



UNIVERSIDADE FEDERAL DE GOIÁS (UFG)/
UNIVERSIDADE FEDERAL DE CATALÃO (UFCAT) em implantação
INSTITUTO DE MATEMÁTICA E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM E OTIMIZAÇÃO

Renan Garcia Rosa

GERAÇÃO DE ASSINATURAS DE IMPEDÂNCIA PARA O
MONITORAMENTO DE INTEGRIDADE ESTRUTURAL UTILIZANDO
Deep Learning

DISSERTAÇÃO DE MESTRADO

CATALÃO – GO, 2022



UNIVERSIDADE FEDERAL DE GOIÁS
UNIDADE ACADÊMICA ESPECIAL DE MATEMÁTICA E TECNOLOGIA

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a [Lei 9.610/98](#), o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

Dissertação Tese Outro*: _____

*No caso de mestrado/doutorado profissional, indique o formato do Trabalho de Conclusão de Curso, permitido no documento de área, correspondente ao programa de pós-graduação, orientado pela legislação vigente da CAPES.

Exemplos: Estudo de caso ou Revisão sistemática ou outros formatos.

2. Nome completo do autor

Renan Garcia Rosa

3. Título do trabalho

GERAÇÃO DE ASSINATURAS DE IMPEDÂNCIA PARA O MONITORAMENTO DE INTEGRIDADE ESTRUTURAL UTILIZANDO DEEP LEARNING

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento SIM NÃO¹

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:

a) consulta ao(à) autor(a) e ao(à) orientador(a);

b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.

O documento não será disponibilizado durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **Jose Dos Reis Vieira De Moura Junior, Orientador**, em



15/07/2022, às 11:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **RENAN GARCIA ROSA, Discente**, em 15/07/2022, às 11:39, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site

[https://sei.ufg.br/sei/controlador_externo.php?](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0)

[acao=documento_conferir&id_orgao_acesso_externo=0](https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **3046953** e o código CRC **885345DB**.

RENAN GARCIA ROSA

GERAÇÃO DE ASSINATURAS DE IMPEDÂNCIA PARA O
MONITORAMENTO DE INTEGRIDADE ESTRUTURAL UTILIZANDO
Deep Learning

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem e Otimização do Instituto de Matemática e Tecnologia, da Universidade Federal Catalão, como requisito final para a obtenção do título de Mestre em Modelagem e Otimização. Área de concentração: Modelagem e Otimização. Linha de Pesquisa: Modelagem Computacional e Otimização.

Orientador:
José dos Reis Vieira de Moura Júnior

CATALÃO – GO

2022

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFCAT.

Rosa, Renan Garcia

Geração de assinaturas de impedância para o monitoramento de integridade estrutural Utilizando Deep Learning / Renan Garcia Rosa.
- 2022.

95, f.: il.

Orientador: Prof. Dr. José dos Reis Vieira de Moura Júnior.

Dissertação (Mestrado) - Universidade Federal de Catalão, Instituto de Matemática e Tecnologia, Catalão, Programa de Pós-Graduação em Modelagem e Otimização, Catalão, 2022.

Bibliografia.

Inclui siglas, fotografias, abreviaturas, símbolos, gráfico, tabelas, lista de figuras, lista de tabelas.

1. Rede Generativa adversária como classificador auxiliar. 2. Monitoramento da Integridade Estrutural. 3. Geração de Dados. 4. Geração de sinais. I. Moura Júnior, José dos Reis Vieira de , orient. II. Título.

CDU 517



UNIVERSIDADE FEDERAL DE GOIÁS

UNIDADE ACADÊMICA ESPECIAL DE MATEMÁTICA E TECNOLOGIA

ATA DE DEFESA DE DISSERTAÇÃO

Ata nº 16 da sessão de Defesa de Dissertação de **Renan Garcia Rosa**, que confere o título de **Mestre(a) em Modelagem e Otimização**, na área de concentração em **Modelagem e Otimização**.

Aos quinze dias do mês de julho de 2022, a partir das 07h30min, na sala web-conferência, realizou-se a sessão pública de Defesa de Dissertação, intitulada "*GERAÇÃO DE ASSINATURAS DE IMPEDÂNCIA PARA O MONITORAMENTO DE INTEGRIDADE ESTRUTURAL UTILIZANDO DEEP LEARNING*" nas dependências da Universidade Federal de Catalão, onde os programas de pós-graduação *Stricto Sensu* em funcionamento encontram-se provisoriamente vinculados à Universidade Federal de Goiás, em virtude de procedimentos técnicos relacionados à CAPES, já sendo realizada a transferência da Biblioteca Digital de Dissertações e Teses (BDTD). Assim, justifica-se os nomes das instituições neste documento, uma no cabeçalho (UFG), outra no corpo do texto (UFCAT). Os trabalhos foram instalados pelo(a) **Orientador, Professor Doutor José dos Reis Vieira de Moura Júnior (PPGMO/IMTec/UFCAT)**, com a participação dos demais membros da Banca Examinadora: **Professor Doutor Celso Vieira Abud (PPGMO/IMTec/UFCAT)**, membro titular interno; **Professor Doutor William Júnio de Lima (UFU)**, membro titular externo. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Dissertação, tendo sido o candidato (X) Aprovado () Reprovado pelos seus membros. Proclamados os resultados pelo **Professor Doutor José dos Reis Vieira de Moura Júnior**, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos quinze dias do mês de julho de dois mil e vinte e dois.

Obs: "*Banca Examinadora de Qualificação/Defesa Pública de Dissertação/Tese realizada em conformidade com a Portaria da CAPES n. 36, de 19 de março de 2020, de acordo com seu segundo artigo:*

Art. 2o A suspensão de que trata esta Portaria não afasta a possibilidade de defesas de tese utilizando tecnologias de comunicação à distância, quando admissíveis pelo programa de pós-graduação stricto sensu, nos termos da regulamentação do Ministério da Educação."

TÍTULO SUGERIDO PELA BANCA



Documento assinado eletronicamente por **Jose Dos Reis Vieira De Moura Junior, Orientador**, em 15/07/2022, às 11:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Celso Vieira Abud, Professor do Magistério Superior**, em 15/07/2022, às 11:17, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **William Júnio de Lima, Usuário Externo**, em 15/07/2022, às 11:18, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

Os Programas de Pós-Graduação *stricto sensu* em funcionamento na Universidade Federal de Catalão (UFCAT), em virtude de procedimentos técnicos relacionados à CAPES, continuam provisoriamente vinculados à Universidade Federal de Goiás (UFG), por isso, todos os elementos pré-textuais do trabalho apresentado estão identificados como Universidade Federal de Goiás/Universidade Federal de Catalão em implantação, em função da migração da BDTD ter ocorrido a partir de 16 de agosto de 2021, assim como pelo fato das pesquisas e produtos serem realizados na UFCAT.

Dedico esse trabalho à Deus que sempre ilumina meus passos, a minha esposa Camila e aos meus pais Carlos e Mirna, por sempre apoiarem meus sonhos.

Dedico (In Memoriam) ao meu avô Augusto, que não conseguiu ver esse momento mas acreditou e me ajudou muito ao decorrer da caminhada.

Agradecimentos

Agradeço em primeiro lugar a Deus, por ter me dado força e determinação para conquistar essa vitória na minha vida.

Agradeço imensamente meu orientador, José dos Reis, pela amizade e carinho. Sem dúvidas, o meu desenvolvimento foi e continuará sendo espelhado em seus vários ensinamentos tanto de vida quanto profissional. Deixo aqui meu agradecimento por todos ensinamentos, conversas e momentos felizes e tristes que passamos e ainda vamos passar.

Ao grande amigo Stanley Washington, deixo aqui meu agradecimento pelo companheirismo e amizade. Todo apoio foi fundamental para o meu desenvolvimento e deste trabalho. Que eu possa contribuir de maneira recíproca ao longo de nossas vidas.

À minha esposa Camila, deixo um agradecimento especial, por todas as lições de amor, companheirismo, amizade, caridade, dedicação, abnegação, compreensão a cada novo dia. Em minha vida a considero meu pilar e sua grande força foi a mola propulsora que permitiu o meu avanço, mesmo durante os momentos mais difíceis. Sinto-me orgulhoso e privilegiado por tê-la.

Aos meus pais Carlos e Mirna, deixo um carinhoso agradecimento, frente a todos os momentos de alegria, amor e amizade que passamos juntos. Vejo muitas características suas em mim e acredito que grande parte da minha capacidade é herança da vossas pessoas. Tenho orgulho e me considero privilegiado por tê-los.

Ao meu irmão Rafael, deixo meu agradecimento. Apesar da distância acredito que você sempre se fez presente de alguma forma. Sou grato pela sua vida e espero estarmos juntos sempre.

A Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela bolsa de pesquisa concedida ao autor desta dissertação de mestrado.

Por fim agradeço aos meus avós Nenê, Augusto (In Memoriam) e Mariana, por terem cuidado de mim quando precisei, por terem me ensinado tanto e por todo o amor e carinho. Saibam que sem o suporte de vocês, não teria chegado até aqui.

“A vitalidade é demonstrada não apenas pela persistência, mas pela capacidade de começar de novo.”

F. Scott Fitzgerald

RESUMO

ROSA, RENAN GARCIA. *Geração de assinaturas de impedância para o monitoramento de integridade estrutural Utilizando Deep Learning*. 2022. 79 f. Dissertação (Mestrado em Modelagem e Otimização) – Instituto de Matemática e Tecnologia, Universidade Federal Catalão, Catalão – GO.

As redes adversárias gerativas (GANs) são capazes de produzir dados artificiais que são iguais aos dados reais, e foram aplicados com sucesso a várias tarefas de geração de imagens como uma ferramenta útil para aumento de dados. Neste trabalho, busca-se desenvolver um gerador de sinais baseado em GAN para que aprenda com os sinais de impedância eletromecânica e gere dados representativos realistas e confiáveis. A abordagem proposta será projetada para produzir sinais sintetizados realistas e os sinais gerados podem ser usados como dados reais para outras aplicações no diagnóstico de falha estrutural. A fim de avaliar o desempenho do modelo gerador, um conjunto de avaliações da qualidade das amostras geradas é introduzida na revisão bibliográfica. Finalmente, após a modelagem de dois estudos de caso, foi possível obter uma GAN eficaz na geração de sinais de impedância eletromecânica muito semelhante às amostras experimentais coletadas e inseridas na rede de treinamento e obtendo um comportamento de sinais representativos de compensação de temperatura

Palavras-chaves: Rede Generativa adversária como classificador auxiliar, Monitoramento da Integridade Estrutural, Geração de Dados, Geração de sinais.

ABSTRACT

ROSA, RENAN GARCIA. *Generating Impedance Signatures for Structural Health Monitoring Using Deep Learning*. 2022. 79 f. Master Thesis in Modelling and Optimization – Instituto de Matemática e Tecnologia, Universidade Federal Catalão, Catalão – GO.

Generative adversarial networks (GANs) are capable of producing artificial data that are the same as real data, and have been successfully applied to various imaging tasks as a useful tool for data augmentation. In this work, we seek to develop a GAN-based signal generator to learn from electromechanical impedance signals and generate realistic and reliable representative data. The proposed approach will be designed to produce realistic synthesized signals and the generated signals can be used as real data for other applications in structural failure diagnosis. In order to evaluate the performance of the generator model, a set of quality assessments of the generated samples is introduced in the literature review. Finally, after modeling two case studies, it was possible to obtain an effective GAN in the generation of electromechanical impedance signals very similar to the experimental samples collected and inserted in the training network and obtaining a behavior of representative signals of temperature compensation.

Keywords: Auxiliary classifier generative adversarial networks, structural health monitoring, Data augmentation , Signal generation .

LISTA DE FIGURAS

Figura 2.1 – Modelo de monitoramento baseado em impedância	20
Figura 3.1 – Estrutura histórica da Inteligência Artificial.	25
Figura 3.2 – Neurônio biológico	29
Figura 3.3 – Neurônio matemático	30
Figura 3.4 – Função de ativação	31
Figura 3.5 – Perceptron	32
Figura 3.6 – Multilayer Perceptron	32
Figura 3.7 – Principais funções de ativação	34
Figura 3.8 – ReLU vs Leaky ReLU	35
Figura 3.9 – Modelo de Rede Neural Convolucional	38
Figura 3.10–Neurônio com recorrência	39
Figura 3.11–Neurônio com recorrência	40
Figura 4.1 – Processo de modelagem generativa	41
Figura 4.2 – Exemplo de modelo de rede geradora de uma GAN	43
Figura 4.3 – Exemplo de modelo de rede discriminadora de uma GAN	44
Figura 4.4 – Esquema do funcionamento da rede GAN	44
Figura 4.5 – curva senoidal	45
Figura 4.6 – Esquema do funcionamento da GAN	45
Figura 4.7 – Esquema do funcionamento da GAN	47
Figura 4.8 – Esquema do funcionamento da GAN	47
Figura 4.9 – Esquema do funcionamento da rede GAN	48
Figura 4.10–Esquema do funcionamento da rede GAN	50
Figura 4.11–Método de envelhecimento facial. (a) aproximação do vetor latente para reconstruir a imagem de entrada; (b) mudar a idade condição na entrada do gerador G para realizar o envelhecimento facial.	51
Figura 4.12–Exemplos de reconstrução facial e envelhecimento	51
Figura 4.13–Exemplos de sinal de impedância eletromecânica	52
Figura 5.1 – Função Gaussiana em Python	54
Figura 5.2 – Sinal Unidimensional com 2 picos	54

Figura 5.3 – Arquitetura rede discriminadora	55
Figura 5.4 – Arquitetura rede geradora	56
Figura 5.5 – Câmara de temperatura EPL-4H	57
Figura 5.6 – 4 corpos de prova utilizados no ensaio.	58
Figura 5.7 – Posicionamento dos corpos de prova no interior da câmara.	58
Figura 5.8 – – Assinaturas de impedância do corpo de prova 1	59
Figura 5.9 – Região das assinaturas utilizadas no experimento	59
Figura 5.10 – Banco de Dados Final	60
Figura 5.11 – Arquitetura rede discriminadora	60
Figura 5.12 – Arquitetura rede geradora	61
Figura 6.1 – Sinal unidimensional gerado pela GAN	64
Figura 6.2 – Arquitetura rede geradora	65
Figura 6.3 – Sinal unidimensional gerado pela GAN	66
Figura 6.4 – Sinal de impedância eletromecânica gerado pela GAN	67
Figura 6.5 – Sinal de impedância eletromecânica: sinal gerado vs sinal real	68
Figura 6.6 – Semelhança entre os sinais: sinal compensado temperatura vs sinal gerado	68
Figura 6.7 – Matriz de Confusão para os sinais gerados vs reais	71

LISTA DE TABELAS

Tabela 6.1 – Métrica R2 aplicada aos sinais gerados nas duas versões da GAN para o Estudo de Caso 01	66
Tabela 6.2 – Métrica R2 aplicada aos diferentes tamanhos de amostras aleatórias	69
Tabela 6.3 – Métrica R2 aplicada aos diferentes tamanhos de amostras aleatórias	70
Tabela 6.4 – Quantidade de sinais com variação de temperatura	70

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivos	16
2	FUNDAMENTAÇÃO TEÓRICA SOBRE MONITORAMENTO DA INTEGRIDADE ESTRUTURAL	17
2.1	Monitoramento da Integridade Estrutural	17
2.2	Método da Impedância Eletromecânica	19
3	CONCEITOS FUNDAMENTAIS SOBRE REDES NEURAIS ARTIFICIAIS	24
3.1	Inteligência Artificial	24
3.2	Conceitos de <i>Machine Learning e Deep Learning</i>	25
3.3	Tarefas realizadas em Deep Learning	26
3.4	Conceitos de Redes Neurais Artificiais	27
3.5	O Neurônio Biológico	28
3.6	O Neurônio Matemático	29
3.7	Funções de Ativação	33
3.8	Tipos de aprendizado Neural	35
3.9	Processo de treinamento de RNAs - o uso do <i>backpropagation</i>	36
3.10	Redes Neurais Convolucionais	38
3.11	Rede LSTM - Long Short Term Memory	39
4	FUNDAMENTAÇÃO TEÓRICA SOBRE A REDE GENERATIVA ADVERSÁRIA - GAN	41
4.1	O que é modelagem generativa?	41
4.2	O que são redes adversárias generativas?	42
4.2.1	Rede Geradora	42
4.2.2	Rede discriminadora	43
4.3	Funcionamento de uma GAN	44
4.4	Rede GAN aplicada no conjunto de dados MNIST	48
4.5	Aumento de Dados (Data Augmentation) no Aprendizado Profundo	49
4.6	O uso de Redes Generativas Adversárias para fazer modificações em sinais	50
5	METODOLOGIA	53

5.1	Estudo de Caso 01 - Sinal Unidimensional Básico	53
5.1.1	Função unidimensional	53
5.1.2	Arquitetura da rede Discriminadora	55
5.1.3	Arquitetura da rede Geradora	55
5.2	Estudo de Caso 02 - Sinal de impedância	57
5.2.1	Arquitetura da rede Discriminadora	60
5.2.2	Arquitetura da rede Geradora	61
5.3	Ferramentas Computacionais Necessárias	62
6	RESULTADOS E DISCUSSÕES	64
6.1	Resultados do Banco de Dados do Estudo de Caso 01	64
6.1.1	Conclusões sobre o Estudo de Caso 01	67
6.2	Resultados do Banco de Dados do Estudo de Caso 02	67
6.2.1	Hipótese 01 - Alteração no tamanho dos pontos aleatórios que entram na rede geradora	69
6.2.2	Hipótese 02 - Tamanho do banco de dados de treinamento	69
6.2.3	Hipótese 03 - Quantidade de amostras com variações de temperatura pode afetar na geração final do modelo	70
6.2.4	Hipótese 04 - Um modelo de classificador de sinais consegue diferenciar um sinal real e um gerado?	70
6.2.5	Conclusões sobre o Estudo de Caso 02	72
7	CONCLUSÕES	73
	REFERÊNCIAS	74

Capítulo 1

INTRODUÇÃO

O aprendizado de máquina (do inglês *Machine Learning - ML*), especialmente o aprendizado profundo (do inglês *Deep Learning - DL*), tem sido utilizado em aplicações de monitoramento e diagnóstico de falhas. Comparado com os métodos convencionais que dependem de recursos selecionados manualmente, o diagnóstico de falhas usando DL tem uma poderosa capacidade de aprendizagem de recursos que compreende representações diretamente do sensor de dados original e seleciona automaticamente as características discriminativas que são benéficas para a classificação precisa das condições da estrutura (MA; SUN; CHEN, 2018).

Os avanços em *Deep Learning* fornece maiores oportunidades para melhorar o desempenho em tarefas de diagnóstico de falhas estruturais, contribuindo para o monitoramento da integridade estrutural (do inglês *Structural Health Monitoring - SHM*) ((ZHAO *et al.*, 2019); (SHAO *et al.*, 2018)).

Para métodos baseados em DL, os dados de treinamento são um fator importante que afeta o desempenho da arquitetura profunda. Um modelo de DL contém várias camadas ocultas e o número de parâmetros que precisam ser treinados são enormes (SHAO *et al.*, 2018).

Para um grande modelo, geralmente, é necessário abundantes dados balanceados. No entanto, na prática, as amostras de treinamento entre os diferentes estados de uma estrutura são geralmente desequilibrados (ZHAO *et al.*, 2019).

Por exemplo, para um sistema mecânico em operação, que está em funcionamento normal, a coleta dos dados pelo sensor são suficientes, enquanto a máquina está operando em falha as amostras coletadas são limitadas em comparação com as amostras da estrutura saudável.. Portanto, há um desequilíbrio entre as amostras de treinamento do baseline e amostras de falhas (SHAO *et al.*, 2018).

Com os dados de treinamento limitados, especialmente dados desequilibrados, trei-

nando um modelo profundo para obter respostas precisas, a previsão das condições da máquina é relativamente difícil (MA; SUN; CHEN, 2018).

Uma alternativa para enfrentar o problema de classificação de dados desequilibrada é aplicar as chamadas redes adversárias generativas (GANs). Tais redes trabalham com a geração de dados artificiais, semelhantes aos originais, gerando um conjunto de dados sintéticos. Além disso, a arquitetura generativa modela o processo de geração de dados e, portanto, ajuda a entender a distribuição de dados original, fornecendo uma nova perspectiva para criação de sistemas preditivos em tarefas de diagnóstico de falha estruturais (GULRAJANI *et al.*, 2017; GOODFELLOW *et al.*, 2014).

1.1 Objetivos

O objetivo geral deste trabalho é a modelagem de uma rede neural profunda, do tipo GAN, para a geração de assinaturas de impedância eletromecânica a partir do fornecimento de assinaturas reais da estrutura.

Como objetivos específicos, tem-se a validação de um método de deep learning que vem sendo bem conceituado e utilizado em criação de imagens. E obter a partir do objetivo geral deste trabalho, um volume maior de dados de impedância eletromecânica que representem a estrutura em estudo, a fim de auxiliar no processo de monitoramento da integridade estrutural.

Capítulo 2

Fundamentação Teórica sobre Monitoramento da Integridade Estrutural

Este capítulo apresenta a contextualização e revisão de alguns trabalhos relacionados ao Monitoramento da Integridade Estrutural e ao uso do Método de Impedância Eletromecânica.

2.1 Monitoramento da Integridade Estrutural

O monitoramento da integridade estrutural busca detectar danos estruturais. Mais conhecida como SHM (do inglês *Structural Health Monitoring*), é uma tecnologia que desenvolve e implementa técnicas para detecção, inspeção e monitoramento de danos em estruturas, que estão sujeitas a longos períodos de uso intermitente e vulneráveis a falhas (HALL, 1999).

Na literatura, os sistemas de monitoramento são definidos como a aquisição, validação e análise de dados técnicos do ciclo de vida de uma estrutura, visando um melhor gerenciamento da utilidade do equipamento/estrutura (HALL, 1999).

A motivação para o uso do SHM é tanto científica quanto econômica. Do ponto de vista científico, monitorar e detectar danos estruturais significa conquistar um elevado grau de segurança. Do ponto de vista econômico, sistemas com essa capacidade permitem uma economia significativa em manutenção, substituindo uma manutenção corretiva por uma preditiva contínua, onde a estrutura é analisada em períodos contínuos e comparada com a estrutura em perfeitas condições, garantindo assim a segurança efetiva e a redução do custo da manutenção (MOURA Jr. *et al.*, 2008).

A manutenção preditiva é a avaliação de uma estrutura que está em uso. Tal avalia-

ção visa o acompanhamento da vida útil da estrutura para maximizar sua utilização e evitar manutenções corretivas, que são necessárias quando a(s) falha(s) estão em um estágio avançado (MOURA Jr.; STEFFEN Jr., 2004).

O funcionamento do SHM é admitir que alterações significativas nas propriedades físicas do sistema e/ou variações das propriedades geométricas que podem vir a mudar as condições de contorno e a conectividade do sistema são ocasionadas pela presença de falhas na estrutura (BAPTISTA *et al.*, 2010).

Um termo muito importante para o SHM é o dano, ou seja, uma alteração da estrutura que afeta o seu desempenho atual ou possa afetar o seu desempenho no futuro. A identificação de um dano é feita através da comparação entre dois estados na estrutura, o estado saudável e o estado atual. Sabe-se que alterações na estrutura como: rigidez e a dissipação de energia alteram a resposta dinâmica da estrutura (sinal de impedância) (BAPTISTA *et al.*, 2010).

A vantagem de se utilizar o SHM é ter a possibilidade de identificar o estágio inicial de um dano, o que no caso de tubulações industriais pode salvar muitas vidas, uma vez que uma falha em tubulações de grande escala gera situações críticas de segurança (MOURA Jr. *et al.*, 2008); (BAPTISTA *et al.*, 2010).

Os sistemas de SHM são, basicamente, conjuntos de sensores para aquisição de dados e processadores que avaliam as condições reais da estrutura. Tal avaliação depende do tipo de monitoramento usado. Se for passivo, a avaliação é feita por sensores passivos (tensão, condição ambiental, etc.), que apenas captam os estímulos mecânicos sofridos pela estrutura, mas não interagem com ela. Caso for ativo, a avaliação é feita por sensores ativos, que excitam a estrutura para detectar a presença ou desenvolvimento de uma falha. Interagindo com a estrutura, os sensores ativos encontram sua integridade e sua confiabilidade (MOURA Jr. *et al.*, 2008).

Os métodos não destrutivos, tradicionais para detectar danos, supõem que o aplicador tenha domínio ou conhecimento da região onde está o dano e se é acessível para análises ou não. Essas limitações fizeram surgir os métodos independentes para serem aplicados em estruturas complexas necessitando de pouca interferência humana, ou seja, são métodos econômicos (material de baixo custo) que monitoram continuamente a estrutura e identificam a presença de dano (BAPTISTA *et al.*, 2010).

Além dos métodos comparativos da resposta vibratória, existem alternativas que associam outras áreas de conhecimento como redes neurais, algoritmos genéticos (um método de otimização) e o uso de materiais inteligentes (os sensores/atuadores), que focam na quantificação do dano a partir de diferentes estados de processamento dos sinais de resposta, elevando a classificação do dano (BAPTISTA *et al.*, 2014), (MOURA Jr. *et al.*, 2008).

2.2 Método da Impedância Eletromecânica

O método de Impedância Eletromecânica é um procedimento para detecção de danos em estruturas que, de forma não destrutiva, monitora as mudanças causadas por falhas. Esse monitoramento ocorre por meio de cerâmicas piezelétricas, transdutores de baixo custo que são acoplados na estrutura, de forma que não interfiram no comportamento natural do sistema dinâmico estrutural (CIANG; LEE; BANG, 2008); (STASZEWSKI; LEE; TRAYNOR, 2007); (MOURA Jr. *et al.*, 2008).

A eficácia do método da impedância eletromecânica se baseia na relação entre as propriedades mecânicas da estrutura e a impedância elétrica do transdutor, por isso as variações podem ser monitoradas medindo a impedância elétrica (CAWLEY, 1984).

A impedância mecânica é a divisão da força harmônica aplicada a um sistema mecânico pela velocidade com que ocorre seu deslocamento no mesmo ponto (MASSOUD, 1984).

A impedância elétrica é a oposição que o circuito oferece a corrente alternada. É dividida em duas vertentes: resistência e reatância. A resistência em circuitos de corrente alternada é igual a de circuitos de corrente contínua, ou seja, respeita a lei de Ohm e é expressa pela razão entre a tensão e a intensidade da corrente, não admitindo ângulo de fase (RAJU, 1997). A reatância é dividida em duas partes: indutiva e capacitiva. A reatância indutiva, presente nos circuitos de corrente alternada, varia conforme a frequência, ou seja, quanto maior for a frequência maior será a reatância indutiva (GIBILISCO, 2002). Essa relação está descrita na Equação (2.1).

$$X_L = 2\pi f L, \quad (2.1)$$

onde X_L é reatância indutiva em ohms, f é a frequência da fonte em Hertz e L é a indutância em Henrys da bobina.

A reatância capacitiva é o oposto da indutiva, também varia com a frequência da fonte porém de forma inversa, ou seja, quanto maior for a frequência menor será a reatância capacitiva (GIBILISCO, 2002). Esta relação está descrita na Equação (2.2).

$$X_C = \frac{1}{2\pi f C}, \quad (2.2)$$

onde X_C é a reatância capacitiva em ohms, f é a frequência da fonte em Hertz e C é o capacitor/capacitância em Farad.

A relação entre a impedância mecânica da estrutura e a impedância elétrica do transdutor piezelétrico pode ser obtido por modelos eletromecânicos. Um modelo utiliza uma pastilha PZT quadrada, cuja impedância elétrica é dada pela Equação (2.3) (RAJU, 1997).

(A sigla PZT vem do latim *Plumbum Zirconate Titanate*, que significa Titanato Zirconato de Chumbo.)

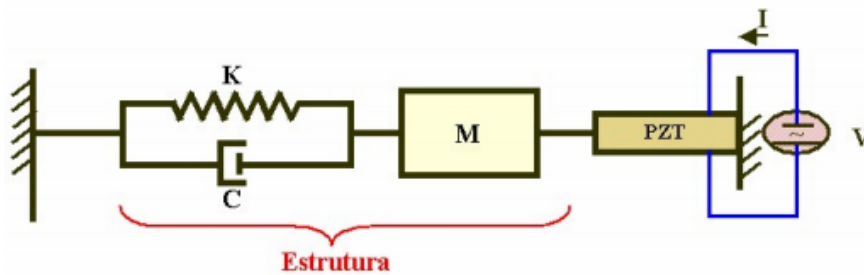
$$Z_E(\omega) = \frac{1}{j\omega C_0} \parallel jZ_T \left(\frac{s_{11}}{d_{31}l} \right)^2 \left[\frac{1}{2} \tanh\left(\frac{kl}{2}\right) - \frac{1}{\sin(kl)} + \frac{Z_S}{j2Z_T} \right] \parallel, \quad (2.3)$$

onde Z_E é a impedância elétrica da pastilha, ω é a frequência angular, C_0 é a capacitância estática para o quadrado de um trecho de l , k é o número da onda, Z_T é a impedância mecânica do transdutor piezelétrico, Z_S é a impedância mecânica da estrutura monitorada, d_{31} é a constante piezelétrica, s_{11} é a conformidade da constante de um campo elétrico, \parallel indica uma conexão paralela e j é a unidade do número imaginário (RAJU, 1997).

Observando a Equação (2.3), é nítida a relação entre a impedância mecânica da estrutura a ser monitorada e a impedância elétrica do transdutor. Então, qualquer modificação na impedância mecânica da estrutura causada por um dano, seja uma trinca, corrosão ou um desgaste, irá apresentar uma variação na impedância elétrica.

Outro modelo eletromecânico, unidimensional, utiliza uma pastilha PZT axial em uma das extremidades da estrutura, como uma viga ilustrada na Figura (1), enquanto a outra está fixada (LIANG, 1994).

Figura 2.1 – Modelo de monitoramento baseado em impedância



Fonte: (MOURA Jr. *et al.*, 2008)

Neste caso considera-se a admitância da pastilha, uma medida que representa a facilidade com que ela conduz corrente alternada (MOURA Jr. *et al.*, 2008). A relação entre a impedância mecânica da pastilha PZT e a da estrutura é descrita na Equação (2.4).

$$Y(\omega) = \frac{1}{V} = i\omega a \left(\epsilon_{33}^{-T} (1 - i\delta) - \frac{Z(\omega)}{Z(\omega) + Z_a(\omega)} d_{3x}^2 \hat{Y}_{xx}^E \right), \quad (2.4)$$

onde V é a voltagem de entrada no atuador PZT, ω é a frequência angular, i é a corrente de saída da pastilha PZT, a é a constante geométrica, Z é a impedância mecânica do PZT, Z_a é a impedância mecânica da estrutura monitorada, d_{3x} é a constante de acoplamento piezelétrico, δ é a perda dielétrica tangencial do PZT, \hat{Y}_{xx}^E indica o módulo de Young e ϵ_{33}^{-T} é a constante dielétrica complexa da pastilha PZT com tensão zero (RAJU, 1997).

A impedância elétrica da pastilha PZT está diretamente relacionada com a impedância mecânica da estrutura, levando a utilizar os sinais de impedância elétrica da pastilha PZT para monitorar a estrutura, representada pela sua impedância mecânica (RAJU, 1997).

Os transdutores piezelétricos são materiais inteligentes de baixo custo frequentemente utilizados no monitoramento de integridade estrutural baseado em impedância, pois sua natureza dielétrica (isolante elétrico que se torna um condutor elétrico caso seu campo elétrico seja quebrado) propicia uma deformação ou tensão quando ocorre a aplicação de um campo elétrico sobre ele. Dessa forma, eles podem ser utilizados como sensores, monitorando a resposta do sistema à perturbações do ambiente, e como atuadores para agir sobre a estrutura e fornecer o comportamento adequado (MAIO, 2011).

Para gerar uma resposta ao campo elétrico exercido sobre os transdutores, o processo de polarização deve ser feito (MOURA Jr. *et al.*, 2008). A polarização é a elevação da temperatura para que os dipolos mudem a orientação da fase sólida, conhecida com temperatura de Curie, e a aplicação de um campo elétrico muito forte sobre a direção que se quer polarizar (MOHEIMANI, 2003).

Então o material é resfriado abaixo da temperatura de Curie e o campo elétrico é mantido, assim o material fica polarizado e com fácil acoplamento eletromecânico, ou seja, o material irá deformar-se quando aplicado um campo elétrico, e o inverso, quando se aplica uma tensão mecânica. (MAIO, 2011).

Quando o material piezelétrico é submetido a um campo elétrico menor do que aquele usado para sua polarização, os dipolos respondem coletivamente, uma expansão macroscópica, gerando o efeito piezelétrico (MOURA Jr. *et al.*, 2008).

O efeito piezelétrico é dividido em dois tipos, o direto e o inverso. O direto acontece quando o material é submetido a uma deformação mecânica, gerando uma deformação dos dipolos, originalmente alinhados, provocando uma diferença de potencial na superfície do material (MOHEIMANI, 2003).

O efeito piezelétrico inverso ocorre quando o material é realmente submetido a um campo elétrico menor do que o utilizado em sua polarização, gerando uma expansão macroscópica dos dipolos (MAIO, 2011).

As equações (2.5) e (2.6) representam o efeito piezelétrico direto e inverso, respectivamente (MOURA Jr. *et al.*, 2008); (MAIO, 2011); (BAPTISTA *et al.*, 2014),

$$D = d\sigma + \epsilon E_1; \quad (2.5)$$

$$S = s\sigma + dE_1; \quad (2.6)$$

onde D é o vetor das deformações (N/m), E_1 é o vetor campo elétrico (V/m), S é o tensor das deformações (N/m^2), σ é o vetor das tensões (N/m^2), d é o tensor da tensão piezelé-

trica (m/V), ϵ indica o tensor dielétrico do material e s indica as propriedades elásticas do material.

A obtenção dos sinais de impedância utiliza os dois efeitos piezelétricos ao mesmo tempo. Logo, para se realizar uma aquisição de sinais, uma diferença de potencial deve ser aplicada, alternando com uma frequência de excitação alta de modo que se obtenha uma resposta dinâmica da estrutura (vibração). Essa resposta é então coletada pelo sensor e transformada em corrente elétrica, possibilitando a coleta da impedância da estrutura a partir da resistência elétrica da pastilha PZT (PALOMINO *et al.*, 2008).

A presença de um dano estrutural é identificada a partir da função resposta em frequência da estrutura. A obtenção desta função é que explica a técnica de monitoramento estrutural baseado em impedância eletromecânica (PARK; INMAN, 2005).

A fixação do transdutor na estrutura gera sua impedância elétrica, que passa a ter picos que correspondem as frequências naturais da mesma. Alterações nesses picos são analisadas para identificar um possível dano estrutural (MARCHI, 2015).

Por ser um material sensível a uma elevada deformação, utiliza-se uma tensão baixa na pastilha PZT. A faixa de frequência utilizada para a análise varia de 10 a 100 kHz, geralmente ocorrendo na casa dos 30 kHz. Para analisar um corpo de prova, a melhor banda de frequência para se analisar é a parte do sinal que existe o maior número de picos (ressonância). Esta escolha da banda de frequência comumente é realizada por meio de tentativa e erro (MOURA Jr. *et al.*, 2008).

A identificação do dano é feita a partir da comparação da impedância elétrica do transdutor medida com a estrutura na condição considerada saudável, com a impedância medida após a estrutura ter sofrido um suposto dano (MOURA Jr. *et al.*, 2008). Essa comparação é feita por métricas de dano, que, na literatura, os mais usados são o RMSD (*Root Mean Square Deviation*) e o CCDM (*Correlation Coefficient Deviation Metric*), e estão representadas nas equações (2.7) e (2.8), respectivamente.

A métrica RMSD trabalha baseado na forma euclidiana (BAPTISTA *et al.*, 2014) e é definida por:

$$RMSD = \sum_{k=\omega_I}^{\omega_F} \sqrt{\frac{[Z_{E,D}(k) - Z_{E,H}(k)]^2}{Z_{E,H}^2(k)}}, \quad (2.7)$$

onde H indica a condição saudável da estrutura, D representa a condição com dano da estrutura, $Z_{E,H}(k)$ é a assinatura de impedância elétrica da estrutura saudável, $Z_{E,D}(k)$ é a assinatura de impedância elétrica da estrutura danificada, k é a frequência, ω_I é a frequência inicial e ω_F é a frequência final.

A métrica CCDM trabalha baseado no coeficiente de correlação (BAPTISTA *et al.*, 2014)

conforme:

$$CCDM = 1 - \frac{\sum_{k=\omega_I}^{\omega_F} [Z_{E,H}(k) - \bar{Z}_{E,H}] [Z_{E,D}(k) - \bar{Z}_{E,D}]}{\sqrt{\sum_{k=\omega_I}^{\omega_F} [Z_{E,H}(k) - \bar{Z}_{E,H}]^2} \sqrt{\sum_{k=\omega_I}^{\omega_F} [Z_{E,D}(k) - \bar{Z}_{E,D}]^2}}, \quad (2.8)$$

onde $\bar{Z}_{E,H}$ e $\bar{Z}_{E,D}$ são as médias das assinaturas de impedância elétrica da estrutura saudável e danificada, respectivamente.

Uma vantagem em utilizar a técnica de impedância eletromecânica é a possibilidade de detectar falhas em seu estado inicial. Mas, uma desvantagem é que o sinal de impedância elétrica é sensível não só pela presença de danos estruturais, mas também aos níveis de ruído nos sinais e de variações das condições ambientais, como diferenças de pressão e temperatura (MARCHI, 2015). Devido a esta desvantagem, vários autores sugerem a utilização apenas da parte real do sinal de impedância, para compensar tais influencias geradas nos sinais (MOURA Jr. *et al.*, 2008); (MARCHI, 2015).

Toda teoria desenvolvida que propõem o uso de impedância eletromecânica no monitoramento da integridade estrutural foi inicialmente proposto por Liang (1994) e em seguida ampliado por vários outros autores, como (CHAUDHRY *et al.*, 1995), (SUN *et al.*, 1995) e (MOURA Jr.; STEFFEN Jr., 2004).

Capítulo 3

Conceitos Fundamentais sobre Redes Neurais Artificiais

Este capítulo é dedicado à apresentação de alguns conceitos fundamentais sobre redes neurais artificiais, incluindo o contexto de aprendizado de máquina, aprendizado profundo e inteligência artificial usados nas formulações das RNAs. A origem das redes neurais também é brevemente descrita, englobando sua idealização e componentes básicos: como funções de ativação, formas de aprendizado e avaliação de modelos neurais e o processo de treinamento mais utilizado no contexto do trabalho proposto.

3.1 Inteligência Artificial

O termo inteligência artificial foi evidenciado pela primeira vez por John McCarthy em 1956, quando ele ocupava o primeiro conferência acadêmica sobre o assunto de que as máquinas eram capazes de resolver problemas que eram difíceis para os humanos resolverem ([HAYKIN, 2001](#)).

Por exemplo, a máquina Enigma foi construída no final da Segunda Guerra Mundial para ser usada em comunicações militares. Alan Turing construiu um sistema de IA que ajudou a quebrar o código do Enigma. Decifrar o código Enigma era uma tarefa muito desafiadora para um ser humano, e poderia demorar semanas para um analista fazer. A máquina de IA foi capaz de decifrar o código em horas ([GOODFELLOW IAN E BENGIO, 2016](#)).

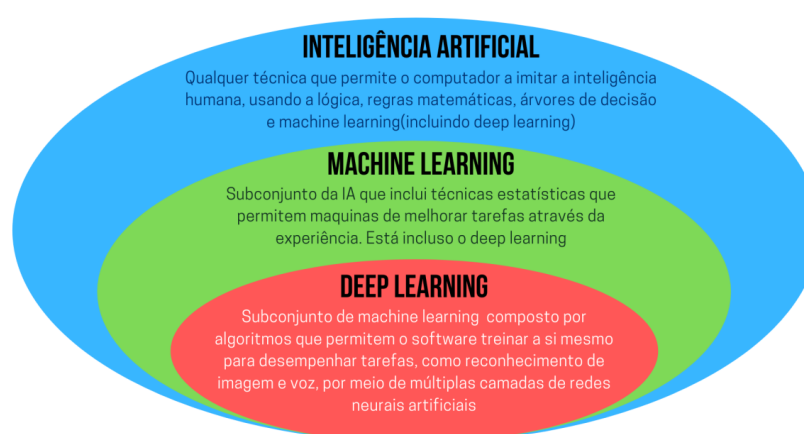
Os computadores têm dificuldade em resolver problemas que são intuitivos para o ser humano, como diferenciar cães e gatos, diferenciar emoções, diferenciar entre caminhão e carro, tomar notas durante um seminário (reconhecimento de fala), ou convertendo notas para outro idioma ([DEKKER; PECHENIZKIY; VLEESHOUWERS, 2009](#)).

Nos primeiros anos da IA, muitos pesquisadores acreditavam que a ela poderia ser alcançada por meio de regras de codificação. Esse tipo de IA é chamado de IA simbólica e

foi útil para resolver problemas lógicos, mas era quase incapaz de resolver problemas complexos, como reconhecimento de imagem, detecção de objetos, segmentação de objetos, tradução de idiomas e tarefas de compreensão de linguagem natural. Abordagens mais recentes de IA, como ML e DL, foram desenvolvidos para resolver esses tipos de problemas (MA; SUN; CHEN, 2018).

Para entender melhor as relações entre IA, *Machine Learning (ML)* e *Deep Learning (DL)*, os círculos concêntricos da Figura 3.1 indica que IA foi a ideia que veio primeiro (o maior), depois a ML (que surgiu mais tarde) e, finalmente, DL, que está impulsionando a explosão da IA.

Figura 3.1 – Estrutura histórica da Inteligência Artificial.



Fonte: o autor (2021).

3.2 Conceitos de *Machine Learning* e *Deep Learning*

Com o aperfeiçoamento dos sistemas de aquisição e armazenamento de dados, ao longo dos anos, o volume de informações coletadas se expandiu consideravelmente ((KUMAR *et al.*, 2021); (BHARTI; NEGI; VERMA, 2019)). A ampliação na quantidade de dados disponíveis, no entanto, acaba dificultando à execução manual de tarefas de abstração, o que, aliado ao recente desenvolvimento tecnológico, incentivou a elaboração de ferramentas automatizadas de análise, as quais constituem hoje o chamado aprendizado de máquina.

A aprendizagem de máquina (ML – do inglês *Machine Learning*) pode ser definida como uma subárea da Inteligência Artificial, voltada para o desenvolvimento de técnicas e/ou algoritmos computacionais, capazes de realizarem a extração de características nos mais variados tipos de problema da atualidade (PROVOST; KOHAVI, 1998).

Em geral, os algoritmos de *Machine Learning* se baseiam na criação de um modelo de inferência programável, em que, a partir de dados amostrais, uma resposta de previsão é obtida (MULLAINATHAN; SPIESS, 2017). Deste modo, tais algoritmos consistem em encontrar,

automaticamente, representações mais úteis dos dados, mapeando uma relação desconhecida dos mesmos em um espaço de hipóteses.

O processo de aprendizado em ML é executado de forma indutiva, onde padrões de respostas são definidos automaticamente, através de experiências já observadas pelo modelo matemático (PROVOST; KOHAVI, 1998). Deste modo, tem-se como principal desafio a formulação de problemas irregulares, nos quais uma lista de regras matemáticas não é facilmente mensurável.

As técnicas de ML, visam extinguir a necessidade de se definir formalmente quais características o classificador deve usar, através de uma representação hierárquica de conceitos (ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012). Assim, o conhecimento é obtido por experiência, possibilitando que o computador faça previsões e/ou classificações de conceitos complexos, por meio da abstração de conceitos mais simples (BURDA; GROSSE; SALAKHUTDINOV, 2015).

As redes neurais artificiais (do inglês = *Neural Network* - RNAs) têm se destacado como uma das principais técnicas de ML, contribuindo ao mesmo tempo para o desenvolvimento de diversas áreas do conhecimento (MENEZES *et al.*, 2009); (BARROS; MORAIS; FERNANDES, 2017); (MHATRE *et al.*, 2017); (SINGH *et al.*, 2017); (ARIA *et al.*, 2020). Sua eficiência se dá principalmente pelo modo como o processamento de dados é executado, em que as operações aplicadas se aproximam particularmente daquelas realizadas pelo cérebro humano.

Quando uma RNA é composta de três ou mais camadas de processamento, essa é caracterizada como uma rede neural profunda, e a sua aplicação compõe o chamado *Deep Learning* (DL) (YU; DAI, 2019).

Na abordagem de deep learning, os níveis de representação dos dados utilizados obedecem a um fluxo hierárquico, isto é, as características de alto nível são obtidas exclusivamente através da composição de níveis inferiores, que se especializam em um determinado tipo de informação (MOHSEN *et al.*, 2018). Para tanto, os modelos de aprendizagem profunda, usualmente, requerem grandes volumes de dados e uma quantidade elevada de camadas ocultas para abstração (MOHSEN *et al.*, 2018). Em consequência, um alto custo computacional também é exigido, com o processamento de dados sendo frequentemente executado na unidade gráfica do computador (YU; DAI, 2019).

3.3 Tarefas realizadas em Deep Learning

No contexto de deep learning, as tarefas realizadas por um modelo neural vão ainda mais longe, sendo descritas em termos de como esse modelo deve processar as instâncias adquiridas para seu treinamento e validação (GOODFELLOW IAN E BENGIO, 2016).

Tais tarefas podem ainda ser comumente qualificadas como atos de predição ou inferência de dados, os quais são avaliados como uma coleção de recursos (*features*), adquiridos de forma vetorial ou matricial, em que cada vetor representa apenas um único estado observado, ou seja, uma amostra (GOODFELLOW IAN E BENGIO, 2016). Em DL, dentre as principais tarefas comumente empregadas, pode-se citar:

- Classificação - em que um modelo (de treinamento geralmente supervisionado) é solicitado a identificar a qual das categorias k pertence uma determinada instância x de dados. Para isso, o modelo é baseado na produção de uma função de mapeamento $f(x)$, que descreve uma relação unívoca dos dados observados em um espaço de hipótese y . A classificação de um conjunto de dados pode ser considerada uma tarefa de inferência, que possui diversos aspectos de avaliação como a classificação binária, a classificação multiclasse, a classificação com entradas ausentes, entre outros.
- Regressão - que ao contrário dos processos de classificação, apresenta um sentido de dependência, ou seja, a relação observada é dada como uma correlação entre os dados de entrada x e uma saída y do modelo (geralmente obtida no espaço R). Tal correlação também é dada por uma função de mapeamento $f(x) \rightarrow y$, porém, como resultado disso, um valor de predição é obtido e não uma classe ou rótulo de inferência.
- Detecção de Anomalias - na qual um conjunto de eventos é avaliado para que correlações entre as variáveis (*features* do conjunto de dados sejam detectadas e combinadas, de forma que variações nessas combinações indiquem a presença de atributos atípicos e/ou anômalos. Essas correlações são usadas para criar um espaço de recursos mais compacto e claro para identificar as diferentes variações dos dados. Para este propósito, ferramentas de extração de recursos são comumente utilizadas em suporte ao modelo neural, como o PCA ou k-means.

3.4 Conceitos de Redes Neurais Artificiais

Uma das áreas de pesquisa em crescimento é a simulação de capacidades cognitivas de um ser humano. Projetam-se máquinas capazes de exibir um comportamento inteligente, como se fossem reações humanas. (RAUBER, 2005)

No cérebro, as entidades básicas são os neurônios, interconectados em redes o que permite a troca de informação entre eles, criando a inteligência biológica. Uma ambição óbvia que surge desses fatos é a tentativa de copiar a estrutura e o funcionamento do cérebro em um ambiente técnico. (RAUBER, 2005)

A Rede Neural Artificial (RNA) é um mecanismo de modelagem poderosa e profunda utilizada em funções não lineares que frequentemente descrevem os sistemas do mundo real. Uma RNA consegue captar as características mais complexas dos dados, o que nem

sempre é possível empregando as técnicas tradicionais de estatística. (BRAGA; CARVALHO; LUDERMIR, 2000)

As RNAs são técnicas computacionais baseadas na estrutura neural de organismos inteligentes e que conseguem habilidades por meio de experiências. Pode ser definida como uma rede grande paralela de componentes interconectados em suas organizações que estão arquitetadas para iterar com objetos do mundo real da mesma maneira que um sistema nervoso biológico o faz (AKKO, 2012).

A RNA busca representar as funções das redes biológicas, implementando em seu comportamento uma semelhança com o sistema biológico básico e sua dinâmica de aprendizado, assim entender melhor como funciona o sistema nervoso (VALENÇA, 2005).

Uma RNA tem duas particularidades: a arquitetura e o algoritmo de aprendizagem. Essa divisão surge naturalmente pelo modo de como a rede é treinada que seria por exemplos de treino. (RAUBER, 2005) São distribuídos em camadas associadas, passando por um processo de treinamento, obtendo a sistemática necessária para executar devidamente o processo desejado dos dados fornecidos. (ZANETTI *et al.*, 2008)

Aptos na resolução de problemas de grande complexidade, as RNAs são modelos utilizados em circunstâncias onde não é possível definir claramente uma lista de regras, ou quando não possuem um padrão de constância. Portanto a RNA especifica padrões e previsões. (NUNES, 2018)

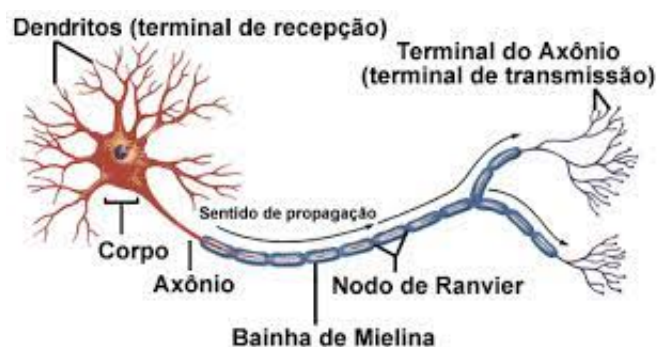
Constituídos de unidades de processamento para determinar funções matemáticas onde podem ser determinadas em conexões (sinapses) que são associadas a pesos (valores atribuídos) e que após o processamento de aprendizagem, armazenam regularmente o conhecimento obtido pela própria rede. (NUNES, 2018)

O funcionamento de RNA é baseado em uma estrutura biológica: o cérebro humano. É composta por um agrupamento de neurônios artificiais que se assemelham à geometria de conexão dos neurônios no cérebro humano, com o propósito de executar uma tarefa com desempenho aprimorado por meio de aprendizagem, treinamento e melhoria contínua. (KOSE UTKU E ARSLAN, 2017)

3.5 O Neurônio Biológico

A unidade básica do processamento é de uma RNA é o neurônio cuja figura representativa pode ser vista na Figura 3.2. Sendo uma célula especializada na transmissão de informações, pois nelas estão introduzidas propriedades de excitabilidade e condução de mensagens nervosas. Nesta podem ser vistas as partes constituintes do neurônio.

Figura 3.2 – Neurônio biológico



Fonte:(HAYKIN, 2001)

O neurônio é constituído por 3 partes principais: a soma ou corpo celular, do qual advêm algumas ramificações chamadas de dendritos, e por uma outra ramificação, porém mais extensa, chamada de axônio. Nas extremidades dos axônios estão os nervos terminais, pelos quais é realizada a transmissão das informações para outros neurônios. Esta transmissão é conhecida como sinapse.

Os dendritos são os receptores, as entradas do neurônio. O axônio é o que transporta o sinal de saída do neurônio. A sinapse que converte um sinal elétrico pré-sináptico em um sinal químico e então de volta em um sinal elétrico pós-sináptico. O corpo celular que é responsável pelo processamento dos sinais de entrada do neurônio. (HAYKIN, 2001)

Quando os valores das entradas atingem um determinado limiar, o neurônio dispara liberando um impulso elétrico que flui do corpo celular para o axônio, que pode estar conectado à entrada de outro neurônio (HAYKIN, 2001).

O corpo celular e os dendritos constituem a parte de entrada e o axônio a superfície de saída do fluxo de informação. Os resultados propagados pelos neurônios são impulsos elétricos. O impulso elétrico é a mensagem que os neurônios transmitem uns aos outros, ou seja, é a disseminação de um estímulo ao longo dos neurônios que pode ser qualquer sinal captado pelos receptores nervosos. A passagem desses sinais não é elétrica, mas química (através da substância serotonina) (HAYKIN, 2001).

Influenciados pelo neurônio biológico, os pesquisadores apresentaram um modelo de neurônio matemático cuja a ideia era reproduzir as redes neurais que formam a inteligência humana em uma Inteligência Artificial. E assim foi desenvolvido o neurônio matemático.

3.6 O Neurônio Matemático

Enquanto o sistema neural biológico pode ser constituído por até bilhões de células nervosas, sendo que cada uma dessas células possui uma complexidade funcional bem

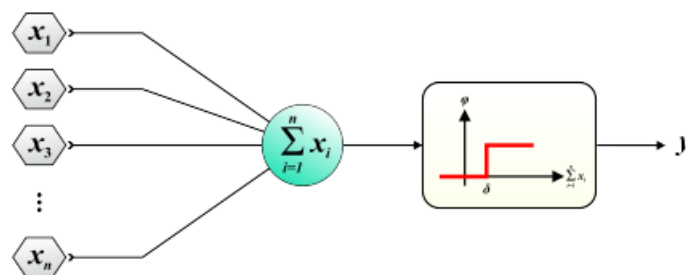
mais elevada, a capacidade de processamento de uma RNA é limitada, visto que a sua modelagem dispõe de dezenas a no máximo milhares de unidades de processamento ((JIANG *et al.*, 2017)).

O primeiro e mais aceito modelo de neurônio artificial que se tem na história, é aquele apresentado no artigo de Warren McCulloch e Walter Pitts de 1943. Os quais descreveram o neurônio artificial como sendo um dispositivo binário, a fim de utilizá-lo para a resolução dos mais variados problemas de circuitos elétricos ((MCCULLOCH WARREN S E PITTS, 1943); (JIANG *et al.*, 2017)).

O modelo MP Neuron ou Discriminador Linear, conforme assim denominado por McCulloch e Pitts na época, realiza em termos simples a agregação de todas as entradas binárias de dados e, a partir do resultado dessa agregação, uma certa decisão é tomada através de uma função de transferência ((MOURA Jr. *et al.*, 2008)).

A saída do neurônio artificial é resultante da ativação ou não ativação da função de transferência, a qual é controlada por um valor de limiar (θ) pré-estabelecido ((HAYKIN, 2001)). A Figura 3.3 apresenta uma esquematização do modelo neural de Warren McCulloch e Walter Pitts de 1943.

Figura 3.3 – Neurônio matemático



Fonte:(REZENDE *et al.*, 2021)

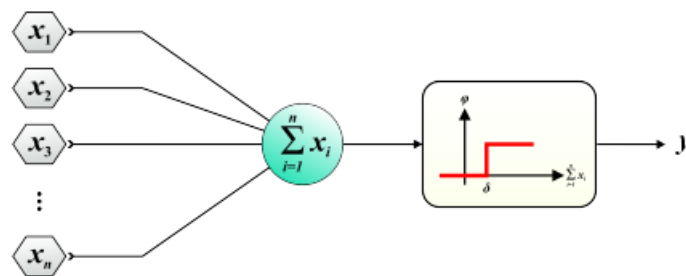
Percebe-se pela Figura 3.3 que o processo de agregação presente no modelo MP Neuron consiste basicamente no somatório de todas as entradas (x_i) de dados, assemelhando-se ao que se ocorre dentro do corpo celular de um neurônio biológico ((MOURA Jr. *et al.*, 2008); (FARIA; CAMSARI; DATTA, 2018)). Quando essa soma atinge um limite superior, o neurônio artificial alcança seu estado de saturação, respondendo conforme a sua função de ativação ().

Os dendritos e axônios são representados matematicamente apenas pelas sinapses, e a intensidade da ligação é representada pelo peso sináptico, simbolizada pela letra w . Quando as entradas, x são enviadas ao neurônio, elas são multiplicadas pelos pesos sinápticos correspondentes, gerando as entradas ponderadas que representa uma das bases matemáticas do funcionamento de uma rede neural artificial, a multiplicação de matrizes:

$$\begin{bmatrix} 1 & x_{11} & \cdots & x_{1d} \\ 1 & x_{21} & \cdots & x_{2d} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & \cdots & x_{nd} \end{bmatrix} X \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_d \end{bmatrix}$$

No modelo de McCulloch e Pitts (REZENDE *et al.*, 2021), a função de ativação usada é do tipo degrau unitário, que pode ser expressa de acordo com a Figura 3.4 ((HAYKIN, 2001)). Nessa função, a ocorrência de um pulso retornará o valor 1 e a sua não ocorrência o valor 0, justificando assim o emprego de tal modelo na abstração de componentes elétricos.

Figura 3.4 – Função de ativação

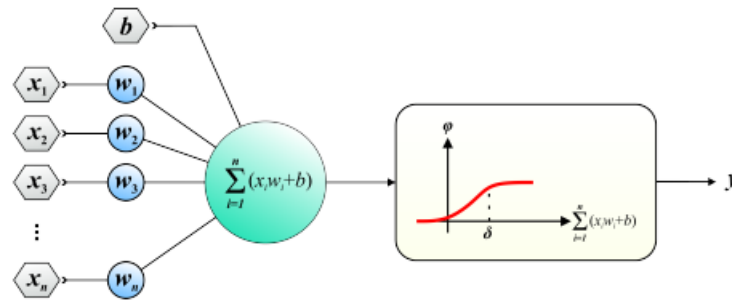


Fonte:(REZENDE *et al.*, 2021)

O Perceptron foi o primeiro tipo de RNA estruturada, isto é, organizada na forma de camada. Tal arquitetura causou um grande impacto na comunidade científica, sendo considerada como o aperfeiçoamento do modelo MP Neuron ((ROSENBLATT, 1960)). O modelo de Rosenblatt acrescentou diversos componentes à estrutura básica do neurônio artificial, possibilitando assim um processo de aprendizado mais próximo ao do sistema neural biológico.

As unidades de processamento contidas no modelo Perceptron são denominadas por *Linear Threshold Unit (LTU)* e possuem as mesmas funcionalidades básicas do modelo MP Neuron. No entanto, nestas são acrescentados pesos de ponderação (w_i) em cada uma das conexões de entrada da rede, além de ser adicionado um dado valor escalar (b) de ajuste, conhecido por bias (HAYKIN, 2001). A Figura 3.5 apresenta o modelo gráfico da rede Perceptron, desenvolvida por Frank Rosenblatt em 1958

Figura 3.5 – Perceptron



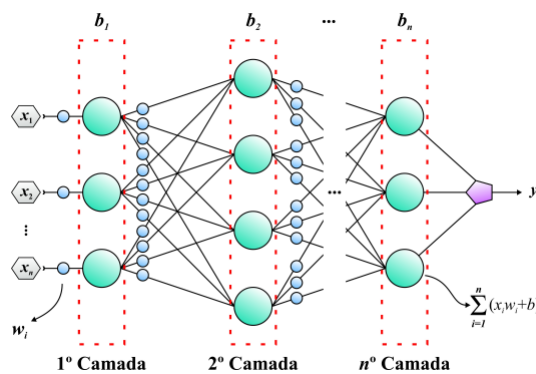
Fonte: (REZENDE *et al.*, 2021)

Convém verificar, que nessa mesma época, outras formulações matemáticas também passaram a ser introduzidas nos estudos de machine learning, como por exemplo: a convolução, a normalização e os estados transientes ((FUKUSHIMA, 1975); (KOHONEN, 1981)). Estas formulações, apoiadas ao avanço tecnológico, deram origem a novas vertentes de RNAs, que posteriormente levaram ao desenvolvimento das atuais redes neurais convolucionais, redes recorrentes e outros tipos de arquitetura profunda ((SCHMIDHUBER; HOCHEITER *et al.*, 1997); (LEE; KIM, 2007)).

No entanto, deve-se ressaltar que o grande impulso no campo das RNAs se deu apenas com o desenvolvimento de novos algoritmos de treinamento mais eficientes, em especial o backpropagation ((RUMELHART; HINTON; WILLIAMS, 1985)). Tais algoritmos de treinamento, possibilitaram não apenas o acoplamento e ajustes dos pesos sinápticos em redes multiníveis, mas também o desenvolvimento de novas arquiteturas neuronais.

A rede Multilayer Perceptron (MLP), conhecida como o aperfeiçoamento do modelo Perceptron, foi a primeira arquitetura neural a ser generalizada para n camadas de neurônios. Esta é formada, essencialmente, por várias unidades menores de processamento, do tipo Perceptron, as quais são distribuídas em paralelo na forma de rede. A Figura 3.6 apresenta um exemplo da arquitetura MLP

Figura 3.6 – Multilayer Perceptron



Fonte: (REZENDE *et al.*, 2021)

As RNAs são estruturas flexíveis, possibilitando assim a sua utilização em diversas aplicações industriais e acadêmicas ((SALAKHUTDINOV, 2015)). Essa flexibilidade proporciona às redes neurais artificiais a capacidade de investigarem diversas hipóteses simultaneamente, tornando-as um recurso de grande valia ((MOURA Jr. *et al.*, 2008)).

As formas de implementação das RNAs são caracterizadas por serem diversificadas, sendo dependentes principalmente do tipo de topologia, do tipo de neurônio artificial (geralmente dado como uma personalização do modelo Perceptron) e das regras de treinamento ou aprendizado utilizadas ((RUMELHART; HINTON; WILLIAMS, 1985); (BRESFELEAN *et al.*, 2008)).

3.7 Funções de Ativação

A função de ativação de um neurônio artificial é basicamente uma transformação do somatório de todas as entradas de dados ((NWANKPA *et al.*, 2018)), as quais são ponderadas pelos valores dos pesos sinápticos e podem ser provenientes de outros neurônios ou da própria entrada da rede neural.

O neurônio realiza os produtos gerando um único resultado chamado de função de combinação. Este resultado é apresentado a uma função de ativação (função de transferência) que tem, dentre outras, a função de evitar o acréscimo progressivo dos valores de saída ao longo das camadas da rede, visto que tais funções possuem valores máximos e mínimos contidos em intervalos determinados ((HAYKIN, 2001)). A função de ativação é a que processa o sinal $u(k)$ para produzir a saída final do neurônio, $y(k)$.

Segundo ((HAYKIN, 2001)), existem seis tipos básicos de função de ativação utilizados em RNAs: função de limiar, função linear por partes, função tangente hiperbólica, função degrau unitário, função sigmoide e função Softmax.

Todas as funções supracitadas bem como suas respectivas derivadas de primeira ordem são apresentadas na Figura 3.7, onde $u = \sum_{i=1}^n x_i w_i + b$ e δ é o valor do limiar de ativação.

Figura 3.7 – Principais funções de ativação

	Nome da Função	$\varphi(u)$	$\varphi'(u)$
Linear	Degrau Unitário	$\varphi(u) = \begin{cases} 1, & \text{se } u \geq \delta \\ 0, & \text{se } u < \delta \end{cases}$	$\varphi'(u) = 0$
	Degrau Bipolar	$\varphi(u) = \begin{cases} 1, & \text{se } u \geq \delta \\ -1, & \text{se } u < \delta \end{cases}$	$\varphi'(u) = 0$
	Linear	$\varphi(u) = au$	$\varphi'(u) = a$
Não Linear	Sigmoidal	$\varphi(u) = \frac{1}{1+e^{-u}}$	$\varphi'(u) = \varphi(u)(1 - \varphi(u))$
	TanH	$\varphi(u) = \frac{2}{1+e^{-u}}(2u) - 1$	$\varphi'(u) = 1 - \tanh^2(u)$
	Softmax	$\varphi(u) = \frac{e^u}{\sum_{k=1}^K e^{u_k}}$	$\varphi'(u) = \begin{cases} \varphi_i(1 - \varphi_k), & \text{se } i = k \\ -\varphi_i\varphi_k, & \text{se } i \neq k \end{cases}$
	ReLU	$\varphi(u) = \max\{0, u\}$	$\varphi'(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases}$
	ELU	$\varphi(u, \alpha) = \begin{cases} u, & \text{se } u \geq 0 \\ \alpha(e^u - 1), & \text{se } u < 0 \end{cases}$	$\varphi'(u, \alpha) = \begin{cases} 1, & \text{se } u \geq 0 \\ \varphi(u, \alpha) + \alpha, & \text{se } u < 0 \end{cases}$
	Leaky ReLU	$\varphi(u, \alpha) = \max\{\alpha u, u\}$	$\varphi'(u, \alpha) = \begin{cases} 1, & \text{se } u \geq 0 \\ \alpha, & \text{se } u < 0 \end{cases}$

Fonte:(REZENDE *et al.*, 2021)

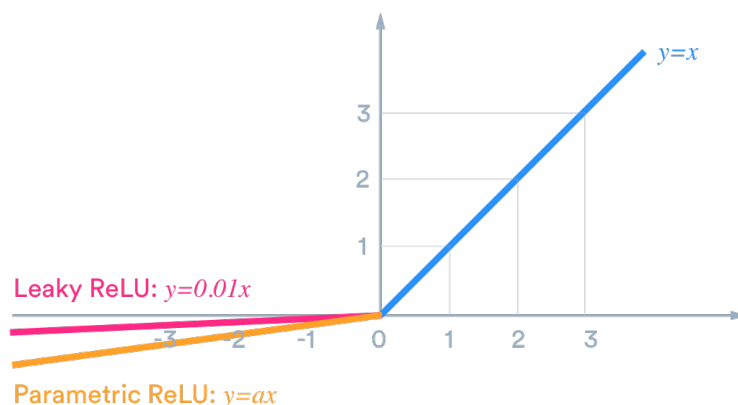
A maioria das aplicações de Deep Learning atualmente faz uso de ReLU em vez de funções de Ativação Logística para Visão Computacional, Reconhecimento de Fala, Processamento de Linguagem Natural e Redes Neurais Profundas, etc (BARELLA *et al.*, 2021). Algumas das variantes do ReLU incluem: Softplus (SmoothReLU), Noisy ReLU, Leaky ReLU, Parametric ReLU e ExponentialReLU (ELU)

O ReLU é uma unidade linear retificada (uma unidade que emprega o retificador também é chamada de unidade linear retificada ReLU) tem saída 0 se a entrada for menor que 0 e saída bruta caso contrário. Ou seja, se a entrada for maior que 0, a saída será igual à entrada. O funcionamento do ReLU está mais próximo do funcionamento dos nossos neurônios biológicos (REZENDE *et al.*, 2021).

ReLU é não linear e tem a vantagem de não ter erros de retropropagação ao contrário da função sigmoid, também para redes neurais maiores, a velocidade de construção de modelos baseados em ReLU é muito rápida em oposição ao uso de Sigmoids (GOODFELLOW IAN E BENGIO, 2016).

Uma desvantagem no ReLU é o problema *Dying ReLU*, onde alguns neurônios ReLU morrem essencialmente para todas as entradas e permanecem inativos, não importa qual entrada seja fornecida, aqui não há fluxos de gradiente e se um grande número de neurônios mortos estiver em uma rede neural, seu desempenho é afetado, isso pode ser corrigido usando o que é chamado de *Leaky ReLU*, onde a inclinação é alterada para a esquerda de $x = 0$, causando um vazamento e estendendo o alcance de ReLU (REZENDE *et al.*, 2021).

Figura 3.8 – ReLU vs Leaky ReLU



Fonte:(REZENDE *et al.*, 2021)

Com *Leaky ReLU* há uma pequena inclinação negativa, então, em vez de não disparar para grandes gradientes, os neurônios produzem algum valor e isso torna a camada muito mais otimizada também (GOODFELLOW IAN E BENGIO, 2016).

3.8 Tipos de aprendizado Neural

Uma RNA adquire conhecimento e melhora seu desempenho em relação a seu ambiente por meio de um processo iterativo de ajuste aplicados a seus pesos sinápticos, o treinamento. A rede se assume mais precisa sobre seu ambiente após cada iteração do processo de aprendizagem ((HAYKIN, 2001))

Existem diversos conceitos de aprendizado de máquina que são aplicados as redes neurais artificiais, sendo cada um deles sujeitos ao modo de como os pesos sinápticos são atualizados e quais são as informações disponíveis para treinamento ((GODEFROID; CÂNDIDO; MORAIS, 2006); (WANG *et al.*, 2019)). No entanto, dentre os principais conceitos de aprendizado, que são direcionados às aplicações de RNAs, se encontram: a aprendizagem supervisionada, a aprendizagem não-supervisionada e a aprendizagem por reforço ((MHA-TRE *et al.*, 2017)).

A maioria das RNAs tem usufruído do treinamento supervisionado. Assim, a saída atual da rede neural é relativa a saída pretendida. Os valores iniciais dos pesos são aleatórios, sendo ajustados de acordo com algoritmo de aprendizagem a cada iteração ((PATTERSON, 1998)).

O peso sináptico é ajustado de acordo com o valor esperado e do sinal atual de saída. O método de aprendizado diminui os erros dos dados em processamento. Essa minimização de erros modifica os pesos até o momento em que a rede alcance uma precisão ((PATTERSON, 1998)).

O aprendizado supervisionado ou aprendizagem com professor é caracterizado pela

presença de um tutor. O estado do ambiente é representado por um vetor que é aplicado à entrada da RNA e ao professor com conhecimento sobre o ambiente o qual é reproduzido por um conjunto de exemplos de entrada e saída fornecendo assim a resposta desejada ((HAYKIN, 2001)).

Na aprendizagem não-supervisionada ou auto-organizada são dados requisitos para executar uma representação que a rede deve aprender, visto que a rede tenha se ajustado às regularidades estatísticas dos dados de entrada, ela desenvolve a capacidade de gerar representações internas para compilar as características da entrada, e assim, cria automaticamente novas classes ((BECKER, 1991)).

No aprendizado não-supervisionado nenhuma informação prévia dos dados de treinamento é utilizada, isto é, os dados de entrada são apenas agrupados tomando-se por base as suas características semelhantes. Para tanto, a arquitetura e o processo de treinamento da rede neural caracterizam-se, respectivamente, por utilizar neurônios classificadores e usualmente ser competitivo e inibitório ((KOHONEN TEUVO E KASKI, 1997); (GOODFELLOW IAN E BENGIO, 2016)).

Já no aprendizado por reforço, o ajuste dos pesos sinápticos é executado com relação às limitações ou informações que a rede possui sobre seu ambiente externo e/ou ações que efetua. Isto é, em uma primeira instância, à rede neural considerada como agente avaliador, executa decisões totalmente aleatórias e, com o passar do tempo suas escolhas (políticas) são ponderadas com base nos resultados de suas ações anteriores ((GOODFELLOW IAN E BENGIO, 2016)).

Algumas abordagens mais simples, como a redução da taxa de aprendizado, podem ser aplicadas para se obter uma melhor resposta por meio do aprendizado contínuo, porém, o uso do aprendizado em lote, quando as características do conjunto de dados são inicialmente desconhecidas, pode gerar resultados mais satisfatórios. Entretanto, o tamanho do lote (*batch_size*) utilizado no treinamento também pode causar perda de acurácia nos modelos neurais, dada sua influência na convergência da rede durante esta etapa ((DUCHI; SINGER, 2009)). Apesar disso, ressalta-se que o tipo de aprendizado empregado é dependente da topologia de rede neural utilizada, sendo dessa forma muito restrita ao tipo de problema a ser tratado.

3.9 Processo de treinamento de RNAs - o uso do *backpropagation*

De acordo com ((HAYKIN, 2001)), em problemas de classificação e regressão de dados, o ajuste dos parâmetros locais de uma RNA é comumente realizado utilizando um algoritmo de retropropagação do erro, também conhecido como *backpropagation* ((HAYKIN, 2001)).

O algoritmo *backpropagation* é responsável por verificar o nível de desempenho da rede neural e, se esse nível estiver abaixo das expectativas, o mesmo ajustará seus parâmetros locais (pesos sinápticos e bias), a fim de melhorar a assertividade do modelo ((HAFEMANN; OLIVEIRA; CAVALIN, 2014)).

No *backpropagation*, o processo de aprendizado ocorre em duas etapas ((RUMELHART; HINTON; WILLIAMS, 1985)). Na fase *forward* o conjunto de dados de entrada são apresentados à primeira camada de RNA, que determina seus sinais de saída e envia os valores para a camada seguinte e assim sucessivamente até a camada de saída calcular as saídas da rede, as quais são comparadas com as saídas desejadas. Já a fase *backward* faz o caminho inverso. Ao começar da camada de saída até a de entrada e os pesos dos neurônios são ajustados de forma que seus erros são diminuídos (os erros das camadas ocultas são calculados utilizando o erro dos neurônios da camada seguinte equilibrado pelo peso da conexão entre eles). Este procedimento é utilizado até atingir o critério de parada.

Após a fase *forward*, a resposta do modelo neural, geralmente dada como um vetor de probabilidades, é então comparada à algum padrão esperado, com uma função de perda sendo definida no final do processo, seja o RMSE, o Erro Médio Absoluto ou a entropia cruzada. ((RUMELHART; HINTON; WILLIAMS, 1985)).

De forma simplificada pode se reunir as etapas do treinamento de uma rede usando *backpropagation* da seguinte forma:

- Inicialização dos pesos, definição dos valores da taxa de aprendizado, momento do critério de parada;
- Inserção das amostras (dados) de treinamento à rede. Para cada amostra, é realizada a série de etapas do treinamento;
- Propagação: supondo que cada amostra de treinamento seja representada por $(x(n), d(n))$, com vetor $x(n)$ sendo apresentado na entrada da rede e o vetor da resposta desejada $d(n)$ apresentada na camada de saída da rede, são calculadas as saídas locais induzidas e os sinais funcionais da rede prosseguindo camada por camada;
- Retropropagação: cálculo dos gradientes locais (em cada neurônio). Ajuste dos pesos sinápticos da rede na(s) camada (s) oculta (s) de acordo com a regra delta generalizada;
- Iteração: iterar as computações de propagação e na retropropagação apresentando novas coletas de amostras de treinamento para rede até que o critério de parada determinado seja alcançado.

3.10 Redes Neurais Convolucionais

As Redes Neurais de Convolução (CNN) são arquiteturas especiais de aprendizagem profunda, voltadas ao tratamento e ao desenvolvimento de um conjunto específico de problemas da visão computacional, em especial à classificação de imagens e áudios (CHOLLET *et al.*, 2015).

Sua formulação neurobiológica teve inspiração na organização e no processo como o córtex visual humano age na abstração de características de um determinado objeto. Para tanto, as CNNs possuem uma estruturação hierarquizada de suas camadas de neurônios das quais em ao menos uma delas é aplicado a operação de convolução (GOODFELLOW IAN E BENGIO, 2016).

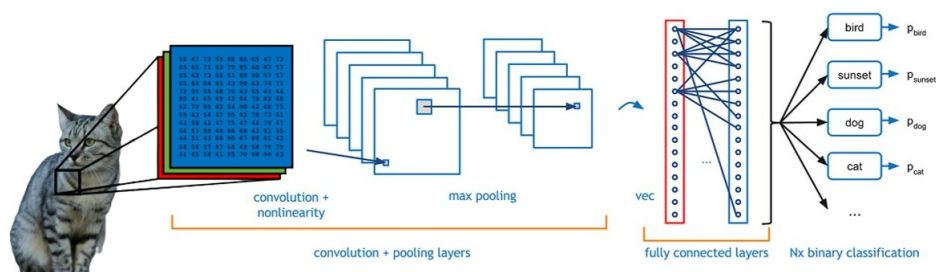
Ainda, para uma CNN, uma imagem computadorizada (ou outro tipo de dado vetorial) pode ser representada por meio de tensores geométricos, os quais descrevem as relações lineares entre suas coordenadas, valores escalares ou outros tensores da mesma imagem. Essas relações podem ser interpretadas pelas camadas de filtros, chamadas convolucionais, de uma CNN (GOODFELLOW IAN E BENGIO, 2016).

Camadas convolucionais, além de empregarem convoluções de seus pesos sinápticos (pelo uso de filtros de amostragem) apresentam apenas uma conectividade local entre o seu conjunto de neurônios artificiais e os núcleos de suas entradas.

As camadas de *pooling*, são camadas de agrupamento, apresentando uma conectividade superior as convolucionais, e servem como pontes detalhadas ao modelo de classificação da rede. Esse, por si, é responsável por catalogar e definir as possíveis saídas da rede, sendo que são definidas em classes ou probabilidades de pertencerem a um determinado grupo ((HAN; PEI; KAMBER, 2011).

A Figura 3.8 apresenta a topologia característica de um modelo de rede neural convolucional.

Figura 3.9 – Modelo de Rede Neural Convolucional



Fonte:(HAYKIN, 2001)

A possibilidade de retratação matemática por meio de tensores, facilitou amplamente a implementação das redes de convolução, bem como otimizou o seu custo computacio-

nal. Tal acontecimento decorre-se também ao desenvolvimento tecnológico das unidades de processamento gráfico, sendo essa disponibilidade computacional o principal contraste das atuais CNNs com os modelos implementados na sua época de criação (FARIA; CAMSARI; DATTA, 2018).

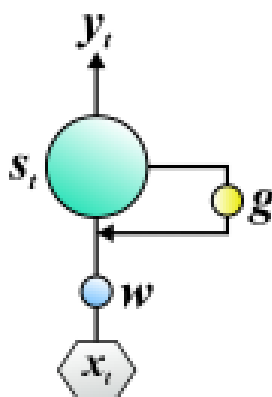
3.11 Rede LSTM - Long Short Term Memory

Em geral, para se caracterizar um sistema dinâmico, a análise de seu comportamento nas condições atuais não é suficiente, pois seu estado anterior geralmente influencia o seu estado atual (SILVA; SPATTI; FLAUZINO, 2010). Nesse contexto, as arquiteturas neurais do tipo feedforward nem sempre se tornam aplicáveis, pois elas possuem apenas uma relação direta entre a entrada corrente e a sua respectiva saída. Assim, com base nesse princípio, é necessário adicionar às RNAs algumas alterações em sua topologia, que lhes permitem armazenar uma relação temporal dos seus estados anteriores para a sua avaliação atual (BISHOP, 1995).

Logo, surgiram as Redes Neurais Recorrentes (RNNs), que são arquiteturas de aprendizagem profunda, que incluem certos mecanismos de memória através do feedback de suas unidades internas de processamento (ZHANG, 2003). Esse recurso, de acordo com Bispo(2018), oferece às RNNs uma melhor capacidade de reconhecer e prever diferentes tipos de padrões em longas cadeias de dados sequenciais (ZHAO *et al.*, 2019).

Para melhor elucidar o fluxo de informações em uma RNN, a Figura 3.9 demonstra o processo de recorrência em um de seus neurônios característicos (BENGIO; SIMARD; FRASCONI, 1994). Nesta imagem, o neurônio artificial processa a entrada $x(t)$ para gerar como resposta o vetor y_t (também conhecido como estado da rede no tempo t).

Figura 3.10 – Neurônio com recorrência



Fonte:(BENGIO; SIMARD; FRASCONI, 1994)

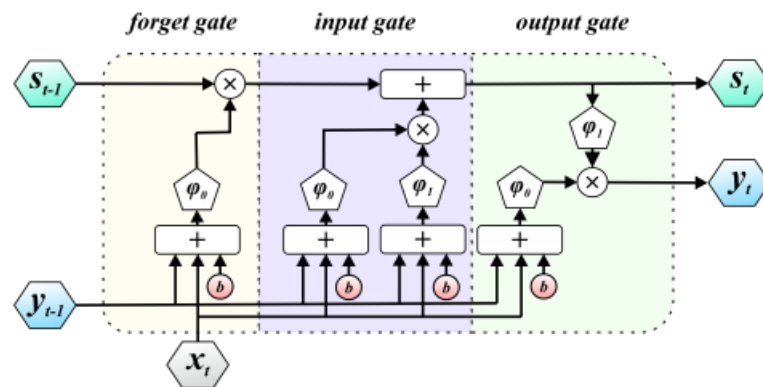
Assim, as redes neurais recorrentes são compostas de neurônios recorrentes que realizam, a cada iteração do processo de treinamento, uma retroalimentação de suas informa-

ções abstraídas para outras partes do modelo de rede neural (ZHANG, 2003).

Portanto, se o processo de treinamento de uma RNA for considerado como um fluxo de memória a longo prazo, a recorrência no escopo das RNNs lhes permitirá operar em problemas muito mais complexos, aplicando conjuntamente recursos de longo e a curto prazo (HOCHREITER; SCHMIDHUBER, 1997).

Em geral, os neurônios (também conhecidos como células ou blocos de memória) de uma rede LSTM são compostos individualmente por 3 portões, que são eles: o portão de esquecimento (*forget gate*), o portão de entrada (*input gate*) e o portão de saída (*output gate*) (YANG *et al.*, 2019). Todos eles possuem uma função de ativação para controlar as informações que fluem em cada sentido da célula (SANTOS, 2012), possibilitando sua esquematização de acordo com a Figura 3.10.

Figura 3.11 – Neurônio com recorrência



Fonte: (HAYKIN, 2001)

A arquitetura LSTM é um modelo de rede neural recorrente desenvolvido em 1997 pelos cientistas da computação Sepp Hochreiter e Jürgen Schmidhuber, a fim de resolver o problema do desaparecimento do gradiente descendente, quando um grande volume de dados sequenciais é avaliado (HOCHREITER; SCHMIDHUBER, 1997). Para isso, o fluxo de erro calculado durante o processamento da rede é mantido constante através da inclusão de unidades especiais em seus neurônios característicos. Essas unidades são popularmente conhecidas como portões (**gates**) (YANG *et al.*, 2019).

Capítulo 4

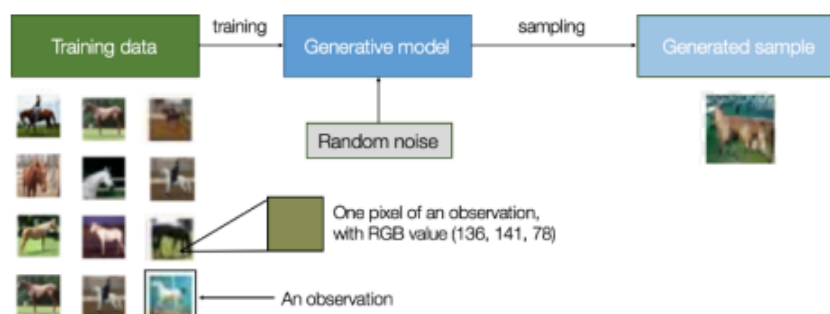
Fundamentação Teórica sobre a Rede Generativa Adversária - GAN

Este capítulo é uma introdução geral ao campo da modelagem generativa. É importante ver o que significa dizer que um modelo é generativo e como ele difere da modelagem discriminativa mais amplamente estudada.

4.1 O que é modelagem generativa?

Suponha que em um conjunto de dados contendo imagens de cavalos, deseja-se construir um modelo que pode gerar uma nova imagem de um cavalo que nunca existiu, mas ainda parece real porque o modelo aprendeu as regras gerais que governam a aparência de um cavalo. Este é o tipo de problema que pode ser resolvido usando modelagem generativa. Um resumo de um processo típico de modelagem generativa é mostrado na Figura 4.1. (GOODFELLOW *et al.*, 2014).

Figura 4.1 – Processo de modelagem generativa



Fonte:(GOODFELLOW *et al.*, 2014)

Primeiro, exigimos um conjunto de dados que consiste em muitos exemplos da entidade que estamos tentando gerar. Isso é conhecido como dados de treinamento, e um

desses pontos de dados é chamado de observação (GOODFELLOW *et al.*, 2014)

Cada observação consiste em muitos recursos para um problema de geração de imagem, os recursos são geralmente os valores de pixel individuais. O objetivo é construir um modelo que possa gerar novos conjuntos de recursos que parecem ter sido criados usando o mesmo conjunto de regras dos dados originais (GOODFELLOW *et al.*, 2014).

4.2 O que são redes adversárias generativas?

Generative Adversarial Networks, ou GANs, são modelos generativos baseados em aprendizado profundo. Geralmente, GANs são uma arquitetura para treinar um modelo generativo, e é mais comum para usar modelos de aprendizado profundo nesta arquitetura.

A arquitetura GAN foi descrita pela primeira vez no artigo de (GOODFELLOW *et al.*, 2014) intitulado *Generative Adversarial Networks*. Essa arquitetura envolve dois submodelos: um modelo gerador para gerar novos exemplos e um modelo discriminador para classificar se os exemplos gerados são reais (do domínio) ou falso (gerado pelo modelo do gerador).

- Gerador: Modelo que é usado para gerar novos exemplos plausíveis do problema.
- Discriminador: Modelo que é usado para classificar exemplos como reais (conjunto real) ou falso (gerado).

Redes adversárias gerativas são baseadas em um cenário teórico em que a rede de geradores deve competir com um adversário. A rede do gerador produz amostras diretamente. Seu adversário, a rede discriminadora, tenta distinguir entre amostras retiradas dos dados de treinamento e amostras retiradas do gerador.

4.2.1 Rede Geradora

O modelo gerador utiliza um vetor aleatório de comprimento fixo como entrada e gera uma amostra no conjunto, como uma imagem. Um vetor é desenhado aleatoriamente a partir de uma distribuição Gaussiana e é usado para semear uma fonte de ruído para o processo generativo. (GOODFELLOW *et al.*, 2014).

Após o treinamento, pontos neste vetor multidimensional no espaço corresponderá a pontos no conjunto do problema, formando uma representação comprimida da distribuição de dados. Esse espaço vetorial é conhecido como espaço latente ou espaço vetorial composta por variáveis latentes ou variáveis ocultas, que são importantes para um conjunto, mas não são diretamente observáveis (RADFORD; METZ; CHINTALA, 2015).

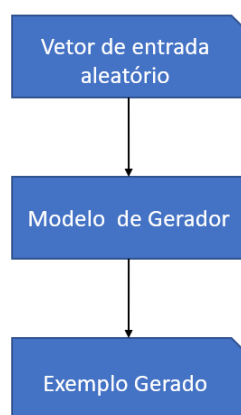
Refere-se a variáveis latentes, ou espaço latente, como uma projeção ou compressão de uma distribuição de dados. Ou seja, um espaço latente fornece uma compressão ou con-

ceitos de alto nível dos dados brutos observados, como a distribuição dos dados de entrada (GOODFELLOW *et al.*, 2014).

No caso de GANs, o modelo do gerador aplica significado a pontos em um espaço latente escolhido, de modo que novos pontos retirados do espaço latente possam ser fornecidos ao modelo do gerador como entrada e usados para gerar novos e diferentes exemplos de saída (GOODFELLOW *et al.*, 2014; RADFORD; METZ; CHINTALA, 2015).

Após o treinamento, o modelo do gerador é mantido e usado para gerar novas amostras. A Figura 4.2 ilustra um modelo de rede gerada em uma GAN

Figura 4.2 – Exemplo de modelo de rede geradora de uma GAN



Fonte: o autor, adaptado de (GOODFELLOW *et al.*, 2014)

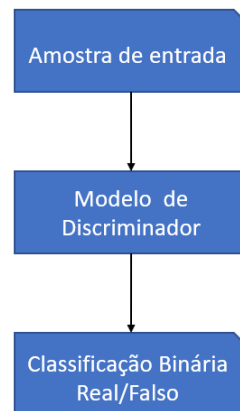
Após algumas iterações, a tendência é que o modelo gerador comece a produzir imagens próximas às imagens reais (GOODFELLOW *et al.*, 2014).

4.2.2 Rede discriminadora

O modelo discriminador pega um exemplo do domínio do problema como entrada (real ou gerada) e prevê um rótulo de classe binária de real ou falso (gerado). O verdadeiro exemplo vem do conjunto de dados de treinamento. Os exemplos gerados são produzidos pelo modelo do gerador (RADFORD; METZ; CHINTALA, 2015).

Após o processo de treinamento, o modelo discriminador é descartado (GOODFELLOW *et al.*, 2014). Às vezes, o gerador pode ser reaproveitado, uma vez que aprendeu a extrair efetivamente recursos de exemplos no domínio do problema. A Figura 4.3 ilustra um modelo de rede discriminadora de uma GAN.

Figura 4.3 – Exemplo de modelo de rede discriminadora de uma GAN

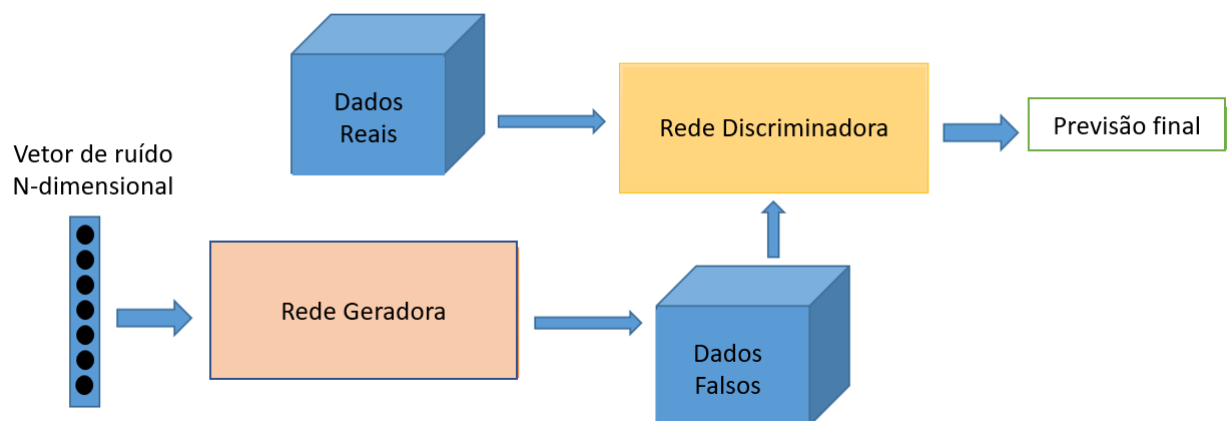


Fonte: o autor, adaptado de (GOODFELLOW *et al.*, 2014)

4.3 Funcionamento de uma GAN

Uma rede geradora gera novas instâncias de dados, enquanto a outra, o discriminador, avalia sua autenticidade; ou seja, o discriminador decide se cada instância de dados que ele analisa pertence ou não ao conjunto de dados de treinamento real (RADFORD; METZ; CHINTALA, 2015). A Figura 4.4 demonstra isso.

Figura 4.4 – Esquema do funcionamento da rede GAN



Fonte:(RADFORD; METZ; CHINTALA, 2015)

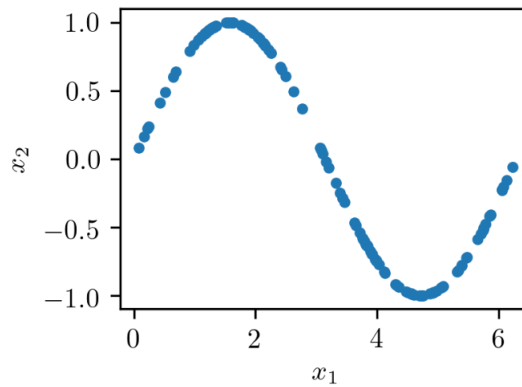
Aqui estão as etapas de uma GAN:

- O gerador considera números aleatórios e retorna uma imagem.
- Essa imagem gerada é inserida no discriminador ao lado de um fluxo de imagens tiradas do conjunto de dados real e verdadeiro.

- O discriminador obtém imagens reais e falsas e retorna probabilidades, um número entre 0 e 1, com 1 representando uma previsão de imagem autêntica e 0 representando previsão de imagens falsas (geradas pela rede generativa).

Para entender como funciona o treinamento GAN, considere um exemplo com um conjunto de dados composto por amostras bidimensionais (x_1, x_2) , com x_1 no intervalo de 0 a 2π e $x_2 = \sin(x_1)$, conforme ilustrado na Figura 4.5:

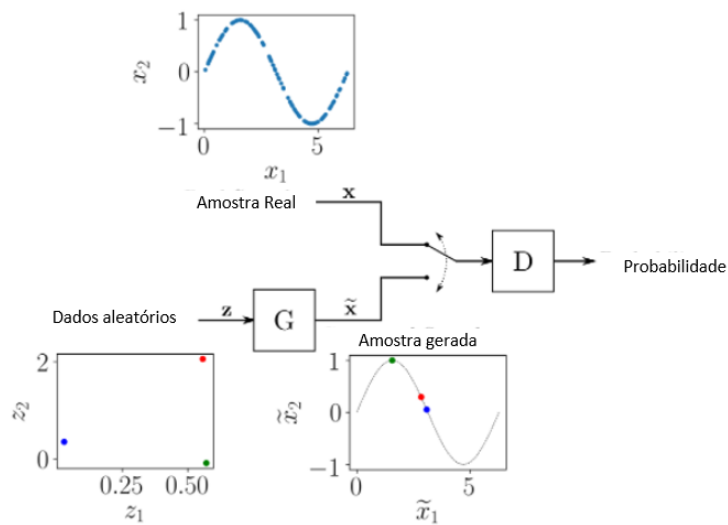
Figura 4.5 – curva senoidal



Fonte: o autor

Este conjunto de dados consiste em pontos (x_1, x_2) localizados sobre uma curva senoidal, tendo uma distribuição muito particular. A estrutura geral de um GAN para gerar pares $(\tilde{x}_1, \tilde{x}_2)$ semelhantes às amostras do conjunto de dados é mostrada na figura 4.6:

Figura 4.6 – Esquema do funcionamento da GAN



Fonte: o autor, adaptado de (Renato Candido, 2020)

O gerador G é alimentado com dados aleatórios de um espaço latente, e sua função é gerar dados semelhantes às amostras reais. Neste exemplo, tem-se um espaço latente bidimensional, de modo que o gerador é alimentado com pares aleatórios (z_1, z_2) e é necessário transformá-los para que se pareçam com as amostras reais (Renato Candido, 2020).

A estrutura da rede neural G pode ser arbitrária, permitindo que você use redes neurais como uma MLP, uma rede neural convolucional (CNN) ou qualquer outra estrutura, desde que as dimensões de entrada e saída correspondam às dimensões do espaço latente e dos dados reais (GOODFELLOW *et al.*, 2014).

O discriminador D é abastecida com amostras reais a partir do conjunto de dados de treino ou amostras geradas fornecidas por pelo gerador G . Sua função é estimar a probabilidade de que a entrada pertença ao conjunto de dados real. O treinamento é executado de forma que D produza 1 quando for alimentado com uma amostra real e 0 quando for alimentado com uma amostra gerada (GOODFELLOW *et al.*, 2014).

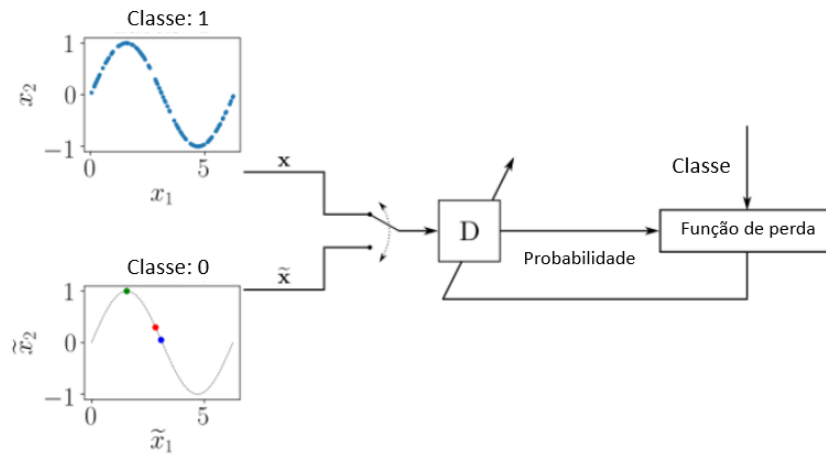
Como acontece com G , pode se escolher uma estrutura de rede neural arbitrária para D , desde que respeite as dimensões de entrada e saída necessárias (RADFORD; METZ; CHINTALA, 2015). Neste exemplo, a entrada é bidimensional. Para um discriminador binário, a saída pode ser um escalar variando de 0 a 1.

O processo de treinamento GAN consiste em um jogo minimax de dois jogadores em que D é adaptado para minimizar o erro de discriminação entre as amostras reais e geradas, e G é adaptado para maximizar a probabilidade de D cometer um erro (GOODFELLOW *et al.*, 2014).

Embora o conjunto de dados contendo os dados reais não esteja rotulado, os processos de treinamento para D e G são realizados de forma supervisionada. A cada etapa do treinamento, D e G têm seus parâmetros atualizados. De fato, na proposta original do GAN, os parâmetros de D são atualizados k vezes, enquanto os parâmetros de G são atualizados apenas uma vez para cada etapa de treinamento. No entanto, para tornar o treinamento mais simples, você pode considerar k igual a 1 (RADFORD; METZ; CHINTALA, 2015; GOODFELLOW *et al.*, 2014).

Para treinar D , em cada iteração rotula-se algumas amostras reais retiradas dos dados de treinamento como 1 e algumas amostras geradas fornecidas por G como 0. Dessa forma, pode-se usar uma estrutura de treinamento supervisionado convencional para atualizar os parâmetros de D a fim de minimizar uma função de perda, conforme mostrado no esquema da Figura 4.7:

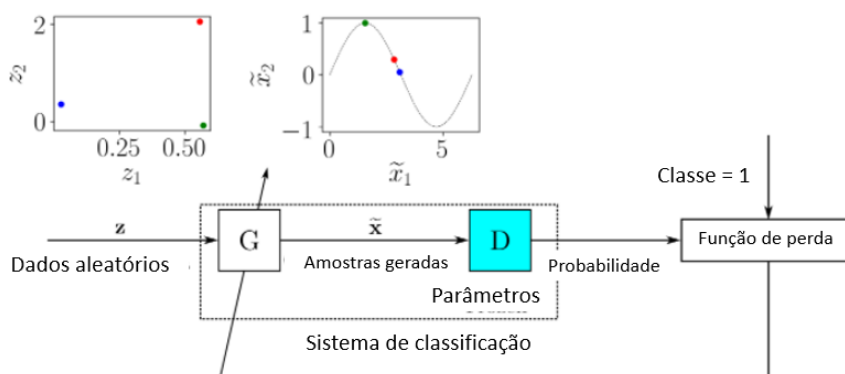
Figura 4.7 – Esquema do funcionamento da GAN



Fonte: o autor, adaptado de (Renato Candido, 2020)

Para cada lote de dados de treinamento contendo amostras reais e geradas rotuladas, atualiza-se os parâmetros de D para minimizar uma função de perda. Depois que os parâmetros de D são atualizados, G é treinado para produzir melhores amostras geradas. A saída de G é conectada a D, cujos parâmetros são mantidos congelados, conforme ilustrado na Figura 4.8:

Figura 4.8 – Esquema do funcionamento da GAN



Fonte: o autor, adaptado de (Renato Candido, 2020)

O sistema é composto de G e D como um único sistema de classificação que recebe amostras aleatórias como entrada e produz a classificação, que neste caso pode ser interpretada como uma probabilidade.

Quando G faz um trabalho bom o suficiente para enganar D, a probabilidade de saída deve ser próxima de 1. Também pode-se usar uma estrutura de treinamento supervisionado convencional aqui: o conjunto de dados para treinar o sistema de classificação composto

de G e D seria fornecido por amostras de entrada aleatórias, e o rótulo associado a cada amostra de entrada seria 1.

Durante o treinamento, conforme os parâmetros de D e G são atualizados, espera-se que as amostras geradas por G se assemelhem mais aos dados reais, e D terá mais problemas para distinguir entre dados reais e gerados (GOODFELLOW *et al.*, 2014).

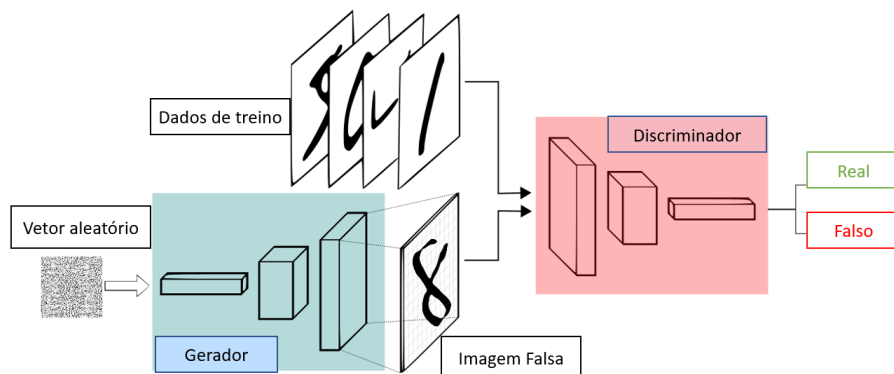
4.4 Rede GAN aplicada no conjunto de dados MNIST

Como exemplificação do funcionamento da rede GAN, foi gerado números escritos à mão, como os encontrados no conjunto de dados MNIST, retirado do mundo real. O objetivo do discriminador, quando mostrada uma instância do verdadeiro conjunto de dados MNIST, é reconhecer aqueles que são autênticos (DENG, 2012).

Enquanto isso, o gerador está criando novas imagens sintéticas que são transmitidas ao discriminador. O gerador gera as imagens fake na esperança de que elas também sejam consideradas autênticas, mesmo sendo falsas. O objetivo do gerador é gerar dígitos manuscritos cada vez melhores. O objetivo do discriminador é identificar imagens falsas do gerador.

Enfim, para ilustração da arquitetura de uma rede GAN, a Figura 4.9 mostra a representação de todo o processo.

Figura 4.9 – Esquema do funcionamento da rede GAN



Fonte: o autor, adaptado de (Renato Candido, 2020)

Para o MNIST, a rede discriminadora é uma rede convolucional padrão que pode categorizar as imagens alimentadas, um classificador binomial que rotula as imagens como reais ou falsas. O gerador é uma rede convolucional inversa, em certo sentido: enquanto um classificador convolucional padrão recebe uma imagem e reduz a amostragem para produzir uma probabilidade, o gerador pega um vetor de ruído aleatório e faz o upsampling para

uma imagem. O primeiro joga fora os dados por meio de técnicas de *downsampling*, como o maxpool, e o segundo gera novos dados.

Ambas as redes estão tentando otimizar uma função objetivo (função de perda) diferente e oposta. À medida que o discriminador muda seu comportamento, o gerador também muda e vice-versa. Suas perdas empurram um contra o outro. Uma ideia simples e genial! Durante o treinamento, a rede generativa vai aprendendo a criar uma imagem fake que fica cada vez mais próxima de uma imagem real. Alguns pesquisadores tem feito o mesmo com a voz, gerando falas que são falsas, mas se parecem muito com falas verdadeiras. Esse é o perigo das *Deep Fakes*, quando a tecnologia é usada para o mal, infelizmente (SAYLER; HARRIS, 2020).

4.5 Aumento de Dados (Data Augmentation) no Aprendizado Profundo

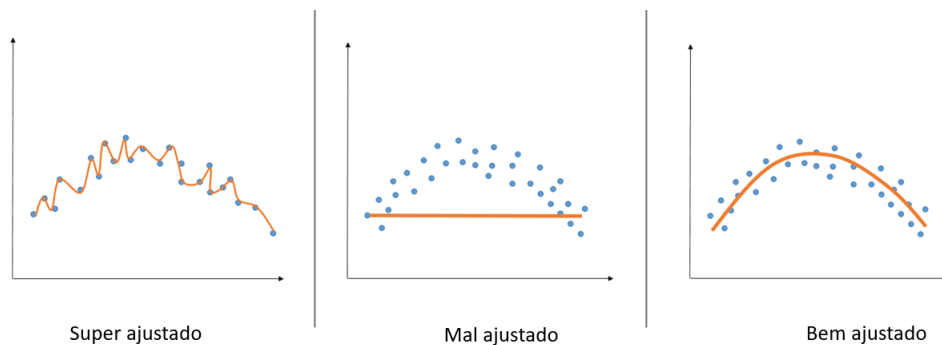
A modelagem e treinamento de novo um modelo para uma tarefa de aprendizado de máquina, independentemente de ser uma classificação ou regressão, tem por objetivo final fazer previsões confiáveis sobre novos dados de entrada que nunca foram vistos antes. Em outras palavras, deseja-se que o modelo generalize bem os novos dados.

Para atingir esse objetivo, deve-se evitar que o modelo seja excessivamente ajustado aos dados de treinamento (*overfitted* - com ajuste excessivo) ou não seja capaz de capturar o padrão nos dados (*underfitted* - ajuste insuficiente). A Figura 4.10 ilustra a caracterização dos cenários de *overfitted*, *underfitted* e a ótimo generalização.

Uma das razões pelas quais o *overfitting* pode ocorrer é a falta de dados. Na verdade, se o treinando do modelo estiver com poucos dados, ele tentará exasperar sua extração de recursos dos dados de treinamento, com o risco de identificar padrões que não existem.

No entanto, muitas vezes acontece que os dados disponíveis são muito poucos e não se pode obter mais amostras. Ou seja, imagine uma empresa de manufatura que deseja examinar períodos de trabalho de suas máquinas com o objetivo de classificá-las como “saudáveis” ou “em risco de quebra”. Para treinar o algoritmo (digamos, uma rede neural convolucional, CNN), a empresa precisará de um monte de imagens pré-rotuladas.

Figura 4.10 – Esquema do funcionamento da rede GAN



Fonte: (PEREZ; WANG, 2017)

O procedimento de coleta de dados demandará tempo, e se a empresa quiser acelerar o processo ao invés de esperar que surjam novas imagens, ela poderia utilizar os dados disponíveis e deles derivar novas imagens, de forma que cada “nova imagem” seja criada de forma consistente em relação às já existentes.

Este processo é chamado de aumento de dados(do inglês - *data augmentation*) e é extremamente poderoso em termos de aumento de precisão do modelo (PEREZ; WANG, 2017).

Existem várias maneiras de criar novos dados a partir das disponíveis. Pensando em sinais, pode-se ter novos dados a partir de:

- deslocamentos nas amplitudes do sinal
- aplicação de perturbações gerando mais picos
- aplicação de filtros para retirar ruídos em sinais ruidosos
- dentre outras técnicas

Ao fim do processo de data augmentation, o conjunto de dados possuirá mais amostras e que são reconhecíveis pelo classificador do modelo.

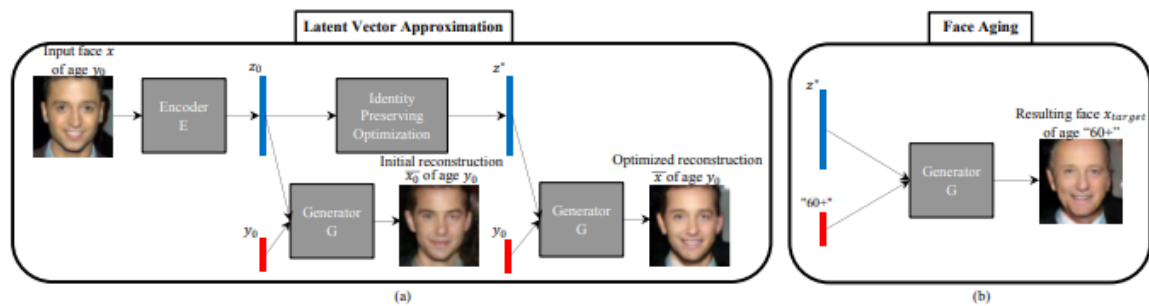
4.6 O uso de Redes Generativas Adversárias para fazer modificações em sinais

Recentemente, no trabalho de (ANTIPOV; BACCOUCHE; DUGELAY, 2017) foi proposto o sistema baseado em GAN para envelhecimento automático da face. Ao contrário de trabalhos anteriores empregando GANs para alterar os atributos faciais como em (YANG;

ZHANG; YIN, 2018), este fez uma ênfase particular na preservação da identidade da pessoa original na versão envelhecida de seu rosto.

O método de envelhecimento facial é baseado no Age-cGAN, um modelo gerador para a síntese de rostos humanos dentro da idade exigida. Uma vez que Age-cGAN é treinado, o envelhecimento facial está concluído em duas etapas como mostra a Figura 4.11 :

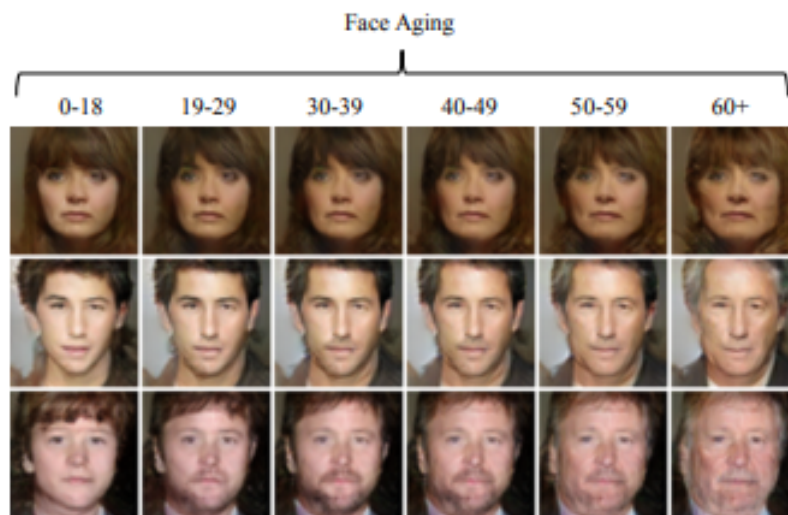
Figura 4.11 – Método de envelhecimento facial. (a) aproximação do vetor latente para reconstruir a imagem de entrada; (b) mudar a idade condição na entrada do gerador G para realizar o envelhecimento facial.



Fonte: (ANTIPOV; BACCOUCHE; DUGELAY, 2017)

Como resultado da aplicação de GAN para envelhecimento facial, a Figura 4.12 ilustra algumas faces que sofreram o processo de envelhecimento.

Figura 4.12 – Exemplos de reconstrução facial e envelhecimento

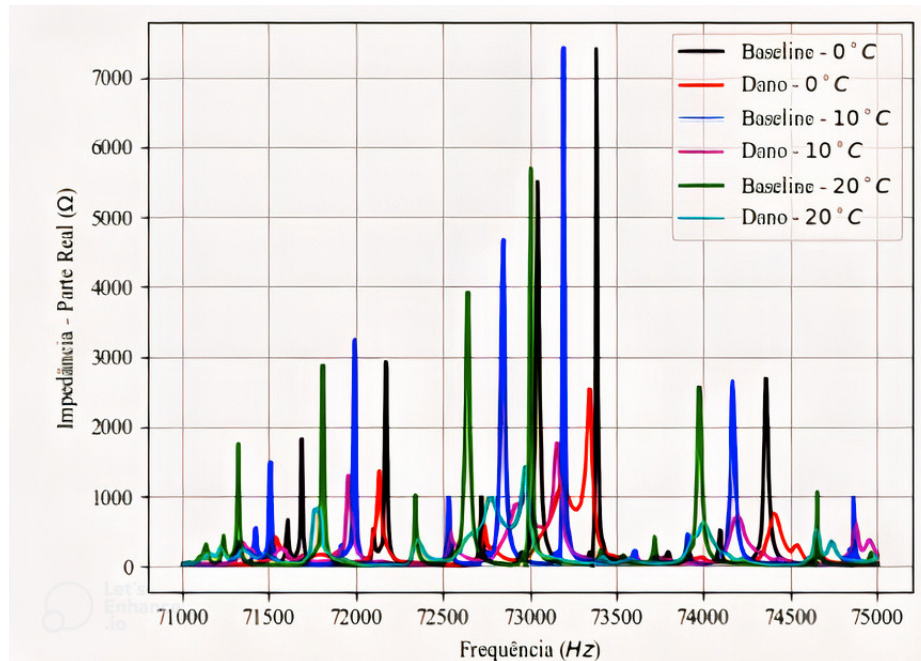


Fonte: (ANTIPOV; BACCOUCHE; DUGELAY, 2017)

Partindo dessa metodologia, este trabalho busca a geração de sinais de impedância eletromecânica visando o aumento de dados a partir da entrada de poucos sinais, mantendo sempre os mesmos padrões dos sinais reais, como ocorre com o método de envelhecimento,

onde as características da face não sofrem distorções a ponto de não representar a imagem real. Em resumo, um sinal de impedância eletromecânica pode ser como na Figura 4.13

Figura 4.13 – Exemplos de sinal de impedância eletromecânica



Fonte: (REZENDE *et al.*, 2021)

Capítulo 5

Metodologia

5.1 Estudo de Caso 01 - Sinal Unidimensional Básico

Neste estudo, foi elaborada uma tarefa simples que fornece um bom contexto para desenvolver uma GAN simples do zero é uma função unidimensional. Isso ocorre porque tanto as amostras reais quanto as geradas podem ser plotadas e inspecionadas visualmente para se ter uma ideia do que foi aprendido.

Uma função unidimensional também não requer modelos sofisticados de rede neural, o que significa que os modelos específicos de gerador e discriminador usados na arquitetura podem ser facilmente entendidos.

5.1.1 Função unidimensional

Para este caso foi selecionou-se uma função unidimensional para modelar, seguindo a equação 5.1:

$$y = f(x) \tag{5.1}$$

Onde x são os valores de entrada e y são os valores de saída da função.

Especificamente, o objetivo é ter uma função que seja possível entender e plotar facilmente. Isso ajudará tanto na definição de uma expectativa do que o modelo deve gerar quanto no uso de uma inspeção visual dos exemplos gerados para ter uma ideia de sua qualidade.

Usou-se uma função Gauss, que pode ser descrita como na equação 5.2, para criar um sinal com 2 picos gaussianos.

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{5.2}$$

Onde:

- μ é a média da distribuição
- σ^2 é a varância da distribuição
- σ é o desvio padrão da distribuição
- X são os valores dentro da distribuição

A função pode ser definida em Python como na Figura 5.1:

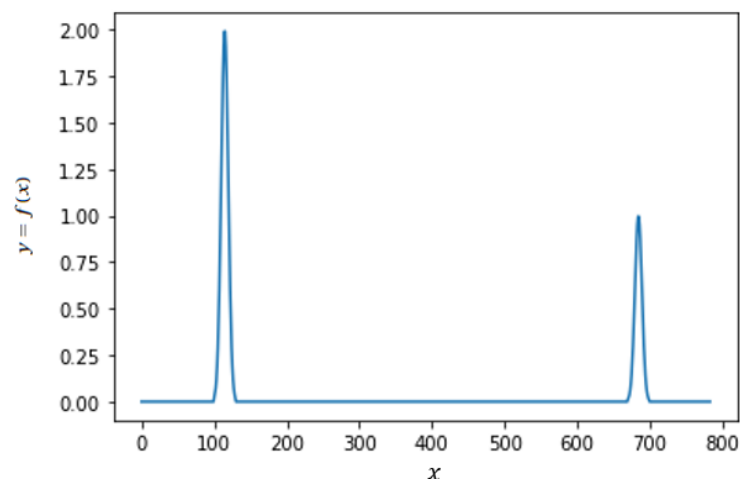
Figura 5.1 – Função Gaussiana em Python

```
def gaussiana(y,mu,sig):  
    return np.exp(-np.power(y-mu,2.)/(2*np.power(sig,2.)))
```

Fonte: o autor

Com a função definida, definiu-se o domínio de entrada entre 0 e 784 e calculando o valor de saída para cada um dos 784 pontos do sinal, adicionando um pico gaussiano com um desvio-padrão pequeno (gerando uma curva fechada, semelhante um pico) entre o intervalo de 100 e 130, e outro pico gaussiano com amplitude menor em relação ao primeiro entre o intervalo de 670 e 700. Depois, é possível visualizar o sinal gerado na Figura 5.2.

Figura 5.2 – Sinal Unidimensional com 2 picos



Fonte: o autor

Após a modelagem da função, novas amostras do sinal podem ser criadas alterando o intervalo entre os pontos onde os picos serão adicionados, de forma aleatória, criando-se assim o banco de dados com as amostras reais do sinal, no qual será expostos posteriormente.

5.1.2 Arquitetura da rede Discriminadora

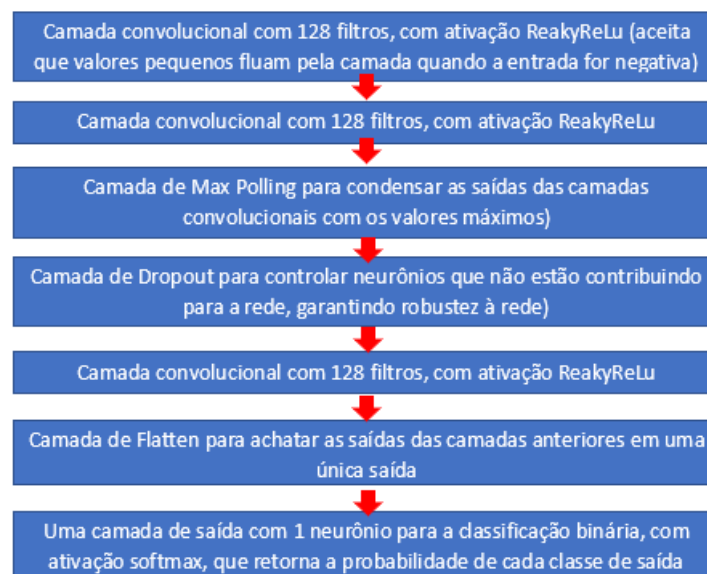
O próximo passo é definir o modelo discriminador.

O modelo deve pegar uma amostra do problema, um vetor com 784 pontos que formam um sinal, e produzir uma previsão de classificação para saber se a amostra é real ou falsa. Este é um problema de classificação binária.

- Entradas: Amostra com 784 valores (pontos).
- Saídas: Classificação binária, probabilidade de a amostra ser real (ou falsa).

A arquitetura da rede discriminadora para este caso foi modelada com a seguinte estrutura da Figura 5.3:

Figura 5.3 – Arquitetura rede discriminadora



Fonte: o autor.

O modelo minimizará a função de perda de entropia cruzada binária, e a versão de Adam do gradiente descendente estocástico será usada porque é muito eficaz em otimizar o treinamento da rede (GOODFELLOW IAN E BENGIO, 2016).

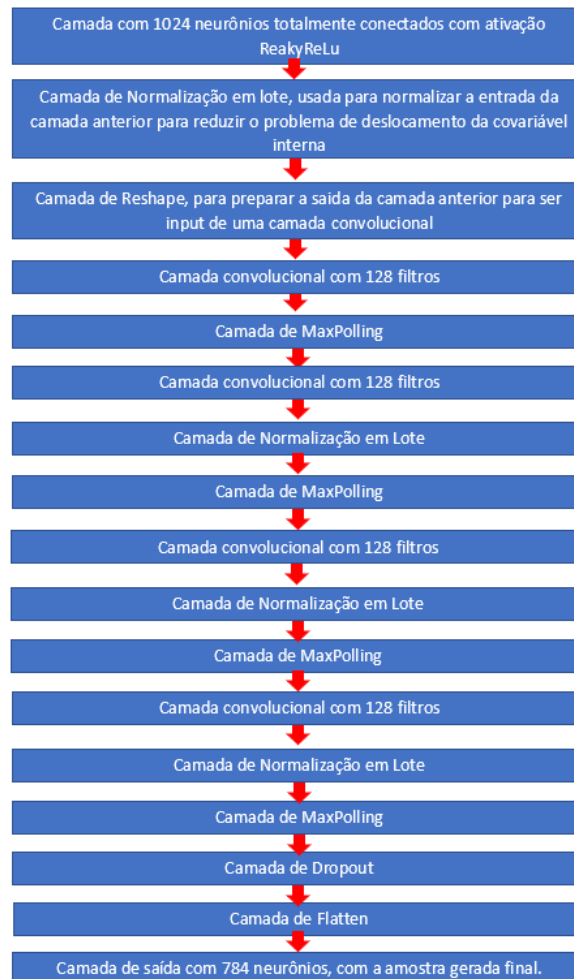
5.1.3 Arquitetura da rede Geradora

O modelo gerador deve pegar uma amostra ruidosa, um vetor pontos aleatórios, e produzir um sinal a partir da distribuição dos dados das amostras reais no qual ele irá aprender, essa distribuição pe chamada de espaço latente.

A arquitetura da rede geradora para este caso foi modelada com a seguinte estrutura ilustrada na Figura 5.4

- Entradas: Ruído aleatório representado por um vetor de pontos aleatórios.
- Saídas: amostra gerada baseada na amostra real (sinal com 2 picos gaussianos).

Figura 5.4 – Arquitetura rede geradora



Fonte: o autor.

O modelo minimizará também a função de perda de entropia cruzada binária, e a versão de Adam do gradiente descendente estocástico será usada porque é muito eficaz em otimizar o treinamento da rede.

Para o treinamento de ambas as redes, foi atribuído classe 1 para sinais reais e classe zeros para sinais gerados.

Os pesos no modelo gerador são atualizados com base no desempenho do modelo discriminador. Quando o discriminador é bom em detectar amostras falsas, o gerador é atualizado mais e quando o modelo discriminador é relativamente ruim ou confuso ao detectar amostras falsas, o modelo gerador é atualizado menos.

Apenas o discriminador se preocupa em distinguir entre exemplos reais e falsos; portanto, o modelo discriminador pode ser treinado de maneira independente em exemplos de

cada um. Já o modelo gerador se preocupa apenas com o desempenho do discriminador em exemplos falsos.

5.2 Estudo de Caso 02 - Sinal de impedância

Neste experimento foram utilizados os dados dos sinais coletados de um sistema de SHM, com variações de temperaturas, conduzido da seguinte forma.

Para a realização do experimento, quatro estruturas formadas a partir de vigas de alumínio foram utilizadas como corpos de prova e uma câmara de controle de temperatura e umidade da cadeia Platinous e uma câmara climática EPLH, conforme mostrado na Figura 5.4. A câmara foi instalada no Laboratório de Mecânica Estrutural (LMEst) da Faculdade de Engenharia Mecânica (FEMEC) da Universidade Federal de Uberlândia (UFU).

Vigas de alumínio de 500 mm de comprimento, 38 mm de largura, e 3,2 mm de espessura foram utilizadas neste experimento. Uma pastilha de PZT com 1 mm de espessura e 20 mm de diâmetro a 100 mm da borda da estrutura foi acoplada em cada uma delas.

Figura 5.5 – Câmara de temperatura EPL-4H



Fonte: (BARELLA *et al.*, 2021)

A Figura 5.5 apresenta os quatro corpos de prova ensaiados na condição de contorno

bi-apoiados. Nota-se que as vigas usinadas perderiam espessura ao longo dos processos de medição, o que poderia contribuir flexionando as vigas em seus dois pontos extremos de apoio. Ao colocá-las apoiadas sob a espessura, e não sobre a largura, seu maior momento de inércia nesta direção impede ou minimiza tais impactos no resultado

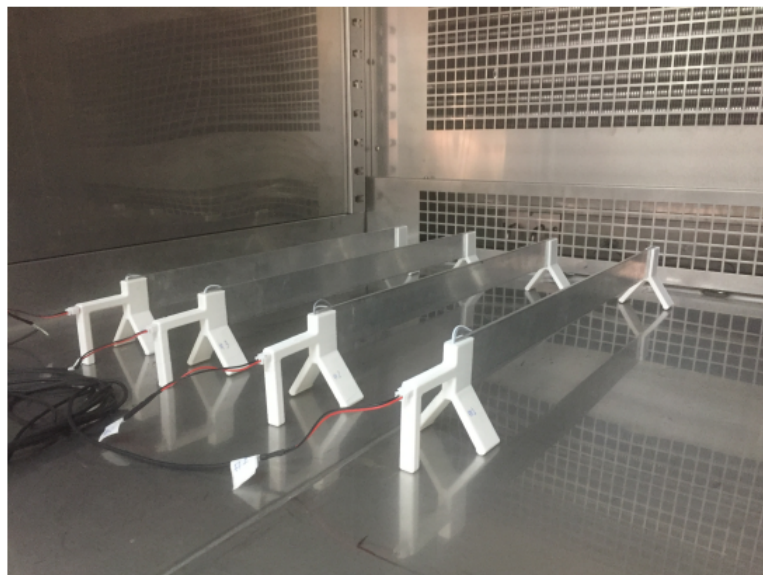
Figura 5.6 – 4 corpos de prova utilizados no ensaio.



Fonte: (BARELLA *et al.*, 2021)

Os pés dos cavaletes, estruturas em plástico projetados para posicionamento das estruturas dentro da câmara, são arredondados para diminuir a interferência do piso da câmara nas estruturas, o que possibilitou o embutimento de conectores nos próprios corpos de prova, na qual a Figura 5.6 apresenta o posicionamento das estruturas na câmara

Figura 5.7 – Posicionamento dos corpos de prova no interior da câmara.

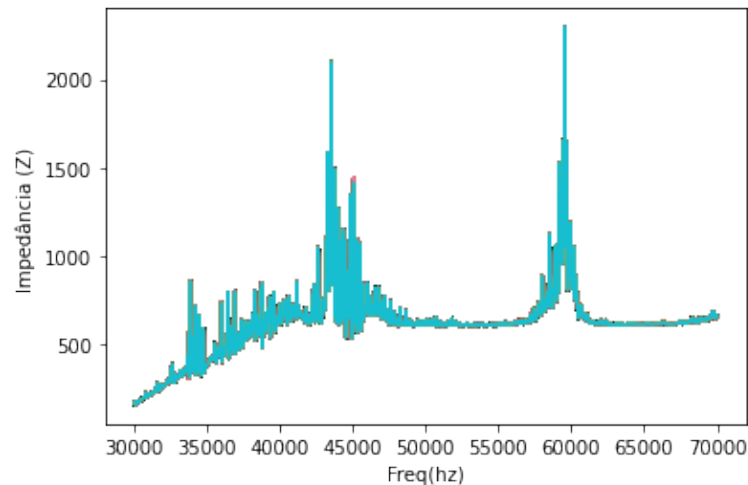


Fonte: (BARELLA *et al.*, 2021)

A faixa de temperaturas considerada variou de forma crescente de 3 °C em 3 °C, abrangendo de 10 °C a 40 °C em 11 ciclos. Por ser uma variação de temperatura pequena, após cada término de coletas das IS a câmara elevava sua temperatura e a mantinha por 30 minutos para estabilização da nova temperatura (equilíbrio térmico dos corpos de prova e do ambiente) antes do início do novo ciclo de coletas.

O intervalo de frequência empregado na aquisição das assinaturas de impedância foi de 30k Hz a 70k Hz, com passo de 10 Hz, gerando um total de 4000 pontos frequenciais. Os sinais de impedância do corpo de prova 1 para cada nível de integridade aplicado são apresentadas pela Figura 5.6, onde todas as assinaturas pertencem ao grupo de temperatura de 30°C.

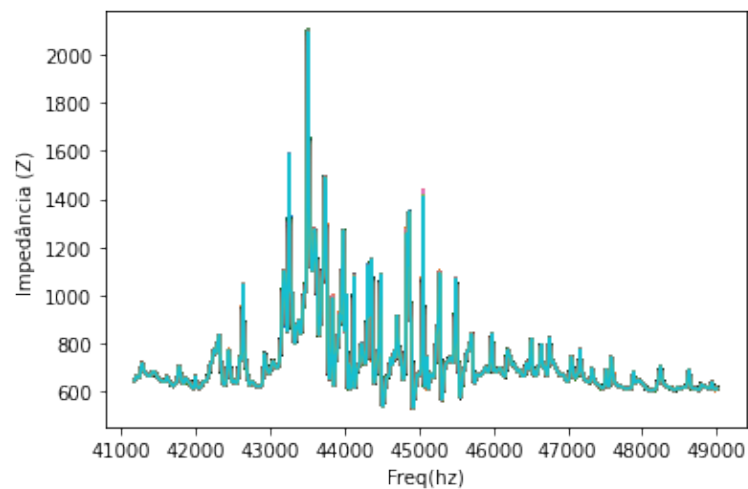
Figura 5.8 – Assinaturas de impedância do corpo de prova 1



Fonte: (BARELLA *et al.*, 2021)

Para melhor geração do sinal da rede geradora, delimitou-se o sinal coletado no range de 41000 a 49000, região com maiores diversidades de picos, proporcionando uma robustez a GAN modelada. O sinal delimitado pode ser visualizado na Figura 5.7

Figura 5.9 – Região das assinaturas utilizadas no experimento

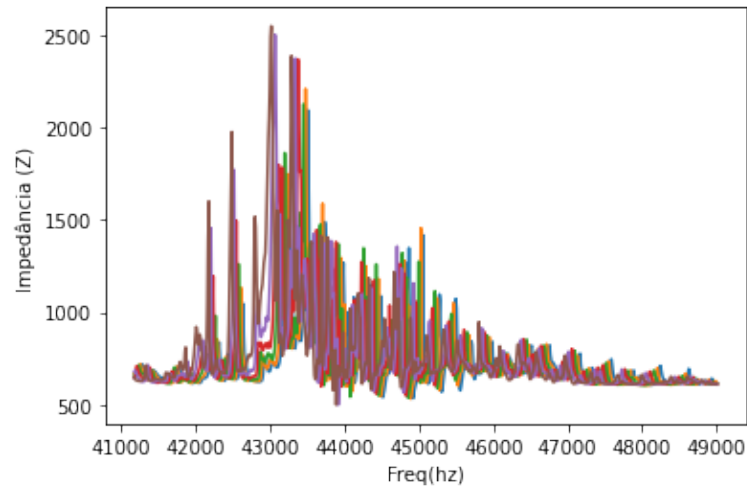


Fonte: o autor

O Banco de Dados final, considerando todas assinaturas com diferentes temperaturas como descrito anteriormente, pode ser ilustrado na Figura 5.8. Como as arquiteturas de deep

learning precisam de uma grande quantidade de dados, foi realizado um aumento numérico entre as amostras experimentais apenas com o intuito de aumentar as amostras que serão passadas para as redes.

Figura 5.10 – Banco de Dados Final



Fonte: o autor

5.2.1 Arquitetura da rede Discriminadora

A arquitetura da rede discriminadora, Figura 5.11, segue o mesmo contexto do estudo de caso 01, ou seja, resolver um problema de classificação binária e possui:

- Entradas: Amostra com 784 valores (pontos).
- Saídas: Classificação binária, probabilidade de a amostra ser real (ou falsa).

Figura 5.11 – Arquitetura rede discriminadora



Fonte: o autor.

O modelo minimizará a função de perda de entropia cruzada binária, e a versão de Adam do gradiente descendente estocástico será usada porque é muito eficaz em otimizar o treinamento da rede.

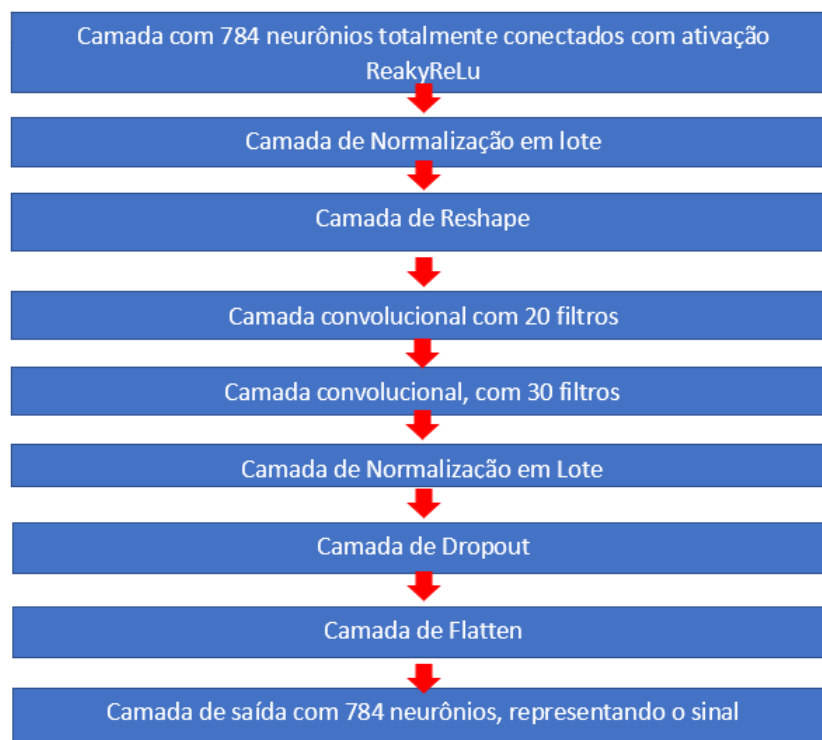
5.2.2 Arquitetura da rede Geradora

Seguindo o mesmo propósito do estudo de caso 01, O modelo deve pegar uma amostra ruidosa, um vetor pontos aleatórios, e produzir um sinal a partir da distribuição dos dados das amostras reais no qual ele irá aprender

- Entradas: Ruído aleatório representado por um vetor de pontos aleatórios.
- Saídas: amostra gerada baseada na amostra real (sinal de impedância).

A arquitetura da rede geradora para este caso foi modelada com a estrutura representada na Figura 5.12:

Figura 5.12 – Arquitetura rede geradora



Fonte: o autor.

O modelo minimizará também a função de perda de entropia cruzada binária, e a versão de Adam do gradiente descendente estocástico será usada porque é muito eficaz em otimizar o treinamento da rede. Para o treinamento de ambas as redes, foi atribuído classe 1 para sinais reais e classe 0 para sinais gerados.

Os pesos no modelo gerador são atualizados com base no desempenho do modelo discriminador. Quando o discriminador é bom em detectar amostras falsas, o gerador é atualizado mais e quando o modelo discriminador é relativamente ruim ou confuso ao detectar amostras falsas, o modelo gerador é atualizado menos.

Apenas o discriminador se preocupa em distinguir entre exemplos reais e falsos; portanto, o modelo discriminador pode ser treinado de maneira independente em exemplos de cada um. Já o modelo gerador se preocupa apenas com o desempenho do discriminador em exemplos falsos.

Ao treinar o gerador, o objetivo é que o discriminador pense que as amostras de saída do gerador são reais, não falsas.

5.3 Ferramentas Computacionais Necessárias

Este trabalho foi realizado no ambiente computacional Google Colaboratory — também conhecido como Google Colab — que é uma ferramenta em nuvem que permite criar e executar códigos na linguagem Python. Com ele, é possível rodar os programas diretamente do navegador, de forma simples e rápida. A escolha por esse ambiente é devido a utilização do hardware de qualidade disponibilizado pelos servidores da Google. Mais precisamente a utilização de Unidades de Processamento Gráfico (GPUs) na hora de executar os códigos, permitindo a execução dos códigos de forma mais rápida com prioridade para uso das GPUs de melhor desempenho e com mais memória RAM para armazenar dados.

Como supracitado, a linguagem de programação utilizada será o Python versão 3.8, uma ferramenta que possui uma coleção de algoritmos de aprendizagem de máquina para diversas tarefas de aprendizado profundo, além de ferramentas para processamento dos dados, classificação, regressão, agrupamentos, regras de associação e visualização de informações.

As bibliotecas com grande potencial de uso:

- NumPy - que fornece ferramentas para cálculos matemáticos
- Pandas - que fornece ferramentas para análise e manipulação de dados.
- Matplotlib - que fornece ferramentas para geração de gráficos 2D a partir de arrays.

Para a implementação das redes GAN, foram utilizadas as bibliotecas TensorFlow, scikit-learn, Keras para TensorFlow.

A biblioteca TensorFlow é um conjunto de ferramentas de código aberto criada para aprendizado de máquina, computação numérica e muitas outras tarefas. Foi desenvolvido

pelo Google em 2015 e rapidamente se tornou uma das principais ferramentas para machine learning e deep learning (([ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012](#))).

A biblioteca scikit-learn dispõe de ferramentas simples e eficientes para análise preditiva de dados, é reutilizável em diferentes situações, possui código aberto, sendo acessível a todos e foi construída sobre os pacotes NumPy, SciPy e matplotlib (([KRAMER, 2016](#))).

Keras é uma API de aprendizado profundo escrita em Python e executada na biblioteca de aprendizado de máquina TensorFlow . Foi desenvolvido com foco em possibilitar experimentação rápida. Ser capaz de ir da ideia ao resultado o mais rápido possível é a chave para fazer uma boa pesquisa (([CHOLLET *et al.*, 2015](#))).

Capítulo 6

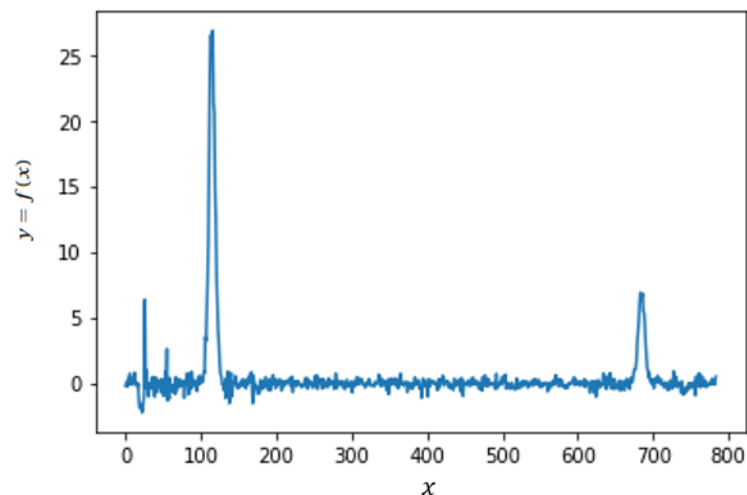
Resultados e Discussões

Este capítulo foi dividido em duas seções principais: Resultados do Banco de Dados do Estudo de Caso 01 (6.1) e Resultados do Banco de Dados do Estudo de Caso 02 (6.2). Em cada seção, é apresentado os resultados obtidos pelos GANs utilizados neste trabalho que foram descritos na metodologia. Por fim, realizou-se alguns experimentos de tamanho de amostras, tamanho do banco de dados e uma Rede classificadora foi modelada para auditar os sinais gerados pela GAN. para validar que os sinais gerados podem ser confundidos com os sinais reais e, por isso, ter uma GAN eficiente na geração dos sinais de impedância.

6.1 Resultados do Banco de Dados do Estudo de Caso 01

Aplicando o treinamento das redes modeladas na metodologia descrita no capítulo anterior, obtém-se os primeiros resultados, onde na Figura 6.1 é ilustrado o sinal gerado pela GAN projetada.

Figura 6.1 – Sinal unidimensional gerado pela GAN



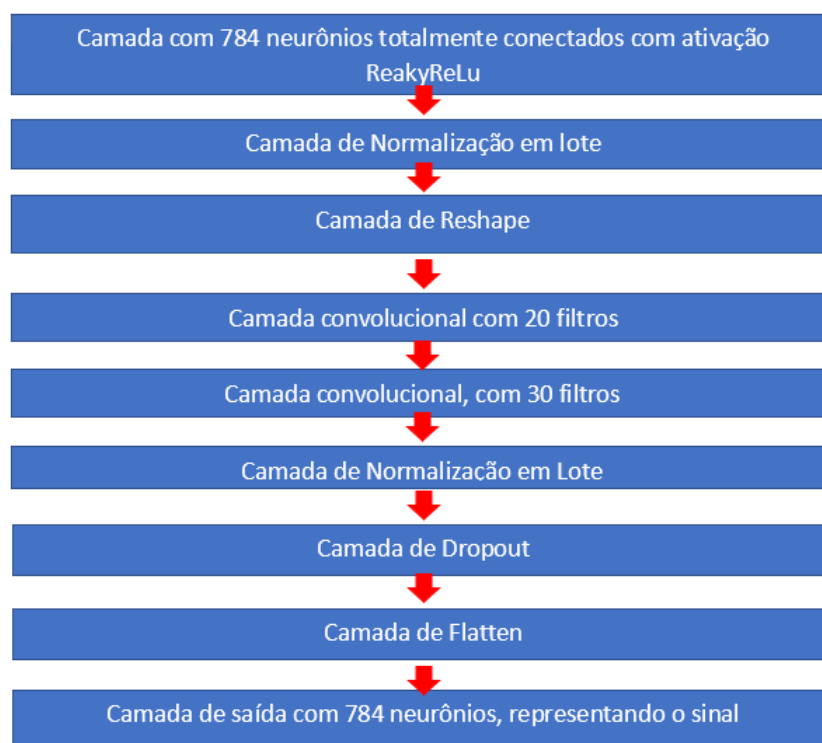
Fonte: o autor

O resultado dos sinais gerados pelas redes acima implementadas foi considerável, porém percebe-se a presença de ruídos em locais inesperados que representam vazamento de dados nas camadas de convolução das redes, fenômeno descrito como “leakage”. Para solucionar, inicialmente a seguinte hipótese foi considerada:

- 1. Arquitetura muito complexa para as amostras do problema

Portanto, uma nova arquitetura para a rede geradora foi implementada e é apresentada a seguir na Figura 6.2.

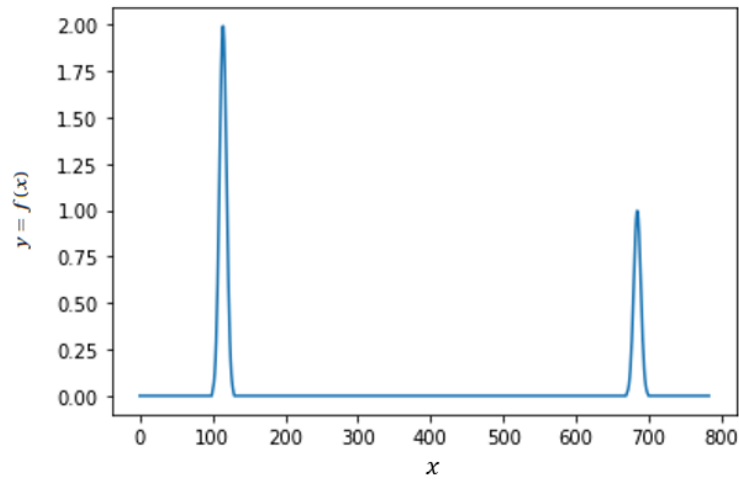
Figura 6.2 – Arquitetura rede geradora



Fonte: o autor.

Esta arquitetura, a rede recebe um ruído com 784 pontos e retorna uma saída também de 784 pontos. As camadas de MaxPollig, Conv2D e o Dropout tem por objetivo garantir que todas características das imagens inseridas sejam extraídas e que não sejam enviesadas durante o processo de treinamento, o que justifica sua utilização aqui. A rede discriminadora foi mantida com a arquitetura anteriormente descrita. O resultado obtido por essa nova configuração é descrita a seguir:

Figura 6.3 – Sinal unidimensional gerado pela GAN



Fonte: o autor

Para uma avaliação quantitativa desse resultado, utilizou-se a métrica R^2 , também conhecida como R-dois ou coeficiente de determinação, representa o percentual da variância dos dados que é explicado pelo modelo. Os resultados variam de 0 a 1, geralmente também são expressos em termos percentuais, ou seja, variando entre 0% e 100%.

Quanto maior é o valor de R^2 , mais explicativo é o modelo em relação aos dados previstos. Na equação 6.1 é mostrado o cálculo desta métrica, no qual y e \hat{y} os valores reais e previstos, respectivamente, e \bar{y} representa a média dos valores reais.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n ((y_i - \hat{y}_i))^2}{\sum_{i=1}^n ((y_i - \bar{y}))^2} \quad (6.1)$$

Onde os valores de y são os valores verdadeiros e o \bar{y} é a média desses valores, enquanto que \hat{y} são os valores preditos. Os resultados de R^2 ficam entre 0 e 1, quanto mais perto de 1 melhor e pior para resultados perto de 0.

Definido o conceito da métrica utilizada, é possível visualizar na Tabela 6.1 o valor do R^2 para o sinal gerado neste estudo de caso, tanto antes dos ajustes quanto após os ajustes realizado.

Tabela 6.1 – Métrica R2 aplicada aos sinais gerados nas duas versões da GAN para o Estudo de Caso 01

Arquitetura da rede	R2-score	R2-score %
GAN inicial	0,75689	76%
GAN após ajustes	0,99654	100%

A partir da Tabela 6.1 é possível visualizar a superioridade do sinal gerado pela GAN que foi reajustada, quantificada pela métrica R^2 . Com uma métrica com aproximadamente

100%, o sinal gerado pela GAN pode ser considerado como um sinal muito parecido com uma amostra real. Por isso, conclui-se que o modelo de GAN desenvolvido neste estudo de caso possibilitou a sintetização de sinais com 2 picos gaussianos idênticos à uma amostra do banco de dados real.

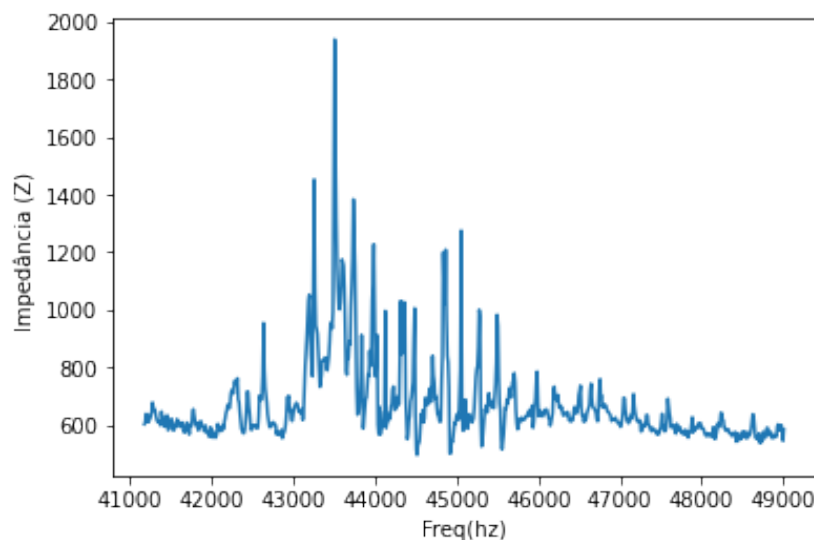
6.1.1 Conclusões sobre o Estudo de Caso 01

Este experimento tinha como objetivo aplicar os conceitos de redes GAN em um problema simples para entendimentos da arquitetura ideal, processo de aprendizagem e geração de novo sinal, bem como ver a aplicabilidade das redes GANS em gerar sinais com picos, visando próximas etapas de geração de sinais voltamos para impedância mecânica

6.2 Resultados do Banco de Dados do Estudo de Caso 02

Aplicando o treinamento das redes modeladas na metodologia descrita no capítulo anterior para este estudo de caso, obtém-se os primeiros resultados, onde na Figura 6.4 é ilustrado o sinal gerado pela GAN projetada.

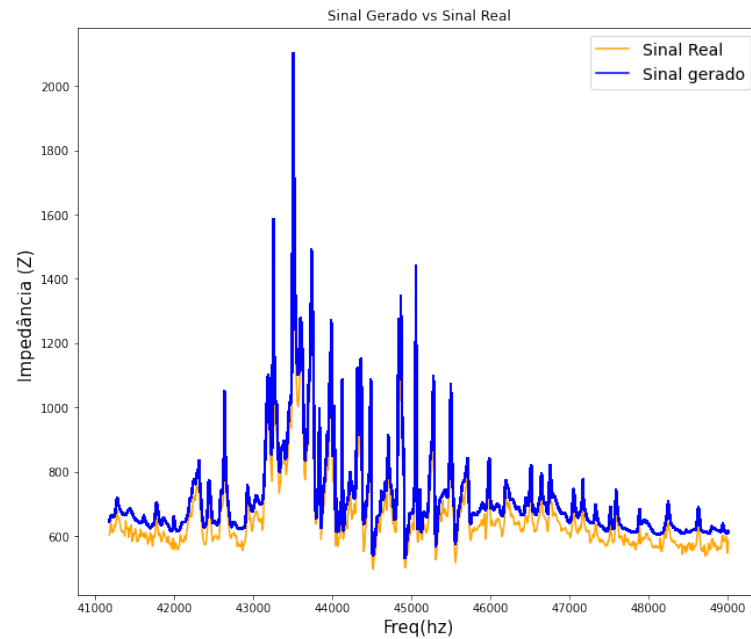
Figura 6.4 – Sinal de impedância eletromecânica gerado pela GAN



Fonte: o autor

O resultado dos sinais gerados pela GAN foi muito considerável, percebendo-se a presença dos picos de diferentes amplitudes. A Figura 6.5 ilustra a comparação entre um sinal gerado e um sinal real, retirado do banco de dados.

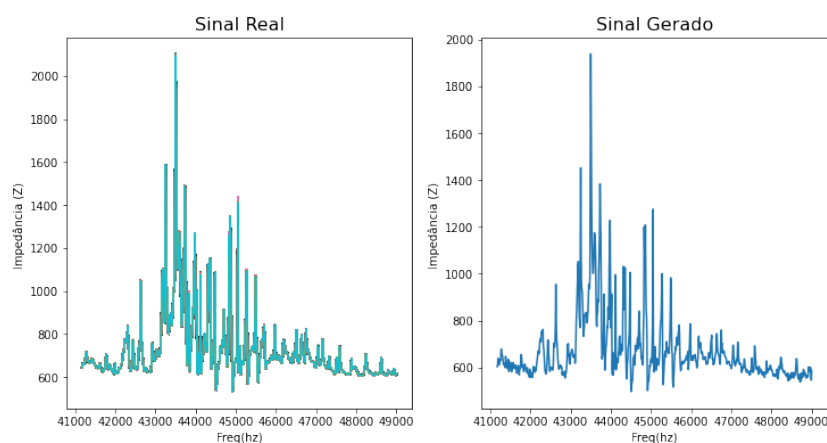
Figura 6.5 – Sinal de impedância eletromecânica: sinal gerado vs sinal real



Fonte: o autor

Pela Figura 6.4 é possível visualizar a semelhança entre os sinais, mesmo com uma pequena diferença entre as amplitudes, o que não acarreta em um descarte dos sinais gerados. Além disso, é possível avaliar que o sinal gerado ficou semelhante aos sinais de impedância que passaram pelo processo de compensação de temperatura, como é possível ver na Figura 6.6.

Figura 6.6 – Semelhança entre os sinais: sinal compensado temperatura vs sinal gerado



Fonte: o autor

Para avaliar o desempenho da geração de sinais de impedância e garantir que a rede está realmente gerando sinais semelhantes aos reais, as seguintes hipóteses foram testadas:

- 1. tamanho da amostra de pontos aleatórios para entrada na rede - testar quantidade de pontos diferentes do ruído aleatório.
- 2. Tamanho do banco de dados de treinamento, qual quantidade deve ser considerada para que a rede faça a geração correta do sinal
- 3. Quantidade de amostras com variações de temperatura pode afetar na geração final do modelo
- 4. Um modelo de classificador de sinais consegue diferenciar um sinal real e um gerado?

Para as hipóteses 1, 2 e 3, a métrica R^2 foi utilizada como avaliadora quantitativa dos resultados. Para a hipótese 4, a matriz de confusão foi elaborada para quantificar a taxa de acerto do modelo.

6.2.1 Hipótese 01 - Alteração no tamanho dos pontos aleatórios que entram na rede geradora

Este teste visa avaliar se o tamanho dos pontos aleatórios que compõe o sinal ruidoso inicial impacta no sinal gerado pela rede. A tabela 6.2 mostra todas configurações testadas para essa hipótese e suas respectivas métricas.

Tabela 6.2 – Métrica R^2 aplicada aos diferentes tamanhos de amostras aleatórias

PONTOS	R2_SCORE
784	90%
576	88%
400	87.50%
100	87.50%
1000	90%

A partir da Tabela 6.2 é possível analisar que o aumento de pontos na amostra não altera o sinal gerado pela rede, ficando com 90% de R^2 , o que significa uma alta semelhança entre os sinais. Também conclui-se que amostras com menos de 784 pontos altera o sinal gerado, porém não há uma diferença muito grande, cerca de 2.5% de diferença no R^2 .

6.2.2 Hipótese 02 - Tamanho do banco de dados de treinamento

Este teste visa avaliar se o tamanho do banco de dados com sinais reais impacta no sinal gerado pela rede. A tabela 6.3 mostra todas configurações testadas para essa hipótese e suas respectivas métricas.

Tabela 6.3 – Métrica R2 aplicada aos diferentes tamanhos de amostras aleatórias

TAMANHO	PONTOS	R2_SCORE
10	784	-21%
100	784	-21%
1000	784	-17.00%
10000	784	53.00%
61200	784	90%

A partir da Tabela 6.3 é possível analisar que o aumentada quantidade de amostras reais no banco de dados de treinamento impacta significativamente no sinal gerado pela rede, ficando com 90% de R^2 , o que justifica a teoria que para algoritmos de aprendizagem profunda, o banco de dados necessita de muitas amostras. Também conclui-se que um banco de dados com poucas amostras gera sinais ruidosos, com nenhuma semelhança com o sinal real.

6.2.3 Hipótese 03 - Quantidade de amostras com variações de temperatura pode afetar na geração final do modelo

Este teste visa avaliar se a quantidade de amostras com variação de temperatura impacta no sinal gerado pela rede. A tabela 6.4 mostra todas configurações testadas para essa hipótese e sua respectivas métricas.

Tabela 6.4 – Quantidade de sinais com variação de temperatura

TAMANHO	PONTOS	R2_SCORE
6	784	90%
60	784	90%
600	784	90.00%
6000	784	89.00%
15000	784	89%

A partir da Tabela 6.4 é possível analisar que o aumentada quantidade de amostras reais com variação de temperatura no banco de dados de treinamento não gera um considerável impacto no sinal gerado pela rede, ficando com 90% de R^2 , o que mantém a geração de sinais compensados pela rede, o banco de dados necessita de muitas amostras. Também conclui-se que um banco de dados com poucas amostras gera sinais ruidosos, com nenhuma semelhança com o sinal real.

6.2.4 Hipótese 04 - Um modelo de classificador de sinais consegue diferenciar um sinal real e um gerado?

Este teste visa avaliar se os sinais gerados podem realmente serem considerados como semelhantes aos sinais reais, uma vez que será colocado a um modelo classificador se o sinal

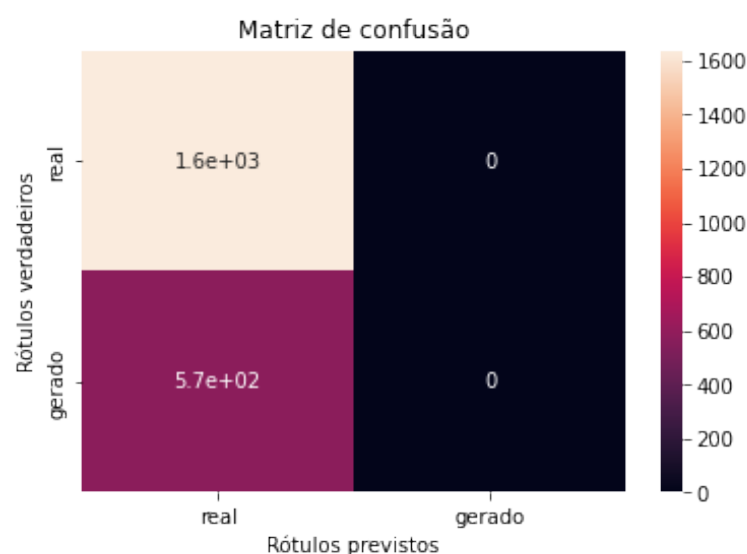
é real ou gerado. Para isso, a seguinte Rede Neural Convolutacional foi modelada com:

- Camada convolutacional com 110 filtros
- Camada de Normalização em lote, usada para normalizar a entrada da camada anterior para reduzir o problema de deslocamento da covariável interna
- Segunda camada convolutacional, com 110 filtros
- Camada de MaxPolling
- Camada de Dropout
- Camada de Flatten
- Camada de saída com 300 neurônios
- Camada de saída com 1 neurônio, com ativação softmax

Essa arquitetura de rede classificadora foi treinada em um conjunto com 10.000 amostras reais e 10.000 geradas pela GAN. Para validação dos resultados dessa rede, um conjunto de teste, contendo 2211 amostras nunca vistas pela rede durante o treinamento, divididas em 1638 amostras reais e 573 amostras geradas pela rede.

O resultado das classificações dessas amostras de testes podem ser visualizadas na matriz de confusão na Figura 6.6

Figura 6.7 – Matriz de Confusão para os sinais gerados vs reais



Fonte: o autor

A partir da Figura 6.6 é possível concluir que a rede classificadora considerou os sinais gerados como da classe dos sinais reais, o que confirma o valor de R^2 em 90% e reafirma que os sinais gerados são semelhantes aos sinais reais de impedância.

6.2.5 Conclusões sobre o Estudo de Caso 02

Este experimento tinha como objetivo aplicar os conceitos de redes GAN em um problema real de sinais de impedância eletromecânica, que por possuir picos em amplitudes diferentes poderia ser um desafio para a rede GAN. Pelos resultados obtidos, é possível concluir que a GAN conseguiu gerar sinais semelhantes aos sinais reais de impedância, com um R^2 de 90%.

Capítulo 7

Conclusões

Neste trabalho, foi apresentado as Redes Generativas Adversárias, uma nova abordagem para geração de dados sintéticos que possam auxiliar em diversas áreas onde o processo de coleta de dados é um pouco limitada.

Um primeiro experimento foi realizado, com o objetivo aplicar os conceitos de redes GAN em um problema simples para entendimentos da arquitetura ideal, processo de aprendizagem e geração de novo sinal, bem como ver a aplicabilidade das redes GANS em gerar sinais com picos, visando próximas etapas de geração de sinais para impedância mecânica.

Com um resultado satisfatório, baseado em uma métrica R^2 com aproximadamente 100%, o sinal gerado pela GAN pode ser considerado como um sinal muito parecido com uma amostra real. Por isso, conclui-se que o modelo de GAN desenvolvido neste estudo de caso possibilitou a sintetização de sinais com 2 picos gaussianos idênticos à uma amostra do banco de dados real.

Em um segundo experimento, com sinais reais de impedância eletromecânica extraídos de um procedimento experimental descrito na Metodologia, os resultados obtidos e todas testes de hipóteses avaliadas, é possível concluir que a GAN também conseguiu gerar sinais semelhantes aos sinais reais de impedância, com um R^2 de 90% e uma matriz de confusão que mostra a quantidade de sinais gerados e que foram classificados como se fossem reais.

Portanto, as redes GAN obtiveram performance considerável na geração de sinais de impedancia eletromecanica semelhantes as amostras reais de uma estrutura não danificada. Há um bom potencial para uso posterior em aplicações na área de SHM e como trabalhos futuros, fica que mais pesquisas são necessárias para a geração de sinais representativos de uma estrutura danificada usando os sinais não danificados (baselines).

REFERÊNCIAS

ABU-MOSTAFA, Y. S.; MAGDON-ISMAIL, M.; LIN, H.-T. *Learning from data*. [S.l.]: AMLBook New York, NY, USA; 2012. v. 4. Citado 2 vezes nas páginas 26 e 63.

AKKO, S. Uma comparação empírica de técnicas convencionais, redes neurais e o modelo híbrido adaptive neuro fuzzy inference system (anfis) para análise de pontuação de crédito: o caso dos dados de cartão de crédito da turquia. v. 222, 2012. Citado na página 28.

ANTIPOV, G.; BACCOUCHE, M.; DUGELAY, J.-L. Face aging with conditional generative adversarial networks. In: IEEE. *2017 IEEE international conference on image processing (ICIP)*. [S.l.], 2017. p. 2089–2093. Citado 2 vezes nas páginas 50 e 51.

ARIA, A. *et al.* Estimating damage size and remaining useful life in degraded structures using deep learning-based multi-source data fusion. *Structural Health Monitoring*, SAGE Publications Sage UK: London, England, v. 19, n. 5, p. 1542–1559, 2020. Citado na página 26.

BAPTISTA, F. *et al.* An experimental study on the effect of temperature on piezoelectric sensors for impedance-based structural health monitoring. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 14, n. 1, p. 1208–1227, 2014. Citado 3 vezes nas páginas 18, 21 e 22.

BAPTISTA, F. G. *et al.* Uma contribuição aos sistemas de monitoramento de integridade estrutural baseados na impedância eletromecânica. Universidade Estadual Paulista (UNESP), 2010. Citado na página 18.

BARELLA, B. P. *et al.* Modelagem com aprendizado de máquina aplicada aos sistemas de monitoramento de integridade estrutural. Universidade Federal de Goiás, 2021. Citado 4 vezes nas páginas 34, 57, 58 e 59.

BARROS, W. K.; MORAIS, D. S.; FERNANDES, M. A. Detecção de câncer de pele com redes neurais artificiais. In: *XIII Congresso Brasileiro de Ciências da Computação*. [S.l.: s.n.], 2017. Citado na página 26.

BECKER, S. Procedimentos de aprendizagem não supervisionados para redes neurais. v. 2, 1991. Citado na página 36.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, IEEE, v. 5, n. 2, p. 157–166, 1994. Citado na página 39.

- BHARTI, A. K.; NEGI, R.; VERMA, D. K. A review on performance analysis and improvement of internet of things application. *International Journal of Computer Sciences and Engineering*, v. 7, n. 2, p. 367–371, 2019. Citado na página 25.
- BISHOP, C. M. e. o. *Redes neurais para reconhecimento de padrões*. [S.l.: s.n.], 1995. Citado na página 39.
- BRAGA, A. d. P.; CARVALHO, A.; LUDERMIR, T. *Redes neurais artificiais: teoria e prática*. Editora LTC, 2000. Citado na página 28.
- BRESFELEAN, V. P. *et al.* Determining students' academic failure profile founded on data mining methods. In: IEEE. *ITI 2008-30th International Conference on Information Technology Interfaces*. [S.l.], 2008. p. 317–322. Citado na página 33.
- BURDA, Y.; GROSSE, R.; SALAKHUTDINOV, R. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015. Citado na página 26.
- CAWLEY, P. The impedance method of non-destructive inspection. *NDT international*, Elsevier, v. 17, n. 2, p. 59–65, 1984. Citado na página 19.
- CHAUDHRY, Z. *et al.* Monitoring the integrity of composite patch structural repair via piezoelectric actuators/sensors. In: *36th structures, structural dynamics and materials conference*. [S.l.: s.n.], 1995. p. 1074. Citado na página 23.
- CHOLLET, F. *et al.* Keras. GitHub, 2015. Disponível em: <<https://github.com/fchollet/keras>>. Citado 2 vezes nas páginas 38 e 63.
- CIANG, C. C.; LEE, J.-R.; BANG, H.-J. Structural health monitoring for a wind turbine system: a review of damage detection methods. *Measurement science and technology*, IOP publishing, v. 19, n. 12, p. 122001, 2008. Citado na página 19.
- DEKKER, G. W.; PECHENIZKIY, M.; VLEESHOUWERS, J. M. Predicting students drop out: A case study. *International Working Group on Educational Data Mining*, ERIC, 2009. Citado na página 24.
- DENG, L. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, IEEE, v. 29, n. 6, p. 141–142, 2012. Citado na página 48.
- DUCHI, J.; SINGER, Y. Efficient online and batch learning using forward backward splitting. *The Journal of Machine Learning Research*, JMLR. org, v. 10, p. 2899–2934, 2009. Citado na página 36.
- FARIA, R.; CAMSARI, K. Y.; DATTA, S. Implementing bayesian networks with embedded stochastic mram. *AIP Advances*, AIP Publishing LLC/AIP Publishing, v. 8, n. 4, p. 045101, 2018. Citado 2 vezes nas páginas 30 e 39.
- FUKUSHIMA, K. Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, Springer, v. 20, n. 3, p. 121–136, 1975. Citado na página 32.
- GIBILISCO, S. *Manual de eletrônica e des telecomunicações*. [S.l.]: Reichmann & Affonso, 2002. Citado na página 19.

GODEFROID, L. B.; CÂNDIDO, L. C.; MORAIS, W. A. Análise de falhas. *ASSOCIAÇÃO BRASILEIRA DE MATERIAIS–ABM. Apostila de curso*, 2006. Citado na página 35.

GOODFELLOW, I. *et al.* Generative adversarial nets. *Advances in neural information processing systems*, v. 27, 2014. Citado 7 vezes nas páginas 16, 41, 42, 43, 44, 46 e 48.

GOODFELLOW IAN E BENGIO, Y. e. C. A. e. B. Y. *Deep learning*. [S.l.: s.n.], 2016. v. 1. Citado 8 vezes nas páginas 24, 26, 27, 34, 35, 36, 38 e 55.

GULRAJANI, I. *et al.* Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*, 2017. Citado na página 16.

HAFEMANN, L. G.; OLIVEIRA, L. S.; CAVALIN, P. Forest species recognition using deep convolutional neural networks. In: IEEE. *2014 22Nd international conference on pattern recognition*. [S.l.], 2014. p. 1103–1107. Citado na página 37.

HALL, S. The effective management and use of structural health data. In: *Proceedings of the 2nd International Workshop on Structural Health Monitoring*. [S.l.: s.n.], 1999. p. 265–275. Citado na página 17.

HAN, J.; PEI, J.; KAMBER, M. *Data mining: concepts and techniques*. [S.l.]: Elsevier, 2011. Citado na página 38.

HAYKIN, S. *Redes neurais: princípios e prática*, 2ª edição, tradução: Paulo martins engel. *Editora: Bookman, Porto Alegre, Cap*, v. 1, n. 2, p. 3, 2001. Citado 9 vezes nas páginas 24, 29, 30, 31, 33, 35, 36, 38 e 40.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 40.

JIANG, F. *et al.* An end-to-end compression framework based on convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, v. 28, n. 10, p. 3007–3018, 2017. Citado na página 30.

KOHONEN, T. *Construction of similarity diagrams for phonemes by a self-organizing algorithm*. [S.l.]: Teknillinen korkeakoulu, 1981. Citado na página 32.

KOHONEN TEUVO E KASKI, S. e. L. H. formação auto-organizada de vários filtros de características invariantes no subespaço adaptativo som. v. 9, 1997. Citado na página 36.

KOSE UTKU E ARSLAN, A. Otimização de autoaprendizagem em cursos de engenharia da computação: um sistema de software inteligente suportado por rede neural artificial e algoritmo de otimização de vórtice. *Computer Applications in Engineering Education*, v. 25, 2017. Citado na página 28.

KRAMER, O. Scikit-learn. In: *Machine learning for evolution strategies*. [S.l.]: Springer, 2016. p. 45–53. Citado na página 63.

KUMAR, N. K. *et al.* An ensemble deep learning method for diabetes mellitus. In: *Emerging Technologies in Data Mining and Information Security*. [S.l.]: Springer, 2021. p. 75–83. Citado na página 25.

LEE, J.; KIM, S. Structural damage detection in the frequency domain using neural networks. *Journal of Intelligent Material Systems and Structures*, Sage Publications Sage UK: London, England, v. 18, n. 8, p. 785–792, 2007. Citado na página 32.

- LIANG, C. Coupled electromechanical analysis of adaptive material system-determination of actuator power consumption and system energy transfer. *Journal of Intelligent Material Systems and Structures*, v. 5, p. 21–20, 1994. Citado 2 vezes nas páginas 20 e 23.
- MA, M.; SUN, C.; CHEN, X. Deep coupling autoencoder for fault diagnosis with multimodal sensory data. *IEEE Transactions on Industrial Informatics*, IEEE, v. 14, n. 3, p. 1137–1145, 2018. Citado 3 vezes nas páginas 15, 16 e 25.
- MAIO, C. E. B. *Técnicas para monitoramento de integridade estrutural usando sensores e atuadores piezoelétricos*. Tese (Doutorado) — Universidade de São Paulo, 2011. Citado na página 21.
- MARCHI, M. Aplicação do método de impedância eletromecânica na detecção da queima no processo de retificação plana. Universidade Estadual Paulista (UNESP), 2015. Citado 2 vezes nas páginas 22 e 23.
- MASSOUD, M. Impedance methods for machine analysis: modal parameters extraction techniques. *The Shock and vibration digest*, Sage, v. 16, n. 5, p. 5–14, 1984. Citado na página 19.
- MCCULLOCH WARREN S E PITTS, W. Um cálculo lógico das idéias imanentes na atividade nervosa. v. 5, 1943. Citado na página 30.
- MENEZES, F. C. d. *et al.* Redes neurais artificiais aplicadas ao processo de coagulação. *Engenharia Sanitaria e Ambiental*, SciELO Brasil, v. 14, p. 449–454, 2009. Citado na página 26.
- MHATRE, N. *et al.* Tree crickets optimize the acoustics of baffles to exaggerate their mate-attraction signal. *Elife*, eLife Sciences Publications Limited, v. 6, p. e32763, 2017. Citado 2 vezes nas páginas 26 e 35.
- MOHEIMANI, S. R. A survey of recent innovations in vibration damping and control using shunted piezoelectric transducers. *IEEE transactions on control systems technology*, IEEE, v. 11, n. 4, p. 482–494, 2003. Citado na página 21.
- MOHSEN, H. *et al.* Classification using deep learning neural networks for brain tumors. *Future Computing and Informatics Journal*, Elsevier, v. 3, n. 1, p. 68–71, 2018. Citado na página 26.
- MOURA Jr., J. dos Reis Vieira de *et al.* Uma contribuição aos sistemas de monitoramento de integridade estrutural aplicada a estruturas aeronáuticas e espaciais. Universidade Federal de Uberlândia, 2008. Citado 9 vezes nas páginas 17, 18, 19, 20, 21, 22, 23, 30 e 33.
- MOURA Jr., J. dos Reis Vieira de; STEFFEN Jr., V. Impedance-based health monitoring: frequency band evaluation. In: *Proceedings of the 22nd International Modal Analysis Conference (IMAC XXII)*, Detroit, Michigan, USA. [S.l.: s.n.], 2004. Citado 2 vezes nas páginas 18 e 23.
- MULLAINATHAN, S.; SPIESS, J. Machine learning: an applied econometric approach. *Journal of Economic Perspectives*, v. 31, n. 2, p. 87–106, 2017. Citado na página 25.
- NUNES, R. T. M. Cálculo preditivo de classificação das notas do enade utilizando redes neurais artificiais. *Revista de Tecnologia Aplicada*, v. 7, n. 2, 2018. Citado na página 28.

NWANKPA, C. *et al.* Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018. Citado na página 33.

PALOMINO, L. V. *et al.* Análise das métricas de dano associadas à técnica da impedância eletromecânica para o monitoramento de integridade estrutural. Universidade Federal de Uberlândia, 2008. Citado na página 22.

PARK, G.; INMAN, D. J. Impedance-based structural health monitoring. *Damage Prognosis for Aerospace, Civil and Mechanical System*, Wiley Online Library, 2005. Citado na página 22.

PATTERSON, D. W. *Redes neurais artificiais: teoria e aplicações*. [S.l.: s.n.], 1998. Citado na página 35.

PEREZ, L.; WANG, J. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017. Citado na página 50.

PROVOST, F.; KOHAVI, R. Glossary of terms. *Journal of Machine Learning*, v. 30, n. 2-3, p. 271–274, 1998. Citado 2 vezes nas páginas 25 e 26.

RADFORD, A.; METZ, L.; CHINTALA, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. Citado 4 vezes nas páginas 42, 43, 44 e 46.

RAJU, V. *Implementing impedance-based health monitoring*. Tese (Doutorado) — Virginia Tech, 1997. Citado 3 vezes nas páginas 19, 20 e 21.

RAUBER, T. W. Redes neurais artificiais. *Universidade Federal do Espírito Santo*, p. 29, 2005. Citado 2 vezes nas páginas 27 e 28.

Renato Candido. *Generative Adversarial Networks: Build Your First Models*. 2020. Disponível em: <<https://realpython.com/generative-adversarial-networks/>>. Acesso em: 01 novembro 2021. Citado 4 vezes nas páginas 45, 46, 47 e 48.

REZENDE, S. W. F. d. *et al.* Técnicas de aprendizado profundo aplicadas ao monitoramento de integridade estrutural por impedância eletromecânica. Universidade Federal de Goiás, 2021. Citado 6 vezes nas páginas 30, 31, 32, 34, 35 e 52.

ROSENBLATT, F. Perceptron simulation experiments. *Proceedings of the IRE, IEEE*, v. 48, n. 3, p. 301–309, 1960. Citado na página 31.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. *Learning internal representations by error propagation*. [S.l.], 1985. Citado 3 vezes nas páginas 32, 33 e 37.

SALAKHUTDINOV, R. Learning deep generative models. *Annual Review of Statistics and Its Application*, Annual Reviews, v. 2, p. 361–385, 2015. Citado na página 33.

SANTOS, H. L. d. Minerando dados de ambientes virtuais de aprendizagem para predição de desempenho de estudantes. 2012. Citado na página 40.

SAYLER, K. M.; HARRIS, L. A. *Deep fakes and national security*. [S.l.], 2020. Citado na página 49.

SCHMIDHUBER, J.; HOCHREITER, S. *et al.* Long short-term memory. *Neural Comput*, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 32.

- SHAO, S. *et al.* Highly accurate machine fault diagnosis using deep transfer learning. *IEEE Transactions on Industrial Informatics*, IEEE, v. 15, n. 4, p. 2446–2455, 2018. Citado na página 15.
- SILVA, I. N. D.; SPATTI, D. H.; FLAUZINO, R. A. Redes neurais artificiais para engenharia e ciências aplicadas curso prático. *São Paulo: Artliber*, 2010. Citado na página 39.
- SINGH, R. *et al.* Prediction of geomechanical parameters using soft computing and multiple regression approach. *Measurement*, Elsevier, v. 99, p. 108–119, 2017. Citado na página 26.
- STASZEWSKI, W.; LEE, B.; TRAYNOR, R. Fatigue crack detection in metallic structures with lamb waves and 3d laser vibrometry. *Measurement Science and Technology*, IOP Publishing, v. 18, n. 3, p. 727, 2007. Citado na página 19.
- SUN, F. P. *et al.* Truss structure integrity identification using pzt sensor-actuator. *Journal of Intelligent material systems and structures*, TECHNOMIC PUBLISHING CO., INC. 851 New Holland Ave., Box 3535, Lancaster, PA . . . , v. 6, n. 1, p. 134–139, 1995. Citado na página 23.
- VALENÇA, M. J. Aplicando redes neurais: um guia completo. *Livro Rápido, Olinda*, 2005. Citado na página 28.
- WANG, H.-L. *et al.* Dorsal raphe dual serotonin-glutamate neurons drive reward by establishing excitatory synapses on vta mesoaccumbens dopamine neurons. *Cell reports*, Elsevier, v. 26, n. 5, p. 1128–1142, 2019. Citado na página 35.
- YANG, H.; ZHANG, Z.; YIN, L. Identity-adaptive facial expression recognition through expression regeneration using conditional generative adversarial networks. In: IEEE. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. [S.l.], 2018. p. 294–301. Citado na página 51.
- YANG, W. *et al.* Towards rich feature discovery with class activation maps augmentation for person re-identification. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 1389–1398. Citado na página 40.
- YU, H.; DAI, Z. Snnrice6ma: a deep learning method for predicting dna n6-methyladenine sites in rice genome. *Frontiers in genetics*, Frontiers, v. 10, p. 1071, 2019. Citado na página 26.
- ZANETTI, S. S. *et al.* Estimação da evapotranspiração de referência no estado do rio de janeiro usando redes neurais artificiais. *Revista Brasileira de Engenharia Agrícola e Ambiental*, SciELO Brasil, v. 12, n. 2, p. 174–180, 2008. Citado na página 28.
- ZHANG, P. Multiple imputation: theory and method. *International Statistical Review/Revue Internationale de Statistique*, JSTOR, p. 581–592, 2003. Citado 2 vezes nas páginas 39 e 40.
- ZHAO, R. *et al.* Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, Elsevier, v. 115, p. 213–237, 2019. Citado 2 vezes nas páginas 15 e 39.