UNIVERSIDADE FEEVALE

FERNANDO BRUNNER

**APLICAÇÃO DO APRENDIZADO DE MÁQUINA NA SAÚDE. DESENVOLVIMENTO DE UM MODELO PREDITIVO PARA O DIAGNÓSTICO DO CÂNCER DE MAMA**

Novo Hamburgo

2022

FERNANDO BRUNNER

**APLICAÇÃO DO APRENDIZADO DE MÁQUINA NA SAÚDE. DESENVOLVIMENTO DE UM MODELO PREDITIVO PARA O DIAGNÓSTICO DO CÂNCER DE MAMA**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do grau de Bacharel em Ciências da computação pela Universidade Feevale

Orientador: Ricardo Ferreira de Oliveira

Novo Hamburgo

2022

FERNANDO BRUNNER

**APLICAÇÃO DO APRENDIZADO DE MÁQUINA NA SAÚDE. DESENVOLVIMENTO DE UM MODELO PREDITIVO PARA O DIAGNÓSTICO DO CÂNCER DE MAMA**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do grau de Bacharel em Ciências da computação pela Universidade Feevale

APROVADO EM: \_\_ / \_\_ / \_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ricardo Ferreira de Oliveira

Orientador - Feevale

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

GABRIEL DA SILVA SIMÕES

Examinador interno – Feevale

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

MARTA ROSECLER BEZ

Examinador interno – Feevale

Novo Hamburgo

2022

**RESUMO**

O câncer de mama, hoje, é um dos tipos de cânceres mais comuns que existem. Este câncer mata todos os anos milhares de pessoas. Este tipo de câncer acomete principalmente mulheres. Se descoberto cedo, o paciente tem grandes chances de sobreviver à doença. Porém, nos dias atuais, mesmo com o avanço tecnológico, muitos casos ainda têm o diagnóstico tardio, aumentando assim, a mortalidade em função da doença. Falhas humanas, com profissionais mal preparados e falta de infraestrutura são algumas das causas que dificultam, principalmente para populações mais pobres, o diagnóstico da doença. Nos últimos anos, o ramo do aprendizado de máquina ganhou muita popularidade na aplicação à saúde, por ser uma solução viável para esses problemas. Capaz de analisar grandes volumes de dados, fazer predições e tomar decisões com uma baixa taxa de erro, com uso de aprendizagem supervisiona, *computer vision* e redes neurais, *machine learning* é usado para detectar e diagnosticar diversos tipos de cânceres, incluindo o câncer de mama. O objetivo deste trabalho é criar modelos e comparar a performance de cada classificador na tarefa de detectar o câncer de mama em imagens obtidas através do exame de Histopatologia e analisar a viabilidade da utilização do uso de modelos *machine learning* para tal tarefa. Os modelos utilizaram técnicas de aprendizado de máquina e aprendizado supervisionado. Com isso, o melhor resultado encontrado foi de 85%, e de 84%, nos dados de teste. Com isso, concluindo a viabilidade do uso de modelos de *machine learning* para o diagnóstico do câncer de mama.

Palavras-chave: Inteligência Artificial. *Machine Learning*. Aprendizado Supervisionado. Câncer de mama

**ABSTRACT**

Breast cancer today is one of the most common types of cancer that exist. This cancer kills thousands of people every year. This type of cancer mainly affects women. If discovered early, the patient has a high chance of surviving the disease. However, nowadays, even with technological advances, many cases still have a late diagnosis, thus increasing the mortality of the disease. Human failures, with poorly prepared professionals and lack of infrastructure are some of the causes that make it difficult, especially for poorer populations, to diagnose the disease. In recent years, the field of machine learning has gained a lot of popularity in the healthcare application, as it is a viable solution to these problems. Capable of analyzing large volumes of data, making predictions and making decisions with a low error rate, using supervisory learning, computer vision and neural networks, machine learning is used to detect and diagnose many types of cancers, including breast cancer. . The objective of this work is to create models and compare the performance of each classifier in the task of detecting breast cancer in images obtained through the Histopathology exam and to analyze the feasibility of using machine learning models for this task. The models used machine learning and supervised learning techniques. Thus, the best result found was 85%, and 84%, in the test data. Thus, concluding the feasibility of using machine learning models for the diagnosis of breast cancer.

Keywords: Artificial Intelligence. Machine Learning. Supervised Learning.

**LISTA DE FIGURAS**

Figura 1 – Carcinoma Ductal in Situ 19

Figura 2 – Diferença entre DCSI vs IDC 20

Figura 3 – Câncer de mama inflamatório 22

Figura 4 – Câncer de mama metastático 23

Figura 5 – Estágio 0 DCIS e LCIS 26

Figura 6 – *Input Command* vs *Input Data* 38

Figura 7 – *Computer science fields* 39

Figura 8 – Modelo tradicional 40

Figura 9 – Modelo *machine learning* 40

Figura 10 – Exemplo de modelo 42

Figura 11 – Exemplo de classificação binária 44

Figura 12 – Exemplo de regressão 45

Figura 13 – Exemplo de dados clusterizados pelo algoritmo *K-means*. 47

Figura 14 – Inteligência artificial, *Machine learning* e *Deep Learning* 48

Figura 15 – Estrutura de um algoritmo de *Deep Learning* 49

Figura 16 – Exemplo da estrutura básica de um neuron 51

Figura 17 – Estrutura do *Perceptron*  55

Figura 18 – Neurônio biológico 56

Figura 19 – Neurônio biológico vs Artificial 57

Figura 20 – Representação de uma *CNN* e suas *layers* 58

Figura 21 – Exemplo de *kernel* aplicando filtro de Sobel 60

Figura 22 – Técnica do *Zero-padding* 60

Figura 23 – Diagrama de atividades do projeto 63

Figura 24 – 3 imagens da classe 1 escolhidas aleatoriamente 64

Figura 25 – 3 imagens da classe 0 escolhidas aleatoriamente 64

Figura 26 – Resultados dos testes para definir o *batch size* 66

Figura 27 – Tensor, *Array* e *Matriz* 69

Figura 28 – *Shape* e *ranking* de uma imagem do *dataset* de treinamento 70

Figura 29 – Exemplo de modelo utilizando keras integrado com *Tensorflow* 71

Figura 30 – Exemplo de *median filter* usado no *dataset* do projeto 72

Figura 31 – Exemplo do *Gaussian filter* aplicado no *dataset* 72

Figura 32 – *N-dimentional Array* normalizado 73

Figura 33 – Exemplo de *data augmentation* 74

Figura 34 – Exemplo de um modelo *Fully connected layer* 76

Figure 35 – Exemplo de *pooling operation* 77

Figura 36 – Exemplo de um modelo utilizando uma *Flatten layer* 78

Figura 37 – *Sigmoid Function* 79

Figura 38 – *Relu Function* 80

Figura 39 – Criação dos tensores de treinamento e testes com quantidade de amostras de cada *dataset*. 84

Figura 40 – Criação do primeiro modelo proposto 84

Figura 41 – *Plot* do modelo 85

Figura 42 – Comparação dos *models* com ajustes 86

Figura 43 – Criação do modelo com base a arquitetura *LeNet* 87

Figura 44 – *Plot* do modelo *LeNet* 88

Figura 45 – Função *create\_model* 89

Figura 46 – Criação do modelo *ResNet* 90

Figura 47 – Criação do modelo com a função *create\_model* 90

Figura 48 – Modelo *ANN 91*

Figura 49 – *Plot* do modelo *Non CNN* 92

Figura 50 – Aplicação do filtro e criação do *dataset* de treino 93

Figura 51 – Chamada do método *fit()* 94

Figura 52 – Comparação de *accuracy* durante o treinamento 96

Figura 53 – Comparação da generalização do aprendizado 97

**LISTA DE TABELAS**

Tabela 1 – Estimativa para o número de novos casos de câncer em mulheres para o ano de 2020 16

Tabela 2 – Fatores que aumentam o risco para o desenvolvimento do câncer de mama 17

**LISTA DE QUADROS**

Quadro 1 – Configurações de *hardware* e *software* disponíveis pelo *Colab* 67

Quadro 2 – Primeira quadro com apresentação de resultados 95

Quadro 3 – Segundo quadro com apresentação de resultados 66

**LISTA DE SIGLAS**

ML *Machine learning*

IDC *Invasive ductal carcinoma*

DCSI *Ductal carcinoma in situ*

TNM *Capability Maturity Model Integration Development*

AJCC *American Joint Committee on Cancer*

UICC *União Internacional de Controle do Câncer*

LCIS *Lobular Carcinoma in Situ*

CNN *Convolutional neural network*

ANN *Artificial neuron network*

**SUMÁRIO**

[**1** **INTRODUÇÃO**](#_gjdgxs) **12**

[1.1 OBJETIVOS](#_30j0zll) 13

[1.2](#_1y810tw) METODOLOGIA**13**

[1.3](#_1fob9te) [E](#_1y810tw)STRUTURA DO TRABALHO14

**2 CÂNCER DE MAMA****14**

[2.1 TUMOR BENIGNO](#_1ci93xb) 17

2.2 TIPOS DE CÂNCER DE MAMA18

**2.2.1 CARCINOMA DUCTAL IN SITU 19**

**2.2.2 CARCINOMA DUCTAL INVASIVO 20**

**2.2.3 CÂNCER DE MAMA TRIPLO-NEGATIVO 21**

**2.2.4 CÂNCER DE MAMA INFLAMATÓRIO 21**

**2.2.4 CÂNCER DE MAMA METASTÁTICO 23**

2.3 ESTÁGIOS DO CÂNCER DE MAMA 24

**2.3.1 ESTÁGIOS 0 E 1 25**

**2.3.2 ESTÁGIO 2 26**

**2.3.3 ESTÁGIO 3 27**

**2.3.4 ESTÁGIO 4 27**

2.4 MÉTODOS DE DIAGNÓSTICO 27

2.5 DETECÇÃO PRECOCE 31

2.6 TRATAMENTO 33

**3 *MACHINE LEARNING* 37**

3.1 APRENDIZADO SUPERVISIONADO 42

3.2 APRENDIZADO NÃO SUPERVISIONADO 46

3.3 *DEEP LEARNING* 48

**4 *ARTIFICIAL NEURAL NETWORKS* 51**

4.1 *ANN* ALGORITMOS 53

4.2 HISTÓRIA DAS *ARTIFICIAL NEURAL NETWORKS* 53

4.3 NEURÔNIOS BIOLÓGICOS VS ARTIFICIAIS 55

**4.3.1 NEURÔNIOS BIOLÓGICOS 56**

**4.3.2 NEURÔNIOS ARTIFICIAIS 57**

**5 *CONVOLUTIONAL NEURAL NETWORKS* 58**

5.1 *PADDING* 60

5.2 *KERNEL SIZE* 61

5.3 *STRIDE* 61

**6 DESENVOLVIMENTO DO PROJETO 63**

6.1 *DATASET* DE IMAGENS 63

6.2 *BATCH SIZE* 65

6.3 AMBIENTE DE DESENVOLVIMENTO 67

**6.3.1 *TENSORFLOW* 68**

**6.3.2 *KERAS* 70**

6.4 PRÉ-PROCESSAMENTO 71

**6.4.1 *MEDIAN FILTER* 71**

**6.4.2 *GAUSSIAN FILTER* 72**

**6.4.3 NORMALIZAÇÃO 73**

**6.4.4 *DATA AUGMENTATION*  74**

6.5 ESTRUTURA 75

**6.5.1 *SEQUENTIAL API* 75**

**6.5.2 *LAYERS* 75**

**6.5.2.1 *DENSE LAYER* 76**

**6.5.2.2 *CONVOLUTIONAL LAYER* 76**

**6.5.2.3 *POOLING LAYER* 77**

**6.5.2.4 *FLATTEN LAYER* 77**

**6.5.3 *ACTIVATION FUNCTION* 78**

**6.5.3.1 *SIGMOID FUNCTION* 79**

**6.5.3.2 *RELU FUNCTION* 80**

6.6 *COMPILE* 81

6.7 CONSTRUÇÃO DOS MODELOS 84

**6.7.1 *LETNET MODEL* 87**

**6.7.2 *RESNET MODEL* 88**

**6.7.3 *EFFICIENTNET MODEL* 90**

**6.7.4 *NON CNN MODEL* 91**

**7 TESTES E RESULTADOS 93**

**8 CONCLUSÃO 98**

1 INTRODUÇÃO

Segundo o INCA (Instituto Nacional de Câncer), entre todas as doenças, o câncer é um problema de grande relevância para a saúde pública, sendo a segunda maior causa de mortes no ano de 2018. Foram mais de 9 milhões de mortes em todo o mundo. Países em desenvolvimento como o Brasil, vêm sofrendo mais com a doença, tendo um aumento na taxa de mortalidade de alguns tipos de cânceres. O Instituto Nacional de Câncer informa que cerca de 80% a 85% dos casos estão relacionados com fatores ambientais e hábitos de consumo. Dentre os tipos de câncer que mais matam no Brasil, está o câncer de mama. O levantamento feito pelo INCA em 2020, aponta para mais de 60 mil novos casos e mais de 18 mil mortes apenas no ano de 2019. O câncer de mama tem múltiplos fatores de riscos, como ambientais, comportamentais, de história reprodutiva, hormonais, genéticos e hereditários. A idade é um dos principais fatores de risco para a doença.

Em seu livro, Khalid Shaikh (2021) fala que aplicações de aprendizado de máquina vêm sendo desenvolvidas e aplicadas na prática de várias áreas da saúde, como método de diagnóstico, protocolo de tratamento, desenvolvimento de drogas e monitoramento de pacientes. Tom Mitchell (1997) descreve *machine learning* como um campo multidisciplinar, resultando da mistura de Inteligência artificial, probabilidade e estatística, *computational complexity* *theory*, *control theory*, information theory, filosofia, psicologia, entre outros. Mitchell também conceitua *machine learning* como sendo algoritmos que são capazes de aprender por experiência. A inteligência artificial vem sendo usada na saúde há algumas décadas. Nos anos 70, por exemplo, um sistema de inteligência artificial foi criado para o tratamento de bactérias, um sistema básico de *input* e *output* com cerca de 600 regras criado na Universidade de Stanford. Desde então, com o aumento tecnológico, grandes avanços foram feitos na área. Com isso, sistemas mais complexos foram desenvolvidos e inúmeros benefícios foram alcançados para o diagnóstico e tratamento de diversas doenças.

A detecção e diagnóstico do câncer de mama também evoluíram muito nas últimas décadas. Ferramentas como mamografia e ultrassom são utilizadas no mundo todo para a detecção de tumores. A triagem dessas imagens leva ao uso de algoritmos de *machine learning* para o diagnóstico do câncer. Segundo Khalid Shaikh (2021) conta em seu livro, muitos problemas na saúde foram efetivamente resolvidos com o uso de processamento digital de imagens. Shaikh descreve que estas tecnologias melhoram muito a qualidade das imagens, facilitando a compreensão e interpretação de humanos e também de algoritmos de *machine learning*, diminuindo, assim, a taxa de erro no diagnóstico do câncer de mama. Possibilita, também, a detecção da doença em diferentes estágios. Com o uso de ML na saúde, temos o benefício de diminuição de custos de operação, diminuição de falha humana, otimização de operações e aceleração no diagnóstico.

Nessa perspectiva, diante dos problemas e dificuldades causadas pela doença e dos benefícios do uso de *machine learning* na área da saúde, observou-se a necessidade de trabalhos na área e este trabalho tem a finalidade de desenvolver um modelo preditivo para classificação e detecção de tumores de mama.

1.1 OBJETIVOS

O objetivo geral deste trabalho é criar um modelo preditivo para o diagnóstico de tumores de mama, utilizando *python*, a biblioteca *Tensorflow* e o *Google Collaboratory* para o desenvolvimento e para o treinamento do modelo utilizando uma base com imagens obtidas através do exame de Histopatologia. O modelo utilizará técnicas de aprendizado de máquina e aprendizado supervisionado.

‘ Para tanto, foram definidos os seguintes objetivos específicos: Realizar pesquisa bibliográfica sobre *machine Learning*; Realizar pesquisa bibliográfica sobre *Machine Learning* aplicado à saúde; Desenvolver os modelos preditivos em *python*; Treinar os modelos utilizando a base de dados; Compara o desempenho dos modelos; Validação do modelo preditivo;

**1.2 METODOLOGIA**

Baseando-se em Prodanov e Freitas (2013) que descreveram os conceitos de metodologia científica, o trabalho desenvolvido tem por característica natureza de pesquisa aplicada, levando em consideração que será desenvolvido um modelo preditivo em *python*, treinado em cima de dados de um dataset, para o diagnóstico de câncer de mama. Sendo pesquisa aplicada definida como "objetivo de gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos''. Envolve verdades e interesses locais" (PRODANOV E FREITAS, 2013)

Para fundamentar os conhecimentos teóricos e servir de base para a construção do modelo preditivo, será realizada a pesquisa bibliográfica. Também serão feitas pesquisas de aprendizado de máquina e aprendizado supervisionado para o serem usados nos modelos e por fim, a etapa de desenvolvimento dos classificadores, treinamento e comparação de resultados.

## **1.3 ESTRUTURA DO TRABALHO**

No primeiro capítulo são descritos o câncer e suas manifestações, aspectos dos tumores malignos versus tumores benignos, apresentação de métodos de diagnóstico de câncer de mama, descrição da detecção precoce e os métodos de tratamento dos tumores.

No segundo capítulo é feita a conceituação de *machine learning*, assim como métodos de aprendizado supervisionado e aprendizado não supervisionado, outros métodos de treinamento, como *deep learning*, será dada uma visão geral de redes neurais e *computer vision*.

No terceiro capítulo falaremos sobre *Deep Learning* e processamento digital de imagens para diagnóstico de câncer de mama, abordagens de *deep learning* para o diagnóstico de câncer, Detecção e diagnóstico, Modelos para classificação de câncer de mama e Atributos comumente usados em sistemas de *machine learning* para detecção de câncer,

**2 CÂNCER DE MAMA**

Segundo o INCA (2020), o câncer engloba mais de 100 tipos de doenças. Elas são caracterizadas por um crescimento anormal de células que afetam tecidos e órgãos. Na mama o câncer é chamado carcinoma quando se inicia em tecidos como os da pele ou mucosas. É chamado de sarcoma quando afeta tecidos conjuntivos. (INCA, 2002)

Essas células formam tumores através da acumulação de células que se multiplicam de maneira descontrolada. Algumas células têm a capacidade de se desprender do tumor e invadir outros tecidos e até mesmo órgãos distantes, caracterizando, assim, a metástase (INCA, 2002). A velocidade que as células se multiplicam e atacam tecidos e órgãos varia de acordo com diferentes tipos de cânceres (INCA, 2020).

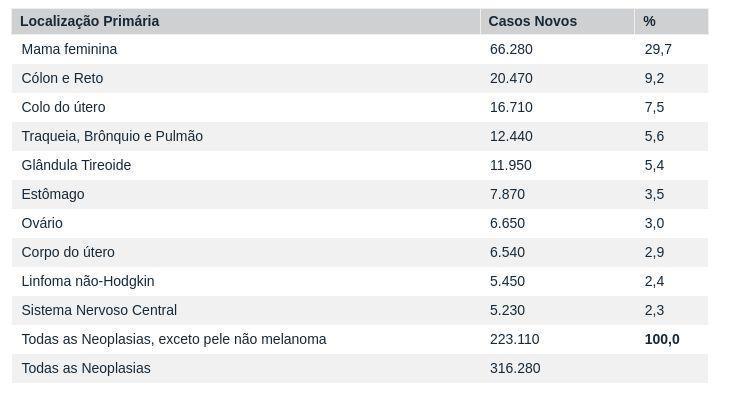
Este processo de formação do câncer é chamado de carcinogênese, e em geral, é um processo bastante lento e chega a levar anos até que um tumor se desenvolva (INCA, 2002). Para que isto aconteça, a formação do câncer passa por diferentes etapas: A etapa de iniciação, que consiste na etapa onde os genes sofrem com os fatores cancerígenos; A etapa de promoção, que acontece quando as células alteradas sofrem o efeito dos agentes cancerígenos; A etapa da progressão, que acontece quando as células cancerígenas se multiplicam descontroladamente. (INCA, 2019)

O câncer se transformou no maior problema de saúde pública mundial, chegando a ser uma das quatro principais causas de morte em pessoas com menos de 70 anos de idade (ESTIMATIVA INCA, 2020). Somente no Brasil, a expectativa é para 625 mil casos novos de câncer por ano (450 mil, excluindo os casos de câncer de pele não melanoma) (ESTIMATIVA INCA, 2020).

A última estimativa mundial demonstrou que apenas no ano de 2018, no mundo todo, foram 18 milhões de novos casos de câncer. O número de óbitos pela doença alcançou o número de 9.6 milhões de pessoas. O câncer de mama foi o câncer com maior incidência entre as mulheres, atingindo 24.2% dos casos em todo o mundo (ESTIMATIVA INCA, 2020).

No Brasil, a incidência varia de acordo com a região. A estimativa feita pelo INCA (2020) mostra que as incidências de câncer no país se concentram na região Sudeste, com cerca de 60%. Logo atrás está a região Nordeste com 27.8% e a região Sul com 23% dos casos. Junto com o câncer de próstata, o câncer de mama é predominante nas regiões Sul e Sudeste. Nas Regiões Norte e Nordeste o câncer de mama figura entre os principais que afetam a população feminina (ESTIMATIVA INCA, 2020).

**Tabela 1 - Estimativa para o número de novos casos de câncer em mulheres para o ano de 2020.**

****Fonte: MS / INCA / Estimativa de Câncer no Brasil, 2020 e MS / INCA / Coordenação de Prevenção e Vigilância / Divisão de Vigilância e Análise de Situação

O câncer de mama, como é conhecido o tumor maligno desenvolvido na região dos seios, pode acometer tanto homens, como mulheres. Apesar de que a incidência de 99% dos casos de câncer de mama ocorre em mulheres (RAMANI et al., 2017).

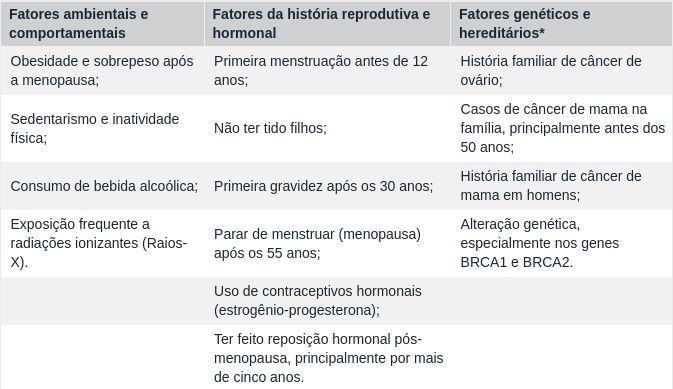
É um conjunto heterogêneo de doenças, e têm comportamentos e desenvolvimentos diferentes. Alguns tipos apresentam um desenvolvimento mais acelerado, enquanto outros se desenvolvem mais lentamente. (INCA, 2021)

Os tecidos mamários, em mulheres, antes da puberdade, são constituídos principalmente por tecido adiposo. Depois da puberdade, os seios desenvolvem glândulas, que podem secretar leite. Os seios de mulheres adultas são constituídos por gordura, tecidos conjuntivos e glândulas (RAMANI *et al*., 2017).

O câncer de mama pode ser classificado em dois grupos: Carcinoma Ductal, o mais comum, segundo o INCA; Carcinoma Lobular, que acomete menos pessoas. O câncer de mama também é categorizado em 3 tipos: invasivo, não invasivo e metastático (RAMANI *et al*., 2017).

O Instituto Nacional do Câncer (2021) informa que nas fases iniciais o câncer se manifesta através dos seguintes sintomas: Nódulo, que é a sua principal manifestação e se apresenta em 90% dos casos em que é percebido pela mulher; Pele da mama avermelhada; Alterações no bico do peito; Pequenos nódulos nas axilas ou no pescoço; Saída espontânea de líquido anormal pelos mamilos. A verificação desses sintomas é parte essencial para a detecção precoce da doença.

O câncer de mama tem múltiplas causas, sendo a idade um dos maiores fatores de risco para a doença.

**Tabela 2 - Fatores que aumentam o risco para o desenvolvimento do câncer de mama**

Fonte: MS / INCA

## **2.1 TUMOR BENIGNO**

Segundo o INCA, nem todo tumor é classificado como câncer. Um tumor é definido como o aumento de alguma parte do corpo. A neoplasia, que é o como é chamada o aumento do crescimento das células, pode ser benigna ou maligna. Quando neoplasias benignas ocorrem, não acontece a metástase. Neoplasias também se desenvolvem de maneira lenta e limitada. O lipoma, composto por células de gordura, é um exemplo de tumor benigno (INCA, 2021).

Essas alterações neoplásicas representam um rápido aumento celular, porém, sem anomalias. Isso significa que possuem as mesmas características de células saudáveis da mama. São exemplos de tumores benignos: Fibroadenoma, comum em mulheres jovens; Fibroadenolipoma, é um tumor de consistência macia; Adenoma, tumor benigno representado pela mudança das características histológicas do tecido; Lesão esclerosante radial, também chamado de cicatriz radial. (INCA, 2019)

**2.2 TIPOS DE CÂNCER DE MAMA**

Segundo o Oncoguia (2020), o câncer de mama pode ser classificado e descrito de múltiplas formas diferentes, de acordo com as células mamárias afetadas e o modo como ele se comporta. Carcinoma é o tipo mais comum de câncer de mama que acomete homens e mulheres. O câncer é classificado carcinoma quando os tumores começam nas células epiteliais que revestem órgãos e tecidos do corpo. Esse tipo de câncer também pode ser classificado como Adenocarcinoma, quando começa nas células de um ducto mamário ou nas glândulas produtoras de leite.

Outro modo de classificar o câncer de mama, é pelas áreas afetadas pelas células cancerígenas. Se ele disseminou ou não. Classificado como câncer de mama *in situ*, é o tipo de câncer que começa no ducto de leite e não se espalha para o restante do tecido mamário. Quando o câncer se espalha pelo resto do tecido mamário circulante, ele é denominado câncer de mama invasivo ou infiltrante (ONCOGUIA, 2020).

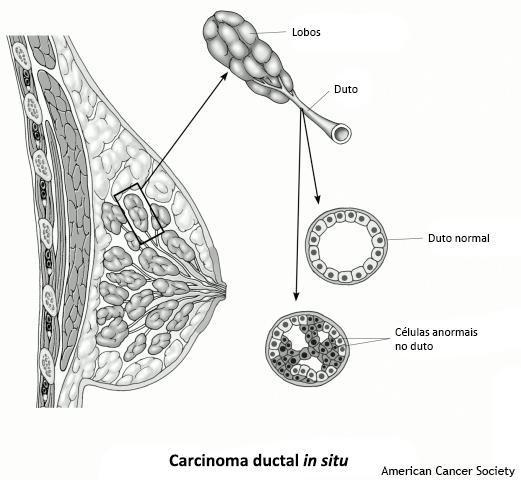
Os tipos mais comuns de câncer de mama invasivos são o carcinoma ductal invasivo e o carcinoma lobular invasivo. Hoje, o câncer de mama do tipo invasivo representa cerca de 80% de todos os casos de câncer de mama (ONCOGUIA, 2020).

O primeiro estágio em que o câncer pode ser classificado, é o carcinoma *in situ*, também conhecido como câncer não invasivo. Neste estágio, o câncer se concentra na camada de tecido do órgão em que se desenvolveu e não se espalha por todo o órgão. Se identificado e tratado durante este estágio, o câncer de mama tem grandes chances de ser curado. Ele é chamado de invasivo quando o câncer progride deste estágio para onde ele invade outras camadas de tecido do órgão de origem e ganha a capacidade de se espalhar para outras partes do corpo. Esse processo também é conhecido como metástase (INCA, 2020).

**2.2.1 CARCINOMA DUCTAL *IN SITU***

Também conhecido como carcinoma intraductal, o Carcinoma Ductal *in Situ* (Figura 1) é classificado como não invasivo ou câncer de mama pré-invasivo. Esse tipo de câncer ocorre quando as células que revestem os ductos são cancerígenas, mas não se espalharam para os tecidos mamários adjacentes. Cerca de 20% dos novos casos de câncer de mama são diagnosticados como Carcinoma Ductal *in Situ*. Quando diagnosticado durante esta etapa, o câncer de mama tem grandes chances de ser curado (ONCOGUIA, 2020). O ideal é ser tratado durante o estágio 0, caso contrário, o câncer pode evoluir e se espalhar para outros tecidos (SHAIKH *et al*., 2021).

**Figura 1: Carcinoma Ductal in Situ**

****

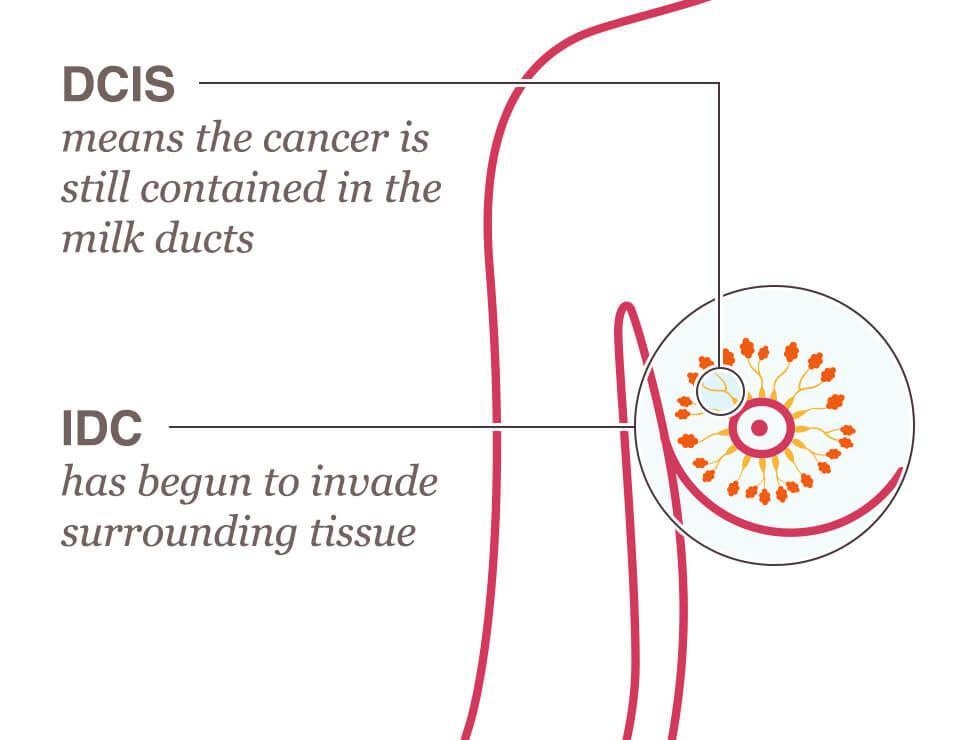
Fonte: Oncoguia, 2020

## Durante esta etapa, o Carcinoma Ductal *in Situ* não pode se espalhar para outros tecidos. Porém, com a possibilidade de se tornar um tipo invasivo, este tipo de câncer pode se espalhar para outras regiões do corpo. Ainda não existe uma forma de prever a evolução do Carcinoma Ductal *in Situ* para invasivo, evidenciando a importância do diagnóstico precoce e tratamento do câncer durante este estágio. Cirurgia conservadora da mama e mastectomia simples são as opções de tratamento para a grande maioria de casos de mulheres que sofrem desse tipo de câncer (ONCOGUIA, 2020).

**2.2.2 CARCINOMA DUCTAL INVASIVO**

Quando o câncer de mama evolui para os tecidos mamários adjacentes, ele é classificado como câncer de mama invasivo (Figura 2), enquanto o DCSI permanece nos ductos de leite e não invade outras áreas (National Breast Cancer, 2020). A grande maioria dos casos que acometem mulheres são classificados como câncer de mama invasivos. O carcinoma ductal invasivo está entre os dois tipos mais comuns de câncer de mama (ONCOGUIA, 2020).

**Figura 2: Diferença entre DCSI vs IDC**

****

Fonte**:** National Breast Cancer, 2020

Células anormais são formadas nos ductos de leite e se espalham para o resto do corpo, por isso esse tipo de câncer é chamado Carcinoma ductal invasivo. É o mais comum tipo de câncer de mama invasivo e ocorre cerca de 70-80 vezes a cada 100 casos. É o tipo mais comum de câncer que acomete homens (SHAIKH *et al*., 2021).

O tipo de tratamento adequado para este tipo de câncer irá depender do estágio em que o câncer foi detectado. Na maioria das vezes algum tipo de cirurgia será necessário. Dependendo do caso, tratamentos adicionais antes ou depois da cirurgia poderão ser recomendados (ONCOGUIA, 2020).

**2.2.3 CÂNCER DE MAMA TRIPLO-NEGATIVO**

Responsável por até 15% dos novos casos de câncer, o câncer de mama triplo-negativo acontece quando células cancerígenas não têm receptores de estrogênio ou progesterona e não produzem a proteína HER2. Esse tipo de câncer é mais comum em mulheres afro-brasileiras ou com menos de 40 anos (ONCOGUIA, 2020).

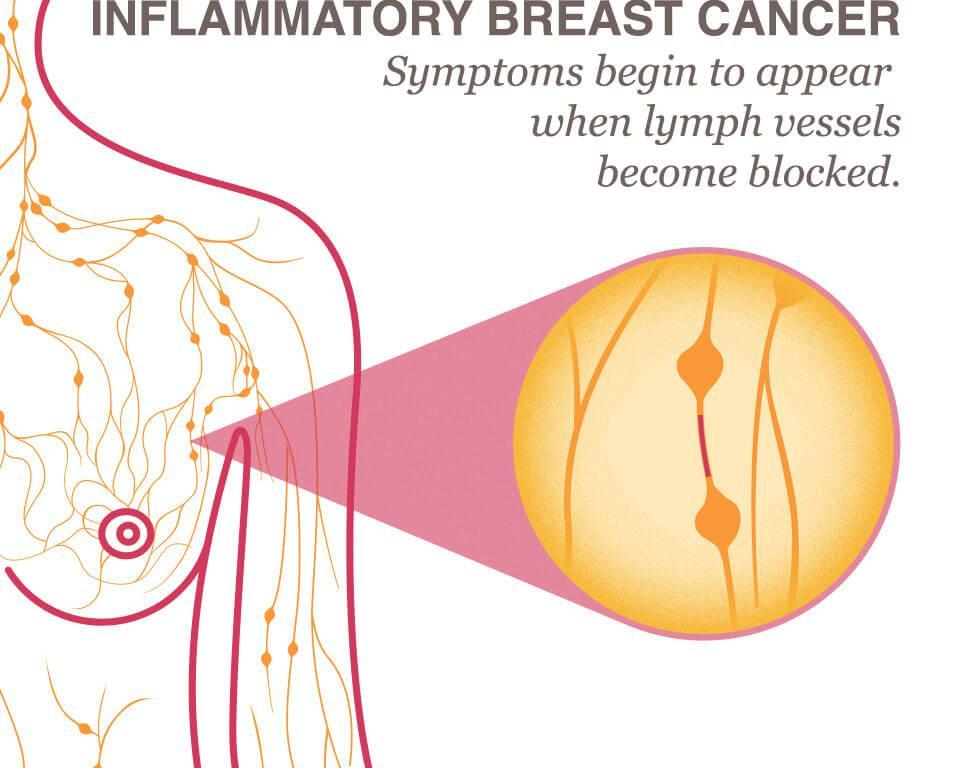
Por conta da ausência de receptores hormonais, tratamentos comuns e drogas que visam estrogênio ou progesterona, como a terapia hormonal, são ineficazes. Uma alternativa de tratamento ainda eficaz para este tipo de câncer é a quimioterapia. Em estágios iniciais do câncer de mama triplo-negativo, a quimioterapia se mostrou mais eficaz do que em outros tipos de câncer de mama (NATIONAL BREAST CANCER, 2020).

Além de difícil e de ter possibilidades limitadas de tratamento, o câncer de mama triplo-negativo é considerado um tipo muito agressivo de câncer, com grande possibilidade de metástase e de recorrência tumoral. O prognóstico irá depender do estágio em que o câncer foi diagnosticado. Novas drogas são testadas especificamente para o tratamento deste tipo de câncer (NATIONAL BREAST CANCER, 2020).

**2.2.4 CÂNCER DE MAMA INFLAMATÓRIO**

Câncer de mama inflamatório (Figura 3) é um tipo de câncer agressivo e de crescimento rápido. É um tipo mais de câncer e ocorre de 1% a 5% dos casos diagnosticados. É classificado como câncer de mama invasivo, mas difere de outros tipos no que se refere a sintomas, tratamento e prognóstico (ONCOGUIA, 2020). O câncer se infiltra na pele da mama e nos vasos linfáticos. Nem sempre apresenta caroços, o que pode dificultar o diagnóstico, mas quando bloqueia os vasos linfáticos, sintomas começam a surgir (NATIONAL BREAST CANCER, 2020).

**Figura 3: Câncer de mama inflamatório**

****

Fonte: National Breast Cancer, 2020

Estágios iniciais do IBC podem apresentar coceira e irritações na pele. A mama pode apresentar inchaço, vermelhidão e dilatação dos poros da pele da mama. Para diagnóstico em casos de suspeita de IBC, poderão ser utilizados exames de mamografia, ultrassom da mama e Ressonância magnética da mama. Para a confirmação do diagnóstico, sempre é necessária a realização da Biópsia (ONCOGUIA, 2020, NATIONAL BREAST CANCER, 2020).

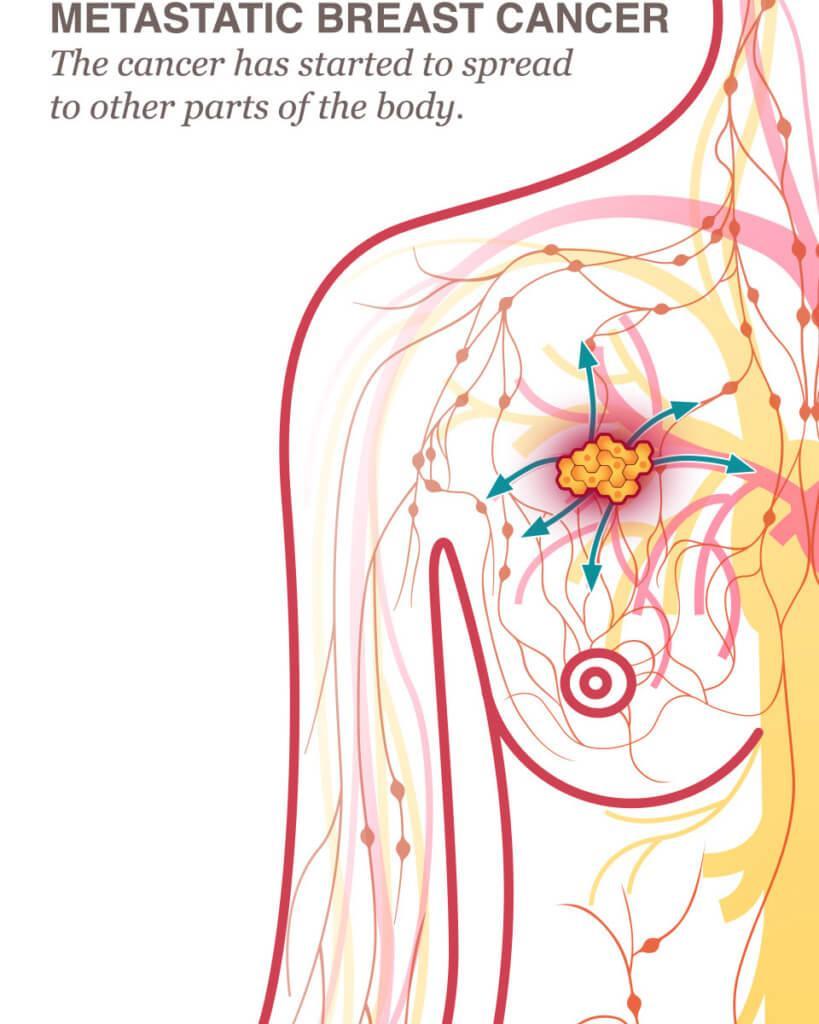
Por conta da alta taxa de crescimento e agressividade deste tipo de câncer, é provável que o IBC já tenha se espalhado no momento do diagnóstico. Também há uma provável chance de recidivar após o tratamento em relação a outros tipos de câncer. O prognóstico tende a ser pior quando comparado com outros tipos de câncer de mama(ONCOGUIA, 2020).

Para medir a probabilidade de sucesso para o tratamento, é utilizada a taxa de sobrevida em 5 anos. Ela se refere a porcentagem de pacientes que vivem por pelo menos 5 anos após o diagnóstico da doença. Em casos em que o tumor se disseminou para estruturas próximas, a taxa de sobrevida é de 52%. Quando o tumor se disseminou para outros órgãos a taxa de sobrevida cai para 18% (ONCOGUIA, 2020).

**2.2.4 CÂNCER DE MAMA METASTÁTICO**

Câncer de mama metastático (Figura 4) é um câncer em estágio 4. Acontece quando a doença se dissemina para outros locais além do local de origem. As células cancerígenas têm a capacidade de se desprender do tumor localizado na mama e entram na corrente sanguínea ou no sistema linfático. Desta forma, se espalham para outras regiões do corpo. Geralmente afetam os pulmões, fígado, ossos ou o cérebro (ONCOGUIA, 2020, NATIONAL BREAST CANCER, 2020).

**Figura 4: Câncer de mama metastático**

****

Fonte: National Breast Cancer, 2020

O câncer de mama pode levar vários passos até contaminar outras regiões do corpo. Quando uma célula saudável é contaminada, ela ganha a habilidade de replicar células anormais, essas células cancerígenas viajam através de vasos linfáticos e vasos sanguíneos, o sistema linfático carrega as células cancerígenas para outras regiões do corpo. Depois de espalhadas, as células cancerígenas começam a formar pequenos tumores (NATIONAL BREAST CANCER, 2020).

Os sintomas do câncer de mama avançado, como também é conhecido o MBC, podem variar e depende do quanto ele se espalhou e de quais as áreas que ele atingiu (SHAIKH *et al*., 2021).

**2.3 ESTÁGIOS DO CÂNCER DE MAMA**

Quando a paciente é diagnosticada com tumor maligno, é importante identificar o estágio da doença. Para isso, testes são feitos com o objetivo de descobrir se as células cancerígenas se espalharam para outros locais do corpo. Esse processo é conhecido como estadiamento e é parte fundamental para determinar qual o tratamento adequado para a paciente (SHAIKH *et al.*, 2021. National Breast Cancer, 2020).

Para determinar o estágio em que o câncer de mama se encontra, são analisados o tamanho do tumor, a localização em que o câncer se originou, se o câncer se espalhou pelo sistema linfático ou outras regiões do corpo, o grau do tumor ou se existe a presença de marcadores biológicos. Para isto, são combinados o sistema *TNM*, grau do tumor e o status dos marcadores biológicos (NATIONAL BREAST CANCER, 2020).

Existem dois tipos de estadiamentos para determinar o estágio do câncer. O primeiro é o estadiamento clínico, que é uma estimativa do estágio da doença que se baseia em resultados de exames físicos, exames de imagens e a biópsia do tumor. Dependendo do tipo de tumor, outros exames também podem ser necessários para o estadiamento clínico (ONCOGUIA, 2020).

O estadiamento cirúrgico, também conhecido como estadiamento patológico, pode ser feito depois do tratamento cirúrgico. Combina exames com observações feitas durante a cirurgia. A cirurgia não só é feita para retirar o tumor, mas também para retirar amostras do tecido cancerígeno e tentar identificar o estado do tumor. Isso com o objetivo de oferecer mais informações que devem ser utilizadas para prever a resposta do tratamento do tumor (ONCOGUIA, 2020).

Apesar de existir diferentes sistemas de estadiamento, o sistema que se mostra mais útil para a maioria dos casos, é o sistema *TNM*. O sistema *TNM* é utilizado pela *AJCC* e a *UNICC* como ferramenta para o estadiamento de diferentes casos de cânceres. O sistema *TNM* é atualizado no máximo a cada 8 anos, desta forma, se mantém atual com os avanços na nossa compreensão sobre a doença. Para cada tipo de câncer, são atribuídas letras e números que ajudam a descrever o tumor, linfonodos e metástases. O T é utilizado para tumores primários. O N para o câncer que se disseminou para linfonodos próximos e o M para Metástase (ONCOGUIA, 2020, NATIONAL BREAST CANCER, 2020).

O grau do tumor é importante porque ajuda a entender e prever as chances de o tumor crescer e se espalhar. Ele mede quão anormal as células cancerígenas aparecem no microscópio. Cânceres com mais alterações, que é denominado diferenciação, tendem a crescer e se espalhar com maior velocidade. Cânceres de baixo grau, têm a diferenciação muito semelhante a células saudáveis, e tendem a ter um crescimento lento. Cânceres de alta diferenciação, além de um crescimento acelerado, costumam ter um prognóstico pior. O tratamento também pode variar de acordo com o grau que o câncer se encontra (ONCOGUIA, 2020, NATIONAL BREAST CANCER, 2020).

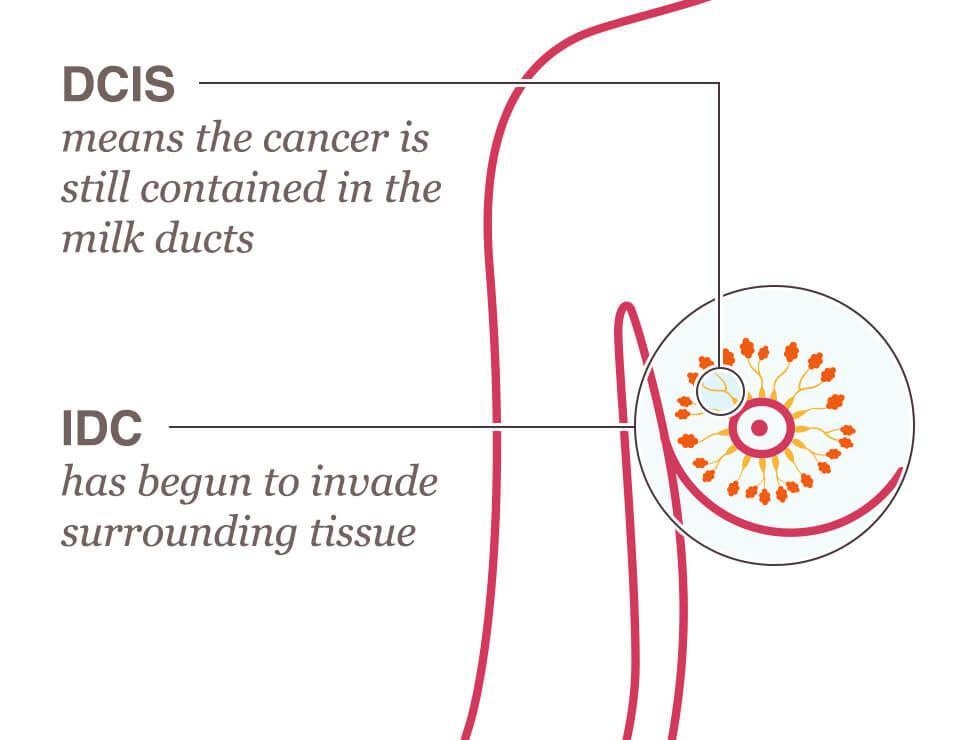
Células saudáveis e alguns tipos de cânceres de mama têm receptores que se ligam aos hormônios estrogênio e progesterona. Células de determinados tipos de cânceres de mama utilizam esses hormônios para crescer e se dividir. Para checar esses receptores, é utilizado o teste de marcadores biológicos, também conhecido como marcadores tumorais. Amostras do tecido do tumor são retiradas através de cirurgia ou biópsia e testadas em laboratório, com o objetivo de encontrar esses receptores hormonais. Outro tipo de receptor que os marcadores biológicos buscam encontrar, são os receptores HER2, também utilizados pelas células cancerígenas para crescer e dividir (NATIONAL BREAST CANCER, 2020).

**2.3.1 ESTÁGIOS 0 E 1**

O estágio serve para indicar tamanho e origem do câncer. Cânceres de estágio 0 recebem o nome de "carcinoma in situ". Carcinoma significa câncer e "*in situ*" significa "local de origem". Existem 3 possíveis tipos de cânceres "*in situ*" que se desenvolvem na região das mamas. Carcinoma Ductal *in situ*, carcinoma lobular *in situ* e Doença de Paget da Mama (NATIONAL BREAST CANCER, 2020).

O estágio 0 é classificado em dois tipos. Estágio 0 DCIS e Estágio 0 LCIS (Figura 5). O câncer de estágio 0 DCIS é um tipo não invasivo localizado nos ductos de leite da mama e não se espalhou. É um tipo de câncer em estado inicial e tem boas chances de tratamento quando diagnosticado cedo. O Estágio 0 LCIS geralmente não é considerado câncer e significa a existência de células anormais crescendo dentro dos lóbulos da mama (NATIONAL BREAST CANCER, 2020).

**Figura 5: Estágio 0 DCIS e LCIS**

****

Fonte: National Breast Cancer, 2020

O câncer de mama de estágio 1 é evidente, embora não tenha se espalhado para outras regiões do corpo. O câncer de mama de estágio 1 é categorizado em dois grupos. Grupo 1A e grupo 1B. Essa categorização é feita de acordo com o tamanho do tumor e com os linfonodos em que as células anormais se encontram.Tumores com até 2 centímetros de tamanho e que não se espalharam para os linfonodos são classificados no grupo 1A. Quando há a presença de células anormais nos linfonodos, ele é classificado com 1B. O câncer de mama de estágio 1 também é facilmente tratado (SHAIKH *et al*., 2021).

**2.3.2 ESTÁGIO 2**

Quando o tumor está em processo de crescimento mas ainda não se espalhou ou se espalhou apenas para linfonodos próximos, ele é classificado como câncer de mama de estágio 2. Tumores nesse estágio podem ser classificados de duas formas. Câncer de mama de estágio 2A ou Câncer de mama de estágio 2B. Para esta classificação são levados em consideração o tamanho do tumor e se ele se espalhou ou não para os linfonodos. Os cânceres nesse estágio normalmente são tratados primeiro com quimioterapia seguida de cirurgia e radioterapia (NATIONAL BREAST CANCER, 2020).

**2.3.3 ESTÁGIO 3**

Tumores de estágio 3 são aqueles que começaram a se espalhar, mas ainda não alcançaram órgãos distantes. Normalmente atingiram linfonodos e tecido muscular. Ainda nesse estágio avançado do câncer, existem possibilidades efetivas de tratamento do tumor. Para este estágio, existem 3 classificações, estágio 3A, estágio 3B e estágio 3C. Para a classificação do câncer de mama de estágio 3, são analisados o tamanho do tumor e o modo como ele se espalhou (NATIONAL BREAST CANCER, 2020).

**2.3.4 ESTÁGIO 4**

O câncer de mama estágio 4 é classificado dessa forma quando o tumor se espalhou e atingiu outras regiões do corpo, como pulmão, cérebro e fígado. Nesse estágio já não existe a possibilidade de cura,embora com o avanço tecnológico, tenham surgido possibilidades de tratamento para estender e melhorar a qualidade de vida da paciente (NATIONAL BREAST CANCER, 2020).

## **2.4 MÉTODOS DE DIAGNÓSTICO**

O controle do câncer de mama é uma das grandes prioridades da política de saúde do Brasil. Desde a criação e a divulgação do Controle do Câncer de Mama: Documento de Consenso, o governo vem trabalhando para fornecer acesso à educação e procedimentos para a detecção do câncer de mama em estágios iniciais, para assim possibilitar o tratamento e aumentar as chances de sobrevida da mulher (INCA, 2007).

Qualquer suspeita deve ser investigada para descobrir se é um câncer de mama. Além do exame clínico, podem ser usados outros métodos de investigação. Exames de imagem como mamografia, ultrassonografia ou ressonância magnética são exemplos de técnicas usadas para a detecção do câncer de mama. A confirmação do diagnóstico só se dá através da realização da biópsia (INCA, 2021).

Mamografia é o exame feito nos seios através de raio-x. A mamografia mostra tecidos adiposos, fibrosos e glandulares da mama e é capaz de mostrar lesões muito pequenas, em estado inicial. É muito usada para verificar sinais iniciais de câncer de mama em mulheres sem sintomas. O exame de mamografia é recomendado pelo menos uma vez a cada dois anos, em mulheres com mais de 50 anos de idade (RAMANI *et al*., 2017).

As imagens feitas pela mamografia são geradas por raio-x. Da mesma forma que imagens de raio-x tradicionais são feitas. Com os seios comprimidos, um feixe de raio-x reflete sobre eles e é registrado com um receptor de imagem. O risco de radiação é mínimo, principalmente se comparado com os benefícios da detecção precoce do câncer de mama. (INCA, 2019)

O MS recomenda que seja feita mamografia de rastreamento, que é o exame de rastreamento para mulheres sem sintomas, em mulheres de 50 a 69 anos a cada 2 anos. Nesta faixa de idade, a mamografia tem uma maior taxa de acerto. Antes desse período, existe um risco maior para falso-positivos e falso-negativos, pois as mamas são mais densas e isso afeta a sensibilidade da mamografia (INCA, 2019).

De acordo com o INCA, os resultados da mamografia são classificados por *Bi-rads* (*Breast Imaging Reporting and Data System*). Esta é uma padronização e utiliza categorias de 0 a 5 para classificação dos resultados e recomendações de conduta para cada uma das classificações.

* Bi-rad 0: Necessita estudo complementar;
* Bi-rad 1: Sem anormalidades detectadas;
* Bi-rad 2: Uma ou mais anormalidades detectadas, mas benignas;
* Bi-rad 3: Uma ou mais anormalidades detectadas, talvez benignas;
* Bi-rad 4: Uma ou mais anormalidades detectadas, provavelmente maligna;
* Bi-rad 5: Anormalidades detectadas, malignas

A ultrassonografia, também conhecida por ultrassom, desempenha um importante papel na detecção do câncer de mama. Sem contraindicações, pode ser usada como primeira etapa no diagnóstico de câncer de mama em jovens. Pode ser usada para verificar os tecidos da mama e até mesmo como guia para realização de biópsias (RAMANI *et al*., 2017).

Para a realização do exame, é utilizado um aparelho chamado transdutor. O aparelho é colocado sobre a mama do paciente para visualizar a parte interna do local desejado. Desta forma, ele é capaz de ajudar a diferenciar nódulos sólidos de cistos. O exame é possível pois ele emite ondas sonoras de alta frequência, causando a vibração dos tecidos. Através da vibração o aparelho capta as imagens e elas são apresentadas pelo monitor (IRPP, 2019).

O exame é não-invasivo para o paciente, e é realizado por um técnico ou médico que seleciona as imagens de acordo com o que for achado. O surgimento de bolsas de ar pode atrapalhar a realização do exame e comprometer os resultados, para evitar este problema, o profissional faz o uso de um gel à base de água. Geralmente o procedimento é rápido e indolor. (IRPP, 2019).

A sonografia desempenha um importante papel para o rastreamento e prevenção do câncer de mama. Faz parte de exames de rotina de mulheres em visitas ao ginecologista. Também pode ser utilizada para completar exames de mamografia, fazer a verificação de caroços palpáveis, pacientes com limitações de radiação e para pacientes que possuem histórico de câncer de mama na família, incluindo homens, apesar de raros os casos de câncer de mama em pessoas do sexo masculino. (IRPP, 2019).

A biópsia é, atualmente, a melhor forma para se ter um diagnóstico definitivo de tecidos anormais na região das mamas. Funciona em conjunto com outros métodos, retirando uma pequena quantidade de tecido da região anormal observada por ultrassom ou mamografia. Depois de removida, a biópsia é realizada por um patologista utilizando um microscópio, que determinará o tipo de câncer que a amostra corresponde (RAMANI *et al*., 2017. ONCOGUIA, 2014).

A biópsia é feita com o auxílio do ultrassom, para guiar o médico que está fazendo uso da agulha. Existem diversos tipos de biópsias e a utilização de cada tipo é definida de acordo com a situação específica. Para decidir qual procedimento usar, o médico responsável irá considerar o tipo de lesão, tamanho, localização, quantidade de tumores entre outras informações (RAMANI *et al*., 2017. ONCOGUIA, 2014).

FNA é um tipo de biópsia que consiste na remoção de uma amostra de um tecido mamário suspeito identificado por ultrassom. É utilizada uma agulha fina junto a uma seringa para a coleta da amostra e depois de pronto, o caroço identificado poderá diminuir um pouco. Para ser possível coletar a quantidade necessária para o exame, esse procedimento poderá ser realizado múltiplas vezes. Depois de coletada, a amostra é analisada em lâminas (RAMANI *et al*., 2017. ONCOGUIA, 2014).

CNB é um tipo de biópsia que consiste no uso de uma agulha maior em conjunto de uma pistola especial para coletar o tecido. Este tipo de biópsia apresenta mais detalhes sobre o tipo do câncer e até mesmo a sua taxa de crescimento do que a biópsia FNA. É executada em conjunto com o ultrassom, mamografia ou ressonância magnética. Se encontrado o tumor, ele será analisado por um patologista. Costuma ser o tipo de biópsia preferido para suspeitas de câncer de mama (RAMANI *et al.,* 2017. ONCOGUIA, 2014).

SB é o tipo de biópsia que consiste na remoção completa da área anormal ou nódulo e uma pequena área de tecido normal para exame. Sendo necessária apenas em casos raros, a SB dá informações mais detalhadas a respeito da área anormal do que a CNB. Porém, muito mais invasiva, retirando uma grande área do tecido mamário. Em caso da confirmação de câncer, uma cirurgia se torna necessária (RAMANI *et al.*, 2017. ONCOGUIA, 2014).

SLNB é um procedimento que consiste na retirada do nódulo localizado mais próximo ao nódulo suspeito. Ele é retirado para a verificação da presença de células cancerígenas. Geralmente esta biópsia é utilizada para verificar a existência de metástase. Se o exame for positivo, isso significa que o câncer atingiu outros nódulos ou até mesmo outros tecidos e órgãos (RAMANI *et al.*, 2017. ONCOGUIA, 2014).

Para decidir o melhor tratamento, é necessário conhecer o tipo de receptor hormonal das células cancerígenas retiradas para amostra durante a biópsia. Este teste define se as células do tumor são receptivas a hormônios ou não. Muitos tipos de câncer de mama têm receptores de estrogênio ou progesterona. Quando esses hormônios se ligam aos receptores do tumor, ele é alimentado, cresce e se multiplica. Por isso, é importante entender se o tumor tem receptores para evitar que ele seja alimentado e para o seu crescimento (RAMANI *et al.*, 2017. ONCOGUIA, 2017).

Níveis anormais de receptores de proteína HER2 podem acelerar o crescimento do câncer. Algumas mulheres têm tumores com níveis altos, e portanto, nesses casos o câncer é chamado de câncer de mama HER2+. Apesar de ter a tendência de crescer e se espalhar com mais facilidade e rapidez, esse tipo de câncer também costuma responder bem ao tratamento. O teste por receptores HER2 é feito através de biópsia. Testes adicionais são indicados em caso de positivo para ajudar no tratamento adequado (RAMANI *et al.*, 2017. ONCOGUIA, 2020).

Ki-67 é um bom marcador para taxar o avanço do câncer de mama. É uma substância liberada durante o crescimento e a divisão celular. Os cânceres que mais crescem, tendem a ter níveis de Ki-67 mais elevados. Como nota de corte, foi definido que tumores com níveis de Ki-67 abaixo de 10% como tumores pouco agressivos e tumores agressivos para aqueles com níveis acima de 30%. Os níveis de Ki-67 também podem ser usados para ajudar a definir o tratamento a ser tomado (RAMANI *et al.*, 2017. CANCER DE MAMA BRASIL, 2019).

*Pet scan* é uma poderosa ferramenta para diagnóstico por imagem. Ele é capaz de detalhar todo o corpo do paciente com precisão. É capaz de fornecer informações detalhadas sobre o quanto as células cancerígenas são ativas e informações metabólicas sobre os tecidos da região mamária. Normalmente este tipo de exame é solicitado pelo médico para detectar tumores ou acompanhar o desenvolvimento de um câncer. Também facilita muito identificar lesões causadas por tumores (RAMANI *et al.,* 2017).

*Cancer staging* é o processo de definir o estado atual do câncer e o quanto de câncer o paciente tem em seu corpo. Informações sobre o tamanho e como ele se espalha pelo corpo da pessoa. Esse procedimento serve para o médico saber a seriedade do caso, as chances de sobrevida do paciente e planejar o tratamento (RAMANI *et al.*, 2017. NATIONAL CANCER INSTITUTE, 2015).

Existem muitos modos de definir o estágio do câncer, como por exemplo exames de raio-x e testes de laboratório. O mais comum se chama TNM e é amplamente recomendado. Este exame pode ser utilizado para definir o estágio de uma grande variedade de cânceres e definir o tratamento adequado para cada um (RAMANI *et al.*, 2017).

**2.5 DETECÇÃO PRECOCE**

A detecção precoce busca identificar o câncer de mama em um estágio inicial, para assim buscar um melhor prognóstico e aumentar muito as chances de sobrevida da paciente. Países de baixa e média renda sofrem com o diagnóstico tardio da doença, o que dificulta em muito o prognóstico, aumenta a morbidade relacionada ao tratamento e piora a qualidade de vida do paciente. No contexto apresentado, estratégias de detecção precoce bem implementadas melhoram significativamente o estágio em que o câncer é detectado (INCA, 2015).

Nas últimas décadas, países de alta renda obtiveram sucesso no combate ao câncer de mama. Estratégias implementadas para o combate da doença resultaram em uma diminuição drástica na mortalidade do câncer de mama. O sucesso alcançado no controle da doença é atribuído a ações que possibilitam a detecção precoce e o tratamento do câncer em estágios iniciais (INCA, 2015).

A prevenção também desempenha um papel importante para a diminuição da mortalidade do câncer de mama. O controle de fatores de risco é fundamental para a prevenção da doença. Mas nem todos os fatores de risco são passíveis de controle, como por exemplo, os fatores hereditários. Riscos que vêm com a má alimentação, excesso de gordura corporal e inatividade física estão entre aqueles possíveis de prevenir. Para isso, é aconselhado a prática de atividades físicas, alimentação balanceada, controle do percentual de gordura e diminuição no consumo de bebidas alcoólicas (INCA, 2021).

No Brasil as diretrizes para combate do câncer de mama foram elaboradas pelo Ministério da Saúde com o auxílio do INCA. Tendo o objetivo de guiar as tomadas de decisões de gestores de saúde no trabalho de combate ao câncer de mama, como os profissionais da área de saúde na prática clínica (INCA, 2015)

São definidas duas estratégias para possibilitar a detecção precoce do câncer de mama. A primeira é o diagnóstico precoce. Trata-se de uma abordagem para reconhecimento do desenvolvimento de sintomas iniciais da doença em pessoas. A segunda é a estratégia do rastreamento. O rastreamento consiste em testes feitos em uma população sem sintomas aparentes a fim de identificar alterações e sintomas anormais e encaminhar estes pacientes para uma investigação mais detalhada (INCA, 2021).

No rastreamento do câncer de mama, o sexo e a faixa etária são os parâmetros que definem qual o grupo de risco. Segundo o INCA (2015), o risco de uma mulher de 74 anos desenvolver este tipo de câncer é de aproximadamente 4.60%. Em regiões com maior desenvolvimento este risco sobe para aproximadamente 7.90%. No Brasil o risco cumulativo é de cerca de 6.30%.

A mamografia é o exame utilizado no rastreamento do câncer de mama. Os melhores resultados na detecção precoce da doença são obtidos com o exame bienal em mulheres na faixa etária de 50 aos 69 anos de idade. A utilização da mamografia é motivo de debates e traz alguns riscos, como por exemplo, o falso-positivo, que causa ansiedade na paciente e causa um excesso de testes. Em alguns poucos casos, até mesmo exposição à radiação ionizante (INCA, 2021).

O rastreamento é classificado em dois tipos. O oportunístico e o organizado. No rastreamento organizado, o exame de mamografia é oferecido para mulheres que estão dentro do grupo de risco. Este estilo de rastreamento parece ser mais eficaz para o controle da doença, mostrando mais controle na qualidade do diagnóstico e monitoramento de todas as etapas do câncer. No rastreamento oportunístico, o exame é oferecido para mulheres que procuram uma unidade de saúde (INCA, 2021).

O rastreamento se mostra eficaz em países que o adotaram. Quando feito de forma adequada, com cobertura do grupo de risco e com qualidade nos exames, o impacto na diminuição do câncer de mama é positivo e justifica a aplicação do rastreamento. No Brasil, a mamografia é o único método de rastreamento que mostrou impacto significativo na diminuição da mortalidade do câncer de mama (INCA, 2021).

O INCA (2021) determina alguns pilares para a obtenção de êxito no rastreamento do câncer de mama. São eles: Informar e mobilizar a população e a sociedade civil organizada; Alcançar a meta de cobertura do grupo de risco; Garantir acesso a diagnóstico e tratamento oportuno; Garantir a qualidade das ações; Monitorar e gerenciar continuamente as ações.

Para tornar o diagnóstico precoce uma estratégia viável, a educação e a disseminação de informação são processos fundamentais. A população deve estar bem informada, ciente e alerta para os sintomas que indicam um caso suspeito de câncer. Não apenas a população,mas profissionais da saúde também precisam ficar atentos e capacitados para o surgimento desses sintomas. Outra parte fundamental para o diagnóstico precoce bem-sucedido é a capacitação de serviços de saúde em oferecer recursos para o diagnóstico com qualidade (INCA, 2015).

**2.6 TRATAMENTO**

Avanços significativos para o tratamento do câncer foram feitos nos últimos anos. Métodos de cirurgias menos invasivas e mutiladoras foram desenvolvidos. Outro importante passo foi a criação do individualismo do tratamento (INCA, 2021). Uma equipe multidisciplinar deve designar o tratamento para cada paciente. O tratamento é definido de acordo com as características de cada paciente. Fatores como idade, peso e altura, tipo e características do câncer, tamanho do tumor, características biológicas do câncer, receptores de hormônios, entre outros são levados em consideração na hora de definir o tratamento (RAMANI *et al*., 2017).

Quando o diagnóstico é feito de modo precoce, o tratamento tem maior chance de sucesso. Quando evidências de metástase, o principal objetivo do tratamento se transforma em melhorar a qualidade de vida do paciente e aumentar a sua sobrevida. Os tratamentos se dividem em dois tipos: Local e sistemático. No tratamento local, o objetivo é tratar o câncer localmente, sem que a terapia afete o resto do corpo. No tratamento sistemático, o paciente recebe medicamentos por via oral ou na corrente sanguínea e o tratamento ataca células cancerígenas em todo corpo (INCA, 2021. ONCOGUIA, 2014).

A cirurgia ocorre na maioria dos casos e pode desempenhar diferentes papéis no tratamento do câncer. A cirurgia pode ser feita para remover o máximo possível da área afetada, para reconstrução da mama depois da retirada do tumor ou até mesmo aliviar os sintomas e trazer melhor qualidade de vida para a paciente. Basicamente existem dois tipos de cirurgia no tratamento do câncer: A cirurgia conservadora da mama e a Mastectomia (RAMANI *et al.*, 2017. ONCOGUIA, 2020).

A Mastectomia parcial, como também é conhecida a cirurgia conservadora da mama, acontece quando é feita a retirada da área afetada, junto com um pouco de tecido normal, para margem de segurança. Geralmente este procedimento é usado em tumores com tamanho de até 2 centímetros. O quanto da mama é retirada depende do tamanho e do estadiamento do tumor (RAMANI *et al.*, 2017. ONCOGUIA, 2020).

Na mastectomia, todo o seio é retirado junto com o tecido adiposo, tecido mamário e algumas vezes outros tecidos próximos. Nódulos são retirados antes da cirurgia para buscar evidências do estágio do câncer. Depois da cirurgia, o patologista informa o tamanho do tumor e a quantidade de nódulos retirados, a fim de fornecer informações a respeito do estágio do câncer. Existem casos em que a retirada dos dois seios é necessária (RAMANI *et al.*, 2017. ONCOGUIA, 2020).

Em estágios iniciais é possível escolher por qualquer uma das duas. Geralmente, na cirurgia conservadora da mama, é necessário complementar o tratamento com uso de radioterapia. No procedimento de mastectomia o uso de radioterapia nem sempre se faz necessário (ONCOGUIA, 2020).

A Lei nº 12.802/2013 determina que mulheres que passaram pela cirurgia de Mastectomia tenham acesso imediato à reconstrução de mama realizada através do SUS (INCA, 2020). Existem vários tipos de cirurgias de reconstrução e dependem da situação clínica e de escolhas pessoais da paciente (ONCOGUIA, 2020).

Na radioterapia o crescimento e o desenvolvimento das células cancerígenas é impedido através do uso de radiações ionizantes. O raio-x é um exemplo de tecnologia que faz uso de radiação ionizante. Existem casos em que o uso de radioterapia não se faz necessário, mas o uso é comum após cirurgia conservadora da mama, em casos específicos de mastectomia e na tentativa de frear o avanço do câncer que se encontra em estágio de metástase (INCA 2019, ONCOGUIA, 2020).

Para remover células cancerígenas que permaneceram nos tecidos mamários após a retirada do tumor por cirurgia, a radioterapia é essencial. A duração da radioterapia pode chegar a 4 semanas e a dose de radiação varia de 40 a 45 gy. São dadas doses menores para que o paciente sofra menos com os efeitos colaterais da radiação. Na radioterapia externa o órgão alvo é irradiado em doses fracionadas. Geralmente este procedimento tem duração de 2 minutos.(RAMANI *et al*., 2017)

As doses são fracionadas em 5 vezes por semana, mas estão sendo desenvolvidas técnicas para diminuir a quantidade de sessões, administrando doses maiores e assim, diminuir o tempo de radioterapia do paciente (ONCOGUIA, 2020).

Quimioterapia é outro tratamento usado em pacientes com câncer. O tratamento pode ser dado por via oral ou via intravenosa. A quimioterapia adjuvante é usada logo após a cirurgia e é usada para acabar com as células cancerígenas restantes. A quimioterapia neoadjuvante é usada antes da cirurgia e tem por fim diminuir o câncer para que a cirurgia ocorra. Para casos de câncer de mama avançados, a quimioterapia pode ser usada como método principal de combate ao tumor. Nessas situações a quimioterapia é aplicada de maneira isolada (ONCOGUIA, 2020).

Considerada uma forma de terapia sistemática, a Hormonioterapia para Câncer de Mama atinge células cancerígenas por todo o corpo. É indicada em casos em que os testes para tumores com receptores de hormônios são positivos. Geralmente usada para diminuir os riscos de recidiva após a cirurgia para retirada do tumor. Costuma ser administrada em pacientes por longos períodos. A maioria dos tipos de hormonioterapia que são usados no combate ao câncer de mama, ajudam a bloquear o estrogênio que atua nas células cancerígenas (ONCOGUIA, 2020).

# 3 *MACHINE LEARNING*

A cada instante um imenso volume de dados é produzido na internet. Projeções indicam que mais de 90% do volume de dados da história da humanidade foram produzidos na última década. Para a exploração desse grande volume de dados, surge a necessidade de meios automatizados de análise. O uso de *machine learning* para análise preditiva vêm ganhando muita popularidade nos últimos anos por ser uma solução viável para este problema ( BAESENS, 2014).

Segundo Kelleher (2015), análise preditiva de dados é a arte de construir e usar modelos que fazem predições baseadas em padrões extraídos de dados. Análise preditiva está por todos os lados. Pode ser usada para diagnóstico médico ou até mesmo determinar a dosagem de um remédio para um paciente específico. Seja qual for o caso, um modelo de *machine learning* é usado para reconhecer padrões utilizando dados e assim, tomar decisões (KELLEHER *et al.,* 2015).

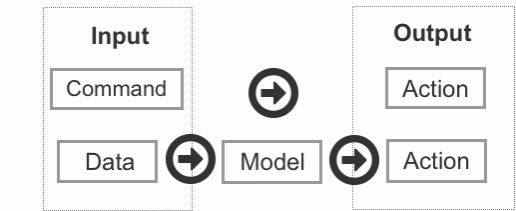
*Machine learning* é um campo da inteligência artificial. O termo *machine learning* faz referência à capacidade de reconhecer padrões em um volume de dados. Tom Mitchell, em 1997 conceituou *machine learning* como sendo algoritmos que possuem a capacidade de aprender através de experiência. A experiência, neste caso, faz referência à informação disponível para o aprendizado do modelo. Para o treinamento do modelo qualquer tipo de informação que possa ser representada na forma de números, pode ser usada. Os fatores determinantes para a qualidade na predição feita pelo modelo, são o tamanho e a qualidade dos dados disponíveis para o treinamento.

É difícil definir quando surgiu o conceito de inteligência artificial. Desde a antiguidade já se discutia a possibilidade de armazenar a 'mente' humana em um corpo mecânico. Os gregos criavam mitos a este respeito. O conceito moderno, como nós conhecemos, surgiu com Alan Turing. Em 1950, Turing lançou um artigo na revista *Mind* chamado *Computing machinery and intelligence*. Neste artigo, Turing propõe um teste para responder se máquinas são capazes de pensar. O teste ficou conhecido como Jogo da Imitação (MUELLER *et al*., 2016 ).

Também na década de 50, Arthur Samuel cunhou o termo *machine learning*. Samuel definiu *machine learning* como sendo o campo de estudos que dá aos computadores a habilidade de aprender sem serem explicitamente programados. Samuel propôs que computadores seriam capazes de aprender qualquer coisa que fossem designados. *Machine learning* nasceu da fusão de múltiplas disciplinas, tais como reconhecimento de padrões e *computer theory*.

Embora não faça parte da conceituação feita por Samuel, um elemento chave do aprendizado de máquina, é o *self-learning*. Isso acontece através da aplicação de modelos estatísticos com o objetivo de encontrar padrões nos dados utilizados para o treinamento e aprimorar a sua performance, sem ser diretamente programado para tal (THEOBALD, 2017).

**Figura 6: *Input Command vs Input Data***

****

Fonte: Machine Learning For Absolute Beginners by O. Theobald 2017

O modelo aprende através de dados que são utilizados para alimentar a máquina, algum tipo de algoritmo é selecionado e hiperparâmetros são ajustados e configurados. O modelo encontra e seleciona padrões nos dados usados para o treinamento por tentativa e erro. Depois de treinado, o modelo pode ser usado para prever futuros valores utilizando novos dados, diferentes daqueles usados durante a etapa de treinamento. Como mostra a figura 6, o modelo recebe dados de *input,* aprende padrões, e utiliza o aprendizado para executar ações. (THEOBALD, 2017).

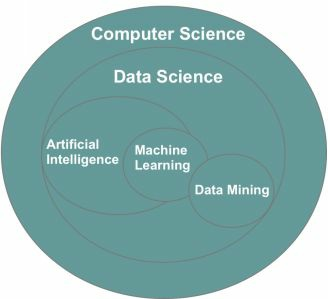
Para o aperfeiçoamento do modelo, parte dos dados são utilizados para validação do treinamento. Para isso, os dados que serão usados para o treinamento são divididos em dois *datasets* diferentes. *Dataset* de treino, que será alimentado à máquina com o objetivo de encontrar padrões, e o *dataset* de teste, que será usado para medir a qualidade dos resultados encontrados durante a etapa de treinamento (THEOBALD, 2017).

Como exemplo, podemos usar um modelo que filtra a caixa de entrada de *e-mails*. Padrões que levam a falsos positivos podem ser detectados durante o treinamento. Por isso, depois de treinado utilizando dados e *e-mails* classificados como *spam* e não *spam*, o modelo deve ser testado em cima de um *dataset* de testes. Depois de alcançar um resultado satisfatório tanto no treinamento quanto na validação feita do *dataset* de treino, o modelo vai estar pronto para filtrar *e-mails* na caixa de entrada (THEOBALD, 2017).

Outra coisa que preciso ser entendida, é como *machine learning* se relaciona com outras áreas, tais como *data mining* e *computer science*. *Machine learning,* *data mining*, programação são alguns dos campos mais relevantes que pertencem à *computer science*, que engloba tudo relacionado com *design* e uso de computadores. Dentro da *computer science*, também temos a *data science*, que engloba *machine learning* e *data mining* e consiste em métodos de extrair conhecimento e *insights* de dados através de computadores. *Data science* também é conceituada com a ciência de extrair valor de dados (THEOBALD, 2017).

*Machine learning* e *data mining* também são ciências que se sobrepõem. Sendo *data mining* a ciência que foca em descobrir padrões em grandes volumes de informação. Enquanto *machine learning* foca em treinar modelos através do *self-learning* para fazer predições, *data mining* foca em encontrar padrões e extrair *insights* do passado, de dados já conhecidos (THEOBALD, 2017). A figura 7 apresenta esta divisão.

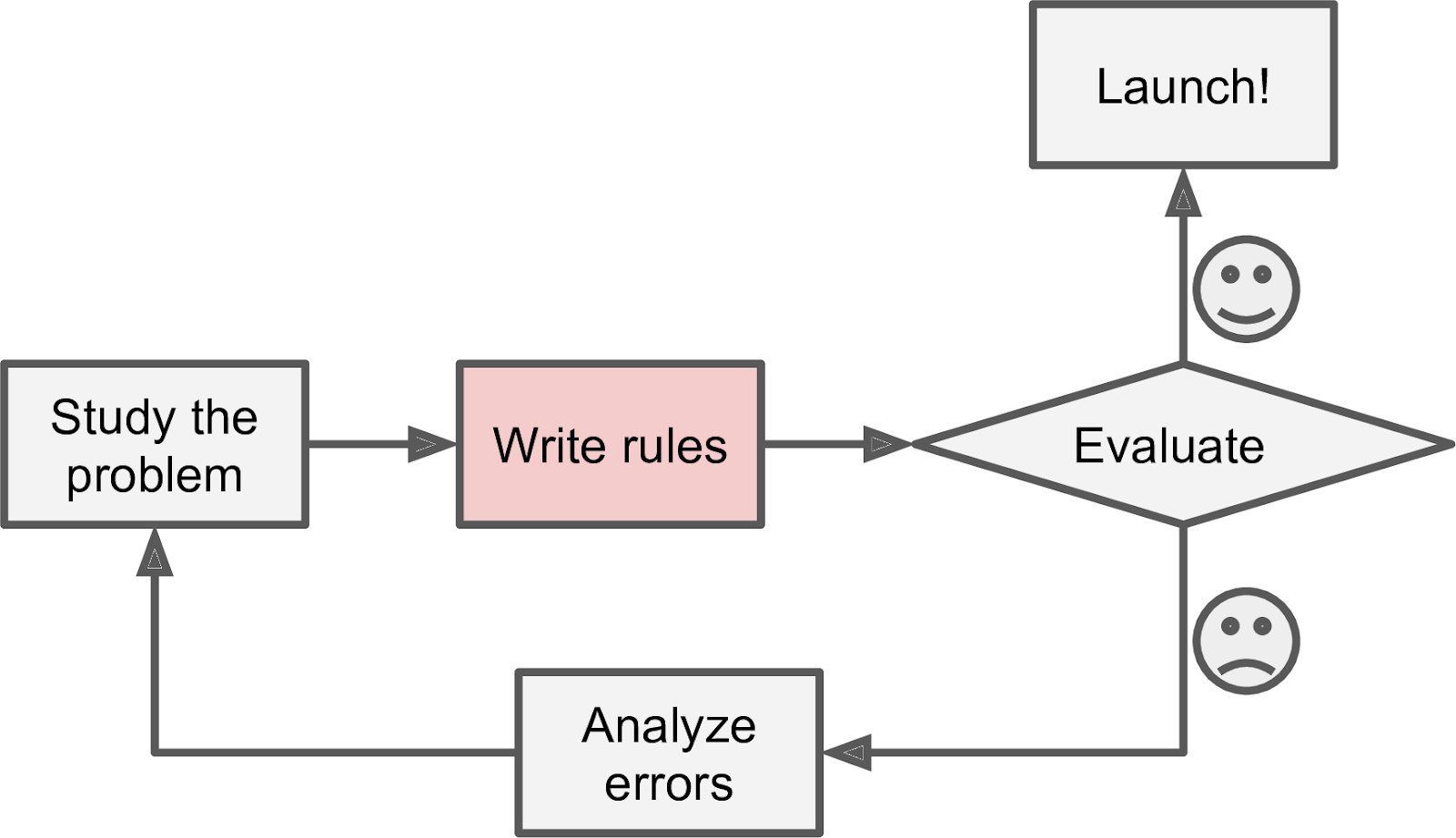
**Figura 7: *Computer science fields***

****

Fonte: Machine Learning For Absolute Beginners by O. Theobald 2017

Os problemas mais comuns solucionados por *machine learning*, são problemas que necessitam da automatização da tomada de decisão. Como mostra o fluxograma da figura 8, sistemas *hardcoded* vão apresentar dificuldades e limitações em mudanças nas tarefas que foram designadas. Sempre que a tarefa mudar, o código precisará ser reescrito. Essas limitações são superadas quando o sistema aprende, e desenvolve o seu próprio conhecimento (MULLER *et al*., 2017).

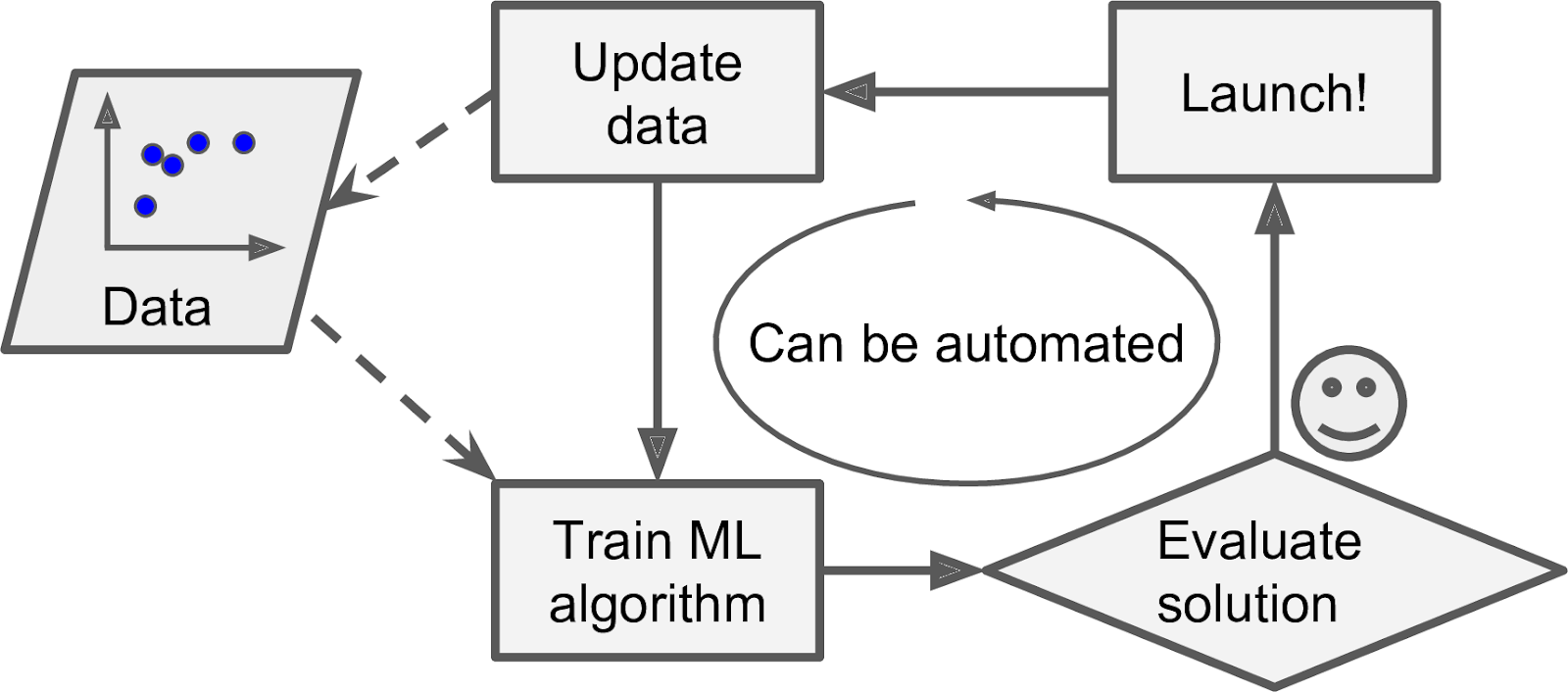
**Figura 8: Modelo tradicional**

****

Fonte: Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow by A.Géron, 2019

Para saber a necessidade de usar modelos de *machine learning* no lugar do modelo tradicional, podemos observar dois aspectos do problema: A Sua complexidade e a necessidade de um modelo que se adapta (GÉRON, 2019). Na figura 9, podemos ver o exemplo de fluxograma de um modelo que se adapta aos novos dados.

**Figura 9: Modelo *machine learning***

****

Fonte: Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow by A.Géron, 2019

Tarefas performadas por pessoas rotineiramente são exemplos de tarefas complexas demais para a definição de um programa *handcoded*. Tarefas como dirigir, reconhecimento de fala e compreensão de imagens são exemplos de tarefas feitas por humanos que modelos de *machine learning* alcançam excelentes resultados quando expostos a uma quantidade suficiente de dados (GÉRON, 2019).

Além de tarefas executadas por pessoas todos os dias, também temos tarefas que possuem nível de complexidade além das capacidades humanas e que se beneficiam do uso de modelos de *machine learning*. Com o aumento da disponibilidade de informação, fica óbvio a incapacidade humana em extrair proveito e *insights* valiosos disso (GÉRON, 2019).

*Machine learning* apresenta uma solução para estes problemas e muitos outros com a sua capacidade de aprender a detectar padrões significativos em grandes e complexos *datasets*. Em conjunto com o avanço tecnológico, memória praticamente ilimitada e velocidade de processamento, *ML* cada vez mais abre novos horizontes (GÉRON, 2019).

Classificação é uma tarefa comum que utiliza sistemas de *machine learning* que são usados para obter uma solução. Neste problema, o sistema precisa categorizar itens. Como por exemplo, categorias de produtos de uma loja virtual. Classificar imagens a suas devidas classes é outro exemplo comum que como *machine learning* é usado para classificação (MOHRI *et al*., 2018).

Outro exemplo de tarefa que pode ser solucionada com o uso de *machine learning* é a regressão. A regressão é definida como a tarefa de prever o valor para cada determinado item. Fazer a previsão de preços de um determinado produto ou o valor de ações são exemplos de problemas de regressão. Dentro de *machine learning*, regressão é um dos conceitos estatísticos mais utilizados. Se caracteriza por uma reta que representa a relação entre duas variáveis (MOHRI *et al*., 2018).

O campo do *machine learning* é vasto. É dividido em uma grande variedade de subgrupos para cada tipo diferente de problema. *O Ml* incorpora centenas de algoritmos baseados em estatísticas e saber qual usar pode ser um desafio para cientistas de dados. Nas próximas seções iremos discutir três dos subgrupos da área. Essas três categorias são su*pervised learning, unsupervised learning e reinforcement learning.*

### 

**3.1 APRENDIZADO SUPERVISIONADO**

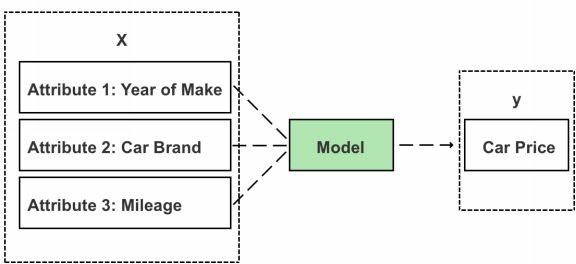
Segundo a *IBM* (2020), aprendizado supervisionado é uma subcategoria de *machine learning*. Um tipo de aprendizado. Algoritmos de aprendizado supervisionado são caracterizados pelo uso de dados rotulados para classificar ou prever resultados. Aprendizado supervisionado é o tipo de aprendizado mais usado em *machine learning*. Organizações utilizam aprendizado supervisionado para uma infinidade de tarefas do dia a dia. Um exemplo é a classificação de *e-mails* de *spam* ou até mesmo a recomendação de vídeos ou *playlists* com base no histórico do usuário.

No aprendizado supervisionado é usado um *dataset* de treino para alcançar os resultados desejados. Neste caso, o algoritmo é alimentado com *inputs* de testes e seus respectivos *outputs*, de forma a aprender os padrões que mapeiam as entradas e os resultados. Durante o treinamento é usada uma função de *loss* para medir o desempenho dos resultados obtidos e ajustes podem ser feitos no modelo para minimizar o *loss*. O *dataset* com *inputs* e seus *outputs* é chamado de t*rain dataset* e é ele que guia o desenvolvimento do modelo (PANESAR, 2019. IBM, 2020).

Um algoritmo de aprendizado supervisionado pode performar predições analisando a relação entre variáveis. Usando como exemplo o preço de carros usados, como mostra a figura 10, seriam analisados os itens e condições do carro e o preço de outros carros vendidos baseado em um histórico de vendas. Sabendo o preço de outros carros vendidos, o algoritmo pode tentar encontrar relação entre características desses carros com o preço que foram vendidos (THEOBALD, 2017).

Depois que a máquina aprende as regras e padrões extraídos dos dados, ela cria um modelo. O modelo é definido como sendo uma equação para produzir um valor de saída com base nas regras que foram aprendidas durante o treinamento utilizado em algum *dataset*. Depois de treinado, o modelo deve ser testado com novos dados, normalmente chamada de *dataset* de teste. Esse teste é feito para testar a precisão das predições que o modelo executa. Depois de passar por essas duas etapas, o modelo está pronto para ser usado em situações do mundo real (THEOBALD, 2017).

**Figura 10: Exemplo de modelo**

****

Fonte: Machine Learning For Absolute Beginners by O. Theobald 2017

O aprendizado supervisionado somente pode ser usado quando o *input* e seu *output* são conhecidos. A grande vantagem do uso do aprendizado supervisionado para a criação de modelos de *machine learning*, é que ele fornece os resultados com maior precisão (PANESAR, 2019. MOHANTY *et al.,* 2021).

Quando trabalhamos com *machine learning* damos de cara com dois tipos de problemas de treinamento supervisionado, que podem ser classificados de duas formas: Problemas de regressão e problemas de classificação (PANESAR, 2019. MOHANTY *et al*., 2021).

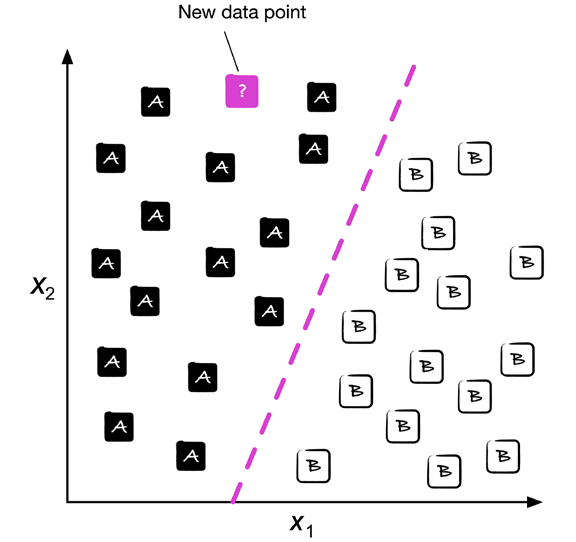
Muitos algoritmos podem ser classificados entre algoritmos de regressão e classificação. Esses algoritmos são utilizados para o aprendizado supervisionado de acordo com cada situação específica (IBM, 2020).

Variáveis são classificadas em dois grupos distintos. Quantitativas e qualitativas, também conhecidas como categóricas. Variáveis quantitativas são representadas por números, como por exemplo, a idade de uma pessoa, a sua altura, seu peso, o valor de um carro usado ou preços de ações. Variáveis qualitativas por sua vez, podem ser representadas com N classes. São exemplos de variáveis qualitativas o estado civil de uma pessoa, a marca de produtos ou até mesmo o diagnóstico de uma doença (IBM, 2020).

Problemas quantitativos são conhecidos como problemas de regressão, enquanto problemas qualitativos são chamados de problemas de classificação. Os métodos estatísticos utilizados para resolver estes problemas são escolhidos de acordo com a classificação dele. Se é um problema de regressão ou classificação. Por exemplo, para um problema de regressão poderá ser utilizado um algoritmo de regressão linear, enquanto para um problema de classificação, poderá ser utilizado um algoritmo de regressão logística.

Classificação é como é chamada a subcategoria do aprendizado supervisionado que tem como objetivo classificar classes de objetos baseados no aprendizado de dados conhecidos. Essas classes são representadas por números discretos e não ordenados. Um exemplo de classificação é a caixa de entrada de e-mail, onde o algoritmo faz a classificação binária em e-mails recebidos. 0 para spam e 1 para não spam (RASCHKA *et al.*, 2022).

**Figura 11: Exemplo de classificação binária**

****

Fonte: Machine Learning with PyTorch and Scikit-Learn by S. Raschka et al. 2022

A figura 11 mostra um exemplo de classificação binária, com 30 itens classificados em 2 classes. 15 itens são classificados como pertencendo a classe A e os outros 15 itens classificados como pertencendo a classe B. Nesse exemplo, nossos dados possuem duas dimensões. Isso significa que cada elemento tem dois valores associados a ele. X1 e X2. Nesse exemplo, a função do modelo de aprendizado supervisionado seria aprender através de padrões a linha pontilhada e dessa forma, ser capaz de classificar novos dados utilizando os dados de entrada X1 e X2 (RASCHKA *et al*., 2022).

Entretanto, a classificação não se limita a classes binárias. Um algoritmo de classificação é capaz de aprender qualquer classe presente em um *dataset*. Um exemplo de classificação multiclasse é o reconhecimento de caracteres escritos à mão. Com acesso a um *dataset* contendo imagens de letras escritas à mão, podemos representar cada letra como uma classe diferente. Depois de treinado, o modelo deve ser capaz de reconhecer e classificar letras utilizadas como *input* com nível de precisão satisfatório (RASCHKA *et al*., 2022).

Baseado no teorema de *Bayes*, o algoritmo *Naive Bayes* é classificado com um algoritmo de classificação. Então, a presença de um *feature* não influencia em outra *feature* na probabilidade do resultado. Ou seja, desconsidera a relação entre *features* (IBM, 2020).

Esse algoritmo geralmente é usado em sistemas de recomendação, classificação de texto e também em sistemas de identificação e *e-mails* de *spam*. Algoritmos *Naive Bayes* também podem ser classificados em três tipos (IBM, 2020).

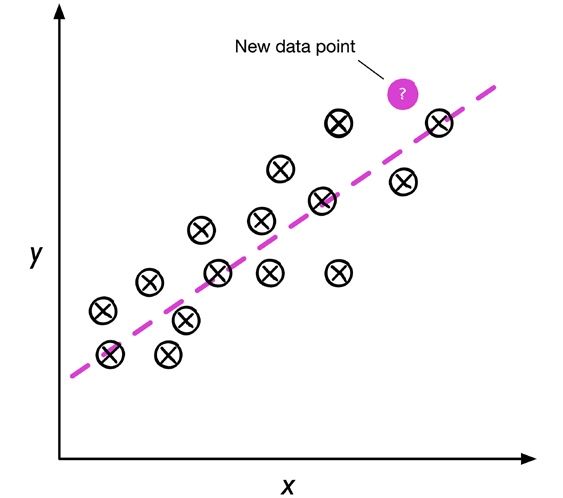
O algoritmo *Naive Bayes* faz parte de uma família de classificadores que se parece com outros algoritmos. Tende a ser um algoritmo que demonstra grande eficiência no treinamento dos modelos. A razão para que o algoritmo *Naive Bayes* ser tão eficiente é que ele analisa *features* individualmente coletando estatísticas de cada uma delas. (MULLER *et al.,* 2017)

Regressão, também conhecida como análise de regressão, é o tipo de aprendizado supervisionado que consiste em prever resultados contínuos. Na análise de regressão usamos uma variável explicativa com seu respectivo resultado e tentamos achar a relação entre essas duas variáveis que permite ao modelo fazer predições (RASCHKA *et al.,* 2022).

O termo regressão foi concebido pelo matemático Francis Galton em 1888 em seu artigo chamado *Regression towards Mediocrity in Hereditary Stature*. Galton descreveu como a variação de altura de uma população não aumenta com o tempo. Ele observou que a altura dos pais não é passada para seus filhos, mas a altura de seus filhos é regressora com a média da população (RASCHKA *et al*., 2022).

É importante saber que no campo do aprendizado de máquina as variáveis de predição são normalmente conhecidas como *features* e as variáveis de resultado são chamadas de target variables (RASCHKA *et al*. 2022).

**Figura 12: Exemplo de regressão**

****

Fonte: Machine Learning with PyTorch and Scikit-Learn by S. Raschka et al. 2022

*Linear regression* (Figura 12) é um algoritmo de regressão. É usado para identificar a relação entre variáveis e fazer predições. Uma variável é considerada a variável independente e a outra variável é considerada a variável dependente. Quando existe apenas uma dependente para uma dependente, o modelo é chamado de *Simple Linear Regression*. Quando existem mais variáveis dependentes, o modelo passa a ser chamado de multiple linear regression (IBM, 2020. MULLER *et al*., 2017)

### 3.2 APRENDIZADO NÃO SUPERVISIONADO

Aprendizado não supervisionado recebe esse nome por utilizar algoritmos de *machine learning* em dados não rotulados. Geralmente esses algoritmos são usados para classificar e agrupar esses dados. Com a proliferação de dados não rotulados, algoritmos de aprendizado não supervisionado vêm ganhando grande popularidade. Podem ser usados em uma grande variedade de aplicações, como reconhecimento de padrões, análise de redes sociais e *web mining* (IBM, 2020. CELEBI, 2016).

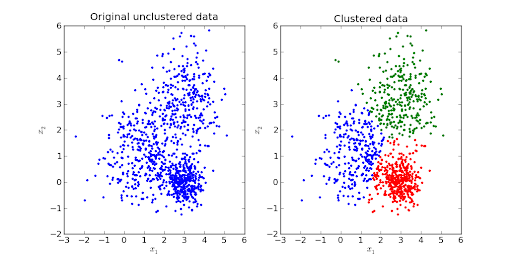
O aprendizado não supervisionado permite que empresas consigam identificar padrões em grandes volumes de dados rapidamente. *Google news* utiliza essa técnica de aprendizado para categorizar artigos, *computer vision* utiliza m*achine learning* com aprendizado não supervisionado para categorizar imagens e graças a isso ele também pode ser usado na prática médica e também muito utilizado para traçar perfil de consumidor para empresas de e-commerce (IBM, 2020).

Por não ter rótulos nos dados, validar o *output* se mostra um grande desafio quando trabalhamos com algoritmos de aprendizado não supervisionado. Dimensionar o resultado do treinamento do modelo é uma tarefa difícil. Como consequência dessa dificuldade, algoritmos de aprendizado não supervisionado acabam sendo muito usados para obter um entendimento maior sobre os dados, ou como etapa de pré-processamento, antes do uso de aprendizado supervisionado (MULLER *et al*., 2017).

*Clustering* é uma técnica para agrupar dados não rotulados em grupos (*clusters*). No *clustering*, o agrupamento é feito baseado em similaridades encontradas entre os grupos. Usado em dados não classificados ou não processados, o objetivo é identificar padrões e estruturas nas informações. Um uso comum para *clustering* é o agrupamento de clientes de uma determinada loja. Podemos classificar algoritmos de *clustering* em subgrupos: *specifically exclusive, overlapping, hierarchical, and probabilistic* (IBM, 2020).

*K-means clustering* (Figura 13) é um dos algoritmos mais simples e comuns de *exclusive clustering*. Ele tem por objetivo encontrar *clusters centers* (centróides) que representam posições nos dados. Os *data points* mais próximos de um determinado *cluster center*, serão classificados como sendo parte daquela categoria. *K-means clustering* é muito usado para segmentação de imagens, compressão de imagens, e agrupamento de documentos (IBM, 2020. MULLER *et al*., 2017).

**FIGURA 13 - Exemplo de dados clusterizados pelo algoritmo *K-means*.**



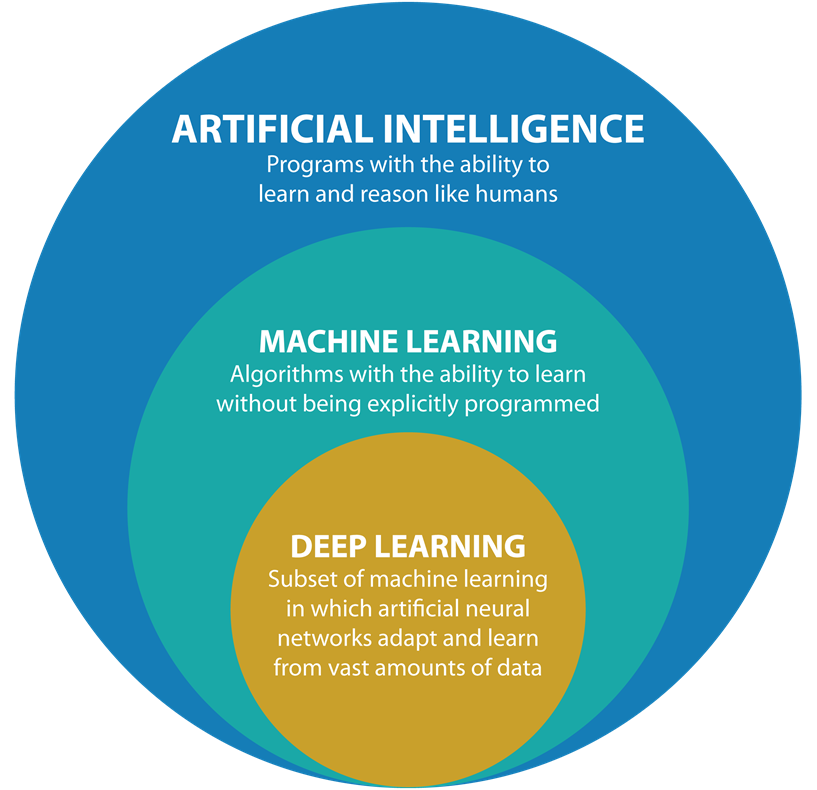
Fonte: Introduction to Machine Learning with Python A Guide for Data Scientists by A. C. Müller et al. 2017

Também conhecido como *hierarchical cluster analysis*, o *Hierarchical clustering* é outro algoritmo de aprendizado não supervisionado utilizado para o agrupamento de dados. Este algoritmo pode ser categorizado em dois tipos: *Agglomerative* ou *divisive*. Divisive clustering utiliza uma abordagem '*Top-down*' e esta categoria não costuma ser muito usada. *Agglomerative* *clustering* utiliza a abordagem '*Down-up'* e é a categoria mais usada. (IBM, 2020, MULLER *et al*., 2017).

### 3.3 *DEEP LEARNING*

Inteligência artificial, *machine learning* e *deep learning* se tornaram termos muito usados nos últimos anos. Inteligência artificial é definida como qualquer técnica que permite simular atividades humanas. *Machine learning*, um *subset* de inteligência artificial, é definida como programas com a capacidade de aprender sem serem explicitamente programados para. *Deep learning*, um conceito que já existe há décadas é um *subset* de *ML*, como mostra a figura 14, consiste essencialmente em redes neurais com três ou mais *hidden layers*. (CHOLLET, 2018. IBM 2020).

**FIGURA 14 - Inteligência artificial, *Machine learning* e *Deep Learning***

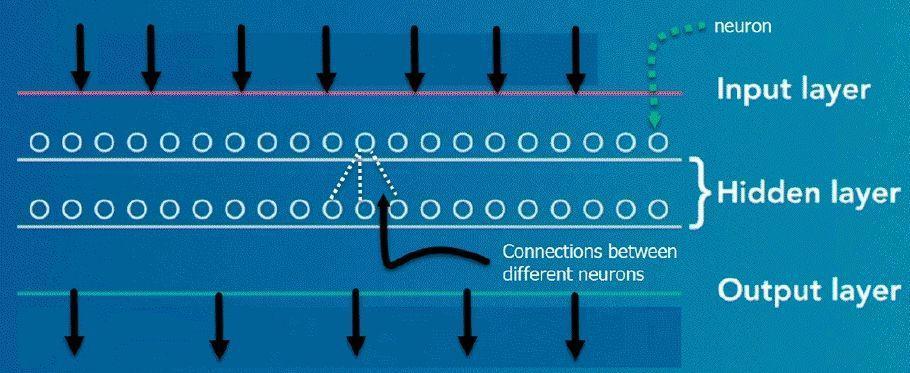
****

Fonte: Medium, 2018, p.2

A técnica de aprendizado profundo recebe este nome pois faz uso de múltiplas *hidden layers* e *neurons*. A ideia por trás do DP é criar um algoritmo capaz de imitar o funcionamento do cérebro humano, simulando redes neurais. Para isso, é feito uso da combinação de data *inputs*, *weights* e *bias*. Esses elementos são usados em conjunto a fim de aumentar a precisão dos resultados alcançados (RUNGTA, 2018. IBM, 2020).

Algoritmos de *deep learning* são formados por *connected layers*. As camadas de *input* e *output* são chamadas de *visible layers* e as camadas do meio são chamadas de *hidden layers*. Algoritmos de aprendizado profundo são formados por pelo menos três *hidden layers*. A camada de *input* é onde o algoritmo recebe os dados e a camada de *output* é onde é feito o resultado de predição ou classificação (RUNGTA, 2018. IBM, 2020). Esta dinâmica é exemplificada na figura 15.

**FIGURA 15: Estrutura de um algoritmo de *Deep Learning***



Fonte: TensorFlow in 1 Day: Make your own Neural Network por Krishna Rungta

Cada *hidden layer* é formada por *neurons* e cada *neuron* é conectado a outros *neurons*. O trabalho deles é receber, processar e propagar os dados de *input* para a próxima camada. Este processo é conhecido como *forward propagation*. Os dados passados para a camada superior dependem do *weight*, *bias* e da *activation function*. O algoritmo recebe uma grande quantidade de dados como *input*, processa esses dados pelas múltiplas *hidden layers* e, desta forma, as redes neurais são capazes de aprender complexas *features* dos dados a cada camada (RUNGTA, 2018. IBM, 2020).

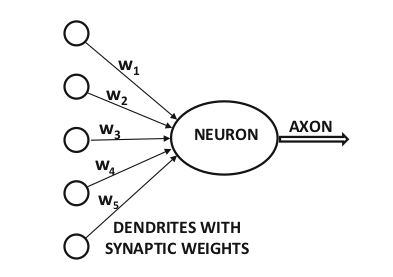
Cada camada representa um nível de aprendizado, quanto mais *layers*, mais profundo é o aprendizado. Esse processo é conhecido como hierarquia de conhecimento e uma rede neural com cinco *layers*, é capaz de aprender *features* e realizar tarefas muito mais complexas do que uma com apenas 3 *layers*. O processo de aprendizado por *layers* tem duas etapas. A primeira etapa é definida pela aplicação de uma transformação não linear e a criação de um modelo estatístico como saída. A segunda etapa consiste na melhora do modelo utilizando um método matemático. Essas duas etapas são repetidas milhares de vezes. Esse processo de repetição é chamado de iteração e a iteração é feita até o modelo atingir um nível de precisão satisfatório (RUNGTA, 2018).

# 4 *ARTIFICIAL NEURAL NETWORKS*

*Artificial neural networks*, como o próprio nome sugere, é uma tentativa grosseira de simular a capacidade de tomada de decisão do cérebro humano. *ANN* tentam simular o modelo *cell-by-cell* se baseando em conhecimentos neurofisiológicos e em neurônios biológicos e nas conexões entre esses neurônios (GRAUPE, 2013).

Por tanto, *Artificial neural network* é uma máquina feita para simular o modo como o cérebro humano executa determinadas tarefas e funções. Para isso, *ANN* emprega uma grande quantidade de unidades simples (Figura 16), também conhecidas como "*neurons*" ou unidades de processamento interconectados (HAYKIN, 2009).

**Figura 16 : Exemplo da estrutura básica de um *neuron***

******

Fonte: Neural Networks and Deep Learning: A textbook by Graupe, 2013

Diferentemente da programação convencional, *Artificial neural network* permite a solução de problemas complexos, problemas matemáticos, não-lineares e problemas estocásticos, utilizando apenas uma série de operações matemáticas simples. Enquanto isso, a programação convencional irá aplicar séries de equações complexas e a solução será aplicada somente a um problema específico. *ANN* oferece uma solução computacionalmente simples, que se adapta e portanto soluciona uma grande variedade de problemas (GRAUPE, 2013).

Os *neurons*, que são unidades computacionais que simulam mecanismos biológicos, estão conectados uns aos outros através de *weights*, que são representações de conexões sinápticas em sistemas biológicos. Os *inputs* são conectados com os *neurons* pelos *weights*, que influenciam no resultado das funções executadas por cada uma dessas unidades computacionais (GRAUPE, 2013).

Uma *artificial neural network* utiliza *weights* como parâmetros intermediários entre os dados de *input* e *output*. Cada unidade computa os valores de *input* e propaga para as próximas camadas. O aprendizado ocorre com o ajuste dos *weights* conectados com cada *neuron* (GRAUPE, 2013).

Para o aprendizado de organismos biológicos são necessários estímulos externos. Em *ANN* esse estímulo externo ocorre com o uso de um *dataset,* contendo exemplo de pares de valores de *input* e seu respectivo *output*. Em um modelo que será treinado para classificar imagens, os dados de *input* serão os *pixels* que compõem a imagem e o *output* o *label*, com a classe da imagem a ser classificada (GRAUPE, 2013).

O *dataset* é usado para alimentar as *Artificial* *neural networks*, que irão tentar achar relações entre os dados de entrada com os *labels* servidos ao modelo. Os dados de *output* fornecem *feedback* do quanto o modelo aprendeu bem e com base nisso, ajustes são feitos nos *weights* que conectam as camadas de *neurons* (GRAUPE, 2013).

O ajuste nos *weights* tem por objetivo modificar os valores que são computados por cada *neuron* para adequar os resultados ao *feedback* recebido e manter as previsões mais corretas em futuras iterações. Os *weights* são ajustados seguindo uma lógica matemática, com o objetivo de reduzir o erro alcançado durante a iteração anterior (GRAUPE, 2013).

Ajustando os *weights* entre as camadas de *neurons*, os valores computados pelos *neurons* são refinados e fornecem predições mais precisas. Se a *ANN* for treinada com uma quantidade suficiente de imagens de uma determinada classe, ela será capaz de classificar com precisão imagens nunca antes vistas (GRAUPE, 2013).

A habilidade de reconhecer imagens nunca vistas depois do modelo ser treinado com uma quantidade suficiente de dados, é conhecida como *model generalization*. O objetivo central de todo modelo de *machine learning*, é a capacidade de generalizar o aprendizado obtido através do treinamento para dados nunca antes vistos pelo modelo (GRAUPE, 2013).

# 4.1 *ANN ALGORITMOS*

Em *ANN*, *neurons* são conectados através de *weights*. Os *weights* são ajustados por algoritmos de aprendizado. Em 1949, *Hebb* propôs o que veio a ser conhecido como *The Hebbian learning rule*. A regra proposta por *Hebb* dizia que o *weight* entre dois *neurons* é reforçado se os dois são ativados ao mesmo tempo. O algoritmo aprende a relação entre os *neurons* através dos seus *weights (ALPAYDIN, 2016)*.

Se um *neuron* checa a existência de um círculo em uma imagem, enquanto outro *neuron* é responsável pela detecção do dígito "6", sempre que o dígito "6" for detectado, o círculo que compõe o dígito também é identificado. Desta forma, a conexão entre estes dois *neurons* é reforçada. Enquanto um *neuron* responsável pela detecção do dígito "7" não é reforçado pela detecção de um círculo (ALPAYDIN, 2016).

*ANN* são estruturadas com *neurons* de *input* e *neurons* de *output*. Utilizando um *dataset* com valores de *input* com os seus respectivos valores de *output*, o que é conhecido como aprendizado supervisionado, o algoritmo propaga a atividade ao longo das *ANN* de acordo com os *weights* e com a estrutura das *Artificial neural networks (ALPAYDIN, 2016)*.

Uma função de *error* é definida pela soma das diferenças entre os valores encontrados durante o treinamento e os valores de *output* contidos no *dataset* utilizado para o treinamento das *ANN*. Dessa forma, os *weights* são atualizados com o objetivo de diminuir o valor do *error*. Diminuindo o valor do *error*, na próxima iteração, encontramos resultados similares aos encontrados no *dataset* de treino (ALPAYDIN, 2016).

Essa importante característica dos algoritmos de *ANN* é conhecida como *Online learning*, e funciona através de pequenos *updates* nos *weights* que conectam os *neurons*, de acordo com os dados de treino (ALPAYDIN, 2016).

# 4.2 HISTÓRIA DAS *ARTIFICIAL NEURAL NETWORKS*

Estudos neurobiológicos são a origem de todo o trabalho feito em *Artificial neural networks* e datam de mais de um século atrás. O funcionamento do sistema nervoso é matéria de estudo a décadas (MEHROTRA *et al.,* 1996).

Até o meio do século passado ainda não entendíamos como nervos respondiam a diferentes tipos de estímulos de correntes elétricas, o limite mínimo de corrente elétrica necessária para a ativação dos nervos, como células nervosas se comunicam ou como diferem de comportamento, como funciona o aprendizado, esquecimento, reconhecimento e como essas tarefas são feitas pelo cérebro humano e de animais. Experimentos psicofísicos nos ajudaram a entender melhor essas questões e serviram de base para o desenvolvimento das *ANN* (MEHROTRA *et al*., 1996).

O primeiro modelo matemático de um *neuron* foi criado em 1943 por McCulloch e Pitts. O modelo criado por McCulloch e Pitts foi amplamente utilizado em outros trabalhos. O principal assunto de pesquisa envolvendo este modelo matemático, era a sua capacidade de generalizar o aprendizado para todos os tipos de tarefas, ao mesmo tempo que se mantinha simples e sem exigir grande capacidade computacional (MEHROTRA *et al*., 1996).

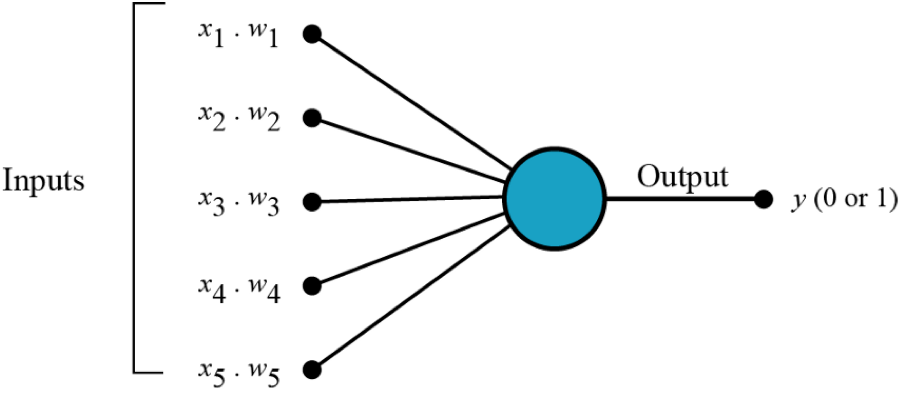
Apesar de métodos de aprendizado e capacidade de execução de tarefas serem as principais matérias de estudos ligadas a *ANN*, o trabalho desenvolvido por McCulloch e Pitts foca mais na construção de um modelo artificial de um neurônio e na suas capacidades computacionais do que na sua habilidade de aprender (BRAGA *et al*., 2000).

Pesquisas envolvendo o aprendizado de *Artificial neural networks* só passaram a ser feitas anos depois do modelo matemático de McCulloch e Pitts. Em 1949, Donald Hebb apresentou o que foi considerado o primeiro trabalho envolvendo o aprendizado de *ANN* (BRAGA *et al*., 2000).

*Hebb's learning rule*, como veio a ser conhecido o trabalho de Donald Hebb, modifica os *weights* de conexão verificando se dois *neurons* estão simultaneamente *on* ou *off*, indicando relação entre esses dois *neurons*. *Hebb's learning rule* ainda é vastamente utilizada (MEHROTRA *et al*., 1996).

Em 1958 o modelo do *perceptron* foi apresentado por Rank Rosenblatt. O *perceptron* tinha a habilidade de reconhecer e classificar certos tipos de padrões. A estrutura básica do *perceptron* apresentado por Rosenblatt possui três *layers* (Figura 17*)*. A primeira *layer* recebe os dados de *input*, a segunda *layer* se comunica com a primeira através de *weights* ajustáveis e envia valores para a *output layer* (BRAGA *et al*., 2000).

**Figura 17: Estrutura d*o Perceptron***

******

FONTE: Data Science, 2020

O modelo inicialmente apresentado do *perceptron* apresentava limitações. Só era capaz de resolver problemas linearmente separáveis. Não conseguia detectar paridade, conectividade e simetria. Esses exemplos representam uma grande parcela de problemas e são conhecidos como *hard learning problems*. Essa limitação do *perceptron* foi exposta em 1969 por Minsky e Papert (BRAGA *et al*., 2000).

Por conta do trabalho de Minsky e Papert e outros fatores, as pesquisas na área esfriaram nos anos 70. Em 1982, John Hopfield publicou um artigo demonstrando as propriedades associativas das *ANN*. *ANN* de múltiplas camadas se mostraram capazes de resolver *hard learning problems* e com isso houve uma nova explosão de interesse na área na década de 80. Junto com o artigo de Hopfield, o avanço tecnológico também foi responsável por essa explosão de interesse (BRAGA *et al*., 2000).

# 4.3 NEURÔNIOS BIOLÓGICOS VS ARTIFICIAIS

O cérebro humano possui bilhões de neurônios. Cada um deles processa e se comunica com milhares de outros neurônios paralelamente e de maneira contínua. A estrutura individual e o comportamento desses neurônios é o que inspira o estudo para o desenvolvimento das *Artificial neural network* (BRAGA *et al*., 2000).

O nosso cérebro é responsável pelo o que conhecemos como emoção, pensamento, percepção e cognição. Também é responsável por executar funções sensório motoras e autônomas. Além disso, nosso cérebro também é capaz de aprender por experiência e reconhecer e relacionar padrões (BRAGA *et al*., 2000).

O objetivo das *ANN* é simular essas funções das redes biológicas do nosso cérebro. Apesar de grandes avanços na área, as *Artificial neural networks* ainda diferem muito das redes biológicas. Porém, as duas ainda possuem similaridades e estas podem ser usadas para compreender melhor o sistema nervoso e buscar mais avanços na área de inteligência artificial (BRAGA *et al.,* 2000).

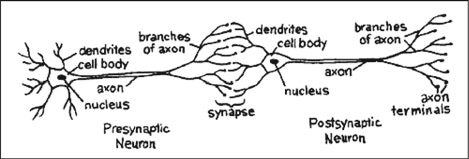
Entre as características comuns, podemos citar as duas mais importantes. A primeira é que os dois sistemas são baseados em unidades de computação paralela e distribuída e que se comunicam por conexões. A segunda é que possuem detectores de características, redundância e modularização das conexões. Graças a estas similaridades as *ANN* podem reproduzir várias funções do nosso cérebro (BRAGA *et al.,* 2000).

**4.3.1 NEURÔNIOS BIOLÓGICOS**

Neurônios biológicos (Figura 18) são divididos em três partes: o corpo da célula, os detritos e o axônio. Cada parte é responsável por funções específicas e essas funções se complementam. Os detritos são responsáveis por receber informações de outros neurônios e enviá-las para o corpo da célula. No corpo da célula, as informações serão processadas e novas informações são geradas e transmitidas para outros neurônios pelo oxônio (BRAGA *et al*. 2000).

Os neurônios são ligados pelas sinapses. Sinapses são os pontos de contato entre a terminação axônica de um neurônio e o dendrito de outro. Essas ligações formam as redes neurais. As sinapses são responsáveis por controlar a transmissão entre informação dos neurônios (BRAGA *et al*. 2000).

**Figura 18: Neurônio biológico**

****

Fonte: Redes neurais artificiais by Braga et al, 2000

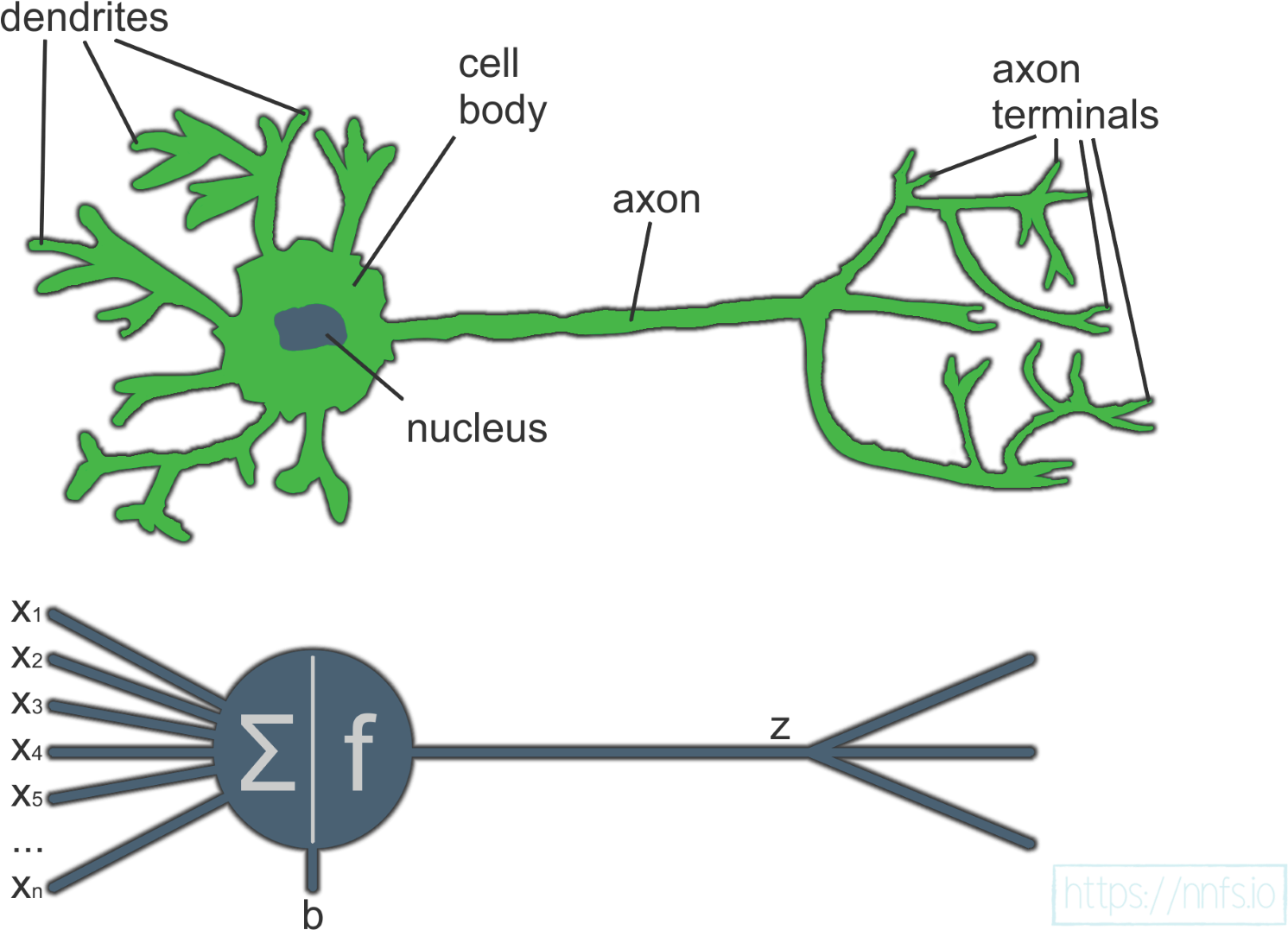
**4.3.2 NEURÔNIOS ARTIFICIAIS**

McCulloch e Pitts criaram o modelo do *perceptron* com base no que se sabia sobre os neurônios biológicos na época. Sua proposta matemática resultou em um modelo que recebia N dados de *input*. Os dados de *input* são uma representação dos dendritos. O modelo contava com apenas um terminal de saída, que por sua vez, representava o axônio (BRAGA *et al*., 2000).

Cada dado de *input* tem um *weight* acoplado. A função do *weight* é simular o funcionamento das sinapses. O valor dos *weights* podem ser negativos ou positivos. Negativos em caso de a sinapse ser inibitória e positivo quando as sinapses são excitatórias. Durante o aprendizado da *Artificial neural network*, o *weight* determina o quanto o *neuron* "importa" naquela conexão (BRAGA *et al*., 2000).

Neurônios biológicos disparam quando os impulsos recebidos ultrapassam os seus limites de excitação. No caso de *ANN*, o corpo do neurônio é simulado por uma função simples que faz a soma dos valores recebidos pelos *inputs* e decide se o *neuron* deve disparar quando comparado com o *threshold* do *neuron* . A saída da função é 1 ou 0. A função responsável pela ativação do *neuron* é conhecida por *activation function* (BRAGA *et al*., 2000)*.* A figura 19 mostra como o neurônio artificial tenta representar os elementos que formam o neurônio biológico.

**Figura 19: Neurônio biológico vs Artificial**

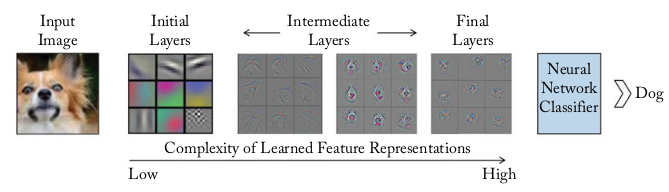
****

Fonte: Hands-On Convolutional Neural Networks with TensorFlow by ZAFAR L. et al 2018

# 5 *CONVOLUTIONAL NEURAL NETWORKS*

*Convolutional neural network* está entre as mais comuns categorias de *ANN*. Principalmente quando se fala se *high-dimentiona*l *data*, como vídeos e imagens, por exemplo. Seu funcionamento se assemelha muito ao modelo tradicional de *Artificial neural networks*. A principal diferença entre *CNNs* e o modelo tradicional está na estrutura das *layers, como demonstra a figura 20*. Cada *CNN layer* funciona como um filtro *multi-dimentional* (KHAN *et al.,* 2018).

**Figura 20: Representação de uma *CNN* e suas *layers***

****

Fonte: A Guide to Convolutional Neural Networks for Computer Vision by Khan et al, 2018

Isso explica o motivo de a *CNN* ser popular para o aprendizado de padrões em *high-dimentional data*. Por ter um *spatial shape* como *input*, *CNNs* são capazes de incorporar o contexto espacial e usar *parameter sharing* para diminuir o número de variáveis de aprendizado (KHAN *et al*., 2018).

Dando um breve contexto histórico, um dos primeiros modelos de *Convolutional neural network* foi proposto em um trabalho feito por Fukushima e Miyake no ano de 1982. O modelo proposto por eles foi chamado de *Neocognitron model* e consistia em múltiplas *layers* que aprendiam abstrações de *features* para reconhecimento de padrões (KHAN *et al*., 2018).

O modelo proposto por Fukushima foi motivado por trabalhos de pesquisa sobre o córtex visual primário. Em 1959, Hubel e Wiesel demonstraram que os neurônios no cérebro são organizados em forma de camadas. Essas camadas de neurônios têm a capacidade de reconhecer padrões extraindo *features* e combinando-as para formar representações de alto nível (KHAN *et al*., 2018).

Em 1989, LeCun contribuiu para o aperfeiçoamento do *Neocognitron model*. Utilizando *error back-propagation* e regras do *reinforcement learning*, o modelo foi capaz de aprender a reconhecer dígitos escritos a mão, um clássico problema de reconhecimento de padrões na área de *machine learning* (KHAN *et al*., 2018).

*Convolutional neural networks* trabalha bem tanto com aprendizado supervisionado quanto não supervisionado. *CNN* aprende a classificar uma imagem com a sua devida classe detectando *features* abstratas e variando das mais simples para as mais complexas. Essas *features* serão usadas pelo modelo para a classificação correta da imagem (KHAN *et al*., 2018).

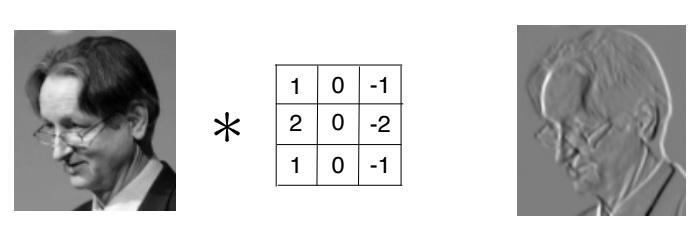
*Two-dimensional convolutions* são utilizados em processamento de imagens para a implementação de filtros. Esses filtros são utilizados para detectar e extrair determinados tipos de *features* (ZAFAR *et al*., 2018).

A *convolutional layer* filtra um *input* utilizando uma matriz. Essa matriz é o que chamamos de *kernel*. O *kernel* desempenha uma função importante em *convolution operations* e define o que será filtrado e irá produzir respostas quando conseguir encontrar o que está filtrando (ZAFAR *et al*., 2018).

Durante o aprendizado, os parâmetros que são modificados são os *weights* da *kernel layer*. Esses valores são ajustados automaticamente com o objetivo de extrair o máximo possível de informação durante a etapa de treinamento (ZAFAR e*t al*., 2018).

No modelo tradicional, os dados de *input* são convertidos em um vetor. Isso causa a perda de informação espacial que a *high-dimentional data* poderia ter. Cada pixel tem um parâmetro para cada *neuron* da camada seguinte, significando que quanto mais dados de *input* e maior o modelo, mais parâmetros de aprendizado nós teremos. A *CNN* oferece uma solução para esse problema utilizando o *kernel*, que procura por padrões específicos, assim como demonstrado na figura 21 (ZAFAR *et al*., 2018).

**Figura 21: Exemplo de kernel aplicando filtro de Sobel**

****

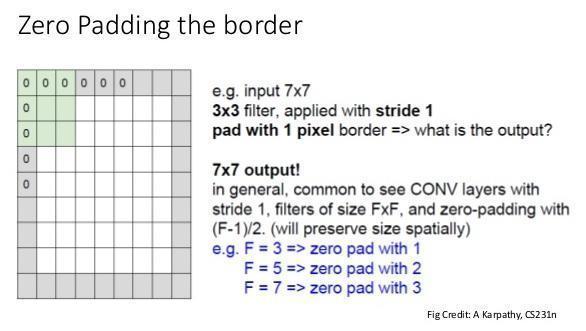
Fonte: Hands-On Convolutional Neural Networks with TensorFlow Solve computer vision problems with modeling in TensorFlow and Python by Zefar et al,2018

**5.1 *PADDING***

O uso de *padding* poderá ser necessário em caso do *kernel* ultrapassar o *activation map*. A função do *padding* é preservar informação nas bordas do *activation map*, com o objetivo de melhorar a performance do aprendizado. O uso do *padding* também preserva o *spatial size* do *input*, dessa forma permitindo uma arquitetura mais complexa de *CNN* (ZEFAR *et al*., 2018).

Existem diferentes tipos de abordagens para *padding*, mas a abordagem mais comum utilizada é a de *zero-padding* (Figura 22). Isso se deve a performance, simplicidade e eficiência computacional. *Zero-padding* é simplesmente adicionar zeros em volta das bordas do *input* (ZEFAR *et al*., 2018).

**Figura 22: Técnica do *Zero-padding***

****

Fonte: Hands-On Convolutional Neural Networks with TensorFlow Solve computer vision problems with modeling in TensorFlow and Python by Zefar et al,2018

**5.2 *KERNEL SIZE***

Se refere ao tamanho da janela do filtro que será aplicado. É um importante hiperparâmetro e tem grande impacto na extração de *features* na classificação de imagens. A escolha de tamanhos menores para o *kernel* pode melhorar significantemente a quantidade de informação extraída em imagens com grande quantidade de *local features* (CNN EXPLAINER, 2020).

Outro ponto positivo na escolha de tamanhos menores do *kernel*, é que ele proporciona uma menor redução nas dimensões das *layers*. Escolher um tamanho maior para o *Kernel* é frequentemente associado a piores performances, porém, podem ser adequados para extração de *features* maiores. A escolha do tamanho do *kernel* deve sempre adequar o *dataset* e o tipo de tarefa a ser executada (CNN EXPLAINER, 2020).

**5.3 *STRIDE***

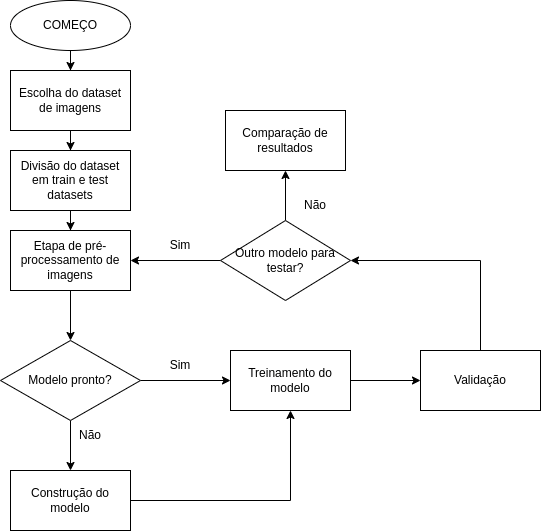
Indica a quantidade de pixels da imagem que deve "deslizar" por vez. Isso significa, quando o *stride* for igual a um, que um *kernel* 3x3 irá produzir o *output* e deslizar um *pixel* para o lado para produzir o próximo (CNN EXPLAINER, 2020)..

Quando o *strive* for menor, mais informação ele conseguirá extrair do *input*, o que também significa maiores *outputs layers* e a possibilidade de modelos mais complexos. Maiores *strides* acabam sendo mais limitados para a tarefa de extração de *features* (CNN EXPLAINER, 2020)..

# 6 DESENVOLVIMENTO DO PROJETO

Este capítulo irá apresentar o desenvolvimento do trabalho, que consiste na construção de 4 modelos de *machine learning*, utilizando 4 estruturas de *neural networks*, o treinamento dos modelos e a comparação dos resultados. O projeto tem como objetivo testar a viabilidade da utilização de *machine learning* no diagnóstico do câncer de mama, melhorando a precisão, velocidade e acesso de classes mais baixas ao diagnóstico da doença.

**Figura 23: Diagrama de atividades do projeto**

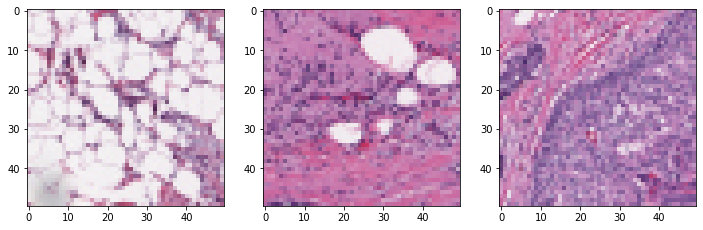
****

Fonte: Imagem gerada pelo Autor

**6.1 *DATASET* DE IMAGENS**

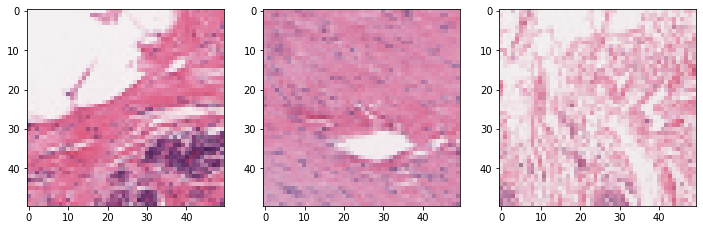
Para o treinamento e testes dos modelos é utilizado um *dataset* de imagens disponível no *site Kaggle*. O *dataset* consistia em 162 *slides* de imagens de Câncer de mama escaneados em 40x. Desses *slides*, foram retiradas 277.524 imagens. Cada imagem tem o *shape* de (50, 50, 3). O que significa que possuem 50X50 *pixels* e cada *pixel* possui 3 canais de cor (*RGB*). As imagens estão divididas em duas classes. A classe 1 indica a existência de *IDC* e a classe 0 indica a não existência de *IDC.*

**Figura 24: 3 imagens da classe 1 escolhidas aleatoriamente**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**Figura 25: 3 imagens da classe 0 escolhidas aleatoriamente**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

Para o treinamento, as imagens foram divididas em duas pastas. A primeira pasta chamada *train*, contendo 80% das imagens divididas em duas classes. Classe 0 e classe 1. A outra pasta chamada *test* possui 20% das imagens contidas no *dataset* e também foi dividida em duas classes. As imagens da pasta *train* foram utilizadas para validar a capacidade de generalização do modelo durante a etapa de treinamento.

Para a criação dos tensores que alimentaram os modelos, foi utilizada a classe *ImageDataGenerator*. A classe, que pertence ao *Keras*, fornece ferramentas para aplicar pré-processamento de imagens, *data augmentation* e a criação de *batches* para o modelo.

**6.2 *BATCH SIZE***

Esta sessão irá discutir e mostrar o processo para definir o tamanho do *batch* utilizado durante o treinamento dos modelos. O *batch size* é utilizado para se referir ao número de exemplos ao qual o modelo será exposto a cada iteração durante o processo de aprendizagem (DEEP LEARNING BOOK, 2021).

Em *machine learning*, o *batch size* é divido em 3 sub-categorias diferentes, de acordo com o seu tamanho. *Batch mode* é utilizado para classificar *batchs* com o tamanho igual a quantidade de dados disponíveis para o treinamento, ou seja, quando não há divisão em lotes dos dados. *Mini-batch mode* é a classificação dada a *batchs* quando o seu tamanho é maior que um, mas menor que a quantidade de dados disponíveis. Neste caso, geralmente, o tamanho do *batch* é definido por um divisor do conjunto total de dados. Por último, o *stochastic mode* é quando o tamanho do *batch* é igual a um (DEEP LEARNING BOOK, 2021).

A definição do *batch size* é um dos hiperparâmetros mais importantes durante a implementação de um algoritmo de *deep learning*. Apesar das *GPU*'s atuais permitirem a utilização de *batchs* maiores, eles se mostram deficientes quando se fala de generalização do aprendizado. Por sua vez, a escolha de um *batch size* menor costuma demonstrar uma convergência mais rápida para soluções boas. Isso pode ser explicado pelo fato de que os *batchs* menores permitem ao modelo o aprendizado antes de que ele seja exposto à totalidade de dados (DEEP LEARNING BOOK, 2021).

Através de testes foi observado que a escolha de *batch sizes* menores, além de proporcionar uma dinâmica de treinamento mais rápido, quando comparada com *batchs sizes* maiores, também demonstra uma generalização mais precisa. Já a escolha do *stochastic mode* para o *batch size* demonstra péssimo desempenho (DEEP LEARNING BOOK, 2021).

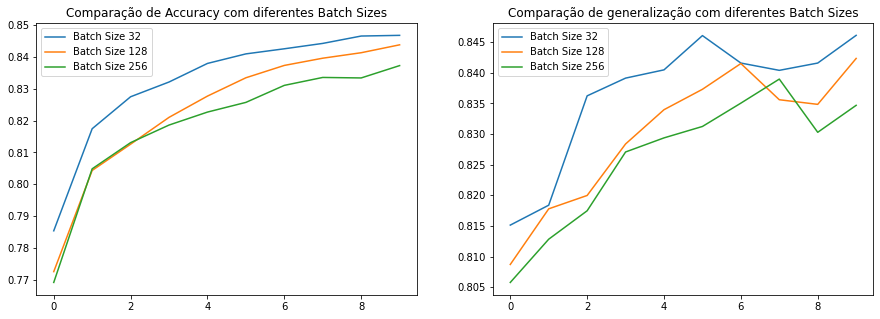
O *Mini-batch mode* é aceito como o intervalo ideal para o *batch* que fornecerá a melhor generalização quando exposto aos dados de testes. Para encontrar exatamente qual o tamanho ideal para o *batch*, deve-se levar em consideração o tipo de dados e o modelo a qual eles serão expostos(DEEP LEARNING BOOK, 2021).

Para evitar *overfitting*, *batch sizes* menores expõe representações ruidosas de todo o conjunto de dados, e dessa forma, causa uma espécie de dinâmica que evita que a *neural network* se ajuste muito bem ao conjunto de dados de treinamento, assim, fornecendo uma generalização pobre quando expostas ao *dataset* de testes (DEEP LEARNING BOOK, 2021).

Para a definição do *batch size* utilizado para o treinamento dos modelos deste trabalho, foram definidos *datasets* de treinamento e de testes com 3 tamanhos diferentes e expostos a um modelo. Os testes para a definição foram feitos com o *batch size* de 32, 128 e 256. Os resultados são mostrados na figura 27.

Os testes foram feitos em cima do primeiro modelo desenvolvido, o *Mini VGG.* Os dados não sofreram nenhuma etapa de pré-processamento, apenas a normalização das imagens. Depois de treinar o modelo 3 vezes, uma para cada tamanho de *batch,* os resultados foram apresentados em plots com a utilização da biblioteca *MatPlotLib.*

**Figura 26: Resultados dos testes para definir o *batch size***

******

Fonte: Imagem gerada pelo autor utilizando o Colab

O primeiro gráfico mostra a *accuracy* do modelo depois do treinamento e o segundo gráfico a capacidade de generalização do modelo quando exposto a dados não conhecidos. O *Batch size* 32 demonstrou melhor desempenho tanto do treinamento, alcançando maior precisão, quanto na validação, conseguindo generalizar melhor o aprendizado alcançado com o *dataset* de treino. Com base nisso, para o treinamento dos modelos deste trabalho, foi utilizado um *batch size* de 32.

**6.3 AMBIENTE DE DESENVOLVIMENTO**

Para o desenvolvimento deste trabalho foi utilizado o *Google Collaboratory*. O *Google Collaboratory* oferece um serviço de armazenamento em nuvens em formato de *notebooks*. Popularmente conhecido como *colab*, ele é focado na criação e execução de códigos em *Python* (KENZIE, 2021).

Os códigos são executados na nuvem diretamente pelo navegador, tornando o desenvolvimento em *machine learning* mais acessível e dispensando qualquer tipo de instalação e configuração na máquina local (KENZIE, 2021).

*Notebooks* são agrupamentos em células de códigos, e neles é possível ler, desenvolver e rodar códigos de maneira interativa, compartilhá-los com outros desenvolvedores, editar e mantê-los salvos na nuvem (KENZIE, 2021)..

O *Google* *Collaboratory* trabalha em parceria com o *Jupyter Notebook*, que é um *software* de código aberto e que disponibiliza um ambiente similar ao *colab* para desenvolvimento de células de código (KENZIE, 2021)..

A grande diferença entre o *colab* e o *Jupyter Notebook* está na instalação local para ser possível utilizar as ferramentas e recursos que o *Jupyter* oferece. Por se tratar de um serviço na nuvem, o colab oferece *hardware* de ponta com *GPU's* com aceleração para o treinamento eficiente dos modelos (KENZIE, 2021)..

**Quadro 1: Configurações de *hardware* e *software* disponíveis pelo *Colab***

| **Componentes** | **Descrição** |
| --- | --- |
| CPU | 1xsingle core hyper threaded Xeon Processors @2.3Ghz i.e(1 core, 2 threads) |
| GPU | 1xTesla K80 , compute 3.7, having 2496 CUDA cores , 12GB GDDR5 VRAM |
| RAM | 12.6 GB Available |
| Disk | 33 GB Available |
| Python Version | 3.6 |
| TensorFlow Version | 2.8.2 |

Fonte: Google Collaboratory

**6.3.1 *TENSORFLOW***

Para a criação dos modelos foi utilizado o *Tensorflow*. O *Tensorflow* é uma biblioteca amplamente utilizada para a implementação de modelos de *machine learning*. Essa ferramenta foi desenvolvida pelo *Google* como parte do projeto que foi chamado de *Google Brain* e mais tarde foi disponibilizado para a comunidade de engenheiros de *machine learning* como uma biblioteca *open source* (SINGH *et al*., 2020).

Por conta da disponibilidade *open source*, o *Tensorflow* passou a ser amplamente utilizado pela comunidade de *machine learning* para o desenvolvimento de ferramentas e produtos. Isso não apenas ajudou desenvolvedores com a criação de modelos de *machine* e *deep* *learning*, como também permitiu a comunidade implementar versões customizadas para uma grande variedade de propósitos. De negócios a pesquisas na área de *AI* (SINGH *et al*., 2020).

O *Tensorflow* é utilizado pelo *Google* em muitos dos seus produtos, como o *Google maps* e o *gmail*, por exemplo. Isso incentivou desenvolvedores a utilizar a biblioteca para uma infinidade de aplicativos. Logo, o *Tensorflow* virou uma das mais populares bibliotecas de *machine learning* (SINGH *et al*., 2020).

Apesar de tudo, no começo a ferramenta tinha algumas limitações, o que atrasou um pouco a adoção, quando comparada com outras bibliotecas como o *PyTorch*, *Theano* e o *OpenCV*. Por ouvir a comunidade, a equipe responsável pelo desenvolvimento do *Tensorflow* passou a trabalhar em melhorias na ferramenta com o objetivo de tornar o *Tensorflow* mais efetivo e simples de se trabalhar. Como resultado, foi lançado o *Tensorflow* 2.0 (SINGH *et al*., 2020).

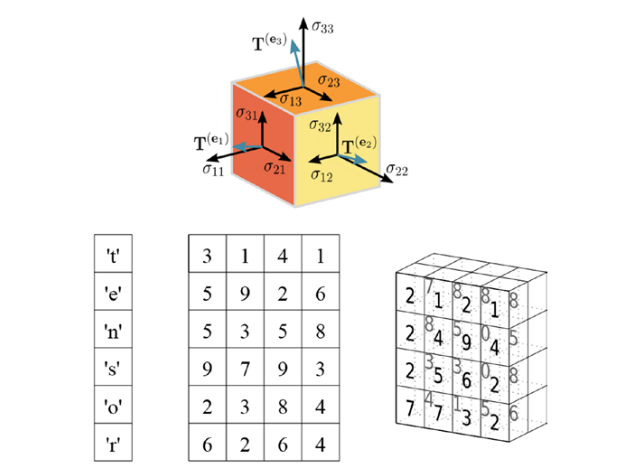
Tensores são a alma do *Tensorflow*. Toda a computação é feita utilizando tensores. De acordo com o *Google’s TensorFlow team*, um tensor é uma generalização de vetores e matrizes para múltiplas dimensões. Um tensor também pode ser definido como dados em forma de números (SINGH *et al*., 2020).

Basicamente, qualquer coisa que pode ser representada desta forma pode ser utilizada para o treinamento de *neural networks*. Internamente, o *Tensorflow* representa tensores como um *nd-Array*, formato disponibilizado pela ferramenta *Numpy* (SINGH *et al*., 2020).

Um vetor é um conceito matemático. Geralmente ele representa algo que tem magnitude e direção. Colocado de forma simples, é uma coleção de valores ordenados. Um *Array*. Sem direção, um tensor se transforma em um valor que é definido como *Scalar*, com apenas magnitude. Um vetor é flexível e pode ser utilizado para representar uma grande variedade de coisas (SINGH *et a*l., 2020).

Um tensor também é um conceito matemático, e assim como um *Scalar*, um vetor e matrizes, ele pode ser utilizado para representar muitas coisas. Em resumo, um tensor é um *array* multidimensional com propriedades dinâmicas (SINGH *et al*., 2020). A figura 27 demonstra a relação entre tensores, *arrays* e matrizes.

**Figura 27: Tensor, Array e Matriz**

****

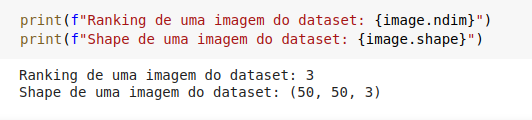
Fonte: Learn TensorFlow 2.0: Implement Machine Learning and Deep Learning

Models with Python by SINGH et al, 2020

*Ranking* é uma propriedade dos tensores. O *rank* do tensor serve para indicar as direções que descrevem um objeto. Também pode ser definido pela quantidade de dimensões que um tensor tem. Um *Scalar*, por não ter nenhuma direção tem *rank* 0. Um vetor, que é definido como uma coleção de dados com uma direção, tem o *rank* 1. Uma matriz é definida como *rank* 2. As imagens utilizadas para o treinamento do modelo possuem *ranking* 3 (SINGH *et al.*, 2020).

O *Shape* de um tensor é a propriedade utilizada para descrever a quantidade de elementos que cada dimensão do tensor tem. Um vetor, ou um tensor com apenas uma dimensão com os valores [5,6,7], teria um *shape* igual a (3). As imagens utilizadas neste trabalho possuem um *shape* igual a (50, 50, 3). 50X50 *pixels* e a última dimensão se refere aos canais de cor *RGB* (SINGH *et al*., 2020). A figura 28 demonstra a utilização dos atributos *tensor.ndim* e *tensor.shape* para verificação do *Ranking* e do Shape das imagens utilizadas durante o treinamento dos modelos.

**Figura 28: Shape e ranking de uma imagem do dataset de treinamento**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

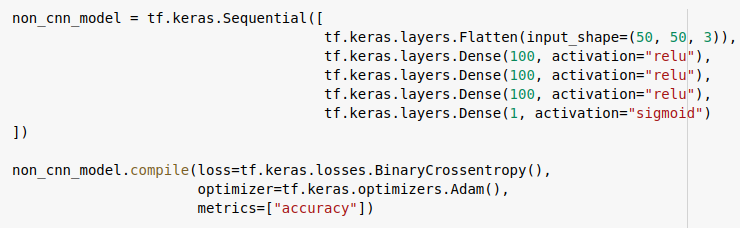
**6.3.2 *KERAS***

Em conjunto com o *Tensorflow*, a criação dos modelos utilizou o *Keras*, uma biblioteca para *machine learning* com uma comunidade de mais de 300.000 desenvolvedores que cresce cerca de 35% ao ano (ATIENZA, 2020).

A biblioteca do *Google*, *Tensorflow*, utiliza o *Keras* como uma *API* de alto nível. Geralmente o *keras* é utilizado com o comando *tf.keras*. Graças a essa integração de alto nível entre o *Tensorflow* e o *Keras*, ela se tornou uma escolha popular para o desenvolvimento de projetos de *Deep learning* (ATIENZA, 2020).

Na biblioteca do *tf.keras*, as *layers* são conectadas umas às outras como um tipo de quebra-cabeça. O resultado disso é um modelo limpo e fácil de compreender. O treinamento também se torna simples, exigindo apenas os dados, a quantidade de *epochs* e o método de validação. Com isso, é possível criar, treinar e validar um modelo com algumas poucas linhas de código, como mostra a figura 29 (ATIENZA, 2020).

**Figura 29: Exemplo de modelo utilizando keras integrado com Tensorflow**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

Como resultado, modelos criados utilizando o *Keras* podem ser implementados utilizando uma quantidade significativamente menor do que outras bibliotecas muito populares como o *PyTorch*. Isso gera um aumento de produtividade pelo tempo salvo durante a etapa de implementação de código (ATIENZA, 2020).

**6.4 PRÉ-PROCESSAMENTO**

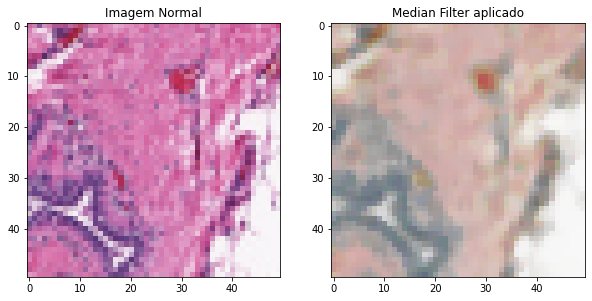
Esta sessão vai falar sobre a etapa de processamento de imagens e as técnicas utilizadas para preparar as imagens do *dataset* para o treinamento, com o objetivo de suavizar as imagens e fornecer uma maior variedade de dados, a fim de extrair o máximo possível de informação e alcançar maior precisão durante o treinamento das *Artificial neural networks*.

**6.4.1 *MEDIAN FILTER***

O *median filter* é um dos filtros mais comuns e básicos quando se fala em processamento digital de imagens. Com o objetivo de suavizar imagens, se trata de um filtro não linear que remove ruídos preto e branco presentes na imagem, encontrando a média utilizando os *pixels* vizinhos (SOLOMON *et al*., 2011).

Para utilizar o *median filter*, nós pegamos a primeira 3x3 *matrix*. Calculamos a mediana dos valores contidos na *matrix* e substituímos o valor central pelo valor da mediana. O próximo passo é andar um *pixel* para o lado e repetir o processo até que todos os *pixels* da imagem sejam cobertos (SOLOMON *et al*., 2011). A figura 30 demonstra o exemplo de *median filter* aplicado a uma imagem aleatória do dataset utilizado neste trabalho.

**Figura 30: Exemplo de *median filter* usado no *dataset* do projeto**

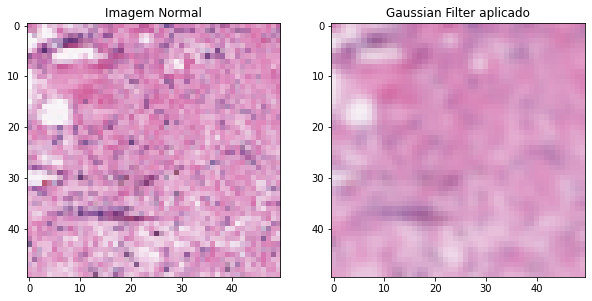
****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**6.4.2 *GAUSSIAN FILTER***

Outra técnica comum em processamento digital de imagens é a do *gaussian filter*. O *gaussian filter* é um filtro de passa-baixa onde é usada a função gaussiana para obter o resultado desejado. O objetivo do *gaussian filter* é diminuir ruídos na imagem e aplicar *blurring* em regiões dessa imagem (SOLOMON *et al*., 2011). A figura 31 demonstra o exemplo de *gaussian filter* aplicado a uma imagem aleatória do dataset utilizado neste trabalho.

**Figura 31: Exemplo do *Gaussian filter* aplicado no *dataset***

****

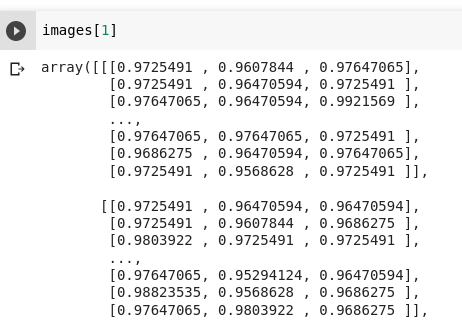
Fonte: Imagem gerada pelo Autor utilizando o Colab

**6.4.3 NORMALIZAÇÃO**

O objetivo de normalizar os dados é fazer com que eles estejam no mesmo padrão das *activations functions,* que normalmente estão entre 0-1. Isso ajuda a diminuir a frequência de gradientes nulos e aumentam a velocidade com que as *neural networks* aprendem (INSIDE MACHINE LEARNING, 2021).

Na figura 32 temos um exemplo de um *n-dimentional array* com os valores *RGB* normalizados. O *n-dimentional array* representa uma imagem aleatória do *dataset* utilizado no treinamento dos modelos deste projeto. Cada linha do *array* representa um *pixel* e os três valores representam o *RGB* normalizado (0-1) deste *pixel*.

**Figura 32: *N-dimentional Array* normalizado**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**6.4.4 *DATA AUGMENTATION***

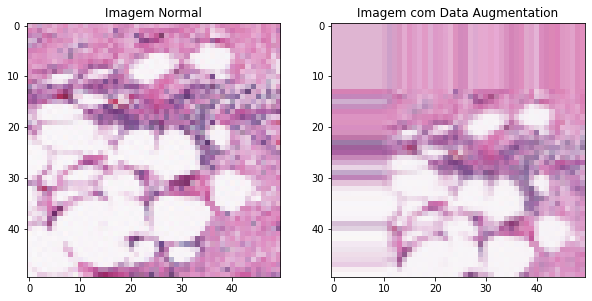
O objetivo de aplicar *data augmentation* em um *dataset* de imagens é aumentar a variedade de dados sem coletar novos dados. Aplicando modificações aleatórias no *dataset* é possível gerar novas informações. Com *data augmentation* o modelo é exposto a uma variedade maior de dados e a aspectos diferentes, isso ajuda a melhorar a generalização do aprendizado (PAPER, 2021).

Esses resultados são alcançados realizando transformações aleatórias nas imagens do *dataset*. Exemplos de transformações são *horizontal* e *vertical* *flipping*, *skewing*, *cropping*, *shearing*, *zooming* e *rotating*. Dessa forma, os dados são capazes de simular diferentes *data points* e fornecer mais informação sobre as imagens ao modelo de aprendizado (PAPER, 2021).

*Data augmentation* também é frequentemente utilizado como uma opção para resolver o problema de *overfitting*. Isso pode acontecer quando a quantidade de dados que o modelo é exposto é limitada (PAPER, 2021).

Na Figura 33 temos a comparação de uma imagem retirada do *dataset* apenas normalizado com a mesma imagem depois de passar pelo processo de *data augmentation*.

**Figura 33: Exemplo de data *augmentation***

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**6.5 ESTRUTURA**

Esta sessão irá abordar os elementos utilizados na construção dos modelos do projeto. Estruturas que compõe as *Layers* utilizadas, *activation function*, *loss*, *optimizer* e *Sequential* API.

**6.5.1 SEQUENTIAL API**

Para a construção dos modelos foi utilizada a *Sequential API*. A S*equential API* permite que o desenvolvedor crie modelo *layer-by-layer* e sua flexibilidade oferece solução para uma grande variedade de problemas (MACHINE LEARNING MASTERY, 2017).

Para criar uma instância da classe, é utilizada a linha *tf.keras.Sequential()*. Depois de instanciada, podemos adicionar *layers* ao modelo utilizando o método *add()*. A primeira *layer* é definida como sendo a *input layer*, as *layers* intermediárias são as *hidden layers* e a última, a *output layer* (HULL, 2021).

**6.5.2 LAYERS**

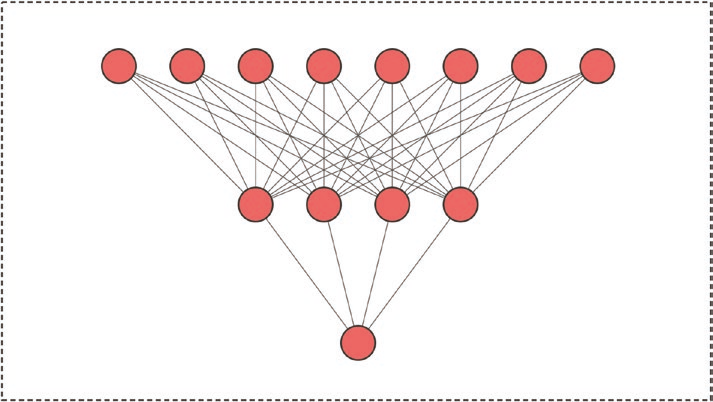
Em um modelo de *machine learning*, *layers* são a parte da estrutura do modelo considerada como a arquitetura. Existem vários tipos de *layers* que podem ser utilizados e cada tipo oferece características diferentes para tipos de problemas diferentes. Na construção dos modelos as *layers* utilizadas foram *Dense*, *Conv2D*, *MaxPool2D* e *Flatten* (ANALYSTICS INDIA MAGAZINE, 2021).

**6.5.2.1 DENSE LAYER**

*Dense layers* também são conhecidas como *fully connected layers (Figura 34)*. Muitos modelos utilizam as *Dense layers* como camadas de *output*. A *Dense layer* é profundamente conectada com a sua *layer* precedente, o que significa que todos os *neurons* ou *hidden units,* são conectados com todos os *neuros* da *layer* precedente (ANALYSTICS INDIA MAGAZINE, 2021).

Cada *hidden unit* na *layer* recebe o *output* gerado nos *neurons* da *layer* anterior, depois de receber o *output*, o *neuron* produz *dot multiplication*. *Dot multiplication* é a operação onde a linha da *matriz* do *output* é igual a coluna da *matriz* da *dense layer*. Para isso, os dois precisam ser do mesmo tamanho (ANALYSTICS INDIA MAGAZINE, 2021).

**Figura 34: Exemplo de um modelo *Fully connected layer***

****

Fonte: Machine Learning for Economics and Finance in TensorFlow 2 by HULL, 2021

**6.5.2.2 CONVOLUTIONAL LAYER**

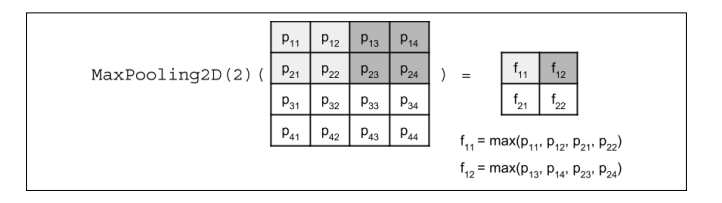
A *Convolutional layer* é o elemento mais fundamental das *CNNs*. Ela contém *weights* que extraem *features* das imagens utilizadas no treinamento do modelo. Para isso, elas utilizam um *set* de filtros que fazem uso do valor de *input* para produzir um *output feature map* (CNN EXPLAINER, 2020).

Com o *output* recebido pela camada anterior, o *neuron* da *Convolutional* *layer* executa o *dot product* elemento por elemento utilizando um *Kernel*. O *convolutional* *neuron* é o resultado de todos os valores intermediários somados com o seu *bias*. Para produzir o output de um convolutional *neuron*, é performado um *dot product* elemento por elemento entre o *output* recebido da camada anterior e o *Kernel* de *weights* aprendidos pela *neural network*(CNN EXPLAINER, 2020).

**6.5.2.3 POOLING LAYER**

Existem vários tipos de *layers* que são chamadas de *pooling*, mas o objetivo de todas elas é o mesmo, apesar de tentar alcançá-lo de formas diferentes. O objetivo é diminuir a extensão espacial da *network*, e com isso diminuir os parâmetros e a quantidade de operações computacionais executadas pelo modelo, como exemplifica a figura 35. Na construção desse modelo foram usadas 2 *pooling layers*. *MaxPooling2D* e *AvaragePooling2D* (CNN EXPLAINER, 2020).

**Figura 35: Exemplo de pooling operation**

****

Fonte: Advanced Deep Learning with TensorFlow 2 and Keras by ATIENZA, 2020

A *pooling layer* recebe 2 parâmetros durante a sua atribuição. O *kernel size* e o *stride length.* Durante o treinamento, a operação desliza o *kernel* utilizando o *stride* especificado sobre o *input*. Durante esse processo, a *MaxPooling* seleciona somente o maior valor do *Kernel*, com uma operação de *Max()*, enquanto a *AvarageMaxPooling* retira a média do *Kernel* (CNN EXPLAINER, 2020).

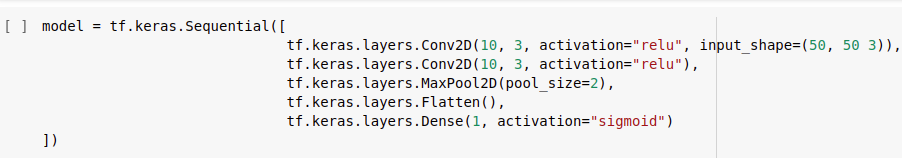
Para a construção dos modelos, foi utilizado o *Kernel* 2x2 e o *stride* de 1, que são valores padrões para esses tipos de *layers*. Esta operação descarta cerca de 75% das ativações e portanto, é computacionalmente mais eficiente.

**6.5.2.4 FLATTEN LAYER**

A função da *Flatten layer* é converter a camada tridimensional na *network* em um vetor de uma única dimensão, para que ele consiga se conectar com a *Dense* *layer* de *output* para executar a classificação. Usando os dados do projeto como exemplo, que ṕossuem um *shape* de (50, 50, 3), seria convertido pela *Flatten layer* em um vetor com tamanho de 7500 (CNN EXPLAINER, 2020).

As *Convolutional layers* são responsáveis pela extração de *features* do *input* recebido, mas para executar a classificação, este *input* é passado para uma *Dense* *layer* (CNN EXPLAINER, 2020). A classificação no problema proposto neste trabalho é chamada de classificação binária. Temos duas classes, IDC(+) e IDC(-). Para a classificação binária utilizamos a *activation function Sigmoid*. Para isso, o uso de uma *Flatter layer* é necessária, como mostra a figura 36, já que esta *activation function* exige um *input* de uma única dimensão.

**Figura 36: Exemplo de um modelo utilizando uma *Flatten layer***

******

Fonte: Imagem gerada pelo Autor utilizando o Colab

**6.5.3 ACTIVATION FUNCTION**

As *activation functions* desempenham uma função fundamental durante o aprendizado das *artificial neural networks*. A função delas é decidir se um *neuron* deve ser ativado ou não. Em outras palavras, a função de uma *activation function* é decidir se a informação que ela está recebendo é importante para o aprendizado ou se deve ser ignorada (DEEP LEARNING BOOK, 2021).

A *activation function* executa uma transformação não linear ao longo do sinal de *input*. Depois de transformada, ela é então enviada para a próxima *layer* de *neurons* como *input*. Quando isso não acontece, os *weights* e o *bias* sofrem apenas uma transformação linear (DEEP LEARNING BOOK, 2021)..

O problema na ausência de uma *activation function*, está na capacidade limitada de aprendizado da equação linear. Um modelo de *neural networks* sem a *activation function* acaba tendo a mesma capacidade de aprendizado de uma regressão linear (DEEP LEARNING BOOK, 2021)..

A *activation function* faz a transformação não linear nos dados de *input*, e com isso o modelo ganha a habilidade de resolver problemas complexos. Exemplos de problemas que podem ser resolvidos com o uso de uma simples *activation function* são o processamento de linguagens naturais e a classificação de imagens (DEEP LEARNING BOOK, 2021).

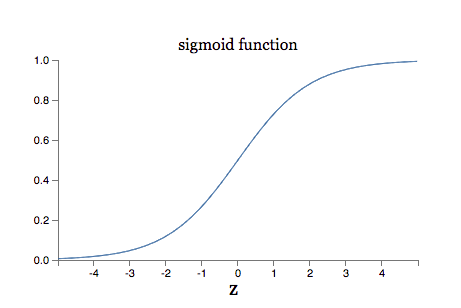
Não existe apenas um tipo de *activation function*. Existem vários e devem ser escolhidas de acordo com o problema que se busca solucionar. Para o desenvolvimento dos modelos, a função *Relu* e a função *sigmoid* foram utilizadas (DEEP LEARNING BOOK, 2021).

**6.5.3.1 SIGMOID FUNCTION**

A *activation function sigmoid* é amplamente utilizada para problemas de classificação binária. Sua forma é: f (x) = 1 / (1 + e-x)

É considerada uma função suave e é continuamente diferenciável. Como dito anteriormente, a maior vantagem no uso de *activation function* está na não linearidade. Isso significa essencialmente que quanto a *activation function sigmoid* é utilizada ela produz uma saída não linear, que varia entre 0 e 1 e tem o formato de S, como pode ser visto na figura 37 (DEEP LEARNING BOOK, 2021).

**Figura 37: *Sigmoid Function***

******

Fonte: Advanced Deep Learning with TensorFlow 2 and Keras by ATIENZA, 2020

A *activation function sigmoid* tenta mandar o valor de *output* para um dos extremos. Essa habilidade é desejável quando queremos classificar objetos em duas classes: 0 ou 1 (DEEP LEARNING BOOK, 2021)..

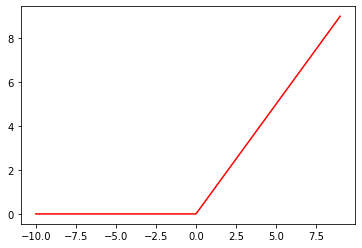
Apesar de ser uma função extremamente popular, esta *activation function* pode apresentar problemas durante o treinamento. Se os gradientes se tornam muito pequenos e se aproximam de 0, isso significa que a rede não está conseguindo aprender nenhum padrão nos dados apresentados a ela (DEEP LEARNING BOOK, 2021).

**6.5.3.2 RELU FUNCTION**

A *activation function Relu* é definida por: *f(x) = max (0, x)*. Durante o desenvolvimento dos modelos, a *activation function* mais utilizada foi a *Relu*. Também é a *activation funcition* mais utilizada em *neural networks*. É uma função não linear e pode ser usada durante a etapa de treinamento para ativar camadas de *neurons* (DEEP LEARNING BOOK, 2021)..

A principal vantagem desta *function* e o que torna ela popular em relação a outras é que ela não ativa todos os *neurons* ao mesmo tempo. Isso significa que quando o dado de entrada for negativo, ela o transforma em 0 e o neuron não é ativado. Este comportamento é demonstrado na figura 38. Isso fornece robustez e eficiência, quando apenas alguns neurons são ativados a *network* reduz a quantidade de processamento computacional (DEEP LEARNING BOOK, 2021)..

**Figura 38: *Relu Function***

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**6.6 COMPILE**

O método *model\_name.compile()* é chamado na etapa de *pre-built* do modelo e nele são especificadas as funções de *loss*, *optimizer* e o *metrics* para a validação do modelo durante o treinamento. Para os modelos em questão, que são modelos que executam a classificação binária, a *loss function* escolhida foi a *BinaryCrossentropy* do *keras*, a *otimizer function* escolhida foi a *Adam*, também pertencente ao *keras*. E como *metrics* foi escolhida *accuracy* (MACHINE LEARNING MASTERY, 2017).

Saber exatamente os valores dos *weights* ideais para uma *neural network* é uma tarefa impossível para uma pessoa. São muitas as possibilidades e fatores desconhecidos para a execução desta tarefa. Para isto, um algoritmo é usado para explorar as possibilidades e tentar alcançar um desempenho suficientemente bom (MACHINE LEARNING MASTERY, 2017).

Se um modelo é treinado utilizando o método *stochastic gradient descent* como *optimizer* e os *weights* e *bias* são calculados através de *backpropagation* como *loss*. O modelo utiliza os *weights* para fazer predições e o *loss* é calculado pela qualidade dessas precisões (MACHINE LEARNING MASTERY, 2017).

A função do método *stochastic* *gradient* *descent*, neste caso, é mudar os *weights* para que a próxima validação encontre um *loss* ainda menor. Isso significa que o modelo está aprendendo (MACHINE LEARNING MASTERY, 2017).

A função da *loss function* é representar todos os aspectos do modelo em um único número de forma que este número sirva de base e guie melhorias no aprendizado do modelo. Este único número possibilita *weigths* candidatos sejam rankeados e comparados (MACHINE LEARNING MASTERY, 2017).

Para calcular o *loss* durante o aprendizado, uma função adequada para o tipo de problema que se busca resolver deve ser escolhida. Para o problema em questão, a escolhida foi a *BinaryCrossentropy()*, uma *loss function* especializada em problemas de classificação binária (MACHINE LEARNING MASTERY, 2017).

A *cross-entropy loss* é também conhecida como “l*ogarithmic loss*” ou “*logistic loss”*. Ela funciona comparando cada predição do modelo durante o treinamento com o a sua *label*: 0 ou 1. Com isso, um *score* é calculado e a predição feita pelo modelo será penalizada de acordo com esse *score* (MACHINE LEARNING MASTERY, 2017).

*Cross-entropy loss* é uma função de minimizar. Isso significa que quanto menor o valor calculado pelo *loss*, melhor as predições que o modelo executa durante as etapas de treinamento e validação. Uma predição perfeita possui o loss igual a 0.0. Para problemas de classificação binária, a *cross-entropy loss* calcula a média entre todos os exemplos (MACHINE LEARNING MASTERY, 2017).

O modelo busca minimizar o *loss* enquanto é exposto aos dados. Se o *loss* demonstra o quão bem o modelo está aprendendo, e um *loss* maior significa que o modelo está performando mal, minimizar o *loss* é o caminho para melhorar a qualidade das predições feitas por ele (MACHINE LEARNING MASTERY, 2017).

Para o treinamento, uma parte chave é revisar os parâmetros matemáticos de cada *neurons* da *network* baseado no quanto esses parâmetros estão corretos na classificação das imagens. Durante a *backpropagation*, as *artificial neural networks* usam um método matemático chamado *gradient descent* para atualizar esses parâmetros matemáticos, de forma a melhorar o desempenho das classificações feitas pela *network* (MACHINE LEARNING MASTERY, 2017).

No método *model\_name.compile()* o parâmetro *optimizer* deve ser definido. O papel desse parâmetro é fazer com que a *backpropagation* seja executada com maior velocidade e de forma mais eficiente. Para os modelos construídos neste trabalho, foi usado o *optimazer* "*Adam*". *Adam* é uma escolha comum para o aperfeiçoamento da *network* durante o aprendizado (TOWARDS DATASCIENCE, 2019).

O nome *Adam* vem de *Adaptive Moment Estimation*. Este algoritmo de otimização é baseado no *stochastic gradient descent* e a sua função é atualizar os *weights* da melhor maneira durante o aprendizado. Algoritmos de otimização como *SGD* mantém um único *learning rate* durante todo o treinamento. O *Adam* difere por atualizar o *learning rate* para cada *weight* individualmente (TOWARDS DATASCIENCE, 2019).

Outros algoritmos serviram de inspiração para o *Adam*. *Adagrad* e *RMS Prop* são exemplos desses algoritmos. Apesar de inspirado, o *Adam* oferece algumas melhorias em relação a esses algoritmos. Por exemplo, no *RMS Prop* o *learning rate* é definido com base no primeiro *moment*, enquanto o *Adam* também utiliza o segundo *moment* para tentar encontrar o melhor *learning rate*(TOWARDS DATASCIENCE, 2019)*.*

Quanto se fala em eficiência, o *Adam* oferece uma série de benefícios. Motivo pelo qual ele é amplamente utilizado na construção de modelos e é aconselhado como *optimizer function* padrão para o treinamento de *artificial neural networks*. Essas vantagens estão na sua facilidade de implementação, velocidade no tempo de execução, baixos requisitos no uso de memória *RAM* e geralmente necessita de menos ajustes para um bom aprendizado(TOWARDS DATASCIENCE, 2019).

Apesar de todas as vantagens e do aconselhamento para o uso do *Adam*, essa *optimizer function* também oferece desvantagens. O *Adam* foca em ter um melhor desempenho quando se fala em tempo de execução computacional, enquanto outros *optimizers* como o *SGD*, focam em *data points.* Como resultado de uma otimização focada em *data points*, modelos que utilizam o *SGD* demonstram uma melhor generalização do aprendizado quando validados no *dataset* de teste. Nos testes feitos neste trabalho, o uso do *Adam* atingiu uma boa generalização(TOWARDS DATASCIENCE, 2019).

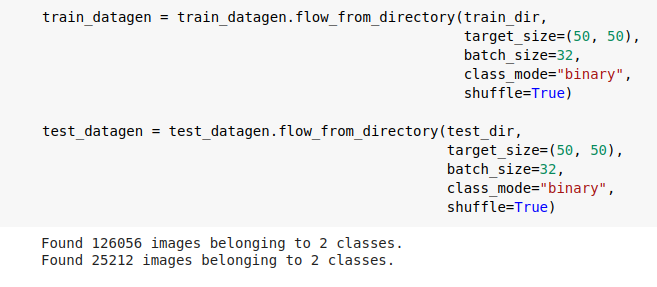
O último parâmetro que o método *model\_name.compile()* recebe é o *metrics*. Ele se refere ao método que será usado para executar a validação do treinamento do modelo. O *metrics* deve ser usado com cuidado, pois nem sempre reflete com precisão o que aconteceu durante o aprendizado(TOWARDS DATASCIENCE, 2019).

Para a etapa de validação dos modelos a métrica utilizada se chama *Accuracy*. *Accuracy* representa a porcentagem de predições corretas que o modelo faz durante o treinamento e durante a etapa de validação. Durante o treinamento a *accuracy* é representada pelo atributo *accuracy*. Durante a etapa de teste, o atributo *val\_accuracy* representa o quão bem o modelo generalizou o aprendizado para o *dataset* de teste(TOWARDS DATASCIENCE, 2019).

Durante a etapa de preparação foi observado que o dataset continha 70% da totalidade dos dados de testes pertencentes a uma única classe. Se o modelo chutasse apenas esta classe, atingiria o *accuarry* de 70%, mesmo sem aprender nenhum padrão das imagens. Para evitar o problema, foi feita uma correção de desequilíbrio e as classes deixadas do mesmo tamanho (TOWARDS DATASCIENCE, 2019).

A figura 39 demonstra a criação dos tensores de treino e de teste utilizados para o treinamento de validação dos modelos, utilizando a classe *ImageDataGenerator* e a divisão dos dados nos 2 *dataset*.

**Figura 39: Criação dos tensores de treinamento e testes com quantidade de amostras de cada *dataset*.**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

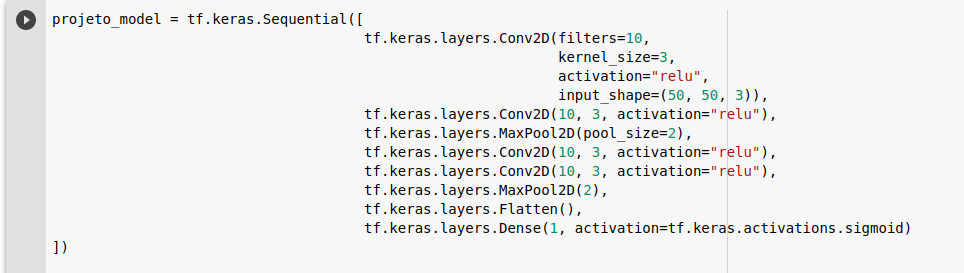
**6.7 CONSTRUÇÃO DOS MODELOS**

Essa sessão irá abordar a construção dos modelos utilizados neste trabalho. Foram construídos 6 modelos e cada modelo foi treinado 4 vezes. Em cada etapa de treinamento técnicas de pré-processamento de imagens foram aplicadas ao dataset, com o objetivo de melhorar o processo de extração de *features* e por consequência atingir maior precisão nas predição performadas pelos modelos.

O primeiro modelo proposto foi uma versão simplificada de estruturas já bem conhecidas e utilizadas para o reconhecimento e classificação de imagens. O modelo foi construído com base na arquitetura *VGG16*, que foi proposta por K. Simonyan and A. Zisserman em um artigo publicado em 2015.

Por motivos de legibilidade foi optado por definir as *layers* durante a criação do modelo utilizando a *Sequential API*, como pode ser visto na figura 40. O modelo consiste em combinações de blocos de *layers* interligados com *Dense Layers* por uma *Flatten Layer*. Cada bloco é formado por 2 *Convolutional layers* e uma *MaxPooling layers*.

**Figura 40: Criação do primeiro modelo proposto**

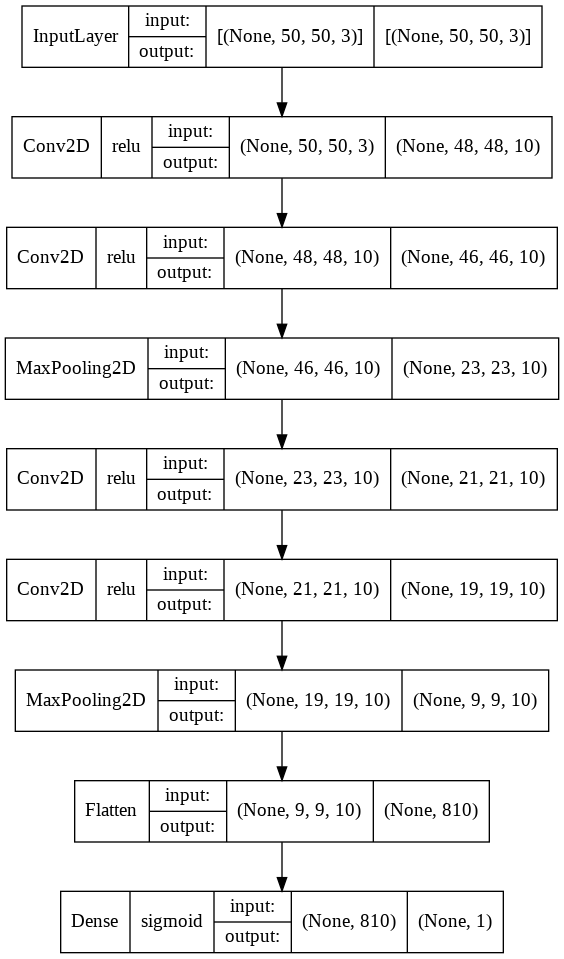
****

Fonte: Imagem gerada pelo Autor utilizando o Colab

Para a definição dos hiperparâmetros das *Convolutions layers* foram definidos *filters* de 10, *strides* de 1 e para a matriz do *Kernel* o tamanho 3. Para todas as funções de ativação das *hidden layers* foi utilizada a *activation function Relu*. Para as camadas *MaxPooling* o hiperparâmetro *pool\_size* foi definido como 2.

As duas últimas *layers* foram formadas por uma *Flatten Layer* e uma *output* *layer*. A *Flatten layer* produz como *output* um *array* de uma única dimensão, necessário para a output executar a perdição. A camada de *output* é formada por um único *neuron* que produz uma saída entre 0 e 1 e pela *activation function Sigmoid.* A figura 41 demonstra o plot do modelo.

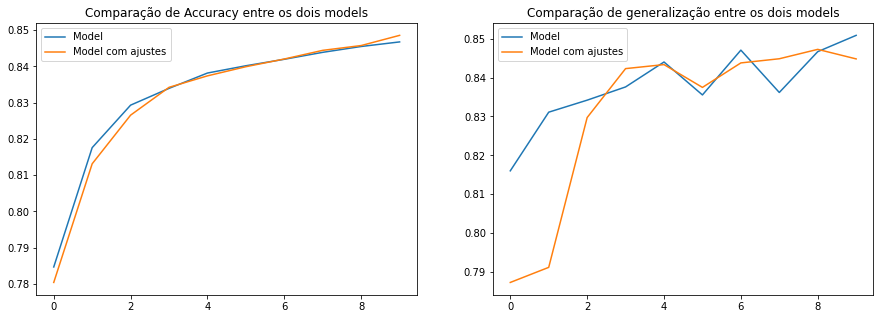
**Figura 41: *Plot* do modelo**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

Em busca de melhorar o desempenho do primeiro modelo proposto, ajustes foram feitos nos hiperparâmetros das *Convolutional layers*. Com o aumento nas quantidades dos *filters* de 10 para 32 e do *Kernel\_size* de 3 para 4, o modelo apresentou um leve ganho de precisão sem grande aumento na sua complexidade e nem no tempo de treinamento. Porém a generalização do aprendizado em cima do *dataset* de teste demonstrou um resultado inferior, como mostra a figura 43.

**Figura 42: Comparação dos *models* com ajustes**

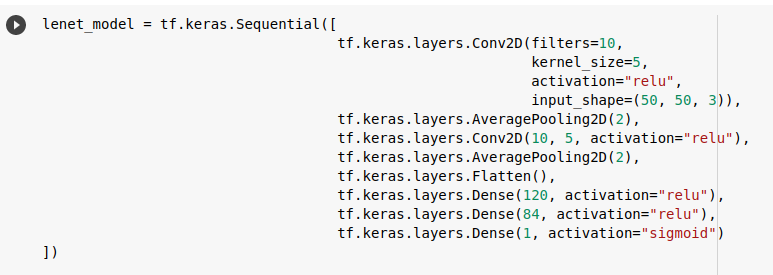
****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**6.7.1 *LETNET MODEL***

O segundo modelo proposto foi construído em cima da arquitetura *CNN LeNet* e édemonstrado na figura 43. *LeNet* foi um dos primeiros modelos de *CNN* propostos. Criado em 1998, apresentava performance no estado da arte para o reconhecimento e classificação de dígitos escritos à mão. É conhecido por ser um modelo pequeno. O modelo criado neste trabalho não chegou nem a 200.000 parâmetros de aprendizado. Diferentemente dos modelos utilizados na época, *LeNet* aprendia por encontrar relações entre os pixels e seus vizinhos (Jha *et al*., 2021).

**Figura 43 : Criação do modelo com base a arquitetura *LeNet***

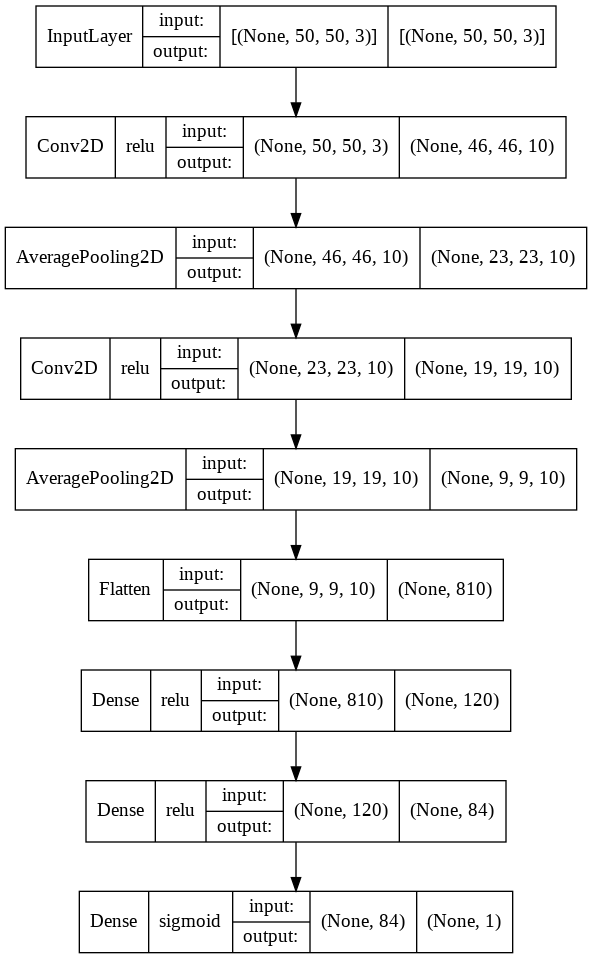
******

Fonte: Imagem gerada pelo Autor utilizando o Colab

O modelo *LeNet* foi criado utilizando a *Sequential API*. Para as *layers* foram usados 2 blocos *Convolutional* e um bloco final de *fully-connected layers*. Para cada bloco *Convolutional*, foi feita uma combinação de uma *Convolutional Layer* com uma *pooling layer*. Como *Convolutional layer* foi utilizada a *Conv2D* e seus hiperparâmetros foram definidos com 10 *filters*, *stride* de 1 e *kernel\_size* de 5. A *activation function* utilizada foi a *Relu*. Para a *pooling layer* foi utilizada a *AveragePooling2D* com *pool\_size de 2* (Jha *et al*., 2021).

Para fazer a ligação entre os blocos *Convolutional* e o bloco *fully-conneted* foi utilizada a *Flatten layer*. O bloco *fully-conneted* foi criado com uma combinação de três *Dense layers*. a Primeira *Dense layer* com 120 *hidden units* e *activation function Relu*. A segunda *Dense layer* com 84 *hidden units* e também fazendo de *activation* *function* *Relu*. E como *output layer*, uma *Dense layer* com apenas uma *hidden unit,* usando *sigmoid function* para classificação binária (Jha *et al*., 2021). A estrutura do modelo é apresentada pelo *plot* na figura 44.

**Figura 44: *Plot* do modelo *LeNet***

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**6.7.2 *RESNET MODEL***

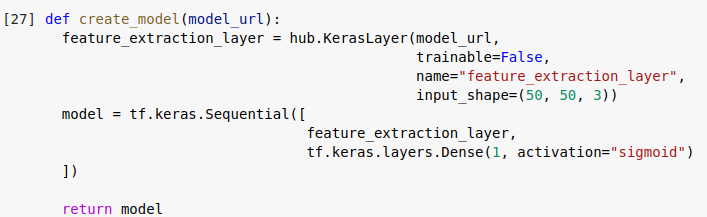
O terceiro modelo proposto neste trabalho foi construído em cima da arquitetura *ResNet* e utilizou o *Tensorflow* *Hub* para a sua construção. *ResNet*, que significa *Residual Network*, foi proposta por Shaoqing Ren *et al.* em 2015. O modelo *Resnet* já é considerado um dos mais bem sucedidos modelos para classificação de imagens e se tornou muito popular nos últimos anos (ANALYTICS VINDHYA, 2021).

Com o passar do tempo e a criação de *neural networks* cada vez mais profundas, um problema passou a ser notado durante o treinamento dessas *networks*. *Networks* mais profundas apresentavam *loss* maior em relação a estruturas não tão profundas. Esse problema era causado por *overftting* e a sua causa era por *exploding gradient* (ANALYTICS VINDHYA, 2021)..

Em ordem de resolver o problema de *exploding* *gradient*, a *ResNet* introduziu o conceito de *Residual Networks*. Para isso, uma técnica chamada *Skip Connections* era usada em conjunto com a arquitetura *ResNet*. *Skip Connections* consiste em pular algumas *layers* durante o treinamento e ir direto para a *layer* de *output* (ANALYTICS VINDHYA, 2021)..

Para a criação deste modelo foi criada uma função chamada *create\_model* (Figura 45)e ela recebe a *url* do modelo que será importado. A função cria uma instância da classe *hub.kerasLayer* utilizando a *url* do modelo escolhido. Depois de criada, a instância dessa classe é utilizada na *Sequencial API* para a criação do modelo. Por fim, a função retorna o modelo criado.

**Figura 45: Função *create\_model***

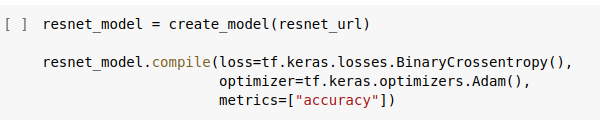
****

Fonte: Imagem gerada pelo Autor utilizando o Colab

O *Tensorflow* *Hub* foi criado para oferecer um repositório de modelos de *machine learning* reutilizáveis criados com *Tensorflow*. Ele provê modelos pré-treinados que podem ser usados para resolver novos problemas com maior eficiência. Maior eficiência não apenas no tempo de treinamento, mas também na etapa de desenvolvimento (TENSORFLOW HUB, 2020).

Para criar o modelo, a função *create\_model* foi chamada utilizando a *url* do modelo escolhido. O classificador escolhido do repositório oferecido pelo *Tensorflow* *Hub* foi o *ResNet\_V2\_50*. 50 faz referência a quantidade de *layers* utilizadas na construção desse modelo. A figura 46 ilustra a facilidade que a criação de um classificador utilizando o *Tensorflow*.

**Figura 46: Criação do modelo *ResNet***

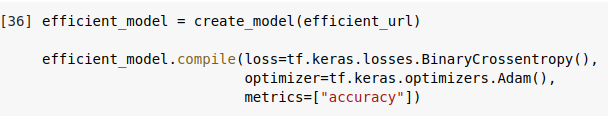
******

Fonte: Imagem gerada pelo Autor utilizando o Colab

**6.7.3 *EFFICIENTNET MODEL***

O quarto modelo proposto neste trabalho foi construído em cima da arquitetura *EfficientNet* e também utilizou o repositório disponibilizado pelo *Tensorflow Hub* para a criação de forma rápida e eficiente do modelo. A criação do modelo é demonstrada na figura 47.

**Figura 47: Criação do modelo com a função create\_model**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

Assim como o modelo anterior, o classificador foi criado utilizando a função *create\_model* implementada pelo autor durante a etapa de desenvolvimento. O modelo escolhido no repositório disponibilizado pelo *Tensorflow Hub*. A escolha para utilização neste trabalho foi o *efficientnet\_b0*.

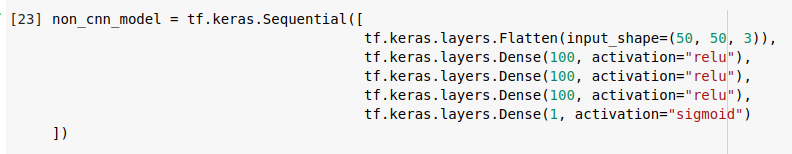
*EfficientNet* é uma estrutura de *CNN*, mas também um método para escalar, que escala todas as dimensões do modelo de maneira uniforme. Ele escala largura, profundidade e resolução através da utilização de um coeficiente composto. Enquanto outras estruturas escalam os modelos de forma arbitrária, o *EfficientNet* utiliza um coeficiente de escala fixo, e como resultado, o modelo é escalado de forma uniforme (PAPER WITH CODE, 2020).

O método composto de escala vem da intuição de que imagens maiores necessitam de *networks* mais robustas, com mais *layers* para aumentar o campo receptivo e mais canais para capturar uma quantidade maior de *features* nas imagens, e como consequência disso, um aprendizado mais refinado (PAPER WITH CODE, 2020).

**6.7.4 *NON CNN MODEL***

Como última proposta de modelo para este trabalho, foi criado um modelo que não utiliza nenhum *Convolution layer* e nem *pooling layers* (Figura 48). O objetivo da proposta é comparar com o desempenho das *CNN's*, medir o quão bem *CNN’s* performam a tarefa de classificação de imagens em relação a um modelo ANN comum, oferecer uma variedade maior de testes e uma maior confiabilidade de resultados.

**Figura 48: Modelo *ANN***

****

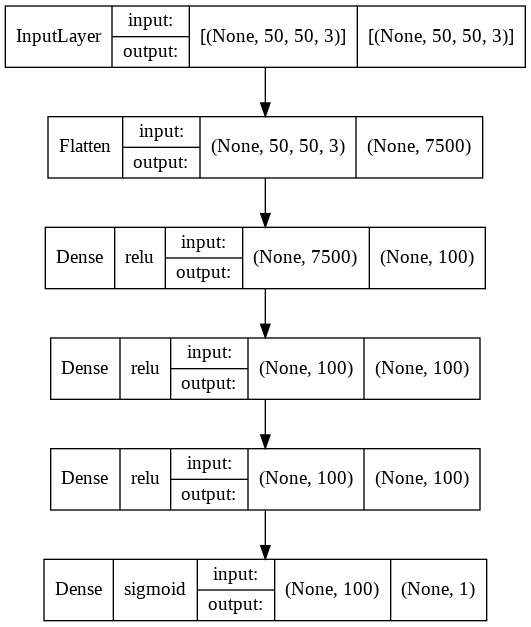
Fonte: Imagem gerada pelo Autor utilizando o Colab

Este modelo foi criado utilizando a *Sequential API*. Consiste na combinação de *fully-conneted layers*. Como *layer* de *input* foi utilizada uma *Flatten layer*. A *Flatten layer* recebeu um único hiperparâmetro chamado *input\_shape*, com o *shape* das imagens utilizadas no treinamento dos modelos. O *shape* de (50, 50, 3). A função dela é transformar o *tensor* que representa a imagem em um único *array* de tamanho 7500, de forma que se torne compatível com o *input* recebido pela camada seguinte.

Depois disso, o modelo é seguido por três *Dense layers*. Cada *Dense layer* recebe dois hiperparâmetros. O primeiro hiperparâmetro define a quantidade de *hidden units* da camada. Para as três *Dense layers* o valor escolhido foi de cem *hidden units*. Esse valor foi escolhido através de testes. Os testes visaram aproximar o desempenho das *CNN's*, mas sem acrescentar muito no tempo de execução durante o treinamento do modelo. O segundo hiperparâmetro é a definição da *activation function*. Para este modelo a *activation function* escolhida foi a *Relu*.

Por fim, temos a camada de *output*. Ela é formada por uma única camada com uma única *hidden unit*. Por ser um problema de classificação binária, a última camada precisa ter uma única *hidden unit*. Para classificação binária, a *activation function* ideal é a *Sigmoid*. Como saída o modelo retorna um único valor entre 0 e 1. Tudo o que foi descrito é demonstrado no plot do modelo na figura 49.

**Figura 49: *Plot* do modelo *Non CNN***

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**7 TESTES E RESULTADOS**

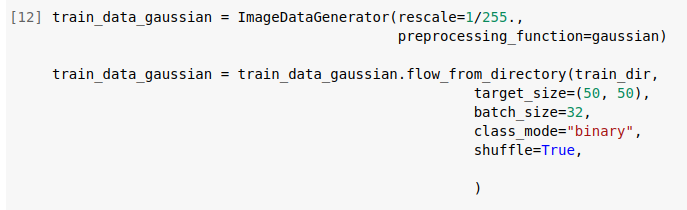
Este capítulo irá abordar os testes feitos com cada modelo desenvolvido durante este trabalho e os resultados alcançados por cada um deles. O objetivo é comparar a performance de cada classificador na tarefa de detectar o câncer de mama em imagens obtidas através do exame de Histopatologia e analisar a viabilidade da utilização do uso de modelos *machine learning* para tal tarefa.

Os testes contaram com 4 etapas de treinamento. Para cada uma das etapas, os *datasets* expostos aos modelos passaram por técnicas diferentes de pré-processamento de imagens, com o objetivo de suavizar, gerar diversidade de dados e quantidade de informação e *features*. O objetivo foi tentar alcançar melhores resultados, maior precisão de classificação e melhorar a generalização de aprendizado durante a validação dos modelos no *dataset* de teste.

Para a aplicação dos filtros de suavizamento de imagens, foi utilizado a classe *skimage*. Esta classe pertence ao *Scikit* e oferece uma coleção de algoritmos para processamento digital de imagens disponíveis gratuitamente. Da classe *skimage.filters,* foram importadas apenas os algoritmos de pŕe-processamento de imagem *gaussian* e *median*.

Para a aplicação do filtro, os algoritmos importados foram passados como parametros durante a criação do *ImageDataGenerator*. Depois disso, o *dataset* é criado utilizando o método *flow\_from\_directory* e o filtro de pré-processamento é aplicado ao *dataset* de treinamento, conforme mostra a imagem 50.

**Figura 50: Aplicação do filtro e criação do *dataset* de treino**

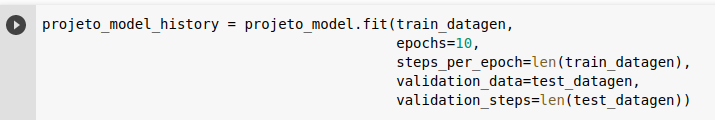


Fonte: Imagem gerada pelo Autor utilizando o Colab

O método *flow\_from\_directory* também recebe outros parâmetros. O diretório das imagens que serão usadas no treinamento, a resolução das imagens do *dataset*, o tamanho do *batch*, que foi definido como 32, o tipo de classificação que será performada pelo classificador e o parâmetro booleano *shuffle*, que define se as instâncias das classes do *dataset* serão ou não embaralhadas.

Depois da criação dos modelos como foi descrito no capítulo anterior e da preparação dos dataset de treino e teste, vem a etapa de treinamento dos classificadores. Para isso, depois de criar e compilar o modelo, basta chamar o método *model\_name.fit(),* como demonstra a figura 51.

**Figura 51: Chamada do método *fit()***

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

O primeiro parâmetro utilizado para o *fit* do modelo foi o *dataset* de treino, gerado pelo *ImageDataGenerator*. O *ImageDataGenerator* rotula todas as imagens de acordo com a pasta em que elas estão localizadas. O segundo parâmetro é a quantidade de *epochs* executadas durante o treinamento. O *epoch* é uma iteração executada em cima de todo o *dataset*. Durante os testes, alguns modelos não mostraram muitos ganhos na *accuracy* com *epochs* acima de 10. Por conta de limitações na quantidade diária de uso disponível da *GPU* no *Colab*, 10 *epochs* se mostrou uma escolha eficiente (TENSORFLOW, 2022).

O terceiro se chama *steps\_per\_epoch*. Esse parâmetro indica a quantidade de *batchs* que cada *epochs* visualiza antes de terminar e passar para a seguinte. O valor passado é a quantidade total de *batchs* do *dataset*, de modo que todo o conjunto de dados seja visualizado a cada *epoch* (TENSORFLOW, 2022).

Os dois últimos parâmetros fazem referência ao processo de validação do modelo. Foi optado por fazer a validação ao mesmo tempo do treinamento. Para isso, o parâmetro *validation\_data* recebe o *dataset* de testes gerado pelo *ImageDataGenerator*. Dessa forma, ao final de cada *epoch*, o treinamento performa a validação do modelo (TENSORFLOW, 2022).

O último parâmetro é similar ao *steps\_per\_epochs*. Recebe a quantidade de *batchs* visualizados ao final de cada *epoch*, para performar a validação do modelo. O valor passado foi a quantidade total de *batchs* no *dataset* de teste (TENSORFLOW, 2022).

Depois do treinamento de todos os modelos, os resultados foram coletados e apresentados em cada uma das tabelas a seguir. O quadro 2 contém os resultados dos treinamentos com os dados apenas normalizados e com *data augmentation* aplicado às imagens do *dataset*. O quadro 3 contém os resultados dos filtros de suavização, *median filter* e *gaussian filter*.

**Quadro 2: Primeira quadro com apresentação de resultados**

|  | | | | **Normalizados** | | | | **Augmentation** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Modelos** | | | | **ACC** | **LOSS** | **V\_LOSS** | **V\_ACC** | **ACC** | **LOSS** | **V\_LOSS** | **V\_ACC** |
| Mini VGG | | | | 84.58% | 36.10% | 36.92% | 83.91% | 83.92% | 37.41% | 37.71% | 83.85% |
| LeNet | | | | 85.11% | 34.95% | 39.27% | 83.23% | 82.79% | 39.63% | 38.63% | 83.26% |
| ResNet | | | | 78.98% | 47.46% | 47.44% | 79.42% | 77.61% | 50.05% | 50.08% | 77.42% |
| EfficientNet | | | | 81.76% | 41.06% | 40.90% | 81.81% | 79.84% | 44.69% | 44.01% | 79.80% |
| Non CNN | | | | 78.47% | 47.46% | 49.93% | 77.95% | 78.08% | 48.51% | 47.33% | 79.01% |

Fonte: Tabela criada pelo Autor

**Quadro 3: Segundo quadro com apresentação de resultados**

|  | | | | **Median** | | | | **Gaussian** | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Modelos** | | | | **ACC** | **LOSS** | **V\_LOSS** | **V\_ACC** | **ACC** | **LOSS** | **V\_LOSS** | **V\_ACC** |
| Mini VGG | | | | 83.29% | 38.69% | 45.18% | 79.79% | 84.07% | 37.10% | 40.12% | 82.61% |
| LeNet | | | | 81.92% | 40.73% | 53.03% | 74.98% | 83.23% | 38.32% | 41.35% | 81.83% |
| ResNet | | | | 80.84% | 44.30% | 55.34% | 75.25% | 79.68% | 46.23% | 51.08% | 77.30% |
| EfficientNet | | | | 83.14% | 38.77% | 74.07% | 63.82% | 81.19% | 42.17% | 49.46% | 77.19% |
| Non CNN | | | | 78.37% | 47.81% | 59.31% | 70.85% | 78.42% | 47.82% | 48.26% | 78.57% |

Fonte: Tabela criada pelo Autor

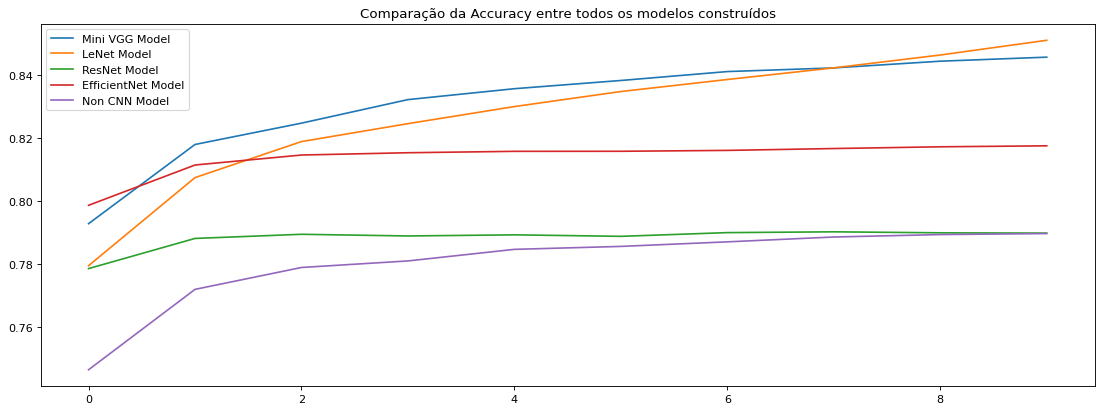
Conforme os resultados nas tabelas mostram, o modelo que melhor desempenhou a função de detectar câncer de mama nas imagens geradas pelo exame Histopatológico, durante o treinamento, foi o modelo *LeNet*. Porém, o modelo *Mini VGG* atingiu resultado semelhante, mas obteve melhor resultado para a generalização do aprendizado em todos os testes feitos. Como a performance durante a generalização é o mais desejável, então o modelo Mini VGG demonstrou o melhor resultado.

Junto com o modelo *ResNet*, o modelo *Non Cnn* teve os piores resultados em todas as métricas durante todos os testes feitos neste trabalho. O modelo *efficientNet* mostou resultado levemente inferior quando comparado com os modelos *LeNet* e *Mini VGG*.

Os melhores resultados foram alcançados com o treinamento dos dados apenas normalizados. Os testes utilizando o *Gaussian filter* no modelo *Mini VGG* atingiram bons resultados durante a etapa de treinamento, mas houve uma queda na generalização. O *Median filter* além de não fornecer bons resultados, ainda aumentou consideravelmente o tempo de treinamento, se mostrando ineficiente.

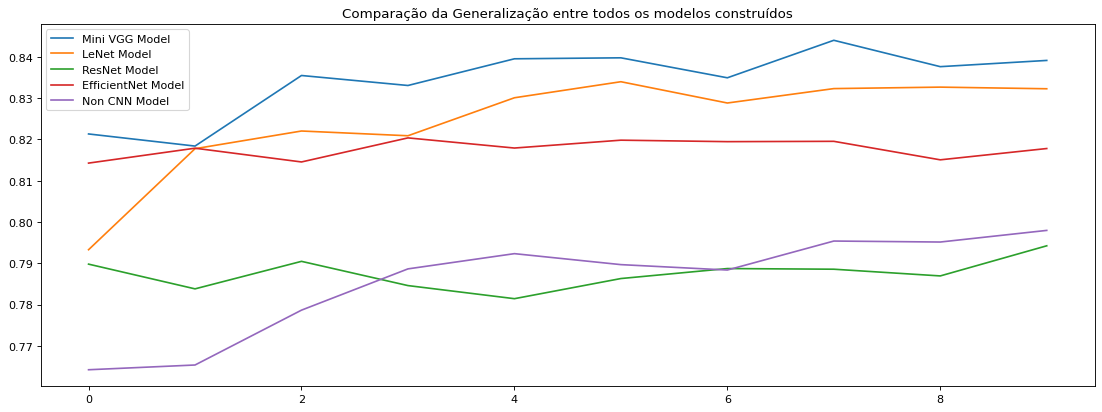
Todos os modelos atingiram melhores resultados durante o treinamento e na generalização do aprendizado quando treinados com o *dataset* apenas normalizado. As Figuras 53 e 54 demonstram os gráficos da diferença de performance entre os modelos treinados apenas com os dados normalizados. A Figura 52 demonstra a *accuracy* de treinamento e a figura 53 a generalização do aprendizado. Levando em consideração a generalização, o modelo com melhor desempenho foi o *Mini VGG*, como os gráficos demonstram.

**Figura 52: Comparação de *accuracy* durante o treinamento**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**Figura 53: Comparação da generalização do aprendizado**

****

Fonte: Imagem gerada pelo Autor utilizando o Colab

**8 CONCLUSÃO**

Este estudo buscou propor a construção de modelos de *machine learning* para fazer a detecção de câncer de mama utilizando um *dataset* com imagens obtidas através do exame de Histopatologia, para o aprendizado das *Artificial neural networks*.

O câncer de mama hoje, é uma das principais causas da morte de mulheres no mundo. A desigualdade e a falta de acesso a exames dificultam o diagnóstico da doença, tornando-a ainda mais mortal entre as classes mais baixas.

Por isso, este estudo visou, com o treinamento das *neural networks,* demonstrar a viabilidade do uso de tal tecnologia para o diagnóstico da doença, oferecendo dessa forma uma opção barata e rápida e tornando o diagnóstico da doença mais acessível.

O melhor resultado obtido pelo treinamento dos modelos foi de 85% de precisão durante a etapa de treinamento e 84% de precisão quando testada a capacidade de generalização do modelo em dados não conhecidos.

A utilização de filtros e outras técnicas de pré-processamento, para este caso, não se mostraram vantajosas. A maioria dos modelos demonstrou performance inferior menos com os dados pré-processados, atingiu uma generalização pior e o tempo de treinamento dos modelos aumentou consideravelmente.

Newman-Toker *et al*., (2020) mostrou em seu estudo que o erro no diagnóstico do câncer de mama varia de 8.9% até 26.3%. No Brasil, segundo o Oncofisio (2011), até 15% dos pacientes chegam ao hospital com o diagnóstico errado.

Baseando-se nos resultados encontrados e dados apresentados, chega-se à conclusão que a tecnologia *machine learning* já é capaz de oferecer bons percentuais de precisão para o diagnóstico da doença.

Durante o desenvolvimento do trabalho, foi observada a dificuldade do treinamento de *networks* mais profundas devido a limitações no uso diário da *GPU* disponibilizada pelo *Colab.* Isto também refletiu na escolha da quantidade de *epochs* utilizadas durante o treinamento.

Alguns modelos desenvolvidos demonstraram potencial para melhores resultados com um treinamento prolongado. Como sugestão para pesquisas futuras, é recomendado o desenvolvimento e testes de *networks* mais profundas, a fim de aproveitar ao máximo o desempenho dessas redes para o problema proposto neste trabalho.

# REFERÊNCIAS BIBLIOGRÁFICAS

**A Complete Understanding of Dense Layers in Neural Networks**, Analytics India mag 19 de Set. de 2021. Disponível em:

<https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/> Acesso em: mar de 2022

ALPAYDIN, E. Título: **MACHINE LEARNING: THE NEW AI** Local de publicação: Massachusetts Institute of Technology, 2016

AMUEL, A. L. **Some Studies in Machine Learning Using the Game of Checkers** 1959. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5392560>

ATIENZA, R. Título: **TENSORFLOW 2 AND KERAS SECOND EDITION** Local de publicação: Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2020

BAESENS, B. Título: **ANALYTICS IN A BIG DATA WORLD** Local de publicação: John Wiley & Sons, Inc., Hoboken, New Jersey 2014

**Beginner’s Guide to Building Neural Networks in TensorFlow**, [Towards Data Science](https://towardsdatascience.com/?source=post_page-----dab7a09b941d--------------------------------) 05 de Jun. de 2019, Disponível em:

<https://towardsdatascience.com/beginners-guide-to-building-neural-networks-in-tensorflow-dab7a09b941d/> Acesso em: mar de 2022

BRAGA, A. de P; CARVALHO, A. P. de L. F; LUDERMIR, T. B. Título: **REDES NEURAIS ARTIFICIAIS: TEORIA E APLICAÇÕES** Local de publicação: LIVROS TÉCNICOS E CIENTÍFICOS EDITORA S.A. Travessa do Ouvidor, 11 Rio de Janeiro, RJ - CEP 20040-040, 2000

**Build ResNet from Scratch With Python!,** Analytics vindhya 07 de Jun. de 2021. Disponível em:

<https://www.analyticsvidhya.com/blog/2021/06/build-resnet-from-scratch-with-python/> Acesso em: mar de 2022

**Biópsia da mama.** Oncoguia, 05 de out. de 2014, Disponível em: <http://www.oncoguia.org.br/conteudo/biopsia-da-mama/1390/264/> Acesso em: mar de 2021

**Câncer de mama.** Institudo Nacional do Câncer, Disponível em: <https://www.inca.gov.br/assuntos/cancer-de-mama/> Acesso em: mar de 2021

**Câncer de Mama Receptor de Hormônio**. Oncoguia, 20 de jun. de 2017, Disponível em: <http://www.oncoguia.org.br/conteudo/cancer-de-mama-receptor-de-hormonio/10879/264/> Acesso em: mar de 2021

**Câncer de mama - versão para Profissionais de Saúde**. Instituto Nacional do Câncer, 29 de mar. de 2021 Disponível em: <https://www.inca.gov.br/tipos-de-cancer/cancer-de-mama/profissional-de-saude> Acesso em: mar de 2021

**Câncer de Mama HER2**, Oncoguia 24 de Jul. de 2020, Disponível em:

<http://www.oncoguia.org.br/conteudo/cancer-de-mama-her2/10880/264/> Acesso em: mar de 2021

**Câncer de Mama Inflamatório**, Oncoguia 24 de Jul. de 2020, Disponível em:

<http://www.oncoguia.org.br/conteudo/cancer-de-mama-inflamatorio/1386/34/> Acesso em: mar de 2022

**Câncer de Mama Triplo-negativo**, Oncoguia 24 de Jul. de 2020, Disponível em:

<http://www.oncoguia.org.br/conteudo/cancer-de-mama-triplonegativo/13793/34/> Acesso em: mar de 2022

**Cancer Staging**, Nacional Cancer Institute 09 de Mar. de 2015, Disponível em:

<https://www.cancer.gov/about-cancer/diagnosis-staging/staging> Acesso em: mar de 2021

**Cirurgia para Câncer de Mama**. Oncoguia, 10 de 0ut. de 2014, Disponível em:

<http://www.oncoguia.org.br/conteudo/cirurgia-para-cancer-de-mama/1397/265/> Acesso em: mar de 2021

**Como surge o câncer?** Instituto Nacional do Câncer, 03 de mar. de 2019 Disponível em: <https://www.inca.gov.br/como-surge-o-cancer/> Acesso em: mar de 2021

**Conceito e Magnitude**. Instituto Nacional do Câncer, 25 de abr. de 2021 Disponível em: <https://www.inca.gov.br/controle-do-cancer-de-mama/conceito-e-magnitude> Acesso em: abr de 2021

**Detecção precoce**, Instituto Nacional do Câncer 27 de Mai. de 2021, Disponível em:

<https://www.inca.gov.br/controle-do-cancer-de-mama/acoes-de-controle/deteccao-precoce> Acesso em: mar de 2021

**Estadiamento do câncer**, Oncoguia 06 de Jun. de 2015, Disponível em:

<http://www.oncoguia.org.br/conteudo/estadiamento/4795/1/> Acesso em: mar de 2022

**Estatísticas de câncer**, Instituto Nacional do Câncer 10 de Jun. de 2021, Disponível em:

<https://www.inca.gov.br/numeros-de-cancer> Acesso em: mar de 2021

**Função de Ativação**, Deep learning book s.d. Disponível em:

<https://www.deeplearningbook.com.br/funcao-de-ativacao/> Acesso em: mar de 2022

GÉRON, A. Título: **HANDS ON MACHINE LEARNING WITH SCIKIT-LEARN, KERAS AND TENSORFLOW** Local de publicação: O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2019

**GOOGLE COLAB**, kenzie 19 de Nov. de 2020 Disponível em:

<https://kenzie.com.br/blog/google-colab/> Acesso em: mar de 2022

GRAUPE, D. Título: **PRINCIPLES OF ARTIFICIAL NEURAL NETWORKS** Local de publicação: World Scientific Publishing Co. Pte. Ltd. 5 Toh Tuck Link, Singapore 596224, 2013

HAYKIN, S. Título: **NEURAL NETWORKS AND LEARNING MACHINES THIRD EDITION** Local de publicação: McMaster University Hamilton, Ontario, Canada

2009

**Hormonioterapia para Câncer de Mama**, Oncoguia 27 de Jul. de 2020, Disponível em:

<http://www.oncoguia.org.br/conteudo/hormonioterapia-para-cancer-de-mama/1404/265/> Acesso em: mar de 2021

HULL, I. Título: **MACHINE LEARNING FOR ECONOMICS** Local de publicação: Nacka, Sweden, 2021

**How to Use the Keras Functional API for Deep Learning**, Machine learning mastery 27 de Out. de 2017. Disponível em:

<https://machinelearningmastery.com/keras-functional-api-deep-learning/> Acesso em: mar de 2022

INSTITUTO NACIONAL DE CÂNCER JOSÈ ALENCAR GOMES DA SILVA. **ABC do Câncer - Abordagens Básicas para o Controle do Câncer.** Disponível em:<https://www.inca.gov.br/sites/ufu.sti.inca.local/files//media/document//livro-abc-6-edicao-2020.pdf>

INSTITUTO NACIONAL DE CÂNCER JOSÈ ALENCAR GOMES DA SILVA. **Ambiente, Trabalho e Câncer: Aspectos Epidemiológicos, Toxicológicos e Regulatórios.** Disponível em**:** <https://www.inca.gov.br/sites/ufu.sti.inca.local/files//media/document//ambiente\_trabalho\_e\_cancer\_-\_aspectos\_epidemiologicos\_toxicologicos\_e\_regulatorios.pdf>

**EfficientNet**, Papers with code s.d. Disponível em:

<https://paperswithcode.com/method/efficientnet#:~:text=EfficientNet%20is%20a%20convolutional%20neural,resolution%20using%20a%20compound%20coefficient.> Acesso em: mar de 2022

INSTITUTO NACIONAL DE CÂNCER JOSÈ ALENCAR GOMES DA SILVA. **Estimativa 2020**. Disponível em: <https://www.inca.gov.br/sites/ufu.sti.inca.local/files//media/document//estimativa-2020-incidencia-de-cancer-no-brasil.pdf>

INSTITUTO NACIONAL DE CÂNCER JOSÈ ALENCAR GOMES DA SILVA. **Dieta, Nutrição, Atividade Física e Câncer: Uma Perspectiva Globa**l. Disponível em: <https://www.inca.gov.br/sites/ufu.sti.inca.local/files//media/document//dieta\_nutricao\_atividade\_fisica\_e\_cancer\_resumo\_do\_terceiro\_relatorio\_de\_especialistas\_com\_uma\_perspectiva\_brasileira.pdf>

INSTITUTO NACIONAL DE CÂNCER JOSÈ ALENCAR GOMES DA SILVA. **Diretrizes para a Detecção Precoce do Câncer de Mama no Brasil**. Disponível em: <https://www.inca.gov.br/sites/ufu.sti.inca.local/files//media/document//diretrizes\_deteccao\_precoce\_cancer\_mama\_brasil.pdf>

INSTITUTO NACIONAL DE CÂNCER JOSÈ ALENCAR GOMES DA SILVA. **ATUALIZAÇÃO EM MAMOGRAFIA 2019**. Disponível em: <https://www.inca.gov.br/sites/ufu.sti.inca.local/files//media/document//2a\_edicao\_atualizacao\_em\_mamografia\_para\_tecnicos\_em\_radiologia\_2019.pdf>

INSTITUTO NACIONAL DE CÂNCER JOSÈ ALENCAR GOMES DA SILVA. **INCIDÊNCIA, MORTALIDADE E MORBIDADE HOSPITALAR POR CÂNCER**

**EM CRIANÇAS, ADOLESCENTES E ADULTOS JOVENS NO BRASIL:**

**INFORMAÇÕES DOS REGISTROS DE CÂNCER E DO SISTEMA DE MORTALIDADE**. Disponível em: <https://www.inca.gov.br/sites/ufu.sti.inca.local/files//media/document//incidencia-mortalidade-morbidade-hospitalar-por-cancer.pdf>

INSTITUTO NACIONAL DE CÂNCER JOSÈ ALENCAR GOMES DA SILVA. **CÂNCER DE MAMA: é preciso falar disso.** Disponível em:

<http://bvsms.saude.gov.br/bvs/publicacoes/cancer\_mama\_preciso\_falar\_disso.pdf>

KELLEHER, J. D. et al. Título: **FUNDAMENTALS OF MACHINE LEARNING FOR PREDICTIVE DATA ANALYTICS** Local de publicação: The MIT Press Cambridge, Massachusetts London, England 10017 2015

KHAN, S. el al. Título: **A GUIDE TO CONVOLUTION NEURAL NETWORKS** Local de publicação: Morgan & Claypool 2018

**Loss and Loss Functions for Training Deep Learning Neural Networks**, Machine learning mastery 05 de Jun. de 2019, Disponível em:

<https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/> Acesso em: mar de 2022

MEHROTRA, K. el al. Título: **ELEMENTS OF ARTIFICIAL NEURAL NETWORKS** 1996

MOHANTY, S. N. Título: **Machine Learning for Healthcare** Applications Local de publicação: John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, USA and Scrivener Publishing LLC, 100 Cummings Center, Suite 541J, Beverly, MA 01915, USA 2021

MITCHELL, T. M. Título: **Machine Learning** Local de publicação: McGraw-Hill Science/Engineering/Math Março 1, 1997

MOHRI, M. el al. Título: **Foundations of Machine Learning second edition** Local de publicação:The MIT Press Cambridge, Massachusetts London, England 2018

MUELLER, J. P; MASSARON, L Título: **MACHINE LEARNING FOR DUMMIES** Local de publicação: John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030-5774, 2016

MULLER, A. C. GUIDO S. Título: **Introduction to Machine Learning with Python: A Guide for Data Scientists Applications** Local de publicação: O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472 2017

MULLER, J. P. MASSARON L. Título: **Machine Learning For Dummies** Local de publicação: John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030-5774 2016

MULLER, A. C; GUIDE, S Título: **INTRODUCTION TO MACHINE LEARNING WITH PYTHON** Local de publicação: O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2017

**O Efeito do Batch Size no Treinamento de Redes Neurais Artificiais**, Deep Learning Book s.d. Disponível em:

<https://www.deeplearningbook.com.br/o-efeito-do-batch-size-no-treinamento-de-redes-neurais-artificiais/> Acesso em: mar de 2022

**Outubro Rosa 2020**. Instituto Nacional do Câncer, 21 de out. de 2020, Disponível em:

<https://www.inca.gov.br/campanhas/cancer-de-mama/2020/outubro-rosa-2020> Acesso em: mar de 2021

**O que é câncer?** Instituto Nacional do Câncer, 30 de nov. de 2020 Disponível em: <https://www.inca.gov.br/o-que-e-cancer> Acesso em: mar de 2021

PANESAR, A. Título: **Machine Learning and AI for Healthcare** Local de publicação:Library of Congress Control Number: 2018967454 2019

**Prevenção**, Instituto Nacional do Câncer 17 de Mai. de 2021, Disponível em:

<https://www.inca.gov.br/controle-do-cancer-de-mama/acoes-de-controle/prevencao> Acesso em: mar de 2021

PRODANOV, C. C.; FREITAS, E. C. De. **Metodologia do trabalho científico: métodos e técnicas da pesquisa e do trabalho acadêmico**. Disponível em: <https://www.feevale.br/Comum/midias/0163c988-1f5d-496f-b118-a6e009a7a2f9/E-book Metodologia do Trabalho Cientifico.pdf>

**Quimioterapia para Câncer de Mama**, Oncoguia 06 de 0ut. de 2014, Disponível em:

<http://www.oncoguia.org.br/conteudo/quimioterapia-para-cancer-de-mama/1405/265/> Acesso em: mar de 2021

**Radioterapia para Câncer de Mama**, Oncoguia 06 de 0ut. de 2014, Disponível em:

<http://www.oncoguia.org.br/conteudo/radioterapia-para-cancer-de-mama/1406/265/> Acesso em: mar de 2021

RAMANI, K. V. et al. Título: **Breast Cancer Medical Treatment, Side Effects, and Complementary Therapies** Local de publicação: Momentum Press, LLC 222 East 46th Street, New York, NY 10017 2017

RASCHKA, S; LIU, Y; MIRJALILI, V. Título: **MACHINE LEARNING WITH PYTORCH AND SCIKIT-LEARN** Local de publicação: Packt Publishing Ltd. Livery Place 35 Livery Street Birmingham B3 2PB, UK. 2022

SANTANA, M. **Deep Learning: do Conceito às Aplicações** 19 de Jul. de 2018 Disponível em:<https://medium.com/data-hackers/deep-learning-do-conceito-%C3%A0s-aplica%C3%A7%C3%B5es-e8e91a7c7eaf/> Acesso em: 22 de Mar. de 2022

SHAIKH K. et al. **Artificial Intelligence in Breast Cancer Early Detection and Diagnosis** Local de publicação: Springer Nature Switzerland AG 2021

SINGH, P; MANURE, A. Título: **LEARN TENSORFLOW 2.0** 2020

**Sobre o Câncer de Mama Avançado**, Oncoguia 02 de Out. de 2020, Disponível em:

<http://www.oncoguia.org.br/conteudo/sobre-o-cancer/6239/826/> Acesso em: mar de 2022

**Supervised Learning**, IBM 19 de Ago. de 2020, Disponível em:

<https://www.ibm.com/cloud/learn/supervised-learning> Acesso em: Abr de 2021

**Stages 0 & 1 Breast Cancer Overview**, National Breast Câncer 15 de Abr. de 2020, Disponível em:

<https://www.nationalbreastcancer.org/breast-cancer-stage-0-and-stage-1> Acesso em: mar de 2022

**Tipos de câncer**. Instituto Nacional do Câncer, 04 de mar. de 2021 Disponível em:

<https://www.inca.gov.br/tipos-de-cancer/cancer-de-mama> Acesso em: mar de 2021

**Tratamento do Câncer de Mama Inflamatório**, Oncoguia 25 de Jul. de 2020, Disponível em:

<http://www.oncoguia.org.br/conteudo/tratamento-do-cancer-de-mama-inflamatorio/13797/265/> Acesso em: mar de 2021

THEOBALD, O. Título: **MACHINE LEARNING FOR ABSOLUTE BEGINNERS 2nd EDITION** 2017

**Tratamento do Câncer de Mama por Estágio**, Oncoguia 25 de Jul. de 2020, Disponível em:

<http://www.oncoguia.org.br/conteudo/imunoterapia-para-cancer-de-mama/13088/265/> Acesso em: mar de 2021

**Tratamento do Câncer de Mama por Estágio**, Oncoguia 25 de Jul. de 2020, Disponível em:

<http://www.oncoguia.org.br/conteudo/tratamento-do-cancer-de-mama-por-estagio/6566/265/> Acesso em: mar de 2021

**Types of Breast Cancer**, Oncoguia 15 de Abr. de 2020, Disponível em:

<https://www.nationalbreastcancer.org/types-of-breast-cancer/> Acesso em: mar de 2022

**What is a Convolutional Neural Network?,** CNN Explainer s.d. Disponível em:

<https://poloclub.github.io/cnn-explainer/> Acesso em: mar de 2022

**Unsupervised Learning**, IBM, 21 de Set. de 2020, Disponível em:

<https://www.ibm.com/cloud/learn/unsupervised-learning/> Acesso em: Abr de 2021

ZAFAR, I. el al. Título: **HANDS ON CONVOLUTIONAL NEURAL NETWORKS WITH TENSORFLOW** Local de publicação: Packt Publishing Ltd. Livery Place

35 Livery Street Birmingham, UK. 2018