



DETECÇÃO DE DESCONTINUIDADES EM RADIOGRAFIAS DE CHAPAS  
SOLDADAS DA INDÚSTRIA DE CALDEIRARIA TÉCNICA NUCLEAR PESADA  
COM AUXÍLIO DE REDES NEURAIAS ARTIFICIAIS

Romulo de Souza Gouvea

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Nuclear, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Nuclear.

Orientador: Alan Miranda Monteiro de Lima

Rio de Janeiro

Fevereiro de 2024

DETECÇÃO DE DESCONTINUIDADES EM RADIOGRAFIAS DE CHAPAS  
SOLDADAS DA INDÚSTRIA DE CALDEIRARIA TÉCNICA NUCLEAR PESADA  
COM AUXÍLIO DE REDES NEURAIAS ARTIFICIAIS

Romulo de Souza Gouvea

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO  
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA  
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM  
ENGENHARIA NUCLEAR.

Orientador: Alan Miranda Monteiro de Lima.

Aprovada por: Prof. Alan Miranda Monteiro de Lima.

Prof. Marcelo Carvalho dos Santos.

Prof. Victor Henrique Cabral Pinheiro.

RIO DE JANEIRO, RJ -BRASIL

FEVEREIRO DE 2024

Gouvea, Romulo de Souza

Detecção de descontinuidades em radiografias de chapas soldadas da industria de caldeiraria técnica nuclear pesada com auxílio de redes neurais artificiais / Romulo de Souza Gouvea - Rio de Janeiro: UFRJ/COPPE, 2024.

X, 61.: il.; 29,7 cm.

Orientador: Alan Miranda Monteiro de Lima

Dissertação (mestrado) - UFRJ / COPPE/ Programa de Engenharia Nuclear, 2024.

Referências Bibliográficas: p. 59-61.

1. Redes Neurais Artificiais 2. Descontinuidades em solda. 3. Detecção de defeitos em solda. I. Lima, Alan Miranda Monteiro de. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Nuclear. III. Título.

## **Dedicatória**

Dedico este trabalho à minha mãe e ao meu pai, que sempre incentivaram os estudos dos filhos, priorizando a educação e as experiências ante todas as outras coisas.

## **Agradecimentos**

Sou grato a Deus sobre todas as coisas, pois Ele concede saúde a mim e a minha família, proporcionando assim força e desenvoltura para suportar as árduas jornadas de trabalho, acadêmica e familiar, concomitantemente. Agradeço ainda a Ele pelo presente que recebi durante o curso de mestrado, a concepção e nascimento do meu filho Heleno, que juntamente do irmão Heitor, meu primogênito, são os grandes orgulhos da minha vida.

Resumo da Dissertação apresentada à COPPE / UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc).

DETECÇÃO DE DESCONTINUIDADES EM RADIOGRAFIAS DE CHAPAS  
SOLDADAS DA INDÚSTRIA DE CALDEIRARIA TÉCNICA NUCLEAR PESADA  
COM AUXÍLIO DE REDES NEURAIIS ARTIFICIAIS

Romulo de Souza Gouvea  
Fevereiro/2024

Orientador: Alan Miranda Monteiro de Lima

Programa: Engenharia Nuclear

Nesta dissertação é apresentado um estudo de verificação da capacidade de uma rede neural artificial identificar descontinuidades em imagens de radiografia de chapas de aços variados, soldadas para a indústria de caldeiraria técnica pesada. Com tais imagens obtidas a partir da fotografia do negatoscópio, visto a impossibilidade de obtenção de amostras de radiografia digitais. Tal experimento comprovou a viabilidade de continuidade desses estudos, e do método adotado, considerando que a rede atingiu uma acurácia superior à 88% (oitenta e oito por cento), mesmo considerando o banco de dados mediano disponível para o experimento. Os resultados obtidos demonstram que essa tarefa é bem resolvida pelas redes neurais artificiais devido à quantidade de radiografias inspecionadas e a qualidade dos resultados, em um espaço de tempo muito menor que o mesmo trabalho realizado pelo especialista humano.

Abstract of Dissertation presented to COPPE / UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc).

DISCONTINUITIES DETECTION IN WELDED PLATES RADIOGRAPHS FOR  
TECHNICAL HEAVY DUTY BOILERMAKING NUCLEAR INDUSTRY WITHIN  
ARTIFICIAL NEURAL NETWORKS AID

Romulo de Souza Gouvea

February/2024

Advisor: Alan Miranda Monteiro de Lima

Department: Nuclear Engineering

In this dissertation, a study is presented to verify the capability of an artificial neural network to identify discontinuities in images of radiographs of various welded steel plates for the heavy technical boiler industry. These images are obtained from photographs of the negatoscope, given the impossibility of obtaining samples of digital radiography. This experiment proved the feasibility of continuing these studies and the adopted method, considering that the network achieved an accuracy of over 88%, even considering the median database available for the experiment. The results obtained demonstrate that this task is well resolved by artificial neural networks due to the quantity of radiographs inspected and the quality of the results, in a much shorter timeframe than the same task performed by a human specialist..

# SUMÁRIO

1.	INTRODUÇÃO .....	1
1.1	O RESUMO HISTÓRICO DA CONSTRUÇÃO COM MATERIAIS METÁLICOS .....	1
1.2	CONTEXTUALIZAÇÃO DO PROBLEMA ENCONTRADO .....	3
1.3	USO DA INTELIGÊNCIA ARTIFICIAL COMO AUXÍLIO PROFISSIONAL.....	4
2.	REVISÃO DA LITERATURA.....	5
2.1.	LIGAÇÃO DOS ELEMENTOS CONSTRUTIVOS POR SOLDAGEM .....	5
2.2.	CONTROLE E INSPEÇÃO DE SOLDAGEM .....	7
2.3.	ENSAIOS NÃO DESTRUTIVOS – END – EM SOLDAS.....	8
2.4.	TIPOS DE DEFEITOS EM SOLDAS .....	9
2.4.1.	Porosidade.....	10
2.4.2.	Inclusões de escórias e/ou tungstênio .....	11
2.4.3.	Falta de fusão .....	12
2.4.4.	Falta de penetração .....	13
2.4.5.	Trincas .....	14
2.5.	ENSAIO RADIOGRÁFICO EM SOLDAS.....	15
3.	INTELIGÊNCIA ARTIFICIAL.....	16
3.1.	O QUE É INTELIGÊNCIA ARTIFICIAL .....	16
3.2.	REDES NEURAIS.....	17
3.3.	CONTEXTUALIZAÇÃO HISTÓRICA DAS REDES NEURAIS ARTIFICIAIS.....	18
3.3.1.	A onda da Cibernética.....	19
3.3.2.	A onda do Conexionismo.....	21
3.3.3.	A onda do Aprendizado Profundo .....	23
3.4.	O PROBLEMA DO DESAPARECIMENTO DE GRADIENTE.....	24
4.	METODOLOGIA DA PESQUISA.....	27
4.1.	O EXPERIMENTO.....	39
4.1.1.	Obtenção das amostras de imagens.....	39
4.1.2.	FLUXOGRAMA DA REDE NEURAL CONVOLUCIONAL UTILIZADA .....	24
4.1.3.	SUBMETENDO AS AMOSTRAS À REDE NEURAL: .....	43
4.2.	ENTENDENDO AS FUNÇÕES UTILIZADAS NA REDE NEURAL.....	27
4.2.1.	DIVISÃO TREINO E TESTE SIMPLES (HOLD-OUT).....	27
4.2.2.	CAMADAS DE CONVOLUÇÃO .....	29
4.2.3.	O QUE SÃO OS NEURÔNIOS DA REDE NEURAL .....	30
4.2.4.	CAMADA DE POOLING ATRAVÉS DA FUNÇÃO MAXPOOL 2D .....	31
4.2.5.	A FUNÇÃO FLATTEN .....	32
4.2.6.	CAMADA Densa E OVERFITTING.....	33
4.2.7.	VALIDAÇÃO CRUZADA.....	34
4.2.8.	TREINANDO A REDE: AS FUNÇÕES FIT E SEQUENTIAL.....	35

4.2.9.	ENTENDENDO A MÉTRICA DE APRENDIZADO ATRAVÉS DAS FUNÇÕES DE PERDA LOSS E ACCURACY E A MELHORIA DOS RESULTADOS ATRAVÉS DO ALGORITMO OTIMIZADOR ADAM .....	36
4.2.10.	AVALIANDO OS RESULTADOS ENCONTRADOS, A FUNÇÃO EVALUATE.....	38
5.	RESULTADOS.....	45
5.1.1.	INTERPRETANDO AS MÉTRICAS PRECISION, RECALL, F1-SCORE, MACRO AVG E WEIGHTED AVG.....	45
5.2.	AVALIANDO OS GRÁFICOS IMPRESSOS .....	47
5.2.1.	EVOLUÇÃO DO TREINAMENTO .....	47
5.2.2.	EVOLUÇÃO DO ERRO .....	48
5.2.3.	EVOLUÇÃO DA ACURÁCIA .....	49
5.2.4.	EVOLUÇÃO DA VALIDAÇÃO CRUZADA.....	50
5.3.	AJUSTANDO ALGUNS DOS HIPERPARÂMETROS DA REDE NEURAL.....	51
5.3.1.	JUSTIFICATIVA PARA AJUSTE NOS HIPERPARÂMETROS .....	51
5.3.2.	RESULTADOS OBTIDOS APÓS AJUSTE.....	53
6.	CONCLUSÃO .....	55
7.	SUGESTÃO DE TRABALHOS FUTUROS.....	57
8.	REFERÊNCIAS.....	59

## LISTA DE FIGURAS

FIGURA 1: EXEMPLO DE POROSIDADE IDENTIFICÁVEL EM UMA RADIOGRAFIA DE SOLDAGEM .....	11
FIGURA 2: EXEMPLO DE INCLUSÃO DE ESCÓRIA IDENTIFICÁVEL EM UMA RADIOGRAFIA DE SOLDAGEM .....	12
FIGURA 3: EXEMPLO DE FALTA DE FUSÃO EM UMA RADIOGRAFIA DE SOLDAGEM .....	13
FIGURA 4: EXEMPLO DE FALTA DE PENETRAÇÃO EM UMA RADIOGRAFIA DE SOLDAGEM..	14
FIGURA 5: DIAGRAMAS EXEMPLOS DE ARQUITETURAS DE REDES NEURAIIS – IMAGEM ORIGINALMENTE DISPONÍVEL EM: <a href="https://www.asimovinstitute.org/neural-network-zoo/">HTTPS://WWW.ASIMOVINSTITUTE.ORG/NEURAL-NETWORK-ZOO/</a> .....	24
FIGURA 6: O FLUXOGRAMA DE FUNCIONAMENTO DA REDE NEURAL OBJETO DESTES ESTUDO .....	26
FIGURA 7: MARCOS NO DESENVOLVIMENTO DAS REDES NEURAIIS. - FIGURA ORIGINALMENTE DISPONÍVEL EM: <a href="http://deeplearningbook.com.br/uma-breve-historia-das-redes-neurais-artificiais/">HTTP://DEEPLARNINGBOOK.COM.BR/UMA-BREVE-HISTORIA-DAS-REDES-NEURAIIS-ARTIFICIAIS/</a> .....	27
FIGURA 8: PROPORCIONALIDADE DA DIVISÃO DOS DADOS PARA TREINO E TESTE PARA UMA RNA HIPOTÉTICA. ....	28
FIGURA 9: PROPORCIONALIDADE DA DIVISÃO DOS DADOS PARA TREINO, TESTE E VALIDAÇÃO PARA UMA RNA HIPOTÉTICA. CRÉDITOS NA IMAGEM. ....	28
FIGURA 10: SINTAXE DA FUNÇÃO <code>TRAIN_TEST_SPLIT()</code> DO SKLEARN .....	29
FIGURA 11: DIAGRAMA HIPOTÉTICO - EXEMPLO DE APLICAÇÃO.....	32
FIGURA 12: FOTOGRAFIA DO NEGATOSCÓPIO REVELANDO A IMAGEM GRAVADA NO FILME RADIOGRÁFICO .....	40
FIGURA 13: IMAGEM PRÉ-PROCESSADA, NA QUAL FORAM DESCARTADAS AS MARGENS FORA DA ÁREA DE INTERESSE.....	41
FIGURA 14: AMOSTRAS DE IMAGEM OBTIDAS A PARTIR DE UMA ÚNICA RADIOGRAFIA. PODE-SE NOTAR QUE A SEQUÊNCIA DE AMOSTRAS EXIBEM CONTINUIDADE DE UMA PARA A OUTRA.....	42
FIGURA 15: RESULTADOS DO TREINAMENTO DA REDE NEURAL, IMPRESSOS EM TELA COM USO DA FUNÇÃO <code>METRICS.CLASSIFICATION_REPORT</code> .....	46
FIGURA 16: GRÁFICO DE EVOLUÇÃO DO TREINAMENTO PARA ERRO E ACURÁCIA.....	47
FIGURA 17: GRÁFICO COMPARATIVO DO ERRO NO TREINAMENTO E NA VALIDAÇÃO CRUZADA .....	49
FIGURA 18: GRÁFICO COMPARATIVO DA ACURÁCIA NO TREINAMENTO E NA VALIDAÇÃO CRUZADA .....	50
FIGURA 19: GRÁFICO DE EVOLUÇÃO DA VALIDAÇÃO CRUZADA PARA ERRO E ACURÁCIA. ....	51
FIGURA 20: RESULTADOS DO TESTE DE TREINAMENTO DA REDE NEURAL, IMPRESSOS EM TELA COM USO DA FUNÇÃO <code>METRICS.CLASSIFICATION_REPORT</code> , APÓS AJUSTE DOS HIPERPARÂMETROS;.....	54
FIGURA 21: RESULTADOS DO TREINAMENTO DA REDE NEURAL, IMPRESSOS EM TELA COM USO DA FUNÇÃO <code>METRICS.CLASSIFICATION_REPORT</code> , APÓS AJUSTE DOS HIPERPARÂMETROS;.....	54
FIGURA 22: GRÁFICO COMPARATIVO DO ERRO NO TREINAMENTO E NA VALIDAÇÃO CRUZADA, APÓS AJUSTE DOS HIPERPARÂMETROS; .....	54
FIGURA 23: GRÁFICO DE EVOLUÇÃO DO TREINAMENTO PARA ERRO E ACURÁCIA, APÓS AJUSTE DOS HIPERPARÂMETROS.....	54
FIGURA 24: GRÁFICO DE EVOLUÇÃO DA VALIDAÇÃO CRUZADA PARA ERRO E ACURÁCIA, APÓS AJUSTE DOS HIPERPARÂMETROS;.....	55
FIGURA 25: GRÁFICO COMPARATIVO DA ACURÁCIA NO TREINAMENTO E NA VALIDAÇÃO CRUZADA, APÓS AJUSTE DOS HIPERPARÂMETROS;.....	55

# 1.INTRODUÇÃO

A construção com materiais metálicos é a base da indústria de engenharia mecânica, especificamente a caldeiraria técnica pesada. E para garantir a integridade estrutural desses equipamentos, a inspeção e o controle de qualidade tem papel fundamental nesse processo, para a segurança e a eficiência operacional. Com a radiografia sendo a das técnicas mais amplamente empregadas para identificar possíveis descontinuidades no interior de juntas soldadas. No entanto, a interpretação profissional das radiografias é um processo trabalhoso, possibilitando almejarmos avanços na automação desse procedimento. A introdução de redes neurais convolucionais, *CNN - convolutional neural network*, pode oferecer solução para essa questão, permitindo a automação da interpretação radiográfica com maior precisão e eficiência. Tais redes são capazes de aprender padrões complexos nas imagens radiográficas e identificar com precisão descontinuidades, contribuindo assim para a melhoria da qualidade e segurança em processos industriais que envolvem materiais metálicos.

A automação da interpretação radiográfica com o uso de CNN poderá representar um avanço significativo na indústria, o que possibilitaria uma análise mais rápida, precisa e consistente das radiografias de soldagem. Ao aprender com um grande volume de dados e padrões complexos nas imagens, tais redes podem identificar descontinuidades e defeitos com uma eficácia comparável ou até superior à análise humana, o que aceleraria o processo de inspeção, reduzindo os tempos de parada e custos associados.

## 1.1 O RESUMO HISTÓRICO DA CONSTRUÇÃO COM MATERIAIS METÁLICOS

Ao analisarmos a relação da humanidade com a engenharia ao longo do tempo, nota-se a imperativa vontade e determinação em conter ou dominar fenômenos e ocorrências naturais como forma de trazer comodidade, facilidade ou segurança para a maneira como vivemos. Basta observarmos o empenho necessário para vencer os vãos e manter pontes e viadutos estáticos, e permitir que encurtem distâncias; na forma como templos são concebidos e construídos com grandes salões para permitirem a reunião de pessoas e comunidades sob a proteção de intempéries; na fabricação de máquinas, no caso de trens a vapor do século XIX, que permitiram o confinamento de vapor e montagem de um robusto sistema de propulsão para deslocar grandes cargas e pessoas; ou a confecção de

equipamentos de conteúdo sob pressão, que é o caso de tanques, vasos de pressão e tubulações em refinarias, indústrias, plantas térmicas e termonucleares.

Desde meados do século XVIII, a construção com auxílio estrutural de metais cresce em todo o mundo, inicialmente usando ferro fundido - caso da construção da ponte *Coalbrookdale*, sobre o rio Severn, na Inglaterra, em 1779 - e posteriormente, com o uso do ferro forjado para a construção de grandes estruturas, dada sua maior segurança e menor propensão à falhas, caso da ponte suspensa *Menai* no país de Gales, construída na década de 1820, já no século XIX. No Brasil, o uso do ferro forjado pode ser visto na ponte sobre o rio Paraíba do Sul, inaugurada em 1857 na cidade de Paraíba do Sul - RJ.

Ainda no século XIX, acidentes envolvendo explosões de caldeiras e vasos de pressão, concebidos na primeira fase da 2ª revolução industrial, assim como acidentes em equipamentos feitos para as primeiras grandes refinarias de petróleo, são o estímulo ideal para que investimentos em segurança e pesquisa sejam implementados. E nesse contexto histórico, especificamente no ano de 1880, que é fundada a *American Society of Mechanical Engineers*, ASME, que passou a ser a principal autoridade em emissão de normas para fabricação, ensaios, qualificação e segurança de equipamentos para conteúdo sob pressão.

Ao adentrarmos no século XX, graças aos avanços científicos e tecnológicos alcançados até então, especificamente o processo Siemens-Martin, com o desenvolvimento de fornos de alta produtividade com lingotamento contínuo, o uso de laminadores, e a entrada da *Carnegie US Steel Company* no mercado norte americano, foi disponibilizada uma grande oferta de aço com preço baixo, por consequência tornando-o o material mais versátil para métodos construtivos, empregado desde a construção de estruturas civis para edificação à equipamentos e máquinas. Contemporaneamente, em 1902, é consolidada a *American Society for Testing and Materials*, ASTM, instituição que normatiza e estabelece critérios e parâmetros para fabricação de aços e testes de qualidade em materiais.

Dada essa ampla disponibilidade do aço, faz-se necessário estabelecer meios de ligação e união para montagem entre chapas, perfis e barras deste material, que são assim ofertados, devido as características da produção continua por lingotamento e laminação. As ligações entre os elementos construtivos podem ser divididas em dois grupos principais: 1. Por elementos de fixação – rebites, parafusos e porcas; 2. Ligações por soldagem – que é nosso alvo de análise neste trabalho – Método de ligação que não era possível para os ferros

fundidos e ferros forjados, dado o elevado teor de carbono destes materiais, e então demandavam uma elaborada e individualizada produção de itens para que fosse possível a montagem em campo. Considerando a expressiva demanda e rápida expansão da necessidade de técnicas de soldagem, é fundada no ano de 1919 a *American Welding Society*, AWS, instituição normativa e parametrizadora dos processos, requisitos tecnológicos e de segurança aos processos de soldagem.

Dado o avanço tecnológico alcançado, estabelecido e difundido desde a fundação da ASME, ASTM e AWS, ao falarmos em construção com utilização de materiais metálicos, sejam equipamentos de conteúdo sob pressão ou construções para uso civil, é inerente pensar em segurança da instalação. E quando se trata de segurança construtiva, robustos requisitos de projeto devem estar alinhados ao controle de qualidade na construção do empreendimento. O que implica dizer que, obedecer a critérios estabelecidos, seguir orientações de boas práticas e normas de regulamentação, é o que permite termos milhares de plantas industriais, siderúrgicas, de mineração, petroquímicas e nucleares, com um ínfimo número de acidentes reportados. Proporcionando oferta de produtos economicamente acessíveis, seguros e de qualidade, conforme as indústrias se desenvolvem.

## **1.2 CONTEXTUALIZAÇÃO DO PROBLEMA ENCONTRADO**

A avaliação de filmes radiográficos de soldagem permanece como um processo moroso e altamente dependente da análise humana devido à complexidade das imagens e à necessidade de discernir entre características sutis de interesse e pontos irrelevantes. Tal método de inspeção requer a perícia e qualificação de profissionais, que examinam meticulosamente cada radiografia objetivando encontrar descontinuidades, que podem se caracterizar como defeitos, ou não. No entanto, mesmo com treinamento especializado, a interpretação humana pode ser suscetível a erros e variações de interpretação, devido à fadiga visual e subjetividade, o que indiretamente, pode tornar o processo demorado e sujeito a inconsistências.

A introdução de tecnologias de inteligência artificial, como as redes neurais convolucionais, pode representar uma oportunidade para melhorar esse cenário. Essas redes podem automatizar e agilizar a análise de filmes radiográficos, fornecendo uma abordagem mais objetiva e consistente para a detecção de descontinuidades de solda. Desde que seja disponibilizado um grande volume de dados para treinamento, as CNN, podem identificar

de forma precisa e eficiente descontinuidades, e possivelmente, num futuro, reduzir a dependência de análise humana, acelerando o processo de inspeção.

A utilização de redes neurais convolucionais, CNN, para a análise de radiografias de solda é amplamente justificada devido à sua capacidade de aprendizado profundo e detecção precisa de padrões complexos em imagens. As CNN têm demonstrado sucesso em diversas aplicações de visão computacional, incluindo a detecção de defeitos em materiais e estruturas industriais. Estudos como o de Zhang et al. (2016) destacam a eficácia das CNNs na identificação de falhas de soldagem com alta precisão e rapidez, contribuindo assim para a segurança e integridade estrutural de equipamentos de caldeiraria técnica pesada.

Adicionalmente, a utilização de CNN pode oferecer benefícios complementares, como a automatização do processo de verificação de filmes e a redução do tempo necessário para análise de radiografias. Isso pode ser particularmente relevante na construção de equipamentos para a indústria nuclear, onde a segurança é a prioridade máxima.

Portanto, a aplicação de redes neurais convolucionais na análise de radiografias de solda não apenas melhora o desempenho e a eficiência do processo de inspeção, mas também desempenha um papel fundamental na garantia da segurança e integridade de equipamentos de caldeiraria técnica pesada, como os utilizados em usinas termonucleares e outras plantas industriais (Zhang et al., 2016).

### **1.3 USO DA INTELIGÊNCIA ARTIFICIAL COMO AUXÍLIO PROFISSIONAL**

Considerando um cenário de aceleração econômica e alta demanda para construção de equipamentos com exigências de verificação das juntas soldadas por radiografia, o auxílio ao especialista humano através do uso de uma ferramenta de inteligência artificial faz-se cabível. Nesse âmbito, potencializar o desenvolvimento e a utilização da tecnologia com vistas a promover o avanço científico e solucionar problemas concretos na área de controle de qualidade para a fabricação de equipamentos de caldeiraria técnica pesada, que hoje é extremamente dependente de profissionais qualificados, é um ponto central deste trabalho. Considerando ainda a sazonalidade das construções dos empreendimentos que exigem equipamentos deste tipo, em face dos vários ciclos econômicos da história do país, este nicho de mercado pode não dispor de profissionais devidamente qualificados para atender uma demanda repentina. Espera-se que a IA possa trazer ganhos na promoção da competitividade e no aumento da produtividade, de forma a avaliar uma quantidade de filmes superior ao que

um humano seja capaz em determinado intervalo de tempo. Permitindo a fabricação de equipamentos mais baratos, fato que pode reduzir os custos de implantação de usinas e indústrias indiretamente.

Existe uma revolução industrial em andamento, baseada em desenvolvimento tecnológico, que objetiva agilizar e automatizar processos, buscando ganhos na operação, e assim gerar diferencial competitivo para a empresa que a aplica. Nesse contexto, as técnicas de visão computacional trazem uma revolução para o processo de formatação de padrões e detecção de defeitos em produtos, tornando os processos de fabricação menos custosos e obtendo resultados cada vez mais padronizados. Habitualmente, ou por exigência normativa, a análise de defeitos em imagens radiográficas de soldas é realizada de forma visual e unitária, ou seja, filme a filme por inspetor qualificado. Considerando o grau de segurança exigido, e as características deste tipo de inspeção, faz-se necessário um processo visual e individualizado, o qual é demandado tempo considerável de inspeção, encarecendo e delongando o tempo de fabricação dos vasos de pressão em função da eficiência da junta soldada no componente. A proposta do uso de uma inteligência artificial tem por objetivo a criação de um modelo de rede neural artificial, especificamente do tipo convolucional, que irá auxiliar no processo de identificação de indicações de soldagem de juntas radiografadas, proporcionando celeridade e menor custo, e por consequência aumentando a eficiência das juntas e assim possibilitando equipamentos mais seguros, com espessuras de parede mais delgadas, que são mais baratos devido a menor quantidade de aço requerida e logisticamente mais viáveis, devido ao custo de manobras de peso envolvidas no processo de instalação.

O que pode-se dizer acerca da expectativa sobre o aprendizado profundo (Deep Learning) não é que os computadores comecem a pensar como humanos. E sim que assumam a capacidade de realizar tarefas que antes eram exclusivamente humanas, como reconhecimento de vozes, reconhecimento de padrões, identificação de defeitos, com tomadas de decisão independentes. E para isso, far-se-á uso de um conjunto suficientemente grande, capacidade de processamento computacional e algoritmos sofisticados o suficiente para desenvolvimento do aprendizado.

## **2. REVISÃO DA LITERATURA**

### **2.1. LIGAÇÃO DOS ELEMENTOS CONSTRUTIVOS POR SOLDAGEM**

Adotemos o entendimento que as normas, especificações e orientações estabelecidas pela ASTM são de responsabilidade do fabricante de matéria-prima, as siderúrgicas, e esse é periodicamente auditado e certificado para o fornecimento destes materiais. Portanto, infere-se que os elementos construtivos obtidos a partir destas matérias-primas são confiáveis, e então a atenção deve-se ser direcionada ao método de ligação entre os elementos, a soldagem.

Nas palavras de Walter e Michèle Pfeil (Pfeil e Pfeil, 2014), a soldagem é a união por coalescência do material, obtida por fusão das partes adjacentes, onde a origem da energia para promover essa fusão pode ser de origem química, óptica, mecânica ou elétrica. Neste trabalho, o foco de análise serão as juntas soldadas com o uso de arco elétrico, que é o método mais amplamente utilizado na indústria.

A soldagem por arco elétrico se dá, basicamente, entre um eletrodo metálico e o componente a ser soldado, denominado metal de base, com desprendimento contínuo de material do eletrodo projetando-se para a poça de fusão, que deve ser obrigatoriamente protegida da atmosfera, inibindo assim a ação oxidante do ambiente no ferro presente tanto no metal de base quanto no eletrodo, e assim, após a solidificação, permitirá a união permanente dos dois elementos.

A soldagem por arco elétrico pode se apresentar de algumas formas distintas, distinguindo-se na forma como o par arco e poça de fusão são protegidos da atmosfera. Dentre as mais comuns, usualmente adotadas na indústria, temos:

- a. SMAW - *Shielded Metal Arc Welding* – Método manual, conhecido no Brasil como soldagem por eletrodo revestido. Onde o revestimento é fundido juntamente do eletrodo, gerando em parte, fumos de solda para a proteção gasosa da poça de fusão, e outra parte transformando-se em escória, que deverá ser removida a cada cordão aplicado;
- b. GMAW - *Gas Metal Arc Welding* – Método semiautomático de soldagem com proteção gasosa, que pode ser MIG – *Metal Inert Gas*, ou MAG – *Metal Active Gas*, usa-se um arame de soldagem sem revestimento como eletrodo, que é inserido na poça de fusão de forma contínua, e durante a fusão, é protegido pelo fluxo de gases que é injetado junto do arame pelo equipamento de soldagem;

- c. FCAW - *Flux-cored arc welding* – Método semiautomático de adição contínua de material, no qual o eletrodo apresenta-se em forma de arame tubular, com o fluxo – nome dado ao material fusível (ou fundível) que gera o fumo para proteção gasosa – no interior do arame, que similar ao método SMAW, o fluxo tem por finalidade ser fundido juntamente do arame, gerando em parte, fumos de solda para a proteção gasosa da poça de fusão, e outra parte transformando-se em escória, que deverá ser removida a cada cordão aplicado;
- d. GTAW - *Gas-Shielded Tungsten Arc Welding* – Também conhecido como TIG - *Tungsten Inert Gas*. Neste processo, diferentemente dos outros citados anteriormente, o eletrodo não é fundido, conseqüentemente não é consumido. O eletrodo estabelece um arco elétrico no metal de base, sob proteção gasosa, que é injetado ao redor do eletrodo, protegendo-o da atmosfera oxidante, e o metal de adição é fundido pelo calor do arco voltaico, assim sendo adicionado a poça de fusão.
- e. SAW – *Submerged Arc Welding* – Soldagem por arco submerso é um método no qual o eletrodo apresenta-se como arame sólido, injetado na poça de fusão de forma contínua, no entanto a proteção da poça de fusão é dada pela adição de um fluxo de material granular que é fundido juntamente do eletrodo, gerando assim os fumos de solda necessários a eliminar a presença de oxigênio na poça, e também pode permitir a adição de elementos de liga à solda.

Considerando a suscetibilidade a defeitos, algumas variáveis das condições de soldagem em juntas metálicas devem ser monitoradas, pois podem interferir na qualidade da execução, e por consequência inserir defeitos no cordão de solda. Cuidados com a variação de corrente elétrica, estabilidade do arco, remoção correta da escória entre as camadas de solda, fluxo de gases adequado, e controle no pré-aquecimento e pós-aquecimento do cordão são partes fundamentais de uma lista de verificações de uma soldagem profissional.

## **2.2. CONTROLE E INSPEÇÃO DE SOLDAGEM**

De forma a garantir a correta execução da soldagem, a AWS e ASME, impõem exigências durante todo o ciclo de execução de uma solda, com requisitos mínimos para aceitação do resultado obtido. E isso tem início ainda na Qualificação do Procedimento de Soldagem – QPS, que deve ser previamente definido, executado em uma amostra do material, inspecionada e ser aprovada nos ensaios não destrutivos – END – e destrutivos, ou ensaios de laboratório. Todas essas etapas são acompanhadas por inspetores também

qualificados pelos critérios destas instituições, que verificam desde os certificados de matéria-prima do material a ser utilizado, passando pela verificação da geometria e integridade da junta a ser soldada (chanfro) até os resultados dos ensaios requeridos, que devem ser comparados aos estabelecidos nas normas de qualificação. Se todos os requisitos previstos em norma forem alcançados, o QPS é considerado aprovado, e então é gerada uma Especificação de Procedimento de Soldagem – EPS, que compilará os parâmetros adotados durante a qualificação, e estes passarão a ser regra a ser seguida durante a fabricação do equipamento.

Posteriormente, durante a utilização daquele procedimento já qualificado, o conteúdo definido na EPS deve ser rigorosamente seguido, e o papel dos inspetores é garantir que os parâmetros ali estabelecidos sejam cumpridos, com isso espera-se que as soldas efetuadas na construção do equipamento apresentem os mesmos resultados daquelas feitas durante a etapa de qualificação.

### **2.3. ENSAIOS NÃO DESTRUTIVOS – END – EM SOLDAS**

Todas as juntas de soldagem para equipamentos e estruturas, conforme exigido nas normas, além de serem supervisionadas durante a execução, devem ser inspecionadas quando finalizadas, e estas inspeções também devem ser executadas por inspetores qualificados, conforme exigência ASME e/ou AWS.

Os END em soldas podem ser divididos em dois grupos: os ensaios superficiais, que tem por finalidade verificar as condições e defeitos nas camadas aparentes dos cordões, e os ensaios volumétricos, que objetivam a verificação da integridade no interior da junta, nos cordões internos e entre as camadas de soldagem internas, e devem garantir que o material adicionado à junta se fundiu de maneira uniforme ao material de base, e não apresenta descontinuidades (defeitos).

O primeiro ensaio superficial a ser executado em uma solda é o Visual de Solda, e é usado para determinar coisas como a condição superficial, alinhamento da junta, altura do reforço, geometria do cordão e presença de mordeduras, que é quando a margem do chanfro funde, mas não é devidamente preenchida pelo metal de adição, deixando uma reentrância na superfície do cordão.

Em seguida, de forma a verificar possíveis defeitos internos que estejam abertos à superfície, e que não sejam perceptíveis ao exame visual, é escolhido um dos métodos de exames de defeitos superficiais, que pode ser por líquido penetrante – da sigla LP, ou PT (*Penetrant Test* em inglês) – ou por partículas magnéticas PM, ou MT (*Magnetic Test* em inglês), não sendo complementares, e sim alternativos um ao outro. Reservando exclusividade ao LP para superfícies não magnetizáveis, como aços austeníticos e demais metais. Para a verificação volumétrica das juntas soldadas tem-se como opções a inspeção por ultrassom US – ou UT (*Ultrasonic Test* em inglês) – e a radiografia RX – RT (*Radiography Test*). – Neste ponto, cabe divulgar uma técnica interessante: apesar de serem amplamente adotadas como métodos de inspeção alternativos um ao outro, o UT pode ser usado de maneira complementar ao RT, quando identificado alguma descontinuidade em solda de equipamentos de parede espessa, assim utiliza-se o método de UT para determinar a profundidade do defeito para que possa ser escolhido o lado mais viável à abertura da junta para remoção do defeito e recomposição da solda de maneira integral, uma vez que a radiografia é capaz de fazer apenas a representação bidimensional da junta analisada.

#### **2.4. TIPOS DE DEFEITOS EM SOLDAS**

Afim de equalizarmos o entendimento e definirmos o conceito de descontinuidade, adotaremos o conceito de Marques (Marques et al, 2006), que define descontinuidade como “*uma interrupção ou uma violação da estrutura típica ou esperada de uma junta soldada*”. Conforme requisitos exigidos a cada projeto, uma descontinuidade pode ser considerada prejudicial para a utilização futura da junta e assim constituindo-se em defeito, que exigirá ação corretiva. Considerando que as juntas soldadas podem apresentar descontinuidades em sua estrutura interna e/ou externa. E a depender das características, dimensões e/ou quantidade destas descontinuidades, podem ser caracterizadas como defeitos, a junta pode ser considerada reprovada ou aprovada.

A depender da criticidade do equipamento para a planta, ou risco de segurança, os ensaios não destrutivos exigidos em norma podem ser mais ou menos rigorosos, alterando-se o critério de aceitação a ser utilizado. Como exemplo, temos os equipamentos do circuito primário, e aqueles do sistema de resfriamento de emergência, de uma usina termonuclear, enquadrados na seção III do código ASME, onde prevê que as juntas para este tipo de componente devam ser cem por cento (100%) radiografadas após a verificação por outros métodos de inspeção superficial. E caso estes equipamentos estivessem em ponto de menor

relevância dentro da planta termonuclear, poderiam ter esse percentual de verificação reduzido, ou talvez suprimido, devido à baixa criticidade para o funcionamento e segurança da planta.

Conforme previsto no *Welding Inspection Handbook*, da AWS, uma solda que não está de acordo com qualquer, ou todos, os requisitos especificados de um procedimento ou de um código, é considerada como uma solda defeituosa. E uma solda com defeito é impossível de avaliar sem referência a algum padrão ou requisito específico. Considerando isso, deve-se ter o entendimento sobre os tipos mais comuns de descontinuidades em cordões de solda, que podem ser divididos da seguinte maneira:

**A. Devido a procedimento/processo:**

- i. Geométrico: desalinhamento, distorção, dimensão final, tamanho da solda, sobreposição, perfil de solda, convexidade, concavidade e reforço de solda;
- ii. Solda/Estrutural: falta de fusão, falta de penetração, mordeduras, preenchimento incompleto, inclusões de escória e/ou tungstênio, trincas de variados tipos, diferentes tipos de porosidade e irregularidades de superfície (rugosidade, respingos e aberturas de arco);

**B. Descontinuidades metalúrgicas:**

- i. Mecânico: Tensões admissíveis, ductilidade e dureza;
- ii. Químico: variações químicas e resistência à corrosão;

**C. Descontinuidades no metal de base, por laminação, delaminação, formação lamelar e/ou emendas;**

Considerando a ampla variedade de descontinuidades, direcionemos o foco às que são verificáveis pelo ensaio volumétrico de RX (RT): Falta de fusão e/ou penetração; Porosidade; Inclusões de escórias e tungstênio; E trincas; Que são descontinuidades internas, imputadas durante a concepção da solda.

#### **2.4.1. POROSIDADE**

É um tipo de descontinuidade geralmente esférica, mas pode ser alongada, e tem como característica a retenção de pequenas bolhas de gás durante o resfriamento da poça de fusão.

Frequentemente causada pela corrente elétrica alta, ou distância inadequada entre eletrodo e metal de base, conforme descrito em WELDING INSPECTION HANDBOOK, (AWS, 2015). A porosidade também pode ser causada devido à contaminação por óleo, graxa ou óxidos presentes na junta a ser soldada, ou também pela evolução dos gases de proteção que podem ficar aprisionados durante a solidificação do metal. Podem ocorrer uniformemente distribuída, em grupos, alinhada ou como porosidade vermicular (em forma de verme). Visualmente se comporta como pequenos vazios (poros). Na radiografia se apresenta visualmente como manchas escuras redondas ou irregulares, em grupos ou fileiras, como descrito em (MARQUES et al, 2009). Ao analisarmos a figura 1, abaixo, pode-se notar que as indicações de porosidade apresentam-se como pontos escuros na radiografia. Isso é causado pela ausência de sólidos naquele ponto permitindo a passagem da radiação, e por consequência atingindo mais o filme. A formação de porosidade pode ser evitada mantendo a limpeza para soldagem, fornecendo barreiras de proteção na prevenção da perda de blindagem de gás e mantendo o equipamento em manutenção, WELDING INSPECTION HANDBOOK, (AWS, 2015).

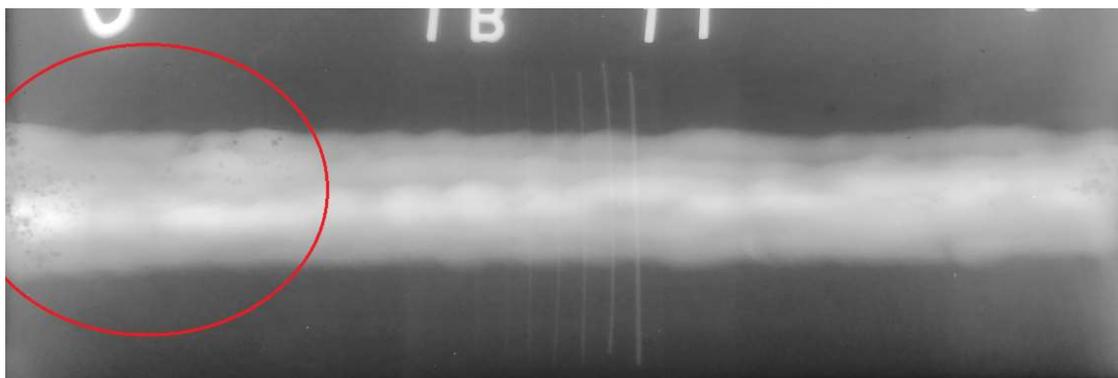


Figura 1: Exemplo de porosidade identificável em uma radiografia de soldagem

#### **2.4.2. INCLUSÕES DE ESCÓRIAS E/OU TUNGSTÊNIO**

Como descrito no WELDING INSPECTION HANDBOOK, (AWS, 2015), existem dois tipos básicos de inclusões relacionadas a soldas, inclusão de escória e inclusão de tungstênio.

Inclusão de escória: Tipicamente, os processos de soldagem que utilizam fluxo, sejam para a formação de fumos de solda para proteção gasosa, ou para adição de elementos de liga, geram escória como parte inerente do processo de soldagem. Naturalmente, essa escória deveria ser separada do metal líquido, solidificando-se na superfície do cordão de solda. No



do metal de base ou do metal de solda previamente depositado. Técnicas de soldagem incorretas, preparação inadequada dos materiais para soldagem ou projetos de junta incorretos promovem a falta de fusão nas soldas, (AWS, 2015).

Como pode-se notar na figura 3, abaixo, na radiografia se apresenta visualmente como uma linha escura orientada na direção da costura da solda, (TELLES, 2010).



Figura 3: Exemplo de falta de fusão em uma radiografia de soldagem

Decorrem, em geral, devido à corrente insuficiente (Pfeil e Pfeil, 2014). Alguns motivos para esta descontinuidade podem ser condução imprecisa do processo de soldagem, contaminação da junta ou energia de soldagem insuficiente (TELLES, 2010). A limpeza insuficiente da junta ou pré-soldagem pode contribuir para a fusão incompleta, mesmo que as condições e a técnica de soldagem sejam adequadas (AWS, 2015).

#### **2.4.4. FALTA DE PENETRAÇÃO**

A falta de penetração é um tipo de defeito que ocorre devido a uma falha no metal de base em se fundir e coalescer ao metal de adição. É uma descontinuidade característica da raiz, pois localiza-se na interface metal de base/metal de solda. A área não fundida permite concentrações de tensão que podem causar falha sem deformação apreciável. Mesmo que as tensões de serviço na estrutura possam não envolver tensão ou flexão neste ponto, as tensões de retração e a conseqüente distorção das peças durante a soldagem poderão causar trincas na área não fundida. Essas fissuras podem progredir, à medida que cordões sucessivos são depositados, até que se estendam através, ou quase através, de toda a espessura da solda, (AWS, 2015).

Embora a falta de penetração na junta possa, em alguns casos, ser devido à falha na dissolução de óxidos e impurezas superficiais, a transferência de calor reduzida na junta é uma fonte mais comum dessa descontinuidade. A causa mais frequente deste tipo de descontinuidade é a utilização de um desenho de chanfro não adequado ao processo de

soldagem escolhido. Outro exemplo é quando uma junta é soldada de um lado apenas, e a penetração completa da junta pode não ser obtida de forma consistente.

Como pode-se notar na figura 4, abaixo, na radiografia se apresenta visualmente como uma área escura com bordas retas bem definidas, (TELLES, 2010).

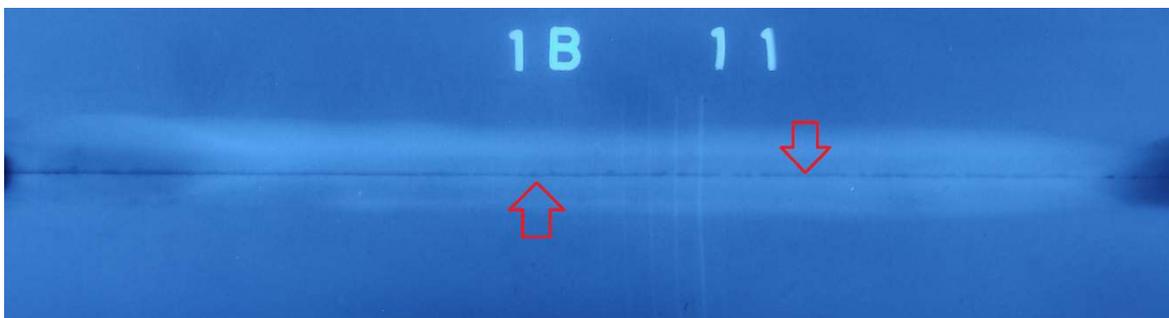


Figura 4: Exemplo de falta de penetração em uma radiografia de soldagem

#### 2.4.5. TRINCAS

As trincas são significativas concentradoras de tensões, e por isso são consideradas como o tipo mais preocupante de descontinuidade, pois podem favorecer a fratura frágil.

Uma forma objetiva para entender esse tipo de descontinuidade é considerar que uma trinca é o resultado da incapacidade do material em responder às solicitações impostas localmente pelas tensões decorrentes do processo de soldagem. Algumas causas para este tipo de ocorrência podem ser: Aquecimento e resfriamento abrupto de materiais com alto teor de carbono, que podem gerar microestruturas no contorno do cordão diferente da microestrutura presente no metal de base, implicando em contrações e distorções no metal, (TELLES, 2010); Devido a essas tensões, uma junta, ou qualquer região adjacente, como a zona termicamente afetada (ZTA) pode ser incapaz de suportar deformação apreciável sem falha. As trincas também podem ocorrer por desprendimento de hidrogênio, ou outros elementos indesejados, que são contaminantes do fluxo de soldagem, (TELLES, 2010).

As trincas que ocorrem durante a solidificação são comumente chamadas de trinca a quente. Trinca a frio são aquelas que ocorrem depois que a solda esfriou até a temperatura ambiente ou depois que a soldagem foi colocada em serviço, (AWS, 2015).

Na radiografia se apresenta visualmente como linhas irregulares denteadas e muitas vezes muito fracas; podem também aparecer como “caudas” nas imagens. As trincas são

detectadas quando essas linhas ou “caudas” tem a sua extensão seguindo uma direção específica, (TELLES, 2010).

## **2.5. ENSAIO RADIOGRÁFICO EM SOLDAS**

Por ser capaz de detectar os defeitos internos de uma solda, a inspeção radiográfica é um processo largamente utilizado para inspeção de soldas, e embora as soldas não radiografadas sejam permitidas, em alguns casos, pela seção VIII, divisão 1, do código ASME, é pratica usual exigir-se pelo menos a radiografia parcial (amostragem) para as soldas principais, em todos os vasos de pressão, (TELLES, 2010). No caso da seção III, divisão 1, subseção NB e NC (Componentes classe 1 e 2 para instalações nucleares), é obrigatória a verificação das juntas por RX, para a fabricação de vasos de pressão.

Uma das principais vantagens da radiografia é o fato de a inspeção resultar em documentos permanentes, que são os filmes radiográficos, (TELLES, 2010), e estes se devidamente armazenados, poderão ser consultados mesmo após vários anos decorridos.

Segundo o código ASME, seção V, artigo 2, os requisitos gerais para execução de um ensaio radiográfico incluem, entre outros:

- A necessidade de um procedimento escrito, que deve reger a maneira como o ensaio é executado. Cada procedimento deve incluir pelo menos as seguintes informações, conforme aplicável: (a) tipo de material e faixa de espessura; (b) isótopo ou tensão máxima de raios-X usada; (c) distância fonte-objeto; (d) distância do lado da fonte do objeto ao filme; (e) tamanho da fonte; (f) marca e designação do filme; e (g) telas usadas;

- Demonstração do procedimento. A demonstração de densidade e o Indicador de Qualidade de Imagem (IQI), requisitos de imagem do procedimento escrito, em produção, ou técnicas radiográficas, deve ser considerada evidência satisfatória de conformidade com esse procedimento;

- Rugosidades ou irregularidades superficiais no interior (quando acessível) e no exterior da solda devem ser removidas por qualquer processo adequado a tal grau que a imagem radiográfica resultante devido a quaisquer irregularidades de superfície, não possa mascarar ou ser confundida com a imagem de qualquer descontinuidade;

- A superfície acabada de todas as juntas soldadas pode ser nivelada com o material de base ou pode ter cordões razoavelmente uniformes, com reforço que não deve exceder o especificado na seção específica para o componente em questão;

- Um símbolo de chumbo "B", com dimensões mínimas de 1/2 pol. (12,7 mm) de altura e 1/16 pol. (1,6 mm) de espessura, deve ser anexado à parte traseira de cada suporte de filme durante cada exposição para determinar se a radiação de retroespalhamento está expondo o filme.

Tais exigências permitirão maior qualidade da imagem gerada, possibilitando uma interpretação mais precisa das indicações apresentadas no filme, permitindo diferenciar entre as indicações quais são descontinuidades reais na soldagem. Caso tais exigências não sejam cumpridas, a radiografia pode ser passível de nulidade.

## 3. INTELIGÊNCIA ARTIFICIAL

### 3.1. O QUE É INTELIGÊNCIA ARTIFICIAL

Como ponto de partida, temos as sentenças abaixo, as quais não há um autor devidamente atribuído, e que podem ser encontradas em diferentes páginas da *internet*, e que, no entanto, sintetizam de modo muito objetivo o conceito de inteligência artificial:

*É o Ramo da ciência da computação que se propõe a elaborar algoritmos que simulem a capacidade humana de raciocinar, perceber e tomar decisões para resolver problemas, isto é, a capacidade de ser inteligente.*

*... é um avanço tecnológico que permite que sistemas simulem uma inteligência similar à humana, indo além da programação de ordens específicas para tomar decisões de forma autônoma, baseadas em padrões de enormes bancos de dados.*

Inteligência Artificial é um conceito muito amplo e envolve inúmeras técnicas, como por exemplo, Algoritmos Evolucionários, Algoritmos de Enxame, Lógica Nebulosa, Sistemas Especialistas, Algoritmos com Inspiração Quântica, Redes Neurais, etc. Este trabalho foca na utilização de Redes Neurais Artificiais – RNA - para reconhecimento de problemas em radiografias de soldas, através da análise das imagens radiográficas.

### 3.2. REDES NEURAIS

Muitas tarefas desempenhadas pelo cérebro ainda intrigam cientistas e estudiosos, como a capacidade do cérebro de reconhecer algo familiar entre um grande número de amostras, ou uma pessoa em uma multidão, somente em algumas frações de segundo. Várias respostas sobre o funcionamento do cérebro não foram encontradas, e ainda hoje permanecem abertas. No entanto, é consenso que o cérebro humano é uma máquina altamente poderosa e complexa, que por sua vez é formado de unidades elementares, entre elas, os neurônios. E é neles que as informações são recebidas, processadas e transmitidas adiante (LECUN et al, 2015).

Fato conhecido sobre o funcionamento do cérebro é a capacidade de desenvolver regras através de experiências e situações anteriormente vivenciadas. Inspirados neste entendimento, diversos pesquisadores sentiram-se estimulados a tentar criar sistemas inteligentes capazes de realizar tarefas de classificação, processamento de imagens, reconhecimento de padrões, entre outras. Como resultado surgiu o modelo de neurônio artificial, para posteriormente o agrupamento destes neurônios interconectados dar origem a chamada rede neural artificial (GOODFELLOW et al, 2016)

As redes neurais artificiais são modelos computacionais que se inspiram nas redes neurais biológicas dos seres vivos para processar informações e realizar tarefas complexas. Essas redes são compostas por uma série de neurônios artificiais interconectados, que são capazes de realizar operações matemáticas simples, mas que, em conjunto, podem executar tarefas sofisticadas, como reconhecimento de padrões, classificação de dados, processamento de linguagem natural, entre outras (RUMELHART et al, 1986).

Essa capacidade de inferir é a principal característica que diferencia as redes neurais artificiais de outros modelos de aprendizado de máquina. Enquanto outros modelos, como as árvores de decisão e as regressões, seguem um caminho determinístico para chegar a uma resposta, as redes neurais têm a capacidade de generalizar a partir de exemplos, permitindo que elas possam lidar com dados complexos e realizar tarefas em que as soluções não são evidentes (GOODFELLOW et al, 2016).

O processo de aprendizado em uma rede neural artificial é baseado no ajuste dos pesos sinápticos, que são responsáveis por regular a intensidade do sinal que passa entre os neurônios. Inicialmente, os pesos são atribuídos aleatoriamente, e a rede passa por um

processo de treinamento supervisionado, em que exemplos rotulados são apresentados à rede para que ela possa aprender a mapear as entradas para as saídas desejadas (MINSKY e PAPERT, 1969).

Durante o treinamento, a rede ajusta os pesos sinápticos para minimizar o erro entre as saídas esperadas e as saídas produzidas pela rede. Esse processo é realizado através do algoritmo de *backpropagation*, que calcula o gradiente do erro em relação aos pesos e ajusta-os de forma a minimizá-lo. Esse processo é repetido para cada exemplo de treinamento até que a rede seja capaz de generalizar para novos exemplos (RUMELHART et al, 1986)..

Uma vez treinada, a rede pode ser usada para realizar inferência em novos exemplos, em que a saída é inferida a partir das entradas apresentadas à rede. A capacidade de inferir a partir de exemplos é o principal aspecto do aprendizado de máquina através das redes neurais artificiais, permitindo que elas possam ser usadas em uma ampla variedade de aplicações, como reconhecimento de voz, detecção de fraudes, diagnóstico médico, entre outras (Dahl et al, 2014).

Em resumo, as redes neurais artificiais são modelos computacionais que se inspiram nas redes neurais biológicas para realizar tarefas sofisticadas de aprendizado de máquina. Elas são compostas por neurônios artificiais interconectados que podem aprender a partir de exemplos rotulados, ajustando seus pesos sinápticos através do algoritmo de *backpropagation*. A capacidade de inferir a partir de exemplos é o principal aspecto do aprendizado de máquina através das redes neurais artificiais, permitindo que elas possam lidar com dados complexos e realizar tarefas em que as soluções não são evidentes.

### **3.3. CONTEXTUALIZAÇÃO HISTÓRICA DAS REDES NEURAS ARTIFICIAIS**

Elencando os acontecimentos ocorridos na história da inteligência artificial em ordem cronológica, e dividindo essa cronologia em épocas, ou ondas, como (GOODFELLOW et al, 2016) sugere, a evolução histórica das pesquisas sobre redes neurais artificiais, ou aprendizado profundo, de um modo geral, em três ondas de desenvolvimento: conhecidas como Cibernética entre as décadas de 1940 a 1960, Conexionismo nas décadas de 1980 a 1990, e aprendizado profundo, propriamente dito, a partir do ano de 2006 aos dias atuais.

A primeira onda começa com a cibernética nas décadas de 1940-1960, com o desenvolvimento de teorias de aprendizagem biológica, onde destacam-se os estudos de

McCulloch e Pitts, 1943; e Hebb, 1949, e implementações dos primeiros modelos, como o perceptron de Rosenblatt, 1958, permitindo o treinamento de um único neurônio.

### 3.3.1. A ONDA DA CIBERNÉTICA

Essa primeira época, ou onda, pode ser enquadrada pelos eventos ocorridos entre os anos 1940 aos anos de 1960. Época caracterizada pelo surgimento dos primeiros modelos de neurônio artificial, e nas hipóteses de funcionamento do neurônio biológico.

Quando em 1943, o matemático Walter Pitts e o neurofisiologista Warren McCulloch escreveram um artigo sobre como os neurônios funcionariam e assim modelaram uma rede neural simples, usando circuitos elétricos, Pitts e McCulloch criaram um modelo para redes neurais baseado em matemática e algoritmos denominado lógica de limiar (*threshold logic*), que abriu o caminho para a pesquisa da rede neural, sendo essa dividida em duas abordagens: uma focada em processos biológicos no cérebro, enquanto a outra na aplicação de redes neurais à inteligência artificial.

Em 1949, Donald Hebb na obra *The Organization of Behavior*, apontou o fato de que os caminhos neurais são fortalecidos toda vez que são usados. Um conceito fundamentalmente essencial para a maneira como os seres aprendem. Se dois nervos dispararem ao mesmo tempo, a conexão entre eles é melhorada, argumentou.

Adicionalmente, com o avanço tecnológico alcançado para os computadores na década de 1950, foi possível simular uma rede neural hipotética. No entanto, infelizmente, as primeiras tentativas, impostas por Nathaniel Rochester nos laboratórios de pesquisa da IBM, não foram bem-sucedidas. Apesar dos resultados, pesquisas continuaram, e em 1956, o Projeto de Pesquisa de Verão *Dartmouth* sobre Inteligência Artificial impulsionou tanto à Inteligência Artificial como às Redes Neurais, resultando em estímulos para a pesquisa em IA na parte de processamento neural.

Imediatamente após o projeto *Dartmouth*, John Von Neumann sugeriu imitar funções de neurônios usando tubos de vácuo ou relés telegráficos. Quando, ao mesmo tempo, o neurobiologista Frank Rosenblatt, intrigado pelo funcionamento dos olhos de moscas, onde grande parte da decisão de fuga deste inseto é feito ainda nos olhos, trabalhava no *Perceptron*. Oriundo desta pesquisa, um *Perceptron* foi fisicamente construído e pode ser considerada a mais antiga rede neural ainda em funcionamento.

Com uma única camada, o *Perceptron* foi útil na classificação de um conjunto de entradas com valor contínuo para saída em duas classes distintas. Este calculava uma soma ponderada das entradas, subtraía um limite e passava um dos dois resultados como saída.

Ainda nos anos 1950, especificamente em 1959, Bernard Widrow e Marcian Hoff, de Stanford, desenvolveram modelos denominados “ADALINE” e “MADALINE”. ADALINE foi desenvolvido para reconhecer padrões binários de modo que, se ele estivesse lendo bits de transmissão de uma linha telefônica, poderia prever o próximo bit. MADALINE foi a primeira rede neural aplicada a um problema do mundo real, usando um filtro adaptativo que elimina ecos nas linhas telefônicas. Embora antigo, assim como os sistemas de controle de tráfego aéreo, MADALINE ainda está em uso comercial.

Considerando o sucesso desses casos, houve uma grande expectativa em torno do potencial das redes neurais e técnicas de inteligência artificial, que não pôde ser correspondida devido, inclusive, às limitações eletrônicas na época. Modelos como o *Perceptron* e *Adaline*, ou nestes baseados, são chamados de modelos lineares, e continuam sendo alguns dos modelos de aprendizado de máquina mais amplamente usados, embora em muitos casos sejam treinados de maneiras diferentes dos modelos originais.

No entanto, modelos lineares apresentam muitas limitações. Sua capacidade de aprendizado é limitada à funções lineares, e operações lógicas como NOT, AND ou OR. O *Perceptron* foi comprovadamente limitado por Marvin Minsky e Seymour Papert, no livro *Perceptrons. An Introduction to Computational Geometry* de 1969, quando mostraram que o *perceptron* é incapaz de resolver funções não lineares, como XOR. Adicionalmente, argumentaram que o *perceptron* poderia ser capaz de lidar com funções não lineares caso fosse dotado de múltiplas camadas, contudo, o modelo de Rosenblatt não funcionava para múltiplas camadas. Os críticos que observaram essas falhas, causaram uma reação contra o aprendizado biologicamente inspirado em geral.

Na mesma época, escritores produziram obras que descreviam possíveis efeitos que teriam “máquinas pensantes” à sociedade. Essa discussão sobre o efeito da Inteligência Artificial na vida humana, que trouxeram críticas a pesquisa em redes neurais. Somando-se aos poucos progressos alcançados, que resultaram em redução drástica do financiamento em pesquisas, assim dando início a uma grande queda na popularidade das redes neurais. Como resultado, até os anos 80, os estudos envolvendo RNA foram diminuindo, com poucas

pesquisas sendo realizadas. Esse período de crescimento reduzido durou até 1981, e ficou conhecido como o Inverno da IA (*AI Winter*).

Nos anos posteriores, uma série de eventos determinariam os novos rumos que o uso da inteligência artificial teria, diminuindo a expectativa de criação de cérebros artificiais, em prol da criação de dispositivos úteis. Adicionalmente, houve avanços significativos no estudo de RNA para modelagem de imagens e de sequências. Em relação ao processamento de imagens, Fukushima, (1980) introduziu um modelo de arquitetura de RNA, chamada *Neocognitron*, inspirado na estrutura do sistema visual dos mamíferos, para processamento de imagens. A *Neocognitron* foi posteriormente utilizada como base para o desenvolvimento da RNA de arquitetura convolucional (LECUN, 1989), que é uma das principais arquiteturas de RNA da atualidade.

### **3.3.2. A ONDA DO CONEXIONISMO**

Na década de 1980, a segunda onda de pesquisa de redes neurais emergiu em grande parte por meio de um movimento chamado conexionismo, ou processamento paralelo distribuído (Rumelhart et al., 1986; McClelland et al., 1995).

A ideia central no conexionismo é que um grande número de unidades computacionais simples pode atingir um comportamento inteligente quando conectadas em rede. Essa percepção se aplica tanto aos neurônios em sistemas nervosos biológicos quanto a camadas ocultas de modelos de aprendizado profundo.

Vários conceitos-chave surgiram durante o movimento do conexionismo da década de 1980 que permaneceram centrais para o aprendizado profundo de hoje. Um desses conceitos é o de representação distribuída (Rumelhart et al., 1986). No entanto, o conceito que pode ser considerado como definidor da era conexionista foi a implementação eficiente do algoritmo de retropropagação (*back-propagation*) para ajuste de pesos. Apesar deste algoritmo ter sido proposto anteriormente por Werbos, (1974), foi somente através do trabalho de Rumelhart et al, (1986) que este se popularizou. A partir deste trabalho, que foi possível o treinamento eficiente de RNAs compostas por múltiplas camadas de *perceptron*, e assim foram nomeadas como redes neurais *perceptron* de múltiplas camadas (do inglês, *multilayer perceptrons* – MLP).

Adicionalmente, durante a década de 1990, durante o conexionismo, pode-se ressaltar que os pesquisadores fizeram avanços expressivos na modelagem de sequências com redes neurais, e na modelagem de imagens através de RNA.

Hochreiter (1991) e Bengio et al. (1994) identificaram e definiram o problema de desaparecimento de gradiente que não permitia que as redes neurais recorrentes conseguissem modelar sequências longas. E posteriormente, Hochreiter e Schmidhuber (1997) introduziram a rede de memória de longo prazo (LSTM), que atualmente é utilizada para tarefas de modelagem de sequência, incluindo tarefas de processamento de linguagem natural.

O problema de desaparecimento de gradiente, não era um obstáculo exclusivo para as redes recorrentes, como também impactava as MLP. Mais especificamente, as MLPs com quatro ou mais camadas ocultas, que não conseguiam ser treinadas de forma eficiente através do algoritmo de retropropagação.

Acontece que o gradiente, que é utilizado para ajustar os pesos (funções responsáveis pelo aprendizado em si) em função do erro obtido na saída da rede, diminuía ao ser retropropagado pelas camadas predecessoras à saída, de modo que, após várias camadas ocultas, o gradiente ao chegar às primeiras camadas da rede estava tão minimizado que o ajuste dos pesos dessas camadas se tornava irrelevante. Em virtude disso, por muito tempo, o uso de RNA com muitas camadas ocultas era inviável, ou seja, a capacidade de aprendizado que as RNA desta época podiam alcançar estava limitada.

A segunda onda de pesquisa em redes neurais durou até meados da década de 1990, quando esforços de empreendimentos baseados em redes neurais e outras tecnologias de IA começaram a fazer reivindicações ambiciosas e irrealistas, enquanto buscavam investimentos. Quando a pesquisa de IA não atendeu a essas expectativas irracionais, os investidores ficaram desapontados. Simultaneamente, outros campos de aprendizado de máquina fizeram avanços. Máquinas de kernel (Boser et al., 1992; Cortes e Vapnik, 1995; Schölkopf et al., 1999) e modelos gráficos (Jordan, 1998) alcançaram bons resultados em muitas tarefas importantes. Esses dois fatores levaram a um declínio na popularidade das redes neurais que durou até 2007

### 3.3.3. A ONDA DO APRENDIZADO PROFUNDO

Pode-se considerar que a terceira onda de pesquisa de redes neurais começou com um pontual avanço em 2006, quando Geoffrey Hinton mostrou que um tipo de rede neural chamada *rede de crença profunda* poderia ser treinada com eficiência usando uma estratégia chamada pré-treinamento “guloso” em camadas (Hinton et al., 2006). Diz-se que esta rede treina de forma “gulosa” (*greedy*), devido ao fato de ela encontrar a melhor solução local, que é considerada suficiente para a aplicação em questão, mas que não necessariamente representa a melhor solução global.

Outros grupos de pesquisa mostraram rapidamente que a mesma estratégia poderia ser usada para treinar muitos outros tipos de redes profundas (Bengio et al., 2007; Ranzato et al., 2007a) e sistematicamente, ajudou a melhorar a generalização em exemplos de teste. Como consequência, as redes neurais profundas superaram os sistemas de IA concorrentes baseados em outras tecnologias de aprendizado de máquina.

Essa terceira onda de popularidade das redes neurais continua até o presente momento, embora o foco da pesquisa de aprendizagem profunda tenha mudado drasticamente durante essa atual onda.

Resumidamente, pode-se dizer que a terceira onda começou com foco em novas técnicas de aprendizado não supervisionado e na capacidade de modelos profundos de generalizar bem a partir de pequenos conjuntos de dados, porém, hoje, há mais interesse em algoritmos de aprendizado supervisionado mais antigos e na capacidade de modelos profundos de alavancar grandes conjuntos de dados rotulados.

Na literatura atual, as redes neurais têm sido extensivamente exploradas devido à sua capacidade de aprendizado e representação de dados complexos. Na Figura 5, demonstra-se uma grande variedade de diagramas representativos de redes neurais conhecidas.

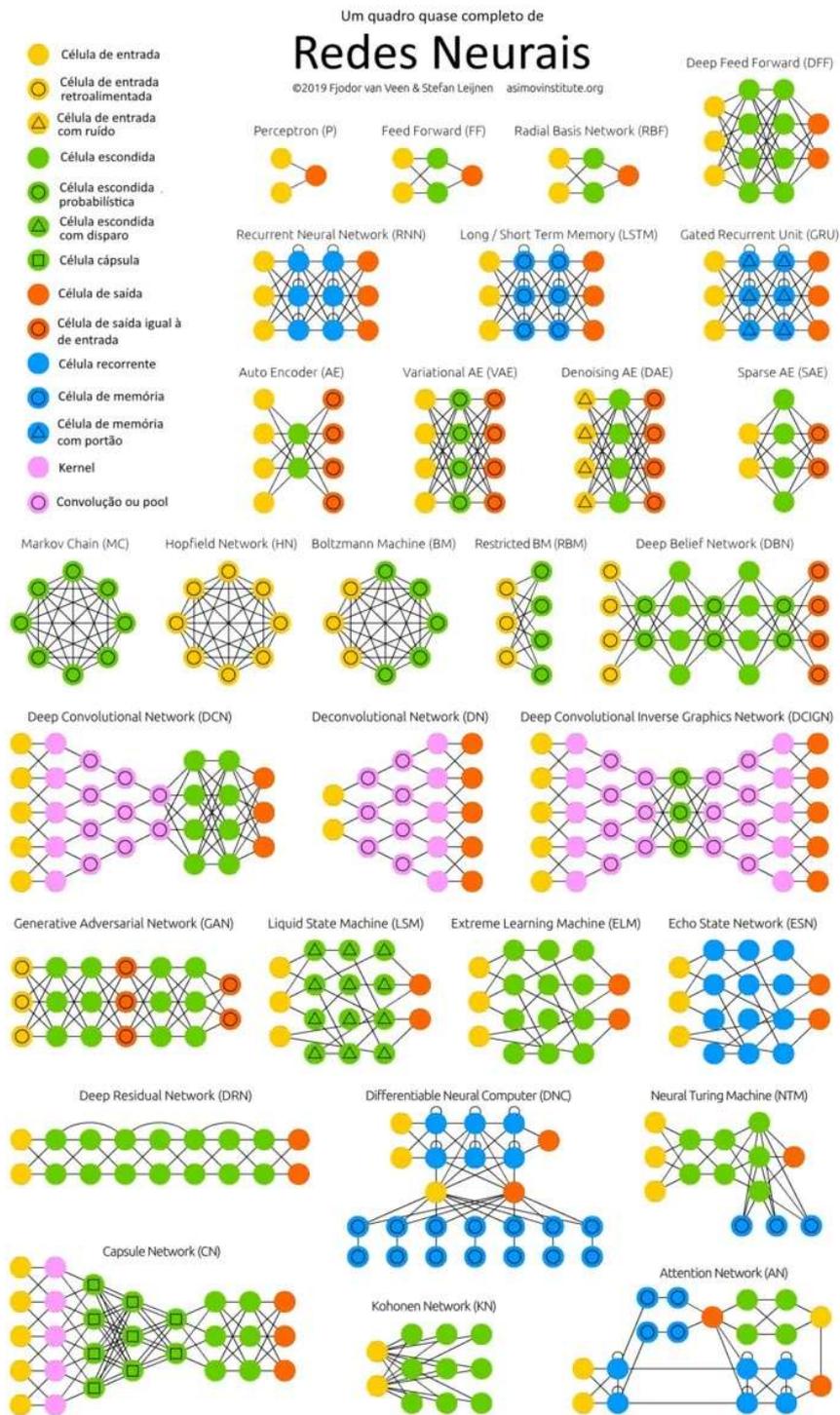


Figura 5: Diagramas exemplos de arquiteturas de Redes Neurais – Imagem originalmente disponível em: <https://www.asimovinstitute.org/neural-network-zoo/>

### 3.4. FLUXOGRAMA DA REDE NEURAL CONVOLUCIONAL UTILIZADA

Não existe um único tipo de fluxograma para redes neurais, no entanto, algumas etapas são obrigatórias para que a rede seja funcional. Ao início, deve-se começar com a preparação

dos dados (amostras), incluindo a limpeza, normalização e transformação. Em seguida, deve ser feita a divisão dos dados em conjuntos de treinamento, validação e teste. Para posteriormente, transitar os dados pelo modelo de rede neural escolhido, definindo-se ainda a escolha das camadas, funções de ativação, número de neurônios, entre outros.

O pré-processamento dos dados é uma etapa importante para garantir que a rede neural funcione corretamente. Os dados de entrada devem ser formatados de uma maneira que estejam compatíveis com a arquitetura da rede neural, e garantir que todos os dados estejam na mesma escala, e/ou outros enquadramentos requeridos pela rede desenvolvida, etapa detalhadamente descrita em 4.1.1. acima.

Durante o treinamento, o desempenho do modelo é avaliado usando o conjunto de validação e os hiperparâmetros são ajustados conforme necessário. O treinamento da rede neural é o processo de ajustar os pesos dos neurônios para que a rede neural possa gerar previsões, conforme a expectativa. O treinamento é feito usando um algoritmo de aprendizado de máquina, como o *backpropagation*, que é parte principal do ajuste automático dos pesos sinápticos do processo de aprendizado.

Mais detalhadamente, falando-se em hiperparâmetros, pode-se dizer que estes são parâmetros que controlam o comportamento do algoritmo de aprendizado de máquina e são definidos antes do treinamento do modelo. Eles são diferentes dos parâmetros do modelo, que são ajustados durante o treinamento para melhorar a precisão do modelo. Os hiperparâmetros são definidos pelo usuário com base em seu conhecimento do problema e do algoritmo, e podem incluir coisas como a taxa de aprendizado, número de camadas, número de neurônios em cada camada, tamanho do *batch*, ou até mesmo o tipo de otimizador.

O teste da rede é feito para avaliar o desempenho desta em dados que não foram usados para treina-la. A rede é testada usando um conjunto de dados de teste, que contém exemplos de dados de entrada que não foram usados durante o processo de treinamento. E então o desempenho é avaliado em relação ao conjunto de dados de teste. Isto é importante para garantir que a rede não esteja apenas aprendendo a gerar previsões precisas nos dados de treinamento, mas também nos dados reais.

O fluxograma básico de funcionamento da rede neural convolucional (CNN) para classificação das amostras de imagens utilizada nesse estudo é exibido resumidamente na figura 6, seguinte:

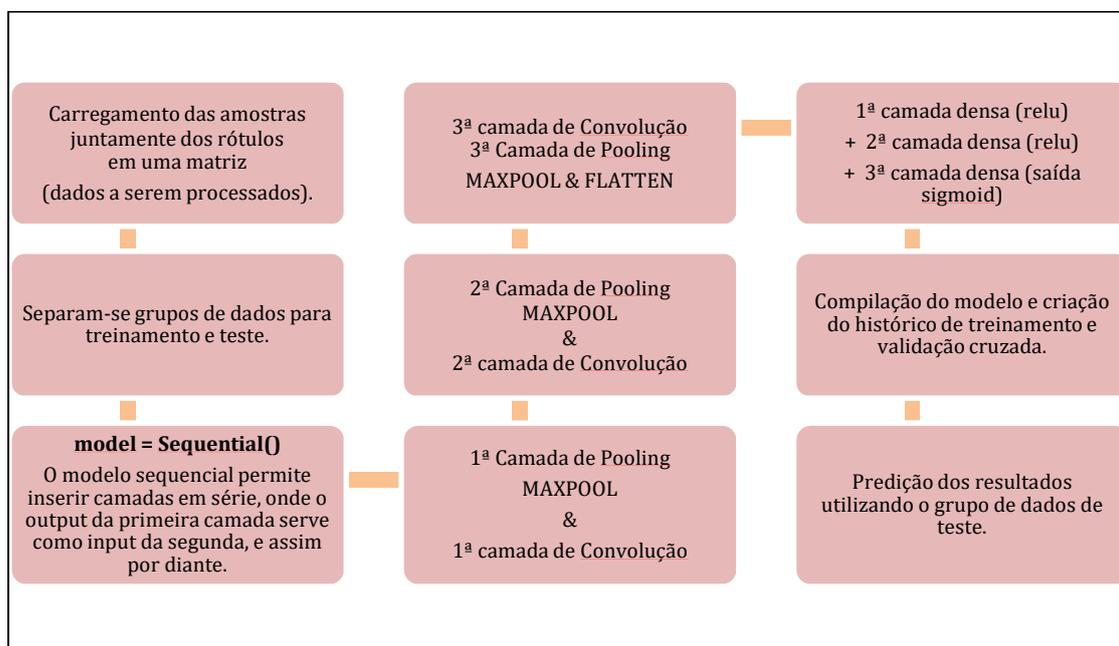


Figura 6: O fluxograma de funcionamento da rede neural objeto deste estudo

### 3.5. O PROBLEMA DO DESAPARECIMENTO DE GRADIENTE

Com o intuito de resolver o problema do desaparecimento de gradiente, foram propostas diferentes soluções na literatura. Uma das primeiras soluções propostas foi a utilização de funções de ativação não saturadas, como a *ReLU (Rectified Linear Unit)*, que evita o problema de saturação em regiões positivas e reduz o risco de desaparecimento de gradiente em regiões negativas (Nair & Hinton, 2010).

Outra abordagem comum é a utilização de inicialização adequada dos pesos sinápticos. Através da inicialização dos pesos com valores próximos a zero, é possível manter a saída dos neurônios na região linear da função de ativação, evitando o problema de saturação (Glorot & Bengio, 2010).

Além disso, a utilização de métodos de regularização, como o *dropout*, também pode contribuir para a prevenção do desaparecimento de gradiente (Srivastava et al., 2014). O *dropout* é uma técnica que desativa aleatoriamente neurônios durante o treinamento, forçando a rede a aprender características redundantes de forma independente.

Por fim, o uso de arquiteturas especiais, como as redes residuais, também se mostrou efetivo na prevenção do desaparecimento de gradiente (He et al., 2016). As redes residuais utilizam conexões de atalho para permitir que a informação flua diretamente de uma camada para outra, evitando que a informação seja filtrada ou perdida em camadas intermediárias.

De maneira resumida, pode-se dizer que o problema do desaparecimento de gradiente foi abordado de diversas formas na literatura, incluindo o uso de funções de ativação não saturadas, inicialização adequada dos pesos, métodos de regularização e arquiteturas especiais. A combinação dessas técnicas permitiu o avanço das redes neurais artificiais e o seu uso em aplicações cada vez mais complexas.

A evolução da inteligência artificial é marcada por diversos marcos históricos que influenciaram seu desenvolvimento. Na Figura 7, destaca-se alguns desses marcos, elucidando a trajetória dessa fascinante disciplina.

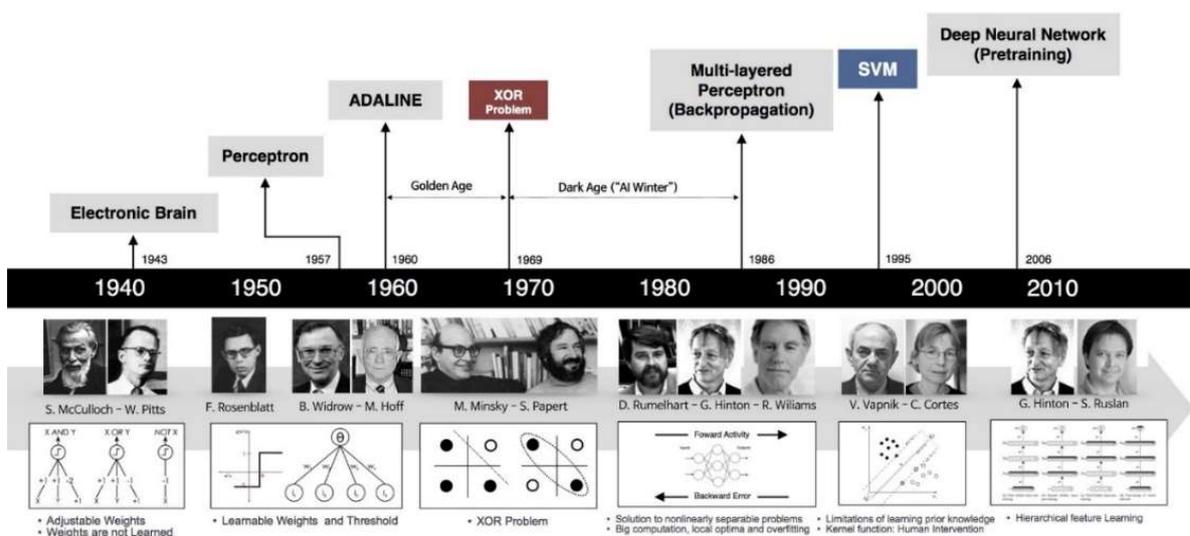


Figura 7: Marcos no desenvolvimento das redes neurais. - Figura originalmente disponível em: <http://deeplearningbook.com.br/uma-breve-historia-das-redes-neurais-artificiais/>

### 3.6. ENTENDENDO AS FUNÇÕES UTILIZADAS NA REDE NEURAL

#### 3.6.1. DIVISÃO TREINO E TESTE SIMPLES (HOLD-OUT)

De maneira a garantir que uma rede neural seja capaz de generalizar bem para novos dados e evitar o *overfitting*, os dados são divididos em três conjuntos: Treinamento, Validação e Teste durante o aprendizado, ou treinamento propriamente dito.

O conjunto treinamento é usado para ajustar os parâmetros do modelo, enquanto o conjunto validação é usado para avaliar o desempenho do modelo durante o treinamento e ajustar os hiperparâmetros: como número de camadas, o tamanho do batch, a taxa de aprendizagem, entre outros. O conjunto teste é usado para avaliar o desempenho final do modelo após o treinamento ser concluído.

Ao separar os dados em conjuntos de treinamento, validação e teste, é possível avaliar a capacidade do modelo de generalizar para novos dados e evitar o overfitting, que ocorre quando o modelo é ajustado demais para os dados de treinamento e perde a capacidade de generalizar para novos dados.

Nesta abordagem, o conjunto original de dados é dividido em treino e teste. Então, ajusta-se o modelo ao conjunto de treino e usando o conjunto de teste mensura-se quão bom é o desempenho do modelo em dados nunca vistos.

Em geral, separa-se 80% do conjunto original para treino e os 20% restantes para teste. Note que, após o modelo ser ajustado, este é usado para prever cada instância do conjunto de teste e sumarizar os erros dessas previsões em um único valor, por exemplo, a média dos erros. Melhor exemplificado graficamente na figura 8:

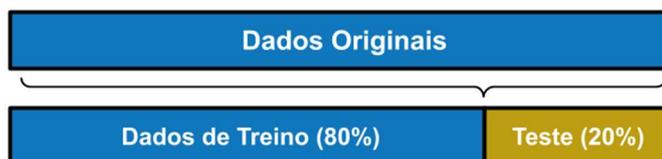
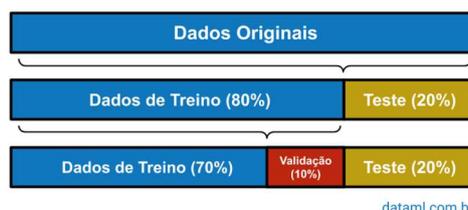


Figura 8: Proporcionalidade da divisão dos dados para treino e teste para uma RNA hipotética.

Muitas vezes é preciso otimizar os hiperparâmetros de um modelo, ou ainda comparar o desempenho de diferentes modelos. Nesses casos separa-se uma parte dos dados para validação, a qual não é usada para treinar, mas sim para calcular os erros nesse conjunto, e modificar os parâmetros de forma a diminuir tais erros. Como exemplificado na figura 9:



[dataml.com.br](http://dataml.com.br)

Figura 9: Proporcionalidade da divisão dos dados para treino, teste e validação para uma RNA hipotética. Créditos na imagem.

No caso da linguagem Python, a qual foi adotada para este estudo, a função `train_test_split()` do `sklearn` é utilizada para dividir o conjunto de dados em dois subconjuntos distintos: um conjunto de treinamento e um conjunto de teste.

A `train_test_split()` do `sklearn`, conforme demonstrado na figura 10, abaixo, apresenta a seguinte sintaxe:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

Figura 10: Sintaxe da função `train_test_split()` do `sklearn`

A função `train_test_split()` recebe quatro parâmetros:

- `X`: um *array* com as características do conjunto de dados
- `y`: um *array* com as variáveis de destino do conjunto de dados
- `test_size`: o tamanho do conjunto de teste, geralmente um valor entre 0 e 1 que representa a proporção do conjunto de teste em relação ao conjunto de dados original. Por exemplo, se o `test_size` for 0.2, 20% dos dados serão usados para testar o modelo.
- `random_state`: é uma semente para o gerador de números aleatórios que controla a divisão dos dados em conjuntos de treinamento e teste. Isso garante que a divisão seja sempre a mesma e, portanto, os resultados do modelo possam ser reproduzidos.

A função retorna quatro objetos:

- `X_train`: o subconjunto de características do conjunto de treinamento
- `X_test`: o subconjunto de características do conjunto de teste
- `y_train`: o subconjunto de variáveis de destino do conjunto de treinamento
- `y_test`: o subconjunto de variáveis de destino do conjunto de teste

### 3.6.2. CAMADAS DE CONVOLUÇÃO

As camadas de convolução são o componente chave das redes neurais convolucionais (CNNs), que permitem o reconhecimento de imagem, segmentação e detecção de objetos. As camadas de convolução funcionam aplicando um conjunto de filtros de convolução a uma entrada de imagem para produzir um conjunto de mapas de características que resumem as informações relevantes presentes na imagem.

As camadas de convolução são compostas por um conjunto de filtros de convolução, que são matrizes de números que deslizam pela imagem de entrada, multiplicando e somando os valores da imagem que se sobrepõem aos elementos do filtro. Isso produz um valor único em cada posição do mapa de características resultante, que representa a resposta do filtro àquela região da imagem. Os filtros podem ser pensados como detectores de recursos que procuram padrões específicos na imagem, como bordas, texturas, padrões ou formas.

As camadas de convolução geralmente incluem algumas etapas adicionais para melhorar o desempenho da rede, como a aplicação de uma função de ativação não-linear (no caso deste estudo usou-se a função *ReLU*) aos valores de saída dos filtros, que adiciona não-linearidade à rede, e/ou ainda a aplicação de *pooling* para reduzir o tamanho dos mapas de características e tornar a rede mais eficiente em termos computacionais.

As camadas de convolução também podem ser empilhadas em uma arquitetura de rede mais profunda, onde as saídas de uma camada de convolução são usadas como entradas para a próxima camada, permitindo que a rede aprenda recursos cada vez mais complexos. Resumidamente, pode-se dizer que as camadas de convolução são um componente fundamental para as CNNs, permitindo que a rede extraia recursos relevantes de imagens para tarefas de visão computacional.

### **3.6.3. O QUE SÃO OS NEURÔNIOS DA REDE NEURAL**

Os neurônios de uma rede neural artificial são unidades de processamento que são responsáveis por aprender padrões nos dados de entrada, eles são os componentes básicos das redes neurais artificiais e são inspirados no funcionamento dos neurônios biológicos do cérebro humano. Eles são responsáveis por receber, processar e transmitir informações através da rede neural (Rosenblatt, 1958).

Cada neurônio é conectado a vários outros neurônios através de conexões sinápticas, que são representadas pelos pesos sinápticos. Esses pesos sinápticos são ajustados durante o processo de treinamento da rede neural, utilizando algoritmos de aprendizado, como o *backpropagation* (Rumelhart et al., 1986).

A capacidade de aprendizado da rede neural é determinada pelo número de neurônios e conexões sinápticas, bem como pelos algoritmos de aprendizado utilizados. Durante o

processo de treinamento, os pesos sinápticos são ajustados gradualmente para que a rede neural possa melhorar seu desempenho na tarefa para a qual foi projetada (Goodfellow et al., 2016).

A evolução do aprendizado em uma rede neural pode ser dividida em duas fases principais: a fase de treinamento e a fase de teste. Durante a fase de treinamento, a rede neural é exposta a um conjunto de dados de treinamento para aprender a relação entre as entradas e as saídas desejadas. Já durante a fase de teste, a rede neural é avaliada em um conjunto de dados que não foram usados no treinamento, para verificar a sua capacidade de generalização (Bishop, 2006).

Resumidamente, pode-se dizer que os neurônios são os componentes básicos das redes neurais artificiais e são responsáveis por processar e transmitir informações através da rede. Os pesos sinápticos são ajustados durante o processo de treinamento da rede neural, utilizando algoritmos de aprendizado, como o *backpropagation*.

#### **3.6.4. CAMADA DE POOLING ATRAVÉS DA FUNÇÃO MAXPOOL 2D**

A camada de *pooling*, através da função *Maxpooling2D* é uma técnica comumente usada neste tipo de rede, leitura de imagens, como é o caso deste estudo, e ela tem a finalidade de reduzir a dimensionalidade dos recursos extraídos pelas camadas convolucionais, trata-se de um processo de discretização baseado em amostra (LeCun et al., 1998). Ou seja, a camada de *pooling*, funciona dividindo a entrada em regiões não sobrepostas e, em seguida, é selecionado o valor máximo em cada região como saída (Scherer et al., 2010). A finalidade é diminuir a resolução espacial da imagem, enquanto mantém as características importantes, de forma a reduzir o número de parâmetros na rede e a aumentar a capacidade de generalização, permitindo que a rede capture características mais gerais das imagens de entrada (Scherer et al., 2010).

Como exemplo de como o *Maxpooling 2D* trabalha, pode-se supor que existe uma matriz 4x4, que representa nossa entrada inicial, a imagem; e um filtro 2x2, que é a própria janela de *maxpooling* de dimensões 2x2; e não sobreporá regiões. A imagem de entrada seria dividida em quatro regiões não sobrepostas de dimensões 2x2, e o valor MÁXIMO em cada região será selecionado como saída (Scherer et al., 2010).

Exemplificando através da figura 11, para cada uma das regiões representadas pelo filtro, pegaremos o MÁXIMO dessa região e criaremos uma nova matriz de saída onde cada elemento é o máximo de uma região na entrada original:

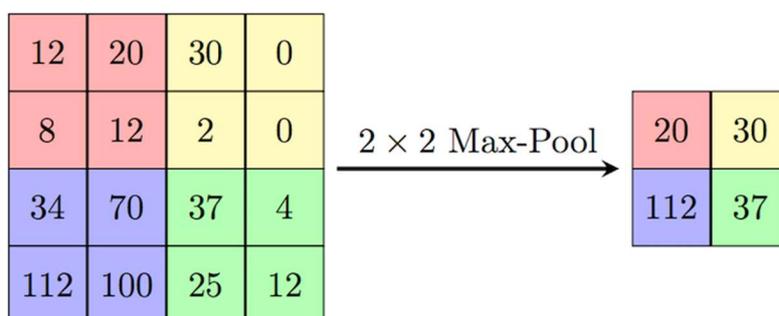


Figura 11: Diagrama hipotético - Exemplo de aplicação.

Adicionalmente, cabe ressaltar, que a maioria das discontinuidades de soldagem são identificadas pela ausência de material, gerando áreas escuras na radiografia (exceção somente para a inclusão de tungstênio, que gera pontos claros devido à alta densidade deste material), devido a este fato, foi notada a necessidade de inverter os pixels das amostras obtidas, pois, durante o processo de aplicação da função *Maxpool2d*, havia uma perda de informação sobre a localização precisa das características, o que tornava difícil identificar pontos escuros em uma imagem. Como solução trivial, adotou-se inverter os pixels da imagem antes de aplicar a função *Maxpool2d*, para que os pontos escuros se tornem pontos claros e possam ser mais facilmente identificados. Essa técnica é amplamente utilizada em várias aplicações de processamento de imagem que envolvem a detecção de características específicas, como bordas ou pontos de interesse.

### 3.6.5. A FUNÇÃO FLATTEN

A função *Flatten* é usada em conjunto com a camada *pooling* em redes neurais convolucionais para transformar os dados de saída da camada de *pooling* em um vetor unidimensional. Como se efetuasse um empilhamento das colunas da matriz que representa a imagem.

Por exemplo, no caso deste estudo utilizamos como entrada amostras de imagens de tamanho 100 x 200 (h x l) pixels, e a camada *pooling* reduzindo a dimensionalidade, mantendo apenas as informações mais relevantes. No entanto, para passar essas informações reduzidas para a próxima camada densa (totalmente conectada), precisamos transformar os dados em um vetor unidimensional.

É aí que a função *Flatten* entra. Ela é usada para achatar (ou "aplanar") o tensor de saída da camada de *pooling* em um vetor unidimensional que pode ser passado para a próxima camada densa. Essa camada densa é usada para fazer a classificação ou previsão final, com base nas informações extraídas pelos mapas de características convolucionais e reduzidas pelas camadas de *pooling*.

De maneira resumida, pode-se dizer que a função *Flatten* é usada para converter os dados em um formato adequado para serem processados pela próxima camada densa, após a redução de dimensionalidade feita pela camada de *pooling*.

### 3.6.6. CAMADA DENSA E OVERFITTING

A camada densa, também conhecida como camada totalmente conectada, é um tipo de camada em rede neural artificial que conecta todos os neurônios de uma camada a todos os neurônios da camada seguinte. Essa conectividade total permite que as camadas densas aprendam relações complexas entre os recursos de entrada e saída da rede.

Na camada densa, cada neurônio recebe como entrada todas as saídas da camada anterior e calcula um valor de saída usando a função de ativação escolhida. Cada conexão entre neurônios é representada pelo peso, o qual determina a força da conexão entre eles. Durante o treinamento, através da técnica de retropropagação do erro, *backpropagation* como explicado anteriormente, a rede ajusta os pesos para otimizar o desempenho do resultado alcançado durante a testagem na saída da rede.

Objetivamente, pode-se classificar as camadas densas como um tipo de camada em redes neurais artificiais que conectam todos os neurônios de uma camada a todos os neurônios da camada seguinte. Elas são úteis para aprender relações complexas entre os recursos de entrada e saída da rede e são comumente usadas para produzir uma saída de tamanho fixo.

As camadas densas podem ser usadas em redes mais profundas, como uma forma de combinar os recursos aprendidos em camadas anteriores para produzir uma representação final da entrada. No entanto, uma das desvantagens das camadas densas é que elas podem exigir um elevado número de parâmetros, o que pode levar a problemas de *overfitting*, especialmente em conjuntos de dados pequenos.

De forma a explicar generalista, entende-se como *Overfitting* como problema comum em aprendizado de máquina no qual um modelo é treinado em um conjunto de dados específico e se ajusta demasiadamente bem a ele, tornando-se incapaz de generalizar para novos dados não vistos durante o treinamento. Diz-se que um modelo que sofre de *overfitting* é aquele que se torna muito específico para os dados de treinamento e não consegue generalizar para novos dados que apresentam diferentes padrões ou variações em relação aos dados de treinamento. Isso pode levar a uma queda na precisão e desempenho do modelo ao ser usado em novos dados.

O *overfitting* geralmente ocorre quando um modelo é muito complexo em relação ao tamanho do conjunto de dados de treinamento, o que faz com que ele se adapte demais aos dados de treinamento e não consiga generalizar adequadamente para novos dados. Outras causas comuns de *overfitting* incluem o uso de muitas características (ou recursos) em relação ao tamanho do conjunto de dados; treinamento excessivo, que ocorre quando o modelo é treinado por um número excessivo de épocas.

Para evitar o *overfitting*, é comum usar técnicas de regularização, como a adição de penalidades nos pesos do modelo, a diminuição do tamanho do modelo, o aumento do conjunto de dados de treinamento, ou o uso de conjuntos de validação e técnicas de validação cruzada para monitorar o desempenho do modelo em dados não vistos. No entanto, para o modelo deste estudo, foi possível contornar o *overfitting* apresentado aplicando validação cruzada, e otimização do número de épocas para treinamento; pois inicialmente, foi verificado que a rede que havia sido iniciada com duzentas épocas de treinamento, após análise dos gráficos de saída do treinamento, notou-se estava entrando em *overfitting*, e assim foram reduzidas gradativamente até chegar-se ao número mais acertado de cem épocas. O que fez que o modelo generalizasse otimamente.

### **3.6.7. VALIDAÇÃO CRUZADA**

A validação cruzada (*cross-validation* em inglês) é a técnica utilizada para avaliar o desempenho dos modelos de redes neurais artificiais, e é esta técnica que foi utilizada para avaliar o modelo deste estudo. Ela é utilizada para estimar a capacidade de generalização do modelo, ou seja, a habilidade de desempenhar bem em novos dados que não foram utilizados para treinar o modelo.

A validação cruzada é particularmente útil para redes neurais, que podem ser muito sensíveis à inicialização dos pesos e à escolha do conjunto de treinamento. Ao usar a validação cruzada, é possível obter uma estimativa mais robusta do desempenho do modelo, reduzindo a dependência da escolha de um único conjunto de treinamento.

Uma das formas mais comuns de validação cruzada é a *k-fold cross-validation*. Nessa técnica, os dados são divididos em *k folds*, onde cada *fold* é utilizado uma vez como conjunto de validação e os outros *k-1 folds* são utilizados como conjunto de treinamento. Ao final, os resultados são combinados para obter uma estimativa do desempenho do modelo.

Trata-se de uma divisão realizada no conjunto de dados em *k* partes iguais, geralmente chamadas de "*folds*". Em seguida, o modelo é treinado *k* vezes, onde em cada iteração um dos *k folds* é utilizado como conjunto de validação e os outros *k-1 folds* são usados como conjunto de treinamento. Ao final das *k* iterações, são obtidos *k* valores de desempenho, que podem ser usados para estimar a média e a variância do desempenho do modelo.

Outra técnica comum é a *Leave-One-Out Cross-Validation*, que é uma variação da *k-fold cross-validation*, onde *k* é igual ao número total de amostras no conjunto de dados. Nessa técnica, o modelo é treinado *k* vezes, onde em cada iteração uma única amostra é utilizada como conjunto de validação e as outras *k-1* amostras são utilizadas como conjunto de treinamento.

Ou seja, após ter um modelo ajustado aos dados, é necessário uma maneira de aferir esse modelo ao prever em dados nunca vistos, então é preciso uma maneira de quantificar sua **capacidade de generalização**, e para isso faz-se o uso da Validação Cruzada.

Em resumo, a validação cruzada é a técnica para avaliar o desempenho de modelos de redes neurais artificiais, que permite estimar a capacidade de generalização do modelo, reduzindo a dependência da escolha de um único conjunto de treinamento.

### 3.6.8. TREINANDO A REDE: AS FUNÇÕES FIT E SEQUENTIAL

Para que os dados transcorram pela rede, passando de uma camada de neurônios para a outra, é necessário uma função de auxílio, para isso faz-se o uso da função *Sequential()* do Keras, que pertence a uma classe de funções que permite criar modelos de redes neurais sequenciais de maneira simples. Usando-a, é possível criar redes neurais sem que seja necessário construir as conexões manualmente, uma vez que a ordem das camadas é definida

de forma sequencial. A função *Sequential()* é uma das principais ferramentas para construir redes neurais utilizando o Keras, já que permite adicionar diversas camadas de maneira rápida.

No entanto, além de transitar os dados entre as camadas, é necessário que numa nova época de treinamento transcorrendo os dados pela rede, os pesos estejam corrigidos, isso é possível com a função *fit()*, que é uma das mais importantes do *Keras*. Nela é treinada a rede neural com os dados disponibilizados para treinamento. A função *fit()* tem como objetivo o ajuste dos pesos da rede para minimizar a função de perda (*loss* – a qual será explicada mais adiante) e melhorar o desempenho do modelo.

Durante o treinamento, a função *fit()* atualiza os parâmetros do modelo usando um algoritmo de otimização (no caso deste estudo, foi utilizado o *ADAM*) para minimizar a perda (*loss*) em relação aos dados de treinamento. Em cada época de treinamento, a função *fit()* divide os dados de treinamento em lotes (*batch*) de tamanho *batch\_size* e ajusta os parâmetros do modelo com base nas amostras em cada lote.

### **3.6.9. ENTENDENDO A MÉTRICA DE APRENDIZADO ATRAVÉS DAS FUNÇÕES DE PERDA LOSS E ACCURACY E A MELHORIA DOS RESULTADOS ATRAVÉS DO ALGORITMO OTIMIZADOR ADAM**

Para calcular a *Accuracy* de uma rede, é necessário separar um conjunto de dados em um conjunto de treinamento e um conjunto de teste. O conjunto de treinamento é usado para treinar a rede e o conjunto de teste é usado para avaliar o desempenho da rede após o treinamento.

A *Accuracy* é uma métrica de desempenho simples e intuitiva que mede a proporção de exemplos classificados corretamente pela rede neural em relação ao número total de exemplos. Ela é amplamente usada em tarefas de classificação, como reconhecimento de imagens, onde o objetivo é classificar imagens em diferentes categorias.

A métrica *Accuracy* é usada para avaliar o desempenho de uma rede neural em tarefas de classificação, como a classificação de imagens em redes neurais. Ela é definida como a proporção de exemplos classificados corretamente em relação ao total de exemplos, em outras palavras, a *Accuracy* mede a taxa de acertos da rede neural em prever a classe correta para uma determinada imagem. Por exemplo, se a CNN classificar corretamente 90 imagens de um total de 100, a *Accuracy* será de 90%.

Depois que a CNN é treinada, ela é avaliada no conjunto de teste. A *Accuracy* é calculada dividindo-se o número de exemplos classificados corretamente pelo total de exemplos no conjunto de teste. Por exemplo, se a CNN classificar corretamente 900 imagens de um total de 1000, a *Accuracy* será de 90%.

A métrica *Accuracy* é muito útil em muitas tarefas de classificação, mas é importante lembrar que ela pode ser enganosa em casos em que o conjunto de dados é desbalanceado, ou seja, quando uma classe tem muito mais exemplos do que as outras. Nesses casos, a *Accuracy* pode ser alta simplesmente porque a CNN está acertando a classificação da classe majoritária, enquanto erra a classificação das classes minoritárias. Nesses casos, outras métricas, como a *Precisão*, *Revocação* ou *F1-score*, podem ser mais apropriadas para avaliar o desempenho da CNN.

Durante o treinamento são impressos na tela quatro resultados diferentes: *loss*, *accuracy*, *val\_loss* e *val\_accuracy*, respectivamente: Trata-se da perda e a acurácia ao final de cada época de treinamento, tanto para os dados de treinamento quanto para o teste de validação cruzada.

A função de perda (*loss function*) é um componente fundamental de qualquer algoritmo de aprendizado supervisionado, incluindo as redes neurais. Ela é usada para medir a diferença entre as saídas previstas pelo modelo e as saídas reais (*labels* – estes fornecidos como dados para treinamento) correspondentes ao vetor de rótulos que identificam as características de cada posição do vetor de treinamento.

O objetivo do treinamento de uma rede neural é ajustar seus parâmetros para minimizar a perda (*loss*) em relação aos dados de treinamento. Isso é feito usando um algoritmo de otimização, que atualiza os parâmetros do modelo de acordo com a derivada da perda em relação a cada parâmetro, no caso da rede deste estudo, trata-se do otimizador *ADAM*.

A escolha da função de perda depende da tarefa de aprendizado específica que se deseja realizar. Para problemas de classificação binária, como no caso deste estudo, a função de perda mais comum é a *binary crossentropy*, a qual foi utilizada; enquanto para problemas de classificação multiclasse, a *categorical cross-entropy* é frequentemente usada. Para problemas de regressão, a *mean squared error* é uma escolha comum.

A função de perda deve ser diferenciável em relação às saídas da rede neural, para permitir que o algoritmo de otimização calcule a derivada da perda em relação aos parâmetros da rede. Além disso, a função de perda deve ser escolhida com cuidado para evitar problemas como o *overfitting* e garantir que o modelo seja capaz de generalizar bem para dados novos e não vistos. Resumidamente, pode-se dizer que a função de perda é um componente crítico do treinamento de uma rede neural e deve ser cuidadosamente escolhida para cada tarefa de aprendizado específica.

Ainda durante o treinamento, e verificado a cada época de treinamento, o otimizador *ADAM* (*Adaptive Moment Estimation*) é um método de otimização popular para redes neurais, que combina as ideias do otimizador *SGD* (*Stochastic Gradient Descent*) com o *RMSProp* (*Root Mean Square Propagation*).

O *ADAM* é um otimizador adaptativo porque ajusta os parâmetros de aprendizagem (*learning rate*) para cada parâmetro de peso individual com base em estimativas de momento de primeira ordem (média das derivadas) e momento de segunda ordem (média das derivadas ao quadrado). Essas estimativas são calculadas a partir dos gradientes calculados em cada iteração do treinamento.

A ideia principal ao utilizar o *ADAM* é ajustar os parâmetros de aprendizagem de forma adaptativa para cada parâmetro individual, em vez de usar um valor global para toda a rede. Isso ajuda a evitar problemas como o *overfitting* e a falta de convergência que podem ocorrer com o uso de um valor fixo para o *learning rate*. Este é, geralmente, considerado como um otimizador eficaz e uma boa escolha para redes neurais, no entanto, para problemas diferentes, devem ser considerados outros otimizadores.

Depois que o treinamento for concluído, você pode avaliar o desempenho do modelo em um conjunto de dados de teste separado usando a função *evaluate()*. A função *evaluate()* calcula as métricas de desempenho do modelo em um conjunto de dados de teste e retorna os valores.

### **3.6.10. AVALIANDO OS RESULTADOS ENCONTRADOS, A FUNÇÃO EVALUATE**

A função *evaluate* foi usada para avaliar o desempenho da rede neural treinada com o conjunto de dados de teste. Fundamentalmente, ela retorna as métricas de desempenho definidas ao compilar o modelo, como a *Accuracy*, *loss* e/ou outras métricas adicionais.

A sintaxe básica da função *evaluate* no *Keras*, *Model.evaluate(x\_test, y\_test)*, onde *x\_test* e *y\_test* são os dados de entrada e saída de teste, respectivamente, tem por finalidade retornar uma lista de valores de métricas calculados na base de teste, na ordem em que foram especificadas ao compilar o modelo. Exemplificando, quando tem-se a compilação do modelo com as métricas *Accuracy* e *loss*, a chamada da função *evaluate* retornaria uma lista com dois valores, o valor da *Accuracy* e o valor de *loss*.

No caso desse estudo, para avaliar o desempenho de um modelo de classificação de imagens, pode-se chamar a função *evaluate*, e passar o conjunto de dados de teste e a métrica *Accuracy*, com *loss*, *accuracy = Model.evaluate(x\_test, y\_test)* em que *x\_test* são as imagens de teste e *y\_test* são as classes correspondentes. A função *evaluate* calcula a *Accuracy* e *loss* para o conjunto de teste, e as retorna como valores de saída.

## 4. METODOLOGIA DA PESQUISA

Nesta dissertação, que descreve o experimento executado, foi investigada a eficácia de uma rede neural, do tipo convolucional, de detectar descontinuidades de solda em radiografias de chapas metálicas. O estudo concentrou-se em verificar se a rede neural é capaz de identificar com precisão e consistência indicações na soldagem, que caracterizariam descontinuidades de preenchimento da solda, crucial para a garantia da integridade estrutural na indústria de caldeiraria e engenharia mecânica. E assim, foi explorado o potencial da rede neural em processar e interpretar as sutis informações visuais presentes nas radiografias de solda reais, não didáticas.

### 4.1. O EXPERIMENTO

#### 4.1.1. OBTENÇÃO DAS AMOSTRAS DE IMAGENS

Para o experimento realizado, foram utilizadas imagens radiográficas físicas avaliadas individualmente no negatoscópio, que em seguida foram pré-processadas com o uso da linguagem Python, onde procedeu-se da seguinte maneira:

##### **Etapa 1 – Seleção de imagens com indicações:**

Consistiu em fazer uma verificação abrangente de centenas de filmes radiográficos utilizados tanto na inspeção de juntas soldadas de equipamentos, quanto no processo de qualificação de soldadores. Nessa verificação inicial foi feita uma seleção de todos os filmes

que continham indicação de defeito de soldagem, ou seja, qualquer imagem que apresentasse uma possível descontinuidade estaria selecionada para a próxima etapa.

## **Etapa 2 – Digitalização das imagens selecionadas:**

Considerou-se, inicialmente, a digitalização dos filmes por um scanner específico para essa finalidade, no entanto, o custo apresentou-se inviável para a execução do experimento. E partiu-se para a busca de uma solução alternativa, de menor aporte financeiro;

Assim foi tentado efetuar a digitalização das radiografias com a utilização de um scanner de bancada convencional, porém este método não foi satisfatório, considerando que os digitalizadores de imagens convencionais são desenvolvidos para digitalizar figuras de películas opacas, como papéis. Uma vez que a luz e o dispositivo que faz a digitalização da imagem incidem pelo mesmo lado do alvo, não possibilitando revelar o conteúdo gravado em filmes radiográficos, que depende de incidência de luz pelo lado posterior ao de visualização.

Descartada essa tentativa, partiu-se para outra possibilidade, que inicialmente acreditava-se não ser praticável devido a uma possível perda de qualidade da imagem, e que consistia em fotografar o filme radiográfico diretamente do negatoscópio. Pois bem, essa possibilidade, posteriormente, mostrou-se satisfatoriamente eficaz e de baixo custo. E assim, foram capturadas sessenta e uma imagens de filmes radiográficos, dispostas no negatoscópio, com uso da câmera fotográfica do celular Redmi Note 9S.

A radiografia da Figura 12 pode ser considerado o ponto de partida crucial para início do experimento para investigar a capacidade da rede neural em identificar descontinuidades.

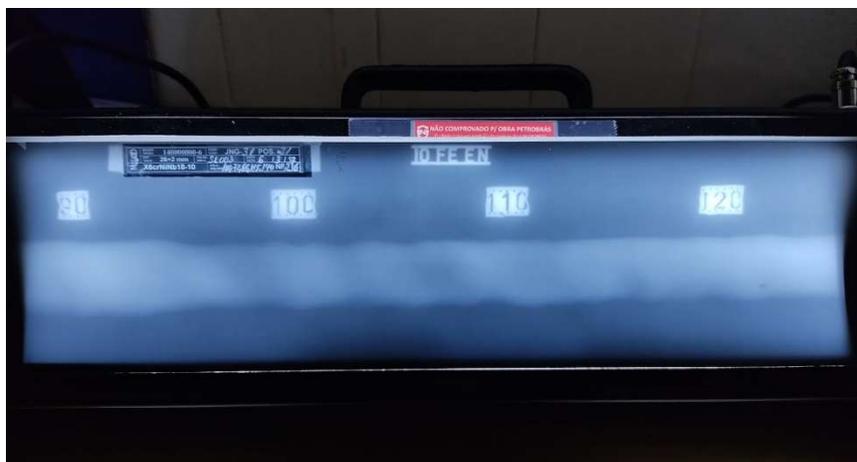


Figura 12: Fotografia do negatoscópio revelando a imagem gravada no filme radiográfico

### Etapa 3 – Pré-processamento das imagens:

Após a digitalização, ainda no aplicativo de fotografia, foram recortadas as bordas, criando uma nova imagem, que continha somente a área de interesse a ser examinada, conforme exibido na figura 13, abaixo. Nesse passo também foi usado o filtro de imagem para transformar a imagem em tons de cinza, porém, posteriormente ao avaliar as propriedades das imagens, foi notado que o aplicativo fez somente uma equalização dos canais RGB, fato que não agregou benefícios ao experimento e foi preciso adicionar um passo ao experimento, o qual foi o próximo passo.

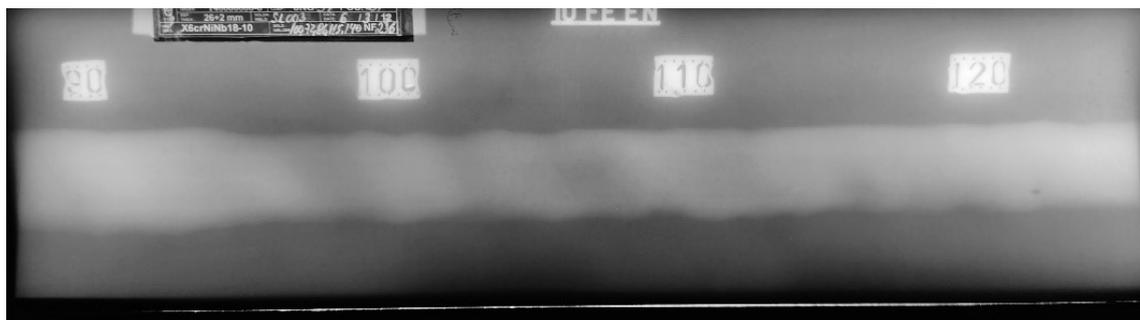


Figura 13: Imagem pré-processada, na qual foram descartadas as margens fora da área de interesse.

### Etapa 4 – Criação de amostras de imagem para análise:

Nesta etapa do experimento, já com o uso da linguagem Python, especificamente da biblioteca PIL, foi desenvolvida uma rotina de programação que percorria todos os arquivos de imagem da pasta onde foram dispostas as imagens pré-processadas no passo 3. Essa rotina, a qual foi batizada de *gerador de amostras*, entregou as amostras, conforme figura 14, retiradas das imagens, obedecendo as seguintes tarefas:

1. Abrir um arquivo de imagem;
2. Determinar as dimensões da imagem, em pixels, largura (`larg_X`) e altura (`alt_Y`);
3. Carregar em uma variável (`px_inicial_X`) o ponto inicial para coleta de amostras daquela imagem, no eixo X, a partir de 0,5% do pixel mais à esquerda da imagem. Isto foi determinado de maneira arbitrária, para eliminar ruídos impostos nas margens do filme radiográfico;
4. Carregar em uma variável (`px_inicial_Y`) o ponto inicial para coleta de imagens, no eixo Y, a partir de 30% do pixel mais inferior da imagem. Isto foi determinado de maneira arbitrária, para eliminar análise no metal de base muito além da Zona Termicamente Afetada (ZTA);

5. Carregar em uma variável a altura pré-definida de amostra de imagem. Nesse caso, quatrocentos pixels ( $px\_final\_Y = px\_inicial\_Y + alt$ ).
6. Determinar a quantidade de amostras possíveis de serem retiradas da imagem em função da largura de amostra definida.
7. Com o auxílio de um contador, enquanto este for menor que o número possível de amostras, a rotina vai seccionando a imagem em fatias menores de 400x200 pixels;  
Em seguida:
  8. Converte-se as imagens para o formato de 16bits de cores;
  9. Compacta-se as amostras com as dimensões 200x100 (desta maneira, mostrou-se uma menor perda de informação da amostra durante a compactação da amostra);
10. Inverte-se os pixels de modo a ressaltar a maioria das descontinuidades. Gerando clareamento dos pontos com ausência de material, que geram áreas escuras nas radiografias.  
– Assunto melhor detalhado ao abordar a função *Maxpool2d*.
11. Salva-se cada amostra individualmente, atribuindo um número sequencial no final do nome do arquivo, convertendo para o formato *TIFF*.
12. Quando o critério de parada do loop é alcançado, a rotina avança para a próxima imagem da pasta, retornando a tarefa 1 acima, assim preparando todas as amostras em pasta pré-definida;

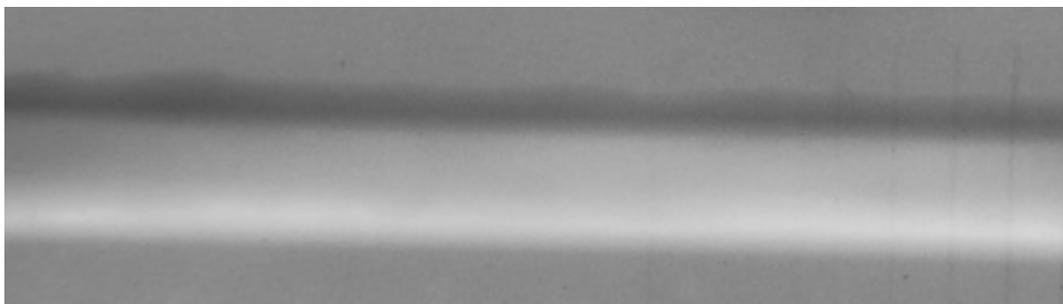


Figura 14: Amostras de imagem obtidas a partir de uma única radiografia. Pode-se notar que a sequência de amostras exibem continuidade de uma para a outra

### **Etapa 5 – Selecionando as amostras:**

Considerando que cada filme radiográfico gerou até vinte e uma amostras ao passar pelo *gerador de amostras* da etapa 4, após o processamento de todas as imagens disponíveis, foram produzidas aproximadamente novecentas amostras, que foram verificadas individualmente, de forma a classificá-las, e isso foi feito com a inclusão de um dígito no nome do arquivo: primeiro dígito do nome “a” significa que é uma amostra *sem indicação*

de defeito, e uma imagem com o primeiro dígito do nome do arquivo “d” sinaliza uma amostra *com indicação de descontinuidade*, melhor exemplificado na tabela 1, abaixo:

Tabela 1: Exemplificação da nomenclatura utilizada nos arquivos de amostras.

	Exemplo 1	Exemplo 2
Nome do arquivo de imagem original	IMG_20210518_200020.jpg	IMG_20210518_164549.jpg
Número de amostras geradas	21	20
Nome de um dos arquivos produzidos no gerador de amostras	IMG_20210518_200020 - 15 de 21.tif	IMG_20210518_164549 - 13 de 20.tif
Avaliação individual	<i>SEM indicação de defeito</i>	<i>COM indicação de defeito</i>
Nome dado ao arquivo após avaliação individual	<i>a</i> IMG_20210518_200020 - 15 de 21.tif	<i>d</i> IMG_20210518_164549 - 13 de 20.tif

As amostras que não receberam como primeiro dígito do nome as letras “a” ou “d”, foram imagens que apresentavam algum tipo de informação que poderia trazer perturbação ou ruído ao algoritmo.

Ao final da seleção das amostras, foram obtidas cento e vinte e oito, 128, amostras ‘SEM indicação de descontinuidades’, e cento e trinta e oito, 138, amostras ‘COM indicação de descontinuidades’.

#### 4.1.2. SUBMETENDO AS AMOSTRAS À REDE NEURAL:

Posteriormente a avaliação e categorização individual das amostras, pode-se considerar que o banco de imagens para treinamento e teste da rede neural está consolidado. E assim pode-se submeter as amostras para avaliação na rede proposta, que resumidamente, pode-se dizer que foi orientada, seguindo os mesmos 9 (nove) passos constantes do fluxograma acima, na seguinte ordem:

**Passo 1: 1a.** Admitir as amostras individualmente e carregá-las como dados em um vetor, e que, a cada loop, adiciona esse vetor como linha de uma matriz, chamada de *matriz0*;

**1b.** Verificar a primeira letra do título do arquivo da amostra e preencher o vetor rótulo com a respectiva informação, '0' para as amostras consideradas sem indicações, e '1' para as amostras consideradas com indicações, e assim fazendo uma correspondência entre o rótulo adicionado ao vetor e a vetor da imagem admitida na *matriz0*;

**1c.** Faz-se os devidos redimensionamentos da *matriz0* e do vetor de rótulos, que possibilitarão o processamentos matemático da matriz e do vetor;

**Passo 2:** Através da função *train\_test\_split()* do *sklearn* realiza-se a divisão do conjunto de dados em dois pares de *amostras/rótulos* distintas: um par para treinamento, e outro para teste ao final do treinamento, a predição de resultados.

Neste ponto, pode-se dizer que os dados estão prontos para passar pela rede neural propriamente dita.

**Passo 3:** Admite-se os dados para a função *Sequential()*, que permitirá que os dados de saída de uma função sejam utilizados como dados de entrada da função seguinte;

**Passo 4: 4a.** Passa-se as amostras, em formato de vetor, pela função *MaxPool2D*, onde os pixels que carregaram valores maiores para o vetor, aqueles que remetem aos pontos claros da amostra, devido a sua intensidade luminosa, tem seu valor assumido regionalmente, dispensando-se os valores vizinhos, por consequência, diminuindo as dimensões da matriz, fato que será detalhado mais adiante neste trabalho. Isso simula uma capacidade de evidenciar a indicação na imagem.

**4b.** Em seguida, passa-se essa informação transformada pela função *Conv2D*, que um filtro de convolução é aplicado a cada região de 2x2 dos dados presentes na matriz de características da imagem, aquela obtida no passo 1.

**4c.** A etapa de convolução é repetida, utilizando-se os mesmos parâmetros adotados anteriormente;

**Passo 5:** Insere-se uma nova etapa de *pooling*, para que possa ser aumentado o destaque dos pontos de interesse na amostra;

**Passo 6:** Considerando que até este ponto trabalhou-se com matrizes, que são dados multidimensionais, seria necessário convertê-los em dados unidimensionais para que possam tramitar pelas camadas densas, que estão no próximo passo. Para isso, os dados provenientes da última camada de *pooling*, são trabalhados na função *flatten()*, mais detalhadamente explicada adiante, e na saída da função *flatten()*, um vetor unidimensional será disponibilizado.

**Passo 7:** A partir deste ponto, tem-se as camadas densas, onde realmente há atividades da rede neural, propriamente dita, com o uso dos neurônios artificiais em si. Com o uso da função *Dense()*, pode-se conectar todos os neurônios da camada anterior a todos os neurônios da camada seguinte. Especificamente nesta rede, a seguinte configuração foi adotada após tentativas e melhorias implementadas: 50 (cinquenta) neurônios na primeira camada, 10 (dez) na segunda camada e, exclusivamente, 1 (um) único neurônio na última camada, visto que o resultado almejado deva ser: “há indicação de defeito” ou “não há indicação de defeito”, ou seja, uma resposta binária.

**Passo 8:** Nesta etapa, faz-se a compilação dos dados apresentados, permitindo o trânsito destes pela rede, vinculando as camadas e os tensores de entrada e saída. É também nesta etapa que permite-se configurar os hiperparâmetros essenciais do modelo, como o otimizador, que é responsável por ajustar os pesos da rede durante o treinamento, a função de perda *loss*, que mede a discrepância entre as previsões do modelo e os valores reais, e as métricas de avaliação, que ajudam a monitorar o desempenho do modelo durante o treinamento.

**Passo 9:** É a etapa crucial no treinamento de uma rede neural, onde, efetivamente, treina-se o modelo usando o conjunto de dados de treinamento e é a última etapa no processo de construção e treinamento da rede neural. É nesta etapa que ocorre o processo de treinamento iterativo da rede neural, onde o modelo é exposto aos exemplos do conjunto de dados de treinamento várias vezes, as chamadas épocas. A cada época, o modelo ajusta os parâmetros (pesos e vieses) com base nos erros cometidos nas previsões em relação aos valores reais do conjunto de treinamento.

Com o auxílio da validação cruzada, é avaliado o desempenho do modelo com o conjunto de validação. Isso permite monitorar a capacidade do modelo de generalização, ou seja, sua capacidade de fazer previsões precisas em dados que não foram usados durante o treinamento.

## 5. RESULTADOS

### 5.1.1. INTERPRETANDO AS MÉTRICAS PRECISION, RECALL, F1-SCORE, MACRO AVG E WEIGHTED AVG

Avaliar o desempenho da rede neural proposta é essencial para verificar a eficácia na solução do problema proposto. Para isso, algumas métricas foram utilizadas, como a medida

de precisão (*precision*), revocação (*recall*) e *f1-score*. Além disso, foram utilizadas as métricas *Macro avg* e *Weighted avg* para avaliar o desempenho geral da rede. Sucintamente, é discorrido abaixo o objetivo de cada uma destas métricas, baseado na documentação oficial do *Scikit-Learn* (*scikit-learn developers*, 2021).

A precisão (*precision*) é a proporção de casos positivos corretamente identificados pelo modelo em relação ao total de casos positivos identificados. Em outras palavras, a precisão mede a capacidade do modelo em identificar corretamente as instâncias positivas.

A revocação (*recall*) é a proporção de casos positivos corretamente identificados pelo modelo em relação ao total de casos positivos existentes. Em outras palavras, a revocação mede a capacidade do modelo em encontrar todas as instâncias positivas.

O *f1-score* é uma medida de equilíbrio entre a *precision* e a *recall*, calculado pela média harmônica entre as duas métricas. O *f1-score* é uma medida útil para comparar modelos com desempenho desigual em *precision* e *recall*.

A métrica *Macro avg* é a média aritmética das medidas de *precision*, *recall* e *f1-score* para cada classe. Essa métrica é útil quando todas as classes têm a mesma importância.

Já a métrica *Weighted avg* leva em consideração o peso de cada classe, sendo a média ponderada das medidas de *precision*, *recall* e *f1-score* para cada classe. Essa métrica é útil quando as classes têm importância desigual.

No cenário inicialmente apresentado neste trabalho, a rede neural obteve uma eficácia de 100% em todas as métricas, conforme figura 15, abaixo. Isso poderia indicar que a rede neural foi capaz de classificar corretamente todas as instâncias em todas as classes, no entanto acende-se um ponto de alerta para a possibilidade da rede apresentar vícios de treinamento. Considerando isso, devemos analisar os demais indicadores de desempenho da rede.

	precision	recall	f1-score	support
SEM indicação de defeitos	1.00	1.00	1.00	7
COM indicação de defeitos	1.00	1.00	1.00	5
accuracy			1.00	12
macro avg	1.00	1.00	1.00	12
weighted avg	1.00	1.00	1.00	12

Figura 15: Resultados do treinamento da rede neural, impressos em tela com uso da função *metrics.classification\_report*

É importante ressaltar que, mesmo quando uma rede neural apresenta uma eficácia de 100%, é necessário avaliar cuidadosamente o desempenho do modelo. Isso inclui avaliar a generalização da rede, sua capacidade de realizar a tarefa em dados não utilizados no treinamento, e a interpretabilidade da rede, ou seja, sua capacidade de explicar suas decisões.

Em resumo, as medidas de *precision*, *recall* e *f1-score* foram essenciais para avaliar o desempenho desta rede neural na classificação dos dados disponíveis. As métricas *Macro avg* e *Weighted avg* são úteis para avaliar o desempenho geral da rede em cenários com classes de importância desigual. No cenário apresentado neste trabalho, a rede neural obteve um desempenho de 100% em todas as métricas, o que deveria indicar alta eficácia na solução de problemas complexos de reconhecimento de padrões.

## 5.2. AVALIANDO OS GRÁFICOS IMPRESSOS

### 5.2.1. EVOLUÇÃO DO TREINAMENTO

Outro ponto de diagnóstico para verificar o resultado apresentado pela rede é o gráfico da evolução da acurácia e do erro durante o treinamento da rede neural artificial em função do número de épocas, figura 16, que pode fornecer informações importantes sobre o desempenho do modelo. Quando a acurácia aumenta continuamente e o erro diminui continuamente, é um sinal de que o modelo está se ajustando bem aos dados de treinamento.

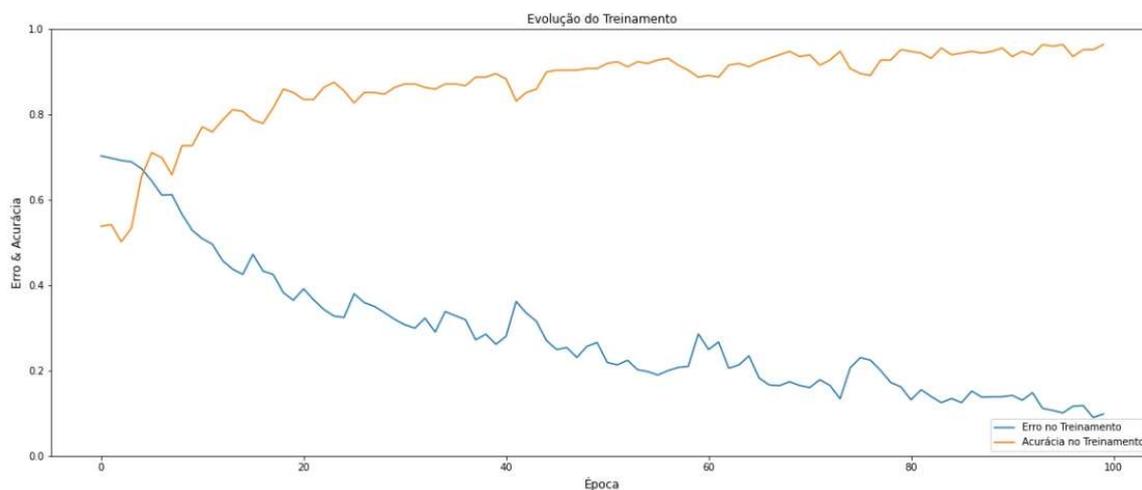


Figura 16: Gráfico de evolução do treinamento para ERRO e ACURÁCIA

A acurácia representa a proporção de classificações corretas do modelo em relação ao total de classificações feitas, enquanto o erro representa a diferença entre as classificações feitas pelo modelo e as classes corretas dos dados de treinamento. Quando a acurácia aumenta, isso significa que o modelo está se tornando cada vez mais preciso em suas

previsões. Já a diminuição do erro indica que o modelo está aprendendo a generalizar melhor, ou seja, a capacidade de fazer previsões precisas em dados que nunca foram vistos antes.

Ponto importante a se ressaltar é que, apesar que uma melhora contínua na acurácia e no erro no treinamento não necessariamente significa que o modelo está se comportando bem em outros conjuntos de dados, como os de validação e de teste. Portanto, é necessário realizar a validação cruzada e testar o modelo em conjuntos de dados diferentes do conjunto de treinamento para verificar se o desempenho é consistente em todos eles.

Além disso, é preciso ter cuidado com o *overfitting*, que ocorre quando o modelo se ajusta muito bem aos dados de treinamento, mas não consegue generalizar bem para outros conjuntos de dados. Nesse caso, é possível observar que a acurácia no treinamento continua a aumentar, enquanto a acurácia na validação cruzada diminui ou se estabiliza. Portanto, é importante encontrar um equilíbrio entre a complexidade do modelo e sua capacidade de generalização.

Em resumo, o gráfico da evolução da acurácia e do erro no treinamento pode fornecer informações valiosas sobre o desempenho de um modelo de rede neural artificial, mas é importante ter em mente que outros conjuntos de dados e medidas de desempenho também devem ser considerados para garantir uma boa generalização do modelo.

### **5.2.2. EVOLUÇÃO DO ERRO**

O gráfico comparativo de evolução do erro no treinamento e na validação cruzada, figura 17, é uma ferramenta importante para avaliar o desempenho da rede neural artificial durante o processo de treinamento. O número de épocas são as iterações do processo de treinamento, e o erro é a diferença entre as saídas previstas pela rede e as saídas desejadas.

O erro de treinamento representa o erro da rede neural em relação aos dados de treinamento, enquanto o erro na validação cruzada representa o erro da rede neural em relação a dados que não foram utilizados no treinamento. A comparação entre os dois erros ao longo das épocas permite avaliar se a rede está tendo um bom desempenho na generalização, ou seja, se está conseguindo fazer previsões precisas para novos dados.

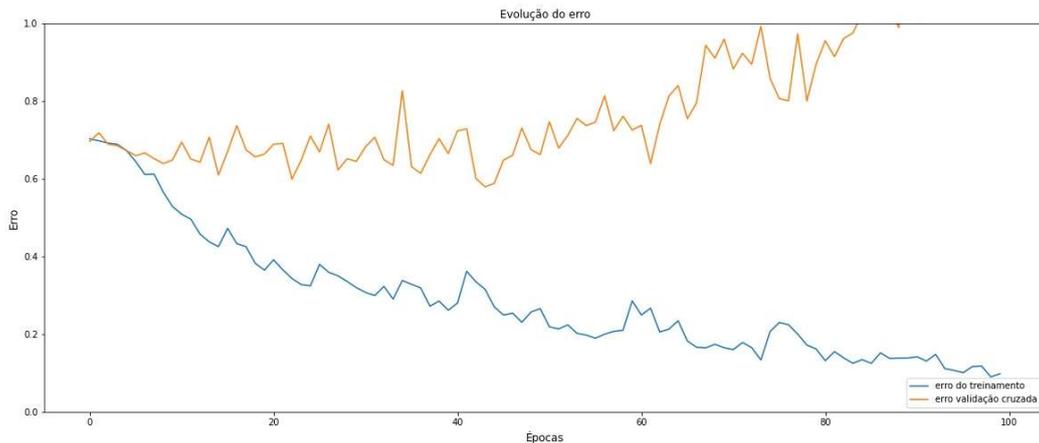


Figura 17: gráfico comparativo do ERRO no treinamento e na validação cruzada

Para o caso deste estudo, o erro de treinamento apresentou uma diminuição ao longo das épocas, porém o erro na validação cruzada aumentou, isto é um forte indício que a rede pode estar em *overfitting*, ou seja, memorizou os dados de treinamento em vez de aprender a relação geral entre as entradas e saídas.

### 5.2.3. EVOLUÇÃO DA ACURÁCIA

O gráfico da figura 18 mostra a evolução da acurácia da rede neural em função das épocas de treinamento, comparação gráfica da acurácia na validação cruzada e no treinamento, enquanto a rede é treinada no conjunto de dados, trata-se de uma importante ferramenta para avaliar o desempenho de uma rede neural artificial. Dado que a acurácia mede a proporção de previsões corretas realizadas pelo modelo e é a métrica fundamental para avaliar a qualidade do modelo de aprendizado de máquina escolhido. A acurácia no treinamento é a porcentagem de instâncias do conjunto de dados que a rede neural acertou. Já na validação cruzada, é a porcentagem de instâncias de um conjunto de dados separado que a rede neural acertou.

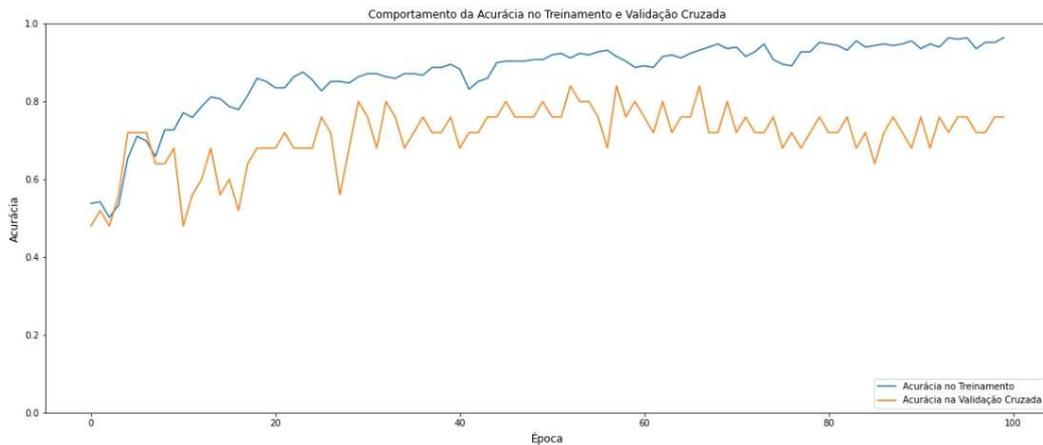


Figura 18: gráfico comparativo da ACURÁCIA no treinamento e na validação cruzada

Interpretando o resultado apresentado pelo gráfico, extrai-se que a acurácia no treinamento aumenta ao longo do tempo, à medida que a rede neural aprende a se ajustar aos dados de treinamento. No entanto, numa observação mais cautelosa, nota-se que a acurácia na validação cruzada apresenta uma tendência de aumento até certo ponto, e depois, entra em tendência de diminuição. Isso indica que a rede neural está a se ajustar aos dados de treinamento e não parece ser capaz de generalizar para dados novos. Isso ocorre porque o modelo está aprendendo os detalhes dos dados de treinamento, mas não está aprendendo as características gerais dos dados. Como resultado, o modelo não será capaz de generalizar para dados novos.

O aumento da variação da acurácia na validação cruzada e um aumento contínuo da acurácia no treinamento em relação ao número de épocas pode indicar que a rede está sofrendo de *overfitting*. No gráfico, o ponto em que a acurácia na validação cruzada começa a diminuir é chamado de ponto de *overfitting*, que sinaliza, em português, o sobre ajuste. Seria o ponto ideal para interromper o treinamento da rede neural antes de atingir esse ponto, a fim de evitar esta ocorrência.

Uma forma possível de evitar o *overfitting* seria utilização de um conjunto de validação maior, alternativa que foi avaliada posteriormente, (conforme veremos adiante) aumentando assim a variabilidade dos dados de validação.

#### 5.2.4. EVOLUÇÃO DA VALIDAÇÃO CRUZADA

Analisemos então o gráfico da evolução do erro e da acurácia na validação cruzada, em função do número de épocas, figura 19, que é uma ferramenta importante para avaliar o desempenho da rede neural. Como anteriormente dito, o erro é uma medida de quão distante

a saída da rede neural está do valor desejado, e a acurácia é uma medida de quão precisamente a rede neural está classificando as amostras de entrada.

Durante as primeiras épocas, era esperado que o erro fosse relativamente alto e a acurácia baixa, pois a rede ainda não saberia a relação entre as entradas e as saídas, e conforme a rede neural fosse treinada por mais épocas, esperava-se que o erro diminuísse e a acurácia aumentasse. Entretanto, após um determinado número de épocas, aparentemente, o modelo se ajustou aos dados de treinamento, levando a uma estabilização na acurácia, com aumento no erro, na validação cruzada.

O fato de a acurácia oscilar dentro de uma determinada faixa de valor e o erro aumentar com o número de épocas pode indicar que a rede está em *overfitting*, que ocorre quando a rede neural se ajusta excessivamente aos dados de treinamento, de modo que acaba perdendo a capacidade de generalizar para dados desconhecidos. Como resultado, a acurácia pode parecer alta no conjunto de treinamento, mas baixa no conjunto de validação. Por outro lado, se a acurácia e o erro estivessem estabilizados em níveis baixos, poderia indicar que a rede estaria bem ajustada aos dados e teria boa capacidade de generalização.

Resumidamente, pode-se dizer que o gráfico de evolução do erro e da acurácia na validação cruzada revelou, valiosamente, que o desempenho desta rede neural artificial sofria com problemas de *overfitting*.

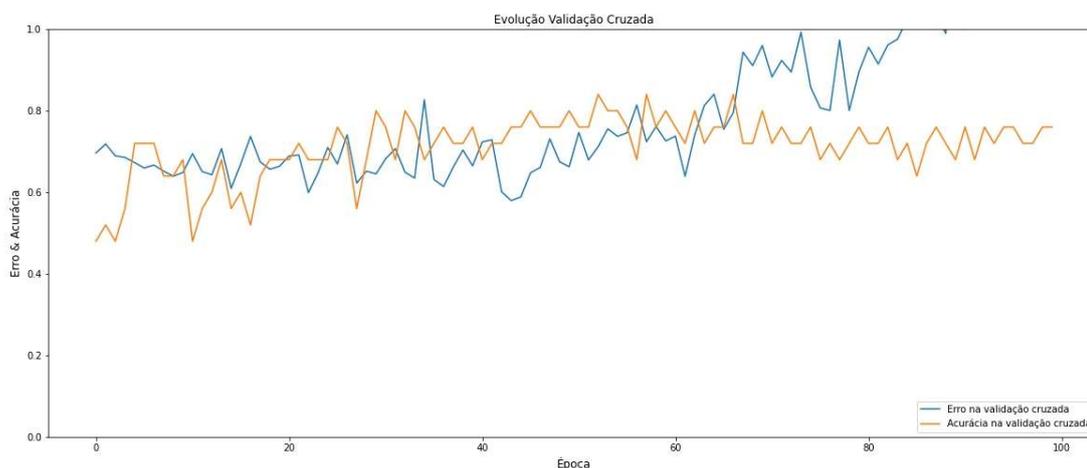


Figura 19: Gráfico de evolução da validação cruzada para ERRO e ACURÁCIA.

### 5.3. AJUSTANDO ALGUNS DOS HIPERPARÂMETROS DA REDE NEURAL

#### 5.3.1. JUSTIFICATIVA PARA AJUSTE NOS HIPERPARÂMETROS

Considerando que o código utilizado na rede neural original apresentou resultados insatisfatórios, com o erro alto e baixa precisão, foram alterados os parâmetros, *seed*, o número de neurônios e a amostragem para validação cruzada.

A *seed* é um valor aleatório usado para inicializar os pesos da rede neural. Ao alterar a *seed*, é possível obter resultados diferentes, pois a rede neural será inicializada de forma diferente. Nesse caso, a *seed* foi alterada para um valor menor, o que resultou em uma rede neural mais robusta e menos propensa a *overfitting*.

O número de neurônios é um parâmetro importante que afeta a capacidade da rede neural de aprender. Se o número de neurônios for muito pequeno, a rede neural pode não ser capaz de aprender os padrões nos dados de treinamento. Se o número de neurônios for muito grande, a rede neural pode acabar aprendendo padrões irrelevantes nos dados de treinamento, o que também pode levar a resultados incorretos. Na alteração proposta, foi reduzido de maneira drástica o número de neurônios.

A amostragem para validação cruzada é um processo usado para avaliar a performance da rede neural em dados de teste. Ao alterar a amostragem para validação cruzada, é possível obter uma avaliação mais precisa da performance da rede neural. Nesse caso, a amostragem para validação cruzada foi alterada para um número maior de *folds*, o que resultou em uma avaliação mais robusta da performance da rede neural.

Antes de apresentar os resultados, é essencial compreender as mudanças feitas nos parâmetros da rede neural. A tabela 2, a seguir, mostra os parâmetros originais e as correções realizadas durante o experimento. Esses ajustes foram fundamentais para otimizar o desempenho da rede, visando uma melhor capacidade de aprendizado e generalização. Analisar esses parâmetros permite uma avaliação mais completa do impacto das mudanças na arquitetura da rede e sua influência nos resultados obtidos.

Tabela 2: Tabela comparativa com os parâmetros utilizados nas duas fases do experimento.

	<b>Fase 1</b> Rede Original	<b>Fase 2</b> Após correção dos hiperparâmetros
Imagens 'SEM indicação de descontinuidade':	92	148
Imagens 'COM indicação de descontinuidade'	89	138

Total de imagens utilizadas	181	286
Gerador de números pseudoaleatórios ( <i>seed</i> ):	400	45
Separação dos dados para teste ( <i>train_test_split</i> ):	0,04	0,18
Imagens utilizadas para teste	12	52
Camada de Pooling ( <i>MaxPool2D</i> ):	<code>pool_size = 6 padding = 'same'</code>	<code>pool_size = 4 padding = 'same'</code>
Camada de Convolução ( <i>Conv2D</i> ):	<code>filters = 16 kernel_size = 2 padding = 'same' activation = 'relu'</code>	<code>filters = 16 kernel_size = 2 padding = 'same' activation = 'relu'</code>
Camada de Convolução ( <i>Conv2D</i> ):	<code>filters = 32 kernel_size = 2 padding = 'same' activation = 'relu'</code>	<code>filters = 16 kernel_size = 2 padding = 'same' activation = 'relu'</code>
Camada de Pooling ( <i>MaxPool2D</i> ):	<code>pool_size = 4 strides = 2 padding = 'valid'</code>	<code>pool_size = 4 strides = 2 padding = 'valid'</code>
Camada Densa ( <i>Dense</i> ):	<code>Nº neurônios = 400 activation = 'relu'</code>	<code>Nº neurônios = 50 activation = 'relu'</code>
Camada Densa ( <i>Dense</i> ):	<code>Nº neurônios = 1000 activation = 'relu'</code>	<code>Nº neurônios = 10 activation = 'relu'</code>
Camada Densa de Saída ( <i>Dense</i> ):	<code>Nº neurônios = 1 activation = 'sigmoid'</code>	<code>Nº neurônios = 1 activation = 'sigmoid'</code>
Etapa de compilação ( <i>model.compile</i> ):	<code>loss = 'binary_crossentropy' optimizer = 'Adam' metrics = ['accuracy']</code>	<code>loss = 'binary_crossentropy' optimizer = 'Adam' metrics = ['accuracy']</code>
Etapa de Treinamento ( <i>model.fit</i> ):	<code>epochs = 100 validation_split = 0.09</code>	<code>epochs = 150 validation_split = 0.18</code>
Acurácia obtida	100%	88%

### 5.3.2. RESULTADOS OBTIDOS APÓS AJUSTE

A seguir, neste capítulo, têm-se as figuras 20 a 25, que representam os mesmo gráficos acima detalhados, no entanto, mostram os resultados dos testes da rede neural após a correção dos hiperparâmetros. As cinco figuras detalham o desempenho da rede, e destacam a variação da acurácia e do erro em função do aumento do número de épocas, que demonstram as relevantes mudanças após as correções. Sendo estes resultados cruciais para avaliar o impacto das alterações nos hiperparâmetros na eficácia e na capacidade de generalização da

rede neural. A análise dessas figuras nos permite uma valorosa avaliação para o aprimoramento contínuo do modelo.

	precision	recall	f1-score	support
SEM indicação de defeitos	0.85	0.97	0.90	29
COM indicação de defeitos	0.95	0.78	0.86	23
accuracy			0.88	52
macro avg	0.90	0.87	0.88	52
weighted avg	0.89	0.88	0.88	52

Figura 20: Resultados do TESTE de treinamento da rede neural, impressos em tela com uso da função `metrics.classification_report`, após ajuste dos hiperparâmetros;

	precision	recall	f1-score	support
SEM indicação de defeitos	0.83	0.93	0.88	119
COM indicação de defeitos	0.92	0.80	0.86	115
accuracy			0.87	234
macro avg	0.87	0.87	0.87	234
weighted avg	0.87	0.87	0.87	234

Figura 21: Resultados do TREINAMENTO da rede neural, impressos em tela com uso da função `metrics.classification_report`, após ajuste dos hiperparâmetros;

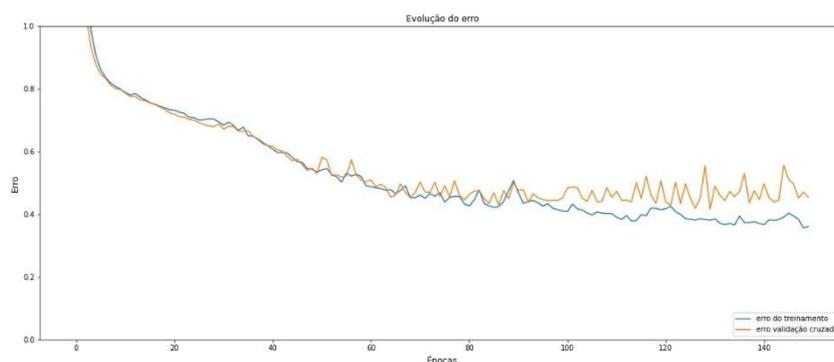


Figura 22: gráfico comparativo do ERRO no treinamento e na validação cruzada, após ajuste dos hiperparâmetros;

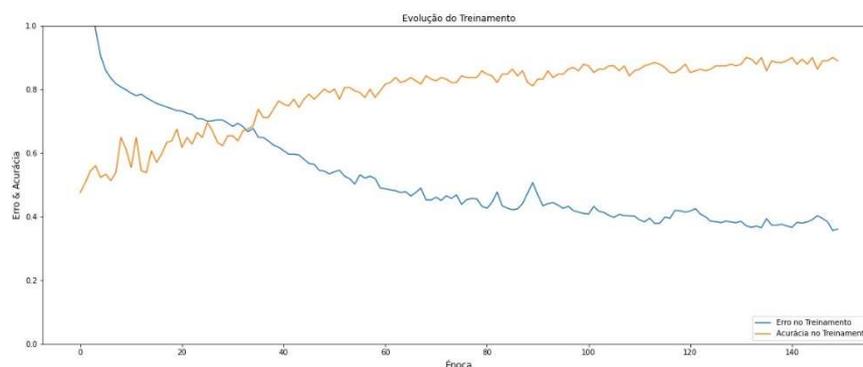


Figura 23: Gráfico de evolução do treinamento para ERRO e ACURÁCIA, após ajuste dos hiperparâmetros.

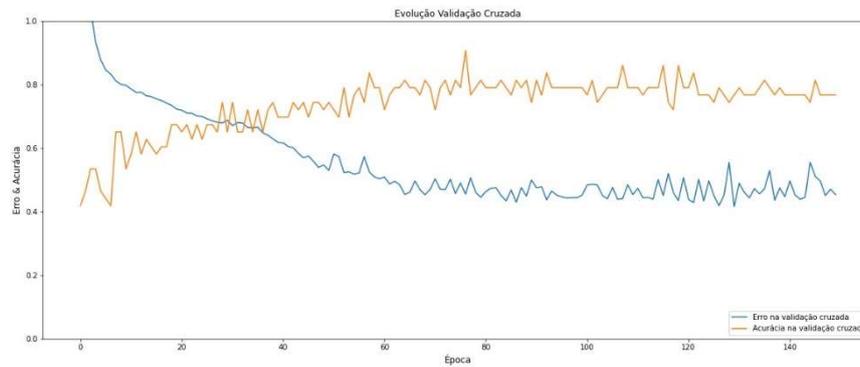


Figura 24: Gráfico de evolução da validação cruzada para ERRO e ACURÁCIA, após ajuste dos hiperparâmetros;

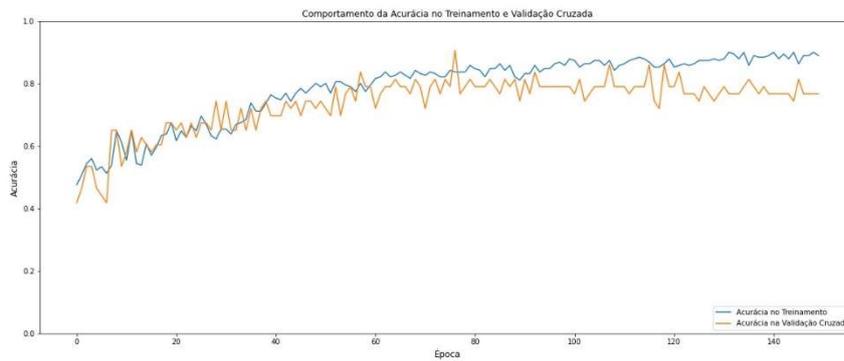


Figura 25: gráfico comparativo da ACURÁCIA no treinamento e na validação cruzada, após ajuste dos hiperparâmetros;

## 6. CONCLUSÃO

O uso de redes neurais artificiais para classificação de imagens, como amplamente noticiado nos últimos tempos, tem se mostrado uma técnica altamente eficaz na solução de problemas variados, e áreas diversas, como medicina, indústria e robótica. Neste trabalho, foi utilizada uma rede neural para classificar imagens, trabalhando com um banco de dados de tamanho médio, e ainda assim pôde alcançar um resultado superior a 88% de eficácia.

Os aspectos sutis de diferença entre as imagens utilizadas foram um desafio particularmente audaz na tarefa de classificação destas imagens, pois exigiu que a rede neural fosse capaz de reconhecer padrões pouco significativos de diferença e realizar previsões com base nos dados de entrada. Há de se ressaltar que uma das preocupações iniciais deste trabalho foi a possível perda de informação ao transitar a imagem pelas funções de *pooling*, pois ao compactar as informações disponibilizadas em pixels, poderiam ser ocultadas informações relevantes das indicações de solda. No entanto, a eficácia superior a 85% obtida na classificação das imagens indica que a rede neural utilizada neste trabalho foi

capaz de reconhecer e distinguir com precisão os aspectos sutis que diferenciam cada imagem.

O resultado obtido neste trabalho é bastante significativo, pois indica a viabilidade do uso de redes neurais artificiais para solucionar problemas complexos de classificação de imagens, ou problemas de captação tênue ou de características amenas, como no caso das imagens deste banco de dados. Intrinsecamente, é sugestivo que a técnica utilizada tem potencial para ser aplicada em diversas áreas, permitindo a agilidade dos peritos humanos, ou ainda minimizando a propensão a erros por fadiga, cansaço, e fatores inerentes à ação humana, de tarefas, com uma significativa redução de custos e tempo em processos de análise de imagens.

Entretanto, há de se destacar que a eficácia da rede neural depende de diversos fatores, como a qualidade dos dados utilizados no treinamento, a arquitetura da rede, o algoritmo de otimização utilizado e a capacidade de processamento da máquina utilizada. Além disso, é preciso avaliar a generalização da rede, ou seja, sua capacidade de realizar a tarefa em dados não utilizados no treinamento.

Outro ponto a ser considerado é a interpretabilidade da rede neural, ou seja, a capacidade de entender como a rede neural chegou a uma determinada decisão. Isso é especialmente importante em áreas como a medicina, em que é necessário entender o raciocínio da rede para validar suas decisões.

É cabível destacar a importância de avaliar cuidadosamente os fatores que influenciam a eficácia da rede neural, a interpretabilidade, ou seja, a capacidade de entender como a rede neural chegou a uma determinada decisão, e suas limitações antes de aplicá-la em outros contextos. É necessário continuar a pesquisa na área de interpretabilidade de redes neurais, a fim de garantir a confiabilidade e segurança das decisões tomadas por este poderoso recurso.

Em conclusão, o resultado obtido neste trabalho demonstra a eficácia do uso de redes neurais artificiais para classificação de imagens e abre portas para o desenvolvimento de novas aplicações em diversas áreas. No entanto, é necessário continuar a pesquisa na área de redes neurais e considerar cuidadosamente seus limites e desafios para garantir a confiabilidade e segurança de sua aplicação.

## 7. SUGESTÃO DE TRABALHOS FUTUROS

O trabalho proposto tem como objetivo verificar a capacidade de uma rede neural artificial na detecção de defeitos reais em processos de soldagem de escala industrial, executado por profissional qualificado, ou seja, defeitos não didáticos. Considerando o papel fundamental que a soldagem tem para a fabricação de equipamentos e componentes nucleares para operação de usinas de geração nucleoeletricas e outras plantas nucleares, é de extrema importância garantir a qualidade e a segurança dos produtos.

Neste contexto, as sugestões de trabalhos futuros são em duas direções principais: A primeira visa à classificação de defeitos de soldagem de diversos tipos, estendendo a capacidade de reconhecimento a uma gama de situações (inclusão de escória, inclusão de tungstênio, falta de fusão, falta de penetração, etc). A segunda sugestão aborda a aplicação de redes neurais no processamento de imagens de filmes radiográficos de soldagem em tamanho padrão, com o objetivo de aprimorar a inspeção por meio da automação. Uma terceira sugestão sugere a integração de técnicas de aprendizado profundo com outros métodos de inspeção, visando uma abordagem multimodal para a detecção de defeitos.

### Sugestão 1: Classificação de defeitos de soldagem de diversos tipos

Uma promissora pesquisa futura seria a expansão do escopo da detecção de defeitos de soldagem nos vários tipos de defeitos conhecidos. O desenvolvimento de uma rede neural artificial capaz de identificar e classificar defeitos em soldas, desde que disponha de um generoso banco de dados para o treinamento da rede, e assim ampliando a capacidade de reconhecimento para as categorias de defeitos reconhecidos, permitiria uma aplicação mais abrangente e eficaz, para que possa ser pensado em futuro na indústria.

### Sugestão 2: Análise de filmes radiográficos de soldagem com redes neurais

Outra sugestão é a aplicação de redes neurais artificiais na análise de imagens radiográficas de soldagens. Considerando que os filmes radiográficos ainda são uma ferramenta comum para inspeção de soldagens, a automação desse processo por meio de redes neurais poderia revolucionar a detecção de defeitos. Um trabalho futuro poderia envolver o desenvolvimento de algoritmos de processamento de imagens e redes neurais

especializadas para analisar filmes radiográficos em tamanho padrão, identificando e classificando defeitos com alta precisão.

Sugestão 3: Integração de técnicas de aprendizado profundo com outros métodos de inspeção

Uma abordagem inovadora para trabalhos futuros poderia ser a integração de redes neurais de aprendizado profundo com outros métodos de inspeção, como ultrassom, partículas magnéticas e líquidos penetrantes. Isso permitiria uma abordagem multimodal na detecção de defeitos, aumentando a robustez do sistema e reduzindo falsos positivos. A pesquisa poderia se concentrar na criação de uma plataforma de inspeção unificada que aproveite as vantagens de diversas técnicas de inspeção, combinadas com redes neurais, para um diagnóstico mais preciso e abrangente de soldagens em diversas condições e materiais.

## 8. REFERÊNCIAS

1. AWS D1.1/D1.1M: 2008, Structural Welding Code-Steel, AMERICAN WELDING SOCIETY, AWS, Miami, Florida, USA. American Welding Society, AWS D1.1;
2. Welding Inspection Handbook, AWS B1C STANDING TASK GROUP, AWS, Miami, Florida, USA. American Welding Society, Fourth Edition, 2015;
3. BENUWA, B. B. et al. A review of deep machine learning. International Journal of Engineering Research in Africa, v. 24, p. 124-136, 2016.
4. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
5. Brownlee, J. (2017). Gentle Introduction to the ADAM Optimization Algorithm for Deep Learning. Disponível em: <https://machinelearningmastery.com/ADAM-optimization-algorithm-for-deep-learning/>.
6. Brownlee, J. (2019). Evaluate the Performance of Deep Learning Models in Keras. Machine Learning Mastery. Disponível em: <https://machinelearningmastery.com/evaluate-performance-deep-learning-models-keras/>
7. Chollet, F. (2018). Deep Learning with Python. Manning Publications.
8. Dahl, G. E., Jaitly, N., and Salakhutdinov, R. (2014). Multi-task neural networks for QSAR predictions. arXiv:1406.1231.
9. Documentação oficial do Keras. Disponível em: <[https://keras.io/api/models/model\\_training\\_apis/#evaluate-method](https://keras.io/api/models/model_training_apis/#evaluate-method)> Acesso em 19/03/2022;
10. Dos Santos, Marcelo Carvalho, SISTEMA BASEADO EM REDES NEURAIAS CONVOLUCIONAIS, RECORRENTES E AUTOCODIFICADORAS PARA CLASSIFICAÇÃO DE ACIDENTES POSTULADOS EM CENTRAIS NUCLEARES COM CAPACIDADE DE DETECÇÃO DE ANOMALIAS E RESPOSTA “NÃO SEI”, Exame de Qualificação ao Doutorado – COPPE/UFRJ, 2020.
11. Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. O'Reilly Media, Inc.
12. Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media.
13. Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (pp. 249-256).
14. GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. MIT Press, 2016.

15. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.
16. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778).
17. HINTON, G. E.; SALAKHUTDINOV, R. R. Reducing the dimensionality of data with neural networks. *Science*, 313 (5786):504–507, 2006.
18. Van Veen, F. & Leijnen, S. (2019). *The Neural Network Zoo*. Retrieved from <https://www.asimovinstitute.org/neural-network-zoo>
19. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning: with Applications in R*. Springer.
20. Keras Code Examples. Disponível em: <https://keras.io/examples/>
21. Kingma, D. P., & Ba, J. (2014). ADAM: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
22. KRAMER, M. A. Autoassociative neural networks. *Computers and Chemical Engineering*, v. 16, p. 313-328, 1992.
23. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
24. LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature (London)*, 521(7553), pp 436-444, 2015.
25. Marques, P.V.; Modenesi, P.J.; Bracarense, A. Q.; Soldagem: Fundamentos e tecnologia, v. 3. Belo Horizonte, Brasil, Editora UFMG, 2009.
26. Minsky, Marvin, & Papert, Seymour A. (1969). *Perceptrons: An Introduction to Computational Geometry*, Cambridge, MA - The MIT Press, January 15, 1969,
27. Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning* (pp. 807-814).
28. NDT RESOURCE CENTER. “Radiograph Interpretation – Welds”. Disponível em: <https://www.nde-ed.org/EducationResources/CommunityCollege/Radiography/TechCalibrations/RadiographInterp.htm>. Acesso em 01 de dezembro de 2020 as 19:00:03.
29. PFEIL, Walter; PFEIL, Michèle. *Estruturas de aço: dimensionamento prático*. 8ª ed., Rio de Janeiro – RJ: Ed. LTC, 2014;
30. RASCHKA, S.; MIRJALILI, V. *Python Machine Learning - Second Edition: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*. Second Edition ed.: Packt Publishing, 2017.

31. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408.
32. Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.
33. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533-536.
34. S. Shankar, V. K. Garg, and R. Cipolla. Deep-carving: Discovering visual attributes by carving deep neural nets. In CVPR, June 2015
35. SAEED, H. A. et al. Novel fault diagnosis scheme utilizing deep learning networks. *Progress in Nuclear Energy*, v. 118, p. 103066, 2020.
36. SCHMIDHUBER, J. Deep Learning in neural networks: An overview. *Neural Networks*, v. 61, p.85-117, 2015.
37. Scikit-learn developers. (2021). Scikit-learn: Machine Learning in Python. Retrieved from <https://scikit-learn.org/stable/index.html>
38. Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
39. Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In International conference on learning representations.
40. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929-1958.
41. SRIVASTAVA, R. K.; GREFF, K.; SCHMIDHUBER, J. Training Very Deep Networks. *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, p. 1–9, 2015.
42. Telles, Pedro Carlos da Silva. *Vasos de Pressão*. 2ª ed., Rio de Janeiro – RJ: Ed. LTC, 2010;
43. Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In European conference on computer vision (pp. 818-833). Springer.
44. Deep Learning Book, Disponível em: <https://www.deeplearningbook.com.br/uma-breve-historia-das-redes-neurais-artificiais/> Acesso em: 15/03/2022
- 45.