

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
CURSO SUPERIOR DE ENGENHARIA ELETRÔNICA**

**ANA CLÁUDIA BANDERCHUK**

**TÍTULO DO TCC**

**Florianópolis, 2021**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
SANTA CATARINA - CÂMPUS FLORIANÓPOLIS  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
CURSO SUPERIOR DE ENGENHARIA ELETRÔNICA**

**ANA CLÁUDIA BANDERCHUK**

## **TÍTULO DO TCC**

Trabalho de conclusão de curso submetido  
ao Instituto Federal de Educação, Ciência  
e Tecnologia de Santa Catarina como parte  
dos requisitos para obtenção do título de  
engenheiro eletrônico

Orientador:  
Prof. Dr. Fernando Santana Pacheco

**Florianópolis, 2021**

Ficha de identificação da obra elaborada pelo autor.

Allebrandt, Diesson Stefano  
**Estudo da redução da interferência eletromagnética  
de um conversor estático do tipo Buck interleaved / Diesson  
Stefano Allebrandt ; orientação de Luis Carlos Martinhago  
Schlichting; coorientação de Joabel Moia.**  
- Florianópolis, SC, 2021.  
82 p.  
**Trabalho de Conclusão de Curso (TCC) - Instituto Federal  
de Santa Catarina, Câmpus Florianópolis. Bacharelado  
em Engenharia Eletrônica. Departamento Acadêmico  
de Eletrônica.**  
Inclui Referências.

1. Compatibilidade eletromagnética. 2. Eletrônica  
de potência. 3. Buck interleaved. 4. Conversor Buck com  
célula de comutação de três estados. I. Schlichting,  
Luis Carlos Martinhago . II. Moia, Joabel . III.  
Instituto Federal de Santa Catarina. Departamento Acadêmico  
de Eletrônica. IV. Título.

# **TÍTULO DO TCC**

**ANA CLÁUDIA BANDERCHUK**

Este Trabalho foi julgado adequado para obtenção do Título de Engenheiro Eletrônico em abril de 2021 e aprovado na sua forma final pela banca examinadora do Curso de Engenharia Eletrônica do Instituto Federal de Educação, Ciência, e Tecnologia de Santa Catarina.

Florianópolis, 12 de março, 2021.

Banca Examinadora:

---

Fernando Santana Pacheco, Dr.

## RESUMO

Este trabalho apresenta

**Palavras-chave:** um. dois.

## **ABSTRACT**

This papper presents

**Keywords:**

## LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplos de problemas e seus níveis de complexidade para computadores ou seres humanos resolvê-los. . . . .	9
Figura 2 – Diagrama de Venn da Inteligência Artificial e suas áreas de estudo. . . . .	10
Figura 3 – Fluxograma que diferencia as áreas de estudo da Inteligência Artificial e suas etapas de processamento de dados. . . . .	11
Figura 4 – Ilustração das camadas de um modelo de Aprendizado Profundo. . . . .	12
Figura 5 – Comparação entre os modelos Aprendizado de Máquina e Aprendizado Profundo. . . . .	12
Figura 6 – Estrutura básica de uma Rede Neural com duas camadas. . . . .	13
Figura 7 – Diagrama de um Neurônio de uma Rede Neural. . . . .	14
Figura 8 – Exemplos de funções de ativação de um neurônio para uma Rede Neural. . . . .	15

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>8</b>
<b>1.1</b>	<b>Justificativa</b>	<b>8</b>
<b>1.2</b>	<b>Descrição do problema</b>	<b>8</b>
<b>1.3</b>	<b>Objetivo geral</b>	<b>8</b>
<b>1.4</b>	<b>Objetivos específicos</b>	<b>8</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>9</b>
<b>2.1</b>	<b>Inteligência Artificial</b>	<b>9</b>
<b>2.2</b>	<b>Redes Neurais</b>	<b>12</b>
2.2.1	Estrutura de Uma Rede Neural	13
2.2.2	Funções de Ativação	14
2.2.3	Redes Neurais Profundas	16
2.2.4	Redes Neurais Convolucionais	16
2.2.4.1	<i>You Only Look Once</i>	16
<b>2.3</b>	<b>Classificação e Detecção de defeitos</b>	<b>16</b>
<b>2.4</b>	<b>Frameworks e Bibliotecas</b>	<b>16</b>
2.4.1	OpenCV	16
2.4.2	Darknet	16
2.4.3	Flask	16
	<b>REFERÊNCIAS</b>	<b>17</b>
	<b>APÊNDICES</b>	<b>19</b>
	<b>APÊNDICE A – NOTAÇÕES DE REDES NEURAIS</b>	<b>20</b>



## **1 INTRODUÇÃO**

Placas de circuito impresso são de longe o método mais utilizado para o projeto de eletrônicos modernos. Elas consistem em um sanduíche de uma ou mais camadas de cobre intercaladas com uma ou mais camadas de material isolante (ZUMBAHLEN, 2008) e servem de suporte para os componentes eletrônicos responsáveis pelo funcionamento de um circuito eletrônico.

À medida em novas tecnologias são desenvolvidas, as placas de circuito impresso estão se tornando cada vez mais sofisticadas e delicadas (HU; WANG, 2020), de modo que a detecção de incertezas, tolerâncias, defeitos e erros de posição relativa associados ao processo de fabricação (LETA; FELICIANO; MARTINS, 2008) deve ser feita com mais cautela a fim de garantir o funcionamento do produto final.

Dessa forma, automatizar a inspeção de defeitos se tornou essencial para aprimorar a qualidade do processo de fabricação, já que técnicas de medição por visão computacional apresentam melhor regularidade, precisão e repetibilidade quando comparada a inspeção humana que, além da imprecisão e não-repetibilidade, está sujeita a subjetividade, fadiga e lentidão e está, ainda, associada a um alto custo (LETA; FELICIANO; MARTINS, 2008).

A inspeção ótica automatizada (AOI) vem sendo amplamente utilizada para detectar defeitos durante o processo de fabricação de uma PCB (CHIN; HARLOW, 1982). Conforme a evolução dessa tecnologia, três principais métodos de detecção tem se destacado: métodos comparativos, não-referenciais e híbridos (WU; WANG; LIU, 1996). O método comparativo, mais utilizado entre eles, está susceptível à interferência da iluminação e ruídos externos, além de necessitar mecanismos de alinhamento precisos para realização da comparação (HU; WANG, 2020).

### **1.1 Justificativa**

### **1.2 Descrição do problema**

### **1.3 Objetivo geral**

### **1.4 Objetivos específicos**

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo serão apresentados os principais conceitos necessários para a compreensão desse trabalho.

### 2.1 Inteligência Artificial

Apesar de ter chamado atenção nos últimos anos com a quantidade enorme de dados adquiridos e processados por grandes corporações como Google, Facebook, Amazon, e Apple (RUSSELL; MOSKOWITZ; JALAIAN, 2020), o termo ‘Inteligência Artificial’ não é tão atual assim. Ele foi utilizado pela primeira vez em 1955 por John McCarthy (ABRAHAM et al., 2021), que conduziu no ano seguinte um workshop cuja premissa central da proposta considera que o comportamento humano inteligente consiste em processos que podem ser formalizados e reproduzidos por uma máquina (DICK, 2019).

Um dos objetivos da Inteligência Artificial, de acordo com Mitchell, Michalski e Carbonell (2013), é fazer com que computadores realizem tarefas mais inteligentes de forma com que não haja necessidade dos seres humanos executá-las. Entretanto, um dos grandes desafios da área de IA atualmente é a execução de atividades consideradas simples e corriqueiras para pessoas, como reconhecimento de objetos e fala (GOODFELLOW; BENGIO; COURVILLE, 2016). Para ilustrar isso, a tabela da Figura 1 apresenta a diferença entre o nível de dificuldade que um computador ou ser humano tem para resolver determinado problema.

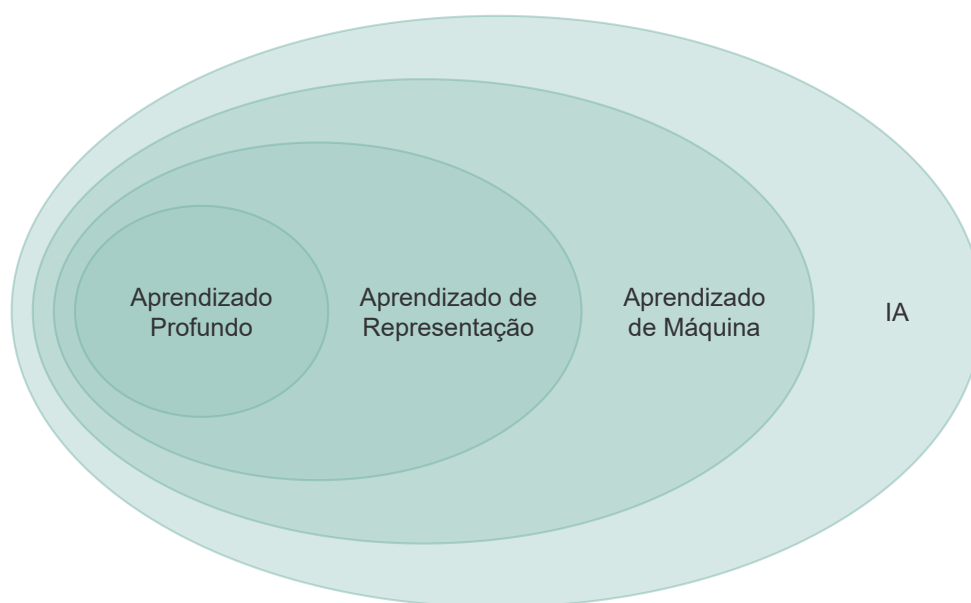
**Figura 1 – Exemplos de problemas e seus níveis de complexidade para computadores ou seres humanos resolvê-los.**

Problema	Computador	Humano
Multiplicar milhares de números grandes rapidamente	FÁCIL	DIFÍCIL
Identificar rostos em uma foto de uma multidão de pessoas	DIFÍCIL	FÁCIL

Fonte: Adaptado de Rashid (2016).

A Inteligência Artificial engloba a área de estudo do Aprendizado de Máquina, que por sua vez englobam as áreas do Aprendizado de Representação e Aprendizado Profundo, conforme o diagrama de Venn apresentado na Figura 2.

Para evidenciar as diferenças entre essas áreas de estudo, em comparação com simples Sistemas Baseados em Regras, Goodfellow, Bengio e Courville (2016)

**Figura 2 – Diagrama de Venn da Inteligência Artificial e suas áreas de estudo.**

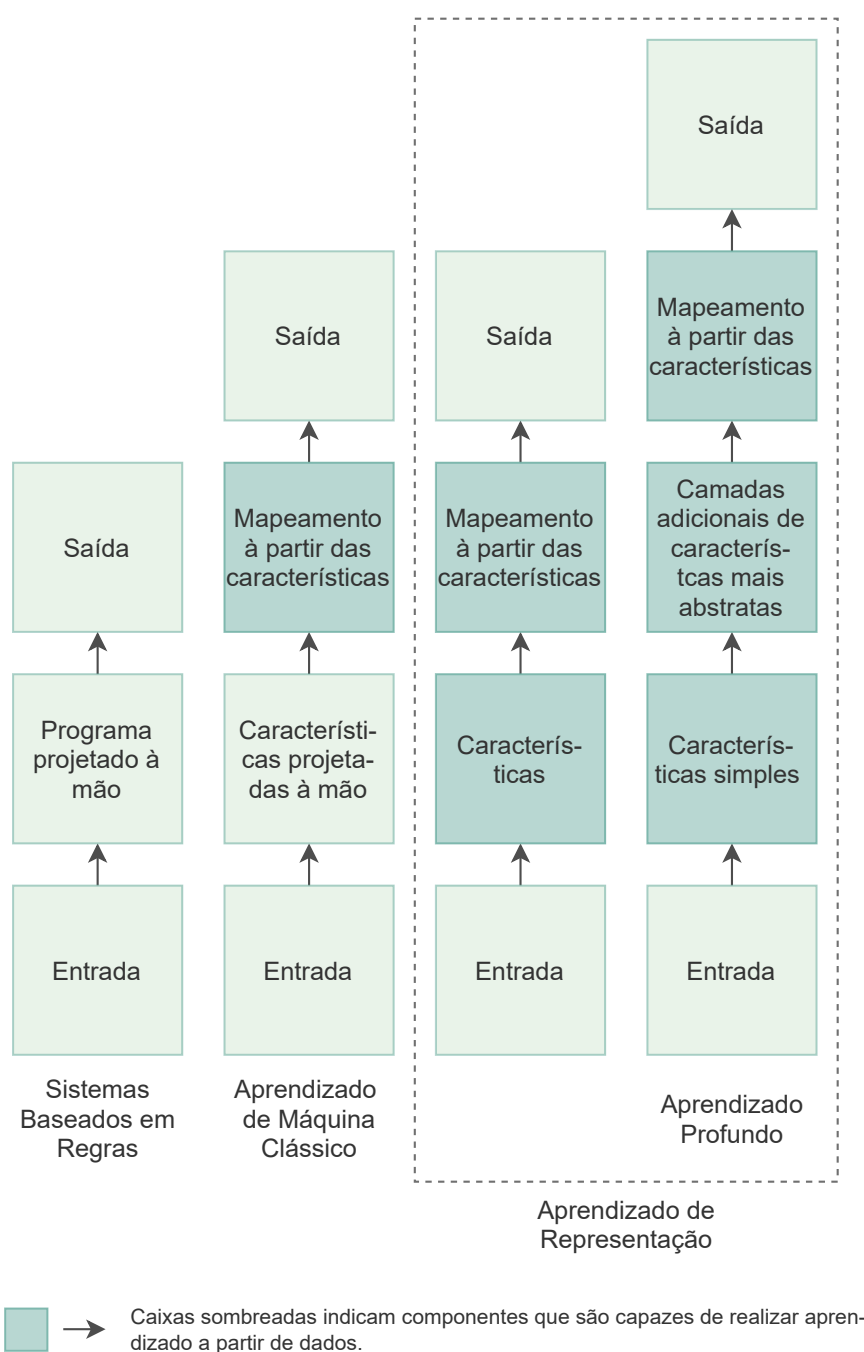
Fonte: Adaptado de Goodfellow, Bengio e Courville (2016).

apresenta um fluxograma com as etapas de processamento entre a entrada e a saída de dados para cada uma delas, presente na Figura 3.

Sistemas Baseados em Regras não possuem componentes capazes de realizar qualquer aprendizado a partir de dados (GOODFELLOW; BENGIO; COURVILLE, 2016) e são utilizados para resolução de problemas e/ou execução de tarefas que podem ser descritos por uma lista de regras formais, como por exemplo jogar Xadrez. Ao contrário desses sistemas, o Aprendizado de Máquina (do inglês *Machine Learning*) utiliza algoritmos computacionais para transformar características reunidas empiricamente em modelos utilizáveis (EDGAR; MANZ, 2017) possuindo a capacidade “de se aprimorar [...], aprendendo novos conhecimentos ao invés de serem programado com eles” (WOOLF, 2009).

No Aprendizado de Representação (do inglês *Feature Learning* ou *Representation Learning*) entretanto, não há necessidade de mapear manualmente essas características (GOODFELLOW; BENGIO; COURVILLE, 2016). Ou seja, por conta própria e de forma abstrata, os algoritmos são capazes de extrair as características importantes para a construção dos modelos utilizando redes neurais (ROBINS, 2020) (LESORT et al., 2018). O Aprendizado Profundo, por sua vez, resolve a dificuldade que o Aprendizado de Representação possui de extrair características abstratas de alto nível, tais como sotaques de um locutor (GOODFELLOW; BENGIO; COURVILLE, 2016). Segundo Mao et al. (2019), o Aprendizado Profundo “imita a função que o

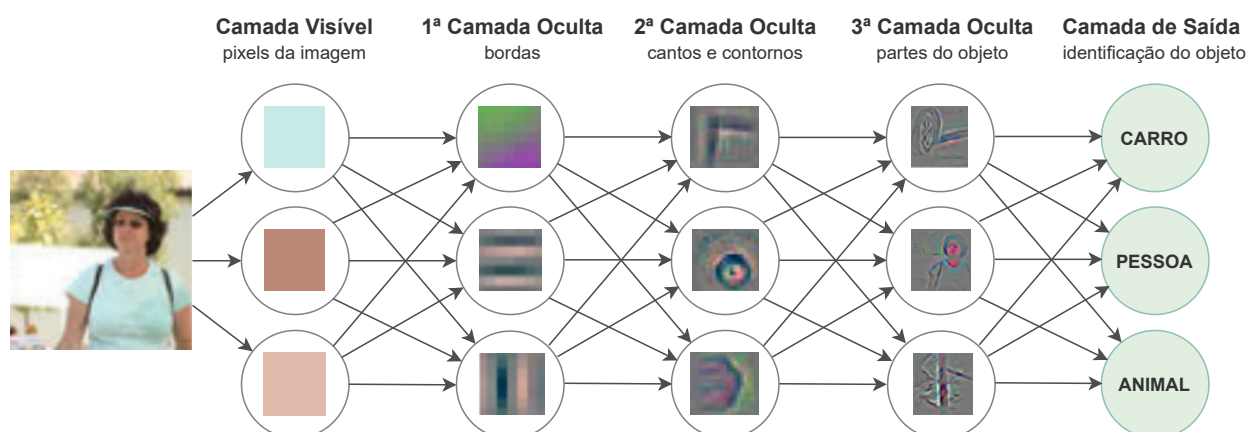
**Figura 3 – Fluxograma que diferencia as áreas de estudo da Inteligência Artificial e suas etapas de processamento de dados.**



Fonte: Adaptado de Goodfellow, Bengio e Courville (2016).

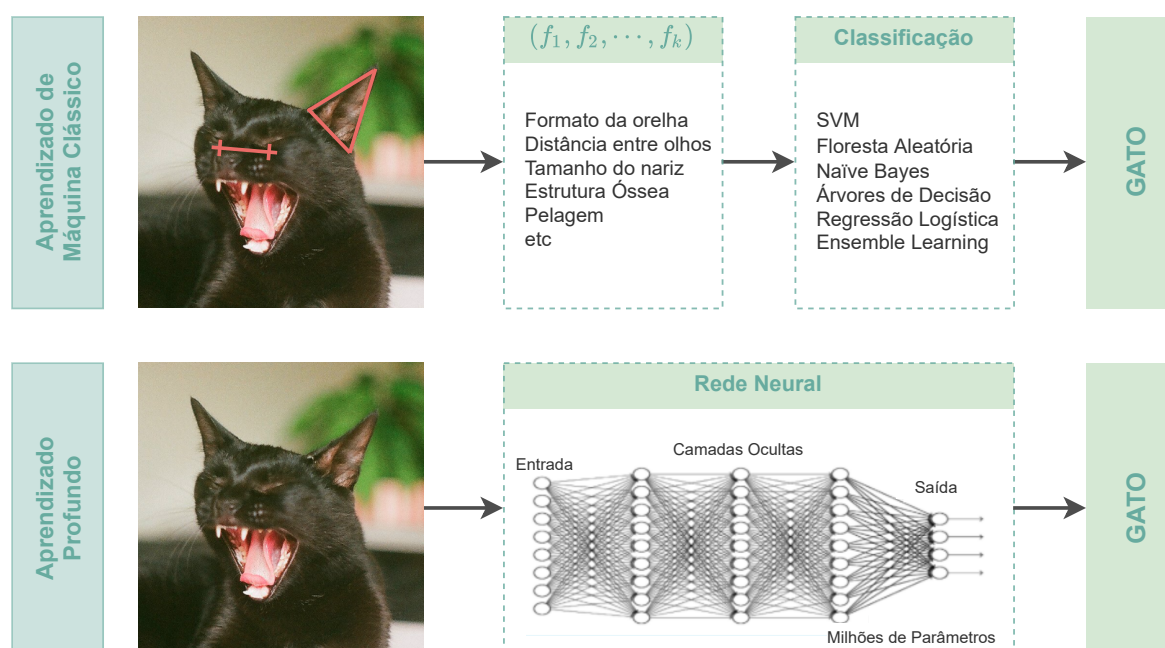
cérebro humano possui de interpretar dados usando redes neurais de várias camadas”. Um exemplo dessas variadas camadas é apresentado na Figura 4, onde características distintas são extraídas por cada camada. Uma comparação com o modelo de Aprendizado de Máquina é ilustrado na Figura 5.

Figura 4 – Ilustração das camadas de um modelo de Aprendizado Profundo.



Fonte: Adaptado de Goodfellow, Bengio e Courville (2016).

Figura 5 – Comparação entre os modelos Aprendizado de Máquina e Aprendizado Profundo.



Fonte: Adaptado de Robins (2020).

## 2.2 Redes Neurais

Ao contrário da abordagem convencional de programação, onde dizemos a um computador o que deve ser feito ao dividir um problema em pequenas tarefas para que ele execute, uma Rede Neural (do inglês *Neural Network*) utiliza dados

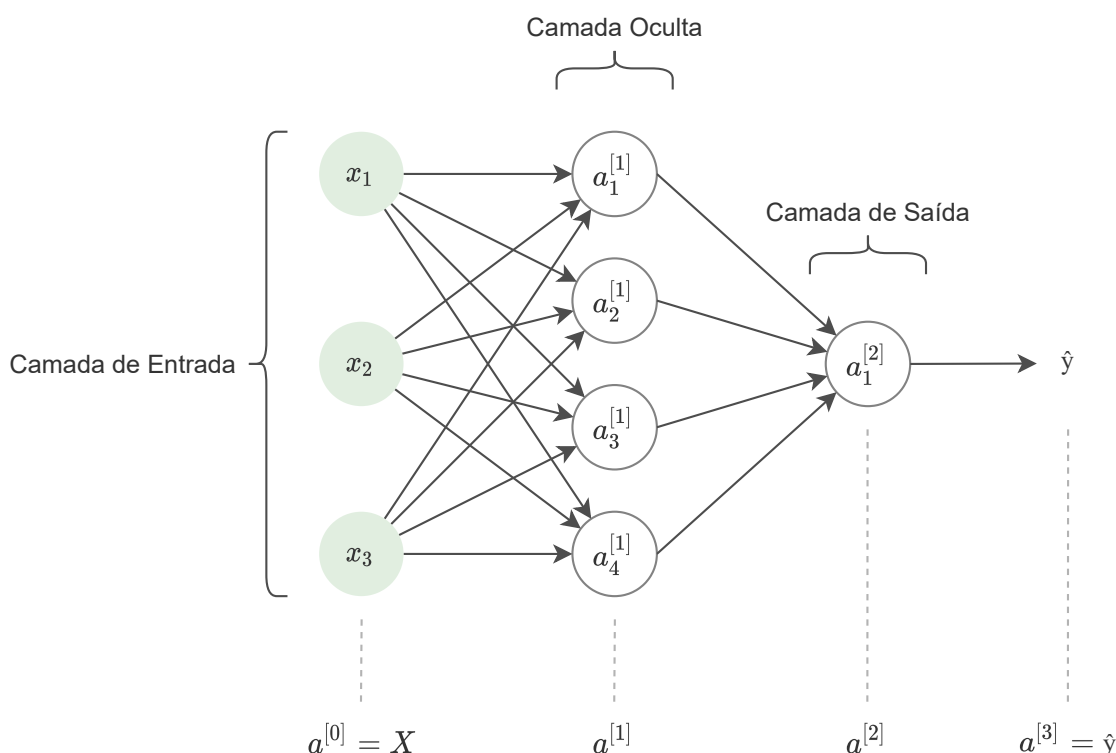
observacionais para aprender como resolver o problema (NIELSEN, 2018).

Walczak e Cerpa (2003) define Redes Neurais Artificiais como modelos que “simulam a atividade elétrica do cérebro e do sistema nervoso”. Porém enquanto alguns tipos de redes neurais tem sido utilizadas para entender o funcionamento do cérebro, na perspectiva de Aprendizado Profundo elas não são projetadas para serem modelos realistas da função biológica (GOODFELLOW; BENGIO; COURVILLE, 2016).

### 2.2.1 Estrutura de Uma Rede Neural

Uma Rede Neural é composta por camadas de nós, conhecidos como neurônios, e conexões que interligam as saídas e entradas desses nós. A estrutura básica de uma Rede Neural é ilustrada na Figura 6, onde a primeira camada de neurônios é a camada de entrada, a última camada é a camada de saída e as camadas entre elas são chamadas de camadas ocultas. Na Figura 6, vetor  $X$  contém os valores de entrada, os vetores  $a^{[l]}$  representam as funções de ativação referentes à  $l$ -ésima camada, e  $\hat{y}$  é o vetor de saída com os valores preditos. As notações utilizadas nesse capítulo respeitam as propostas por Ng (2020) presentes no Apêndice A.

**Figura 6 – Estrutura básica de uma Rede Neural com duas camadas.**

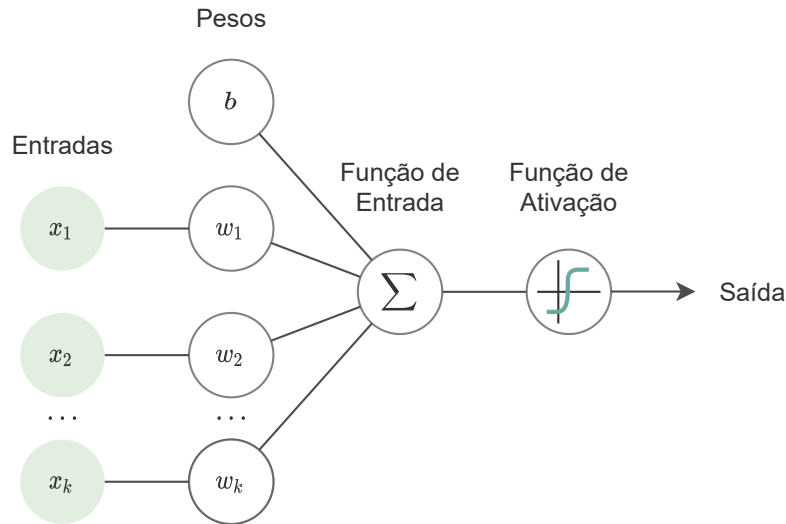


Fonte: Elaboração própria (2021).

São nos neurônios que ocorrem as computações. A Figura 7 mostra um

diagrama de um nó de uma Rede Neural, onde os pesos representados por  $b$  e  $w_k$  são responsáveis por atribuir significância às entradas com relação à tarefa que o algoritmo está tentando aprender (NICHOLSON, 2020). Esses produtos são então somados e passam por uma função de ativação.

**Figura 7 – Diagrama de um Neurônio de uma Rede Neural.**



Fonte: Adaptado de Nicholson (2020).

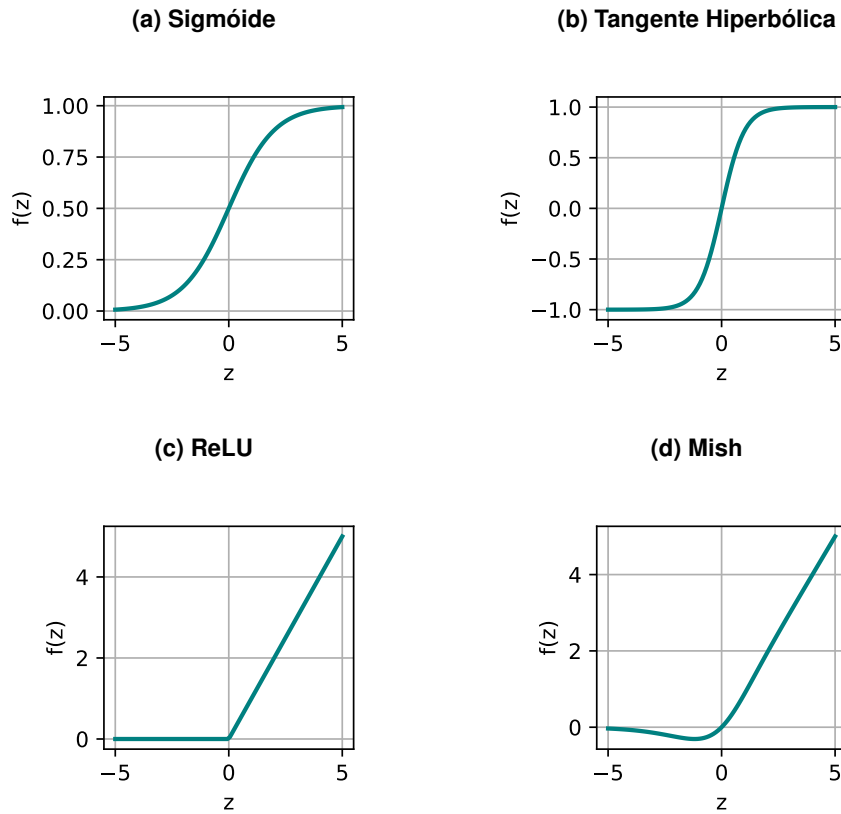
### 2.2.2 Funções de Ativação

As funções de ativação desempenham um papel crucial na dinâmica de desempenho e treinamento em redes neurais (MISRA, 2019), determinando se um sinal deve progredir e em que medida ele deve progredir através da rede (NICHOLSON, 2020). Alguns exemplos de função de ativação estão na Figura 8.

A função Sigmóide da Figura 8a, definida por Sharma (2017) pela Equação 2.1, é geralmente utilizada em classificações binárias, pois o resultado está sempre no intervalo entre zero e um (NG, 2020).

$$f(z) = \sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.1)$$

Para a otimização dos pesos durante o treinamento de uma rede neural, em geral é necessário calcular o gradiente das funções de ativação. Por isso, de acordo com Sharma (2017), o uso da tangente hiporbólica como função de ativação é preferível pois tem gradientes que não estão restritos a variar em uma certa direção e são mais

**Figura 8 – Exemplos de funções de ativação de um neurônio para uma Rede Neural.**

Fonte: Elaboração própria (2021).

acentuados quando comparado à Sigmóide. A definição da tangente hiperbólica da Figura 8b está na Equação 2.2.

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.2)$$

Porém, para valores muito grandes ou muito pequenos de  $z$ , a derivada resultante, utilizada no gradiente, tende a ser nula tanto para a sigmóide quanto para a tangente hiperbólica (NG, 2020) o que geralmente causa lentidão no treinamento (MISRA, 2019). Para que isso não ocorra, utiliza-se a função ReLU, do inglês *Rectified Linear Unit*, plotada na Figura 8c e definida na Equação 2.3. A ReLU é a função de ativação mais utilizada (NG, 2020) e é a função padrão recomendada para uso com a maioria das redes neurais (GOODFELLOW; BENGIO; COURVILLE, 2016).

$$f(z) = \text{ReLU}(z) = \max(0, z) \quad (2.3)$$

Com a evolução das Redes Neurais na última década, diversas funções de ativação vem sendo propostas com o objetivo de melhorar o desempenho do



treinamento. Um exemplo delas é a Mish (Figura 8d), proposta por Misra (2019) cuja função é definida pela Equação 2.4.

$$f(z) = x \cdot \tanh(\ln(1 + e^x)) \quad (2.4)$$

Segundo Misra (2019), ao contrário da ReLU, a função Mish é continuamente diferenciável, evitando efeitos colaterais indesejados quando utiliza-se a otimização baseada em gradiente.

### 2.2.3 Redes Neurais Profundas

### 2.2.4 Redes Neurais Convolucionais

#### 2.2.4.1 *You Only Look Once*

## 2.3 Classificação e Detecção de defeitos

## 2.4 Frameworks e Bibliotecas

### 2.4.1 OpenCV

### 2.4.2 Darknet

### 2.4.3 Flask

## REFERÊNCIAS

ABRAHAM, T. et al. Contributors. In: COHEN, S. (Ed.). *Artificial Intelligence and Deep Learning in Pathology*. Elsevier, 2021. p. xiii–xiv. ISBN 978-0-323-67538-3. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780323675383010022>>. Nenhuma citação no texto.

CHIN, R. T.; HARLOW, C. A. Automated visual inspection: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4, n. 6, p. 557–573, 1982. Nenhuma citação no texto.

DICK, S. Artificial intelligence. *Harvard Data Science Review*, v. 1, n. 1, 7 2019. <https://hdsr.mitpress.mit.edu/pub/0aytgrau>. Disponível em: <<https://hdsr.mitpress.mit.edu/pub/0aytgrau>>. Nenhuma citação no texto.

EDGAR, T. W.; MANZ, D. O. Chapter 6 - machine learning. In: EDGAR, T. W.; MANZ, D. O. (Ed.). *Research Methods for Cyber Security*. Syngress, 2017. p. 153–173. ISBN 978-0-12-805349-2. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780128053492000066>>. Nenhuma citação no texto.

GOODFELLOW, I. J.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Nenhuma citação no texto.

HU, B.; WANG, J. Detection of pcb surface defects with improved faster-rcnn and feature pyramid network. In: . [S.l.: s.n.], 2020. v. 8, p. 108335–108345. Nenhuma citação no texto.

State representation learning for control: An overview. *Neural Networks*, v. 108, p. 379–392, 2018. ISSN 0893-6080. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0893608018302053>>. Nenhuma citação no texto.

LETA, F. R.; FELICIANO, F. F.; MARTINS, F. P. R. Computer vision system for printed circuit board inspection. In: *ABCM Symposium Series in Mechatronics*. [S.l.: s.n.], 2008. v. 3, p. 623 – 632. Nenhuma citação no texto.

MAO, S. et al. Opportunities and challenges of artificial intelligence for green manufacturing in the process industry. *Engineering*, v. 5, n. 6, p. 995–1002, 2019. ISSN 2095-8099. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2095809919300074>>. Nenhuma citação no texto.

MISRA, D. Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*, 2019. Nenhuma citação no texto.

MITCHELL, R.; MICHALSKI, J.; CARBONELL, T. *An artificial intelligence approach*. [S.l.]: Springer, 2013. Nenhuma citação no texto.

NG, A. Notas de Aula, *Introduction To Deep Learning*. 2020. COURSERA: Neural Networks and Deep Learning. Disponível em: <<https://www.coursera.org/learn/neural-networks-deep-learning>>. Nenhuma citação no texto.

NICHOLSON, C. misc, *A Beginner's Guide to Neural Networks and Deep Learning*. Pathmind A.I. Wiki, 2020. Disponível em: <<https://wiki.pathmind.com/neural-network#define>>. Nenhuma citação no texto.

NIELSEN, M. A. misc, *Neural Networks and Deep Learning*. Determination Press, 2018. Disponível em: <<http://neuralnetworksanddeeplearning.com/>>. Nenhuma citação no texto.

RASHID, T. *Make Your Own Neural Network*. 1. ed. [S.l.]: CreateSpace Independent Publishing Platform, 2016. ISBN 1530826608,9781530826605. Nenhuma citação no texto.

ROBINS, M. The difference between artificial intelligence, machine learning and deep learning. *Intel Artificial Intelligence*, 5 2020. Disponível em: <<https://www.intel.com.br/content/www/br/pt/artificial-intelligence/posts/difference-between-ai-machine-learning-deep-learning.html>>. Nenhuma citação no texto.

RUSSELL, S.; MOSKOWITZ, I. S.; JALAIAN, B. Chapter 4 - context: Separating the forest and the trees—wavelet contextual conditioning for ai. In: LAWLESS, W. F.; MITTU, R.; SOFGE, D. A. (Ed.). *Human-Machine Shared Contexts*. Academic Press, 2020. p. 67–91. ISBN 978-0-12-820543-3. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780128205433000043>>. Nenhuma citação no texto.

SHARMA, S. Activation functions in neural networks. *towards data science*, v. 6, 2017. Nenhuma citação no texto.

WALCZAK, S.; CERPA, N. Artificial neural networks. In: MEYERS, R. A. (Ed.). *Encyclopedia of Physical Science and Technology (Third Edition)*. Third edition. New York: Academic Press, 2003. p. 631–645. ISBN 978-0-12-227410-7. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B0122274105008371>>. Nenhuma citação no texto.

WOOLF, B. P. Chapter 7 - machine learning. In: WOOLF, B. P. (Ed.). *Building Intelligent Interactive Tutors*. San Francisco: Morgan Kaufmann, 2009. p. 221–297. ISBN 978-0-12-373594-2. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780123735942000071>>. Nenhuma citação no texto.

WU, W.-Y.; WANG, M.-J. J.; LIU, C.-M. Automated inspection of printed circuit boards through machine vision. *Computers in Industry*, v. 28, n. 2, p. 103–111, 1996. ISSN 0166-3615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0166361595000631>>. Nenhuma citação no texto.

ZUMBAHLEN, H. Chapter 12 - printed circuit-board design issues. In: ZUMBAHLEN, H. (Ed.). *Linear Circuit Design Handbook*. Burlington: Newnes, 2008. p. 821–895. ISBN 978-0-7506-8703-4. Disponível em: <<https://www.sciencedirect.com/science/article/pii/B9780750687034000122>>. Nenhuma citação no texto.

## APÊNDICES

## **APÊNDICE A – NOTAÇÕES DE REDES NEURAIS**

## Standard notations for Deep Learning

This document has the purpose of discussing a new standard for deep learning mathematical notations.

### 1 Neural Networks Notations.

#### General comments:

- superscript (i) will denote the  $i^{th}$  training example while superscript [l] will denote the  $l^{th}$  layer

#### Sizes:

- $m$  : number of examples in the dataset
- $n_x$  : input size
- $n_y$  : output size (or number of classes)
- $n_h^{[l]}$  : number of hidden units of the  $l^{th}$  layer

In a for loop, it is possible to denote  $n_x = n_h^{[0]}$  and  $n_y = n_h^{[\text{number of layers} + 1]}$ .

- $L$  : number of layers in the network.

#### Objects:

- $X \in \mathbb{R}^{n_x \times m}$  is the input matrix
- $x^{(i)} \in \mathbb{R}^{n_x}$  is the  $i^{th}$  example represented as a column vector

- $Y \in \mathbb{R}^{n_y \times m}$  is the label matrix

- $y^{(i)} \in \mathbb{R}^{n_y}$  is the output label for the  $i^{th}$  example

- $W^{[l]} \in \mathbb{R}^{\text{number of units in next layer} \times \text{number of units in the previous layer}}$  is the weight matrix, superscript [l] indicates the layer

- $b^{[l]} \in \mathbb{R}^{\text{number of units in next layer}}$  is the bias vector in the  $l^{th}$  layer

- $\hat{y} \in \mathbb{R}^{n_y}$  is the predicted output vector. It can also be denoted  $a^{[L]}$  where  $L$  is the number of layers in the network.

#### Common forward propagation equation examples:

- $a = g^{[l]}(W_x x^{(i)} + b_1) = g^{[l]}(z_1)$  where  $g^{[l]}$  denotes the  $l^{th}$  layer activation function

$$\hat{y}^{(i)} = \text{softmax}(W_h h + b_2)$$

- General Activation Formula:  $a_j^{[l]} = g^{[l]}(\sum_k w_{jk}^{[l]} a_k^{[l-1]} + b_j^{[l]}) = g^{[l]}(z_j^{[l]})$

- $J(x, W, b, y)$  or  $J(\hat{y}, y)$  denote the cost function.

#### Examples of cost function:

- $J_{CE}(\hat{y}, y) = -\sum_{i=0}^m y^{(i)} \log \hat{y}^{(i)}$

- $J_1(\hat{y}, y) = \sum_{i=0}^m |y^{(i)} - \hat{y}^{(i)}|$