Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

Comparação entre diferentes modelos de redes neurais convolucionais para classificação de melanoma

Autor: Elias Queiroga Vieira
Orientador: Prof. Dr. Renan Utida Barbosa Ferreira

Brasília, DF
2022

Elias Queiroga Vieira

Comparação entre diferentes modelos de redes neurais convolucionais para classificação de melanoma

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Renan Utida Barbosa Ferreira

Brasília, DF

2022

Elias Queiroga Vieira

Comparação entre diferentes modelos de redes neurais convolucionais para classificação de melanoma/ Elias Queiroga Vieira. – Brasília, DF, 2022-

80 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Renan Utida Barbosa Ferreira

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2022.

1. Redes Neurais. 2. Melanoma. I. Prof. Dr. Renan Utida Barbosa Ferreira. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Comparação entre diferentes modelos de redes neurais convolucionais para classificação de melanoma

CDU 02:141:005.6

Elias Queiroga Vieira

Comparação entre diferentes modelos de redes neurais convolucionais para classificação de melanoma

Monografia submetida ao curso de graduação em Engenharia Eletrônica da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia Eletrônica.

Trabalho aprovado. Brasília, DF, 11 de maio de 2022:

Prof. Dr. Renan Utida Barbosa Ferreira
Orientador

Prof. Dr. Cristiano Jacques Miosso
Convidado 1

Prof. Dr. Bruno Luiggi Macchiavello Espinoza
Convidado 2

Brasília, DF
2022

Agradecimentos

Gostaria de agradecer imensamente o apoio incondicional e essencial da minha família em todo esse processo de graduação, como também em outros processos difíceis que enfrentei em minha vida.

Também gostaria de agradecer a todas as minhas amizades, que sempre se mostraram presentes e tornaram essa jornada acadêmica mais prazerosa, mesmo com todos esses eventos que estamos vivendo nos últimos anos.

E por fim, gostaria de agradecer ao corpo docente da FGA, por proporcionar um ensino de qualidade e abrir caminho para meu aprendizado e capacitação. Em especial, gostaria de agradecer ao meu orientador Renan Utida, que sempre se mostrou presente e foi um guia imprescindível na construção deste trabalho.

"I must not fear.
Fear is the mind-killer.
Fear is the little-death that brings total obliteration.
I will face my fear. I will permit it to pass over me and through me.
And when it has gone past, I will turn the inner eye to see its path.
Where the fear has gone, there will be nothing.
Only I will remain."
(Frank Herbert, Dune, 1965)

Resumo

O câncer de pele é o mais comum no Brasil atualmente, correspondendo a cerca de 30% dos tumores malignos encontrados. O melanoma é um câncer de pele incomum que constitui apenas 3% das neoplasias de pele. Entretanto, o melanoma tem alta possibilidade de atingir metástase por ser bastante agressivo e de evolução rápida, sendo considerado o câncer de pele mais grave e com pior prognóstico se não for identificado cedo. Com o passar dos anos foram surgindo novas técnicas dermatológicas para diagnóstico do câncer de pele, uma delas é o diagnóstico auxiliado por computador. Atualmente, com o avanço do poder computacional e das inovações em aprendizado de máquina, já é bastante comum o estudo de redes neurais para classificação de melanoma, principalmente o uso de redes neurais convolucionais. Este trabalho tem como objetivo comparar o estado da arte das redes neurais convolucionais no que diz respeito à classificação de melanomas. Foi utilizada a arquitetura EfficientNet-B7, treinada e testada com os conjuntos de dados ISIC 2019 e ISIC 2020, e feita a comparação com as redes EfficientNet-B4, EfficientNet-B5, EfficientNet-B6 e DenseNet201, que também utilizaram o mesmo conjunto de dados para treinamento e teste. Toda a implementação do trabalho foi feita usando a plataforma online Kaggle, utilizando Python e as bibliotecas TensorFlow e Keras. A EfficientNet-B7 obteve pontuação AUC de 0.9467, que foi gerada a partir do *ensemble* de todos os modelos treinados, comparados com a pontuação da DenseNet201 de 0.9250 e da EfficientNet-B5+B6 de 0.9411, o que demonstra a eficiência e capacidade da EfficientNet-B7 em classificar o melanoma em imagens dermatoscópicas de lesões de pele.

Palavras-chaves: Melanoma. Classificação. Aprendizado de máquina. Redes neurais convolucionais.

Abstract

Skin cancer is currently the most common in Brazil, corresponding to about 30% of malignant tumors found. Melanoma is an uncommon skin cancer that constitutes only 3% of skin neoplasms. However, melanoma has a high possibility of metastasis because it is very aggressive and has a rapid evolution, being considered the most serious skin cancer and with the worst prognosis if it is not identified early. Over the years, new dermatological techniques for the diagnosis of skin cancer have emerged, one of which is computer-aided diagnosis. Currently, with the advance of computational power and innovations in machine learning, the study of neural networks for melanoma classification is quite common, especially the use of convolutional neural networks. This work aims to compare the state of the art of convolutional neural networks regarding the classification of melanomas. The EfficientNet-B7 architecture was used, trained and tested with the ISIC 2019 and ISIC 2020 datasets, and a comparison was made with the EfficientNet-B4, EfficientNet-B5, EfficientNet-B6 and DenseNet201 networks, which also used the same datasets for training and testing. The entire implementation of the work was done using the online platform Kaggle, Python was used along with TensorFlow and Keras libraries. EfficientNet-B7 had an AUC score of 0.9467, which was generated from the ensemble of all trained models, compared to the DenseNet201 score of 0.9250 and the EfficientNet B5+B6 score of 0.9411, which demonstrates the efficiency and capacity of the EfficientNet-B7 in classifying melanoma in dermoscopic images of skin lesions.

Key-words: Melanoma. Classification. Machine learning. Convolutional neural networks.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ABCD	Assimetria, Borda, Cor e Diâmetro
AdaGrad	Algoritmo de Gradiente Adaptativo, do inglês *Adaptive Gradient Algorithm*
ADALINE	Elemento Linear Adaptativo, do inglês *Adaptive Linear Element*
Adam	Estimativa de Momento Adaptativo, do inglês *Adaptive Moment Estimation*
ANN	Redes Neurais Artificiais, do inglês *Artificial Neural Networks*
API	Interface de Programação de Aplicação, do inglês *Application Programming Interface*
AUC	Área Sob a Curva, do inglês *Area Under the Curve*
BCE	Entropia Cruzada Binária, do inglês *Binary Cross-Entropy*
CAD	Diagnóstico Auxiliado por Computador, do inglês *Computer-aided Diagnosis*
CNN	Rede Neural Convolucional, do inglês *Convolutional Neural Network*
CNTK	Conjunto de Ferramentas Cognitivas da Microsoft, do inglês *Microsoft Cognitive Toolkit*
CPU	Unidade de Processamento Central, do inglês *Central Processing Unit*
DNA	Ácido Desoxirribonucleico, do inglês *Deoxyribonucleic Acid*
EDP	Equações Diferenciais Particiais
FN	Falsos Negativos
FNN	Rede Neural *Feedforward*, do inglês *Feedforward Neural Network*
FP	Falsos Positivos
GPU	Unidade de Processamento Gráfico, do inglês *Graphics Processing Unit*
GT	Verdade Básica, do inglês *Ground Truth*
IA	Inteligência Artificial

ILSVRC	Desafio de Reconhecimento Visual em Grande Escala ImageNet, do inglês *ImageNet Large Scale Visual Recognition Challenge*
INCA	Instituto Nacional de Câncer
ISIC	Colaboração Internacional de Imagens da Pele, do inglês *International Skin Imaging Collaboration*
LM	Lentigo Maligno
LMM	Lentigo Maligno Melanoma
ML	Aprendizado de Máquina, do inglês *Machine Learning*
MLA	Melanoma Lentiginoso Acral
MLP	*Perceptron* de Multicamada, do inglês *Multilayer Perceptron*
MN	Melanoma Nodular
MXU	Unidade Multiplicadora de Matrizes, do inglês *Matrix Multiplying Unit*
NV	Negativos Verdadeiros
PV	Positivos Verdadeiros
RAM	Memória de Acesso Aleatório, do inglês *Random Access Memory*
ReLU	Unidade Linear Retificada, do inglês *Rectified Linear Unit*
RGB	Vermelho, Verde e Azul, do inglês *Red, Green and Blue*
RMSProp	Propagação Quadrática Média, do inglês *Root Mean Squared Propagation*
RMSE	Erro Quadrático Médio, do inglês *Root Mean Square Error*
RNN	Rede Neural Recorrente, do inglês *Recurrent Neural Network*
ROC	Característica de Operação do Receptor, do inglês *Receiver Operating Characteristic*
SGD	Descida de Gradiente Estocástico, do inglês *Stochastic Gradient Descent*
SSM	Melanoma Extensivo Superficial, do inglês *Superficial Spreading Melanoma*
tanh	Tangente Hiperbólica

TFP	Taxa de Falsos Positivos
TNV	Taxa Negativa Verdadeira
TPV	Taxa de Positivos Verdadeiros
TPU	Unidade de Processamento de Tensores, do inglês *Tensor Processing Unit*
UV	Ultravioleta
VPU	Unidade de Processamento de Vetores, do inglês *Vector Processing Unit*

Lista de símbolos

$\mathbf{b}$	Vetor de polarização
C_i	Canal
ϕ	Função de ativação
$\mathcal{F}_i$	Operador da camada
H_i	Altura da imagem
$h_{\mathbf{W,b}}$	Saída da camada densa
I	Imagem discretizada
k	Classificação dos positivos
L_i	Camada
n_{neg}	Número de amostras negativas
n_{pos}	Número de amostras positivas
p_j	j-ésimo valor escalar na saída do modelo
t_j	Valor alvo correspondente
w	Função peso do *kernel*
$\mathbf{W}$	Matriz de pesos
W_i	Largura da imagem
$\mathbf{X}$	Matriz de características

Sumário

1 Introdução

1.1 Contextualização

O câncer de pele é o câncer de maior incidência no Brasil, representando cerca de 30% de todos os tumores malignos registrados. O melanoma compõe 3% do total e é o terceiro tipo mais comum de câncer de pele. No entanto, é a principal causa de morte por câncer de pele por ser o tipo mais agressivo, com rápida progressão e evolução para metástase (ONCOGUIA, 2020). Segundo o Instituto Nacional de Câncer - INCA, a estimativa de novos casos no Brasil é de 8450 por ano, sendo 4200 homens e 4250 mulheres (INCA, 2021). O número de mortes anual é de 1978, sendo 1159 homens e 819 mulheres. A taxa de sobrevida em 5 anos do melanoma é de 25% quando o mesmo já atingiu um estágio avançado (ONCOGUIA, 2020).

Biópsias ainda são o método mais utilizado para confirmação de diagnóstico de melanoma (SOMMER, 2008), porém, na década de 80 houve um aumento na incidência de câncer de pele que demandou métodos não invasivos para detecção prematura dessas lesões malignas. Esses métodos permitem que os médicos reexaminem a mesma área repetidamente ao longo do tempo, sem ferir ou alterar o tecido. Por outro lado, fazer várias biópsias pode danificar o tecido e a formação de cicatrizes pode dificultar o estudo posterior (BAKOS *et al.*, 2018).

A instrumentação óptica para imagens é um componente importante do progresso das ciências biológicas e médicas. No campo da dermatologia, as tecnologias de imagem não invasivas são muito úteis no monitoramento de doenças de pele, principalmente quando as condições cutâneas apresentam características ambíguas, levando a atrasos no tratamento. Alguns desses métodos são: dermatoscopia, microscopia confocal, tomografia de coerência óptica, interferometria, espectroscopia vibracional, microscopia de fluorescência, entre outros (TKACZYK, 2017).

As instrumentações citadas servem de assistência no exame físico dermatológico, provendo uma acurácia acima de 80%. A dermatoscopia por exemplo, atingiu 89% de sensibilidade (a capacidade de um teste para identificar corretamente os positivos verdadeiros) e 84% especifidade (a capacidade de um teste para identificar os negativos verdadeiros) nos melhores testes clínicos *in vivo* (TKACZYK, 2017), mas podendo obter valores baixos como em diagnósticos de melanoma facial, que mesmo com a dermatoscopia atingiram uma mínima de 54% de sensibilidade, como visto em Wolner *et al.* (WOLNER *et al.*, 2017).

Idealmente, uma inspeção por um dermatologista especialista detectaria com pre-

cisão os melanomas em estágio inicial, no entanto, não é prático para todos os pacientes receberem exames intensivos por dermatologistas. Para superar esse problema, muitos estudos estão em andamento para desenvolver diagnósticos auxiliados por computador ou CAD (do inglês *computer-aided diagnosis*) (FUJISAWA; INOUE; NAKAMURA, 2019). A expectativa é que o CAD possa facilitar o diagnóstico precoce pela capacidade de analisar um grande número de imagens, podendo ser aplicados em triagens comuns de hospital ou até mesmo em serviços de teleconsulta (DICK *et al.*, 2019).

Uma das dificuldades em detectar câncer de pele por meio de imagens, utilizando métodos computacionais, é a variação que cada tipo de pele pode apresentar, como por exemplo a tonalidade. Existem também os desafios de lidar com a imagem em si, que pode conter diversos tamanhos e formas, assim como ruídos na lesão a ser avaliada, como pêlos, marcas de nascença, lesões benignas de baixo contraste, baixa iluminação ou reflexo. Tudo isso torna o processo de detecção computadorizada de câncer de pele bastante desafiador (ALI *et al.*, 2021).

Tradicionalmente, o CAD começa com o pré-processamento, nessa etapa o foco é remover artefatos e ruídos da imagem dermatoscópica da pele. Logo após vem a fase de análise, que inclui a segmentação, extração e seleção de características importantes para efetuar a classificação da imagem, que é a última etapa desse processo. Existem muitos algoritmos diferentes disponíveis para esta tarefa de classificação, como máquina de vetores e suporte, árvores de decisão, regressão logística e redes neurais artificiais, também conhecidas como ANN (do ingles *artificial neural networks*) (FUJISAWA; INOUE; NAKAMURA, 2019).

Hoje em dia está sendo estudado o uso de redes neurais e aprendizagem profunda (*deep learning*) para classificação de imagens médicas. Redes neurais convolucionais ou CNNs (do inglês *Convolutional Neural Networks*) podem aprender e determinar automaticamente quais informações são importantes para a classificação a partir das imagens de treinamento. A extração e seleção das características para classificação é um componente fundamental dos métodos tradicionais citados acima, e também a parte mais difícil. Assim, usando CNNs, o pré-processamento de imagem complicado não é mais necessário para obter valores de recursos ideais para a classificação da imagem (LI *et al.*, 2014).

1.2 Problema de pesquisa e justificativa

Resultados disponíveis em Brinker *et al.* (BRINKER *et al.*, 2019) e Haenssle *et al.* (HAENSSLE *et al.*, 2018; HAENSSLE *et al.*, 2020) mostram que a classificação de melanoma automatizada através de redes neurais atinge uma precisão significativamente superior à de dermatologistas, com ou sem experiência de trabalho. Diversas competições surgem todos os anos evidenciando o que há de mais novo nessa tecnologia, competições

como a ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) (IMAGENET, 2017) e o desafio ISIC (*International Skin Imaging Collaboration*) (ISIC, 2020) acontecem regularmente e fomentam o avanço das técnicas utilizadas em redes neurais através da comunidade. Redes como AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), GoogLeNet (SZEGEDY *et al.*, 2014), VGG (SIMONYAN; ZISSERMAN, 2015) e ResNet (HE *et al.*, 2015) já são bastante populares por vencerem diversas dessas competições.

Com o avanço do poder de processamento das máquinas, tanto em quesito de CPU (*Central Processing Unit*), GPU (*Graphics Processing Unit*), e mais recentemente, TPU (*Tensor Processing Unit* acelerador específico para inteligência artificial) (JOUPPI *et al.*, 2017), novos métodos de aprendizado estão surgindo e com mais eficiência durante o treinamento. Novos métodos de armazenar e acessar dados, assim como otimizar o algoritmo de aprendizado também foram descobertos, levantando a necessidade de aplicarmos essas novas tecnologias no campo da identificação de melanoma e compararmos o estado da arte com as redes mais recentes que venceram competições de detecção de melanoma.

1.3 Objetivos

1.3.1 Objetivo geral

Realização do treinamento da rede neural convolucional EfficientNet-B7 utilizando pesos pré-treinados. Após treinamento, obter pontuações de desempenho e compará-las a outros modelos encontrados na literatura, que utilizam condições de treino similar e também realizam a tarefa de classificação de melanoma.

1.3.2 Objetivos específicos

Com o intuito de atingir o objetivo geral, são propostos os seguintes objetivos intermediários:

- Obter informações do que há de mais novo no campo das redes neurais convolucionais para classificação de imagem.
- Fazer a utilização de *backbones* e adaptar a rede para o problema de classificação de melanoma atráves da técnica de *transfer learning*.
- Selecionar os bancos de dados a serem utilizados para treinar a rede.
- Ajustar os parâmetros da rede para que se obtenha a maior precisão sem prejuízo na otimização da mesma.
- Comparar os resultados obtidos com estudos recentes que utilizam o mesmo banco e condições de treinamento semelhantes porém usando redes diferentes.

1.4 Estrutura do texto

No Capítulo 2 - Referencial Teórico, é realizado um breve estudo a cerca do melanoma, tal como suas causas, tratamentos disponíveis e sua classificação no meio clínico. Neste mesmo capítulo é apresentada uma revisão bibliográfica com a descrição de técnicas de aprendizado de máquina e aprendizado profundo, arquiteturas das redes estudadas e as métricas usadas para a avaliação de desempenho das mesmas.

No Capítulo 3 - Metodologia, é descrita a metodologia de treinamento e como será feita a comparação das diferentes arquiteturas. Também estão descritas as ferramentas de *hardware* e *software* utilizadas para a execução deste trabalho e o procedimento realizado para a obtenção das métricas de desempenho.

No Capítulo 4 - Resultados e discussões, são apresentadas as métricas obtidas ao final do treinamento de cada rede, assim como outros dados analíticos. Também é feita uma análise dos resultados obtidos.

E por fim, no Capítulo 5 - Conclusão, é apresentanda uma síntese de todo o processo e dos resultados obtidos a cerca do desempenho de cada rede treinada, tal como a proposta de projetos futuros.

2 Referencial teórico

Para que haja completa compreensão do trabalho, primeiro serão definidos alguns conceitos sobre o melanoma. Logo após, serão apresentados conceitos de imagem digitais e a aplicação de filtros. Por fim, serão abordados os conceitos de aprendizado de máquina até redes neurais convolucionais e um detalhamento da arquitetura a ser utilizada.

2.1 Melanoma

2.1.1 Etiologia e patogênese

O câncer surge quando há um crescimento desordenado de células atípicas decorrente de danos no DNA ou algum defeito genético congênito (WEINBERG, 2014). A integridade do genoma de todos os organismos vivos é constantemente ameaçada por agentes exógenos e endógenos que danificam o DNA. O melanoma é um tipo de câncer de pele que se desenvolve quando os melanócitos (as células que dão à pele sua cor bronzeada ou marrom) começam a crescer descontroladamente. O melanoma é muito menos comum do que alguns outros tipos de câncer de pele, mas é mais perigoso porque é muito mais provável que se espalhe para outras partes do corpo se não for detectado e tratado precocemente (LEE; FARIES, 2020). Os principais fatores de risco associados ao melanoma são descritos a seguir.

2.1.1.1 Fatores de risco

- **Exposição ao sol** - Fatores genéticos e ambientais estão relacionados à patogênese do melanoma e certamente nem todos os melanomas são relacionados ao sol. Há, no entanto, evidências claras e convincentes de que a exposição ao sol, e mais especificamente a exposição ultravioleta (UV), é uma das principais causas ambientais de melanoma, especialmente em populações de alto risco (MASSI; LEBOIT, 2014).

- **Fenótipo da pele** - Pigmentação da pele clara, cabelo loiro ou ruivo, olhos azuis ou verdes, tendência proeminente de sardas e tendência a queimaduras solares são características fenotípicas associadas a um risco aumentado de melanoma. O melanoma ocorre com muito menos frequência em pessoas de pele mais escura, sugerindo que o pigmento da pele desempenha um papel protetor (MASSI; LEBOIT, 2014).

- **Nevo melanocítico** - Há um risco aumentado de melanoma associado a nevos (pequenos tumores cutâneos), tanto de maneira quantitativa (ou seja, número de nevos) e qualitativa (ou seja, nevos típicos vs. atípicos). Adultos com mais de 100

nevos de aparência clinicamente típica, crianças com mais de 50 nevos de aparência típica e qualquer paciente com nevos atípicos estão em risco. A presença de um nevo displásico solitário pode dobrar o risco de melanoma (WOLFF *et al.*, 2011).

- **Histórico pessoal** - Um histórico anterior de melanoma aumenta o risco de outro melanoma primário, com 5% a 15% dos indivíduos desenvolvendo múltiplos melanomas primários. Em pacientes com múltiplos melanomas primários, cerca de metade desenvolve um segundo tumor primário na mesma região de o corpo (ou seja, tronco, extremidade, cabeça e pescoço) e cerca da metade desenvolve um segundo melanoma primário no primeiro ano do diagnóstico inicial (MASSI; LEBOIT, 2014).

- **Genética** - Alguns genes quando sofrem mutação podem apresentar predisposição para desenvolvimento do melanoma, como por exemplo o gene CDKN2A-CDK4-TP53, o gene BRAF e MAPK, e o gene MITF (WOLFF *et al.*, 2011).

2.1.1.2 Progressão do tumor

Cinco estágios de transformação maligna e progressão tumoral em melanócitos foram sugeridos, com base nas propriedades clínicas, histopatológicas, imunopatológicas, citogenéticas e *in vitro*: (1) nevos melanocíticos benignos; (2) nevos atípicos; (3) melanoma maligno primário, fase de crescimento radial; (4) melanoma maligno primário, fase de crescimento vertical; e (5) melanoma maligno metastático. Acredita-se que a cada etapa sucessiva da tumorigênese, um novo clone de células emerge com vantagens de crescimento sobre o tecido circundante, resultando em "expansão clonal". Postulou-se que uma etapa crítica na progressão tumoral do melanoma pode ser a transição das fases de crescimento radial para vertical (MASSI; LEBOIT, 2014).

2.1.2 Diagnóstico diferencial

2.1.2.1 Melanoma extensivo superficial

O melanoma extensivo superficial (SSM, do inglês *Superficial Spreading Melanoma*) é o subtipo mais comum, sendo responsável por aproximadamente 70% de todos os melanomas cutâneos. É diagnosticado mais comumente em áreas intermitentemente expostas ao sol, mais freqüentemente nas extremidades inferiores das mulheres e na parte superior das costas dos homens (MASSI; LEBOIT, 2014). Sua aparência clínica clássica se encaixa melhor nos critérios do ABCD (assimetria, borda, cor e diâmetro), com bordas irregulares e pigmentação irregular, mas pode se apresentar sutilmente como uma área focal discreta de escurecimento dentro de um nevo preexistente. A faixa de aparência do SSM é ampla. Embora tons variados de marrom tipifiquem a maioria das lesões melanocíticas, aspectos marcantes de marrom escuro a preto, cinza-azulado, rosa, vermelho e

branco-acinzentado (que podem representar regressão) podem ser encontrados no melanoma. SSM é o subtipo de melanoma mais comumente associado a nevos preexistentes. O histórico do SSM costuma ser de uma lesão que muda lentamente ao longo de meses ou anos. Pode ser confundido com um nevo atípico ou ceratose seborreica (ROSENDAHL; MAROZAVA, 2020).

2.1.2.2 Melanoma nodular

O melanoma nodular (MN) é o segundo subtipo de melanoma mais comum e representa aproximadamente 15% a 30% de todos os melanomas. O tronco é o local mais comum. O MN é notável pela rápida evolução, freqüentemente surgindo ao longo de várias semanas ou meses. Na maioria das vezes, o MN carece de uma fase de crescimento radial aparente. É mais comum para o MN começar do zero do que surgir em um nevo preexistente. Segundo Hendi e Martinez (HENDI; MARTINEZ, 2011), o MN tipicamente aparece como uma lesão uniformemente elevada em preto-azulado ou vermelho-azulado, mas 5% são amelanóticas. Uma proporção substancial dos melanomas espessos é do tipo nodular. As lesões iniciais geralmente carecem de assimetria, têm bordas regulares e têm uma cor uniforme. Lesões amelanóticas podem ser confundidas com carcinoma basocelular, granuloma piogênico ou hemangioma, enquanto lesões pigmentadas podem ser confundidas com nevos azuis ou carcinomas basocelulares pigmentados (WOLFF *et al.*, 2011).

2.1.2.3 Lentigo maligno ou melanoma *in situ*

Lentigo maligna (LM) é um subtipo de melanoma *in situ* (no local original de formação) com uma fase de crescimento radial prolongada que pode progredir para um *Lentigo Maligna Melanoma* (LMM) invasivo com o tempo. LMM invasivo constitui de 10% a 15% dos melanomas cutâneos. LM e LMM são diagnosticados mais comumente na sétima à oitava décadas em uma população mais velha do que outros tipos de melanoma, incomum antes dos 40 anos (MASSI; LEBOIT, 2014). A localização mais comum é no rosto cronicamente exposto ao sol, principalmente nas bochechas e no nariz, pescoço, couro cabeludo e orelhas nos homens. Acredita-se que sua patogênese esteja relacionada à exposição cumulativa ao sol, e não à exposição intermitente. O LM é uma mácula achatada, marrom, de aumento lento, semelhante a sardas, com forma irregular e diferentes tons de marrom e castanho, geralmente surgindo em um fundo de fotodano (WOLFF *et al.*, 2011).

2.1.2.4 Melanoma lentiginoso acral

O melanoma lentiginoso acral (MLA) é um subtipo de melanoma com diferenças distintas nas frequências observadas entre grupos étnicos. O MLA constitui apenas 2% – 8% dos melanomas em caucasianos, mas representa a forma mais comum em indivíduos

Figura 1 – Representação dos 4 subtipos de melanoma mencionados: (A) melanoma extensivo superficial; (B) melanoma nodular; (C) lentigo Maligno e (D) melanoma lentiginoso acral. Fonte: (WOLFF *et al.*, 2011).

com pigmentação mais escura (60% – 72% afrodescendentes e 29% – 46% asiáticos). Embora a proporção de MLA observada em indivíduos com pigmentação mais escura seja maior, a incidência de MLA é semelhante para caucasianos e outras etnias. O MLA é diagnosticado com mais frequência em uma população mais velha, com idade média de início de 65 anos. O local mais comum para MLA é a sola, seguido da palma e da localização subungueal. Nem todos os melanomas palmares ou plantares são MLAs. A aparência clínica da MLA pode ser marrom, preta, castanha ou vermelha com variações de cor e bordas irregulares. No entanto, a cor mais comum é marrom-preto (WOLFF *et al.*, 2011).

2.2 Imagens digitais

Uma imagem digital pode ser considerada uma representação discreta de dados que possuem informações espaciais e de intensidade (cor). A imagem digital bidimensional e discreta $f(x, y)$ representa a resposta dos sensores de imagem (fotografia) em uma série de posições fixas ($x = 0, 1, 2, ..., M - 1; y = 0, 1, 2, ..., N - 1$) em uma sistema

cartesiano de coordenadas e é derivado de um sinal espacial contínuo através de um processo de amostragem frequentemente referido como discretização. Desta forma, uma imagem digital pode ser descrita como uma matriz de dimensão $M \times N$ (GONZALEZ; WOODS, 2018), mostrada a seguir:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}. \tag{2.1}$$

Cada elemento numérico representado na matriz da Equação 2.1 é chamado de *pixel*, os valores de *pixels* individuais na maioria das imagens correspondem a alguma resposta física no espaço 2-D real (por exemplo, a intensidade óptica recebida no sensor de uma câmera ou a intensidade de ultrassom em um transceptor) (SOLOMON; BRECKON, 2011). O número de *pixels* presentes na matriz está diretamente relacionado com a resolução espacial da imagem.

Uma imagem possui um ou mais canais de cor que definem a intensidade ou cor num determinado *pixel* $f(x,y)$. O valor numérico de cada *pixel* varia entre preto (0) até o branco (máximo da escala). No caso da escala de cinza (*greyscale*), o valor máximo é 255, pois a mesma possui uma intensidade de 8 *bits* para seu canal de cor. No caso de imagens coloridas, geralmente são utilizados 24 *bits* de intensidade, logo seu valor máximo seria de $2^{24} - 1 = 16777215$, possuindo um alcance muito grande para representar cada cor (GONZALEZ; WOODS, 2018).

Imagens coloridas possuem 3 mapas de cores, logo teremos uma matriz $M \times N$ para cada canal de cor no espectro RGB (*red*, *green* e *blue*). A imagem colorida então será o resultado da interpolação linear dessas três matrizes, possuindo então a seguinte dimensão: $M \times N \times 3$ (MCANDREW, 2015).

2.2.1 Filtros

As operações de processamento de imagem podem transformar os valores dos *pixels*. As operações de processamento de imagem podem ser executadas apenas em um *pixel* isolado ou podem ser executadas na vizinhança do *pixel* em questão para atingir novos valores. O processo de filtragem está profundamente relacionado com o processamento de vizinhança (MCANDREW, 2015).

Em filtros espaciais lineares, o valor novo ou filtrado do *pixel* de destino é determinado como alguma combinação linear dos valores de *pixels* em sua vizinhança. Qualquer outro tipo de filtro é, por definição, um filtro não linear. A combinação linear específica

obtida dos *pixels* vizinhos é determinada pelo *kernel* do filtro (geralmente chamado de máscara)(MCANDREW, 2015).

A mecânica da filtragem espacial linear expressa de forma discreta um processo chamado convolução. Por esse motivo, muitos *kernels* de filtro às vezes são descritos como *kernels* de convolução. Formalmente, podemos expressar a equação de convolução discreta bidimensional entre um *kernel* e uma imagem da seguinte maneira:

$$f(x,y) = \sum_{i=I_{min}}^{I_{max}} \sum_{j=J_{min}}^{J_{max}} w(i,j)I(x+i,y+j), \tag{2.2}$$

sendo w o peso do kernel do filtro e I a imagem em a ser filtrada (SOLOMON; BRECKON, 2011).

2.3 Inteligência artificial e aprendizado de máquina

Inteligência artificial (IA) é um campo geral que abrange aprendizado de máquina e aprendizado profundo, mas que também inclui muitas outras abordagens que não envolvem nenhum aprendizado. Os primeiros programas de xadrez, por exemplo, envolviam apenas regras codificadas elaboradas por programadores e não se qualificavam como aprendizado de máquina. Essa abordagem é conhecida como IA simbólica e foi o paradigma dominante em IA dos anos 1950 até o final dos anos 1980 (CHOLLET, 2018).

Em particular, definimos o aprendizado de máquina (mais comumente chamado de ML, do inglês *machine learning*) como um conjunto de métodos que podem detectar automaticamente padrões nos dados e, em seguida, usar os padrões descobertos para prever dados futuros ou para realizar outros tipos de tomada de decisão sob incerteza (RUSSELL; NORVIG, 2021).

O ML geralmente é dividido em três tipos. O primeiro tipo é a abordagem de aprendizagem preditiva ou supervisionada, o objetivo é gerar um mapeamento através de entradas x para saídas y, dado um conjunto rotulado de pares de entrada-saída $\mathcal{D} = (x_i, y_i)_{i=1}^N$, aonde $\mathcal{D}$ é chamado de conjunto de treinamento e N é o número de amostras de treinamento. Na situação mais simples, cada entrada de treinamento x_i é um vetor de dimensão D contendo números, representando, por exemplo, a altura e o peso de uma pessoa. Estes são chamados de características, atributos ou covariáveis. No geral, x_i pode ser um objeto estruturado complexo, como uma imagem, frase, mensagem de e-mail, uma forma molecular, "*etc*" (GOODFELLOW; BENGIO; COURVILLE, 2017).

Da mesma forma, a forma da variável de saída ou resposta pode, em princípio, ser qualquer coisa, mas a maioria dos métodos assume que y_i é uma variável categórica ou nominal de algum conjunto finito, $y_i \in \{1, \ldots, C\}$ (como masculino ou feminino), ou que y_i é um escalar com valor real (como nível de renda). Quando y_i é categórico, o problema

é conhecido como classificação ou reconhecimento de padrão, e quando y_i tem valor real, o problema é conhecido como regressão (GOODFELLOW; BENGIO; COURVILLE, 2017).

O segundo tipo de ML é a abordagem de aprendizado descritivo ou não supervisionado. Neste modelo, recebemos apenas entradas, $\mathcal{D} = \{x_i\}_{i=1}^N$, e o objetivo é encontrar "padrões interessantes" nos dados. Isso às vezes é chamado de descoberta de conhecimento. Este é um problema muito menos bem definido, uma vez que não somos informados de que tipos de padrões procurar, e não há uma métrica de erro óbvia para usar (ao contrário do aprendizado supervisionado, onde podemos comparar nossa previsão de y para um determinado x com o valor observado) (GOODFELLOW; BENGIO; COURVILLE, 2017).

O terceiro tipo de aprendizado de máquina, conhecido como aprendizado por reforço, é menos comumente usado. Isso é útil para aprender como agir ou se comportar quando recebe recompensas ocasionais ou sinais de punição (por exemplo, considere como um bebê aprende a andar) (GOODFELLOW; BENGIO; COURVILLE, 2017).

2.3.1 Classificação

O objetivo da classificação é obter um aprendizado através do mapeamento das entradas x para as saídas y, onde $y \in \{1, \cdots, C\}$, sendo C o número de classes. Se $C = 2$, isso é chamado de classificação binária (nesse caso, geralmente assumimos $y \in \{0, 1\}$). Se $C > 2$, isso é chamado de classificação multiclasse. Se os rótulos (*labels*) de classe não forem mutuamente exclusivos (por exemplo, alguém pode ser classificado como alto e forte), isso é chamado de classificação de vários rótulos (*multilabel*), mas que pode ser apresentado na saída com a previsão de vários *labels* binários relacionados (chamado modelo de saída múltipla), obtendo várias saídas binárias ao invés de apenas uma saída *multilabel* (MURPHY, 2012).

Existem outras variantes de classificação, por exemplo, onde y produz uma distribuição de probabilidade sobre as classes. Um exemplo de tarefa de classificação é o reconhecimento de objeto, em que a entrada é uma imagem (geralmente descrita como um conjunto de valores de brilho de *pixel*) e a saída é um código numérico que identifica o objeto na imagem (MURPHY, 2012).

A classificação torna-se mais desafiadora se não for garantido ao *software* de que todas as medições no vetor de entrada sempre serão fornecidas. Para resolver essa tarefa de classificação, o algoritmo de aprendizagem precisa definir um mapeamento de função única a partir de uma entrada vetorial para uma saída categórica. Quando algumas das entradas podem estar faltando, em vez de fornecer uma única função de classificação, o algoritmo de aprendizagem deve aprender um conjunto de funções. Cada função corresponde à classificação de x com um subconjunto diferente de suas entradas ausentes. Esse tipo

de situação surge com frequência no diagnóstico médico, porque muitos tipos de exames médicos são caros ou invasivos (GOODFELLOW; BENGIO; COURVILLE, 2017).

Uma maneira de definir com eficiência um conjunto tão grande de funções é aprender uma distribuição de probabilidade sobre todas as variáveis relevantes e, em seguida, resolver a tarefa de classificação marginalizando as variáveis ausentes. Com n variáveis de entrada, podemos agora obter todas as 2^n funções de classificação diferentes necessárias para cada conjunto possível de entradas ausentes, mas o *software* precisa aprender apenas uma única função que descreve a distribuição de probabilidade conjunta (GOODFELLOW; BENGIO; COURVILLE, 2017).

2.3.2 Modelos paramétricos e não paramétricos

Um modelo paramétrico possui um número de parâmetros fixo, já no modelo não paramétrico, o número de parâmetros cresce de acordo com os dados de treinamento. Os modelos paramétricos têm a vantagem de geralmente serem mais rápidos de implementar e a desvantagem de fazer suposições mais fortes sobre a natureza das distribuições dos dados. Modelos não paramétricos são mais flexíveis, porém muitas vezes computacionalmente intratáveis para grandes conjuntos de dados (*datasets*) (MURPHY, 2012).

2.3.3 Capacidade, sobreajuste e sub-ajuste

O desafio central no aprendizado de máquina é que o algoritmo deve ter um bom desempenho em entradas novas e não vistas, não apenas naquelas nas quais nosso modelo foi treinado. A capacidade de um bom desempenho em entradas não observadas anteriormente é chamada de generalização (MURPHY, 2012).

Normalmente, ao treinar um modelo de aprendizado de máquina, tem-se acesso a um conjunto de treinamento (*training set*). O erro surgido na etapa de treinamento será minimizado o máximo possível durante o processo. Contudo, é esperado que o erro de generalização, ou erro de teste, também seja mínimo. Normalmente estimamos o erro de generalização de um modelo de aprendizado de máquina medindo seu desempenho em um conjunto de teste (*test set*) de amostras que foram coletadas separadamente do conjunto de treinamento (MURPHY, 2012).

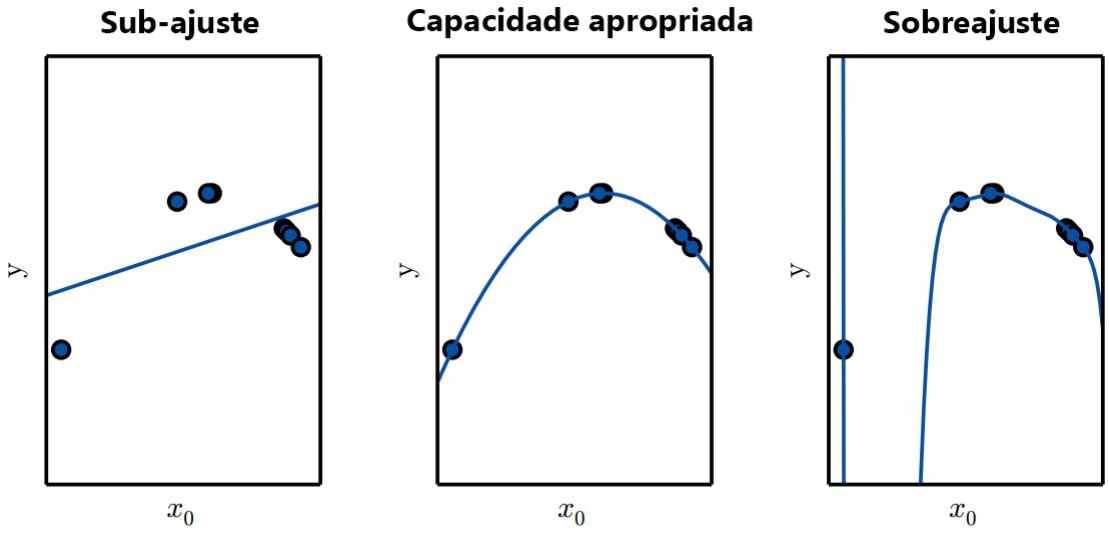


Figura 2 – Preditor linear, quadrático e de grau 9 tentando ajustar um problema onde a verdadeira função subjacente é quadrática. Adaptado de: (GOODFELLOW; BENGIO; COURVILLE, 2017)

Sub-ajuste (*Underfitting*) ocorre quando o modelo não é capaz de obter um valor de erro suficientemente baixo no conjunto de treinamento. Sobreajuste (*Overfitting*) ocorre quando a lacuna entre o erro de treinamento e o erro de teste é muito grande. A capacidade de um modelo é sua capacidade de se ajustar a uma ampla variedade de funções (Figura 2). Os modelos com baixa capacidade podem ter dificuldade em se ajustar ao conjunto de treinamento. Os modelos com alta capacidade podem se ajustar demais ao memorizar propriedades do conjunto de treinamento que não os atendem bem no conjunto de teste (GOODFELLOW; BENGIO; COURVILLE, 2017).

2.3.4 Conjuntos de validação, hiperparâmetros e validação cruzada

A maioria dos algoritmos de aprendizado de máquina tem hiperparâmetros, configurações que podemos usar para controlar o comportamento do algoritmo. Os valores dos hiperparâmetros não são adaptados pelo próprio algoritmo de aprendizagem. Às vezes, uma configuração é escolhida para ser um hiperparâmetro pois o algoritmo não é capaz de aprender devido à difícil otimização requerida (MURPHY, 2012).

Mais frequentemente, a configuração deve ser um hiperparâmetro porque não é apropriado aprender o mesmo no conjunto de treinamento. Isso se aplica a todos os hiperparâmetros que controlam a capacidade do modelo. Se aprendidos no conjunto de treinamento, tais hiperparâmetros sempre escolheriam a capacidade máxima possível do modelo, resultando em *overfitting*. Para resolver este problema, precisamos de um conjunto de validação com amostras que o algoritmo de treinamento não observa (MURPHY, 2012).

É importante que as amostras de teste não sejam usadas de forma alguma para

fazer escolhas sobre o modelo, incluindo seus hiperparâmetros. Por esse motivo, nenhuma amostra do conjunto de teste pode ser usada no conjunto de validação. Portanto, sempre construímos o conjunto de validação a partir dos dados de treinamento. Especificamente, dividimos os dados de treinamento em dois subconjuntos separados. Um desses subconjuntos é usado para aprender os parâmetros (MURPHY, 2012). O outro subconjunto é o nosso conjunto de validação, usado para estimar o erro de generalização durante ou após o treinamento, permitindo que os hiperparâmetros sejam atualizados de acordo (CHARNIAK, 2019).

O subconjunto de dados usado para aprender os parâmetros ainda é normalmente chamado de conjunto de treinamento, embora possa ser confundido com o conjunto maior de dados usado para todo o processo de treinamento. O subconjunto de dados usado para orientar a seleção de hiperparâmetros é chamado de conjunto de validação. Normalmente, usa-se cerca de 80% dos dados de treinamento para o treinamento em si e 20 % para validação. Após a conclusão de toda a otimização de hiperparâmetros, o erro de generalização pode ser estimado usando o conjunto de teste (MURPHY, 2012).

Dividir o conjunto de dados (do inglês *dataset*) em um conjunto de treinamento fixo e um conjunto de teste fixo pode ser problemático se resultar em um conjunto de teste pequeno. Um pequeno conjunto de testes implica em incerteza estatística em torno do erro de teste médio estimado, tornando difícil afirmar que o algoritmo A funciona melhor do que o algoritmo B na tarefa dada. Quando o *dataset* tem centenas de milhares de exemplos ou mais, esse não é um problema sério. Quando o *dataset* é muito pequeno, procedimentos alternativos permitem usar todos os dados na estimativa do erro médio do teste, a custo do aumento da demanda computacional (GOODFELLOW; BENGIO; COURVILLE, 2017).

Esses procedimentos são baseados na ideia de repetir o treinamento e testar a computação em diferentes subconjuntos ou divisões escolhidos aleatoriamente do *dataset* original. O mais comum deles é o procedimento de validação cruzada *k-fold*, no qual uma partição do *dataset* é formada dividindo-o em k subconjuntos não sobrepostos. O erro de teste pode então ser estimado tomando o erro de teste médio em k tentativas(GOODFELLOW; BENGIO; COURVILLE, 2017).

2.4 Aprendizado profundo e redes neurais

O aprendizado profundo, ou *deep learning*, é um subconjunto do aprendizado de máquina (Figura 3). Na indústria, o aprendizado profundo é usado para resolver tarefas práticas em uma variedade de campos, como visão computacional (imagem), processamento de linguagem (texto) e reconhecimento automático de fala (áudio). Resumindo, o aprendizado profundo é um subconjunto de métodos na caixa de ferramentas do aprendi-

zado de máquina, principalmente usando redes neurais artificiais, que são uma classe de algoritmo vagamente inspirada no cérebro humano (TRASK, 2019). O termo "profundo" se refere a quantidade de camadas envolvidas no processo de treinamento de uma rede neural (CHOLLET, 2018).

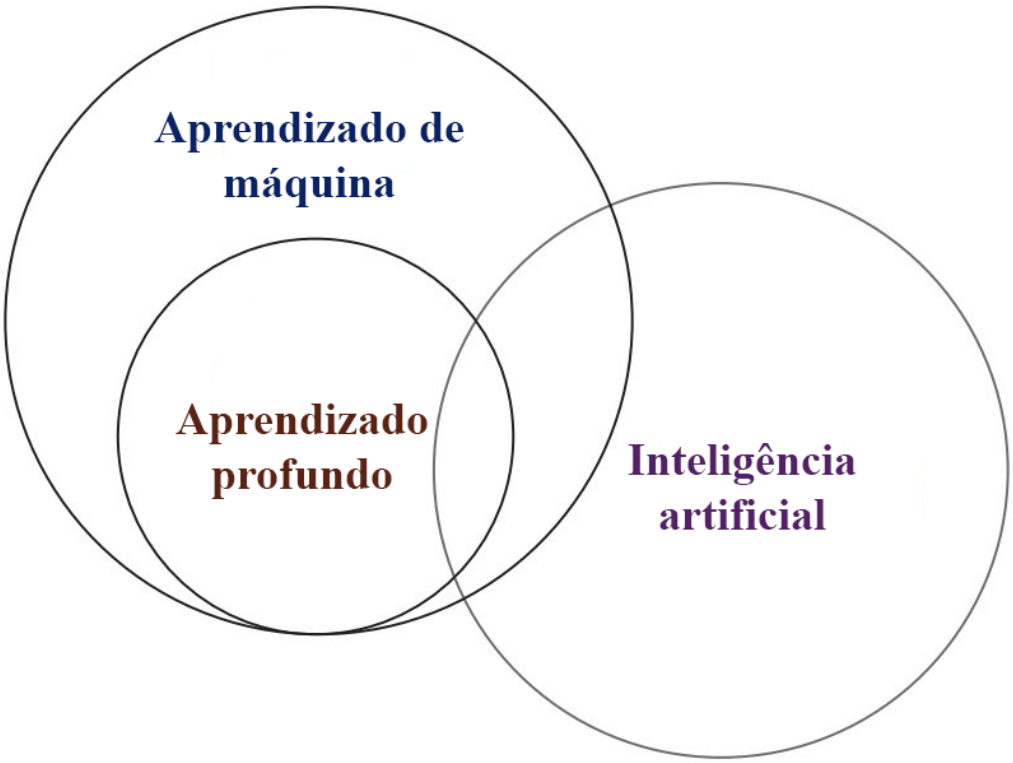


Figura 3 – Inteligência artificial, aprendizado de máquina e aprendizado profundo. Adaptado de: (TRASK, 2019)

O neurônio McCulloch-Pitts (MCCULLOCH; PITTS, 1943) foi um dos primeiros modelos de função cerebral artificial. Este modelo linear podia reconhecer duas categorias diferentes de entradas testando se $f(x, w)$ é positivo ou negativo. Para que o modelo correspondesse à definição desejada das categorias citadas, os pesos precisavam ser definidos corretamente. Esses pesos podiam ser definidos pelo operador humano.

Um dos modelos precursores do surgimento do aprendizado profundo foi o *perceptron*, postulado em 1958 por Rosenblat (ROSENBLATT, 1958) (Figura 4). A teoria foi desenvolvida para um sistema nervoso hipotético, ou máquina, chamado de *perceptron*, termo muito utilizado ainda hoje para definir algoritmos de aprendizagem supervisionada com classificadores binários (GOODFELLOW; BENGIO; COURVILLE, 2017). Logo após surgiu o ADALINE (*Adaptive Linear Element*) que retornava o próprio valor de $f(x)$ para prever um número real (WIDROW, 1960). Durante muito tempo foi utilizado o termo ANN (*Artificial Neural Networks*) para definir o que hoje chamamos de aprendizado profundo.

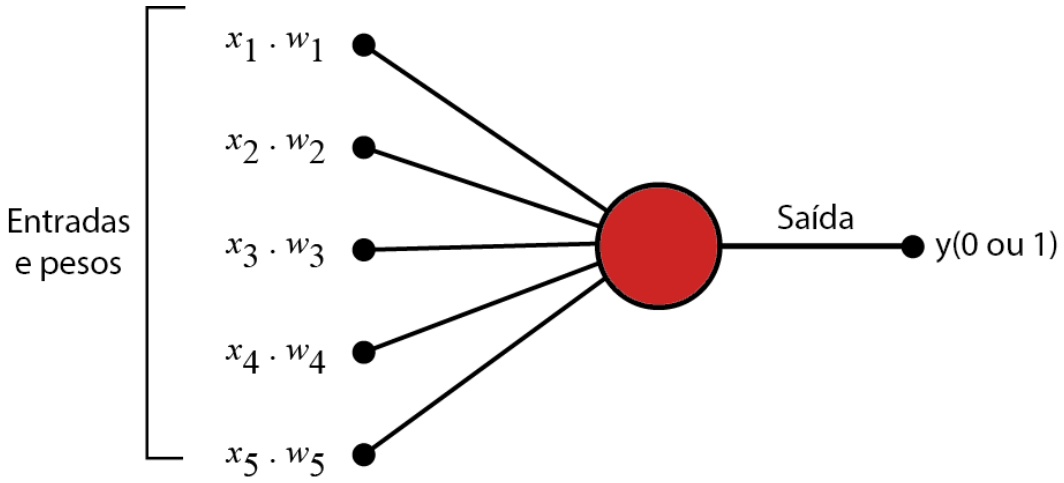


Figura 4 – Estrutura de um *perceptron*. Adaptado de: (MINSKY; PAPERT, 1969)

Por mais que esses conceitos já existam há muitos anos, apenas recentemente o *deep learning* ganhou força e popularidade devido ao aumento expressivo do poder computacional para executar códigos pesados e de alta profundidade. Também pelo fato de que os dados são mais acessíveis e estão em maior abundância que antigamente, assim como os algoritmos que também estão mais robustos e bem implementados (CHOLLET, 2018).

2.4.1 Redes de *Feedforward*

Redes neurais *feedforward* (FNNs ou *Feedforward Neural Networks*), ou *perceptrons* de multicamada (MLPs ou *multilayer perceptrons*), são os modelos de *deep learning* quintessenciais. O objetivo de uma rede *feedforward* é aproximar alguma função f^*. Por exemplo, para um classificador, $y = f^*(x)$ mapeia uma entrada x para uma categoria y. Esta rede neural define um mapeamento $y = f(x;\theta)$ e aprende o valor dos parâmetros θ que resultam na melhor aproximação de função (GOODFELLOW; BENGIO; COURVILLE, 2017).

Esses modelos são chamados de *feedforward* porque a informação flui através da função que está sendo avaliada de x, através dos cálculos intermediários usados para definir f e, finalmente, para a saída y. Não há conexões de *feedback* nas quais as saídas do modelo são realimentadas. Quando as FNNs são estendidas para incluir conexões de *feedback*, elas são chamadas de redes neurais recorrentes (RNNs ou *Recurrent Neural Networks*) (CHOLLET, 2018). As FNNs são de extrema importância para os profissionais de aprendizado de máquina. Elas formam a base de muitas aplicações comerciais importantes. Por exemplo, as redes convolucionais usadas para reconhecimento de objetos a partir de fotos são um tipo especializado de rede *feedforward* (GOODFELLOW; BENGIO; COURVILLE, 2017).

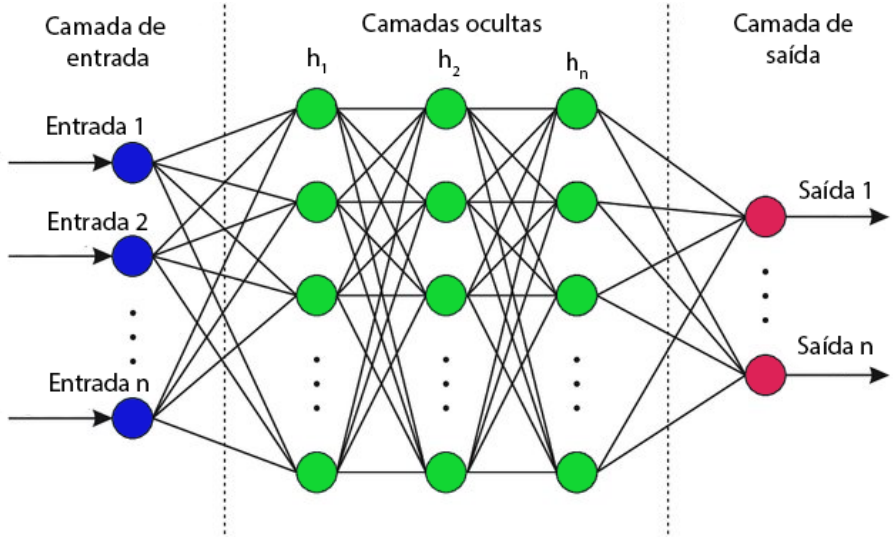


Figura 5 – Estrutura de uma FNN.

Como podemos ver na Figura 5, uma FNN é composta de uma camada de entrada (passagem), uma ou mais camadas de processamento, chamadas de camadas ocultas, e uma camada final chamada de camada de saída. As camadas próximas à camada de entrada são geralmente chamadas de camadas inferiores e as próximas às saídas são geralmente chamadas de camadas superiores. Cada camada, exceto a camada de saída, inclui um neurônio de polarização e está totalmente conectada à próxima camada (GéRON, 2019).

2.4.2 Funções de ativação

Quando todos os neurônios em uma camada estão conectados a todos os neurônios da camada anterior (ou seja, seus neurônios de entrada), a camada é chamada de camada totalmente conectada (*fully connected layer*) ou camada densa (*dense layer*). Para calcular a saída de uma camada densa temos a seguinte equação (GéRON, 2019):

$$h_{\mathbf{W},\mathbf{b}}(\mathbf{X}) = \phi(\mathbf{XW}+\mathbf{b}). \tag{2.3}$$

Na Equação 2.3, $\mathbf{X}$ representa a matriz das características de entrada, possui uma linha por instância e uma coluna por característica. A matriz peso $\mathbf{W}$ contém todos os pesos de conexão, exceto aqueles do neurônio de polarização. Tem uma linha por neurônio de entrada e uma coluna por neurônio artificial na camada. O vetor de polarização $\mathbf{b}$ contém todos os pesos de conexão entre o neurônio de polarização e os neurônios artificiais. A função ϕ é chamada de função de ativação (*activation function*) (GéRON, 2019).

Sem uma função de ativação, a camada densa consistiria em duas operações lineares: um produto escalar e uma adição. Portanto, a camada só poderia aprender transformações lineares dos dados de entrada e este espaço hipotético é muito restrito e não

se beneficiaria de várias camadas de representações, pois uma pilha profunda de camadas lineares ainda implementaria uma operação linear. Para obter acesso a um espaço hipotético muito mais rico que se beneficiaria de representações profundas, você precisa de uma não linearidade, ou função de ativação (CHOLLET, 2018).

As funções de ativação mais comumente usadas são: função logística ou sigmóide (sig), tangente hiperbólica (tanh), unidade linear retificada (ReLU) e Softmax (MATTMANN, 2021). Seguem as equações para cada função de ativação citada, respectivamente: $\sigma(z) = 1/(1 + \exp(-z))$, $\tanh(z) = 2\sigma(2z) - 1$, $ReLU(z) = max(0, z)$ e $\sigma(z)_i = \exp^{z_i} / (\sum_{j=1} \exp^{z_j})$ (GéRON, 2019).

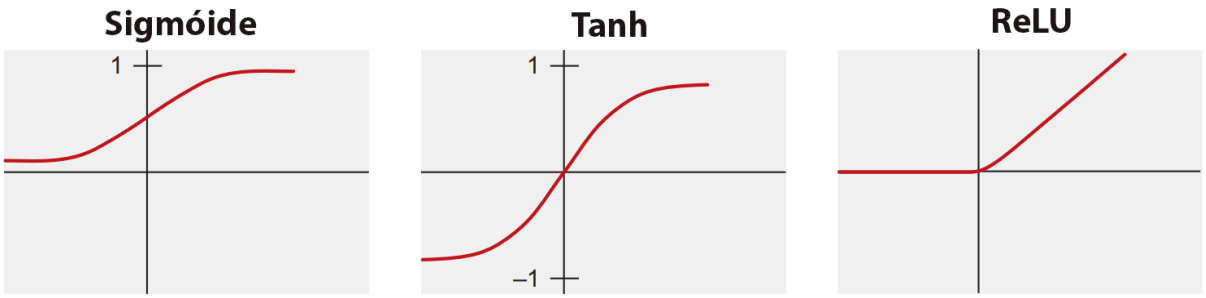


Figura 6 – Representação gráfica das três funções de ativação. Adaptado de: (MATTMANN, 2021)

2.4.3 Função de perda de entropia cruzada binária

As funções de perda mostram como a previsão do algoritmo se desvia do resultado real. Existem muitas funções para encontrar a perda com base no valor previsto e real, dependendo do problema. Uma dessas funções se chama função de perda de entropia cruzada, também chamada de perda logarítmica, perda de log ou perda logística. Cada probabilidade de classe prevista é comparada com a saída verdadeira de classe: 0 ou 1. Uma pontuação/perda é calculada de formar a penalizar a probabilidade com base em quão distante ela está do valor real esperado. A penalidade é logarítmica por natureza, resultando em uma grande pontuação para grandes diferenças perto de 1 e pequena pontuação para pequenas diferenças tendendo a 0 (GOODFELLOW; BENGIO; COURVILLE, 2017).

A perda de entropia cruzada, assim como outras funções de perda, é usada para ajustar os pesos do modelo durante o treinamento. O objetivo é minimizar a perda, ou seja, quanto menor a perda, melhor o modelo. Um modelo ideal não tem perda de entropia cruzada (GOODFELLOW; BENGIO; COURVILLE, 2017). A entropia cruzada binária (BCE, *Binary Cross-Entropy*) é frequentemente calculada como a entropia cruzada média em todos as amostras de dados. Na Equação 2.4 temos N pontos de dados onde t_j é o valor alvo correspondente, p_j é o j-ésimo valor escalar na saída do modelo (GéRON, 2019):

$$BCE = -\frac{1}{N}\left[\sum_{j=1}^{N}[t_j \log(p_j) + (1 - t_j)\log(1 - p_j)]\right]. \tag{2.4}$$

2.5 Conceitos de treinamento

2.5.1 Descida de gradiente estocástico

O algoritmo de descida de gradiente estocástico (*Stochastic Gradient Descent* ou SGD) é um algoritmo de otimização que estima o gradiente de erro para o estado atual do modelo usando amostras do *dataset* de treinamento e, em seguida, atualiza os pesos do modelo usando o algoritmo de retropropagação de erros. A quantidade de atualização dos pesos durante o treinamento é chamada de taxa de aprendizagem (*learning rate*) (GOODFELLOW; BENGIO; COURVILLE, 2017).

A taxa de aprendizagem é um hiperparâmetro configurável usado no treinamento de redes neurais que possui um pequeno valor, muitas vezes na faixa entre 0 e 1. A taxa de aprendizagem controla a rapidez com que o modelo se adapta ao problema. Taxas menores requerem mais épocas (mais conhecida como *epochs*) devido às mudanças menores feitas nos pesos a cada atualização, enquanto taxas de aprendizagem maiores resultam em mudanças rápidas e requerem menos períodos de treinamento. Uma taxa de aprendizagem muito grande pode fazer com que o modelo convirja muito rapidamente para uma solução abaixo do ideal, enquanto uma taxa de aprendizado muito pequena pode fazer com que o processo pare (MATTMANN, 2021).

Existem dois otimizadores que se originaram da SGD, o AdaGrad e o RMSProp. O algoritmo AdaGrad, adapta individualmente as taxas de aprendizagem de todos os parâmetros do modelo, escalando-os de forma inversamente proporcional à raiz quadrada da soma de todos os valores quadrados do gradiente (DUCHI; HAZAN; SINGER, 2011) . O algoritmo RMSProp modifica o AdaGrad para ter um melhor desempenho na configuração não convexa, alterando o acúmulo de gradiente em uma média móvel exponencialmente ponderada (GOODFELLOW; BENGIO; COURVILLE, 2017).

2.5.2 Adam

Adam (derivado de *Adaptive Moment Estimation*) é um algoritmo de otimização que pode ser usado em vez do procedimento clássico de descida de gradiente estocástico para atualizar os pesos da rede com base nos dados de treinamento. O algoritmo calcula taxas de aprendizagem adaptativas individuais para diferentes parâmetros baseado nas estimativas dos gradientes. O Adam se baseia no AdaGrad no que se trata de taxa de aprendizagem variável com base na mudança de parâmetros e se baseia no RMSProp pois

o algoritmo calcula uma média móvel exponencial do gradiente e também do gradiente quadrado, o que torna o Adam bastante poderoso e veloz (KINGMA; BA, 2017).

2.5.3 Tamanho do lote e épocas

O tamanho do lote (*batch size*) é um hiperparâmetro que define o número de amostras a serem trabalhadas antes de atualizar os parâmetros internos do modelo. Pense em um lote como um *loop* **for** iterando sobre uma ou mais amostras e fazendo previsões. No final do lote, as previsões são comparadas às variáveis de saída esperadas e um erro é calculado. A partir desse erro, o algoritmo de atualização é usado para melhorar o modelo, por exemplo, descer ao longo do gradiente de erro. Um *dataset* de treinamento pode ser dividido em um ou mais lotes (GéRON, 2019).

O número de épocas é um hiperparâmetro que define o número de vezes que o algoritmo de aprendizado rodará em todo o *dataset* de treinamento. Uma época (*epoch*) é composta por um ou mais lotes e permite que cada amostra no *dataset* de treinamento atualize os parâmetros do modelo ao menos uma vez. Por exemplo, se o meu modelo está sendo treinado utilizando 10 épocas com o tamanho do lote igual a 128, isso significa que meu algoritmo irá rodar por todo o *dataset* de treinamento 10 vezes, e durante cada etapa de treinamento, o *dataset* está dividido em 128 lotes com amostras aleatórias (GéRON, 2019).

2.6 Regularização de redes neurais

Um problema central no aprendizado de máquina é como fazer um algoritmo que terá um bom desempenho não apenas nos dados de treinamento, mas também em novas entradas. Muitas estratégias usadas no aprendizado de máquina são projetadas explicitamente para reduzir o erro de teste, possivelmente às custas de um aumento no erro de treinamento. Essas estratégias são conhecidas coletivamente como regularização (GOODFELLOW; BENGIO; COURVILLE, 2017).

2.6.1 Aumento de dados

O aumento de dados (Figura 7) adota a abordagem de gerar mais dados de treinamento a partir de amostras existentes, aumentando-as por meio de uma série de transformações aleatórias que geram imagens distintas porém úteis para etapa de treinamento. Isso reduz o *overfitting*, tornando esta uma técnica de regularização (CHOLLET, 2018).

Por exemplo, pode-se deslocar, girar e redimensionar levemente cada imagem no conjunto de treinamento em vários valores e adicionar as imagens resultantes ao conjunto de treinamento. Isso força o modelo a ser mais tolerante às variações na posição, orientação

e tamanho dos objetos nas imagens. Para um modelo que é mais tolerante a diferentes condições de iluminação, pode-se gerar de forma semelhante muitas imagens com vários contrastes. No geral, também é possível virar as imagens horizontalmente (exceto para texto e outros objetos assimétricos) (GéRON, 2019).

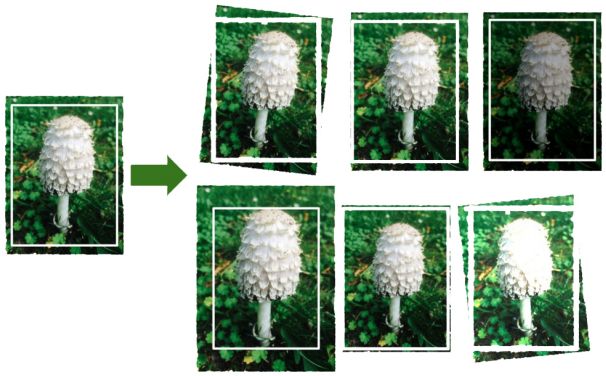


Figura 7 – Exemplo de aumento de dados. Fonte: (GéRON, 2019)

2.6.2 Parada antecipada

Ao treinar grandes modelos com capacidade de representação larga o suficiente, frequentemente observa-se que o erro de treinamento diminui continuamente com o tempo, mas o erro do *set* de validação começa a aumentar novamente em um dado momento. Cada vez que o erro no *set* de validação diminui, armazena-se uma cópia dos parâmetros do modelo. Quando o algoritmo de treinamento termina, retornam-se os parâmetros que obtiveram menor erro durante o treinamento. O algoritmo termina quando nenhum parâmetro melhora em relação ao menor erro de validação registrado para um número pré-especificado de iterações. Essa estratégia é conhecida como parada antecipada, ou *early stopping*. É provavelmente a forma de regularização mais usada no aprendizado profundo. Sua popularidade se deve tanto à sua eficácia quanto à sua simplicidade (GOODFELLOW; BENGIO; COURVILLE, 2017).

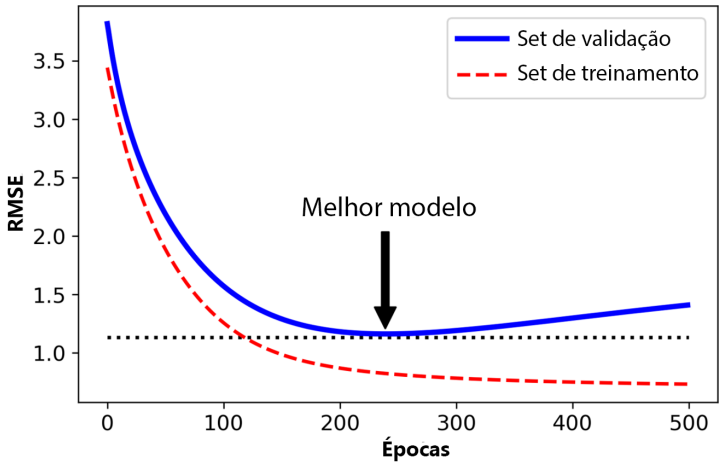


Figura 8 – Exemplo de *early stopping*. Adaptado de: (GéRON, 2019)

Pode-se observar na Figura 8 que conforme as épocas passam, o algoritmo aprende e seu erro de predição (RMSE) no conjunto de treinamento diminui, junto com seu erro de predição no conjunto de validação. Depois de um tempo, o erro de validação para de diminuir e começa a aumentar. Isso indica que o modelo começou a sobrecarregar com os dados de treinamento. Com a parada antecipada, você simplesmente interrompe o treinamento assim que o erro de validação atinge o mínimo (GéRON, 2019).

2.6.3 *Ensemble*

Ensemble (Figura 9) é uma técnica para reduzir o erro de generalização pela combinação de vários modelos (BREIMAN, 1996). A ideia é treinar vários modelos diferentes separadamente e, em seguida, fazer com que todos os modelos votem na saída para amostras de teste. Este é um exemplo de uma estratégia geral em aprendizado de máquina chamada média de modelo (GOODFELLOW; BENGIO; COURVILLE, 2017).

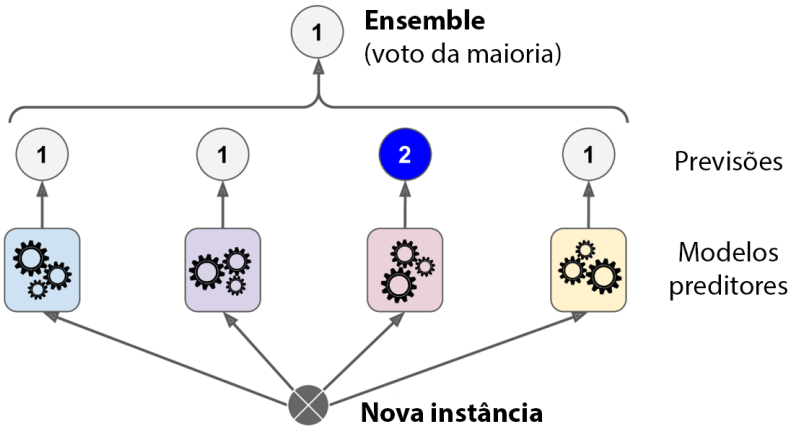


Figura 9 – Exemplo de *ensemble*. Adaptado de: (GéRON, 2019)

A razão pela qual a média do modelo funciona é que modelos diferentes geralmente não cometerão todos os mesmos erros no conjunto de teste (GOODFELLOW; BENGIO; COURVILLE, 2017). Surpreendentemente, esse classificador de votação geralmente atinge uma precisão maior do que o melhor classificador do conjunto. Na verdade, mesmo se cada classificador for de aprendizado fraco, o conjunto ainda pode obter um aprendizado forte (alcançando alta precisão), desde que haja um número suficiente de aprendizados fracos e eles sejam suficientemente diversos (GéRON, 2019).

2.6.4 *Dropout*

O *Dropout* (SRIVASTAVA *et al.*, 2014), aplicado a uma camada, consiste em eliminar aleatoriamente (modificando para zero) uma série de características de saída das camadas durante o treinamento. Digamos que uma determinada camada normalmente retornaria um vetor $[0.2, 0.5, 1.3, 0.8, 1.1]$ para uma determinada amostra de entrada durante

o treinamento. Após aplicar o *Dropout*, este vetor terá algumas entradas zero distribuídas aleatoriamente: por exemplo, $[0, 0.5, 1.3, 0, 1.1]$ (CHOLLET, 2018).

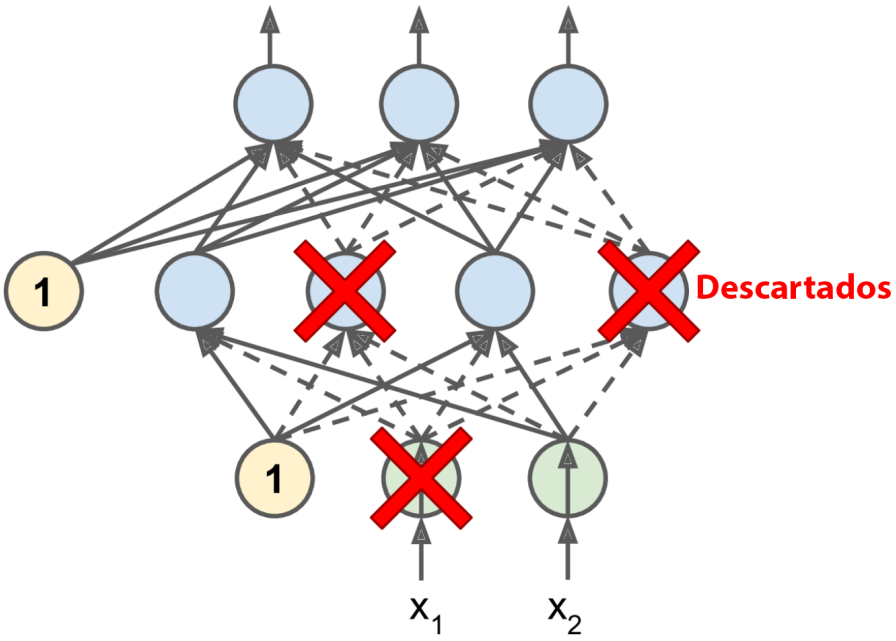


Figura 10 – Exemplo de *Dropout*. Adaptado de: (GéRON, 2019)

É um algoritmo bastante simples: a cada etapa de treinamento, cada neurônio (incluindo os neurônios de entrada, mas sempre excluindo os de saída) tem uma probabilidade p de ser temporariamente "descartado", como podemos ver na Figura 10, o que significa que será totalmente ignorado durante esta etapa de treinamento, mas pode estar ativo durante a próxima etapa. O hiperparâmetro p é chamado de taxa de abandono (*dropout rate*) e é normalmente definido entre 10% e 50%: próximo a $20 - 30\%$ nas redes neurais recorrentes e próximo a $40 - 50\%$ nas redes neurais convolucionais. Após o treinamento, os neurônios não são mais descartados (GéRON, 2019).

2.7 Redes Neurais Convolucionais

Redes neurais convolucionais, também popularmente conhecidas como CNNs ou ConvNets, foram apresentadas inicialmente por Lecun *et al.* (LECUN *et al.*, 1989), que deu origem à popular CNN chamada *LeNet* (LECUN *et al.*, 1998). CNNs são um tipo especializado de rede neural para processamento de dados que possui uma topologia semelhante a uma grade. Por exemplo, dados de uma imagem, que podem ser considerados como uma grade $2D$ de *pixels*. As redes convolucionais têm sido extremamente bem-sucedidas em aplicações práticas. O nome "rede neural convolucional" indica que a rede emprega uma operação matemática chamada convolução. A convolução, conforme definido na subseção 2.2.1, é um tipo especializado de operação linear. As redes convolucionais são simples-

mente redes neurais que usam convolução no lugar da multiplicação geral da matriz em pelo menos uma de suas camadas (GOODFELLOW; BENGIO; COURVILLE, 2017).

Na terminologia de rede convolucional, o primeiro fator (no caso da Equação 2.2, a função I) para a convolução é a entrada e o segundo fator (a função w) são os pesos, que são representados como o fator multiplicativo dos filtros (também chamados de *kernels* de convolução) (GOODFELLOW; BENGIO; COURVILLE, 2017). A arquitetura da CNN tem alguns blocos de construção que já foram apresentados, como camadas totalmente conectadas (*dense layers*) e funções de ativação, mas também introduz dois novos blocos de construção: camadas convolucionais e camadas de *pooling* (GéRON, 2019).

2.7.1 Camadas convolucionais

O bloco de construção mais importante de uma CNN é a camada convolucional: os neurônios na primeira camada convolucional não estão conectados a cada *pixel* na imagem de entrada, mas apenas aos *pixels* em seus campos receptivos (Figura 11). Por sua vez, cada neurônio na segunda camada convolucional está conectado apenas a neurônios localizados dentro de um pequeno retângulo na primeira camada. Essa arquitetura permite que a rede se concentre em pequenos recursos de baixo nível na primeira camada e, em seguida, monte-os em recursos maiores de nível superior na próxima camada e assim por diante. Essa estrutura hierárquica é comum em imagens do mundo real, o que é uma das razões pelas quais as CNNs funcionam tão bem para reconhecimento de imagens (GéRON, 2019).

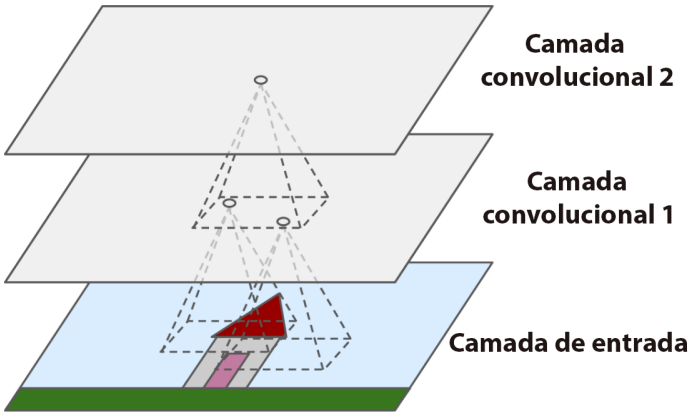


Figura 11 – Camadas de uma CNN com campos receptores locais retangulares. Adaptado de: (GéRON, 2019)

Um recurso essencial em qualquer implementação de rede convolucional é a capacidade de preencher com zeros implicitamente a entrada para torná-la mais larga (Figura 12). Sem esse recurso, a largura da saída diminui em um *pixel* com relação à largura do filtro em cada camada. O preenchimento com zeros (*zero padding*) na entrada permite o controle da largura do filtro e o tamanho da saída de forma independente. Sem o preenchimento com zeros, é necessário reduzir a dimensão espacial da rede neural ou utilizar

filtros de pequena largura, ambos os cenários limitam significativamente o poder da rede (GOODFELLOW; BENGIO; COURVILLE, 2017).

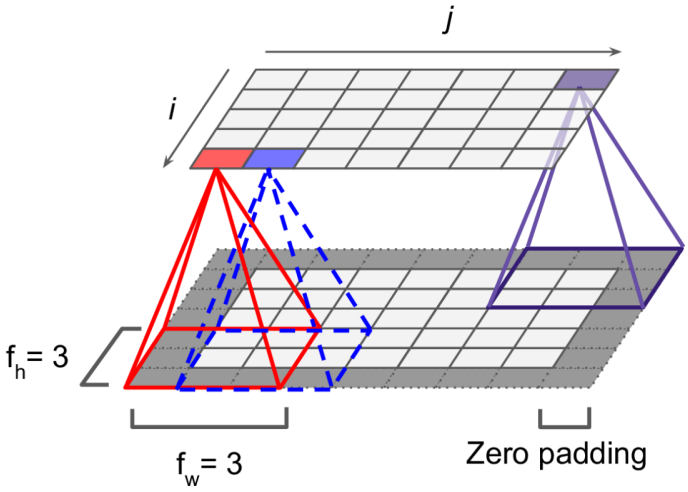


Figura 12 – Utilização de *zero padding*. Fonte: (GéRON, 2019)

O outro fator que pode influenciar o tamanho da saída é o conceito de *strides*. O *stride* consiste no número de *pixels* descolados pelo filtro na matriz de entrada. Quando o *stride* é 2, por exemplo, o filtro irá fazer a varredura na matriz de entrada saltando a distância de 2 *pixel* por vez, como pode ser observado na Figura 13 (CHOLLET, 2018).

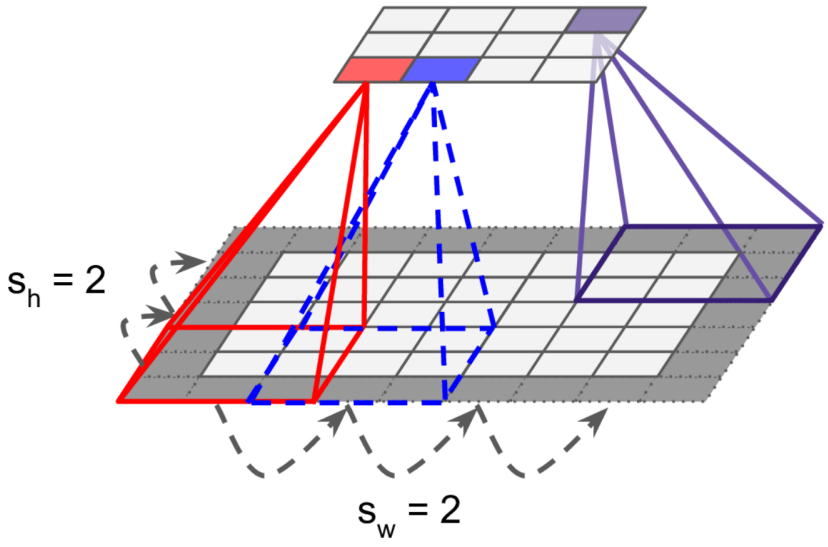


Figura 13 – Representação visual do filtro passando na entrada com *stride* 2. Fonte: (GéRON, 2019)

A saída é pela sobreposição de todos os mapas de características (*feature maps*) obtidos durante o treinamento (Figura 14). Uma camada com neurônios usando o mesmo filtro produz um mapa de características, que destaca as áreas em uma imagem que mais ativam o filtro. Obviamente, não é preciso definir os filtros manualmente, em vez disso,

durante o treinamento, a camada convolucional aprenderá automaticamente os filtros mais úteis para sua tarefa e as camadas seguintes aprenderão a combiná-los em padrões mais complexos (GéRON, 2019).

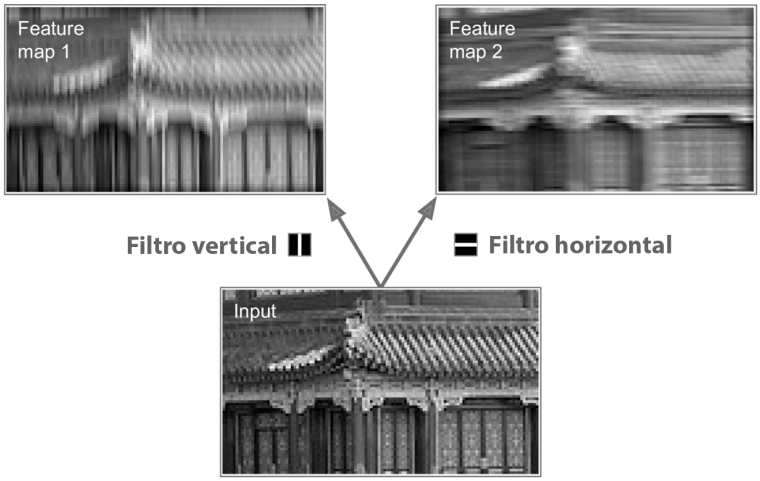


Figura 14 – Aplicando 2 filtros diferentes e obtendo 2 mapas de características distintos. Adaptado de: (GéRON, 2019)

2.7.2 *Pooling*

Pooling significa subamostrar a imagem de entrada a fim de reduzir a carga computacional, o uso de memória e o número de parâmetros (limitando assim o risco de *overfitting*). Assim como nas camadas convolucionais, cada neurônio em uma camada de *pooling* está conectado às saídas de um número limitado de neurônios na camada anterior, localizados dentro de um pequeno campo receptivo retangular. Ainda deve ser definido o tamanho, o *stride* e o tipo de preenchimento (*padding*), como antes. No entanto, um neurônio de *pooling* não tem pesos, tudo o que ele faz é agregar as entradas usando uma função, como função de máximo ou de média (GéRON, 2019). A operação de *pooling* máximo (Figura 15) (ZHOU; CHELLAPPA, 1988), por exemplo, obtêm o valor máximo dentre os *pixels* em uma vizinhança retangular (GOODFELLOW; BENGIO; COURVILLE, 2017).

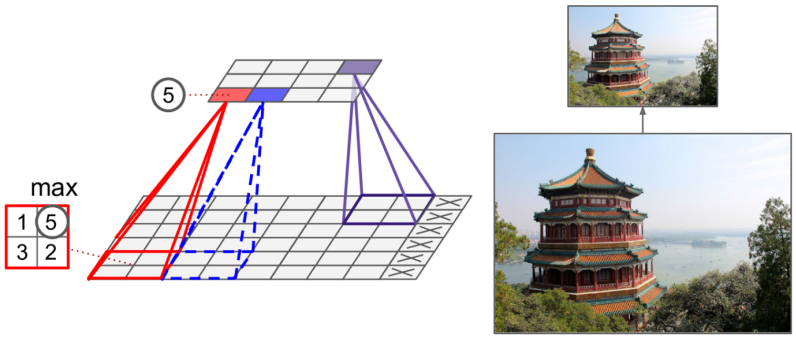


Figura 15 – Camada de *pooling* máximo (*kernel* de *pooling* 2×2, *stride 2*, sem *padding*). Fonte: (GéRON, 2019)

Uma última camada a ser comentada é a camada de achatamento, ou *flatten layer*, conforme pode ser observada na última etapa da Figura 16. Achatar é converter os dados em uma matriz unidimensional para inseri-los na próxima camada. Nivelando a saída das camadas convolucionais para criar um único vetor de características, e o mesmo está conectado ao modelo de classificação final, que é uma camada densa, ou camada totalmente conectada (MATTMANN, 2021).

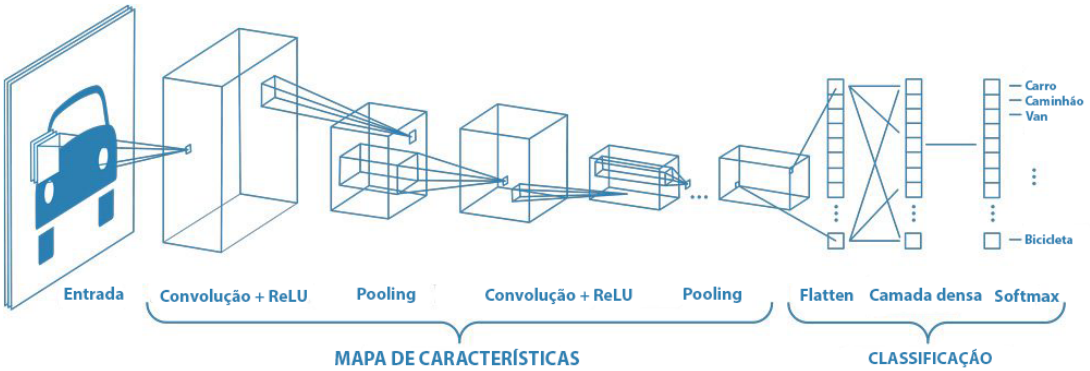


Figura 16 – Exemplo de uma rede com muitas camadas convolucionais. Adaptado de: (MATHWORKS, 2021)

2.7.3 Propriedades estruturais de uma CNN

A convolução aproveita de três ideias importantes que podem ajudar a melhorar um sistema de aprendizado de máquina: interações esparsas, compartilhamento de parâmetros e representações equivariáveis. Além disso, a convolução fornece um meio de trabalhar com entradas de tamanho variável. Camadas de redes neurais tradicionais usam multiplicação de matriz através de uma matriz de parâmetros com um parâmetro separado que descreve a interação entre cada unidade de entrada e saída. Isso significa que cada unidade de saída interage com cada unidade de entrada. Redes convolucionais, no entanto, normalmente têm interações esparsas (também chamadas de conectividade esparsa ou pesos esparsos). Isso é feito tornando o *kernel* menor do que a entrada (GOODFELLOW; BENGIO; COURVILLE, 2017).

O compartilhamento de parâmetros refere-se ao uso do mesmo parâmetro para mais de uma função em um modelo. Em uma rede neural convolucional, cada membro do *kernel* é usado em todas as posições da entrada (exceto talvez alguns *pixels* de borda). O compartilhamento de parâmetros usado pela operação de convolução significa que, em vez de aprender um conjunto separado de parâmetros para cada local, aprende-se apenas um conjunto. A existência do compartilhamento de parâmetros faz com que as camadas tenham uma propriedade chamada representação equivariante. Dizer que uma função é equivariante significa que se a entrada muda, a saída muda da mesma maneira. Especificamente, uma função $f(x)$ é equivariante a uma função g se $f(g(x)) = g(f(x))$. No caso da convolução, se g for qualquer função que interprete a entrada, ou seja, a

desloque, então a função de convolução é equivariante a g (GOODFELLOW; BENGIO; COURVILLE, 2017).

2.8 Arquiteturas de interesse

2.8.1 EfficientNet

Aumentar a dimensão de CNNs é amplamente usado para obter melhor precisão. Por exemplo, a ResNet (HE *et al.*, 2015) pode ser ampliada de ResNet-18 para ResNet-200 usando mais camadas. Recentemente, a GPipe (HUANG *et al.*, 2018) alcançou top-1 no ILSVRC com 84,3% de precisão ao ampliar um modelo básico quatro vezes mais. No entanto, o processo de escalonamento de CNNs nunca foi bem compreendido e atualmente existem muitas maneiras de fazê-lo. A maneira mais comum é aumentar sua profundidade (HE *et al.*, 2015) ou largura (ZAGORUYKO; KOMODAKIS, 2016). Outro método menos comum, mas cada vez mais popular, é aumentar a escala de modelos através da resolução da imagem (HUANG *et al.*, 2018).

É fundamental balancear todas as dimensões da rede, como largura, profundidade e resolução. Surpreendentemente, esse equilíbrio pode ser alcançado simplesmente dimensionando cada uma delas com proporção constante. Com base nesta observação, a EfficientNet utiliza um método de escalonamento composto (Figuras 17 e 18). Ao contrário da prática convencional que dimensiona arbitrariamente esses fatores, o escalonamento composto dimensiona uniformemente a largura, a profundidade e a resolução da rede com um conjunto de coeficientes de dimensionamento fixos (TAN; LE, 2020).

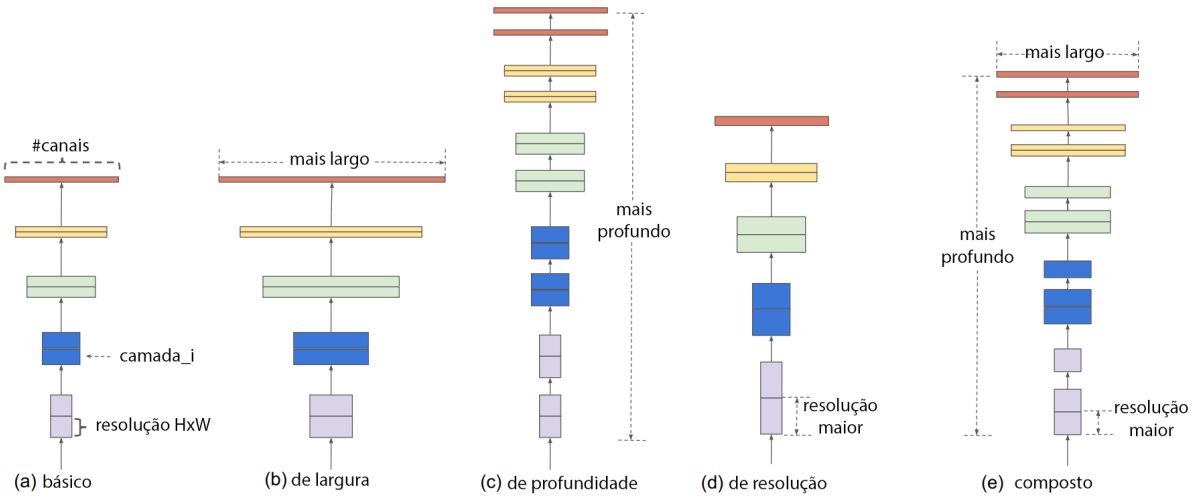


Figura 17 – Diferenças estruturais entre tipos de escalonamento das CNNs. Adaptado de: (TAN; LE, 2020)

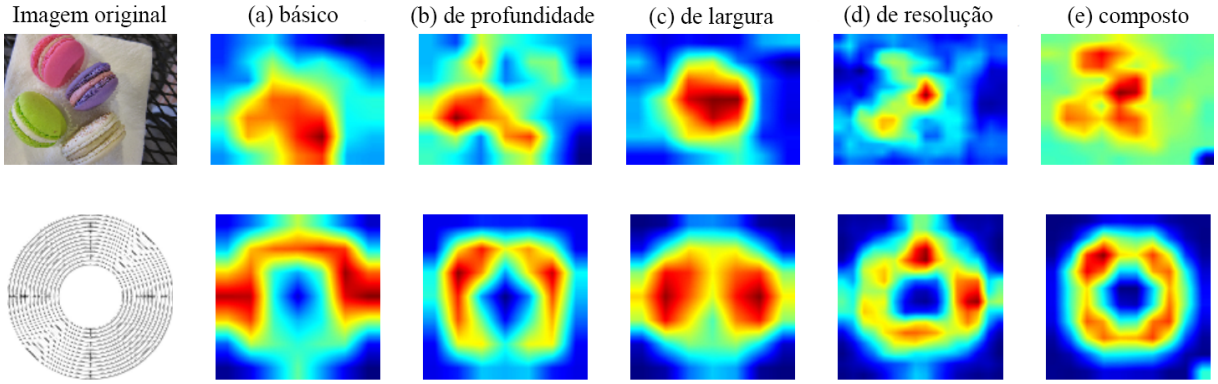


Figura 18 – Mapa de ativação de classe para modelos com métodos de escalonamento diferentes. Adaptado de: (TAN; LE, 2020)

A EfficientNet recebe a nomenclatura EfficientNet-BX, sendo X o número da versão da rede. Até o presente momento existem 8 redes indo de B0 até B7. A Tabela 1 mostra a arquitetura da EfficientNet-B0, seu principal bloco é o MBConv, que também é chamado de bloco invertido residual, foi desenvolvido por Sandler *et al.* (SANDLER *et al.*, 2018) na rede MobileNetV2 e depois aprimorado por Tan *et al.* (TAN *et al.*, 2018) na rede MnasNet. Um bloco invertido residual, às vezes chamado é um tipo de bloco residual usado para modelos de imagem que usa uma estrutura invertida por razões de eficiência. Um bloco residual tradicional tem uma estrutura ampla/estreita/ampla com o número de canais. Em contraste, o MBConv segue uma abordagem estreita/larga/estreita (SANDLER *et al.*, 2018).

Abaixo, podemos ver na Tabela 1 a estrutura da rede B0, ficando evidente a utilização do escalonamento composto, pois tanto a resolução, largura e a profundidade são variáveis dependendo do estágio da rede (TAN; LE, 2020).

Tabela 1 – Estrutura da rede EfficientNet-B0.

Estágio i	Operador $\mathcal{F}_i$	Resolução $H_i \times W_i$	Canais C_i	Camadas L_i
1	Conv3x3	224×224	32	1
2	MBConv1,k3x3	112×112	16	1
3	MBConv6,k3x3	112×112	24	2
4	MBConv6,k5x5	56×56	40	2
5	MBConv6,k3x3	28×28	80	3
6	MBConv6,k5x5	14×14	112	3
7	MBConv6,k5x5	14×14	192	4
8	MBConv6,k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

A Figura 19 apresenta uma divisão em módulos para melhor compreensão de como são estruturadas as EfficientNets, visto que elas podem possuir um número de camadas muito alto, a EfficientNet-B7, por exemplo, possui 813 camadas (TAN; LE, 2020).

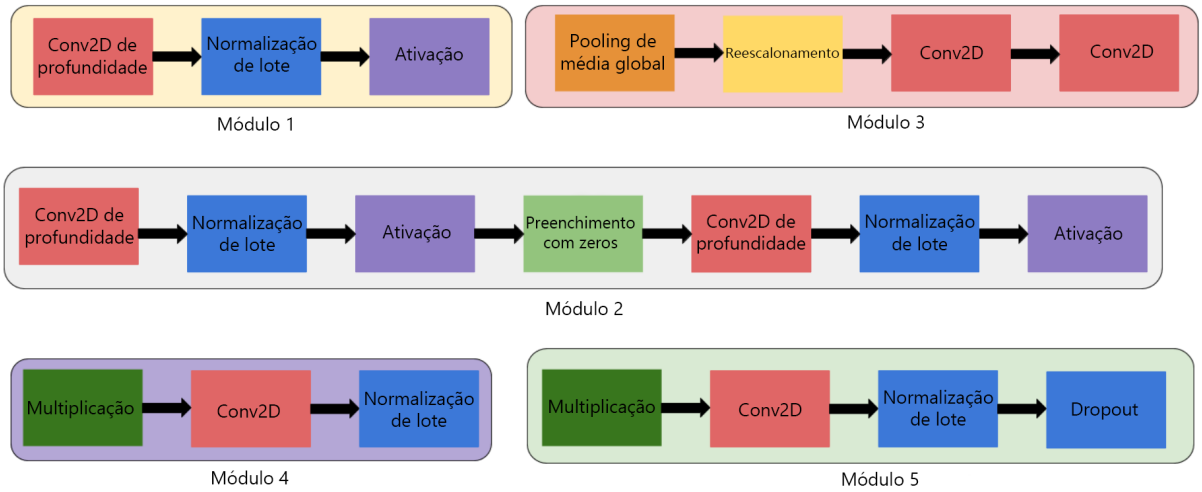


Figura 19 – Módulos que compõem a arquitetura EfficientNet. Diagrama construído com base em (TAN; LE, 2020)

Esses módulos são combinados para formar sub-blocos (Figura 20) que serão usados na descrição das redes (Figuras 21, 22, 23 e 24).

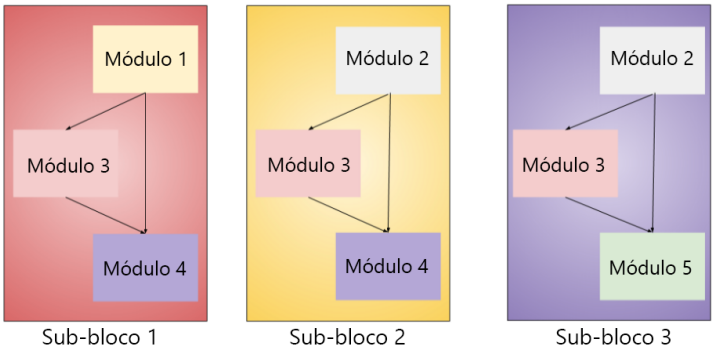


Figura 20 – Sub-blocos utilizando os módulos descritos acima

- **EfficientNet-B4**

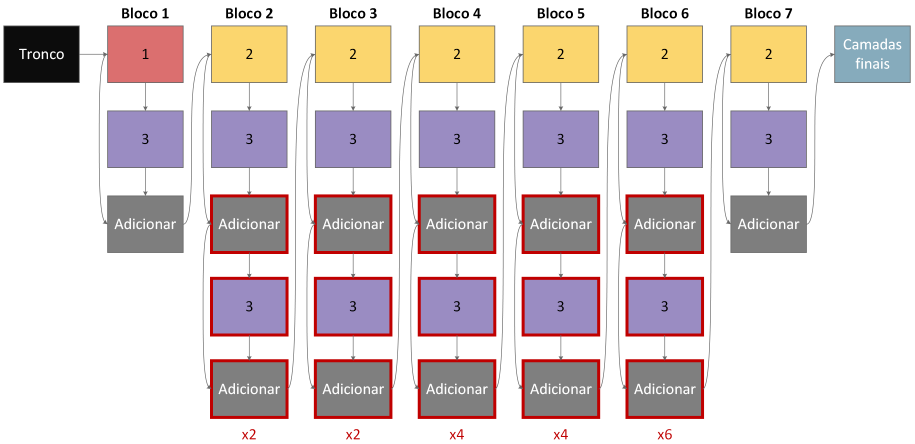


Figura 21 – Composição da EfficientNet-B4.

- **EfficientNet-B5**

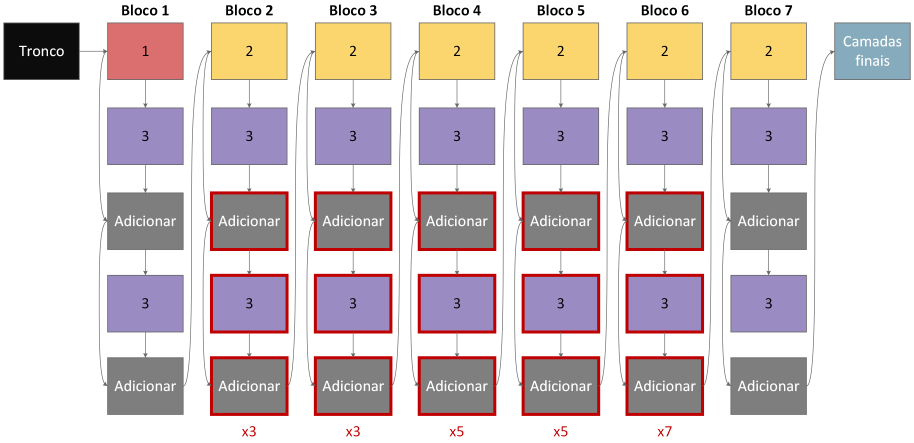


Figura 22 – Composição da EfficientNet-B5.

- **EfficientNet-B6**

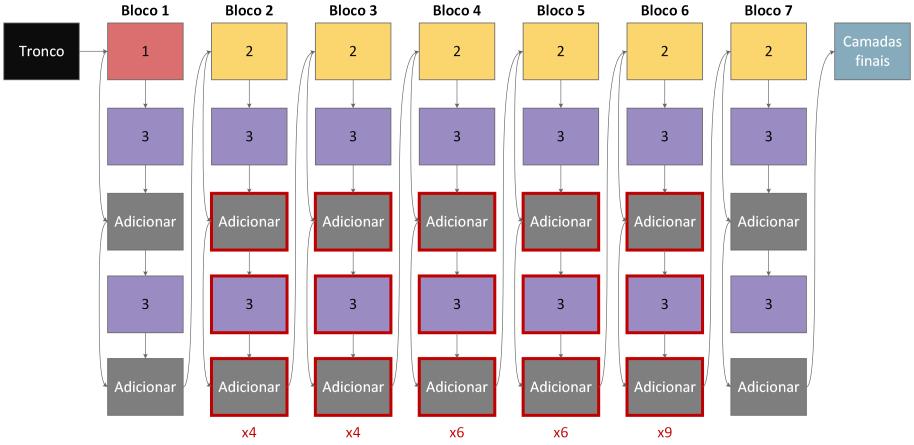


Figura 23 – Composição da EfficientNet-B6.

- **EfficientNet-B7**

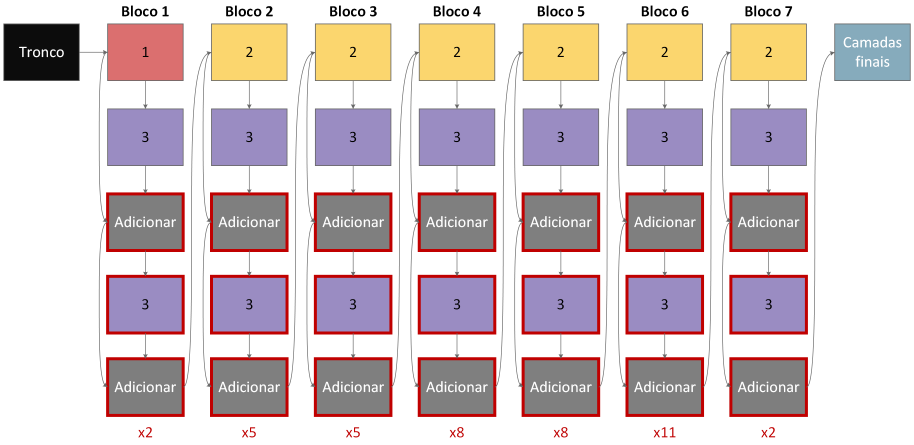


Figura 24 – Composição da EfficientNet-B7.

2.8.2 DenseNet

A arquitetura DenseNet foi proposta por Huang, Liu e Weinberger (HUANG; LIU; WEINBERGER, 2016) juntamente com membros da equipe de IA do Facebook e usa os mesmos conceitos de convoluções, *pooling* e a função de ativação ReLU para funcionar. O detalhe importante e a inovação nesta arquitetura foram os blocos densos (Figura 25). Para preservar a natureza de rede *feedforward*, cada camada obtém entradas adicionais de todas as camadas anteriores e passa seus próprios mapas de características para todas as camadas subsequentes (HUANG; LIU; WEINBERGER, 2016).

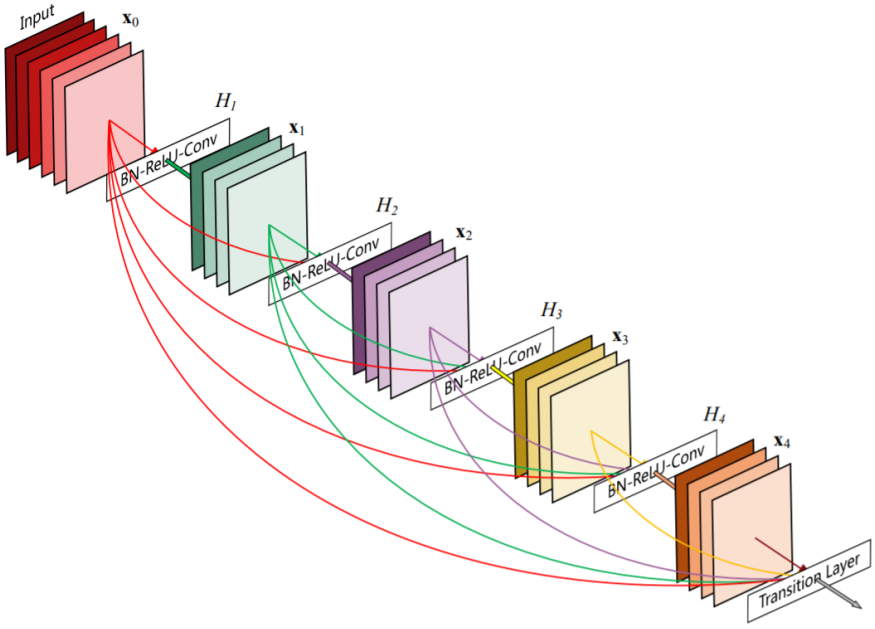


Figura 25 – Bloco denso de 5 camadas. Fonte: (HUANG; LIU; WEINBERGER, 2016)

Crucialmente, ao contrário das ResNets (HE *et al.*, 2015), as características não são combinadas por somatório antes de serem passadas para uma camada, e sim combinadas por concatenação. Consequentemente, a ℓ-ésima camada tem ℓ entradas, consistindo nos mapas de características de todos os blocos convolucionais anteriores. Seus próprios mapas de características são passados para todas as camadas $L - \ell$ subsequentes. Isso introduz conexões $\frac{L(L+1)}{2}$, em vez de apenas L, como nas arquiteturas tradicionais. Por causa de seu padrão de conectividade denso, essa abordagem é referida como Rede Convolucional Densa (DenseNet) (HUANG; LIU; WEINBERGER, 2016).

2.9 Métricas de desempenho

2.9.1 Matriz de confusão

A matriz de confusão é uma tabela que compara como as respostas previstas se comparam às reais. Os itens que são corretamente previstos como positivos são chamados de positivos verdadeiros (PV). Aqueles que são incorretamente previstos como positivos são chamados de falsos positivos (FP). Se o algoritmo acidentalmente predizer que um elemento é negativo quando na realidade é positivo, chamamos essa situação de falso negativo (FN). Finalmente, quando a previsão e a realidade concordam que um item de dados é um rótulo negativo, é chamado de negativo verdadeiro (NV). Uma matriz de confusão permite que você veja facilmente com que frequência um modelo confunde duas classes que está tentando diferenciar (MATTMANN, 2021).

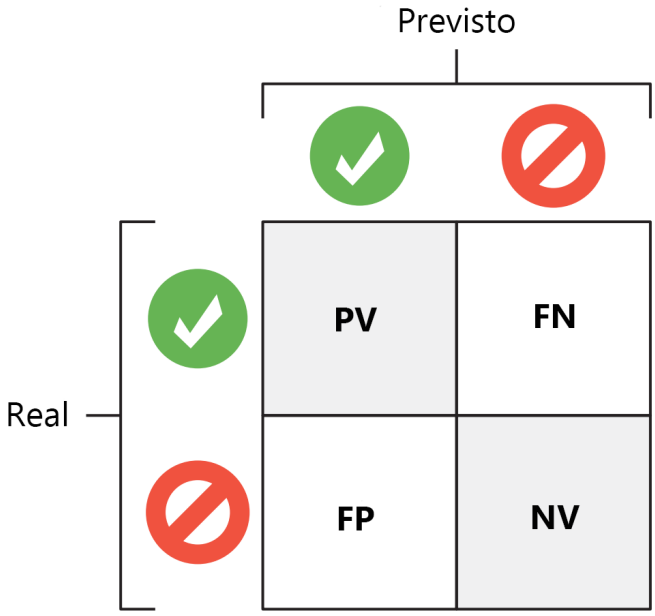


Figura 26 – Matriz de confusão. Adaptado de: (MATTMANN, 2021)

A proporção de positivos verdadeiros para o total de exemplos positivos é a precisão - uma pontuação da probabilidade de uma previsão positiva estar correta. A coluna da esquerda na Figura 26 é o número total de previsões positivas (PV + FP). A proporção de positivos verdadeiros para todos os positivos possíveis é o *recall*, que mede a proporção de verdadeiros positivos encontrados, é uma pontuação de quantos positivos verdadeiros foram previstos com sucesso. A linha superior na Figura 26 é o número total de positivos (PV + FN), as equações para o cálculo de precisão e *recall* se encontram abaixo (MATTMANN, 2021):

$$precisão = \frac{PV}{PV + FP}, \tag{2.5}$$

$$recall = \frac{PV}{PV + FN}. \tag{2.6}$$

2.9.2 Curva ROC

A curva de característica de operação do receptor (ROC, do inglês *Receiver Operating Characteristic*) é uma ferramenta comum usada com classificadores binários (Figura 27. É muito semelhante com a curva precisão/*recall*, mas ao invés disso, a curva ROC traça a taxa de positivo verdadeiro (TPV, outro nome para recall) contra a taxa de falso positivo (TFP, *False Positive Rate*). A TFP é a proporção de ocorrências negativas classificadas incorretamente como positivas e é igual a: 1− a taxa negativa verdadeira (TNV), que é a proporção de instâncias negativas que são classificadas corretamente como negativas. A TNV também é chamada de especificidade. Uma maneira de comparar os classificadores é medir a área sob a curva (AUC, *Area Under the Curve*). Um classificador perfeito terá um ROC AUC igual a 1, enquanto um classificador puramente aleatório terá um ROC AUC igual a 0,5 (GéRON, 2019). A expressão da AUC é dada por:

$$AUC = \frac{\sum_{positivos} k - \frac{n_{pos}(n_{pos}+1)}{2}}{n_{pos} n_{neg}}, \tag{2.7}$$

aonde k é a classificação dos positivos, n_{pos} representa o número de amostras positivas e n_neg o número de amostras negativas (ZHANG; WANG, 2021).

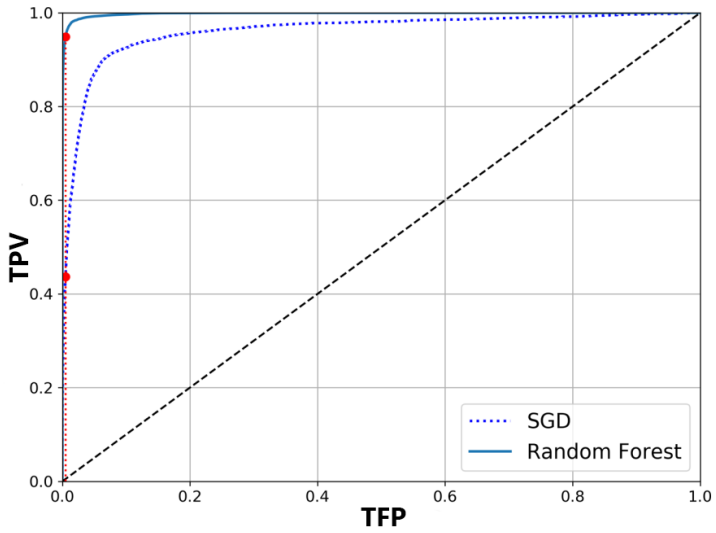


Figura 27 – Exemplo de curva ROC. Fonte: (GéRON, 2019)

2.10 Trabalhos relacionados

Os trabalhos a seguir utilizam métricas de desempenho e *datasets* iguais aos que foram utilizados para treinar a rede deste trabalho (EfficientNet-B7), para efeito de comparação de arquitetura de forma justa e com mesma aplicabilidade.

2.10.1 Karki, Kulkarni e Stranieri (2021).

O trabalho de Karki, Kulkarni e Stranieri (KARKI; KULKARNI; STRANIERI, 2021) faz uma comparação das EfficientNets B4, B5 e B6. Os modelos são treinados com dados da competição ISIC 2020, somados com imagens de melanoma do ISIC 2019 e 2018. O conjunto de teste utilizado é o do ISIC 2020, que possui 11000 imagens. O trabalho também apresenta a pontuação pública, que utiliza 30% do conjunto de teste, e a pontuação privada, que utiliza os outros 70%. Essa pontuação pode ser automaticamente obtida ao enviar um arquivo de formato .csv com os dados previstos pela rede neural para a plataforma de avaliação no Kaggle, que automaticamente compara com o valor real de cada imagem no banco de teste.

Para preprocessamento e aumento de dados do conjunto de treinamento, as imagens foram giradas, cortadas, ampliadas e deslocadas aleatoriamente. Também foram feitos *flips* horizontais aleatórios, alteração de contraste, saturação, brilho e matiz (KARKI; KULKARNI; STRANIERI, 2021). Todos os modelos foram treinados em TPUs Kaggle (KAGGLE, 2021). Os modelos foram treinados usando validação cruzada em dobras (k-*fold*) com $k = 5$. Os dados são divididos de forma que a proporção de imagens malignas para benignas em cada dobra seja igual à proporção de malignas para benignas no conjunto de dados geral (KARKI; KULKARNI; STRANIERI, 2021).

Todas as EfficientNets são treinadas utilizando diferentes resoluções (Tabela 2) e os melhores modelos, baseados na pontuação ROC AUC para cada dobra, são selecionados e é feito *Ensemble* com os mesmos. O otimizador Adam é usado durante o treinamento com uma taxa de aprendizagem personalizada (KARKI; KULKARNI; STRANIERI, 2021).

Tabela 2 – Desempenho dos modelos através de *Ensemble*. Fonte: (KARKI; KULKARNI; STRANIERI, 2021)

Rede	Resolução	Pontuação privada	Pontuação Pública
B6	512×512	0.9369	0.9337
B6	384×384	0.9373	0.9397
B6	256×256	0.9332	0.9364
B5	512×512	0.9364	0.9353
B5	384×384	0.9363	0.9345
B5	256×256	0.9259	0.9232
B4	512×512	0.9300	0.9306
B4	384×384	0.9347	0.9292
B4	256×256	0.9304	0.9287

2.10.2 Zhang, Wang (2021).

Neste artigo, Zhang e Wang (ZHANG; WANG, 2021) proporam o uso da DenseNet para realizar o reconhecimento do melanoma em fotos de lesões cutâneas. Foi utilizado o modelo DenseNet201, que possui 201 camadas e foi treinado no conjunto de dados ISIC

2020 fornecido pela plataforma de competição Kaggle (KAGGLE, 2021). Os experimentos mostraram que, em comparação com as redes VGG16 e ResNet50, este modelo pode identificar o melanoma com mais eficácia e ajudar os dermatologistas clínicos a melhorar a precisão do diagnóstico do melanoma (ZHANG; WANG, 2021).

A DenseNet201 possui uma taxa de aprendizado de 1^{-4} e utiliza o Adam como otimizador, assim como Karki, Kulkarni e Stranieri (KARKI; KULKARNI; STRANIERI, 2021). A métrica de desempenho utilizada também é a pontuação ROC AUC (Eq. 2.7):

Tabela 3 – Resultado de diferentes modelos usando o *dataset* do ISIC 2020 para classificação de melanoma. Fonte: (ZHANG; WANG, 2021)

Modelos	Pontuação AUC ROC
VGG16	0.891
ResNet50	0.922
DenseNet201	0.925

3 Metodologia

3.1 Base de dados

Cada vez mais *datasets* estão sendo disponibilizados aos pesquisadores. Um problema que certamente deve ser abordado neste contexto consiste na escolha dos *datasets*, que frequentemente são criados para oferecer suporte a tipos específicos de análise, portanto serão utilizados conjuntos específicos para classificação de melanoma. Os *datasets* do ISIC são fornecidos para competições de classificação de melanoma e também utilizados na literatura, como visto em Culell-Dalmau *et al.* (CULLELL-DALMAU *et al.*, 2021), Zhang e Wang (ZHANG; WANG, 2021) e Karki, Kulkarni e Stranieri (KARKI; KULKARNI; STRANIERI, 2021).

Em um primeiro momento, a rede será treinada utilizando apenas o *dataset* do ISIC 2020, em seguida ela será treinada utilizando a junção dos *datasets* ISIC 2020 e ISIC 2019. As imagens serão utilizadas no formato TFRecord.

O formato TFRecord é o mais adequado do TensorFlow para armazenar grandes quantidades de dados e lê-los com eficiência. É um formato binário que contém apenas uma sequência de registros binários de tamanhos variados (cada registro é composto por um comprimento, uma soma de verificação CRC para verificar se o comprimento não foi corrompido, os dados reais e, finalmente, uma soma de verificação CRC para os dados) (GéRON, 2019).

- **ISIC 2020** - O *dataset* de treino (Figura 28) está representado por 2.056 pacientes (20,8% com pelo menos um melanoma, 79,2% com zero melanoma) de três continentes com uma média de 16 lesões por paciente, consistindo em 33.126 imagens dermatoscópicas e 584 (1,8%) melanomas confirmados histopatologicamente em comparação com lesões que mimetizam um melanoma, como nevos, proliferação melanocítica atípica, mancha café com leite, lentigo simples, lentigo solar, ceratose liquenóide e ceratose seborreica. Já o *dataset* de teste contém 10.982 imagens. As imagens de treinamento consistem em 12.743.090 *pixels* em média, mas variam de 307.200 a 24.000.000. Os metadados para cada imagem incluem idade aproximada do paciente no momento da captura da imagem, sexo biológico, local anatômico geral da lesão, número de identificação do paciente anônimo, categoria benigna/maligna e o diagnóstico específico, se disponível, com base em um método de confirmação de *Ground Truth* (GT). Foi lançado sob a licença *Creative Commons* de Atribuição-Não-Comercial 4.0 Internacional (CC BY-NC 4.0) e está permanentemente acessível ao público através do ISIC 2020 (ROTEMBERG *et al.*, 2021).

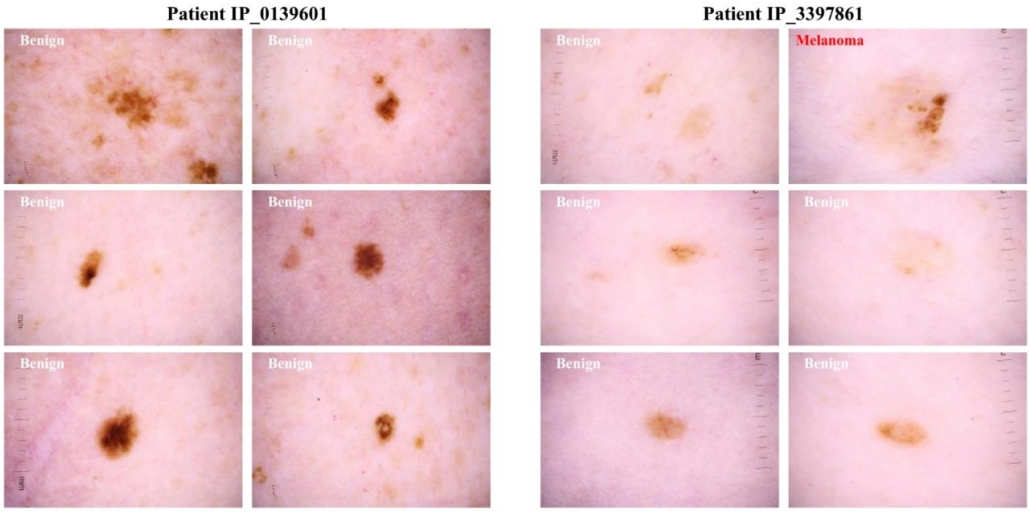


Figura 28 – Exemplo de imagens encontradas no *dataset* do ISIC 2020 (ROTEMBERG *et al.*, 2021).

- **ISIC 2019** - O *dataset* de treinamento do ISIC 2019 (Figura 29) é composto pela junção de 3 outros *datasets*, sendo eles: 1) MSK (CODELLA *et al.*, 2017), que foi utilizado no desafio do ISIC de 2017; 2) BCN20000 (COMBALIA *et al.*, 2019) e 3) HAM10000 (TSCHANDL; ROSENDAHL; KITTLER, 2018). No total, o conjunto do ISIC 2019 contém 25.331 imagens e estão disponíveis para treinamento em 8 categorias diferentes:

  - Melanoma
  - Nevo melanocítico
  - Carcinoma basocelular
  - Ceratose actínica
  - Ceratose benigna (lentigo solar / ceratose seborreica / ceratose semelhante a líquen plano)
  - Dermatofibroma
  - Lesão vascular
  - Carcinoma de células escamosas

  As imagens e metadados do conjunto "ISIC 2019: Treinamento" estão sob a licença *Creative Commons* de Atribuição-Não-Comercial 4.0 Internacional (CC BY-NC 4.0) e está disponível para *download* após cadastro de *e-mail* no site do ISIC 2019 (ISIC, 2019).

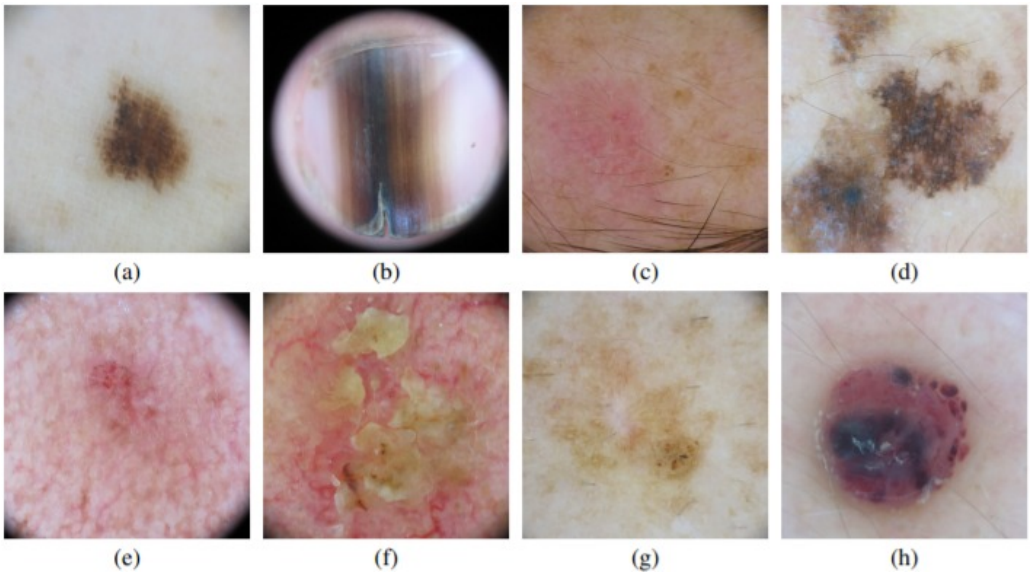
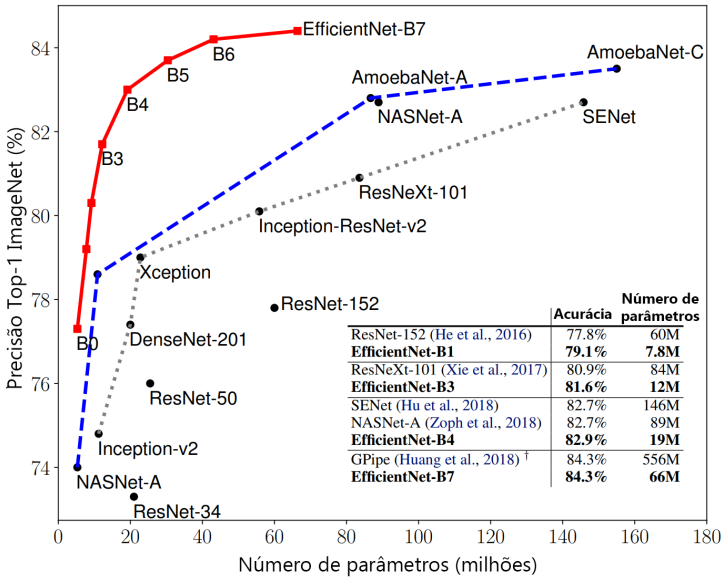


Figura 29 – Amostras do *dataset* BCN20000 correspondendo: (a) nevo, (b) melanoma, (c) carcinoma basocelular, (d) ceratose seborreica, (e) ceratose actínica, (f) carcinoma de células escamosas, (g) dermatofibroma e (h) lesão vascular (COMBALIA *et al.*, 2019).

3.2 Arquitetura utilizada

A arquitetura selecionada para realizar a classificação de melanoma foi a EfficientNet-B7, que atinge o estado da arte com 84.3% de precisão e primeiro lugar no topo do ImageNet (Figura 30), sendo 8,4x menor e 6,1x mais rápida em inferências que as melhores CNNs existentes (TAN; LE, 2020). Foi utilizada a técnica de *transfer learning* para adequar essa CNN para o propósito de classificação de melanoma.



	Acurácia	Número de parâmetros
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

Figura 30 – Precisão no ImageNet vs tamanho do modelo. Adaptado de: (TAN; LE, 2020)

3.2.1 Aumento de dados

Foram feitos treinamentos com e sem aumento de dados e a utilização do mesmo se mostrou eficaz e necessária para aumentar a confiança da rede. As transformações feitas nas imagens do banco de treinamento estão listadas na Tabela 4 a seguir:

Tabela 4 – Transformações utilizadas no banco de treinamento.

Transformações
Flip horizontal e vertical da imagem
Rotação da imagem em até 180^o
Alterações nos níveis de matiz
Aumento nos cabelos
Alterações nos níveis de contraste
Mudanças no brilho e na saturação

Podemos observar na Figura 31, aplicações aleatórias em uma imagem do *dataset* das diversas transformações supracitadas:

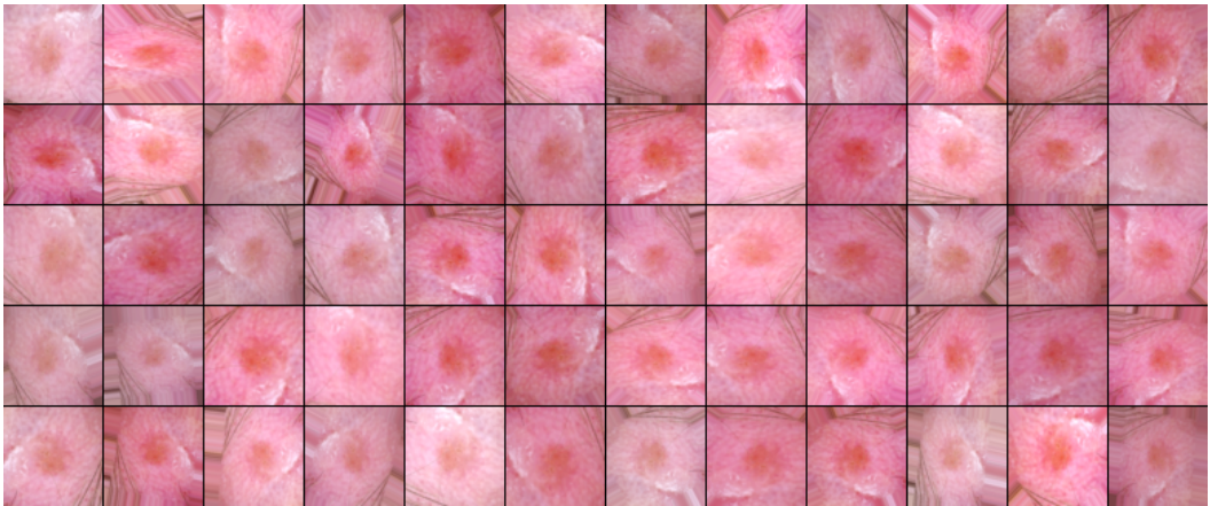


Figura 31 – Aumento de dados aleatórios aplicados em uma imagem do *dataset*.

3.2.2 Hiperparâmetros

O treinamento foi realizado em diferentes resoluções e para cada resolução de entrada há um número diferente de épocas, conforme pode ser observado na Tabela 5. O número de épocas adequado foi determinado utilizando técnicas de parada antecipada.

Tabela 5 – Hiperparâmetros da rede de acordo com a resolução da imagem de entrada.

Hiperparâmetros			
Conjunto	Resolução de entrada	Épocas	Tamanho do lote
2020	256 × 256	13	128
2020	384 × 384	15	128
2020	512 × 512	15	128
2020	768 × 768	20	32
2019-2020	256 × 256	25	128
2019-2020	384 × 384	25	128
2019-2020	512 × 512	12	128
2019-2020	768 × 768	20	32

Foi utilizado o algoritmo otimizador Adam e a função de ativação ReLU para a não-linearidade da rede. O recurso de *Dropout* utilizado foi o de *Coarse Dropout*, que basicamente é uma técnica que ignora pequenos retângulos da imagem e pode ser útil na remoção de ruídos como cabelos nas imagens das lesões. A taxa de aprendizagem foi definida inicialmente em 0.000003, com mínima de 0.000001 e máxima de 0.000020. Podemos ver um exemplo de variação da taxa de aprendizagem durante o treinamento na Figura 32.

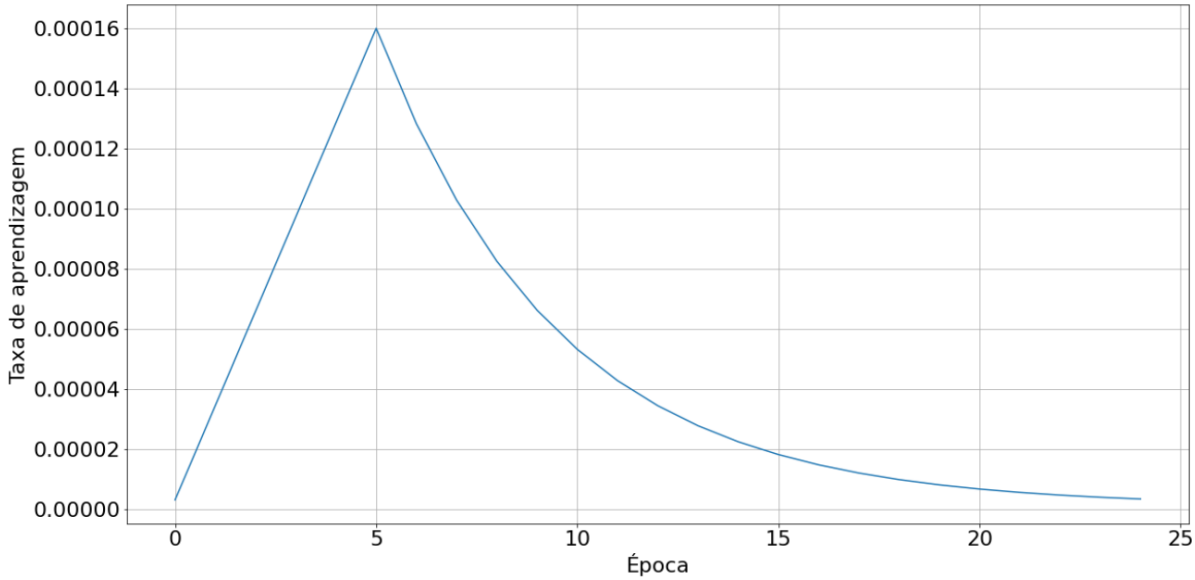


Figura 32 – Variação da taxa de aprendizagem no treinamento do modelo de resolução 384x384 com *dataset* de 2019-2020.

3.2.3 Métricas de desempenho e função de perda

No aprendizado de máquina, a medição de desempenho é uma tarefa essencial. Portanto, quando se trata de um problema de classificação, um dos métodos mais utilizados é a pontuação AUC da curva ROC, conforme visto na seção 2.9.2 (Equação 2.7). A

função de perda a ser utilizada é a de entropia cruzada binária, conforme visto na seção 2.4.3 (Equação 2.4).

3.2.4 Treinamento

Quatro resoluções de entrada foram utilizadas separadamente ao longo do treinamento. Além disso, também foram efetuados cortes nas imagens para eliminar as bordas das mesmas, como podemos ver na Tabela 6:

Tabela 6 – Tamanho do corte realizado em cada imagem de entrada de acordo com sua resolução.

Resolução original	Resolução após corte
256×256	250×250
384×384	370×370
512×512	500×500
768×768	750×750

O modelo então foi treinado oito vezes, pois foi feito o treinamento com cada resolução separadamente para cada *dataset* (ISIC 2020 e ISIC 2019-2020). No que tange o *transfer learning*, o treinamento foi efetuado com pesos iniciais já definidos. O *NoisyStudent* (XIE *et al.*, 2020), que é o conjunto de pesos utilizado no treinamento de várias EfficientNets e também atinge alta acurácia no ImageNet.

Após o treinamento de todos os modelos foi feito o *ensemble* e gerado apenas um arquivo .csv que foi comparado ao *Ground Truth* de teste para obter a pontuação final da rede.

3.3 Recursos computacionais

3.3.1 Hardware

Como tarefas de aprendizado profundo demandam, em sua grande maioria, alto poder computacional, foi escolhido o ambiente Kaggle, que é um *site* especializado no treinamento de redes neurais, para realizar todo o processo de treinamento da EfficientNet-B7. O *site* oferece gratuitamente dois tipos de serviços:

- **GPU** - O Kaggle oferece 30 horas semanais de acesso a sua GPU, que é uma NVIDIA TESLA P100.
- **TPU** - As TPUs são aceleradoras de hardware especializadas em tarefas de *deep learning*. Um núcleo de TPU possui uma unidade multiplicadora de matrizes (MXU - *Matrix Multiplying Unit*) e uma unidade de processamento de vetores (VPU -

Vector Processing Unit), tornando a TPU uma ferramenta ideal para trabalhar com tensores. A TPU fornecida pelo Kaggle possui 8 núcleos e 128 GB de memória RAM, atingindo 420 *teraflops* de desempenho. A mesma é compatível com o TensorFlow 2.1 por meio da API de alto nível Keras. Também fica disponível gratuitamente por 20 horas semanais (KAGGLE, 2021).

3.3.2 Software

Foi utilizada a linguagem Python para programar a EfficientNet, todas as ferramentas e bibliotecas também foram implementadas em Python. Foram utilizadas as bibliotecas TensorFlow e Keras, que são utilizadas para tarefas de aprendizado de máquina.

- **TensorFlow** - Criado pela equipe de IA da Google, o TensorFlow é uma biblioteca de código aberto para computação numérica e aprendizado de máquina em grande escala. O TensorFlow agrupa uma série de modelos e algoritmos de aprendizado de máquina e aprendizado profundo e utiliza Python para fornecer uma API de *front-end* conveniente para construir aplicativos com a estrutura. O TensorFlow pode treinar e executar redes neurais profundas para classificação de dígitos manuscritos, reconhecimento de imagem, incorporação de palavras, redes neurais recorrentes, modelos de sequência a sequência para tradução automática, processamento de linguagem natural e simulações baseadas em EDPs (equações diferenciais parciais) (MATTMANN, 2021).

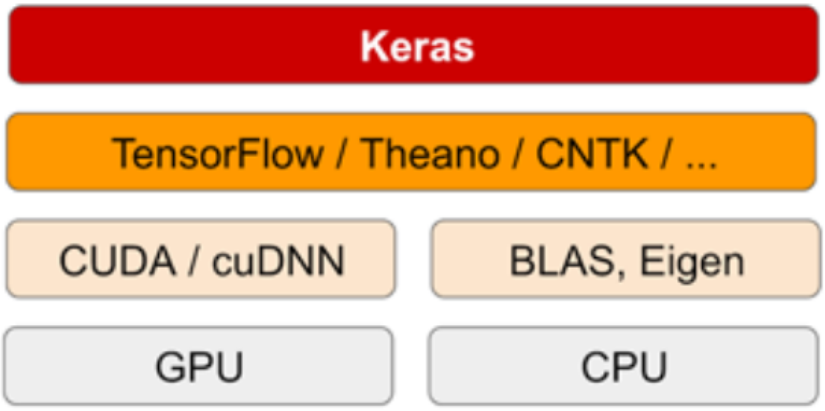


Figura 33 – Módulos de *software* e *hardware* de aprendizado profundo. Fonte: (CHOLLET, 2018)

- **Keras** - O Keras é uma biblioteca (API) que fornece blocos de construção de alto nível para o desenvolvimento de modelos de aprendizado profundo. Ele não lida com operações de baixo nível, como manipulação e diferenciação de tensores. Em vez disso, ele conta com uma biblioteca de tensores especializada e bem otimizada

para fazer isso, servindo como o mecanismo de *back-end* do Keras. Em vez de escolher uma única biblioteca de tensores e vinculá-la ao Keras, o problema pode ser tratado de maneira modular (Figura 33), assim, vários mecanismos de *back-end* diferentes podem ser conectados perfeitamente ao Keras. Algumas implementações de *back-end* existentes são TensorFlow, Theano e Microsoft Cognitive Toolkit (CNTK) (CHOLLET, 2018).

4 Resultados e Discussões

A EfficientNet-B7 foi treinada com o padrão de 64.100.241 parâmetros (estando de acordo com a Figura 30), sendo destes, 63.789.521 parâmetros treináveis e 310.720 não-treináveis. A pontuação AUC de validação de todos os modelos podem ser observados na Figura 34, assim como a perda dos mesmos. Podemos observar que nos modelos que contém apenas o *dataset* de 2020, a pontuação AUC e a perda é levemente menor que nos modelos que possuem o *dataset* de 2019-2020.

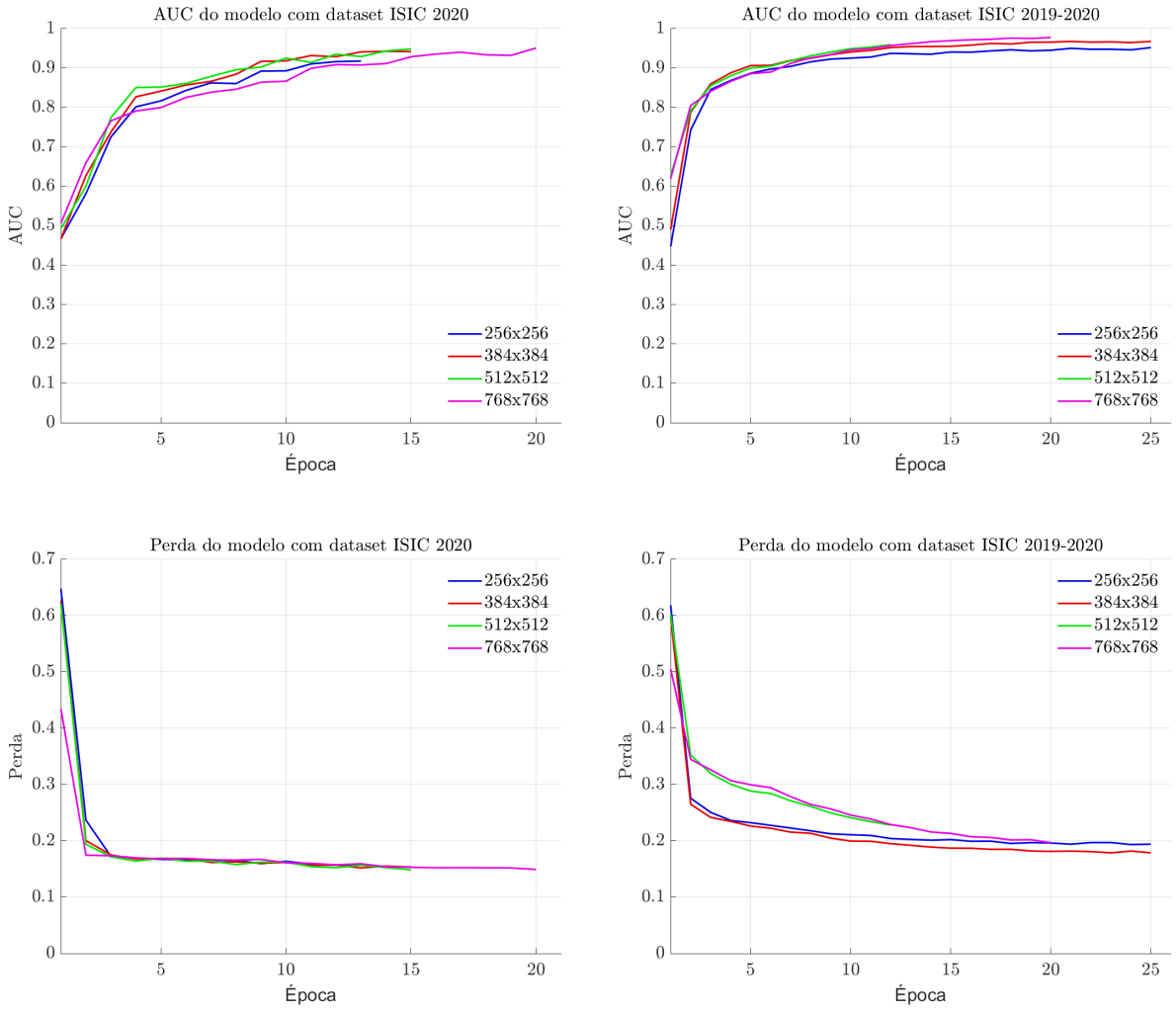


Figura 34 – AUC e perda dos modelos treinados.

Por mais que a perda do modelo com *dataset* 2019-2020 seja um pouco maior, ela obtém uma melhor pontuação AUC devido ao fato de que a AUC é uma medida relativa de ordenação interna, em vez de uma medida absoluta de um conjunto de previsões. A perda de entropia cruzada binária, entretanto, captura até que ponto as probabilidades previstas divergem dos rótulos de classe. Ou seja, por mais que a perda esteja maior pois

o erro absoluto aumentou, a AUC aumenta porque existe uma ordenação de classificação maior, por causa da variabilidade proporcionada pela junção dos 2 *datasets*. De qualquer maneira, o *ensemble* tem a função de mitigar ainda mais a presença de falsos positivos e negativos.

Assim como no trabalho de Karki, Kulkarni e Stranieri (KARKI; KULKARNI; STRANIERI, 2021), também foram geradas as pontuações privadas e públicas para cada resolução, após a realização do *ensemble* em cada uma delas. Abaixo na Tabela 7 podemos observar a pontuação da EfficientNet-B7 com relação às redes treinadas por Karki, Kulkarni e Stranieri (KARKI; KULKARNI; STRANIERI, 2021):

Tabela 7 – Desempenho da EfficientNet-B7 comparada aos modelos treinados por Karki, Kulkarni e Stranieri (KARKI; KULKARNI; STRANIERI, 2021).

Rede	**Resolução**	**Pontuação privada**	**Pontuação Pública**
B7	768×768	0.9288	0.9527
B7	512×512	0.9369	**0.9636**
B7	384×384	**0.9456**	0.9530
B7	256×256	0.9408	0.9434
B6	512×512	0.9369	0.9337
B6	384×384	0.9373	0.9397
B6	256×256	0.9332	0.9364
B5	512×512	0.9364	0.9353
B5	384×384	0.9363	0.9345
B5	256×256	0.9259	0.9232
B4	512×512	0.9300	0.9306
B4	384×384	0.9347	0.9292
B4	256×256	0.9304	0.9287

Além do *ensemble* realizado separadamente para cada resolução, também foi realizado *ensemble* que engloba todos os modelos e gera uma pontuação final para a rede. No trabalho de Karki, Kulkarni e Stranieri (KARKI; KULKARNI; STRANIERI, 2021) é feito um *ensemble* apenas da rede B6 e outro *ensemble* da rede B6 com um modelo da B5 que foi treinado utilizando apenas a resolução 368x368. Essas pontuações podem ser usadas para comparar resultados com a rede apresentada no artigo de Zhang e Wang (ZHANG; WANG, 2021), conforme observamos na Tabela 8:

Tabela 8 – Desempenho da EfficientNet-B7 comparado às redes dos trabalhos relacionados.

Modelos	Pontuação AUC ROC
DenseNet201	0.9250
EfficientNet-B6	0.9409
EfficientNet-B5+B6	0.9411
EfficientNet-B7	**0.9467**

Outro aspecto importante a ser observado é o tempo de execução de cada modelo, que pode ser visto na Tabela 9:

Tabela 9 – Tempo de execução do treinamento de cada modelo.

Conjunto	Resolução de entrada	Tempo de execução (hh:mm:ss)
2020	256 × 256	00:18:42
2020	384 × 384	00:36:24
2020	512 × 512	01:16:04
2020	768 × 768	06:57:07
2019-2020	256 × 256	01:02:06
2019-2020	384 × 384	01:50:16
2019-2020	512 × 512	01:48:43
2019-2020	768 × 768	08:30:04

Ambos os modelos com resolução 768x768 levaram um tempo substancialmente maior para serem treinadas. O tamanho do lote precisou ser reduzido em quatro vezes, assim como feito o aumento do número de épocas para esses modelos, visto que estava ocorrendo extrapolação da memória disponível, já que as imagens tem resolução maior e a EfficientNet-B7 é uma rede muito profunda com mais camadas que as suas versões predecessoras (B4-B6).

Os gráficos de validação para as pontuações AUC e para as perdas dos modelos, conforme mostrado na Figura 34, evidenciam o bom treinamento da rede e da escolha dos hiperparâmetros. Tanto a curva de perda quanto a curva de AUC não apresentam comportamento que indiquem algum tipo de divergência da rede que desse indícios da presença de *overfitting*.

Os resultados da Tabela 7 refletem as medidas de precisão do ImageNet mostradas na Figura 30. Há um pequeno aumento na pontuação da EfficientNet-B7 com relação à rede EfficientNet-B6 (KARKI; KULKARNI; STRANIERI, 2021), entretanto, o tempo de execução da EfficientNet-B7 é cerca de 1,5 vezes maior, se ambas as redes forem treinadas utilizando TPUs de 8 núcleos.

Levando em consideração as pontuações mostradas na Tabela 7 e o tempo de execução de cada modelo, representados na Tabela 9, podemos afirmar que o modelo com resolução de entrada de 384x384 é o ideal para o treinamento das EfficientNets. Tendo pontuação privada de 0.9456 neste trabalho e 0.9373 no trabalho de Karki, Kulkarni e Stranieri (KARKI; KULKARNI; STRANIERI, 2021), obtendo a maior pontuação em ambos.

É também importante ressaltar os resultados apresentados na Tabela 8, evidenciando que a EfficientNet-B7 obteve uma pontuação maior que redes bastante consagradas no quesito de classificação de imagens médicas, como a DenseNet201, demonstrando a eficácia da EfficientNet-B7 para esse tipo de classificação. Vale ressaltar que o trabalho de Zhang e Wang (ZHANG; WANG, 2021) também traz as pontuações da ResNet50 (0.922) e VGG16 (0.891), que são CNNs conceituadas no ramo de classificação de imagens.

5 Conclusão

A classificação de lesões de câncer de pele em classes malignas e benignas é um trabalho desafiador e demorado para os olhos humanos. Considerando um cenário de escassez de especialistas ou de atendimento, o diagnóstico automatizado de câncer de pele é essencial. Este trabalho propôs o treinamento da rede neural convolucional EfficientNet-B7, que atinge o estado da arte em outras tarefas de classificação de imagem, para a tarefa de classificação de melanoma. Também feitas comparações com a literatura, com outras CNNs executando a mesma tarefa e utilizando o mesmo *dataset*.

A métrica de desempenho escolhida foi a pontuação AUC da curva ROC. As pontuações finais de cada modelo foram obtidas utilizando a técnica de *ensemble*. Várias técnicas de aumento, como adição de cabelo e rotação, foram usadas como pré-processamento para melhorar o desempenho da classificação e aumentar a generalização da rede.

O melhor resultado obtido foi o de 0.9467, que foi gerado a partir do *ensemble* de todos os 8 modelos treinados. Outro resultado relevante é o de 0.9456, que foi gerado a partir do *ensemble* dos dois modelos com resolução 384x384, um com apenas o *dataset* de 2020 e outro com o *dataset* de 2019-2020. Esses resultados são levemente melhores que as pontuações obtidas nas redes EfficientNet-B4, EfficientNet-B5, EfficientNet-B6 e DenseNet201. Por meio destes resultados é possível concluir que a EfficientNet é uma arquitetura bastante viável para classificação de imagens dermatoscópicas de melanoma.

Muito tempo computacional e recursos são necessários para determinar um modelo de aprendizado de máquina com melhor desempenho. Portanto, como sugestão para trabalhos futuros, seria interessante avaliar o desempenho de outras técnicas de otimização que estão surgindo, como a técnica de otimização de convergência angular. Também é válido averiguar a convergência da EfficientNet utilizando outros pesos além do *NoisyStudent*, visto que esse é um campo que evolui rapidamente conforme novas ideias e modelos vão surgindo.

Referências

ALI, M. S. *et al.* An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models. **Machine Learning with Applications**, v. 5, p. 100036, 2021. ISSN 2666-8270. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2666827021000177>. Citado na página 26.

BAKOS, R. M. *et al.* Noninvasive Imaging Tools in the Diagnosis and Treatment of Skin Cancers. **American journal of clinical dermatology**, Springer International Publishing, v. 19, n. 30374899, p. 3–14, nov. 2018. ISSN 1175-0561. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6244601/>. Citado na página 25.

BREIMAN, L. Bagging Predictors. **Machine Learning**, v. 24, n. 2, p. 123–140, 1996. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1018054314350>. Citado na página 46.

BRINKER, T. J. *et al.* Deep neural networks are superior to dermatologists in melanoma image classification. **European Journal of Cancer**, v. 119, p. 11–17, 2019. ISSN 0959-8049. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0959804919303491>. Citado na página 26.

CHARNIAK, E. **Introduction to Deep Learning**. 1. ed. [*S.l.: s.n.*]: The MIT Press, 2019. ISBN 0262039516; 9780262039512. Citado na página 38.

CHOLLET, F. **Deep Learning with Python**. [*S.l.: s.n.*]: Manning, 2018. ISBN 9781617294433; 1617294438. Citado 10 vezes nas páginas 14, 34, 39, 40, 42, 44, 47, 49, 67 e 68.

CODELLA, N. C. F. *et al.* Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), Hosted by the International Skin Imaging Collaboration (ISIC). **CoRR**, abs/1710.05006, 2017. Disponível em: <http://arxiv.org/abs/1710.05006>. Citado na página 62.

COMBALIA, M. *et al.* **BCN20000: Dermoscopic Lesions in the Wild**. 2019. Disponível em: <https://arxiv.org/abs/1908.02288>. Citado 3 vezes nas páginas 14, 62 e 63.

CULLELL-DALMAU, M. *et al.* Convolutional Neural Network for Skin Lesion Classification: Understanding the Fundamentals Through Hands-On Learning. **Frontiers in Medicine**, v. 8, p. 213, 2021. ISSN 2296-858X. Disponível em: <https://www.frontiersin.org/article/10.3389/fmed.2021.644327>. Citado na página 61.

DICK, V. *et al.* Accuracy of Computer-Aided Diagnosis of Melanoma: A Meta-analysis. **JAMA Dermatology**, v. 155, n. 11, p. 1291–1299, nov. 2019. ISSN 2168-6068. Disponível em: <https://doi.org/10.1001/jamadermatol.2019.1375>. Citado na página 26.

DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. **J. Mach. Learn. Res.**, JMLR.org, v. 12, n. null, p. 2121–2159, jul. 2011. ISSN 1532-4435. Citado na página 43.

FUJISAWA, Y.; INOUE, S.; NAKAMURA, Y. The Possibility of Deep Learning-Based, Computer-Aided Skin Tumor Classifiers. **Frontiers in Medicine**, v. 6, p. 191, 2019. ISSN 2296-858X. Disponível em: <https://www.frontiersin.org/article/10.3389/fmed.2019.00191>. Citado na página 26.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. 4th ed.. ed. [*S.l.: s.n.*]: Pearson, 2018. ISBN 9780133356724,0133356728. Citado na página 33.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [*S.l.: s.n.*]: MIT, 2017. Citado 18 vezes nas páginas 13, 34, 35, 36, 37, 38, 39, 40, 42, 43, 44, 45, 46, 48, 49, 50, 51 e 52.

GéRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 2. ed. [*S.l.: s.n.*]: O'Reilly Media, 2019. ISBN 9781492032649. Citado 13 vezes nas páginas 13, 14, 41, 42, 44, 45, 46, 47, 48, 49, 50, 58 e 61.

HAENSSLE, H. A. *et al.* Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. **Annals of Oncology**, Elsevier, v. 29, n. 8, p. 1836–1842, ago. 2018. ISSN 0923-7534. Disponível em: <https://doi.org/10.1093/annonc/mdy166>. Citado na página 26.

HAENSSLE, H. A. *et al.* Man against machine reloaded: performance of a market-approved convolutional neural network in classifying a broad spectrum of skin lesions in comparison with 96 dermatologists working under less artificial conditions. **Annals of Oncology**, Elsevier, v. 31, n. 1, p. 137–143, 2020. ISSN 0923-7534. Disponível em: <https://doi.org/10.1016/j.annonc.2019.10.013>. Citado na página 26.

HE, K. *et al.* Deep Residual Learning for Image Recognition. **CoRR**, abs/1512.03385, 2015. Disponível em: <http://arxiv.org/abs/1512.03385>. Citado 3 vezes nas páginas 27, 52 e 56.

HENDI, A.; MARTINEZ, J. C. **Atlas of Skin Cancers: Practical Guide to Diagnosis and Treatment**. [*S.l.: s.n.*]: Springer, 2011. v. 1. ISBN 9783642133992. Citado na página 31.

HUANG, G.; LIU, Z.; WEINBERGER, K. Q. Densely connected convolutional networks. **CoRR**, abs/1608.06993, 2016. Disponível em: <http://arxiv.org/abs/1608.06993>. Citado 2 vezes nas páginas 14 e 56.

HUANG, Y. *et al.* Gpipe: Efficient training of giant neural networks using pipeline parallelism. **CoRR**, abs/1811.06965, 2018. Disponível em: <http://arxiv.org/abs/1811.06965>. Citado na página 52.

IMAGENET. **ImageNet Large Scale Visual Recognition Challenge (ILSVRC)**. 2017. Acessado em: 28 de outubro de 2021. Disponível em: <https://www.image-net.org/challenges/LSVRC>. Citado na página 27.

INCA. **Câncer de pele melanoma**. 2021. Acessado em: 13 de outubro de 2021. Disponível em: <https://www.inca.gov.br/tipos-de-cancer/cancer-de-pele-melanoma>. Citado na página 25.

ISIC. **Skin Lesion Analysis Towards Melanoma Detection**. 2019. Disponível em: <https://challenge2019.isic-archive.com/>. Citado na página 62.

ISIC. **The 2020 Live Challenge is open!** 2020. Acessado em: 28 de outubro de 2021. Disponível em: <https://challenge.isic-archive.com>. Citado na página 27.

JOUPPI, N. P. *et al.* In-datacenter performance analysis of a tensor processing unit. *In*: **2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)**. [*S.l.: s.n.*], 2017. p. 1–12. Citado na página 27.

KAGGLE. **Tensor Processing Units (TPUs) Documentation**. 2021. Acessado em: 28 de outubro de 2021. Disponível em: <https://www.kaggle.com/docs/tpu>. Citado 3 vezes nas páginas 59, 60 e 67.

KARKI, S.; KULKARNI, P.; STRANIERI, A. Melanoma Classification Using EfficientNets and Ensemble of Models with Different Input Resolution. *In*: **2021 Australasian Computer Science Week Multiconference**. New York, NY, USA: Association for Computing Machinery, 2021. (ACSW '21). ISBN 9781450389563. Disponível em: <https://doi.org/10.1145/3437378.3437396>. Citado 6 vezes nas páginas 15, 59, 60, 61, 70 e 71.

KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. 2017. Citado na página 44.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *In*: PEREIRA, F. *et al.* (ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2012. v. 25. Disponível em: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>. Citado na página 27.

LECUN, Y. *et al.* Backpropagation Applied to Handwritten Zip Code Recognition. **Neural Computation**, v. 1, n. 4, p. 541–551, 1989. Citado na página 47.

LECUN, Y. *et al.* Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 47.

LEE, D. J.; FARIES, M. B. **Practical Manual for Dermatologic and Surgical Melanoma Management**. [*S.l.: s.n.*]: Springer, 2020. ISBN 9783030273996. Citado na página 29.

LI, Q. *et al.* Medical image classification with convolutional neural network. *In*: **2014 13th International Conference on Control Automation Robotics Vision (ICARCV)**. [*S.l.: s.n.*], 2014. p. 844–848. Citado na página 26.

MASSI, G.; LEBOIT, P. E. **Histological Diagnosis of Nevi and Melanoma**. [*S.l.: s.n.*]: Springer, 2014. v. 1. ISBN 9783642373114. Citado 3 vezes nas páginas 29, 30 e 31.

MATHWORKS. **What is a Convolutional Neural Network?** 2021. Acessado em: 28 de outubro de 2021. Disponível em: <https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html>. Citado 2 vezes nas páginas 13 e 51.

MATTMANN, C. A. **Machine Learning with TensorFlow**. 2. ed. [*S.l.: s.n.*]: Manning Publications Co., 2021. ISBN 9781617297717. Citado 7 vezes nas páginas 13, 14, 42, 43, 51, 57 e 67.

MCANDREW, A. **A Computational Introduction to Digital Image Processing**. 2nd edition. ed. [*S.l.: s.n.*]: CRC Press, 2015. ISBN 9781482247350; 1482247356. Citado 2 vezes nas páginas 33 e 34.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. v. 5, n. 4, p. 115–133, 1943. ISSN 1522-9602. Disponível em: <https://doi.org/10.1007/BF02478259>. Citado na página 39.

MINSKY, M.; PAPERT, S. **Perceptrons.** Oxford, England: M.I.T. Press, 1969. Citado 2 vezes nas páginas 13 e 40.

MURPHY, K. P. **Machine Learning: A Probabilistic Perspective**. 1. ed. [*S.l.: s.n.*]: The MIT Press, 2012. (Adaptive Computation and Machine Learning). ISBN 9780262018029. Citado 4 vezes nas páginas 35, 36, 37 e 38.

ONCOGUIA. **Taxa de Sobrevida para Câncer de Pele Melanoma**. 2020. Acessado em: 13 de outubro de 2021. Disponível em: <http://www.oncoguia.org.br/conteudo/taxa-de-sobrevida-para-cancer-de-pele-melanoma/7062/187>. Citado na página 25.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65 6, p. 386–408, 1958. Citado na página 39.

ROSENDAHL, C.; MAROZAVA, A. **Dermatoscopy and Skin Cancer: A handbook for hunters of skin cancer and melanoma**. 4. ed. [*S.l.: s.n.*]: Scion Publishing Ltd, 2020. v. 1. ISBN 1911510339. Citado na página 31.

ROTEMBERG, V. *et al.* A patient-centric dataset of images and metadata for identifying melanomas using clinical context. **Scientific Data**, v. 8, n. 1, p. 34, 2021. ISSN 2052-4463. Disponível em: <https://doi.org/10.1038/s41597-021-00815-z>. Citado 3 vezes nas páginas 14, 61 e 62.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach, Global Edition**. 4. ed. [*S.l.: s.n.*]: Pearson, 2021. ISBN 9780134610993. Citado na página 34.

SANDLER, M. *et al.* Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. **CoRR**, abs/1801.04381, 2018. Disponível em: <http://arxiv.org/abs/1801.04381>. Citado na página 53.

SIMONYAN, K.; ZISSERMAN, A. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. 2015. Citado na página 27.

SOLOMON, C.; BRECKON, T. **Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab**. 1. ed. [*S.l.: s.n.*]: Wiley-Blackwell, 2011. ISBN 0470844728; 9780470844724; 0470844736; 9780470844731; 9780470689783; 0470689781; 9780470689776; 0470689773. Citado 2 vezes nas páginas 33 e 34.

SOMMER, C. Skin biopsy as a diagnostic tool. **Current opinion in neurology**, England, v. 21, p. 563–8, out. 2008. ISSN 1350-7540. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/18769250>. Citado na página 25.

SRIVASTAVA, N. *et al.* Dropout: A Simple Way to Prevent Neural Networks from Overfitting. **Journal of Machine Learning Research**, v. 15, n. 56, p. 1929–1958, 2014. Disponível em: <http://jmlr.org/papers/v15/srivastava14a.html>. Citado na página 46.

SZEGEDY, C. *et al.* **Going Deeper with Convolutions**. 2014. Citado na página 27.

TAN, M. *et al.* Mnasnet: Platform-aware neural architecture search for mobile. **CoRR**, abs/1807.11626, 2018. Disponível em: <http://arxiv.org/abs/1807.11626>. Citado na página 53.

TAN, M.; LE, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. 2020. Disponível em: <https://arxiv.org/abs/1905.11946>. Citado 6 vezes nas páginas 13, 14, 52, 53, 54 e 63.

TKACZYK, E. Innovations and Developments in Dermatologic Non-invasive Optical Imaging and Potential Clinical Applications. **Acta dermato-venereologica**, Suppl 218, n. 28676880, p. 5–13, 2017. ISSN 0001-5555. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5943168/>. Citado na página 25.

TRASK, A. W. **Grokking Deep Learning**. [*S.l.: s.n.*]: Manning, 2019. ISBN 9781617293702. Citado 2 vezes nas páginas 13 e 39.

TSCHANDL, P.; ROSENDAHL, C.; KITTLER, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. **Scientific Data**, v. 5, n. 1, p. 180161, 2018. ISSN 2052-4463. Disponível em: <https://doi.org/10.1038/sdata.2018.161>. Citado na página 62.

WEINBERG, R. A. **The biology of cancer**. 2. ed. [*S.l.: s.n.*]: Garland Science, 2014. ISBN 9780815342205. Citado na página 29.

WIDROW, B. An Adaptive 'Adaline' Neuron Using Chemical 'Memistors'. **Stanford Electronics Laboratories Technical Report**, 1960. Citado na página 39.

WOLFF, K. *et al.* **Fitzpatrick's Dermatology in General Medicine**. 4. ed. [*S.l.: s.n.*]: McGraw-Hill, 2011. ISBN 9780071669054. Citado 4 vezes nas páginas 13, 30, 31 e 32.

WOLNER, Z. J. *et al.* Enhancing Skin Cancer Diagnosis with Dermoscopy. **Dermatologic clinics**, v. 35, p. 417–437, out. 2017. ISSN 1558-0520. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/28886798/>. Citado na página 25.

XIE, Q. *et al.* Self-training with Noisy Student improves ImageNet classification. 2020. Disponível em: <https://arxiv.org/abs/1911.04252>. Citado na página 66.

ZAGORUYKO, S.; KOMODAKIS, N. Wide residual networks. **CoRR**, abs/1605.07146, 2016. Disponível em: <http://arxiv.org/abs/1605.07146>. Citado na página 52.

ZHANG, Y.; WANG, C. SIIM-ISIC Melanoma Classification With DenseNet. *In*: **2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)**. [*S.l.: s.n.*], 2021. p. 14–17. Disponível em: <https://ieeexplore.ieee.org/document/9389983>. Citado 7 vezes nas páginas 15, 58, 59, 60, 61, 70 e 71.

ZHOU, Y.; CHELLAPPA, R. Computation of optical flow using a neural network. **IEEE 1988 International Conference on Neural Networks**, p. 71–78 vol.2, 1988. Citado na página 50.