



# **DEEP LEARNING**

## **UNIDADE I** **AS ORIGENS DO *DEEP LEARNING***

## **Elaboração**

Natasha Sophie Pereira

## **Produção**

Equipe Técnica de Avaliação, Revisão Linguística e Editoração

# SUMÁRIO

## UNIDADE I

AS ORIGENS DO <i>DEEP LEARNING</i> .....	5
--	---

### CAPÍTULO 1

INTELIGÊNCIA ARTIFICIAL.....	6
------------------------------	---

### CAPÍTULO 2

APRENDIZADO DE MÁQUINA .....	14
------------------------------	----

### CAPÍTULO 3

REDES NEURAIS ARTIFICIAIS.....	21
--------------------------------	----

REFERÊNCIAS .....	31
-------------------	----



# AS ORIGENS DO *DEEP LEARNING*

## UNIDADE I

Nos últimos anos, a inteligência artificial (do inglês *Artificial intelligence*, AI) tem se tornado muito popular. O Aprendizado de Máquina (do inglês *Machine Learning*, ML), o Aprendizado Profundo (do inglês *Deep Learning*, DL), e a AI vêm sendo abordados não apenas em publicações voltadas para a área tecnológica e científica, mas várias reportagens vêm apresentando tecnologias, tais como carros autônomos e assistentes virtuais, como inevitáveis no futuro da humanidade. Algumas vezes a AI é apresentada como benéfica, mas por várias vezes a máquina inteligente é retratada como vilã que ocupará o lugar do homem em diversas atividades, tornando-o obsoleto (CHOLLET, 2018).

É essencial que o perpetrador do aprendizado de máquina reconheça a diferença entre os desenvolvimentos tecnológicos que transformam o mundo e as reportagens sensacionalistas. Como estudante deste curso, você passa a ter um papel essencial nesta transformação, visto que se tornará um desenvolvedor de agentes inteligentes!

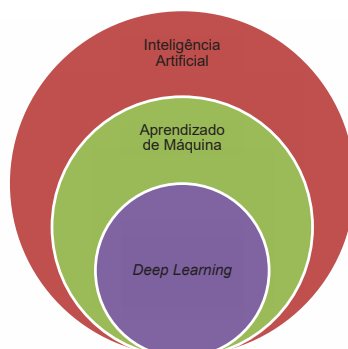


A partir disso, reflita sobre algumas questões:

- » Quais são os avanços que o *Deep Learning* já alcançou?
- » Em termos de impacto na sociedade, quão significativos são estes avanços?
- » Quais são os próximos passos que a Inteligência Artificial irá tomar?
- » Como você deve analisar as notícias sobre os avanços da tecnologia inteligente?

Esta primeira unidade busca esclarecer os principais pontos sobre Inteligência Artificial (AI) e Aprendizado de Máquina (ML), que são a base para o *Deep Learning* (DL). Para entender melhor a correlação entre eles, veja a Figura 1.

Figura 1. Relação entre Inteligência Artificial, Aprendizado de Máquina e *Deep Learning*.



Fonte: Adaptado de Patterson; Gibson, 2017; Chollet, 2018.

# CAPÍTULO 1

## INTELIGÊNCIA ARTIFICIAL

Inteligência Artificial, ou *Artificial Intelligence* (AI), é o termo utilizado para qualquer operação computacional que realize tarefas que seriam consideradas inteligentes caso fossem realizadas pelo ser humano. A AI é uma área de pesquisa da Ciência da Computação que objetiva o desenvolvimento de métodos e metodologias que, utilizando o computador, realizem ou reforcem a capacidade humana de se comportar de forma inteligente, seja na resolução de problemas, aquisição e representação de conhecimento, reconhecimento de padrões, entre outros (LIMA; PINHEIRO; SANTOS, 2014). Para Chollet (2018, p. 4), a melhor definição para AI seria “o esforço para automatizar tarefas intelectuais normalmente realizadas por seres humanos”.

De acordo com Lima, Pinheiro e Santos (2014), um sistema pode ser considerado inteligente quando é capaz de adquirir e armazenar conhecimento, planejar eventos, resolver problemas, representar informações, se comunicar utilizando linguagem coloquial e aprender. A AI surgiu quando os primeiros estudiosos em Ciência da Computação começaram a indagar a capacidade de “pensar” dos computadores, até que ponto os computadores podem agir como seres humanos, desenvolvendo a capacidade de pensar logicamente? Essa indagação perdura ainda hoje.

A Inteligência Artificial é uma área de estudo ampla, que engloba o aprendizado de máquina e, conseqüentemente, o aprendizado profundo. Porém, também inclui muitas outras abordagens que não envolvem diretamente o aprendizado. A tomada de decisão realizada por sistemas, muitas vezes, é apenas um conjunto de regras codificadas, não envolvendo o aprendizado da máquina, como é o caso dos primeiros programas de xadrez. A manipulação do comportamento computacional a partir de regras definidas é conhecida como *AI Simbólica* (*symbolic AI*), e foi o paradigma dominante na área entre os anos 1950 e 1980, quando os pesquisadores acreditavam que, se um conjunto suficientemente grande de regras explícitas fosse implementado, o computador poderia pensar como humano.

No final dos anos 1980, com o aumento significativo do desenvolvimento dos sistemas especialistas, a AI Simbólica atingiu seu ápice, visto que essa abordagem é muito adequada para a solução de problemas lógicos. Porém, ela não é adequada no caso de problemas complexos e difusos, por exemplo, a classificação de imagem, o reconhecimento de fala, ou a tradução de linguagem natural. E foi com base nessa lacuna que surgiu a necessidade de se expandir os conceitos da AI, fazendo com que os computadores possam tomar decisões baseadas em conhecimento previamente adquirido, é o que chamamos de **Aprendizado de Máquina**.

## Histórico

A década de 1940 foi marcada pela Segunda Guerra Mundial, quando surgiu a necessidade de desenvolvimento de tecnologias voltadas para a análise balística, quebra de códigos inimigos, realização de cálculos para projetar armas nucleares etc. A partir dessa necessidade, surgiram os primeiros estudos sobre AI, quando foram desenvolvidos os primeiros computadores próximos ao que conhecemos hoje, que eram utilizados na realização de cálculos.

Após a Segunda Guerra Mundial, os computadores deixaram de ser restritos ao uso militar e científico, e gradativamente passaram a fazer parte do dia a dia das empresas, indústrias e universidades, o que impulsionou as pesquisas em *software*, *hardware*, linguagens de programação, e, consequentemente, em Inteligência Artificial.

Em 1943, McCulloch e Pitts (1943) se dedicaram a pesquisas voltadas ao desenvolvimento de modelos de neurônios artificiais, possibilitando a criação de máquinas capazes de aprender.

Em 1950, Alan Turing (1950) publicou o artigo intitulado “*Computing Machinery and Intelligence*”, no qual apresentou um teste para descobrir se uma máquina poderia ou não imitar o pensamento humano. Este teste, conhecido como “Teste de Turing”, consiste em três indivíduos, um interrogador (humano) e dois interrogados, um humano e um computador. Os três indivíduos são alocados em espaços distintos, de modo que não se conheçam. O interrogador, por meio de um diálogo indireto (usando um teclado, por exemplo), irá questionar os interrogados a fim de distinguir quem é o humano e quem é o computador. Se, ao final do interrogatório, o interrogador não conseguir distinguir quem é o humano, conclui-se, segundo o Teste de Turing, que o computador pode “pensar”. Segundo Russell e Norvig (2010), é muito complexo programar um computador para passar no Teste de Turing, para isso, o computador deverá ser capaz de: I) processar linguagem natural de modo a se comunicar satisfatoriamente; II) representar conhecimento para armazenar o que sabe ou aprende; III) raciocinar de forma automatizada para usar a informação armazenada na elaboração de resposta às questões e desenvolvimento de conclusões; IV) aprender (*Machine Learning*), adaptando-se às novas circunstâncias, detectando e superando padrões.

Em 1956, John McCarthy, Marvin Minsky, Allan Newell e Herbert Simon denominaram de Inteligência Artificial as técnicas desenvolvidas pelo homem para criar máquinas de comportamento inteligente. Isso aconteceu durante um seminário realizado no Colégio Dartmouth, quando vários pesquisadores renomados se juntaram a McCarthy para

estudar o tema. Este evento ficou conhecido como um marco na fundação da área de AI, e, a partir dessa definição, o estudo nesta área se intensificou.

Durante as décadas de 1950 e 1960 aconteceram diversos avanços na área de Inteligência Artificial: os primeiros programas capazes de jogar xadrez como humanos foram desenvolvidos, computadores foram utilizados para provar teoremas de lógica e emular o raciocínio humano, para planejar tarefas, para se comunicar em linguagem natural, para analisar as estruturas de moléculas, foi possível programar um computador para aprender por analogia, enfim, foram tempos de muito entusiasmo e consequente desenvolvimento para a AI.

Entre 1970 e 1980, problemas com armazenamento de informações e tempo de processamento começaram a ser percebidos pelos pesquisadores de AI. Ainda durante essas décadas, vários *softwares* para reconhecimento de linguagem natural foram desenvolvidos, focando na representação e lógica do conhecimento necessário para o entendimento da linguagem. Os problemas estudados incluíam a representação de situações estereotipadas, descrever a organização da memória humana, entender planejamento e objetivos, assim como outros diversos problemas do mundo real. Nessa época foram desenvolvidas diversas linguagens de programação voltadas para a Inteligência Artificial.

Na década de 1980, a Inteligência Artificial atingiu um novo patamar, quando o primeiro sistema especialista em nível comercial foi lançado, e, a partir daí, diversos sistemas especialistas foram desenvolvidos utilizando conceitos de AI, praticamente todas as grandes corporações dos Estados Unidos já possuíam seu próprio grupo de pesquisadores na área para pesquisar e desenvolver seus sistemas especialistas. A Inteligência Artificial ainda impulsionou o desenvolvimento de *chips* computacionais e sistemas com interface amigável para usuários comuns. No final da década de 1980, várias empresas desenvolviam sistemas especialistas, sistemas de visão computacional, robôs, e *softwares* e *hardwares* voltados para AI. Infelizmente, com a empolgação dos pesquisadores da área, várias empresas prometiam desenvolvimento de sistemas impossíveis, e não conseguiam entregar o prometido. Com isso, muitas delas foram esquecidas, e veio o que é chamado de “o inverno da AI” (*AI Winter*). Em 1986, a área da AI conhecida como Redes Neurais (do inglês *Neural Networks*, NN) foi dividida em duas áreas de estudo; uma delas busca desenvolver arquiteturas de redes e algoritmos eficazes, e entender suas propriedades matemáticas; a outra área visa à modelagem das propriedades empíricas a neurônios e conjuntos de neurônios reais. Em 1987, a AI passou de um patamar de desenvolvimento de novas teorias para a comprovação científica de teorias já existentes, buscando o desenvolvimento de soluções para problemas reais. A partir dessa época, a AI passou a ser mais bem firmada nos métodos de pesquisa científicos, ou seja, para que um teorema



fosse aceito, ele deveria ser baseado em uma hipótese, passar por rigorosos experimentos e os resultados deveriam vir acompanhados de análises estatísticas que comprovassem sua importância e eficácia.

Durante a década de 1990, com o surgimento da internet, um dos ambientes mais importantes para os agentes inteligentes, a utilização de sistemas de inteligência artificial se tornou muito comum em aplicações *web*, e o conceito de agentes inteligentes emergiu. Nessa época, surgiram os sistemas de busca, sistemas de recomendação de produtos baseado nas preferências do usuário, entre outros.

Na década de 2000 os fundadores da AI (Jonh McCarthy, Marvin Minsky, Nils Nilsson, Patrick Winston e outros) expressaram sua insatisfação pelos rumos que a AI vem tomando, eles pensam que a AI deveria enfatizar menos a criação de sistemas cada vez melhores em realizar uma tarefa específica, ao invés disso, os pesquisadores de AI deveriam voltar às raízes do conceito e desenvolver uma máquina pensante, capaz de aprender e criar conhecimento. Eles chamaram esse movimento de AI em nível humano (*human-level AI*). Seguindo a mesma ideia, existe um subcampo da AI conhecido como Inteligência Artificial Genérica (do inglês *Artificial General Intelligence*, AGI), que busca o desenvolvimento de um algoritmo universal para que as máquinas possam aprender e agir em qualquer ambiente. Porém, será que essa AGI será nociva aos humanos?

A partir da década de 2000, uma grande quantidade de dados passou a ser disponibilizada, e, ao contrário dos 60 anos anteriores de história da AI, em que os algoritmos eram o foco do estudo, os dados passaram a ser o foco principal de pesquisa em diversas áreas. É possível encontrar gigantescas bases de dados de palavras, imagens, sequências genéticas, fala, entre outras. Alguns pesquisadores sugerem que o “gargalo do conhecimento” na AI, ou seja, o problema de como expressar todo o conhecimento necessário para um sistema, pode ser resolvido através do método de aprendizado ao invés das regras codificadas manualmente, desde que os algoritmos tenham dados suficientes para acumular conhecimento. A partir disso, notou-se que existem diversas novas aplicações para AI, e já se fala em uma nova primavera que está tomando o espaço do inverno da AI.

## Principais aplicações

Existem, atualmente, tantas atividades envolvendo a inteligência artificial nos mais diversos campos de estudo, que chega a ser complicado listar todas as aplicações possíveis para essa tecnologia. Então, baseado em Russell e Norvig (2010) e em Joshi (2017), são apresentadas algumas aplicações que perduram nos estudos de pesquisadores atuais.

- » **Jogos:** o primeiro programa de computador a derrotar um jogador de xadrez profissional, Garry Kasparov, foi o *Deep Blue*, da IBM. A partir daí, diversas outras partidas entre homem e computador já aconteceram, nestas, as vitórias são predominantemente do computador. A AI é utilizada para projetar agentes inteligentes que podem competir com humanos, ou na construção de jogos que interagem com humanos de forma inteligente.
- » **Processamento de Linguagem Natural:** algumas aplicações são os mecanismos de busca de conteúdo na *web*, e programas traduzem automaticamente textos de um idioma a outro, permitindo maior acessibilidade e repercussão. Este campo lida com a manipulação e interpretação de texto, de modo que o ser humano possa interagir com a máquina através da inserção de frases em linguagem natural.
- » **Reconhecimento de Fala:** ao telefonar para uma companhia aérea, para reservar um voo, você pode ter toda a conversa guiada por um sistema automatizado de reconhecimento de fala e gerenciamento de diálogos. São sistemas capazes de “ouvir” e interpretar a fala humana, interagindo com o interlocutor de forma inteligente.
- » **Robótica:** a empresa *iRobot Corporation* desenvolveu os aspiradores robóticos Roomba para uso doméstico. Ela também desenvolve o PackBot, que é um robô mais robusto para manuseio de materiais perigosos, explosivos, e outras finalidades militares. Atualmente, a área da robótica combina diversos conceitos de AI, permitindo que o sistema realize diversas tarefas diferentes dependendo da situação. Os robôs podem “enxergar” coisas à sua volta, fazer medições no ambiente (temperatura, gases, entre outras), se movimentar, entre outras capacidades. Os processadores embarcados permitem a realização de diversas tarefas em tempo real, em resposta a um estímulo, permitindo que os robôs se adaptem a novos ambientes.
- » **Sistemas Especialistas:** utilizam bases de dados da área do conhecimento desejado, como finanças, medicina, *marketing*, entre outras, para aconselhar o usuário em seus procedimentos, ou tomar decisões automaticamente para a solução de determinado problema. Alguns exemplos são:
  - › **Combatente de Spam:** diariamente, são classificados mais de um bilhão de mensagens como sendo *spam* por algoritmos de aprendizado de máquina, assim, eles evitam que o destinatário necessite analisar manualmente cada uma dessas mensagens.
  - › **Planejamento Logístico:** em 1991, a Agência de Projetos de Pesquisa Avançada de Defesa dos Estados Unidos (DARPA, do inglês *Defense Advanced Research Project Agency*) implantou uma Ferramenta de Análise Dinâmica de

Planejamento chamada DART (*Dynamic Analysis and Replanning Tool*), essa ferramenta planejava e programava a logística de transporte durante a crise do Golfo Pérsico. Os planejamentos levavam em conta pontos de partida, destinos, rotas e resolução de conflitos, entre outros parâmetros. Em poucas horas, as técnicas de AI geravam um plano que poderia levar semanas para ser traçado através dos métodos convencionais.

- » **Veículos Autônomos:** Stanley é um Volkswagen Touareg que foi equipado com câmeras, radares, e telômetros a *laser* para detectar o ambiente, além de um *software* embarcado para comandar a direção, frenagem e aceleração. Em 2005, Stanley competiu o DARPA *Grand Challenge*, no terreno acidentado do deserto de Mojave, onde realizou um percurso de mais de 210 quilômetros e venceu a competição. Em 2006, o BOSS, da CMU, dirigiu pelas ruas de uma base fechada da Força Aérea Americana, obedecendo às regras de trânsito e evitando pedestres e outros veículos. Hoje em dia é ainda mais comum encontrar relatos de carros autônomos, que vêm sendo utilizados por empresas de entrega em domicílio, empresas de transporte de alimentos, entre outras. Leia mais sobre o Stanley em Thrun *et al.* (2006).
- » **Visão Computacional:** a Google utiliza um sistema de pesquisa reversa de imagens para buscar por imagens visualmente similares pela *web*. A visão computacional trata dos sistemas que manipulam dados visuais como imagens e vídeos. Eles são capazes de entender o conteúdo desses dados e extrair informações baseados no problema apresentado.



Em 27/2/2019, Bill Gates publicou em sua página do *Linkedin* um artigo intitulado “10 tecnologias que farão manchetes em 2019”. Entre elas, duas são do ramo da AI: Tecnologia para destreza robótica e Assistentes de IA com fala natural.

Você pode verificar a chamada de Bill Gates no *Linkedin*. Disponível em: <https://www.linkedin.com/pulse/10-technologies-make-headlines-2019-bill-gates/>. Ou checar a lista completa no *site* do MIT. Disponível em: <https://www.technologyreview.com/lists/technologies/2019/>.

## Áreas de atuação

É necessário compreender as diversas ramificações da inteligência artificial, a fim de escolher corretamente qual paradigma utilizar para a solução de dado problema do mundo real. Com isso, de acordo com Joshi (2017), segue uma lista de áreas de atuação dominantes no campo da AI:

- » **Aprendizado de Máquina e Reconhecimento de Padrões:** um dos campos mais populares da AI, nesta área, são desenvolvidos *softwares* que podem aprender

com os dados que lhe são fornecidos. Baseado nos modelos de aprendizagem, é possível fazer inferências e previsões em novos dados. Neste campo da AI, o foco principal são os dados, se o conjunto de dados é demasiadamente pequeno, isso pode limitar os modelos de aprendizagem.

- » **Busca:** técnicas de busca são frequentemente utilizadas em programas de Inteligência Artificial, são sistemas que analisam um grande número de possibilidades e escolher a que melhor se adéqua ao problema. Essas técnicas são utilizadas especialmente em jogos de estratégia (como xadrez), redes, alocação de recursos, agendamento automático etc.
- » **Heurística:** esta é uma técnica utilizada na resolução rápida de determinado problema prático, porém, não garante que a solução apresentada é a mais ideal. É aplicada quando não há a possibilidade de analisar todas as possibilidades para escolher a melhor opção. Comumente utilizada em robótica, mecanismos de busca etc.
- » **Inteligência Artificial Baseada em Lógica:** também conhecido como AI simbólica, neste campo, a AI se utiliza de lógica matemática para executar programas de computador. Um sistema desenvolvido baseado em lógica é composto por um conjunto de regras lógicas para determinado problema a ser solucionado. É usada principalmente em correspondência de padrões, análise de idioma, análise semântica etc.
- » **Planejamento:** esta vertente da AI busca realizar o planejamento ideal, trazendo o máximo de retorno com o mínimo custo. São programas que recebem fatos sobre uma situação específica e um objetivo a ser alcançado, além de estarem cientes de diversos fatos do mundo real, a fim de saberem quais regras devem ser seguidas. Com base nessas informações, os sistemas traçam o plano mais ideal para atingir o objetivo determinado.
- » **Programação Genética:** é a forma de desenvolver programas que solucionem uma tarefa, a partir da combinação de outros programas e seleção dos mais aptos. Os programas são codificados como um conjunto de genes, e utilizam um algoritmo para selecionar aquele que seja capaz de realizar melhor a tarefa proposta.
- » **Representação do Conhecimento:** é necessário que os acontecimentos do mundo real sejam representados de forma que façam sentido para os sistemas, geralmente, essa representação é feita através da lógica matemática, de modo que, quando o conhecimento é representado de forma eficiente ao sistema, ele se torna mais inteligente. Geralmente, a representação formal das propriedades e

relacionamentos das entidades de um domínio específico é feita através de uma taxonomia ou estrutura hierárquica.



Na Inglaterra vitoriana, Lady Ada Lovelace era uma amiga e colaboradora de Charles Babbage, o inventor do **Mecanismo Analítico**: o primeiro computador mecânico de propósito geral. Embora visionário e muito à frente de seu tempo, o mecanismo analítico não foi concebido como um computador de uso geral quando foi projetado nas décadas de 1830 e 1840, porque o conceito de computação de propósito geral ainda não havia sido inventado. Significava apenas uma maneira de usar operações mecânicas para automatizar certos cálculos do campo da análise matemática – daí o nome Mecanismo Analítico.

Em 1843, Ada Lovelace comentou sobre a invenção: “O Mecanismo Analítico não tem pretensões de originar nada. Ele pode fazer apenas o que nós sabemos como instruí-lo a fazer... Seu papel é nos ajudar a disponibilizar aquilo com que já estamos familiarizados.”

Esse comentário foi mais tarde citado pelo pioneiro da AI, Alan Turing, como “objeção de Lady Lovelace”, em seu importante trabalho de 1950, “*Computing Machinery and Intelligence*” (TURING, 1950), que introduziu o teste de Turing, assim como os principais conceitos que viriam a formar a inteligência artificial.

Turing estava citando Ada Lovelace enquanto ponderava se os computadores de uso geral poderiam ser capazes de aprender e originar coisas, e chegou à conclusão de que eles poderiam.

O aprendizado de máquina surge a partir dessas perguntas:

- » Um computador poderia ir além do “que nós sabemos como instruí-lo a fazer” e aprender sozinho como realizar uma tarefa específica?
- » Um computador poderia nos surpreender?
- » Em vez de os programadores criarem manualmente as regras de processamento de dados, um computador poderia aprender automaticamente essas regras observando os dados?

# CAPÍTULO 2

## APRENDIZADO DE MÁQUINA

Desde que os computadores foram inventados, o ser humano tem buscado como programá-los para que aprendam automaticamente, com sua própria experiência. O Aprendizado de Máquina (ML, do inglês *Machine Learning*) é uma das áreas da Inteligência Artificial que mais vem se desenvolvendo na atualidade. Imagine os benefícios de se ter um computador que aprenda, a partir de registros médicos, qual é o melhor tratamento para as doenças de pacientes; uma casa totalmente automatizada que aprende, a partir da rotina de seu usuário, como otimizar os gastos de energia, a reposição de produtos, a solicitação de serviços; um computador que consegue analisar imagens de satélites e mapear áreas de desmatamento, de crescimento exagerado das cidades, e propor meios para minimizar os impactos ambientais... Enfim, as aplicações deste campo de estudo são inúmeras, e podem trazer muitos benefícios e facilidades para o ser humano.

Aprendizado de Máquina pode ser definido como um conjunto de métodos computacionais que se utilizam da experiência adquirida para melhorar seu desempenho na realização de tarefas, ou fazer previsões consistentes sobre informações não conhecidas (MITCHELL, 1997). Os sistemas adquirem experiência através de informações já analisadas, de modo que, quanto mais dados são analisados pelo sistema, mais ele aprende. O principal objetivo do ML é desenvolver algoritmos que sejam eficientes e mostrem eficácia e precisão em suas previsões. Os algoritmos de ML objetivam relacionar dados desconhecidos com dados previamente analisados, eles buscam sempre aumentar ao máximo a capacidade do sistema de se generalizar, ou seja, de dar respostas corretas com relação a dados não presentes na fase de treinamento. Infelizmente, ainda não conseguimos fazer os computadores aprenderem da mesma forma que o ser humano, mas os algoritmos desenvolvidos são eficazes para o aprendizado de tarefas específicas.

O principal objetivo do Aprendizado de Máquina é o desenvolvimento de previsões corretas para itens desconhecidos e, principalmente, produzir algoritmos que sejam eficientes e robustos para realizar essas previsões, inclusive para problemas em grande escala, ou com grande quantidade de dados para serem analisados. Para atingir estes objetivos, diversas técnicas foram desenvolvidas, incluindo o **Aprendizado Profundo**.

### Conceitos matemáticos básicos

- » **Escalar:** diz respeito a um elemento de um vetor. É um número real utilizado para definir um espaço vetorial. É um local de armazenamento de valores, e pode ser considerado como sinônimo de “variável”.

- » **Vetor:** é uma estrutura que contém  $n$  valores (ou escalares) ordenados sequencialmente. A quantidade de elementos que um vetor contém é chamada de ordem (ou comprimento).
- » **Matriz:** é um conjunto de vetores de mesma dimensão (número de colunas) agrupados. Uma matriz é composta por duas dimensões (bidimensional), linhas e colunas. Pode-se dizer que uma matriz possui tamanho  $n \times m$ , onde  $n$  é a quantidade de linhas e  $m$  é a quantidade de colunas da matriz.
- » **Tensor:** é uma matriz multidimensional, de modo que, além de possuir linhas e colunas, ainda possui outras dimensões.

## Principais aplicações

Segundo Mohri, Rostamizadeh e Talwalkar (2012), os algoritmos de Aprendizagem têm obtido êxito em muitas aplicações, incluindo a classificação de documentos (textos, imagens, áudio etc.); processamento natural de linguagem; reconhecimento e síntese de fala; reconhecimento de caracteres (OCR, do inglês *Optical Character Recognition*); biologia computacional; visão computacional; detecção de fraudes (cartão de crédito, *spam*, telefonemas etc.); desenvolvimento de jogos; veículos autônomos (robôs; sistema de navegação etc.); diagnósticos médicos; mecanismos de busca e sistemas de recomendações, entre diversas outras. É impossível apresentar uma lista fechada de todas as aplicações de Aprendizado de Máquina, principalmente porque, diariamente, as técnicas de ML são aplicadas em novos contextos.

É possível, então, subdividir o campo do Aprendizado de Máquina em subáreas que englobam técnicas com possibilidades infinitas de aplicação, dependendo do contexto dos dados. São elas:

- » **Classificação:** o sistema é capaz de determinar uma categoria para cada objeto analisado. As categorias são definidas pelo usuário do sistema, ou pelo próprio sistema, a partir do aprendizado prévio. O número de categorias pode variar, mas é relativamente pequeno na maioria dos casos. Porém, dependendo do problema, esse número pode ser bastante grande, ou até ilimitado, em casos específicos, em especial nas áreas de OCR, classificação textual e reconhecimento de fala.
- » **Regressão:** permite a previsão de um valor real para cada item analisado. O sistema aprende com as previsões erradas através de penalidades que são aplicadas de acordo com o tamanho da diferença entre os valores tidos como corretos e os previstos. Geralmente essas técnicas são aplicadas na previsão de valores de estoque ou variações econômicas.



- » **Ranqueamento:** possibilita a ordenação dos dados de acordo com critérios preestabelecidos. É muito comum seu uso em mecanismos de busca, que retornam um conjunto de possíveis respostas a uma consulta, classificadas de acordo com a relevância em determinados contextos.
- » **Agrupamento:** separa os dados em grupos homogêneos. São aplicadas a grandes conjuntos de dados a serem analisados, de modo a agrupar estes dados de acordo com suas características relevantes ao problema proposto. Para verificar a eficácia de um agrupamento, é importante entender que os elementos de um grupo devem ter sempre características semelhantes (análise intragrupo), e cada grupo deve conter dados muito diferentes dos demais (análise intergrupos), caso isso ocorra, o agrupamento foi eficaz.
- » **Redução de Dimensionalidade ou Aprendizado Múltiplo:** manipula os dados, reduzindo seu tamanho, mas mantendo suas propriedades iniciais relevantes. Um exemplo de uso muito comum desta técnica é o pré-processamento de imagens digitais, no campo da visão computacional.

## Definições e terminologias

No âmbito do Aprendizado de Máquinas, é importante definir algumas terminologias, a fim de esclarecer a utilização e a avaliação de seus algoritmos. Seguem alguns conceitos intrínsecos de ML:

- » **Exemplos:** itens ou instâncias de dados usados na fase de treinamento para que o algoritmo possa aprender com suas características, ou na fase de avaliação, para testar o algoritmo.
- » **Características:** conjunto de atributos associados aos exemplos. Comumente são apresentados em forma de vetor.
- » **Rótulos:** valores ou categorias que são atribuídos aos exemplos. No caso de Algoritmos de Classificação, os exemplos são definidos em categorias, já na Regressão, os exemplos recebem valores reais já predefinidos.
- » **Conjunto de Treinamento:** são exemplos utilizados no treinamento, levando o algoritmo a aprender algo. O conjunto de treinamento pode variar de acordo com o problema a ser abordado.
- » **Conjunto de Validação:** são exemplos utilizados para ajustar os parâmetros (um ou mais) presentes nos algoritmos de Aprendizado de Máquina que trabalham com dados rotulados.



- » **Conjunto de Teste:** são exemplos utilizados para avaliar a performance do algoritmo de Aprendizado de Máquina. O conjunto de testes deve ser diferente dos conjuntos de treinamento e validação, sendo desconhecida pelo algoritmo. O conjunto de teste é apresentado ao algoritmo que faz a previsão de rótulos para seus elementos levando em consideração o que já aprendeu nas fases de treinamento e validação, essa previsão é, então, comparada aos rótulos já conhecidos do conjunto de testes, para medir o desempenho do algoritmo.
- » **Função de Perda:** é a função que mede a diferença, ou perda, entre o rótulo previsto e o rótulo real no conjunto de testes, de modo que, ao se denotar o conjunto de todos os rótulos como  $y$ , e o conjunto de predições como  $y'$ , a função de perda  $L$  pode ser escrita como  $L : y \times y' \rightarrow \mathbb{R}_+$ . Na maior parte dos casos,  $y' = y$  e a função de perda é limitada, mas nem sempre. Alguns exemplos de Funções de Perda mais utilizadas são:
  - › Zero-um, também conhecido como classificação incorreta é definida como  $\{-1, +1\} \times \{-1, +1\}$  por  $L(y, y') = 1_{y' \neq y}$ .
  - › Perda quadrática que é definida como  $I \times I$  por  $L(y, y') = (y' - y)^2$ , neste caso, quando  $I \subseteq \mathbb{R}$  o intervalo é limitado.
- » **Conjunto de Hipótese:** conjunto de funções para mapear as características (vetor de características) para o conjunto de rótulos  $y$ . Mais comumente, são funções de mapeamento de características para um novo conjunto  $y'$ . Podem ser utilizadas, por exemplo, as funções lineares.

A partir dessas terminologias, é possível estipular uma linha de execução-padrão para os algoritmos de Aprendizado de Máquina. Inicialmente, tem-se um conjunto de dados rotulados que deverá ser subdividido em três subconjuntos, um para o treinamento do algoritmo, um para validação e outro para teste. O tamanho de cada subconjunto depende de vários fatores, por exemplo, a quantidade de parâmetros que devem ser ajustados (influência diretamente no tamanho do conjunto de validação), a quantidade de dados disponíveis, entre outros. Lembrando que é interessante que o conjunto de treinamento seja maior que os demais, em especial quando há escassez de dados, visto que é nessa fase que o algoritmo aprenderá.

Após a criação dos conjuntos de dados, são associados a eles as características relevantes à solução do problema. Este passo é essencial para as soluções usando ML, visto que características bem definidas tornam o aprendizado de dados mais efetivo. A má definição de características pode levar o algoritmo a aprender de forma incorreta. É possível deixar

que o usuário faça a definição das características, uma vez que ele possui conhecimento prévio sobre a tarefa a ser executada.

Com as características já definidas, é possível treinar o algoritmo de inteligência artificial fazendo alterações nos parâmetros livres a fim de que o algoritmo defina hipóteses para a solução do problema. Durante a fase de validação, a melhor hipótese para a solução do problema é definida. A partir dela, o algoritmo faz a predição dos rótulos na etapa de teste. Utilizando Funções de Perda, é possível avaliar a performance do algoritmo para a solução daquela tarefa. Sendo assim, a performance do algoritmo é definida na fase de teste, e não na de treinamento.

## Cenários de aprendizagem

Apesar do fluxo geral de funcionamento de um sistema de aprendizado de máquina descrito acima, nem sempre os cenários de aprendizado são iguais. Eles podem se diferenciar no tipo de dados disponíveis para treinamento, na forma como os dados para treinamento são obtidos, e nos tipos de dados disponíveis para avaliar o aprendizado do algoritmo. Seguem os principais cenários em ML, de acordo com Mohri, Rostamizadeh e Talwalkar (2012):

- » **Aprendizado Supervisionado:** o sistema recebe um conjunto de dados já rotulados para a fase de treinamento e faz previsões para os demais dados não conhecidos. É o cenário mais comum ao se tratar problemas de classificação, regressão e ranqueamento.
- » **Aprendizado Não Supervisionado:** o sistema recebe um conjunto de dados não rotulados, e faz previsão para todos eles. Visto que não existem dados rotulados, é complicado inferir sobre a performance do aprendizado do algoritmo. Problemas como agrupamento e redução de dimensionalidade são comumente resolvidos utilizando aprendizado não supervisionado.
- » **Aprendizado Semi-supervisionado:** o sistema recebe um conjunto de treinamento com dados rotulados e não rotulados e faz previsões para todos os dados desconhecidos. É utilizado quando dados rotulados têm alto custo para aquisição, enquanto a obtenção de dados não rotulados é relativamente simples. Pode ser aplicado em diversas situações, incluindo classificação, regressão e ranqueamento. Utiliza-se este cenário a fim de obter um aprendizado melhor que apenas o cenário supervisionado, porém, pesquisadores ainda estudam em quais situações ele é realmente eficaz.

- » **Inferência Transdutiva:** o sistema recebe para treinamento um conjunto de dados rotulados juntamente com o conjunto de teste sem os respectivos rótulos. O objetivo deste cenário é obter rótulos exclusivamente para os dados de teste. É um cenário simples, e pode ser utilizado na solução de diversos problemas modernos. Porém, assim como no aprendizado semi-supervisionado, ainda se tem dúvidas sobre a eficácia deste cenário.
- » **Aprendizado On-line:** ao contrário dos outros cenários, este envolve múltiplas rodadas e as fases de treinamento e teste são mescladas. A cada rodada o sistema recebe um conjunto de treinamento não rotulado, faz a predição, recebe os rótulos verdadeiros e verifica a perda. O objetivo é minimizar a perda acumulada durante todas as rodadas.
- » **Aprendizado Reforçado:** as fases de treinamento e teste se misturam e, para coletar informações, o sistema interage ativamente com o ambiente e, em alguns casos, interfere no ambiente, e recebe uma recompensa por cada ação. O objetivo é aumentar sua pontuação durante as ações e interações com o ambiente. Porém, não há fornecimento de recompensa em longo prazo pelo ambiente, então, o sistema deve decidir entre realizar ações desconhecidas e receber maior pontuação, ou explorar informações já coletadas.
- » **Aprendizado Ativo:** o sistema coleta, de forma adaptativa ou interativa, exemplos de treinamento, geralmente consultando um guia para solicitar rótulos para novos dados. O objetivo é alcançar uma performance comparável ao cenário supervisionado, porém, com menos exemplos rotulados. Este cenário é comumente utilizado quando dados rotulados têm custo elevado.

## Quando usar

Para identificar para quais tarefas os algoritmos de Aprendizado de Máquina podem ser úteis, Simeone (2018) se baseou no artigo de Brynjolfsson e Mitchell (2017) para definir oito critérios que devem ser observados:

1. A tarefa envolve funções que mapeiam de forma bem definida entradas para saídas.
2. Existem, ou podem ser criados, grandes conjuntos de dados contendo pares de entrada e saída.
3. A tarefa define um *feedback* com metas e métricas claramente definidos.
4. A tarefa não envolve longas cadeias lógicas que dependem de vasto conhecimento prévio sobre temas diversos ou de senso comum.

5. A tarefa não exige explicações detalhadas sobre a forma como a decisão foi tomada.
6. A tarefa tem certa tolerância a erros e não exige soluções comprovadamente corretas ou ótimas.
7. O fenômeno ou função que o sistema está aprendendo não deve ser alterado rápida e significativamente ao longo do tempo.
8. Não é necessária nenhuma destreza especializada, habilidades físicas ou mobilidade.

Estes critérios são úteis ao verificar se certa tarefa pode ou não ser realizada utilizando métodos de ML.

# CAPÍTULO 3

## REDES NEURAIS ARTIFICIAIS

O cérebro é o que possibilita ao ser humano guardar memórias e sentimentos, sonhar, pensar, ou seja, o que nos torna inteligentes. Desde bebê, o cérebro aprende com todo o ambiente onde a pessoa está inserida, com suas próprias experiências, ou através da observação da experiência de outras pessoas. O maior sonho da humanidade é criar máquinas que sejam capazes de fazer o mesmo que o cérebro humano. Porém, as tarefas que o cérebro pode resolver tão rapidamente, como reconhecer um rosto ou uma palavra dentro de um contexto, para os computadores não são tão simples, e não podem ser programadas de forma comum.

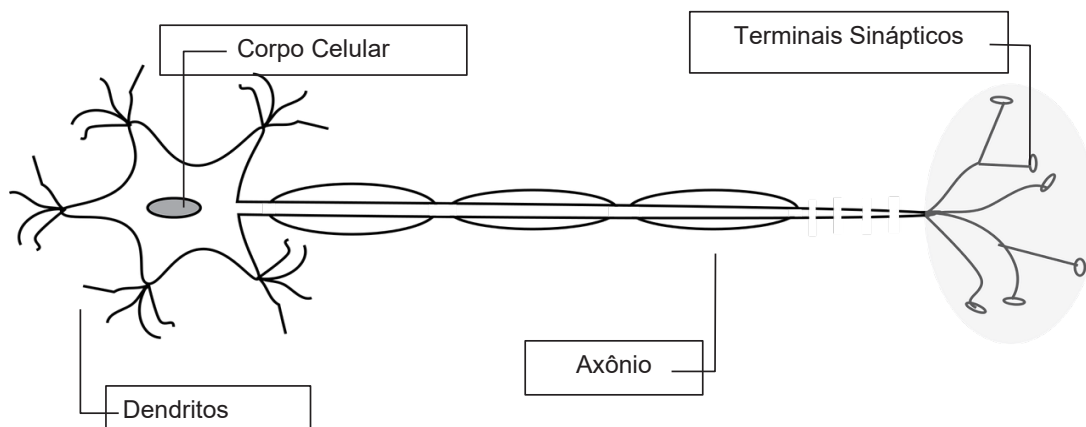
Programas de computador são muito bons em duas coisas. A primeira delas é executar rapidamente cálculos matemáticos, e a segunda é seguir uma lista de instruções, que também são conhecidas como algoritmos. Mas, e se nosso problema for algo mais abstrato do que realizar um cálculo?

Quando crianças, nós aprendemos através de exemplos. Apresentam-nos diversos exemplares de algo específico, e somos corrigidos quando erramos o reconhecimento. De acordo com Buduma (2017), quando nascemos, nosso cérebro cria modelos que descrevem como devemos ver o mundo e nos ajudam a entender todos os objetos contidos nele. Quando vamos crescendo, vários exemplos são apresentados e, quando acertamos o reconhecimento, o modelo é reforçado, quando erramos, nós corrigimos nosso modelo, adicionando a experiência negativa, tornando-o mais adequado àquela representação. Isso significa que, no decorrer de nossa vida, nossos modelos são constantemente aprimorados.

*Deep Learning*, que é o tema deste módulo, é um subcampo do campo de Inteligência Artificial conhecido como Aprendizado de Máquina, cujo propósito principal é a ideia de fazer a máquina aprender através de exemplos. Não simplesmente ensiná-la a interpretar uma lista de regras para realizar cálculos, mas dar-lhe modelos e instruções para que possa aprimorar estes modelos através de exemplos, erros e acertos. Porém, quanto mais exemplos são apresentados ao modelo, mais ele fica especializado em reconhecer certo objeto e, para manipular essa grande massa de dados, é necessário o *Deep Learning*.

A parte principal do cérebro são os neurônios. Os neurônios se conectam com outros neurônios, criando uma rede biológica que nos permite perceber o mundo ao nosso redor. Os neurônios recebem informações de outros neurônios, processam essa informação e encaminham os resultados para outros neurônios. Um neurônio, como pode ser visto na Figura 2, possui os dendritos, o corpo celular, o axônio e o terminal do axônio, também conhecido como terminal sináptico.

Figura 2. Estrutura de um Neurônio Biológico.

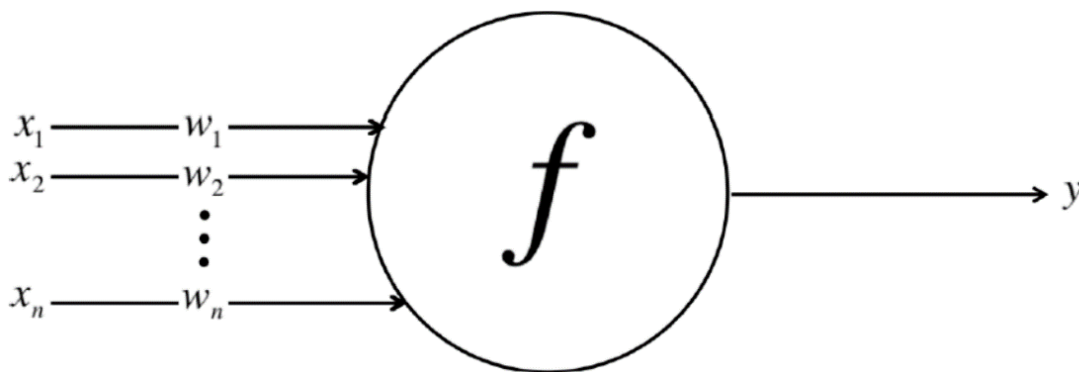


Fonte: Adaptado de Buduma, 2017, p. 7.

Durante o processo de aprendizado biológico, um neurônio recebe as entradas através dos dendritos, que são estruturas em forma de antenas. Cada conexão de entrada pode ser mais ou menos forte, dependendo da frequência de uso, e a força dessa conexão determinará qual contribuição a entrada terá na saída do neurônio. Após receber o peso baseado na força da conexão, todas as entradas são somadas dentro do corpo celular. Essa soma é novamente transformada em sinal que é propagado através dos axônios e são enviados para outros neurônios através dos terminais sinápticos.

É possível representar o funcionamento dos neurônios através de um modelo artificial (Figura 3), que poderá ser representado pelo computador. Assim como no modelo biológico, o modelo artificial recebe diversos valores de entrada ( $x_1, x_2, \dots, x_n$ ), e cada um deles é multiplicado por um peso específico ( $w_1, w_2, \dots, w_n$ ). Então, todas as entradas, já com seus respectivos pesos, são somadas ( $z = \sum_{i=0}^n w_i x_i$ ), produzindo o *logit* do neurônio, que também inclui o *bias* (uma constante). O *logit* passa pela função de transformação  $f$  produzindo a saída  $y = f(z)$ , que poderá ser transmitida para outros neurônios.

Figura 3. Estrutura de um Neurônio Artificial.



Fonte: Adaptado de Buduma, 2017, p. 7.

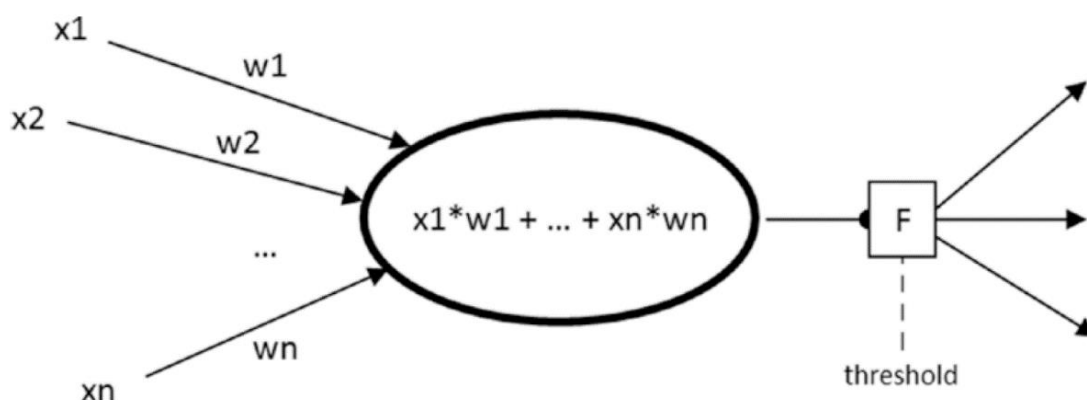
## Modelo perceptron

O modelo Perceptron é um classificador linear, ou seja, uma rede neural que não possui camadas ocultas. Neste modelo, cada perceptron recebe diversas entradas, calcula a soma ponderada dessas entradas vezes seus respectivos pesos, e produz uma única saída 0 ou 1, que mapeia a entrada  $x$  através da função de ativação:

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

onde  $T$  é um valor conhecido como limite (*threshold*), que é utilizado para comparar a soma ponderada das entradas com o valor de limite, a fim de definir a saída. Sendo assim, o perceptron calcula a soma ponderada das entradas e pesos, submete esse valor a uma função de ativação  $F$ , e envia o resultado aos neurônios da próxima camada, caso o resultado seja maior ou igual ao valor de limite, conforme apresentado na Figura 4.

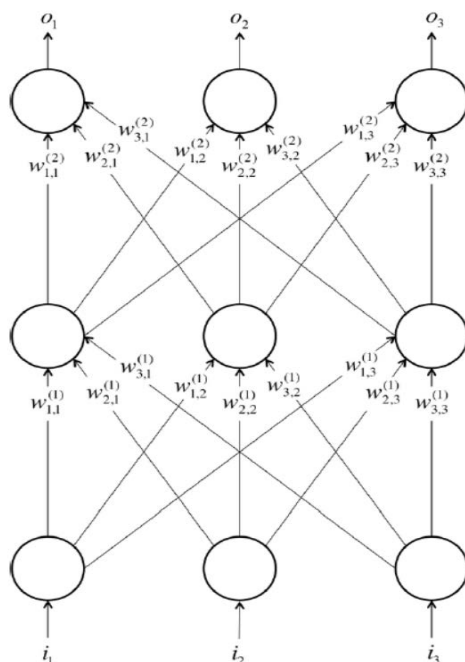
Figura 4. Modelo Perceptron.



Fonte: Pérez Castaño, 2018, p. 415.

## Redes neurais feedforward

No cérebro humano, os neurônios são organizados em camadas, pois um único neurônio não é capaz de resolver problemas complexos. As informações são passadas de uma camada a outra, até que os sinais de entrada sejam transformados em algo compreensível. Tendo como base estes conceitos biológicos, é possível construir uma Rede Neural Artificial.

Figura 5. Rede Neural *Feedforward* com três camadas.

Fonte: Buduma, 2017, p. 10.

Uma Rede Neural Artificial acontece quando ligamos neurônios uns aos outros, aos dados de entrada e aos nós de saída, que são a resposta da rede a certo problema de aprendizado. A Figura 5 mostra um exemplo de Rede Neural *Feedforward* com três camadas (entrada, uma camada oculta e saída) e três neurônios por camada, nela, a camada inicial (base) recebe a entrada de dados, e a última camada (superior) de neurônios são os nós de saída, que calculam a resposta final. A camada intermediária, que em outras redes do mesmo tipo podem ser mais de uma, são chamadas de camadas ocultas. Nas camadas ocultas são definidos os pesos  $w$  de modo que  $w_{ij}^{(k)}$  é o peso na conexão entre o neurônio  $i$  da camada  $k$  com o neurônio  $j$  da camada  $k + 1$ . Os pesos formam o vetor de parâmetros  $\theta$ , e a capacidade da Rede Neural de resolver um problema está em sua propriedade de encontrar os melhores valores para conectar em  $\theta$ .

O exemplo apresentado na Figura 5 mostra uma rede em que os neurônios da camada de entrada se conectam com a próxima camada até a conexão com a camada de saída. Porém, como pode ser observado, não há conexão de um neurônio com outro de mesma camada, tampouco há uma conexão reversa entre camadas superiores com camadas inferiores. Este tipo de rede é chamado de Redes *Feedforward*, que são um dos tipos mais simples de Redes Neurais Artificiais.

Buduma (2017) apresenta algumas observações importantes sobre esse tipo de Rede Neural Artificial:

- » Nas camadas ocultas, que estão entre a primeira camada de neurônios (camada de entrada) e a última camada de neurônios (camada de saída), é onde as características



dos dados de entrada são processadas a fim de gerar uma saída mais apropriada. O processo de extração das características da entrada é feito de forma automática pelas camadas ocultas, o que permite que a Rede Neural aprenda essas características.

- » Apesar de no exemplo apresentado pela Figura 5 as camadas ocultas apresentarem a mesma quantidade de neurônios em todas as camadas, geralmente as camadas ocultas possuem menos neurônios que as camadas de entrada, a fim de que a rede aprenda representações compactadas da entrada original.
- » Não é necessário que um neurônio tenha sua saída conectada como entrada de cada um dos neurônios da próxima camada.
- » As entradas e saídas são representações vetorizadas. Ou seja, mesmo que os dados brutos sejam tridimensionais, eles devem ser transformados em vetores antes de serem apresentados a uma Rede Neural.
- » É possível representar uma rede neural matematicamente como uma série de operações entre vetores e matrizes. Considerando que a entrada para a camada  $i$  de uma rede seja o vetor  $\mathbf{x} = [x_1, x_2, \dots, x_n]$ , espera-se obter, a partir da propagação da entrada pelos neurônios, como saída, o vetor  $\mathbf{y} = [y_1, y_2, \dots, y_m]$ . É necessário considerar a matriz de pesos  $W$  de tamanho  $n \times m$  e o vetor *bias* de tamanho  $m$ . Na matriz resultado da multiplicação entre o vetor de entrada e a matriz de pesos, cada coluna corresponde a um neurônio, de modo que o elemento  $j$  da coluna corresponde ao peso multiplicado pelo elemento  $j$  do vetor de entrada. Ou seja, a função de transformação  $\mathbf{y} = f(W^T \mathbf{x} + \mathbf{b})$  será aplicada a cada elemento do vetor de entrada.

## Neurônios Sigmoid, Tanh e ReLU

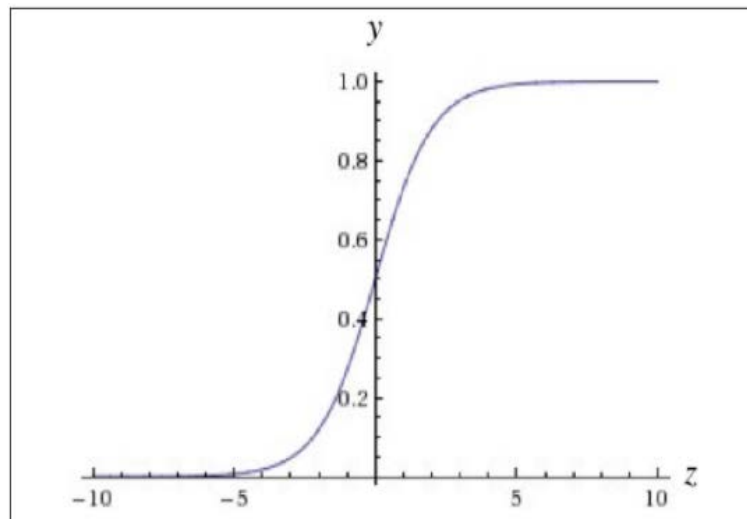
Uma rede neural sem camadas ocultas não é eficiente na solução de problemas complexos, visto que são as camadas ocultas que permitem à rede aprender as características importantes dos dados de entradas. Essas camadas ocultas, então, possuem neurônios especiais, que possibilitam o aprendizado de relações complexas através da aplicação de funções não lineares. Existem três principais tipos de neurônios que são utilizados para cálculos não lineares, o neurônio Sigmoid, o neurônio Tanh, e o neurônio de unidade linear restrita (ReLU, do inglês *Restricted Linear Unit*). Vejamos as características de cada um deles.

O neurônio Sigmoid utiliza a função:

$$f(z) = \frac{1}{1 + e^{-z}}$$

A utilização dessa função possibilita que, quando o logit é muito pequeno, a saída de um neurônio logístico será muito próxima de zero, e quando o logit é muito grande, a saída de um neurônio logístico será muito próxima de um. No entremeio destes extremos, a função assumirá a forma de um S, como pode ser visto na Figura 6.

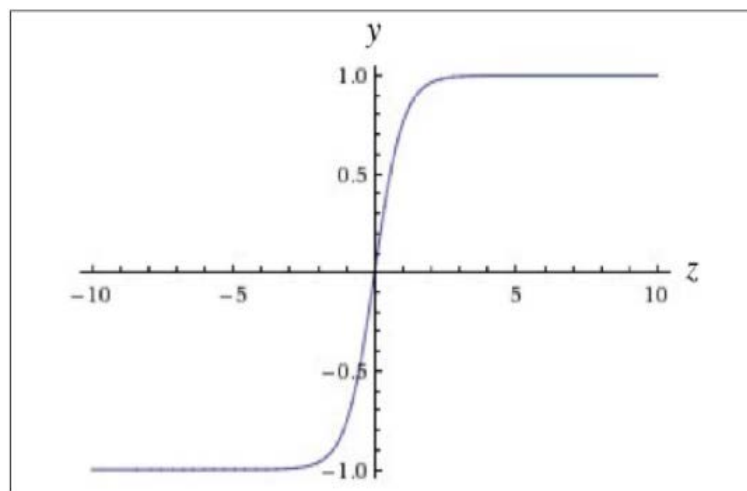
Figura 6. Neurônio Sigmoid.



Fonte: Buduma, 2017, p. 13.

O neurônio Tanh utiliza uma função não linear em formato S similar ao Sigmoid, porém, ao invés de valores entre 0 e 1, a saída dos neurônios Tanh varia entre -1 e 1, pois utiliza a função  $f(z) = \text{Tanh}(z)$ , que retorna a tangente hiperbólica de z. Quando funções não lineares de formato S são utilizadas, geralmente o neurônio Tanh é utilizado, pois tem seu centro em zero, como pode ser observado na Figura 7.

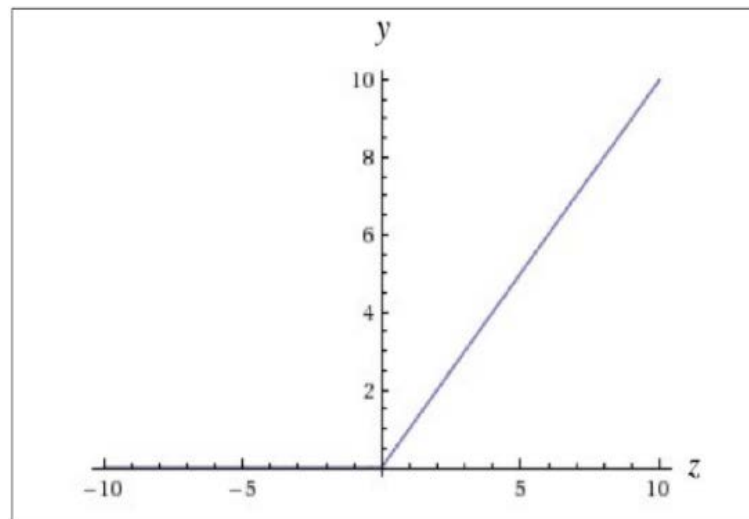
Figura 7. Neurônio Tanh.



Fonte: Buduma, 2017, p. 14.

O neurônio ReLU promove um tipo de diferente de não linearidade. Ele utiliza a função  $f(z) = \max(0, z)$ , e gera um gráfico com formato de barra, como é apresentado na Figura 8. Atualmente, este é um dos tipos de neurônios mais utilizados para resolver diversos problemas de Inteligência Artificial, em especial de Visão Computacional.

Figura 8. Neurônio ReLU.



Fonte: Buduma, 2017, p. 14.

## Camada de saída softmax

Alguns problemas trazem a necessidade de que a saída da Rede Neural seja uma distribuição probabilística para um conjunto específico de rótulos mutuamente exclusivos. Ou seja, dado um vetor de possíveis respostas, a entrada deverá ser classificada de modo que a saída da rede seja a probabilidade de que a entrada pertença a cada um dos valores presentes no vetor de respostas. Dessa forma, é possível considerar que o vetor de saída da rede deverá obedecer à seguinte regra  $\sum_{i=0}^n P_i = 1$ , onde  $[P_1, P_2, \dots, P_n]$  é o vetor de probabilidades de a entrada ser igual ao elemento  $i$  do vetor de possíveis respostas, que tem tamanho  $n$ .

Essa distribuição probabilística é obtida através de uma camada de saída chamada *Softmax*. Neste tipo de camada, a saída do neurônio depende da saída de todos os outros neurônios da camada, visto que a soma de todas as saídas deve ser, obrigatoriamente, igual a 1. Desta forma, sendo  $z_i$  o logit do  $i$ -ésimo neurônio softmax, sua normalização é obtida através da função:

$$y_i = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

De modo que uma boa acurácia na previsão possui apenas um dos valores do vetor de saída próximos a 1 e os demais, próximos a 0. Uma previsão que possui vários rótulos com probabilidades parecidas é considerada fraca ou de pouca acurácia.

## Processo de treinamento

A fim de tornar uma Rede Neural eficiente na solução de determinada tarefa, é necessário que ela seja treinada com diversos valores. Assim como o ser humano, que aprende com seus erros e acertos, uma rede neural faz o mesmo. Quanto maior o conjunto de treinamento, mais exemplos a rede terá para aprender, e o objetivo é que ela consiga minimizar seus erros, tornando-os próximos a 0. Desta forma, quanto mais perto de 0 é o erro da rede, melhor o modelo é considerado.

Para atingir erros baixos, é necessário que a rede seja capaz de selecionar o melhor conjunto de pesos para os neurônios. Por exemplo, para minimizar o erro quadrático, é necessário que  $E$  seja próximo de 0 na função:

$$E = \frac{1}{2} \sum_i (t^{(i)} - y^{(i)})^2$$

Onde,  $i$  é um elemento do conjunto de treinamento,  $T^{(i)}$  é a resposta correta,  $y^{(i)}$  é o valor calculado pela rede neural.

A fim de alcançar a redução do erro quadrático ao classificar um conjunto de treinamento  $x$ , o algoritmo leva em consideração que o gradiente, ou seja, o conjunto de todas as derivadas parciais, indica a direção do maior aumento para  $E$ . Então, ao multiplicar o gradiente por  $-1$ , é possível de se obter a direção do decréscimo mais acentuado para  $E$ , o que nos leva ao menor erro obtido. Essa forma de calcular o menor erro é chamada de Gradiente Descendente.

O algoritmo de busca do erro mínimo pelo Gradiente Descendente determina o melhor vetor de pesos ao minimizar o erro global da rede para aquele conjunto de treinamento. Inicialmente, o vetor de pesos contém valores aleatórios, que vão sendo ajustados a cada interação até que o valor do erro seja considerado aceitável.

À medida que o algoritmo se aproxima do erro mínimo, os ajustes dos pesos vão sendo cada vez mais refinados, o que pode levar muito tempo em busca do erro zero. Então, é comum multiplicar o gradiente por um fator  $\epsilon$ , também conhecido como taxa de aprendizado (*learning rate*). A taxa de aprendizado determina o quanto os coeficientes do gradiente podem ser alterados em cada iteração para ajuste dos pesos.

Similar ao algoritmo do Gradiente Descendente, o algoritmo de *Backpropagation* também busca descobrir o menor valor para erro da rede, porém, eles trabalham de forma diferente. O algoritmo de *Backpropagation* é um mecanismo que transporta, fazendo ajustes a cada camada, o erro obtido na camada de saída (quando o erro já está minimizado e os pesos ajustados) para a última camada oculta e, então, para a penúltima camada oculta, e assim por diante, fazendo o caminho inverso até atingir a primeira camada oculta. Ou seja, a saída é calculada a partir da camada de entrada, passando por todas as camadas ocultas e finalizando na camada de saída, então, assim que a saída é obtida, os pesos são ajustados no caminho inverso, ou seja, desde a saída até a primeira camada oculta (logo após a camada de entrada).

## Rede neural multicamadas

Uma rede neural multicamadas é composta de neurônios organizados em camadas de modo que a saída de um neurônio é a entrada de outro neurônio da próxima camada. A camada de entrada recebe suas entradas a partir do conjunto de treinamento  $x$ , porém, todas as outras camadas recebem como entradas os valores gerados pelas funções de ativação das camadas anteriores.

Quando as redes neurais possuem mais de uma camada, elas são chamadas de Redes Neurais Profundas, ou *Deep Neural Network*, que são comumente utilizadas em *Deep Learning*. Pérez Castaño (2018, p. 438) traz o Quadro 1, que apresenta a relação entre a quantidade de camadas ocultas e o poder da Rede Neural:

Quadro 1. Relação entre a quantidade de camadas e o Poder da RNA.

Nº de camadas ocultas	Resultado
0	Capaz de representar apenas funções lineares separáveis ou decisões.
1	Pode aproximar qualquer função que contenha um mapeamento contínuo de um espaço finito para outro.
2	Pode representar um limite de decisão arbitrário para uma precisão arbitrária com funções racionais de ativação e pode aproximar qualquer mapeamento suave a qualquer precisão.
>2	Camadas adicionais podem aprender representações complexas.

Fonte: Pérez Castaño, 2018, p. 438.



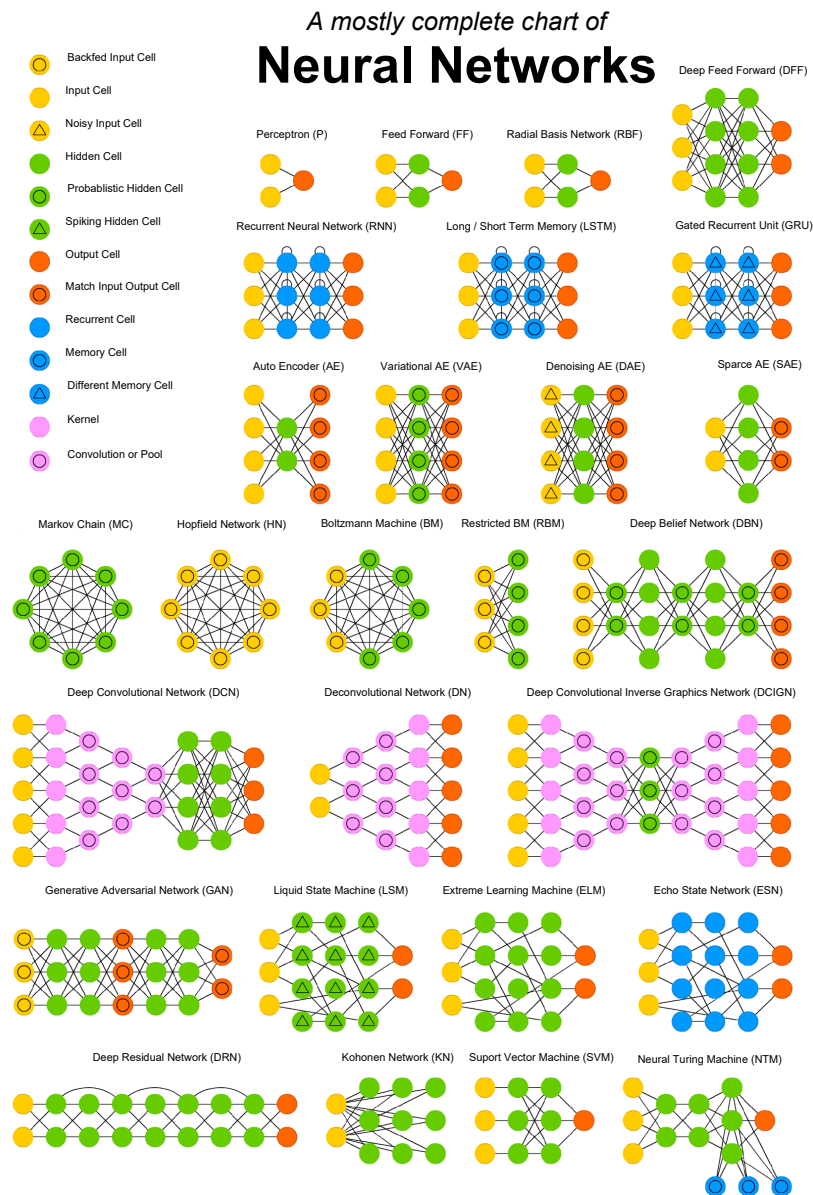
O capítulo 10 do livro digital *Deep Learning Book* (DATA SCIENCE ACADEMY, 2017) apresenta as 10 principais Arquiteturas de Redes Neurais. Algumas são apresentadas neste material, outras não.

Da mesma forma, a publicação *The Neural Network Zoo* (THE ASIMOV INSTITUTE, 2016) apresenta a Figura 9, que mostra, em forma de gráficos, os diversos tipos

de Redes Neurais. O *site* também traz a explicação para cada uma dessas redes, e indicações de artigos que as utilizam.

Como o foco de estudo deste material é *Deep Learning*, verifique os materiais para aprender mais sobre os tipos de Redes Neurais Artificiais que não trataremos aqui.

Figura 9. Diversas Redes Neurais em forma de gráfico.



Fonte: Adaptado de The Asimov Institute, 2016.

# REFERÊNCIAS

- AREL, I.; ROSE, D. C.; KARNOWSKI, T. P. Deep machine learning – a new frontier in artificial intelligence research. **IEEE Computational intelligence magazine**, v. 5, n. 4, pp. 13-18, 2010.
- BENGIO, Y. Learning deep architectures for AI. **Foundations and trends® in machine learning**, v. 2, n.1, p. 1-127, 2009.
- BENGIO, Y.; COURVILLE, A.; VINCENT, P. representation learning: a review and new perspectives. **IEEE Transactions on pattern analysis and machine intelligence**, v. 35, n.8, pp. 1.798-1.828, 2013.
- BRYNJOLFSSON, E.; MITCHELL, T. M. What can machine learning do? Workforce Implications. **Science**, v. 358, n. 6.370, pp. 1.530-1.534, 2017.
- BUDUMA, N. **Fundamentals of deep learning**: designing next-generation machine intelligence algorithms. Sebastopol, CA: O'Reilly Media, 2017.
- CHO, K.; MERRIËNBOER, B. van; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *In*: Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar. **Anais...** Doha, Qatar: 2014.
- CHOLLET, F. **Deep learning with python**. New York: Manning Publications, 2018.
- DATA SCIENCE ACADEMY. **Deep learning book**. [s.l.] Data Science Academy, 2017.
- DECHTER, R. Learning while searching in constraint-satisfaction-problems. *In*: Conference Proceedings of the Association for the Advancement of Artificial Intelligence 1986, **Anais...**1986.
- DENG, L. An overview of deep-structured learning for information processing. *In*: Proceedings of Asian-Pacific Signal & Information Processing Annual Summit and Conference (APSIPA-ASC), Xi'an. **Anais...** Xi'an: 2011. Disponível em: <https://www.microsoft.com/en-us/research/publication/an-overview-of-deep-structured-learning-for-information-processing/>. Acesso em: 27/05/2019.
- DENG, L. *et al.* Recent advances in deep learning for speech research at Microsoft. *In*: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing – Proceedings, Vancouver, BC, Canadá. **Anais...** Vancouver, BC, Canadá: 2013.
- DENG, L.; YU, D. Deep Learning: Methods and Applications. **Foundations and trends® in signal processing**, v. 7, n. 3-4, pp. 197-387, 2013.
- DENTON, E.; CHINTALA, S.; SZLAM, A.; FERGUS, R. Deep generative image models using a laplacian pyramid of adversarial networks. *In*: Neural Information Processing Systems (NIPS), Montreal, Canadá. **Anais...** Montreal, Canadá: 2015. Disponível em: <http://arxiv.org/abs/1506.05751>.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep learning**. [s.l.] MIT Press, 2016.
- GOODFELLOW, I.; POUGET-ABADIE, J.; MIRZA, M.; XU, B.; WARDE-FARLEY, D.; OZAIR, S.; COURVILLE, A.; BENGIO, Y. Generative adversarial nets. *In*: Advances in Neural Information Processing Systems 27 (NIPS 2014), **Anais...**2014. Disponível em: <https://papers.nips.cc/paper/5423-generative-adversarial-nets>. Acesso em: 27/5/2019.



## REFERÊNCIAS

- HINTON, G. E. Learning Multiple Layers of Representation. **Trends in cognitive sciences**, v. 11, n. 10, pp. 428-434, 2007.
- HINTON, G. E. *et al.* Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. **IEEE Signal Processing Magazine**, v. 29, n. 6, pp. 82-97, 2012.
- HINTON, G. E.; OSINDERO, S.; TEH, Y. W. A fast learning algorithm for deep belief nets. **Neural computation**, v. 18, n. 7, pp. 1.527-1.554, 2006.
- JAIN, S. **An Overview of Regularization Techniques in Deep Learning (with Python code)**. 2018. Disponível em: <https://www.analyticsvidhya.com/blog/2018/04/fundamentals-deep-learning-regularization-techniques/>. Acesso em: 27/05/2019.
- JOSHI, P. **Artificial intelligence with python: build real-world artificial intelligence applications with python to intelligently interact with the world around you**. Birmingham, UK: Packt Publishing, 2017.
- KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Convolutional neural networks imagenet classification with deep convolutional neural network. **Communications of the Acm**, v. 60, n. 6, pp. 84-90, 2017.
- LECUN, Y. **Generalization and network design strategies technical report CRG-TR-89-4**. [s.l: s.n.].
- LEE, H.; GROSSE, R.; RANGANATH, R.; NG, A. Y. Unsupervised learning of hierarchical representations with convolutional deep belief networks. **Communications of the ACM**, v. 54, n. 10, pp. 95-103, 2011.
- LIMA, I.; PINHEIRO, C. A. M.; SANTOS, F. A. O. **Inteligência artificial**. Rio de Janeiro: Elsevier, 2014.
- MCCULLOCH, W. S.; PITTS, W. A Logical Calculus of the Ideas Immanent in Nervous Activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, pp. 115-133, 1943.
- MIRZA, M.; OSINDERO, S. **Conditional generative adversarial nets**. [s.l: s.n.]. Disponível em: <http://arxiv.org/abs/1411.1784>. Acesso em: 27/05/2019.
- MITCHELL, T. M. **Machine learning**. [s.l: s.n.]
- MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. Cambridge, MA: The MIT Press, 2012.
- OHLSSON, S. **Deep learning: how the mind overrides experience**. Cambridge: Cambridge University Press, 2011.
- PATTERSON, J.; GIBSON, A. **Deep Learning: A practitioner's approach**. Sebastopol, CA: O'Reilly Media, 2017.
- PÉREZ CASTAÑO, A. **Practical artificial intelligence: machine learning, bots, and agent solutions using C#**. New York, NY: Apress, 2018.
- RADFORD, A.; METZ, L.; CHINTALA, S. **Unsupervised representation learning with deep convolutional generative adversarial networks**. [s.l: s.n.]. Disponível em: <http://arxiv.org/abs/1511.06434>. Acesso em: 27/05/2019.
- RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach**. 3ª ed. New Jersey: Pearson Education, 2010.



- SIMEONE, O. **A brief introduction to machine learning for engineers**. London, England: King's College London, 2018.
- SUTSKEVER, I.; VINYALS, O.; LE, Q. V. Sequence to Sequence Learning with Neural Networks. *In*: NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, Canadá. **Anais...** Montreal, Canadá: 2014. Disponível em: <http://arxiv.org/abs/1409.3215>. Acesso em: 27/05/2019.
- THE ASIMOV INSTITUTE. **The neural network zoo**. Disponível em: <http://www.asimovinstitute.org/neural-network-zoo/>. Acesso em: 27/5/2019.
- THRUN, S. *et al.* Stanley: The robot that won the DARPA grand challenge. **Journal of Field Robotics**, v. 23, n. 9, pp. 661-692, 2006.
- TURING, A. M. Computing Machinery and Intelligence. **MIND**, v. 59, n. 236, pp. 433-460, 1950.
- VASILEV, I.; SLATER, D.; SPACAGNA, G.; ROELANTS, P.; ZOCCA, V. **Python deep learning**: exploring deep learning techniques and neural network architectures with pytorch, keras and tensorflow. 2ª ed. [s.l.] Packt Publishing, 2019.
- VINYALS, O.; TOSHEV, A.; BENGIO, S.; ERHAN, D. **Show and tell**: a neural image caption generator. *In*: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA. **Anais...** Boston, MA: 2015.
- WERBOS, P. J. **Backpropagation Through time**: what it does and how to do it. *In*: Proceedings of IEEE, **Anais...**1990.
- YU, D.; DENG, L. Deep Learning and its applications to signal and information processing. **IEEE Signal Processing Magazine**, v. 28, n. 1, pp. 145-150, 2011.