre o axônio de um neurônio e o dendrito de outro desencadeia um processo de reações, elétricas ou químicas, chamadas de sinapses.

A sinapse gera um impulso nervoso que define o fluxo de informações entre um axônio e vários dendritos. No corpo celular, os impulsos vindos dos dendritos são combinados e formam o sinal de entrada do neurônio. Dependendo da força desse sinal, o neurônio é capaz de impulsioná-lo pelo axônio, dando continuidade ao fluxo de informação. A sinapse pode ainda ser excitatória ou inibitória, o que indica que pode haver uma alteração nos sinais transmitidos por axônios e recebidos pelos dendritos (SILVA; PERES; BOSCARIOLI, 2016).

Quando falamos sobre neurônio artificial, estamos falando de uma função que mapeia entradas e saídas. Embora os primeiros modelos tenham sido propostos em 1943, o Perceptron, proposto em 1955 (figura 3.1 (b)) é o modelo mais interessante.

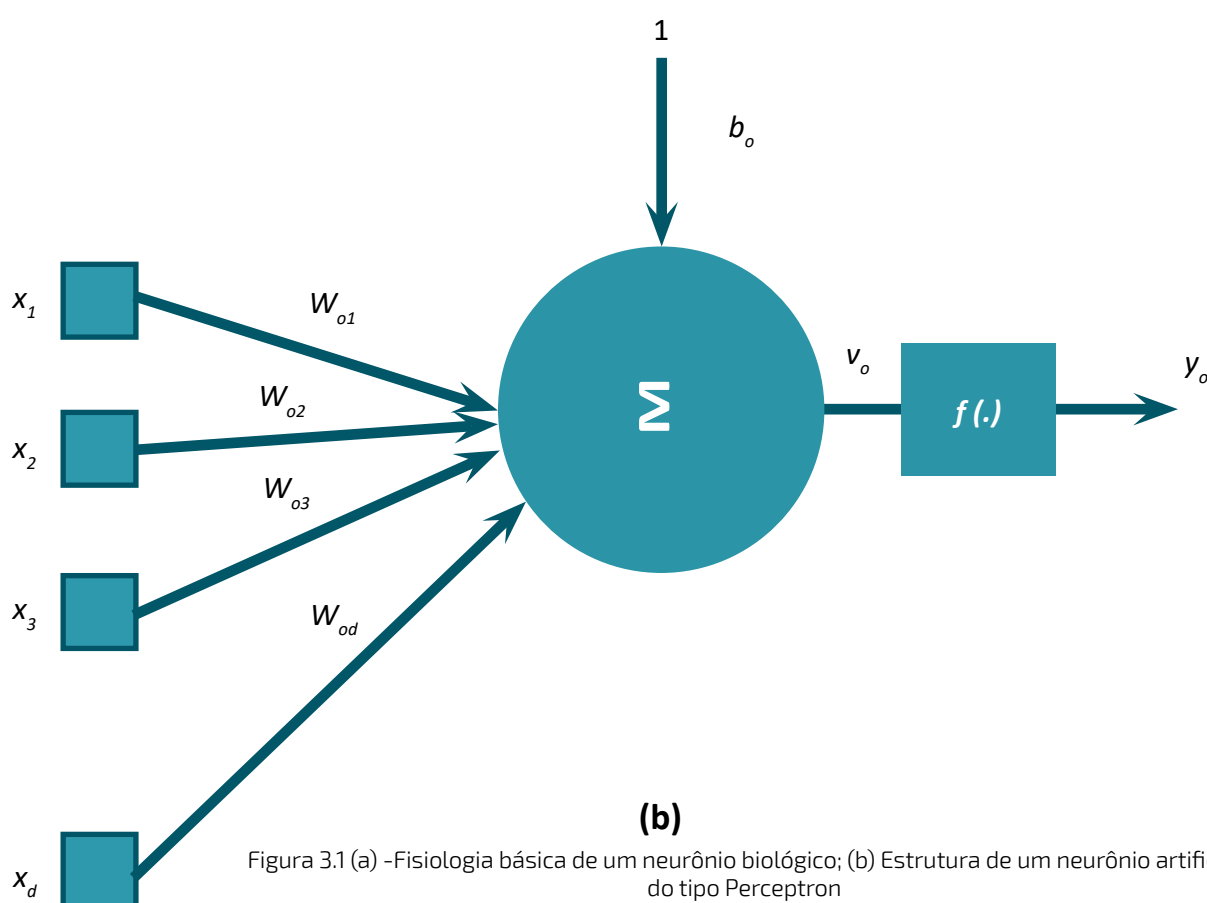
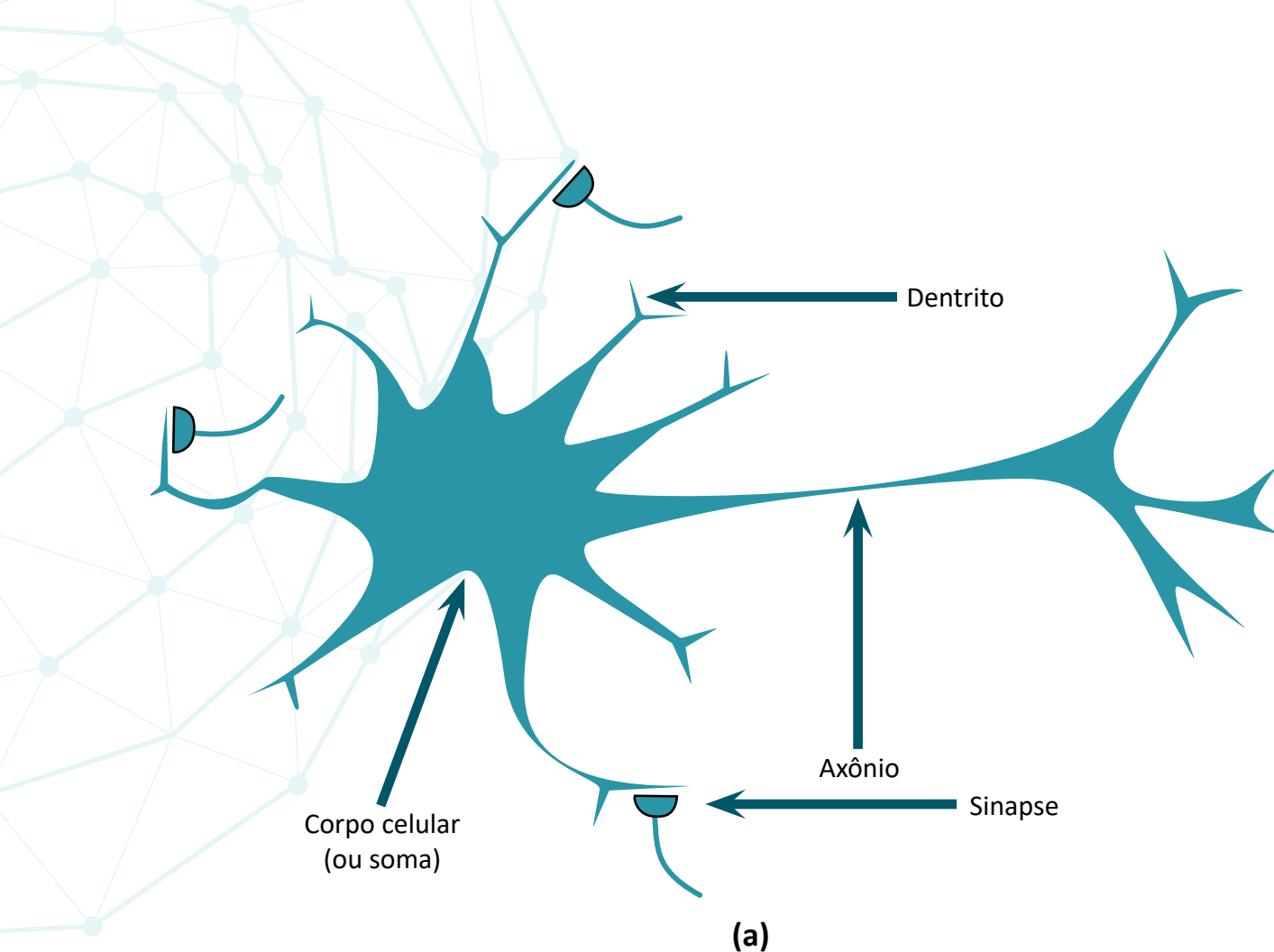


Figura 3.1 (a) -Fisiologia básica de um neurônio biológico; (b) Estrutura de um neurônio artificial do tipo Perceptron
Fonte: Silva, Peres e Boscaroli (2016)

É possível fazer um paralelo entre o funcionamento do neurônio artificial Perceptron e um neurônio biológico: os dendritos, ou cada atributo descritivo $\{xi1, xi2, \dots xij, \dots, xid\}$ de um exemplar de entrada \vec{X}_i , estão associados a um neurônio o por meio dos pesos sinápticos (\vec{W}_o); o núcleo do corpo celular é representado por uma função somatória que combina as entradas do neurônio; ao resultado do somatório ou campo induzido (U_o) é aplicada a função de ativação $f(.)$ gerando um valor y_o de saída no axônio do neurônio (SILVA; PERES; BOSCAROLI, 2016).

O principal ponto a ser considerado quando tratamos do uso de neurônios artificiais na resolução de problemas, é determinar cada uma das variáveis que compõem o modelo neuronal, de acordo com as especificações do problema. Cada um dos atributos descritivos de um exemplar de entrada é associado a cada uma das entradas ($X_j, j = 1 \dots d$) do neurônio, possibilitando a apresentação de uma informação referente ao problema, e, portanto, o valor da variável d depende do problema sob resolução; a função de ativação é escolhida pelo projetista que configura o neurônio artificial, observando também algumas características do problema; cada uma das sinapses (ou pesos sinápticos) ($W_{oj}, j = 1 \dots d$) é ajustada por meio de um algoritmo de treinamento (ou de aprendizado), de tal forma que a saída do neurônio y_o seja a saída esperada na resolução do problema (SILVA; PERES; BOSCAROLI, 2016).

Opcionalmente, um neurônio Perceptron pode ter uma entrada extra chamada *bias*, cujo valor (também chamado de valor de ativação) é sempre 1 e cujo peso associado é denotado por b_o . Na existência do bias, o algoritmo de treinamento faz o ajuste do peso com o ajuste do conjunto de pesos. A existência do bias aumenta ou diminui o sinal no campo induzido, dependendo se o valor do peso associado é positivo ou negativo, respectivamente.

Para entender como um neurônio pode resolver tarefas de classificação de dados, por exemplo, considere o problema de implementação de uma porta lógica AND. A figura 3.2 ilustra o problema informando cada uma das entradas da porta lógica e as respectivas saídas esperadas. Um neurônio projetado para resolver esse problema receberá e como entradas, ponderará essas entradas por meio da aplicação dos pesos sinápticos, as combinará no somatório e aplicará a função de ativação para produzir uma saída, que deve ser igual à saída desejada.

| X1 | X2 | Y |
|----|----|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Figura 3.2 - Função lógica AND representando um problema a ser resolvido por meio de um neurônio artificial do tipo Perceptron
Fonte: Silva, Peres e Boscaroli (2016)

Para que o neurônio possa ser preparado para resolver o problema proposto, é necessário definir para ele uma função de ativação. Uma função degrau, conforme ilustrado na figura 3.3, pode ser usada. Ela produzirá a saída 1 quando o campo induzido do neurônio for maior que 0, e produzirá o valor 0, caso contrário. A regra de processamento do neurônio pode então ser definida: o cálculo do sinal que entra no neurônio é o $v_o = \sum_j^d = 1(X_j * W_{oj}) + b_o$ e a saída do neurônio é dada por $y_o = f(v_o)$

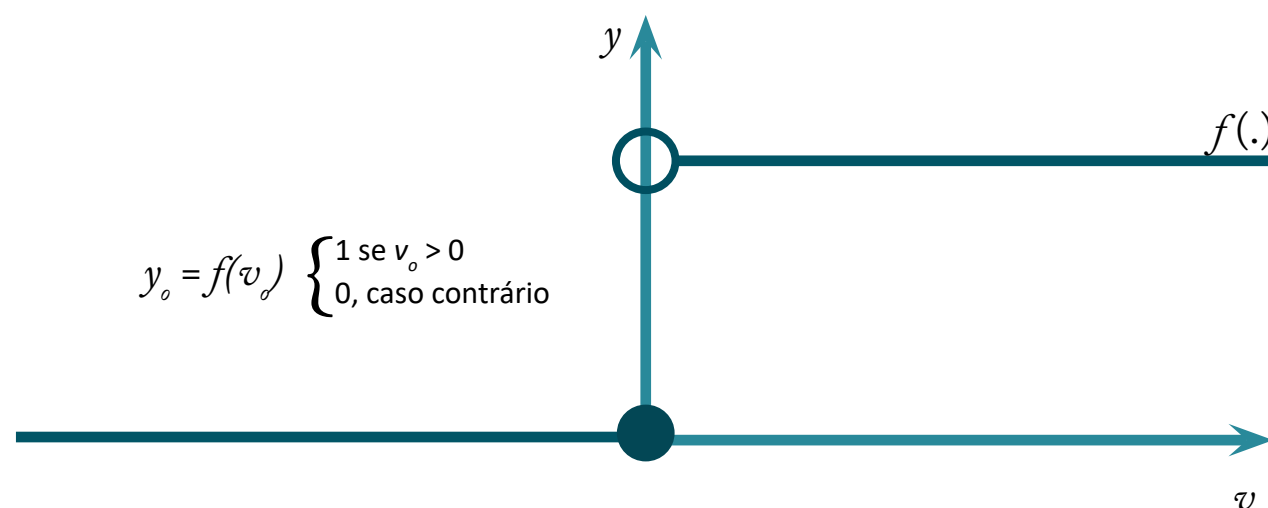


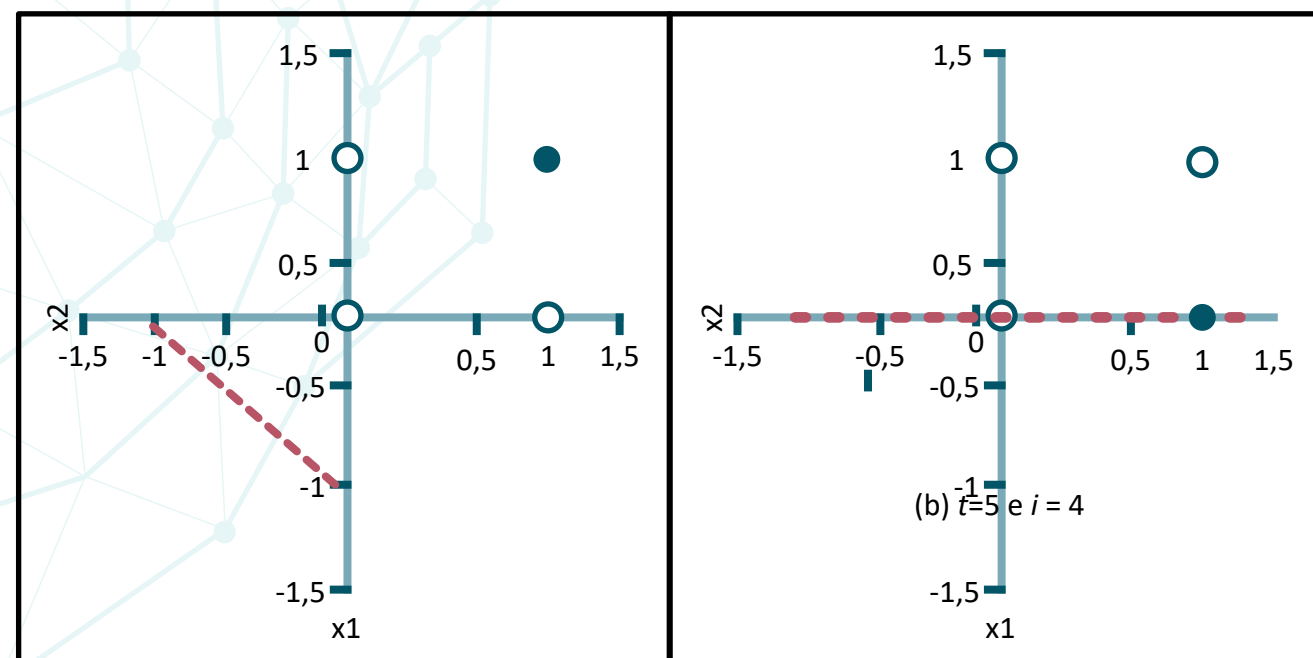
Figura 3.3 - Função de ativação do tipo degrau
Fonte: Silva, Peres e Boscaroli (2016)

Definidas a função de ativação e a regra de processamento do neurônio, resta agora definir como o ajuste dos pesos sinápticos (o aprendizado) deve ser feito. Há diferentes princípios de aprendizado que podem ser aplicados a um neurônio, e um deles ajusta o conjunto de pesos sinápticos iterativamente a partir do erro $e_o(t)$, produzido no tempo t , entre a saída desejada (y) e a saída produzida pelo neurônio (y_o). Desta forma, a regra de aprendizado do Perceptron é $\Delta w_{oj}(t) = n * x_j * e_o(t)$ em que $e_o(t) = y - y_o(t)$, x_j é a entrada associada ao peso w_{oj} sendo processada pelo neurônio no tempo t , e n é uma taxa de aprendizado que tem a função de ponderar o ajuste de pesos de forma que o aprendizado não se torne especializado no exemplar de entrada. Caso o **bias** esteja sendo usado no neurônio, o ajuste de pesos sinápticos deve também ser aplicado a ele por meio de $\Delta b_o(t) = n * e_o(t)$. A partir do cálculo das variações Δ , os pesos podem ser efetivamente atualizados por meio da adição de aos seus valores atuais (SILVA; PERES; BOSCAROLI, 2016).

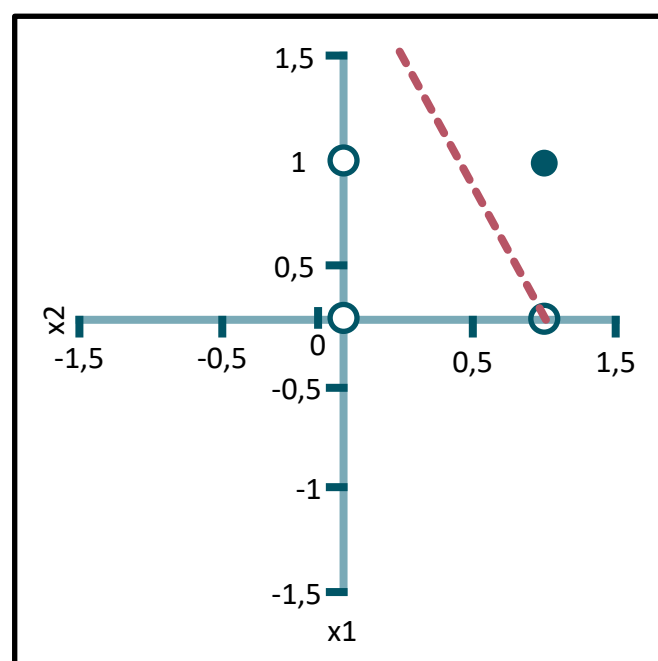
As atualizações dos pesos são realizadas com o objetivo de minimizar o erro produzido pelo neurônio e representam, portanto, o aprendizado (nesse caso, supervisionado). Esse processo é chamado também de treinamento, por se tratar de um processo iterativo que deve ser repetido muitas vezes (épocas) até que o erro produzido seja nulo ou suficientemente pequeno.

Os ajustes de pesos sinápticos são feitos em função do erro cometido na saída do neurônio: quando o erro é nulo, nenhum ajuste de pesos é necessário, pois o neurônio respondeu com a saída desejada. Um neurônio terá conseguido aprender a solução para o problema da porta lógica AND quando, dentro de uma época ou dentro de um limite de erro permitido, ele responder corretamente para todas as entradas presentes no conjunto de treinamento.

O processamento de um neurônio Perceptron pode ser interpretado também geometricamente como resultado de um hiperplano de separação do espaço d -dimensional dos exemplares de entrada. Se considerarmos o mesmo problema da porta lógica AND, para a qual o espaço dos exemplares de entrada é 2-dimensional (x_1, x_2) (figura 3.4), o processamento do neurônio resulta em uma reta de separação do espaço.



(a) $t=1$ e $i=4$



(c) $t=10$ e $i=4$

Figura 3.4 - Representação geométrica do problema da porta lógica AND: os círculos brancos representam a classe 0 e o círculo preto representa a classe 1; a reta pontilhada é a reta de separação que o neurônio representa
Fonte: Silva, Peres e Boscaroli (2016)

Para o treinamento realizado sobre o problema da porta lógica AND, os pontos de interseção da reta nos eixos x_1 e x_2 e, segundo as informações da figura 3.2, serão, respectivamente:

$$x_1 = -\frac{b}{w_1} = \frac{0,5}{0,5} = 1,0 \quad \text{e} \quad x_2 = -\frac{b}{w_2} = \frac{0,5}{0,2} = 2,5$$

(reta plotada na Figura 3.4(c))

Como ilustração, as retas de separação obtidas após a primeira e a segunda épocas do treinamento são mostradas na Figura 3.5(a) e (b) (SILVA; PERES; BOSCARIOLI, 2016).

Após o treinamento, a equação do hiperplano de separação pode ser usada para classificar exemplares novos, para os quais a informação de rótulo é desconhecida. O hiperplano de separação é considerado o modelo de predição.

O uso do neurônio Perceptron como modelo, até o momento está restrito a conjuntos com exemplares organizados em classes que podem ser separadas por uma reta (ou um hiperplano), ou seja, problemas linearmente separáveis. Em situações nas quais essa restrição não é obedecida, a aplicação do Perceptron como gerador de um modelo de decisão fica prejudicada.

Para exemplificar esse problema, considere a modelagem da porta lógica OU EXCLUSIVO, também chamada de XOR, ilustrada na Figura 3.7(a). A separação dos exemplares de cada classe (classe 0 e classe 1) é não linearmente separável, conforme exibido na Figura 3.7(b). Nessa figura, é possível observar que a separação das classes é feita a partir da combinação de duas retas e, para que isso seja possível, é necessário combinar mais de um neurônio Perceptron, criando uma rede de neurônios organizados, por exemplo, em múltiplas camadas – Multilayer Perceptron.

| x_1 | x_2 | y |
|-------|-------|-----|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

(a)

Figura 3.5 - (a) problema XOR;
Fonte: Fonte: Silva, Peres e Boscaroli (2016)

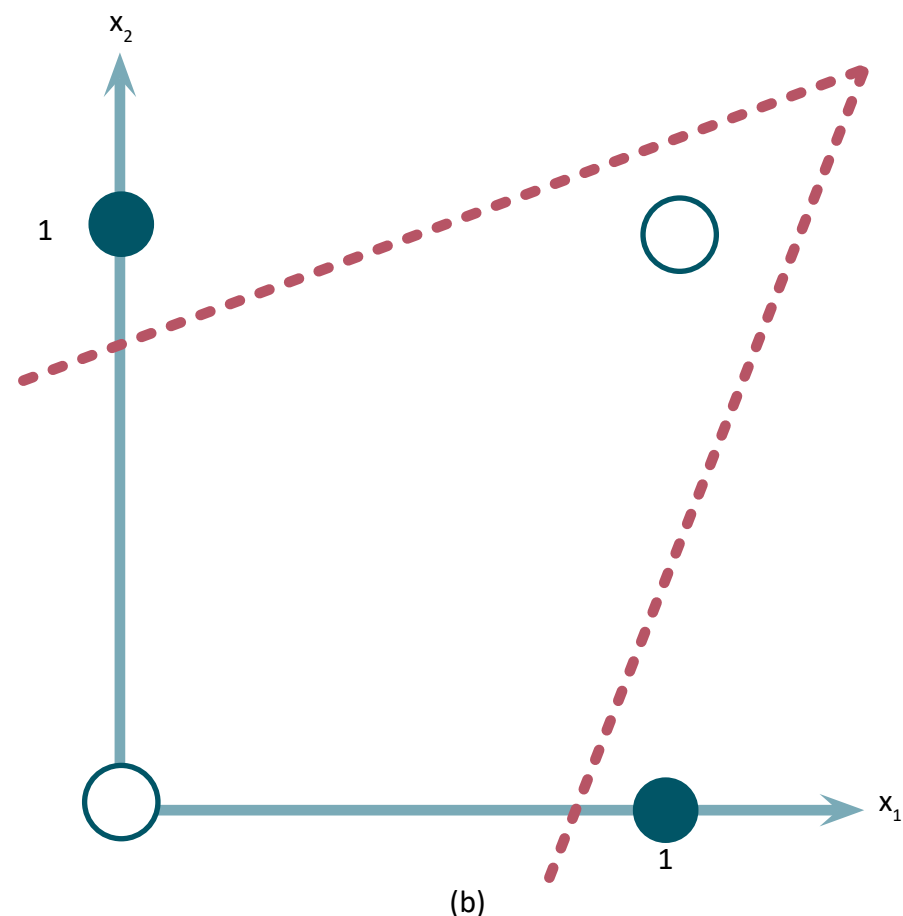


Figura 3.5 –(b) representação geométrica do problema XOR
Fonte: Silva, Peres e Boscaroli (2016)

A partir da possibilidade de combinação de neurônios em uma rede de múltiplas camadas, é possível a obtenção de estruturas mais complexas, como a arquitetura para rede neural ilustrada na figura 3.6(a). Nessa figura, aparece o conceito de camadas: a camada de entrada é formada por unidades responsáveis apenas por receber os valores descritivos dos exemplares processados na rede neural, não havendo processamento em tais unidades; na camada de saída, estão os neurônios que processam a informação descritiva dos exemplares, implementado mapeamentos especiais que possibilitam a resolução de problemas complexos. Essas camadas possuem uma série de neurônios que recebem sinal dos neurônios da camada anterior e enviam sinal para os neurônios da próxima camada, caracterizando um fluxo de informação do tipo feedforward (alimentação para frente).

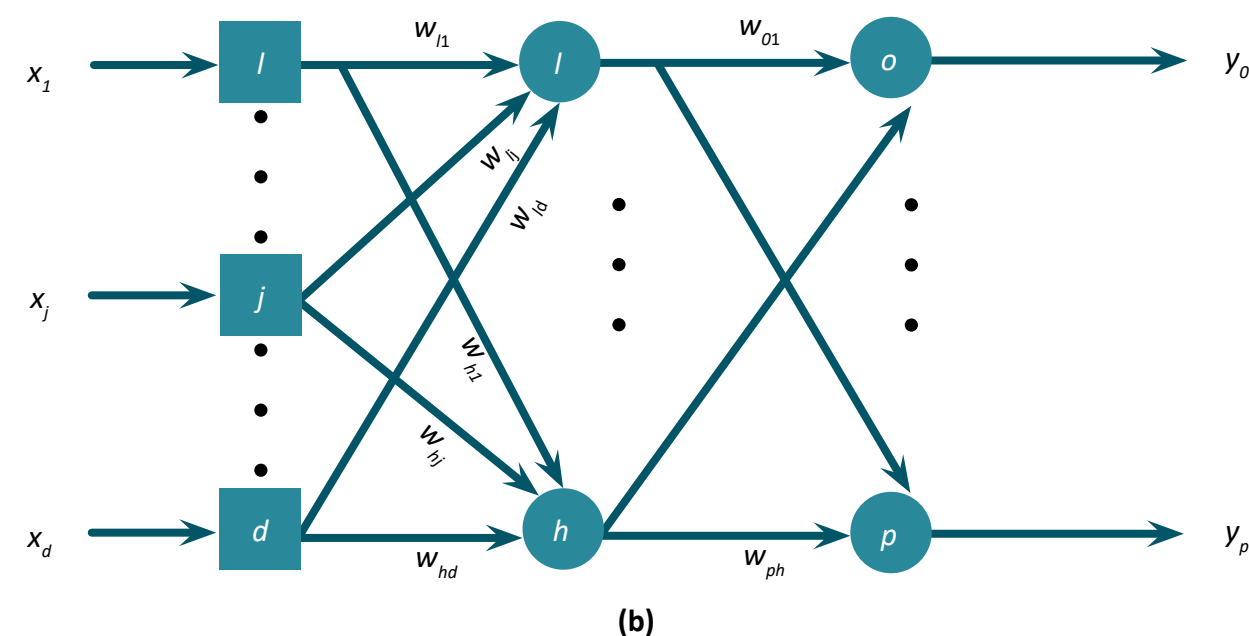
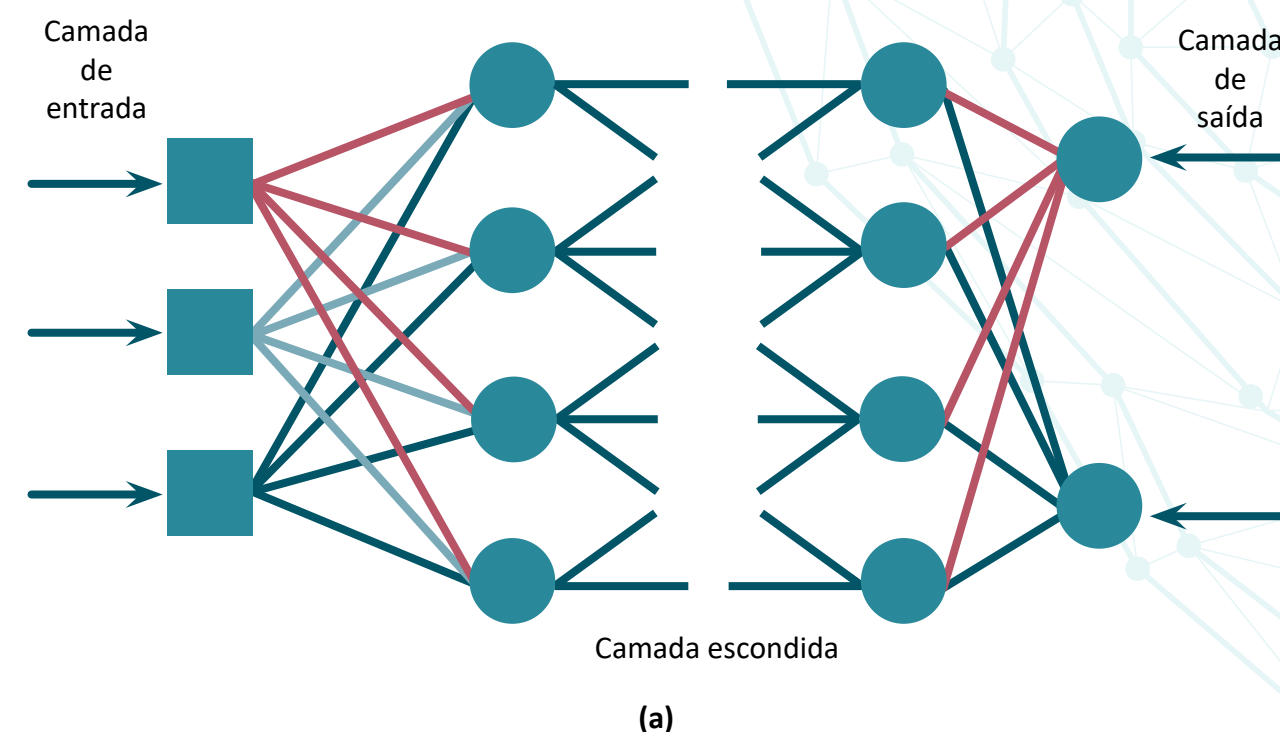


Figura 3.6 - (a) Arquitetura genérica de uma rede neural de múltiplas camadas; (b) Arquitetura da rede Multilayer Perceptron com uma única camada escondida

Originalmente, o ajuste de pesos do Perceptron é feito com base na informação sobre o erro cometido na sua saída e, também, considerando a informação de entrada na rede neural. Agora, nessa nova arquitetura, o erro (calculado na camada de saída) não é mais associado apenas às entradas da rede, mas também ao processamento realizado nas camadas escondidas; não há como calcular o erro ocorrido nos neurônios das camadas escondidas de forma direta, uma vez que não existe a resposta desejada para tais neurônios. Sendo assim, uma nova forma de implementar o ajuste de pesos precisa ser utilizada, e o algoritmo para ajuste de pesos nessa arquitetura é conhecido como aprendizado por retropropagação do erro (em inglês, error backpropagation) (SILVA; PERES; BOSCARIOLI, 2016).

O funcionamento desse algoritmo de treinamento é composto por três fases: na primeira, um exemplo do conjunto de dados usado para treinamento da rede neural é apresentado na camada de entrada da rede e é propagado para as próximas camadas por meio do processamento nos neurônios (neurônios do tipo Perceptron) até que, na camada de saída, o resultado final é produzido; na segunda fase, a resposta produzida na camada de saída é comparada com a resposta desejada, o erro obtido é calculado e, então, propagado de volta até a camada de entrada, propiciando que as alterações necessárias nos pesos das sinapses sejam calculadas; na terceira fase, o ajuste de pesos é, de fato, realizado a partir das alterações calculadas na segunda fase (SILVA; PERES; BOSCARIOLI, 2016).

As topologias das redes a serem utilizadas dependem do problema que precisa ser solucionado e da representação escolhida para os dados. De maneira geral, as aplicações de data mining, por exemplo, recebem na camada de entrada os dados pré-processados de cada registro de um banco de dados. Os registros então são processados pela rede de forma que seja produzida uma saída cuja natureza varia de acordo com a aplicação.

Em uma RN, os pesos das conexões entre os neurônios determinam os padrões que essa rede pode reconhecer. Nesse caso, é na fase de treinamento que os pesos das conexões serão definidos. Dentre as formas de atribuição dos pesos das conexões, podem ser destacados o aprendizado supervisionado e o aprendizado não supervisionado (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

Em RN treinadas com aprendizado supervisionado, a entrada corresponde aos atributos previsores enquanto a saída do modelo corresponde ao atributo-alvo do problema. Desta forma, o algoritmo de aprendizado pode estimar o erro, ou distância, entre a saída produzida pela rede e a saída desejada. Em função desse erro, o algoritmo ajusta os pesos das conexões existentes na rede tornando a saída gerada tão próxima quanto seja possível da saída desejada. Modelos neurais deste tipo são muito úteis em geral para reconhecimento de padrões e tarefas de mineração que envolvam predição (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

Por outro lado, as RN treinadas com aprendizado não supervisionado são adequadas para tarefas em que se deseja organizar os dados, assim como ocorre em tarefas de clusterização. Nesse caso, durante o treinamento, não há informação relativa à saída desejada associada a cada registro. Em vez disso, os pesos das conexões são definidos única e exclusivamente com base nos registros de entrada. A partir desses registros, as similaridades entre os registros de entrada são usadas pelo algoritmo de aprendizado para determinar os grupos naturalmente existentes no conjunto de dados (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

3.2 - ALGORITMOS BASEADOS EM REDES NEURAIS

No tópico anterior falamos sobre alguns dos principais conceitos relacionados às Redes Neurais (RN) e aos neurônios artificiais que as constituem. Falamos também sobre os neurônios do tipo Perceptron que, embora não sejam os únicos tipos de neurônios artificiais, constituem o principal deles. A partir de agora, falaremos um pouco mais sobre outras abordagens baseadas nos conceitos de Redes Neurais, tratando especificamente de dois algoritmos amplamente utilizados no campo da IA: Back-Propagation e Mapas de Kohonen.

Antes, porém, de falarmos dos algoritmos, é importante abordarmos alguns conceitos fundamentais sobre a composição das RN, tais como as topologias e algumas particularidades relacionadas à forma como os algoritmos aprendem. Para que você compreenda melhor, é a topologia da rede que determina, por exemplo, a composição estrutural dessa rede, incluindo o número de camadas, assim como a quantidade de neurônios presentes nas camadas de entrada, intermediária e de saída. A escolha da topologia da rede deve considerar que há uma dependência entre a topologia adotada, o problema que precisa ser solucionado e a representação escolhida para os dados (PALMIERE, 2016).

Quando falamos em data mining, por exemplo, as aplicações recebem, na camada de entrada, os dados pré-processados de cada registro de um banco de dados. Os registros então são processados pela rede de forma que seja produzida uma saída cuja natureza varia de acordo com a aplicação. Na RN, os pesos das conexões entre os neurônios determinam os padrões que essa rede pode reconhecer. Nesse caso, é na fase conhecida como treinamento que os pesos das conexões serão definidos (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

O treinamento da RN (figura 3.7) consiste em fazer com que ela adote pesos e valores baseados em amostras de dados de entrada. A ideia é que, a partir de qualquer outra amostra de dados futura, a rede consiga classificá-la corretamente. Desta forma, ao receber uma amostra de dados e um resultado que se pretende obter, a rede vai modificando os pesos dos dados para que possa alcançar o resultado esperado. A partir daí, ao receber futuros conjuntos de dados, a rede devidamente treinada sempre irá produzir um resultado igual ou, pelo menos, próximo ao resultado que se espera (PALMIERE, 2016).

Durante o processo de treinamento, é interessante realizar a divisão das amostras de dados, utilizando uma parte para o treinamento propriamente dito e a outra parte para a validação desse treinamento. Cada vez que um determinado conjunto de dados é apresentado à rede, temos a chamada época de treinamento. Esses treinamentos podem ser classificados como sendo supervisionados, não supervisionados, com reforço, por lote de padrões e padrão por padrão (PALMIERE, 2016).

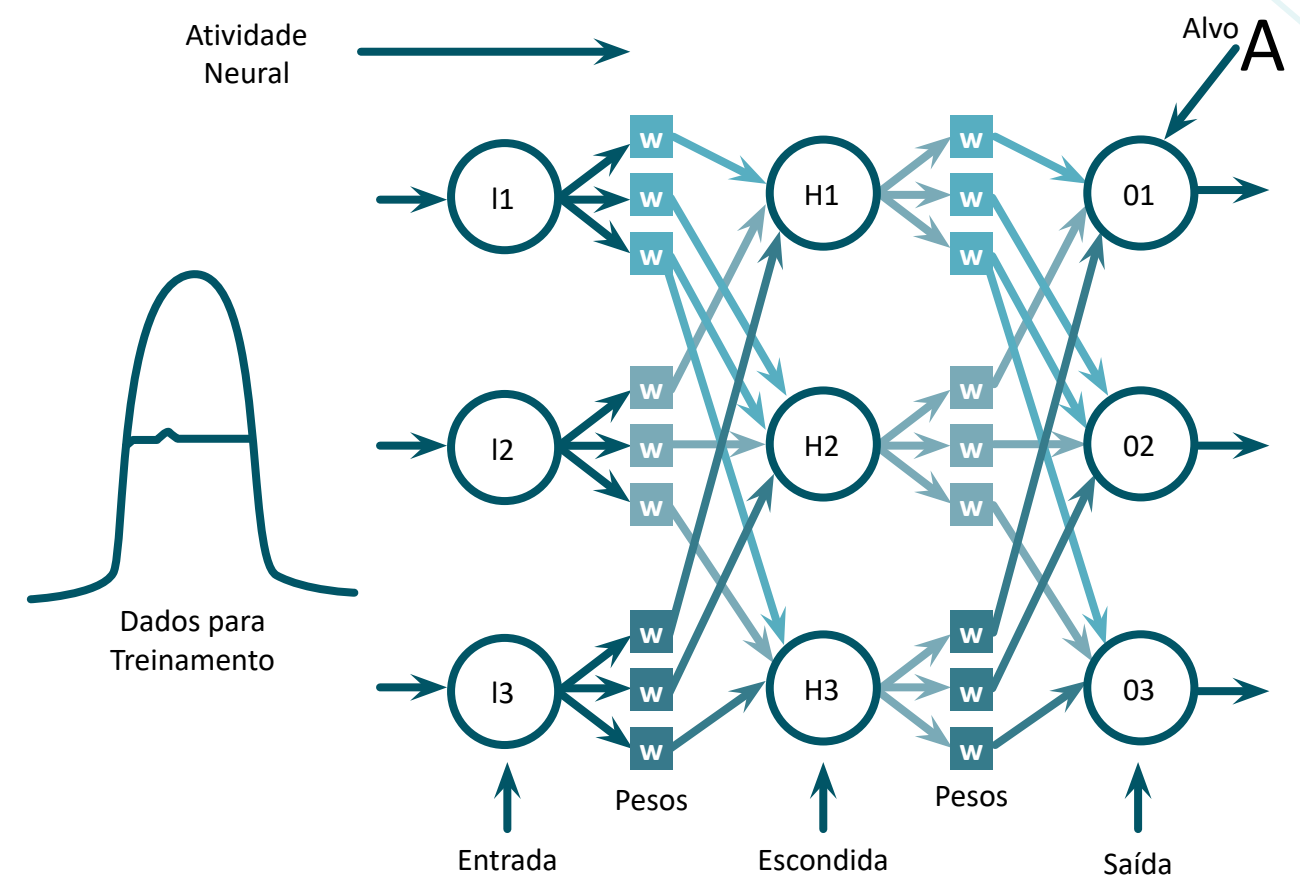


Figura 3.7 - Ilustração do processo de treinamento de uma Rede Neural Multi Layer Perceptron
Fonte: Goldschmidt, Passos e Bezerra (2015)

Aprendizado supervisionado: Nesse tipo de aprendizado, é necessário, além da amostra de dados, que o algoritmo tenha as informações da saída esperada. A partir daí os pesos dos dados serão ajustados pelo algoritmo para que aqueles resultados sejam alcançados;

Aprendizado não supervisionado: Nesse modelo, o algoritmo deve se organizar de forma automática a partir dos conjuntos de dados que são submetidos a ele, de modo que possa identificar padrões ou similaridades entre os dados ou conjuntos de dados. Após treinado, o algoritmo tentará classificar novos conjuntos de dados aos padrões identificados durante a fase de treinamento.

Aprendizado com reforço: Esse tipo de aprendizado é similar ao aprendizado supervisionado. A diferença é que o aprendizado com reforço busca ajustar os seus pesos tendo como base as informações originadas a partir de interações anteriores com o sistema mapeado. A ideia aqui é validar ou reforçar os resultados encontrados.

Lote de padrões: Nesse tipo de aprendizado a rede passa a buscar por padrões apenas após a exposição a todo o conjunto de amostras de dados.

Padrão por padrão: Esse tipo de treinamento é bastante útil em situações em que o sistema mapeado sofre alterações ao longo tempo, exigindo, portanto, um número maior de amostras. Os pesos, nesse caso, são atualizados após a apresentação de cada amostra e estas, por sua vez, são descartadas após a conclusão do treinamento.

Em RN treinadas com aprendizado supervisionado, a entrada corresponde aos atributos previsores enquanto a saída do modelo corresponde ao atributo-alvo do problema. Desta forma, o algoritmo de aprendizado pode estimar o erro, ou distância, entre a saída produzida pela rede e a saída desejada. Em função desse erro, o algoritmo ajusta os pesos das conexões existentes na rede tornando a saída gerada tão próxima quanto seja possível da saída desejada. Modelos neurais deste tipo são muito úteis, em geral, para reconhecimento de padrões e tarefas de mineração que envolvam predição (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

Por outro lado, as RN treinadas com aprendizado não supervisionado são adequadas para tarefas em que se deseja organizar os dados, assim como ocorre na clusterização. Nesse caso, durante o treinamento, não há informação relativa à saída desejada associada a cada registro. Em vez disso, os pesos das conexões são definidos única e exclusivamente com base nos registros de entrada. A partir desses registros, as similaridades entre os registros de entrada são usadas pelo algoritmo de aprendizado para determinar os grupos naturalmente existentes no conjunto de dados (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

As RN possuem inúmeros algoritmos que permitem a resolução dos mais diversos tipos de problemas, principalmente aqueles que exigem o reconhecimento de padrões. Dentre os principais exemplos, temos o Back-Propagation, os Mapas de Kohonen, Adaline, Perceptron, dentre outros. Cada um desses algoritmos possui uma especialidade diferente. Falaremos com um pouco mais detalhe agora sobre o Back-Propagation e sobre os Mapas de Kohonen.

3.2.1 - BACK-PROPAGATION

O Back-Propagation (retropropagação) é um algoritmo de aprendizado supervisionado cuja aplicação se dá em tarefas de classificação, regressão ou previsão de séries temporais. Seu objetivo principal é minimizar a função de erro entre a saída gerada pela RN e a saída real desejada, utilizando o método gradiente descendente. Esse algoritmo se destaca dos demais porque ele opera com várias camadas, além de poder resolver problemas classificados como não-linearmente separáveis, coisa que outros algoritmos não podem fazer (LANHELLAS, 2013).

Os problemas não-linearmente separáveis são aqueles em que não possível separar classes distintas apenas traçando uma reta no plano eixo cartesiano bidimensional. Note na figura 3.8 que há classes representadas por X e por \bullet . A diferença entre o Back-Propagation e os demais algoritmos é que ele consegue reconhecer os dois tipos de classes e separar os dados em cada uma delas, enquanto os outros não (LANHELLAS, 2013).

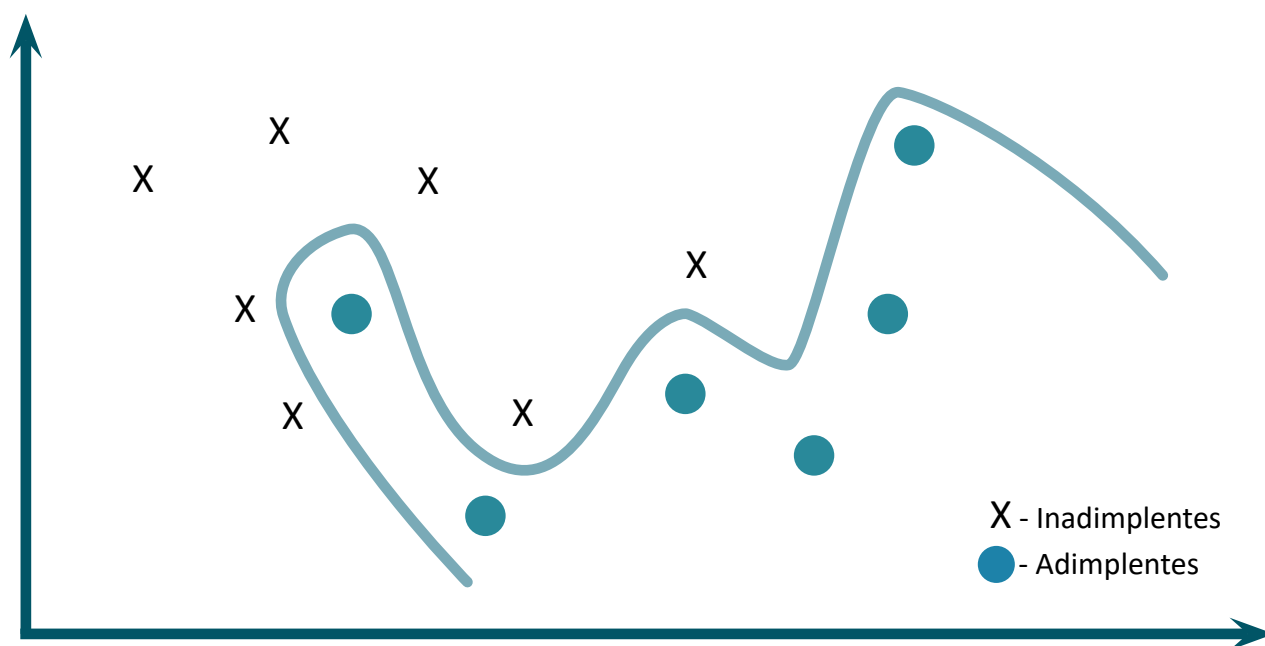


Figura 3.8 - Ilustração de um problema não linear
Fonte: Elaborado pelo autor (2019)

A topologia de uma RN não linear cujo comportamento seja codificado pelo Back-Propagation geralmente é composta por uma camada de entrada, uma camada de saída e um número arbitrário de camadas intermediárias (figura 3.9). Cada neurônio de uma camada, com exceção da camada de entrada, se encontra conectado a todos os neurônios presentes na camada imediatamente anterior à sua. A fase de treinamento do algoritmo consiste em duas etapas realizadas para cada registro de entrada apresentado: processamento para a frente e processamento para trás.

No processamento para a frente, correspondente à primeira etapa, o fluxo do processamento parte das unidades na camada de entrada em direção às unidades na camada de saída. Os neurônios da camada de entrada recebem os valores no padrão de entrada. Em seguida, a função de ativação é aplicada produzindo a saída de cada neurônio desta camada. Nesta etapa, os pesos sinápticos não são alterados. Os valores de entrada dos neurônios nas demais camadas são calculados pela regra de propagação escolhida durante a modelagem da rede. Já a segunda etapa prossegue com o repasse dos sinais de erro da direita para a esquerda, calculando o gradiente local associado a cada neurônio não pertencente à camada de saída, e ajustando os pesos das conexões associadas a ele (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

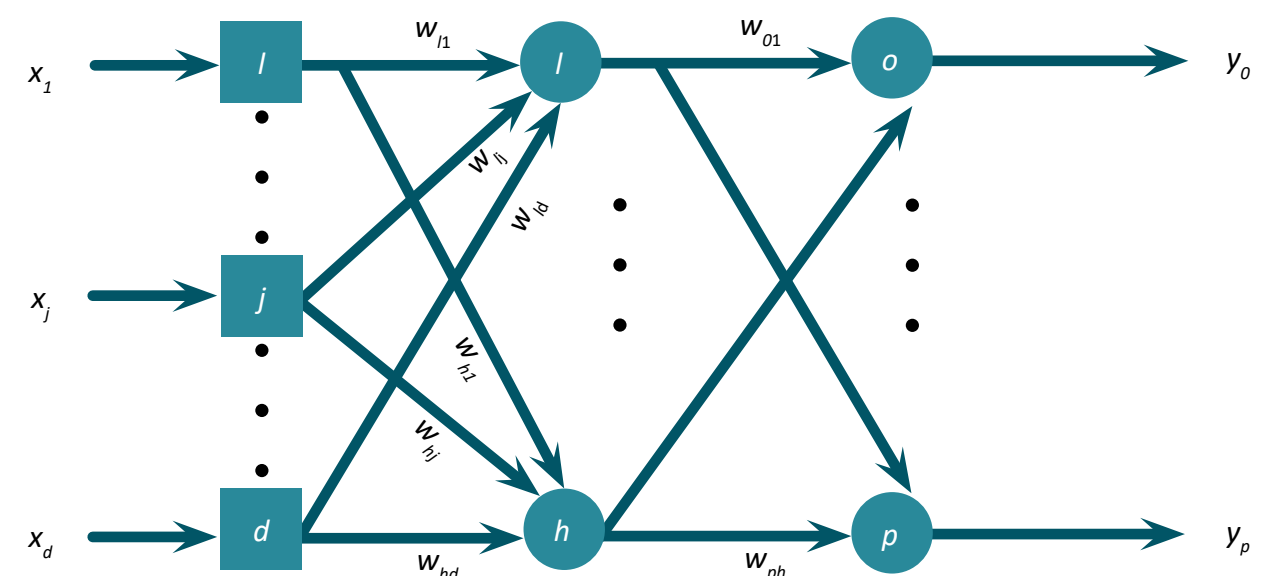


Figura 3.9 - Representação de uma Rede Neural Artificial com Back-Propagation
Fonte: Silva, Peres e Boscardoli (2016)

Geralmente são duas as condições de parada do treinamento de uma RN com Back-Propagation: um número máximo de interações definido pelo usuário ou a convergência da rede. A convergência da rede ocorre quando o somatório dos erros dos neurônios da camada de saída tende a zero, ficando abaixo de um limiar de tolerância de erro especificado pelo usuário antes do início do treinamento.

No fim do treinamento de uma rede, os pesos das conexões entre os neurônios representam o conhecimento gerado por ela. Esse conjunto então pode ser utilizado pela rede para processar novos casos e, em função do conhecimento descoberto, apresentar resultados. Uma vez que o conhecimento armazenado pela matriz de pesos de uma RN treinada não pode ser interpretado diretamente pelo homem, a qualidade do desempenho desta rede deve ser avaliada por meio de experimentos a fim de verificar a adequação deste conhecimento na realização da tarefa desejada (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

APRENDA UM POUCO MAIS

O algoritmo Back-Propagation possui uma importância particular no contexto das Redes Neurais, sendo utilizado na resolução de diversos tipos de problemas. Para conhecer um pouco melhor esse algoritmo e compreender por que ele é tão importante, você poderá fazer a leitura deste texto:



[Algoritmo Backpropagation Parte 2 – Treinamento de Redes Neurais](#)

Para complementar a sua experiência de aprendizado, aproveite também para pesquisar um pouco mais sobre Deep Learning e as suas possibilidades de aplicação. Bons estudos!

3.2.2 - MAPAS DE KOHONEN

Os Mapas de Kohonen (figura 3.10) pertencem à classe das RN Auto-Organizáveis (Self-Organized-Maps - SOM). Nesse tipo de rede, o treinamento é do tipo não supervisionado, geralmente baseado em uma forma de competição entre os elementos processadores. As principais tarefas que podem ser realizadas por esse tipo de algoritmo estão a sumarização e o agrupamento.

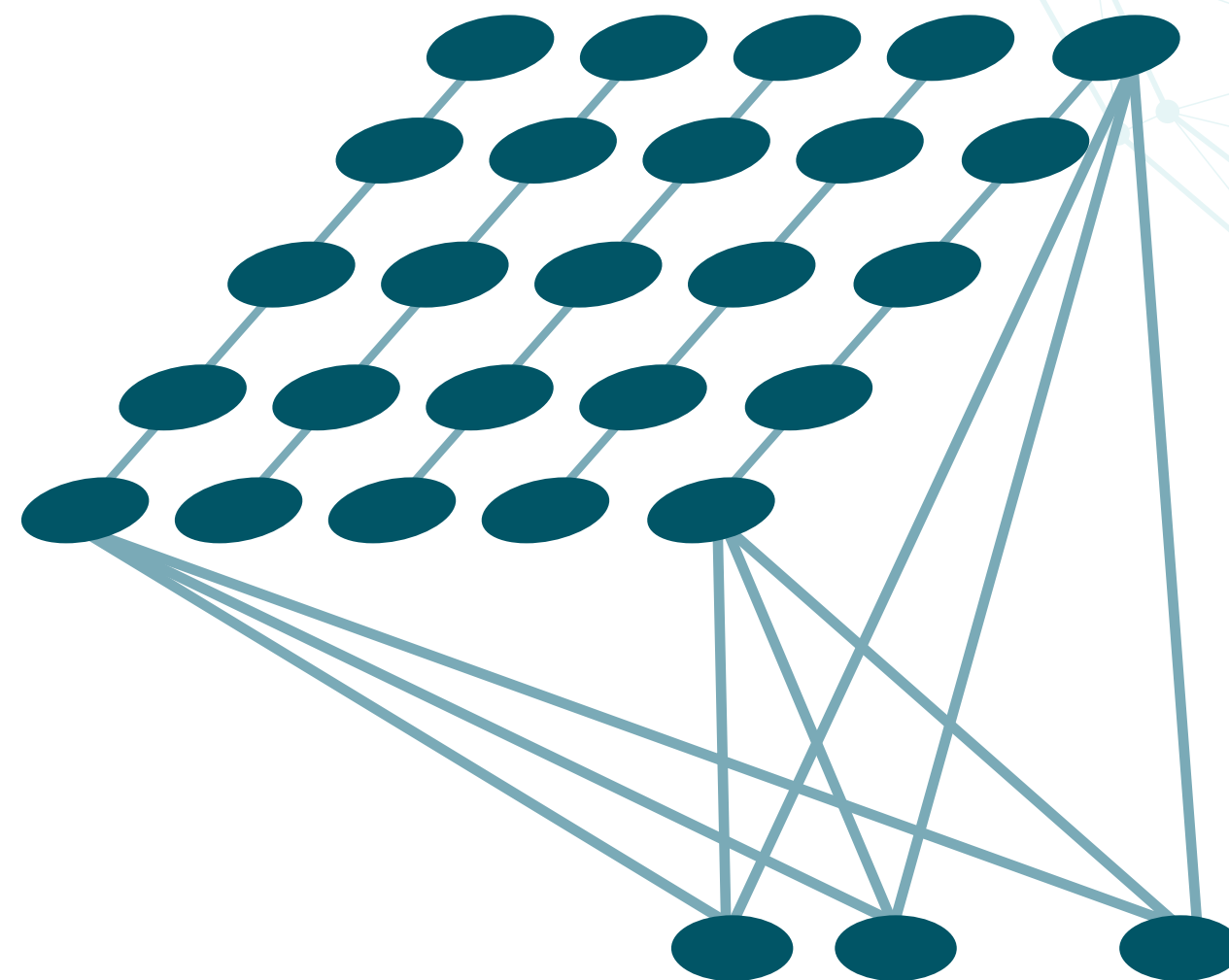


Figura 3.10 - Ilustração gráfica de um Mapa de Kohonen
Fonte: Goldschmidt, Passos e Bezerra (2015)

Nesse tipo de rede, existe uma unidade de entrada para cada um dos componentes do vetor de entrada. Cada uma das unidades de saída representa uma estrutura chamada de cluster. Durante o processo de auto-organização do mapa realizado pelo algoritmo, o cluster cujo valor for o mais próximo dos padrões de entrada pré-determinados é definido como o vencedor. A partir daí aquele cluster e os seus vizinhos têm os seus pesos atualizados.

O método mais comum de aprendizado das Redes Auto-Organizáveis é denominado aprendizado por competição (Competitive Learning), que consiste em dividir o conjunto de padrões de entrada em grupos inerentes aos dados. Para tanto, em sua abordagem mais simples, este método considera que os neurônios de saída da rede competem entre si, resultando em apenas um neurônio vencedor (com maior ativação). Esse tipo de rede geralmente possui algumas características particulares, conforme descrito a seguir (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

Topologia: Os neurônios são dispostos em uma única camada estruturada em uma (figura 3.11a) ou duas (figura 3.11b) dimensões. Todos os neurônios dessa camada recebem todos os valores dos padrões de entrada.

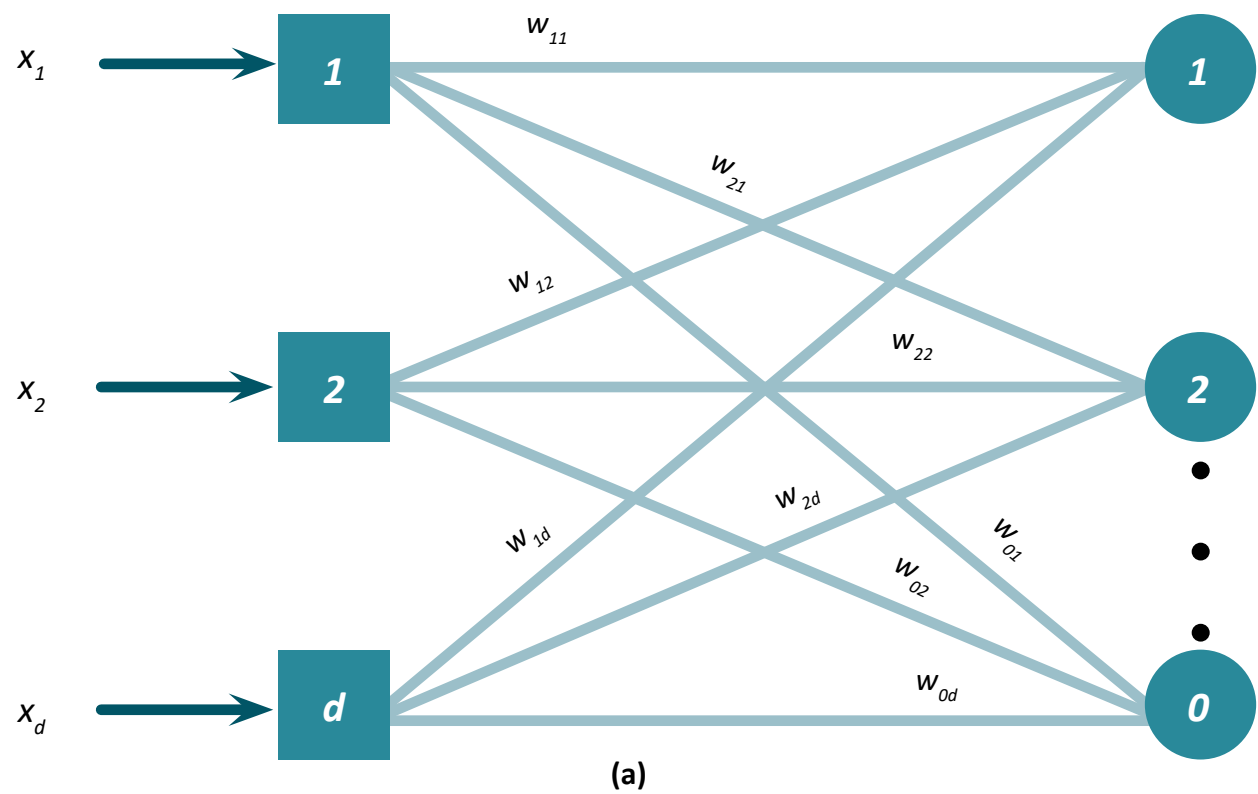


Figura 3.11 - Exemplos de arquitetura de um SOM. (a) SOM unidimensional. (b) SOM bidimensional.
Fonte: Silva, Peres e Boscaroli (2016)

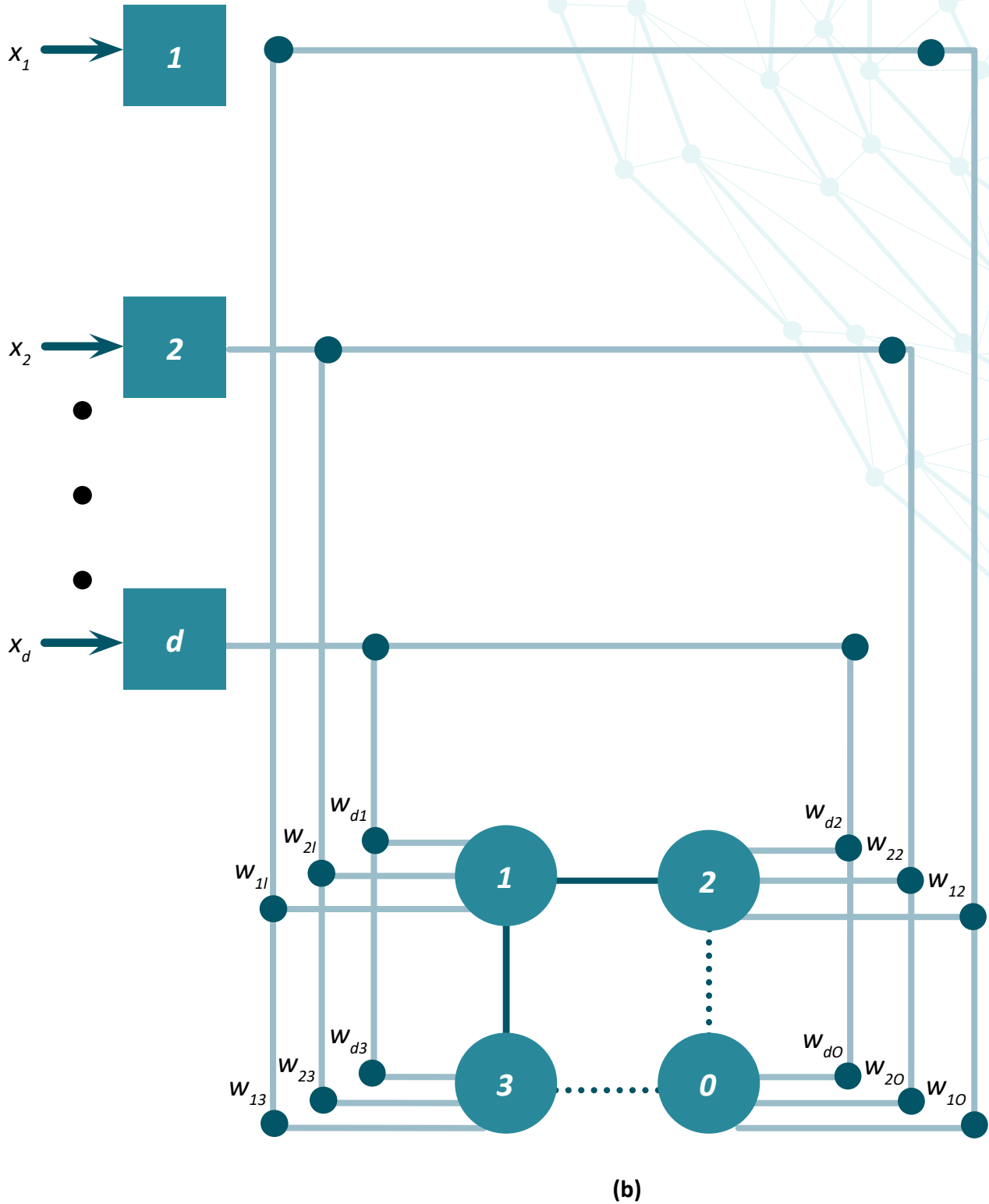


Figura 3.11 - Exemplos de arquitetura de um SOM. (a) SOM unidimensional. (b) SOM bidimensional.
Fonte: Silva, Peres e Boscaroli (2016)

Regra de Propagação: Produto escalar entre as entradas do elemento processador e os respectivos pesos associados às conexões afluentes ao referido elemento.

Função de ativação: Função Degrau, aplicada apenas ao neurônio vencedor, ou seja, somente o neurônio com maior potencial de ativação é submetido à função de ativação.

Regra de Aprendizado: Segundo esta regra, o aprendizado é não supervisionado, e ocorre de tal forma que a direção de atualização dos pesos minimiza a diferença entre o vetor de pesos e o vetor de entrada do elemento processador vencedor. O funcionamento desta regra requer que os vetores estejam normalizados, sendo que somente os pesos do neurônio vencedor são atualizados. Esta regra de aprendizado tem como principal objetivo ajustar os pesos de tal maneira que vetores de entrada similares ativem sempre o mesmo neurônio.

O tipo de competição que ocorre entre os neurônios é o winner takes all, em que apenas um neurônio pode vencer a disputa. Além disso, esses algoritmos utilizam conexões inibitórias e o conceito de vizinhos topológicos (figura 3.12), ou seja, os vizinhos mais próximos do neurônio vencedor também têm seus pesos ajustados.

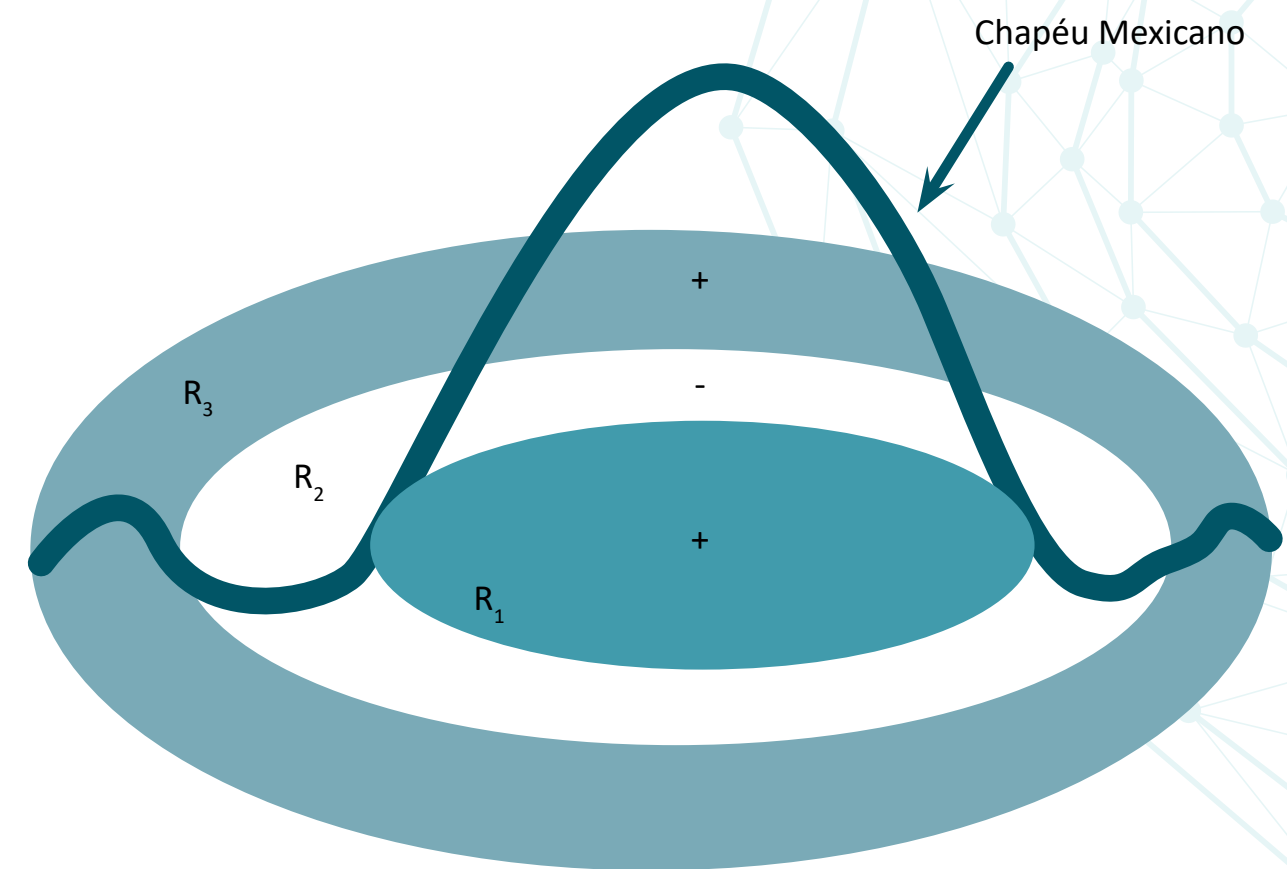


Figura 3.12 – Conceito de vizinhos topológicos (Chapéu Mexicano)
Fonte: Goldschmidt, Passos e Bezerra (2015)

Cada neurônio recebe todos valores de entrada, funcionando como um discriminador ou extrator de características. Desta forma, quanto mais semelhante a um vetor de pesos uma entrada for, maior será a saída gerada por esse neurônio (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

Com isso chegamos ao final do nosso papo sobre os algoritmos baseados em RN. No tópico a seguir falaremos sobre as formas de aplicação desses algoritmos e das Redes Neurais de maneira geral.

3.3 - APLICAÇÃO DAS REDES NEURAIS

Agora que você já sabe o que são as Redes Neurais e quais são os principais algoritmos que se utilizam dos conceitos das RN para resolver problemas, chegou o momento de você conhecer algumas de suas principais formas de aplicação. Essas aplicações ocorrem nas mais diversas áreas, mas algumas delas recebem um certo destaque, tais como a medicina e financeira, por exemplo. Contudo é importante que você tenha uma ideia de todo o potencial que a utilização das RN nos oferece, e como elas podem ser utilizadas para nos ajudar a solucionar uma os mais diversos tipos de problemas.

Algumas das principais aplicações das Redes Neurais Artificiais na atualidade são a detecção de fraudes em cartões de crédito, otimização logística no setor de transportes, processamento de linguagem, incluindo caracteres e voz, diagnósticos médicos, marketing, previsões na área financeira, sistemas de controle na área de robótica, previsão de demandas de energia, controle de qualidade e processos, identificação de compostos químicos, avaliação de ecossistemas, dentre outras (SAS, 2019a).

Agora vamos falar com um pouco mais de detalhes sobre algumas delas:

Detecção e fraudes: As fraudes, principalmente aquelas aplicadas em transações financeiras, têm sido um problema cada vez mais frequente na área financeira. Entretanto, esse não é um problema exclusivo dos bancos. O número de transações online, por exemplo, tem sido cada vez maior, principalmente nos sites de e-commerce. Para você ter uma ideia, o número de fraudes no Brasil em 2017 foi de aproximadamente 1,8 milhão entre janeiro e novembro. Contra os bancos e financeiras, esse número foi de aproximadamente 428 mil, incluindo fraudes em cartões de crédito, cheques e até empréstimos contraídos de forma ilícita (CANALTECH, 2017; NOVAES, 2018).

A boa notícia é que diversas soluções antifraude têm sido criadas e evoluídas para tornar esse tipo de transação cada vez mais segura. Em grande parte, essas soluções são baseadas no aprendizado de máquina (Machine Learning) ou no aprendizado profundo (Deep Learning). A ideia principal é analisar o comportamento das transações, entender como elas funcionam e, a partir daí, identificar padrões possivelmente fraudulentos. Conforme o sistema vai “treinando”, ele vai aprendendo com os seus erros e acertos e se ajustando para identificar e tratar diferentes tipos de situações.

A grande vantagem aqui é que esse tipo de ferramenta consegue identificar características e padrões nos dados que um ser humano não conseguiria identificar com facilidade. Outro ponto importante a favor dos sistemas antifraude está na sua grande assertividade. Isso acontece porque eles possuem um nível de refinamento de regras muito maior do que se essas regras fossem estipuladas por humanos e simplesmente replicadas pelo sistema (CANALTECH, 2017; NOVAES, 2018).

A cerejinha do bolo talvez seja o tempo necessário para fazer um grande volume de validações sobre uma transação: tudo pode ser feito em alguns milissegundos, sem a necessidade de fazer um bloqueio no cartão ou interromper a compra de um usuário de e-commerce, por exemplo. Contudo, nem tudo são flores, não é mesmo? Para se protegerem, estima-se que os bancos brasileiros invistam em torno de 2 bilhões de reais por ano em tecnologias que lhes permitam minimizar ataques e fraudes (NOVAES, 2018).

De qualquer forma, os investimentos deverão continuar sendo feitos, afinal, não são apenas os bancos e instituições financeiras que se especializam e evoluem: os criminosos também têm buscado conhecimento e aprimoramento de suas técnicas, gerando um custo à economia global que fica próximo aos 600 bilhões de dólares (DSA, 2018). Por essas e outras razões, as empresas de segurança irão investir cada vez mais em IA, principalmente em técnicas de Machine Learning e Deep Learning.

Otimização de logística: Muitos são os benefícios relatados pelas empresas que passaram a utilizar técnicas de IA em seus processos logísticos. Dentre essas vantagens, podemos citar a redução de falhas em processos, aumento da produtividade e efetividade das entregas, redução de custos, além de melhorias na forma como o transporte é realizado e em como as mercadorias são armazenadas.

Isso tudo tem sido possível porque essa área tem se tornado cada vez mais proativa, buscando tecnologias e investindo em ferramentas que contribuam com a melhoria das suas atividades administrativas, operacionais e focadas no atendimento aos clientes. A principal matéria-prima para todas essas evoluções é o grande volume de dados coletados a partir das milhões de operações logísticas realizadas todos os dias.

Algumas das empresas que mais se beneficiam com toda essa evolução são os e-commerce. A utilização da IA nesse contexto permite que elas se programem com as quantidades de produtos em estoque, respeitando a sazonalidade e as características que os produtos precisam ter. Os principais efeitos ocorrem sobre processos que essas empresas executam. Grande parte deles pode ser sentida na melhor oferta de produtos e serviços aos clientes e, conseqüentemente, no engajamento ou fidelização.

De forma prática, sensores podem ser usados para monitorar as condições dos equipamentos utilizados em todo o processo, de forma que possíveis falhas possam ser detectadas e prevenidas, impedindo que elas tenham impacto sobre a operação. Os recursos de roteamento e envio inteligente também são bastante importantes nesse contexto, afinal, eles determinam a melhor rota e forma de envio das mercadorias, além de buscar a prevenção da ocorrência de atrasos e outros contratempos que impactem nos prazos e na qualidade das entregas.

A gestão do estoque também é fundamental. É a partir dela que as empresas sabem os produtos que têm à disposição e com que frequência eles precisam ser repostos. A localização e forma de armazenamento dos produtos também facilitam a busca e coleta de itens no estoque, tornando o processo mais ágil e menos suscetível a erros (ROMEDER, 2019).

Processamento de linguagem: O Processamento de Linguagem Natural (PLN) é uma área da Inteligência Artificial dedicada à geração e compreensão automática de linguagens naturais humanas, incluindo a fala, a escrita e o reconhecimento de símbolos. A partir do momento em que as máquinas adquirem essas capacidades, elas permitem que uma série de problemas possa ser resolvida.

Dentre as principais aplicações do PLN podemos destacar as tarefas de sumarização automática, análise de discursos, tradutores automáticos, reconhecimento e execução de comandos de voz, respostas a perguntas, análises de sintaxe e gramática, extração de informações, reconhecimento de sentimentos, dentre outras. É válido ressaltar que o PLN não é uma “ciência” muito recente, entretanto, ela começou a evoluir de forma mais significativa a partir do momento em que a computação se tornou mais poderosa e os algoritmos foram sendo aprimorados (RODRIGUES, 2017; SAS, 2019b).

A partir da utilização do PLN, um volume cada vez maior de dados pode ser processado, a fim de identificar sentimentos, emoções, encontrar padrões, relações e a relevância dos dados. O melhor disso tudo é que os dados não precisam estar estruturados, o que acaba sendo muito comum, considerando toda a complexidade da linguagem humana nas suas mais diversas formas. Em níveis mais complexos, o PLN utiliza um conjunto de subtarefas, incluindo (SAS, 2019b):

- **Categorização de conteúdo:** Permite que os conteúdos sejam identificados, catalogados e sumarizados;
- **Descoberta e modelagem de tópicos:** Permite a descoberta do significado e os temas de correlação entre os conteúdos analisados;
- **Extração contextual:** Permite a extração de informações estruturadas, de forma automática;
- **Análise de sentimentos:** Permite a identificação de sentimentos e opiniões em meio a grandes volumes de texto;
- **Conversão de fala em texto/texto em fala:** Permite transformar comandos de texto em fala, ou de fala em texto;
- **Sumarização:** Permite a geração automática de resumos a partir de grandes volumes de texto;
- **Tradução de máquina:** Permite traduzir texto ou fala para outros idiomas.

Dentre as principais aplicações do PLN, podemos identificar os filtros antispam, textos sugeridos em pesquisas, tradutores automáticos, reconhecimento de sentimentos em redes sociais, dentre outras.

Diagnósticos médicos: Os diagnósticos médicos talvez sejam uma das principais aplicações da IA na medicina, porém não são a única. O tratamento de doenças, por exemplo, tem sido aprimorado ao longo do tempo a partir da utilização, principalmente, do Watson, da IBM. Como ele consegue fazer isso? Bem... Ele usa o Deep Learning para buscar e analisar publicações científicas sobre o problema que se pretende resolver, além de informações sobre o paciente e sobre outros casos semelhantes. A partir desses dados, ele apresenta os possíveis tratamentos, além das consequências e efeitos colaterais de cada um deles.

Outra aplicação interessante é a interpretação de resultados de exames de imagens, tais como as tomografias e mamografias, por exemplo. E falando em interpretação de resultados, há também algoritmos capazes de combinar vários sintomas para diagnosticar doenças. Um exemplo disso é a biblioteca TensorFlow, do Google, que permite a identificação de doenças por meio da comparação entre imagens e com o histórico médico do paciente.

É importante ressaltar que, embora essas ferramentas tragam muitos avanços para a medicina, é muito cedo para se falar em substituição dos médicos. De maneira geral, essas ferramentas visam auxiliar esses profissionais na realização do seu trabalho. Um exemplo disso são os mecanismos de recuperação de dados utilizados para armazenamento e recuperação de conteúdos de laudos médicos e prontuários, tornando as atividades médicas mais organizadas e simples. Além disso, a busca por casos específicos torna-se mais simples e objetiva, tornando as avaliações médicas muito mais rápidas.

O fornecimento de informações para os médicos e equipes de apoio não para por aqui: atualmente temos ferramentas capazes de monitorar os pacientes e, em tempo real, comunicar as equipes médicas em casos de evoluções no quadro desse paciente. Além disso, é possível que a ferramenta analise os resultados de exames feitos por esse paciente, e faça recomendações sobre uso de medicamentos ou sobre outras formas de tratamento (DE ROBERTO, 2018; STEFANINI, 2018).

Marketing: Muitos benefícios têm sido trazidos também à área de Marketing pela utilização da IA: alguns deles vem por meio dos chatbots que apoiam no atendimento aos clientes, do reconhecimento de linguagem, que facilita a realização de buscas, dos motores de recomendação de produtos, preços e de conteúdo. Dentre os benefícios que essas técnicas podem trazer, podemos destacar o engajamento dos clientes, maior efetividade de campanhas e conteúdos disponibilizados, além do incremento nos resultados das vendas (MORENO, 2018).

Outros recursos interessantes são a produção de conteúdo para sites de forma totalmente automática, incluindo artigos e opiniões, criação de conteúdo personalizado e análises preditivas. Em relação à geração de leads, o processo de nutrição e contato com os futuros clientes também pode ser automatizado e incrementado por meio da utilização da IA. O foco principal aqui é proporcionar experiências cada vez melhores para os clientes, de forma que estes se tornem cada vez mais engajados e tragam resultados cada vez melhores para as empresas (CHAFFEY, 2019).

Área financeira: Da mesma forma que ocorre em outras áreas, a financeira também busca investir em tecnologias que lhe permitam ser mais efetiva e incrementar resultados a partir de suas operações. Um exemplo disso é a disputa pelo consumidor, que agora passa a ser focada no perfil de cada cliente, de modo que se possa oferecer a ele as melhores soluções de acordo com o seu perfil. Além disso, os consumidores têm à sua disposição diversas ferramentas, incluindo chats e outros recursos, que lhes conferem maior autonomia e independência.

Outro exemplo interessante são as parcerias entre empresas, que acabam sendo constituídas para ofertar melhores serviços e experiências para um número maior de clientes. Um dos casos mais conhecidos é o da parceria formada entre a Amazon, JPMorgan e Berkshire Hathaway, que criaram uma seguradora de saúde para os seus funcionários. Nesse caso, as empresas colaboram entre si para fornecer serviços aos seus funcionários, mas também podem gerar e ofertar serviços para outras empresas/consumidores (FAUSTINO, 2018).

Mas a coisa não para por aí. A plataforma Warren Brasil é um exemplo bem interessante quando o tema é finanças pessoais. A ideia principal da plataforma é direcionar os usuários para investirem seu dinheiro naquilo que está mais próximo dos seus objetivos e preferências. Outro exemplo é o chat Erica, do Bank of America, que conversa com os seus clientes e sugere opções para ajudá-los a tomarem decisões sobre investimentos e economia. Alguns bancos brasileiros também têm investido nesse tipo de tecnologia, visando fornecer aos clientes uma experiência cada vez mais autônoma e com melhores resultados (MAGNUS, 2018).

Outras vantagens obtidas a partir da utilização da IA na área financeira são o atendimento virtual a clientes, que tem aumentado gradativamente, a automação de processos internos por meio de robôs, aumento da agilidade de processos por meio da retroalimentação de informações, redução do número de fraudes, redução de custos operacionais, dentre outras.

Setor energético: No setor energético, um dos principais investimentos está relacionado à utilização de robôs para automação de processos. O que se espera com isso é que os trabalhos repetitivos deixem de ser realizados por humanos e passem a ser realizados por máquinas, liberando os humanos para exercerem atividades criativas. Dentre as atividades que os robôs poderão realizar estão o saneamento de cadastros, removendo inconsistências, o apoio no processo de operação das redes de distribuição, atendimento aos consumidores, gestão de ativos, dentre outras.

Assim como acontece com as áreas financeira e médica, no caso da área de energia um dos focos também será um relacionamento mais próximo e mais sólido com os clientes, buscando compreender as suas necessidades e criando produtos e serviços sob medida para eles. Dentre esses serviços, podemos destacar as emissões de 2ª via de faturas, pedidos de ligação, alterações em datas de vencimento e manutenções de contratos. Isso pode ser resolvido, em partes, por meio do processamento de linguagem natural e da computação cognitiva.

Identificação de compostos químicos: Você já deve ter ouvido falar da tabela periódica dos elementos, não é mesmo? Bem... Algoritmos baseados no processamento de linguagem natural conseguiram analisar os padrões químicos dos elementos da tabela e reorganizá-los em algumas horas. Isso seria impossível se tivesse que ser feito por seres humanos, afinal, a tabela periódica levou décadas para ser organizada por humanos. Nosso trabalho acaba sendo o de alimentar o algoritmo com as informações que temos sobre os elementos, e ele se vira para produzir informações relevantes a partir desses dados.

A expectativa é de que possamos enviar alguns parâmetros ou informações para o algoritmo, conhecido como Atom2Vec, e ele poderá descobrir e projetar novos materiais. Isso poderá trazer benefícios para as mais diversas áreas, incluindo a médica e a de energia, por exemplo (ZAP, 2018).

Em um outro exemplo, uma ferramenta chamada Reinforcement Learning for Structural Evolution (ReLeaSE), desenvolvida nos EUA, foi capaz de criar um medicamento totalmente do zero. Inicialmente, a rede neural vai recebendo informações e, a partir daí, avalia as moléculas geradas, como num processo de aprendizagem. O processo é repetido até que as moléculas geradas apresentem a qualidade esperada. Isso poderá trazer resultados extraordinários para a indústria farmacêutica em todo o mundo, além de apoiar cientistas na condução de suas pesquisas (WAKKA, 2018).

É importante ressaltar que, embora tenhamos sobre várias possibilidades de aplicação da IA, em especial das Redes Neurais Artificiais, há ainda muitas outras aplicações interessantes nas mais diversas áreas, inclusive para resolver problemas que nem passam pelas nossas cabeças. Por isso é sempre bom dar uma pesquisada e atualizar os seus conhecimentos.

3.4 - OUTROS MÉTODOS BIOINSPIRADOS

Quando falamos em solucionar problemas por meio da utilização de Inteligência Artificial, devemos considerar que esses problemas possuem algumas características em comum, incluindo o espaço de busca e a função de avaliação, por exemplo. Em alguns casos temos até os resultados esperados. Nesse contexto, como já vimos em outros momentos aqui em nosso e-book, há uma série de métodos que podem ser utilizados para resolver os mais diversos tipos de problemas.

Na introdução desta unidade falamos um pouco sobre os métodos bioinspirados para que você entendesse de onde vem os conceitos das Redes Neurais e de outros métodos baseados na natureza. A partir de agora, falaremos um pouco mais sobre outros métodos bioinspirados: os Algoritmos Genéticos e a Lógica Nebulosa. Esses métodos são utilizados para resolver problemas mais complexos, em que os métodos que estudamos até o momento acabam não sendo tão eficazes.

3.4.1 - ALGORITMOS GENÉTICOS

Os Algoritmo Genéticos (AG) são baseados em processos relacionados com organismos biológicos. Na natureza, por meio de muitas gerações, populações de uma mesma espécie evoluem de acordo com os princípios da seleção natural e sobrevivência do mais adaptado, inicialmente estudados por Charles Darwin. Os AG são, portanto, baseados na teoria da evolução e na genética.

A partir da imitação do processo de adaptação dos seres vivos, os AG são capazes de se adaptarem aos problemas e encontrar soluções a partir dessas adaptações. Pensando sob a perspectiva da adaptação, apenas os indivíduos mais adaptados sobrevivem e, mais que isso, eles podem transmitir a suas características para as gerações seguintes.

APRENDA UM POUCO MAIS

Antes de seguir com a sua leitura, faça uma pausa para compreender um pouco melhor o que é o Darwinismo e como ocorreu a evolução dessa teoria. Isso fará com que você compreenda melhor os conceitos ligados a ele e, conseqüentemente, compreenda melhor alguns conceitos que iremos abordar a seguir sobre os Algoritmos Genéticos. Siga a dica de um artigo para que você aprenda um pouco mais. Basta acessar este link



[Darwinismo](#)

Boa leitura!

Em geral, os AG são úteis em problemas complexos que envolvam otimização. Quando falamos em otimização, as abordagens clássicas trabalham com um único elemento candidato que, de forma iterativa, é manipulado por meio de um conjunto de heurísticas que são associadas àquele problema que se pretende resolver. No caso dos AG, os algoritmos atuam sobre uma população de indivíduos em paralelo. A vantagem disso é que esses indivíduos podem atuar na busca em várias regiões do espaço de estados simultaneamente, com grupos de membros adequados para procurar soluções nessas diversas regiões.

Em alguns contextos, como a da mineração de dados, por exemplo, os AG podem ser aplicados a diversas tarefas, incluindo a sumarização. Geralmente, essas tarefas ou problemas envolvem três componentes: variáveis, restrições e funções-objetivo. (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

As variáveis descrevem os aspectos do problema. Cada possível combinação de valores das variáveis é um elemento contido em um conjunto normalmente denominado espaço de busca (search space). As restrições indicam os valores que as variáveis podem assumir. Por último, as funções-objetivo, descritas em um problema de otimização, definem a geografia básica do espaço de busca, e determinam que técnicas podem ser utilizadas.

Técnicas baseadas em modelos heurísticos como os AG não podem garantir que sempre encontrarão a solução ótima para o problema. Porém, essas técnicas podem conseguir soluções próximas, ou aceitáveis (subótimas). A grande vantagem dessas técnicas está no fato de que elas não impõem tais restrições quanto outros métodos. É isso que faz com as soluções encontradas por elas sejam bastante eficazes. Outro ponto importante é que, pelo fato de serem derivados de organismos biológicos, os algoritmos genéticos possuem a capacidade de identificar fatores ambientais e utilizá-los como base na resolução dos problemas.

O funcionamento desses algoritmos ocorre mais ou menos da seguinte forma: inicialmente é gerado um conjunto aleatório de indivíduos que podem ser vistos como a solução do problema. A esse conjunto dá-se o nome de população. Conforme o processo evolutivo vai ocorrendo, essa população vai sendo avaliada. Nesta avaliação, cada indivíduo recebe uma nova que reflete a sua capacidade de adaptação ao ambiente. Aqueles indivíduos mais adaptados seguem o processo, enquanto o resto é eliminado.

Os membros que são mantidos podem sofrer mudanças nas suas características além de cruzamentos (crossover) e recombinações genéticas (mutação). Esse processo, também chamado de reprodução, ocorre até que uma solução satisfatória para o problema seja encontrada.

Na modelagem por AG, cada cromossoma da população representa uma regra, e cada gene representa um atributo do banco de dados. Cada regra, por sua vez, pode ser avaliada com relação à sua confiança e à sua abrangência. Quanto maiores forem a confiança e a abrangência de uma regra, melhor é o indivíduo que representa esta regra. Estas medidas podem ser utilizadas para avaliar a quantidade das regras geradas durante a execução do Algoritmo Genético (GOLDSCHMIDT; PASSOS; BEZERRA, 2015).

Uma vez definida a modelagem de uma regra por meio de um cromossoma, a próxima definição a ser feita é acerca da forma de realizar o cruzamento genético (crossover) entre um par de cromossomas. A seguir temos um exemplo que ilustra o cruzamento genético entre duas regras, gerando duas novas regras. Consideremos como operador de cruzamento o crossover de um ponto (falaremos mais dele daqui a pouco) cujo ponto de corte tenha ocorrido na posição 2.

Cromossomas pais:

| | | | | |
|--------------------------------------|--------------------------------------|-----------------|-----------------|----------------|
| Receita Serviço 1 5000 < R\$ 7000 | Receita Serviço 2 7000 < R\$ 8000 | CodAtividade=13 | 10<#_Filiais<50 | #_Empreg > 100 |
| Receita Serviço 1 3000 < R\$ 4000 | Receita Serviço 2 8500 < R\$ 9500 | CodAtividade=12 | 5<#_Filiais<10 | #_Empreg < 100 |

Cromossomas filhos:

| | | | | |
|--------------------------------------|--------------------------------------|-----------------|-----------------|----------------|
| Receita Serviço 1 5000 < R\$ 7000 | Receita Serviço 2 7000 < R\$ 8000 | CodAtividade=12 | 5<#_Filiais<10 | #_Empreg < 100 |
| Receita Serviço 1 3000 < R\$ 4000 | Receita Serviço 2 8500 < R\$ 9500 | CodAtividade=13 | 10<#_Filiais<50 | #_Empreg > 100 |

Esse conjunto de operações realizado com os membros de uma população, na busca por populações melhores, é fundamental para que as soluções dos problemas sejam encontradas e melhoradas. Para realização dessas operações, são necessários alguns conjuntos de operadores genéticos, sendo eles os operadores de cruzamento (crossover) e operadores de mutação. São esses mesmos operadores que garantem que, embora possamos ter populações novas, essas populações sempre herdarão as características de seus pais.

Os operadores de cruzamento predominam nas operações, sendo muito mais utilizados do que os operadores de mutação. Eles são os responsáveis pela recombinação das características dos pais durante a reprodução, e fazem com que os filhos herdem as características dos pais. Algumas das formas de utilização desses operadores são (CARVALHO, 2009):

Um ponto: apenas um ponto de cruzamento é definido e, a partir dele, as informações dos pais serão trocadas (recombinadas) para gerar novos indivíduos (Figura 3.15).

Multipontos: Muitos pontos de cruzamento podem ser utilizados. Pode ser considerada uma derivação do cruzamento de um ponto.

Uniforme: Não utiliza nenhum ponto de cruzamento. Nesse caso há uma definição, por meio de um parâmetro global, da probabilidade de cada uma das variáveis ser trocada entre os pais.

Outro ponto importante que precisamos considerar quando falamos dos AG são os parâmetros genéticos. Esses parâmetros definem comportamentos para os algoritmos, considerando a complexidade dos problemas que precisamos resolver e os recursos computacionais disponíveis para execução do algoritmo. Os parâmetros são os seguintes (CARVALHO, 2009):

Tamanho da população: Este parâmetro afeta diretamente o desempenho e eficiência dos AG. Isso acontece porque, quanto maior a população, maior o espaço de busca coberto por cada grupo de indivíduos previamente definido. O problema de termos grandes populações é que são necessários mais recursos computacionais.

Taxa de cruzamento: Esta taxa deve ser equilibrada, uma vez que é ela que determina a velocidade com que novos indivíduos são inseridos na população. Se a taxa for muito alta, há um risco de perda de indivíduos com alta aptidão. Se a taxa for muito baixa, o algoritmo tende a ser muito lento.

Taxa de mutação: Esta taxa garante que as posições não fiquem com valores estagnados, permitindo uma busca mais abrangente no espaço de busca. O problema aqui é que, se a taxa for muito alta, a busca realizada no espaço acaba sendo uma busca aleatória.

Intervalo de geração: Este parâmetro define o percentual da população que será substituída na próxima geração. O comportamento aqui deve ser semelhante ao que temos com a taxa de cruzamento. Se muitos indivíduos forem substituídos de uma vez, é possível que haja a perda de indivíduos com alta aptidão, ao passo que, se poucos indivíduos forem substituídos, o algoritmo pode se tornar muito lento.

Os exemplos mais simples de GA utilizam crossover e mutação, quando ocorre, de ponto único. Versões mais complexas e mais recentemente elaboradas desses algoritmos possuem algumas outras revisões e melhorias (DAVIS, 1991):

Normalização linear: A aptidão dos indivíduos é classificada de forma que seja possível determinar o número de descendentes esperados de cada um deles.

Elitismo: Determina que o melhor indivíduo da população sempre seja salvo na próxima geração.

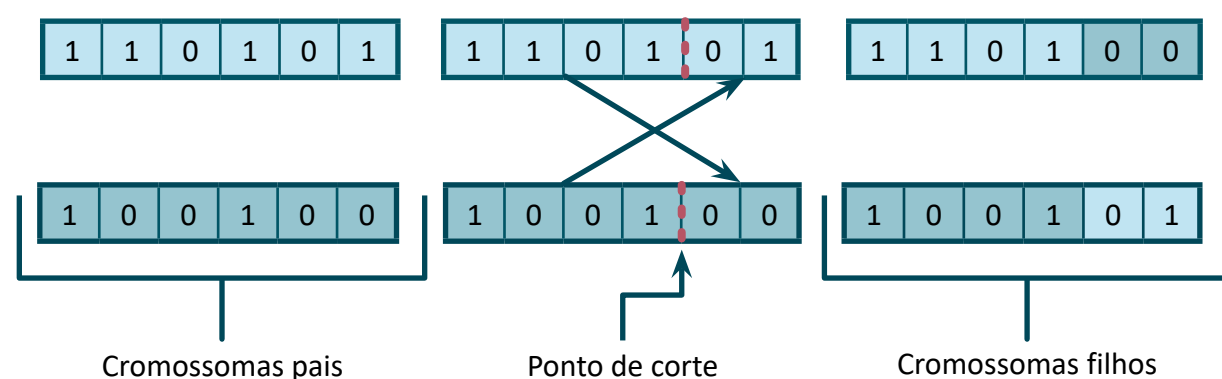


Figura 3.13 - Exemplo de crossover de um ponto
Fonte: Elaborado pelo autor (2019)

3.4.2 - LÓGICA NEBULOSA

Reprodução em estado estacionário: Determina que apenas um ou dois indivíduos de cada população sejam reposicionados no espaço de estado de cada vez.

Cruzamento uniforme: Em vez de escolher um ponto único de crossover, cada gene é escolhido de forma aleatória entre os pais.

Seleção de operadores com base na aptidão: A adequação de cada operador (crossover ou mutação) ocorre de acordo com o desempenho da capacidade de adaptação de cada indivíduo.

Adaptar a adequação do operador: São geradas estatísticas sobre cada operador, a fim de avaliar a sua capacidade de gerar melhorias na população. Caso a sua taxa de aplicação e, portanto, o valor que ele gera, seja alta, esse operador é considerado adequado.

Como você pode observar, o funcionamento desses algoritmos é bastante dinâmico e é isso que permite que eles sejam aplicados na resolução de problemas nas mais diversas áreas. Alguns exemplos de sua aplicação são os controles de sistemas dinâmicos, simulação de modelos biológicos, evolução interativa de imagens, busca por novos modelos e topologias de redes neurais artificiais e biológicas, dentre outras.

A Lógica Nebulosa (LN) é uma técnica que permite construir sistemas que lidem com informações imprecisas ou subjetivas. Diferentemente da lógica clássica, a lógica nebulosa oferece flexibilidade na definição e na avaliação de conceitos, tendo suas origens nos sistemas multivalorados e em funções de pertinência.

A LN é baseada no princípio de que o pensamento humano é estruturado em classes de objetos cuja transição entre pertencer ou não a um determinado conjunto é gradual. Isso acontece porque o raciocínio humano não trabalha apenas de forma dicotômica (verdadeiro ou falso), mas sim utilizando uma linguagem natural em que há multiplicidade de sentidos (SAMPAIO; OLIVEIRA; IGNACIO, 2007).

Diversos métodos de mineração de dados, por exemplo, foram adaptados de forma que pudessem incorporar a flexibilidade proporcionada pela Lógica Nebulosa. Dentre esses métodos, podemos citar as versões nebulosas do K-means e do C4.5. Nestas versões, os registros da base de dados podem pertencer a diversos grupos e classes simultaneamente, com diferentes graus de pertinência.

De forma mais ampla, a LN pode ser usada para lidar com informações que não podem ser representadas por valores numéricos precisos. São informações que envolvem, além de um certo grau de incerteza, interpretações subjetivas e que podem até variar de um indivíduo para outro. Um exemplo dessa subjetividade pode ser observado na Figura 3.14.



Figura 3.14 - Em que momento o copo deixa de estar vazio?
Fonte: Gomes e Rodrigues (2015)

O exemplo da Figura 3.16 nos faz refletir sobre a seguinte questão: supondo que o copo esteja sendo enchido gota a gota, até que esteja completamente cheio, a partir de que momento podemos considerar que ele já não está mais vazio? Talvez o principal ponto aqui seja o fato de que o copo vai sendo, gradativamente, migrado de uma classificação para a outra: de vazio para cheio, porém há uma série de estados intermediários.

Considerando esse exemplo, podemos afirmar que a Lógica Nebulosa, também conhecida como Lógica Fuzzy, nos permite representar modelos mais complexos e imprecisos e que, de certa forma, se assemelham mais a modelos de problemas do mundo real. A partir daí, é possível construirmos algoritmos que simulem de forma mais aproximada a forma como cérebro humano funciona, principalmente sob o aspecto de que um problema não precisa ser tratado de forma dicotômica (GOMES; RODRIGUES, 2015).

Bem... até o momento deixamos claro que a LN é recomendada para solução de problemas em que o tratamento baseado em fatores numéricos é complicado. Isso pode acontecer em virtude de diversos fatores, incluindo a quantidade de variáveis envolvidas, além da incompatibilidade dos resultados encontrados em relação àquilo que se espera. Por isso a LN está baseada em palavras e não em números, ou seja, os resultados são representados de forma linguística (SAMPAIO; OLIVEIRA; IGNACIO, 2007).

Outra característica interessante sobre a LN, é que ela se utiliza de uma série de modificadores de predicados, tais como muito, mais, menos, pouco, bastante e médio. Além disso, ela usa um conjunto de quantificadores, tais como poucos, vários ou em torno de. A LN também usa probabilidades linguísticas, tais como provável e improvável (SAMPAIO; OLIVEIRA; IGNACIO, 2007).

A LN é utilizada por diversas razões, dentre as quais podemos destacar o fato de se aproximar da forma como os seres humanos pensam, além de permitir a avaliação/observação de variáveis que podem conter diversas possibilidades de valores. Isso facilita a aquisição e a formação de bases de conhecimento utilizadas na solução de problemas. Além disso, sua flexibilidade permite que ela seja utilizada em conjunto com outros sistemas que trabalhem com dados imprecisos. De forma resumida, a LN é a opção ideal quando há a necessidade de se lidar com a imprecisão e com a falta de linearidade dos problemas (SAMPAIO; OLIVEIRA; IGNACIO, 2007).

Dentre os principais conceitos que envolvem a LN, podemos destacar os conjuntos nebulosos, as funções de pertinência, os operadores lógicos, as regras nebulosas e o ciclo dos processos nebulosos (SAMPAIO; OLIVEIRA; IGNACIO, 2007), sobre os quais falaremos um pouco melhor a seguir:

Conjuntos nebulosos: Um conjunto nebuloso é um conjunto preciso, porém com limites imprecisos. Eles são usados porque a LN trabalha com grau de pertinência dos elementos a um determinado conjunto. Se tivermos, por exemplo, um conjunto que armazene dados de pessoas que medem 1,75m, um conjunto nebuloso poderia armazenar pessoas que medem entre 1,60m e 1,75m, e determinar o grau de pertinência de cada elemento ao conjunto em questão.

Função de pertinência: É a função responsável por determinar o nível de verdade de uma determinada variável, ou seja, o quão pertinente ela é a um determinado conjunto nebuloso. A pertinência pode ser total ou parcial, porém são considerados os intervalos de valores entre o total e o inexistente.

Operadores lógicos: Os operadores lógicos utilizados em conjuntos nebulosos são os de conjunção, disjunção e negação. Porém quando transportados para o modelo Fuzzy, são representados da seguinte forma:

Conjunção (e): $m(A \text{ e } B) = \min(m(A), m(B))$

Disjunção (ou): $m(A \text{ ou } B) = \max(m(A), m(B))$

Negação (não): $m(\text{não } A) = 1 - m(A)$

Regras nebulosas: As regras nebulosas utilizam os operadores lógicos como base para modelar raciocínios imprecisos, semelhantes àqueles característicos dos seres humanos. A partir daí é possível fazer inferências e testes com base nas variáveis de entrada.

Ciclo dos processos nebulosos: Nesse ciclo são realizadas diversas tarefas, incluindo a transformação das variáveis de entrada em conjuntos nebulosos (fuzificação), o direcionamento do conhecimento por meio de regras (base de regras), avaliação da relevância das regras (inferência), obtenção de conjuntos de saída a partir de conjuntos de entrada (agregação) e a conversão dos resultados encontrados em resultados quantitativos (desfuzificação).

Os sistemas baseados nessa lógica podem ser utilizados nas mais diversas áreas, incluindo a engenharia, matemática, biologia, medicina, dentre outras. Dentre os problemas que podem ser resolvidos, podemos citar os sistemas de apoio à decisão, reconhecimento de padrões (incluindo o reconhecimento facial), diagnósticos médicos, previsão do tempo, cálculo de riscos, controle de tráfego, carros autônomos, dentre outras. Além disso, há uma série de equipamentos que utilizam a LN, incluindo televisores, máquinas de lavar, dentre outros (SAMPAIO; OLIVEIRA; IGNACIO, 2007).

E assim, caro (a) aluno (a), chegamos ao final de mais uma unidade. O objetivo aqui foi o de trazer até você um pouco mais de conhecimento sobre alguns métodos amplamente utilizados para resolver problemas com alto nível de complexidade e nas mais diversas áreas. A partir daí você poderá ampliar seus conhecimentos sobre os temas que mais lhe interessaram, além de conhecer outras formas de aplicação de cada um deles. Como uma sugestão de continuidade dos seus estudos, você pode se aprofundar um pouco mais nos conceitos da computação natural.

O QUE APRENDEMOS

Caro (a) aluno (a), nesta terceira unidade do nosso e-book tratamos sobre as Redes Neurais Artificiais, seus objetivos, aplicações, além de suas principais características. Pudemos conhecer os principais conceitos ligados às redes neurais (RN), alguns dos principais algoritmos que implementam os conceitos de RN, algumas das principais formas de aplicação das redes neurais e, por fim, conhecemos outros algoritmos baseados em métodos bioinspirados.

REFERÊNCIAS

CANALTECH. **Inteligência Artificial chega aos sistemas antifraude com aprendizado de máquina.** Disponível em: <<https://bit.ly/2VMYjqL>>. Acesso em: 24 maio. 2019.

CARVALHO, A. P. DE L. F. DE. **Algoritmos Genéticos.**

CHAFFEY, D. **15 Applications of Artificial Intelligence in Marketing.** Disponível em: <<https://bit.ly/2A0EN3P>>. Acesso em: 26 maio. 2019.

DAVIS, L. D. **Handbook of Genetic Algorithms.** 1a ed. New York: Van Nostrand Reinhold, 1991.

DSA. **Inteligência Artificial e o futuro da detecção de fraudes financeiras.** Disponível em: <<https://bit.ly/2W3gwFa>>. Acesso em: 24 maio. 2019.

FAUSTINO, R. **Como a inteligência artificial está mudando o sistema financeiro.** Disponível em: <<https://glo.bo/2W4Oh9r>>. Acesso em: 26 maio. 2019.

FERRARI, D. **Computação Natural: inspirando-se na natureza para resolver problemas.** Disponível em: <<https://goo.gl/yh9cFp>>. Acesso em: 25 set. 2018.

GOMES, D. DA S. M.; RODRIGUES, M. DE C. **Introdução à Lógica Fuzzy com Java.** Disponível em: <<https://www.devmedia.com.br/introducao-a-logica-fuzzy-com-java/32444>>. Acesso em: 2 jun. 2019.

GOLDSCHMIDT, R.; PASSOS, E.; BEZERRA, E. **Data Mining: Conceitos, técnicas, algoritmos, orientações e aplicações.** 2. ed. Rio de Janeiro: Elsevier, 2015.

LANHELLAS, R. **Redes Neurais Artificiais: Algoritmo Backpropagation.** Disponível em: <<https://bit.ly/2YySHC3>>. Acesso em: 18 maio. 2019.

MAGNUS, T. **Como a Inteligência Artificial pode ajudar no controle financeiro?** Disponível em: <<https://bit.ly/2JF7fN1>>. Acesso em: 26 maio. 2019.

MORENO, Y. **6 aplicações da Inteligência Artificial no Marketing.** Disponível em: <<https://bit.ly/2JENllk>>. Acesso em: 26 maio. 2019.

NOVAES, F. **IA e machine learning no combate às fraudes bancárias.** Disponível em: <<https://bit.ly/2YRejtJ>>. Acesso em: 24 maio. 2019.

PALMIERE, S. E. **Arquiteturas e Topologias de Redes Neurais Artificiais.** Disponível em: <<https://bit.ly/2VFzTEi>>. Acesso em: 16 maio. 2019.

RODRIGUES, J. **O que é Processamento de Linguagem Natural.** Disponível em: <<https://bit.ly/2prbuzY>>. Acesso em: 25 maio. 2019.

ROMEDER, S. **5 maneiras de otimizar a cadeia de suprimentos usando AI.** Disponível em: <<https://bit.ly/2YKpzb3>>. Acesso em: 25 maio. 2019.

SAMPAIO, L. M. D; OLIVEIRA, M. J. F. DE; IGNACIO, A. A. V. **Lógica Nebulosa: aplicações e tendências: SPOLM.** Rio de Janeiro - Brasil: [s.n.].

SAS. **Redes Neurais: o que são e qual a sua importância?** Disponível em: <<https://bit.ly/2W2OPwr>>. Acesso em: 24 maio. 2019a.

SAS. **Processamento de Linguagem Natural: O que é e qual sua importância?** Disponível em: <<https://bit.ly/2JG8NX2>>. Acesso em: 25 maio. 2019b.

SILVA, L. A. DA; PERES, S. M.; BOSCARIOLI, C. **Introdução à Mineração de Dados com aplicações em R.** 1. ed. Rio de Janeiro: Elsevier, 2016.

SILVA, L. A. DA; PERES, S. M.; BOSCARIOLI, C. **Introdução à Mineração de Dados com aplicações em R.** 1. ed. Rio de Janeiro: Elsevier, 2016.

STEFANINI, M. **Medicina do futuro: IA para auxiliar em diagnósticos e aprimorar a relação médico-paciente.** Disponível em: <<https://bit.ly/2Wnnwwd>>. Acesso em: 26 maio. 2018.

WAKKA, W. **Inteligência artificial é capaz de criar do zero novos compostos para remédios.** Disponível em: <<https://bit.ly/2Qs1e6X>>. Acesso em: 26 maio. 2019.

ZAP. **Humanos para quê? Inteligência Artificial recria a tabela periódica.** Disponível em: <<https://bit.ly/2VQBims>>. Acesso em: 26 maio. 2019.



CENTRO UNIVERSITÁRIO



EAD

CENTRO UNIVERSITÁRIO FAG

EDUCAÇÃO A DISTÂNCIA

www.ead.fag.edu.br