



DESENVOLVIMENTO DE REDES NEURAIS ARTIFICIAIS PARA  
CLASSIFICAÇÃO DE ENSAIOS NÃO DESTRUTIVOS DE ULTRASSOM EM  
PVDF

Guilherme Lopes Londres

Dissertação apresentada ao Programa de Pós-graduação em Engenharia Metalúrgica e de Materiais, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Metalúrgica e de Materiais.

Orientador: Gabriela Ribeiro Pereira

Rio de Janeiro  
Setembro de 2018

DESENVOLVIMENTO DE REDES NEURAIAS ARTIFICIAIS PARA  
CLASSIFICAÇÃO DE ENSAIOS NÃO DESTRUTIVOS DE ULTRASSOM EM  
PVDF

Guilherme Lopes Londres

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

Examinada por:

---

Prof.<sup>a</sup> Gabriela Ribeiro Pereira, D.Sc.

---

Prof. Celio Albano da Costa Neto, Ph.D.

---

Prof. Sergio Damasceno Soares, D.Sc.

RIO DE JANEIRO, RJ – BRASIL  
SETEMBRO DE 2018

Londres, Guilherme Lopes

Desenvolvimento de Redes Neurais Artificiais para Classificação de Ensaios Não Destrutivos de Ultrassom em PVDF/Guilherme Lopes Londres. – Rio de Janeiro: UFRJ/COPPE, 2018.

XV, 100 p.: il.; 29,7cm.

Orientador: Gabriela Ribeiro Pereira

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Metalúrgica e de Materiais, 2018.

Referências Bibliográficas: p. 81 – 86.

1. Redes Neurais. 2. Ultrassom. 3. PVDF. I. Pereira, Gabriela Ribeiro. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Metalúrgica e de Materiais. III. Título.

*Para o meu amigo Prof. Bastian  
(in memoriam), também mestre  
e mentor. Que precisou sair  
mais cedo, certamente para  
ajudar alguém, porque era isso  
que ele sempre estava fazendo.*

# Agradecimentos

Agradeço primeiramente à minha mãe, Lúcia, e meu pai, Manoel, por terem me apoiado e incentivado mais uma vez. Não bastasse o longo caminho da graduação, o mestrado também apresentou suas pedras.

À minha namorada, Virgínia Bertolo, que sempre esteve do meu lado me apoiando. Me levando à razão quando essa talvez estivesse escapando, me acompanhando nas aventuras dessa vida. O velho mundo nos espera agora, vamos em frente!

À minha família, que sempre serviu de suporte, sempre apoiou e comemorou comigo. Vocês também têm uma parcela nessa conquista.

Ao amigo Rafael Cidade, que já foi orientador e agora caça moedas virtuais do outro lado do mundo. Mesmo longe geograficamente, sempre esteve disponível para me ajudar. Falando coisas que não entendo, mas, que vejo com boa vontade.

Ao mentor e também grande amigo, Professor Bastian, que infelizmente teve de sair mais cedo. À Dona Vera e Eduardo, esposa e filho do professor, que me apoiaram e auxiliaram nas aplicações para meu novo caminho.

Ao grande irmão, Gustavo Cabral, sempre presente para o que for preciso. E para o que não for preciso também.

À amiga Natalie Siqueira, que me recebeu ao cair de paraquedas no LNDC, e virou companheira de trabalho e conversas. Dá o gás, Natalie! Muita coisa boa te espera. E pega mais leve nas correções...

À professora Gabriela Pereira, orientadora desse trabalho, que me deu a oportunidade de terminar o mestrado e me apoiou quando resolvi tentar a vida em outro país.

Ao professor Celio Costa, membro da banca, orientador por 2 semanas, mas que

ajudou muito quando conversou comigo.

Aos professores João Marcos e Damasceno, pela gentileza em participarem da banca deste trabalho.

À Netflix, sempre presente.

A todos, meu muito obrigado!

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

DESENVOLVIMENTO DE REDES NEURAIAS ARTIFICIAIS PARA  
CLASSIFICAÇÃO DE ENSAIOS NÃO DESTRUTIVOS DE ULTRASSOM EM  
PVDF

Guilherme Lopes Londres

Setembro/2018

Orientador: Gabriela Ribeiro Pereira

Programa: Engenharia Metalúrgica e de Materiais

A utilização de ensaios não destrutivos (END) de ultrassom (UT) é amplamente difundido na indústria. A aplicação desta técnica em polímeros é, porém, quase inexistente devido à dificuldade na interpretação dos resultados. O polifluoreto de vinilideno (PVDF) tem, dentre inúmeras aplicações, sido utilizado como camada de estanqueidade em risers na indústria do petróleo. Um dos problemas é justamente a inspeção por meio de ensaios não destrutivos deste material. Um efeito presente em alguns polímeros é denominado *whitening*, que decorre da deformação desse material, e, dessa forma, pode ser utilizado como um indicador de defeito. Visando uma solução para este problema, é proposta neste trabalho a utilização de inteligência artificial na classificação e detecção de *whitening* em corpos de prova de PVDF inspecionados por UT. Os dados obtidos foram analisados e processados, escolhendo-se utilizar apenas a porção retroespalhada do sinal. Foram utilizadas as transformadas de Fourier e discreta dos cossenos, além do filtro de Savitzky-Golay no processamento dos sinais antes da alimentação da rede. A técnica utilizada é a de redes neurais artificiais como um perceptron multicamadas. Foi feito o treinamento fazendo uso da linguagem *Python* e suas bibliotecas de aprendizado de máquina, como *TensorFlow* e *Keras*, resultando numa rede classificadora com mais de 95% de acertos.

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

DEVELOPMENT OF ARTIFICIAL NEURAL NETWORKS FOR  
CLASSIFICATION OF NON-DESTRUCTIVE ULTRASOUND TESTS IN PVDF

Guilherme Lopes Londres

September/2018

Advisor: Gabriela Ribeiro Pereira

Department: Metallurgical and Materials Engineering

The use of nondestructive (NDT) ultrasound (UT) test is widely diffused in industry. The application of this technique to polymers is, however, almost non-existent due to the difficulty in interpreting the results. Polyvinylidene polyfluoride (PVDF) has, among many applications, been used as a sealing layer in risers in the petroleum industry. One of the problems is precisely the inspection by non-destructive testing of this material. A certain polymers phenomenon is called whitening, which follows the deformation of the material, and thus can be used as a fault indicator. Aiming for a solution to this problem, it is proposed in this work the use of artificial intelligence in the classification and detection of whitening in PVDF specimen inspected by UT. The obtained data were analyzed and processed, choosing to use only the backscattered portion of the signal. The Fourier and discrete cosine transforms were used, in addition to the Savitzky-Golay filter in the processing of the signals before the feeding the network. The technique used is that of artificial neural networks as a multilayer perceptron. The training was done using the Python language and its machine learning libraries, such as TensorFlow and Keras, resulting in a classification network with more than 95% of accuracy.

# Sumário

<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>Lista de Símbolos</b>	<b>xiv</b>
<b>Lista de Abreviaturas</b>	<b>xv</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivo . . . . .	2
<b>2 Revisão Bibliográfica</b>	<b>4</b>
2.1 <i>Risers</i> . . . . .	4
2.2 Polifluoreto de vinilideno - (PVDF) . . . . .	5
2.2.1 Cavitação . . . . .	7
2.2.2 <i>Whitening</i> . . . . .	8
2.3 Ensaaios Não Destrutivos . . . . .	9
2.3.1 Ultrassom - UT . . . . .	10
2.4 Processamento de Sinais . . . . .	13
2.4.1 Transformada de Fourier . . . . .	13
2.4.2 Transformada Discreta dos Cossenos - DCT . . . . .	14
2.4.3 Filtro de Savitzky-Golay . . . . .	15
2.5 Inteligência Artificial . . . . .	15
2.6 Redes Neurais Artificiais . . . . .	15
2.6.1 Elementos da rede . . . . .	17
2.6.2 Treinamento . . . . .	20
2.6.3 Extração de Características . . . . .	26
2.7 Estado da Arte . . . . .	26
<b>3 Materiais e Métodos</b>	<b>33</b>
3.1 Metodologia . . . . .	33
3.2 Fabricação dos Corpos de Prova - PVDF . . . . .	34

3.3	Ensaio de Tração . . . . .	35
3.4	Ensaio de UT . . . . .	37
3.5	Processamento de Sinais . . . . .	39
3.6	Desenvolvimento da RNA . . . . .	40
<b>4</b>	<b>Resultados e Discussão</b>	<b>42</b>
4.1	Ensaio de UT e Tração . . . . .	42
4.2	Redes Neurais . . . . .	45
4.2.1	Análise Exploratória dos Dados . . . . .	45
4.2.2	Divisão das Classes para Treinamento . . . . .	48
4.2.3	Processamento de Sinais . . . . .	48
4.2.4	Topologia das Redes . . . . .	50
4.2.5	Resultados das Classificações . . . . .	51
4.2.6	Classificação com <i>A-Scan</i> . . . . .	52
4.2.7	Classificação com FFT . . . . .	55
4.2.8	Classificação com DCT . . . . .	59
4.2.9	Comparação das Redes . . . . .	62
4.3	Estudos de Casos . . . . .	63
4.3.1	Caso 01 - Particionamento Forçado . . . . .	63
4.3.2	Caso 02 - Outros Corpos de PVDF . . . . .	67
4.3.3	Caso 03 - <i>Polyfit</i> . . . . .	72
4.4	Considerações Finais . . . . .	77
<b>5</b>	<b>Conclusão</b>	<b>79</b>
<b>6</b>	<b>Sugestões para Trabalhos Futuros</b>	<b>80</b>
	<b>Referências Bibliográficas</b>	<b>81</b>
<b>A</b>	<b>Códigos</b>	<b>87</b>

# Lista de Figuras

2.1	Exemplo de <i>riser</i> . . . . .	5
2.2	Polimerização do PVDF . . . . .	6
2.3	Esferulito - Micrografia . . . . .	6
2.4	Exemplo do ensaio de UT. . . . .	11
2.5	Exemplo do ensaio de UT em imersão. . . . .	12
2.6	Exemplo de Transformada de Fourier . . . . .	14
2.7	Exemplo de classe não linearmente separável. . . . .	16
2.8	Exemplo de RN. . . . .	17
2.9	Formato da curva da ReLU. . . . .	19
2.10	Exemplo de MLP. . . . .	22
3.1	Fluxograma das etapas do projeto. . . . .	33
3.2	Molde e dimensões dos corpos de prova. . . . .	34
3.3	Fluxograma das etapas de degasagem e carregamento. . . . .	35
3.4	Corpos de prova de PVDF. . . . .	36
3.5	Sistema automatizado para ensaio de UT em imersão. . . . .	38
3.6	Pontos de inspeção do UT no corpo de prova. . . . .	39
3.7	Sinal retroespalhado. . . . .	40
4.1	Curva de Tensão x Deformação. . . . .	44
4.2	Comparação de <i>A-scan</i> . . . . .	47
4.3	Comparação de sinais filtrados. . . . .	49
4.4	Comparação de sinais filtrados. . . . .	50
4.5	Topologia das Redes. . . . .	51
4.6	Estatísticas do treinamento da rede. . . . .	53
4.7	Matriz confusão - 2 classes. . . . .	54
4.8	Matriz confusão - 4 classes. . . . .	54
4.9	Espectro FFT. . . . .	56
4.10	Estatísticas do treinamento da rede. . . . .	57
4.11	Matriz confusão - 2 classes. . . . .	58
4.12	Matriz confusão - 4 classes. . . . .	58

4.13	Espectro DCT. . . . .	59
4.14	Estatísticas do treinamento da rede. . . . .	60
4.15	Matriz confusão - 2 classes. . . . .	61
4.16	Matriz confusão - 4 classes. . . . .	61
4.17	Estatísticas do treinamento da rede. . . . .	64
4.18	Matriz confusão - 2 classes. . . . .	65
4.19	Matriz confusão - 4 classes. . . . .	65
4.20	Matriz confusão - Excluídos - 2 classes. . . . .	66
4.21	Matriz confusão - Excluídos - 4 classes. . . . .	67
4.22	Matriz confusão NR1 - 4 classes. . . . .	68
4.23	Matriz confusão NR2 - 2 classes. . . . .	69
4.24	Matriz confusão NR2 - 4 classes. . . . .	69
4.25	Matriz confusão - DCT - NR1 - 4 classes. . . . .	70
4.26	Matriz confusão DCT - NR2 - 2 classes. . . . .	71
4.27	Matriz confusão - DCT - NR2 - 4 classes. . . . .	71
4.28	Estatísticas do treinamento da rede - ReLU x <i>tanh</i> . . . . .	73
4.29	Estatísticas do treinamento da rede - ReLU x <i>tanh</i> . . . . .	74
4.30	Matriz confusão - ReLU - 2 classes. . . . .	75
4.31	Matriz confusão - <i>tanh</i> - 2 classes. . . . .	75
4.32	Matriz confusão - ReLU - 4 classes. . . . .	76
4.33	Matriz confusão - <i>tanh</i> - 4 classes. . . . .	76

# Lista de Tabelas

2.1	Resumo de artigo de <i>whitening</i> . . . . .	9
2.2	Resumo de artigos - UT e RNs . . . . .	32
3.1	Relação dos corpos de prova. . . . .	37
4.1	Relação dos corpos de prova. . . . .	44
4.2	Velocidade do som calculada. . . . .	45
4.3	Divisão do conjunto de dados. . . . .	46
4.4	Divisão das classes para treinamento e quantidade de dados disponíveis. . . . .	48
4.5	Propriedades das Redes Neurais. . . . .	51
4.6	Número de neurônios nas redes. . . . .	51
4.7	Escala da qualidade de classificação. . . . .	52
4.8	Comparação da classificação das redes. . . . .	62
4.9	Corpos de Prova Retirados . . . . .	63
4.10	Comparação de classificações . . . . .	77

# Lista de Símbolos

$D$	diâmetro efetivo do transdutor
$F$	função objetivo
$N$	tamanho do campo próximo
$V$	velocidade do som no material
$Z$	impedância acústica
$\alpha$	passo de treinamento
$\lambda$	comprimento de onda
$\rho$	densidade do material
$\tilde{y}_n$	saída $n$
$\varphi(\cdot)$	função de ativação
$b_0^k$	bias da camada $k$
$f$	frequência da onda
$u_n^k$	resultado do combinador linear do neurônio $n$ da camada $k$
$v$	velocidade da onda
$v_n^k$	potencial de ativação do neurônio $n$ da camada $k$
$w_{ni}^k$	peso da sinapse da camada $k$ entre o neurônio $i$ e o neurônio $n$
$x_n$	entrada $n$

# Lista de Abreviaturas

DCT	Transformada Discreta dos Cossenos
DFT	Transformada Discreta de Fourier
DWT	Transformada Discreta de Wavelet
FFT	Transformada Rápida de Fourier
LNDC	Laboratório de Ensaio Não Destrutivos, Corrosão e Soldagem
LPCM	Laboratório de Processamento e Caracterização de Materiais
MLP	Perceptron Multicamadas
PP	Polipropileno
PROPMEC	Laboratório de Propriedades Mecânicas
PVDF	Polifluoreto de vinilideno
RNA	Rede Neural Artificial
RN	Rede Neural
UT	Ultrassom
XOR	Função ou-exclusivo
hiPP	Polipropileno de alto impacto

# Capítulo 1

## Introdução

Ensaaios não destrutivos (END) é o nome dado ao conjunto de técnicas utilizadas para inspeção de materiais de forma que não haja dano no corpo testado, ou, que seja o menor possível e não permanente. Isto é, ao final do ensaio, a peça testada continuará em condições de uso. As técnicas de END podem ser empregadas na determinação de propriedades de materiais, detecção, caracterização e localização de defeitos e dimensionamento de peças, por exemplo.

Sua utilização na indústria é amplamente difundida. Dentre os inúmeros sistemas de teste, discutiremos a técnica de ultrassom (UT) cuja uma das aplicações é na inspeção de *risers* flexíveis, empregados no setor de óleo e gás. São dutos compostos por diferentes camadas de materiais: metais, compósitos e polímeros, usados normalmente para o transporte do petróleo do poço até a plataforma [1].

Os *risers* são geralmente inspecionados por UT [2], porém, na avaliação da camada polimérica o método não é aplicado. As respostas obtidas pelo ensaio são muito sutis, demandando muito tempo e experiência para que sejam analisadas por um operador. Além disso, são escassas ou inexistentes as referências na literatura para a interpretação dos resultados de UT em polímeros, conforme apresentado em [3], que realizou um primeiro estudo para obtenção de parâmetros em amostras de polifluoreto de vinilideno (PVDF), polímero utilizado em *risers*.

Atualmente, para que seja realizada uma análise da camada polimérica, é necessário que sejam removidas as camadas mais externas às de polímero [2]. Então

faz-se um reparo no duto e recoloca-se as camadas.

Um fenômeno que ocorre com o PVDF é o *whitening*, que é quando uma região do material torna-se mais opaca, esbranquiçada. Esse efeito acontece quando o material é submetido à uma tensão trativa de módulo quase coincidente com sua capacidade máxima de carregamento [4]. Em outras palavras, o aparecimento dessas áreas esbranquiçadas pode, dependendo da severidade, servir como indicador de que o material está prestes a colapsar.

O *whitening* é uma consequência da formação de microcavidades na estrutura do polímero, e dependendo do grau de severidade pode ser observado a olho nu. Utilizando corpos de prova de polímeros que foram submetidos a ensaios de tração até deformações pré-determinadas, e corpos não ensaiados, será possível realizar ensaios de UT. Em seguida, pode-se extrair e combinar características diversas dos sinais obtidos, fazendo uso de técnicas de processamento de sinais como a transformada de Fourier para o treinamento de uma rede neural.

Este trabalho tem como proposta dar continuidade ao que foi iniciado por NASCIMENTO FILHO, E. S. [3], dessa vez utilizando inteligência artificial, no caso, redes neurais artificiais, nos *A-Scan* de ensaios de UT em polímeros; com o objetivo de criar um classificador capaz de detectar de forma autônoma a presença ou não do efeito *whitening*. Dessa forma, tornando possível o ensaio de UT nos polímeros utilizados em risers.

## 1.1 Objetivo

Em sua dissertação de mestrado, Nascimento Filho, E. S. [3], utilizou diversas técnicas nos dados obtidos com ensaios de ultrassom em amostras de PVDF. Os resultados mostraram que é possível caracterizar o aparecimento do *whitening* no polímero, em estágios bem anteriores ao limite de escoamento.

Dando continuidade ao estudo iniciado em [3], este trabalho tem a proposta de utilizar técnicas de inteligência artificial, mais especificamente, redes neurais artificiais nos dados obtidos pelo ensaio de UT no mesmo polímero (PVDF), de forma a

se criar um classificador automático, capaz de indicar a presença ou não do efeito *whitening* no material inspecionado.

# Capítulo 2

## Revisão Bibliográfica

### 2.1 *Risers*

*Risers* são dutos, que podem ser rígidos ou flexíveis, utilizados na prospecção de petróleo. Esses tubos são compostos por várias camadas de materiais distintos, cada uma com uma função específica, sendo compostas por materiais poliméricos e metálicos.

Observa-se na Figura 2.1 um exemplo de *riser*: composto por 5 camadas, sendo 3 metálicas e 2 poliméricas. Entre as poliméricas estão a camada externa (revestimento + isolamento), que serve para proteger o duto contra corrosão e abrasão que possam ser causados pelo ambiente externo. A outra camada de polímero é onde normalmente se emprega o PVDF, a barreira de pressão; sua função é manter a estanqueidade do duto, evitando vazamentos e diminuir o atrito entre a carcaça e a armadura de pressão, ambas metálicas. Além da resistência mecânica, é importante que seu material seja quimicamente inerte<sup>1</sup> e resistente à temperatura de operação, já que muitas vezes os fluidos transportados são aquecidos.

Nas camadas metálicas: a armadura de tração tem por objetivo suportar esforços trativos no qual o duto é submetido. Sua estrutura é formada por tiras metálicas posicionadas de forma helicoidal. A armadura de pressão, por sua vez, auxilia a

---

<sup>1</sup>Diz-se que é quimicamente inerte o material que não é afetado, ou dificilmente altera suas propriedades, quando exposto a um agente químico.

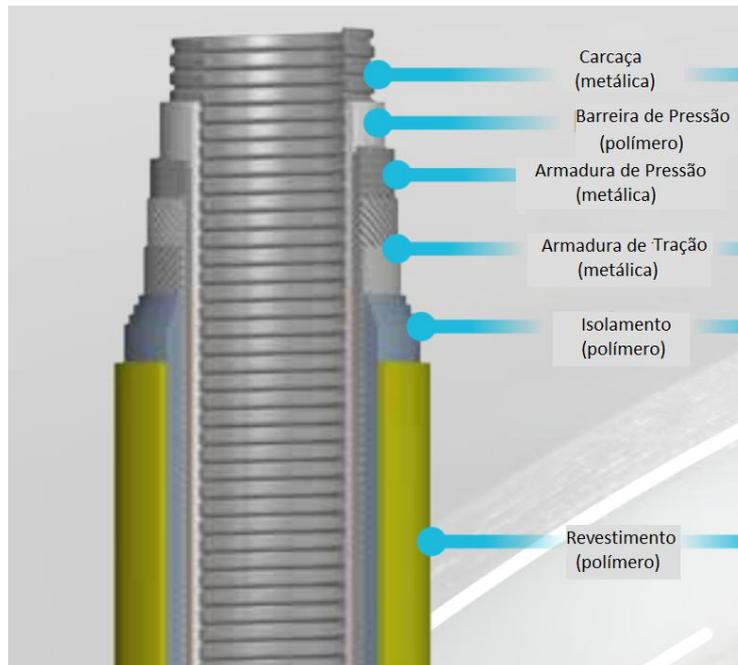


Figura 2.1: Exemplo de *riser*. Adaptado de [1].

carcaça a suportar os esforços compressivos. Finalmente, a carcaça, camada mais interna e com contato direto com o fluido tem por objetivo suportar as extremas pressões externas e também as pressões internas exercidas pelo fluido.

## 2.2 Polifluoreto de vinilideno - (PVDF)

O polifluoreto de vinilideno (PVDF), é um polímero linear com grau de cristalinidade entre 40% e 60%, classificado como semicristalino. É obtido pelo processo de polimerização por adição (poliadição) via radical livre do fluoreto de vinilideno, conforme mostrado na Figura 2.2. Característica do processo de poliadição, o PVDF apresenta alto grau de polimerização, com 2000 unidades repetitivas e massa molar entre 60 e 70 Kg/mol [5–8].

Uma característica muito importante do PVDF é a combinação de suas propriedades piro e piezoelétricas com a facilidade de processamento, resistência mecânica e ser quimicamente inerte. Essa peculiaridade faz com que o PVDF tenha diversas aplicações, tais como construção de transdutores, sensores, detectores, membranas poliméricas, aplicações biomédicas como enxerto ósseo, entre outras [6].

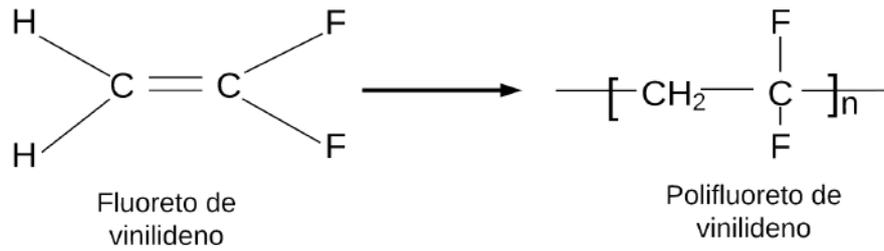


Figura 2.2: Reação de polimerização do PVDF

A cristalização do PVDF ocorre com o resfriamento controlado, lento, do polímero fundido. Dessa forma, os cristais vão sendo gerados como uma estrutura denominada esferulítica. Esferulitos são agregados esféricos de monocristais lamelares. Esses monocristais crescem no sentido radial à partir de um núcleo comum e são interligados pela fase amorfa. O tamanho dos esferulitos tem ligação direta com as propriedades do material. Uma maneira de se verificar a presença dessa estrutura é pela técnica de microscopia óptica de luz polarizada, conforme mostrado na Figura 2.3 [5, 6, 8, 9].

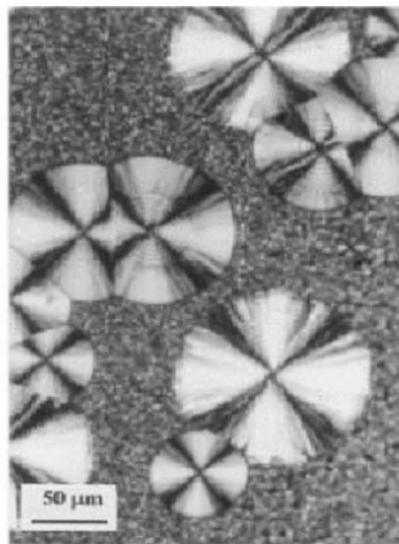


Figura 2.3: Esferulito visto numa micrografia óptica de luz polarizada. Retirado de [5]

Outra característica importante do PVDF é denominada polimorfismo. Dependendo das condições de processamento, principalmente a temperatura e a presença ou não de campo elétrico, o material irá apresentar 4 diferentes fases:  $\alpha$ ,  $\beta$ ,  $\gamma$  ou  $\delta$

[5, 6, 9].

$\alpha$  é a fase mais comum e é apolar, obtida com o super-resfriamento do polímero fundido. Confere ao material maior resistência mecânica e, a partir desta fase, realizando-se tratamentos térmicos, mecânicos ou elétricos, obtêm as demais.

$\beta$  é uma fase polar, obtida ao se aplicar um esforço mecânico na fase  $\alpha$ . Suas principais características são a piro e piezoeletricidade.

$\gamma$  também é uma fase polar, obtida a partir do polímero fundido, porém realizando-se a cristalização em elevada temperatura ( $> 160^{\circ}C$ ) por um longo período de tempo ( $> 6h$ ). A principal característica é uma temperatura de fusão maior do que as outras fases.

$\delta$  tem a mesma estrutura da fase  $\alpha$ , porém é polar. É obtida de forma semelhante à  $\alpha$  mas sendo exposta a um campo elétrico de alta intensidade ( $\geq 150MV/m$ ).

Nas aplicações em *risers* deseja-se que a fase em maior quantidade seja a  $\alpha$ , devido à sua característica de elevada resistência mecânica. Como já citado, entre outras propriedades, o que faz o PVDF ser escolhido para utilização em dutos flexíveis é, além das propriedades mecânicas, ser quimicamente inerte, resistir bem ao envelhecimento e à elevadas temperaturas [5, 10].

### 2.2.1 Cavitação

O mecanismo de deformação do PVDF ocorre em vários estágios. De forma geral, quando submetido à uma tensão trativa, depois de alcançada a tensão limite de escoamento do material, começa a ocorrer o escoamento, que é uma deformação irreversível, definido por norma [11]. Em materiais frágeis, nesse momento há a formação de uma trinca, seguido da propagação da mesma e terminando com a ruptura do material. Num polímero dúctil, como o PVDF, pode haver o processo chamado de cavitação.

Na cavitação, há a formação de uma cavidade de escala micrométrica (ou até nanométrica), que, diferente de uma trinca, não coalesce. A microcavidade formada é envolta por fibrilas, que acabam por exercer um papel estrutural, sendo capaz de sustentar tensões e impedindo o crescimento dos vazios. Esse processo acaba alterando a densidade na região onde ocorre, o que implica numa falta de homogeneidade óptica, alterando os índices de refração no material, dando origem a uma faixa opaca [12, 13].

### **2.2.2 *Whitening***

Ao aparecimento da região esbranquiçada que surge no processo de deformação do polímero, dá-se o nome de efeito *whitening*. É importante ressaltar que esse fenômeno ocorre, geralmente, antes da ruptura do material - num caso em que a tensão trativa continua sendo aplicada, como num ensaio de tração. Outra importante característica é, dependendo da intensidade, ser identificado numa simples inspeção visual.

Não há uma definição sobre o motivo do aparecimento do *whitening*, porém diversos autores relacionam seu aparecimento com a formação das microcavidades, no processo de cavitação [3, 5, 14].

A seguir, na Tabela 2.1 é apresentado um resumo dos estudos publicados em alguns artigos:

Tabela 2.1: Resumo de artigos de *whitening*.

Autor	Material	Conclusão
Laiarinandrasana <i>et al.</i> [14]	PVDF	<i>Whitening</i> ocorre por causa da variação de volume e formação de microcavidades no material, percebido pelo aumento da porosidade.
Brusselle-Dupend <i>et al.</i> [15]	PVDF	A aparição do branqueamento caracteriza a cavitação, na região amorfa restrita sob tensão tratativa.
Koch e Seidler [16]	PVDF	Não conclui sobre a causa do <i>whitening</i>
Liu <i>et al.</i> [17]	PP isotático	Relaciona o aparecimento da região esbranquiçada à alta quantidade de microcavidades e vazios na estrutura do material e também à uma estrutura altamente fibrilar.
Liu <i>et al.</i> [18]	hiPP	Diz que o branqueamento ocorre na fase amorfa e é devido à fissuras na matriz do polímero ou formação de vazios na estrutura.

## 2.3 Ensaios Não Destrutivos

É denominada por ensaio não destrutivo a técnica de inspeção de um corpo que não provoca dano ao mesmo. Pode ser usada para localização e detecção de falhas ou para caracterizar um material que foi exposto a uma condição que difere da qual foi produzido originalmente.

### 2.3.1 Ultrassom - UT

A técnica de ensaio não destrutivo de ultrassom (UT), consiste na emissão de ondas sonoras em frequências maiores do que as audíveis por humanos, por isso denominadas ondas ultrassônicas. Ao se propagar pelo corpo que está sendo inspecionado, a onda irá percorrer um caminho sônico até que haja uma diferença na impedância acústica  $Z$ , que é resultante do produto da densidade do material  $\rho$  e da velocidade do som no material  $V$ .

$$Z = \rho \times V \quad (2.1)$$

Essa diferença na impedância  $Z$  fará com que a onda seja mais refletida ou menos refletida. Quando no interior de um corpo, essas diferenças ocorrem em descontinuidades na estrutura do material, que podem ser defeitos, porosidades ou quando se chega na extremidade do corpo. Dessa forma, toda vez que a onda encontrar uma borda ela sofrerá uma reflexão, que será detectada pelo cristal piezoelétrico usado para a produção e recepção das ondas ultrassônicas.

A leitura e interpretação do ensaio de UT é feito com auxílio de um osciloscópio que desenha curvas de intensidade *versus* tempo, denominada *A-Scan*. Também existem outras formas de curvas como *B-Scan*, *C-Scan* e *D-Scan*. Neste trabalho, utilizaremos apenas as curvas *A-Scan*. Um exemplo dessa curva pode ser observado na Figura 2.4, onde o ponto de maior intensidade é o pulso inicial, que ocorre assim que se faz o contato entre o transdutor e a superfície do corpo a ser inspecionado. Em seguida, a onda se propaga até a outra extremidade do corpo onde será refletida e, ao ser detectada pelo transdutor, será representada pelo segundo maior pico do gráfico, chamado de eco de fundo. Caso a onda encontre alguma descontinuidade no caminho, ocorrerá uma reflexão prematura, representada por um pico entre o pulso inicial e o eco de fundo, chamado de eco do defeito.

O esquema apresentado é do método de inspeção por UT chamado pulso-eco, um dos mais simples e utilizados. Como dito, as ondas ultrassônicas são geradas por um cristal piezoelétrico, localizado no interior do transdutor. Embora algumas

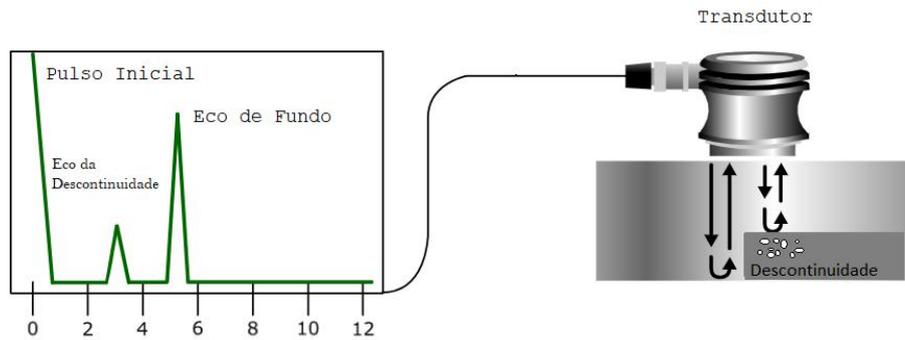


Figura 2.4: Exemplo do ensaio UT. Do lado esquerdo observa-se o espectro de um *A-Scan*, e do lado direito um transdutor utilizado para inspecionar uma placa com um defeito. Adaptado de [19].

superfícies pareçam lisas ao olhar, em escala microscópica podem haver inúmeros defeitos que resultam em regiões rugosas, o que atrapalha a transmissão das ondas ultrassônicas para o interior do material. Além disso, entre a cabeça do transdutor e a superfície a ser inspecionada geralmente há uma fina camada de ar, o que também dificulta ou impede a propagação das ondas devido à sua impedância acústica  $Z$ . Para que haja uma melhor transmissão das ondas geradas pelo cristal, utiliza-se algum tipo de acoplante, que costuma ser algum tipo de líquido ou gel, como óleos diversos, parafinas, água, mel, entre outros. O acoplante deverá ter  $Z$  semelhante ao do material, dessa forma garantindo a transmissão da onda do transdutor para o corpo a ser analisado.

### O Problema do Campo Próximo

Quando a onda ultrassônica é gerada pelo cristal piezoelétrico, ela não se origina de um único ponto, mas a partir de toda a superfície do cristal [19]. Assim é formado o feixe sônico, que pode ser dividido em três regiões principais: zona morta, campo próximo e campo distante. Na zona morta, não há detecção, por ser muito próxima ao transdutor, enquanto o cristal ainda está vibrando para gerar uma onda, já está recebendo uma reflexão e, assim, não consegue distinguir os sinais. No campo próximo, a interação entre as ondas é muito alta, resultando em muita interferência

tanto construtiva quanto destrutiva; é possível que haja detecção, mas não há como caracterizar o sinal obtido pois não há controle da energia que está chegando no detector. Já no campo distante, uma frente de onda única já foi formada e quase não há interferência, portanto é a região ideal para a inspeção. O comprimento  $N$  do campo próximo gerado por um transdutor pode ser calculado pela Lei de Fresnel (eq. 2.2), sabendo-se o diâmetro efetivo  $D$  do transdutor e o comprimento de onda  $\lambda$  da onda gerada ou então a frequência  $f$  e a velocidade da onda no material,  $v$ .

$$N = \frac{D^2}{4\lambda} = \frac{D^2 f}{4v} \quad (2.2)$$

Dependendo das dimensões do corpo a ser inspecionado, especialmente os de espessura delgada, é difícil fazer uma boa leitura devido ao fato do campo próximo estar dentro do material. Para contornar esse problema, utiliza-se uma maior camada de acoplante, o que dá origem a outro método de inspeção, o de imersão, representado na Figura 2.5.

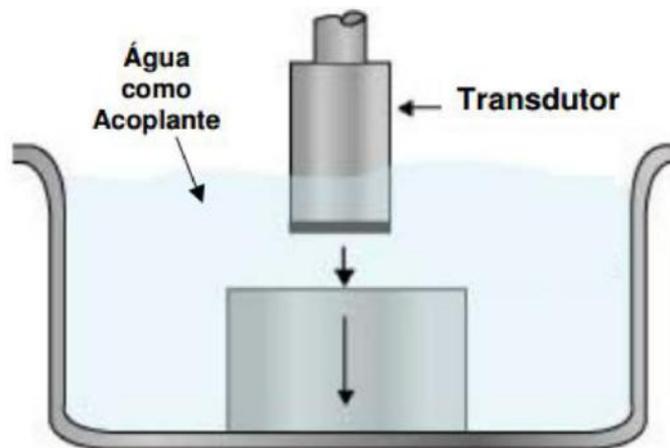


Figura 2.5: Exemplo do ensaio UT em imersão. Retirado de [3].

Nesse método o acoplante mais utilizado é água. Controlando a distância entre o transdutor e o corpo de prova, ou seja, determinando o tamanho da lâmina d'água, é possível ajustar o feixe sônico de forma a fazer com que apenas o campo distante fique no material [20]. Outra característica dessa técnica é não ter contato com o corpo de prova, o que facilita os processos de automação da inspeção.

## 2.4 Processamento de Sinais

Existe uma infinidade de métodos e algoritmos para tratamento de sinais disponíveis tanto na literatura quanto em pacotes prontos para ser utilizados em *Python* ou *MATLAB*, por exemplo. Entre os diversos algoritmos, a Transformada de Fourier é um dos mais utilizados e será brevemente descrito a seguir.

### 2.4.1 Transformada de Fourier

Uma transformada é um dispositivo matemático semelhante à uma função - que tem uma ou mais variáveis de entrada e uma variável de saída -, porém com múltiplas saídas, ao invés de uma única. A transformada de Fourier é a ferramenta que permite descrever qualquer sinal (ou função) contínuo periódico numa soma de senoides [21].

A forma mais simples e elementar de apresentação dos sinais obtidos no ensaio de UT, na tela do osciloscópio, é a curva *A-Scan* cujos eixos são intensidade *versus* tempo. Portanto, diz-se que este sinal de UT está no domínio do tempo. Em algumas medições em materiais com alta densidade pode ocorrer forte atenuação do sinal, dificultando análise. Outro complicador pode ser a espessura do corpo de prova que, se muito fino pode resultar na sobreposição dos ecos, prejudicando a leitura [22]. Uma forma de se contornar esses problemas é representar o sinal obtido no domínio das frequências, o que pode ser feito aplicando-se a transformada de Fourier [23].

A transformada de Fourier irá fazer a conversão do sinal do domínio temporal para o domínio das frequências, conforme pode-se observar na Figura 2.6. Explorando de maneira muito simples, a representação no domínio das frequências é como um resumo de todas as frequências e suas respectivas intensidades presentes no sinal obtido. Caso queira retornar ao sinal original de amplitude *versus* tempo, basta aplicar a inversa da transformada no novo sinal.

São quatro classificações distintas para a Transformada de Fourier, variando de acordo com as características do sinal de entrada - contínuo e aperiódico, contínuo

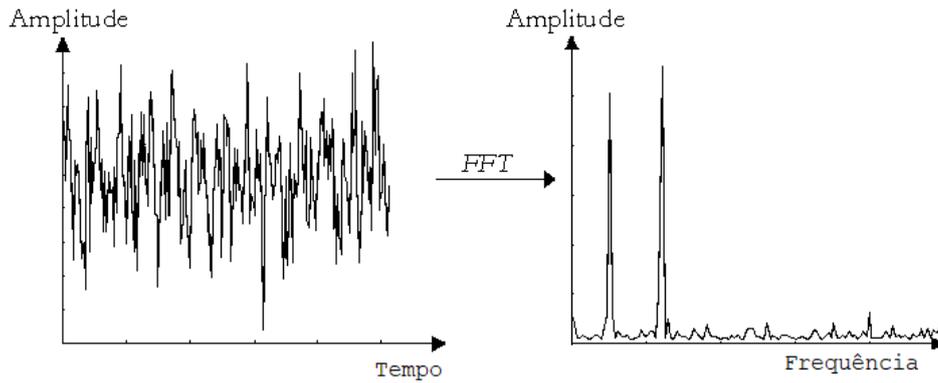


Figura 2.6: Exemplo da aplicação da transformada de Fourier em um sinal. Adaptado de [24].

e periódico, discreto e aperiódico e discreto e periódico. No caso dos sinais de ultrassom, quando captados são convertidos em sinais digitais, ou seja, sinais discretos. Como este tipo de onda se repete periodicamente, temos um sinal discreto e periódico, o que faz com que utilizaremos a transformada denominada transformada discreta de Fourier (DFT). Quando se fala em processamento de sinais, utiliza-se a transformada rápida de Fourier (FFT), um algoritmo muito eficiente que permite o cálculo numérico (computacional) da transformada de Fourier [21]. O detalhamento matemático da DFT ou do algoritmo da FFT não fazem parte do escopo deste trabalho e, portanto, não serão aqui abordados.

## 2.4.2 Transformada Discreta dos Cossenos - DCT

Outros método bastante empregado é a transformada discreta dos cossenos (DCT). O funcionamento e o princípio do DCT é muito semelhante à DFT também convertendo um sinal do domínio temporal para o das frequências. Porém, utiliza exclusivamente séries de cossenos, utiliza apenas números reais e é simétrico. Essa transformada tem extensa aplicação na compactação de imagens. O fato de utilizar apenas séries de cossenos é significativo nas aplicações de compressão pois demanda menos funções para representar um sinal [25, 26]

### 2.4.3 Filtro de Savitzky-Golay

Normalmente a captura e conversão de sinais analógicos para digitais envolve o aparecimento de ruídos. Dependendo da severidade, a leitura e utilização do sinal ficam prejudicadas. De forma de retirar essa perturbação, aplica-se um filtro. Um exemplo, é o filtro de Savitzky-Golay. Publicado em 1964 [27], num periódico de química analítica, o método proposto realiza um ajuste linear não-ponderado de mínimos quadrados fazendo uso de um polinômio de ordem a ser determinada pelo usuário. O filtro apresenta como resultado uma curva suavizada com significativo aumento do relação sinal-ruído [28].

## 2.5 Inteligência Artificial

Conforme reportado em [29], de 1996 até 2017 houve um aumento de 9x no número de artigos publicados anualmente sobre inteligência artificial em todo o mundo, junto a isso, o número de robôs utilizados na indústria global aumentou em 2,5x desde o ano 2000. Sistemas autônomos estão presentes cada vez mais no cotidiano das pessoas e das empresas; um exemplo da rápida evolução é quando se fala em visão computacional, as máquinas já superaram os humanos em 2015 [30, 31].

## 2.6 Redes Neurais Artificiais

Redes neurais artificiais (RNAs), ou simplesmente redes neurais (RNs), são um conjunto de técnicas computacionais inspiradas no funcionamento do cérebro humano, principalmente na sua capacidade de aprendizagem [32]. A estrutura cerebral humana é basicamente composta por neurônios interconectados pelo que se chama de sinapses. Assim, quando ocorre um estímulo no dendrito, com voltagem maior do que a necessária para haver a polarização do axônio, o impulso elétrico é então transmitido de um neurônio para outro.

O primeiro modelo neurônio artificial foi introduzido por McCulloch e Pitts em 1943 [33]. No início muito desenvolveu-se sobre o assunto até que boa parte dos

estudos estagnou devido à limitação daquela estrutura em separar classes não linearmente separáveis, observe a Figura 2.7. Esse tipo de classe é muito comum nas aplicações de eletrônica, é a chamada função ou-exclusivo (XOR).

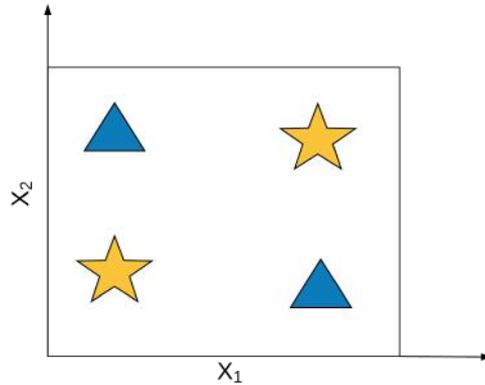


Figura 2.7: Exemplo de um caso de classe não linearmente separável. De forma bem simples, imagine, com apenas uma linha reta, separar as estrelas dos triângulos. Não é possível pois se trata de uma classe não linearmente separável. Para resolver esse problema, precisaremos de, ao menos, duas linhas distintas.

Em 1986 com o desenvolvimento do algoritmo de treinamento *backpropagation*, torna-se possível a separação desse tipo de classes, sendo possível a utilização de redes com mais de uma camada escondida, o que hoje chama-se perceptron multicamadas (MLP) [32]. As RN costumam ser utilizadas tanto como classificadoras, separando diferentes classes, quanto como aproximadoras, onde uma função é definida para descrever um conjunto de dados.

Agora, observe a Figura 2.8, o exemplo mais simples de uma rede neural: uma camada de entrada, uma camada escondida e uma saída. O neurônio na camada de entrada recebe um sinal  $x_n$  que é multiplicado por um peso sináptico  $w_{ni}^k$ <sup>2</sup> e então transmitido para a camada escondida, onde um combinador linear faz o somatório de todas as entradas já multiplicadas por seus respectivos pesos, isso inclui o *bias*  $b_0^k$ , que também foi multiplicado por um peso  $w_{ni}^k$ , resultando em  $v_n^k$ , usado como variável de entrada na função de ativação  $\varphi(v_n^k)$ . A saída da função de ativação será  $z_n^k$  que, nessa rede já será o sinal de saída  $\tilde{y}_n$ .

<sup>2</sup>Na notação  $w_{ni}^k$ , indica o peso  $w$  que faz a sinapse do neurônio  $i$  da camada  $k$  até o neurônio  $n$  na próxima camada.

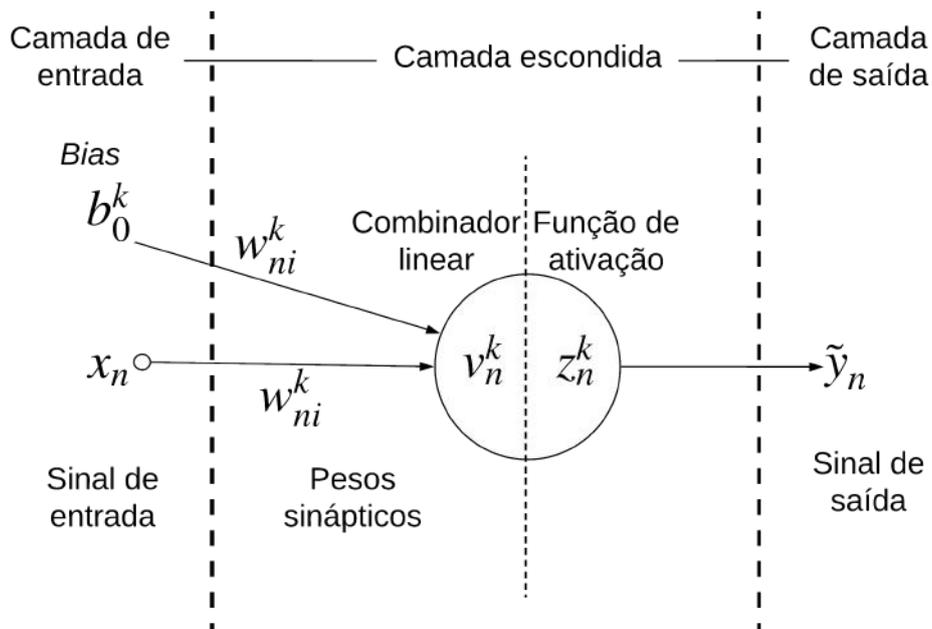


Figura 2.8: Exemplo de rede neural com uma entrada, uma camada escondida e uma saída.

Uma rede neural pode ter diversas entradas, camadas escondidas e saídas. A arquitetura da rede irá variar de acordo com a necessidade do problema a ser resolvido.

### 2.6.1 Elementos da rede

Tomando como base o resumo apresentado por Nunes em [34], apresenta-se nesta seção os principais elementos na estrutura de uma rede neural.

**Sinais de entrada** são os sinais recebidos para serem processados. Esses sinais costumam ser normalizados para que se incremente a eficiência do treinamento da rede.

**Pesos sinápticos** são iniciados aleatoriamente com valores entre 0 e 1. Conforme ocorre o treinamento da rede, esses valores vão sendo ajustados. Note que, caso o valor do peso seja 0, ou muito próximo de 0, o valor de entrada conectado por aquela sinapse será nula ou muito pouco relevante, respectivamente. O contrário também deve ser notado, isto é, em caso de pesos iguais a 1, ou muito próximos de 1, a entrada será muito relevante.

**Combinador Linear** faz a combinação linear de todas as entradas ajustadas por seus respectivos pesos sinápticos, conforme mostra a (eq. 2.3).

$$v_n^k = \sum_{i=1}^m x_i w_{ni}^k \quad (2.3)$$

**Bias** é um valor iniciado igual a 1, ajustado no decorrer do treinamento. É somado no combinador linear, gerando o potencial de ativação  $v_n^k$  que servirá como a variável de entrada para a função de ativação. O *bias* é útil para deslocar a curva da função de ativação lateralmente, dessa forma facilitando o ajuste da curva em relação aos dados de entrada, seja se tratando de um separador ou do ajuste de uma função.

**Potencial de ativação** é a variável de entrada da função de ativação  $z_n^k$ , resultado do combinador linear.

**Função de Ativação** funcionam como limitadores ou compactadores, restringindo os valores de saída num limite, normalmente entre -1 e 1 ou entre 0 e 1. Essas funções podem ser divididas em dois tipos: linear e não-linear [35]. No caso da utilização apenas de funções de ativação lineares, a rede torna-se um combinador linear. Porém, é mais usual o uso das não-lineares, como a tangente hiperbólica (*Tanh*) (eq. 2.4), um tipo de função sigmoide (eq. 2.5).

$$z_n^k = \varphi(v_n^k) = \tanh(v_n^k) = \frac{1 - \exp(-2v_n^k)}{1 + \exp(-2v_n^k)} \quad (2.4)$$

$$z_n^k = \varphi(v_n^k) = \frac{1}{1 + \exp(-v_n^k)} \quad (2.5)$$

A função sigmoide (eq. 2.5) também é conhecida como função logística.

Com o avanço das técnicas de treinamento, surgiram novas funções de ativação e, atualmente, conforme reportado na literatura e nos manuais das bibliotecas

de aprendizado de máquina as funções mais utilizadas são ReLU e *Softmax*[36–39].

ReLU (eq. 2.6) é o acrônimo para *Rectified Linear Unit*, o formato de sua curva é exibido na Figura 2.9.

$$f(x) = x^+ = \max(0, x) \quad (2.6)$$

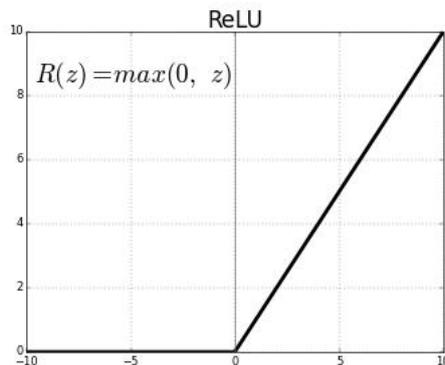


Figura 2.9: Formato da curva da ReLU. Retirado de [40].

A principal diferença entre as funções tangente hiperbólica e ReLU é o intervalo da imagem das duas. A primeira varia de -1 a 1, enquanto a ReLU vai de 0 a  $+\infty$ . Essa característica faz com que em redes profundas, em outras palavras, com muitas camadas - *Deep Learning*, gradientes negativos continuem sendo atualizados, o que torna-se muito dificultado caso fosse feito uso da *Tanh*.

A função *Softmax* (eq. 2.7), é uma generalização da função logística, sendo muito útil especialmente quando inserida na camada de saída da rede. Essa função tem sua maior vantagem em relação à sigmoide devido ao fato da última tender a saturar rapidamente os valores em seus limites, no caso 0 e 1. Já a *Softmax*, faz uma distribuição mais suave, tendo como resultado uma probabilidade de pertencimento a uma certa classe categórica.

$$\text{softmax}_i(a) = \frac{e^{a_i}}{\sum_{i=1}^K e^{a_i}} \quad (2.7)$$

Diferentes funções de ativação podem ser combinadas numa mesma RN de

forma a se obter diferentes resultados com uma mesma arquitetura.

## 2.6.2 Treinamento

Assim como um humano precisa aprender a falar, comer, andar e outras tarefas mais complexas, a rede neural também precisa de treinamento. O cérebro animal, assim como o artificial, aprende através de exemplos. Por exemplo, um bebê aprendendo o nome dos brinquedos: a mãe mostra para ele um carrinho e diz que aquilo é um carrinho. Caso ele chame aquele objeto de caneta, corrige-se a criança reafirmando que aquilo é um carrinho. Normalmente quando ela acerta, chamando o carrinho pelo nome correto, recebe um retorno positivo, um incentivo e então salva aquele padrão. A rede artificial funciona da mesma forma, pode-se mostrar a ela a imagem de um carrinho e dizemos que aquilo se chama carrinho. Ela então vai processar aquela informação até que na saída vai dar uma probabilidade daquilo ser um carrinho ou não. Como foi apresentado o objeto correto à ela, vai poder comparar o resultado e calcular o erro; caso o erro seja menor do que o limite estipulado pelo programador, o treinamento é terminado. Caso seja maior, o treinamento continua.

A cada ciclo de treinamento conta-se uma época. Isto é, a cada vez que alimenta-se a rede com os sinais de entrada, ela os processa e faz os ajustes nos pesos e *bias*, conta-se como um ciclo, ou uma época. Os tipos de treinamento podem ser divididos em: **supervisionado** e **não-supervisionado**.

**Supervisionado** é o treinamento que tem um professor. Alimenta-se a rede com os dados de entrada e com os dados de saída, ou seja, conta-se para a rede a resposta do problema. A rede processa os dados de entrada e gera um resultado, compara com a resposta correta que já lhe foi apresentada e calcula o erro daquela saída.

**Não-supervisionado** a rede aprende sozinha buscando padrões semelhantes dentro do conjunto de dados que lhe foram apresentados. Ela então tenta agrupar esses padrões de acordo com uma medida de similaridade.

Existem vários algoritmos de treinamento e, conforme já mencionado acima, foi o *backpropagation* que permitiu que redes neurais pudessem classificar classes não linearmente separáveis, reacendendo o assunto que estava estagnado e, atualmente é um dos mais pesquisados pelo mundo. O processo de treinamento é, na verdade, um processo de otimização, onde procura-se minimizar (ou otimizar) uma função objetivo  $F$  através do ajuste dos pesos sinápticos [41].

Como exemplo de funções objetivo podemos citar: erro médio quadrático, erro médio absoluto, entropia cruzada - binária, esparsa e categórica. A entropia cruzada é uma das funções que apresenta resultados melhores especialmente quando está se lidando com conjunto de dados muito grandes [38, 42]. Basicamente a função (eq. 2.8) calcula a distância  $D$  entre o resultado  $S$ , predito pelo classificador e o valor verdadeiro, ou esperado  $L$ .

$$D(S, L) = - \sum_i L_i \log(S_i) \quad (2.8)$$

Diz-se de otimização de primeira ordem o algoritmo que em seu procedimento utiliza apenas derivadas de primeira ordem. De forma análoga, um algoritmo de segunda ordem, utiliza derivadas de até segunda ordem. As derivadas em questão são as derivadas parciais da função objetivo escolhida em relação às suas variáveis. Dessa forma, é possível determinar-se o ponto mínimo daquela função.

Um fator extremamente importante a ser determinado no processo de treinamento é o passo de treinamento  $\alpha$ . Caso escolha-se um  $\alpha$  muito grande, pode-se nunca alcançar a convergência. Por outro lado, um  $\alpha$  muito pequeno pode levar muito tempo para que o treinamento convirja ou ficar preso num mínimo local. Existem métodos de  $\alpha$  variável, onde inicia-se o treinamento com um valor elevado e vai-se diminuindo com a evolução [43].

Apresenta-se aqui um algoritmo de treinamento supervisionado: o gradiente descendente ou regra delta, um processo de otimização de primeira ordem, contra o gradiente. Lembrando que o vetor gradiente aponta para o máximo da função. Por isso, um processo de otimização contra o gradiente funciona somando valores nega-

tivos na direção contrária ao vetor gradiente, de forma que se caminhe no sentido contrário, isto é, na direção do valor mínimo [32, 41, 43]. Iremos utilizar o MLP da Figura 2.10, que possui  $m$  entradas, duas camadas escondidas e uma saída, as funções de ativação dos neurônios da primeira cada escondida são do tipo  $\tanh(\cdot)$  e do neurônio da segunda camada escondida é linear. Considere também um passo de treinamento igual a  $\alpha$ .

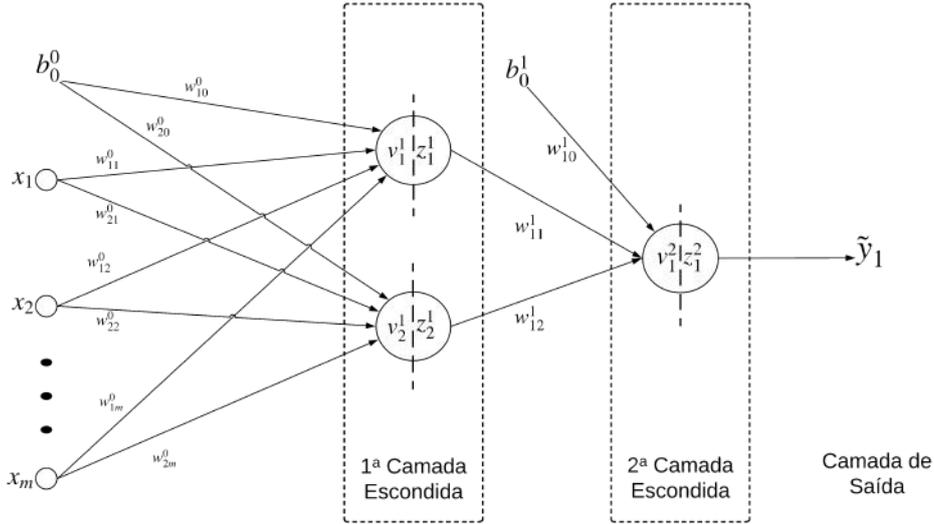


Figura 2.10: Exemplo de rede neural com  $m$  entradas, duas camadas escondidas e uma saída.

Usualmente a função escolhida como objetivo  $F$  (eq. 2.9) é o erro médio quadrático na saída da rede, onde  $m$  é o número total de pares de entrada-saída,  $\tilde{y}_k$  é a saída produzida pela rede e  $y_k$  é a saída desejada da rede (aquela introduzida pelo professor).

$$F = \frac{1}{m} \sum_{n=1}^m \varepsilon_n^2 \quad \varepsilon_n = y_n - \tilde{y}_n \quad (2.9)$$

A primeira parte do treinamento é a propagação do sinal. Começando da entrada ( $k = 0$ ) para a primeira camada escondida ( $k = 1$ ), onde o *bias*  $b_0^0$  e cada entrada  $x_n$  serão multiplicados por seus respectivos pesos  $w_{ni}^0$  e somados pelo combinador linear, resultando no potencial de ativação  $v_n^k$ , como mostra a (eq. 2.10)<sup>3</sup>. Como a rede em

<sup>3</sup>Note que  $x_0 = 1$  aparece como condição na eq. 2.10. Para simplificação do algoritmo costuma-

questão possui 2 neurônios na primeira camada escondida, teremos dois potenciais de ativação ( $v_1^1$  e  $v_2^1$ ), conseqüentemente dois resultados da função de ativação são gerados ( $z_1^1$  e  $z_2^1$ ) respectivamente (eq. 2.13).

$$v_n^k = \sum_{i=0}^m (x_i w_{ni}^{k-1}), \quad k = 1; x_0 = 1 \quad (2.10)$$

$$z_n^k = \varphi(v_n^k) \quad (2.11)$$

Continuando,  $z_1^1$  e  $z_2^1$ , assim como o *bias*  $b_0^1$  são novamente multiplicados pelos respectivos pesos das sinapses que ligam a primeira camada escondida ( $k = 1$ ) com a segunda camada escondida ( $k = 2$ ), a equação que rege essa operação é idêntica à eq. 2.10, apenas atualiza-se a variável a ser multiplicada pelo peso sináptico (2.12).

$$v_n^k = \sum_{i=0}^m (z_i w_{ni}^{k-1}), \quad k = 2; z_0 = 1 \quad (2.12)$$

Como só há um neurônio na segunda camada escondida, apenas um valor  $z_1^2$  é obtido e, este já será a saída da rede:

$$\tilde{y}_1 = z_1^2 = \varphi(v_1^2) \quad (2.13)$$

Encerra-se aqui o processo de propagação do sinal. Para completar uma época de treinamento é necessário corrigir os valores dos pesos sinápticos e dos *biases* de todas as camadas anteriores, é o que se chama de retropropagação do erro (ou *error backpropagation*).

Para começar a retropropagação, o erro é calculado de acordo com a eq. 2.14.

$$\varepsilon_1 = y_1 - \tilde{y}_1 \quad (2.14)$$

---

se representar o *bias* como se fosse uma entrada, com índice  $i = 0$ , no caso  $x_0$ .

Em seguida, são atualizados os pesos sinápticos e o *bias* da camada imediatamente anterior à saída, no caso, a segunda camada escondida. Essa atualização é feita de acordo com a eq. 2.15:

$$\Delta w_{ni}^2 = 2\alpha\varepsilon_1 z_n^1, \quad z_0 = 1 \quad (2.15)$$

Continua-se agora com a correção dos elementos da primeira camada escondida:

$$\Delta w_{ni}^1 = 2\alpha\varepsilon_1 w_{ni}^2 (1 - z_n^{1^2}) x_n, \quad x_0 = 1 \quad (2.16)$$

Finalmente uma época de treinamento está completa. Note que, devido ao fato das funções de ativação serem diferentes na primeira e na segunda camada escondida, a atualização dos elementos é diferente. Isso acontece pois no passo de *backpropagation* deve-se efetuar a derivada parcial das funções de ativação, o que resulta em modelos diferentes.

Uma evolução do algoritmo *backpropagation* é denominado Adam. É largamente utilizado ultimamente devido a uma combinação de benefícios. Trata-se de um método bastante efetivo e de rápida convergência. Tem passo de treinamento variável, combinando métodos de otimização de primeira e segunda ordem [44, 45].

**Conjuntos de treinamento** um ponto extremamente importante no treinamento de uma RN é a divisão de seus dados de entrada em três diferentes grupos: **treino**, **validação** e **teste**.

**Treino** - utilizados para o treinamento da rede até que se atinja um valor de erro menor do que o estipulado inicialmente.

**Validação** - o conjunto de validação é usado após o treinamento da rede com o conjunto de treino. São dados que ainda não foram apresentados à rede. Assim, são inseridos e processados pela RN recém-treinada e calcula-se o erro. Caso

o percentual de erro seja elevado, refaz-se o sorteio dos conjuntos e reinicia-se o treinamento.

**Teste** - esse é o último conjunto a ser apresentado à rede, após o treinamento e a validação. Novamente, são dados que não foram usados nem no treinamento nem na validação, servirão apenas para testar a eficiência da rede, sua capacidade de generalização.

Usualmente divide-se os dados na proporção de 60% para treino, 20% para validação e 20% para teste ou então em 70%, 15% e 15% respectivamente. É recomendável que os percentuais utilizados sejam semelhantes a esses para que se obtenha uma RN com boa capacidade de generalização, isto é, evite o **overfitting** [37, 38].

**Overfitting ou Overtraining** é um problema muito comum na utilização das RN e, caso não se tenha cuidado, pode passar despercebido. Voltando à comparação com o bebê, imagine agora que a criança tenha crescido e já frequenta a escola. Sua rede neural, ou seu cérebro, está sendo intensamente treinado e retreinado na aula de matemática quando lhe é ensinado a multiplicação dos números. A professora então pergunta a tabuada do número 2, ele responde com facilidade desde  $2 \times 1$  até  $2 \times 10$ . Porém, quando lhe é perguntado o resultado de  $8 \times 2$ , pensa e não sabe responder. Se os pais da criança não estão habituados às redes neurais, dirão que ela **decorou** a sequência da tabuada do 2 e quando lhe perguntam  $8 \times 2$ , ela acha que não sabe. Está aí um exemplo claro de *overfitting*.

Algumas vezes a RN chega a acertos de 99% - 100%, pode ser que o treinamento tenha sido muito bem feito e a rede é extremamente eficiente, mas é preciso prestar atenção. Assim como a criança, ela pode ter decorado aquele conjunto de dados, então quando for utilizada com dados novos, o percentual de acertos torna-se muito pequeno. Por isso é importante a separação dos dados nos conjuntos de treino, validação e teste, para se ter uma noção de como será o comportamento com dados inéditos.

Afim de se evitar o *overtraining*, pode-se tentar algumas estratégias como aumentar o conjunto de dados para treinamento<sup>4</sup>, o que geralmente é complicado ou por não se ter como obter mais dados ou ser muito dispendioso. Outra tática é modificar a arquitetura da rede, aumentando ou diminuindo o número de camadas escondidas ou o número de neurônios nas camadas, assim como as funções de ativação dos neurônios. O problema nesse caso pode ser no aumento do tempo de processamento da rede, quanto maior, mais pesado computacionalmente será o processo de treinamento da RN. Outra alternativa a ser tentada é fazer a regularização da rede, essa técnica serve para penalizar as matrizes de pesos sinápticos que sejam muito grandes, com isso, as funções de ativação tendem a ser mais lineares, deixando de detectar pontos singulares e, assim, diminuindo a tendência de *overfit*.

### 2.6.3 Extração de Características

Um ponto extremamente importante no desenvolvimento de uma rede neural, ou qualquer técnica de aprendizado de máquina é o *feature extraction*, ou extração de características. Depois da aquisição do sinal, é necessário que seja feita uma seleção de quais características são mais relevantes de forma a melhor representar aquele conjunto de dados. À determinação das características, dá-se o nome de extração de características; à seleção das mais relevantes, seleção de características [25]. Usualmente fala-se apenas de extração de características, mas o processo envolve essas duas etapas. Extraídas as características, inicia-se o processo de treinamento da RN.

## 2.7 Estado da Arte

Nesta seção será realizado um apanhado geral sobre o que foi publicado nos últimos anos sobre sistemas inteligentes integrados a ensaios não destrutivos de ultrassom.

---

<sup>4</sup>Aumentar o conjunto de dados para treinamento significa aumentar a quantidade de dados disponível para treinamento da rede, o que é muito diferente de aumentar o percentual do conjunto de treino dos dados disponíveis para treinamento.

Wang [46], faz a classificação de sinais de UT em compósitos de matriz polimérica reforçada com fibra de carbono (CFRP) utilizando máquinas de vetor de suporte (SVM). Os sinais são processados usando DWT, cujos coeficientes são utilizados para a extração dos vetores de características, que são selecionados pela técnica de análise das componentes principais (PCA), e então são classificadas.

O autor [46], cita que existem duas principais estratégias para extração de características em sistemas automatizados de classificação de sinais de UT, a primeira extrai do sinal parâmetros estatísticos como média, variância, assimetria e curtose. A outra estratégia alimenta o classificador com o sinal completo da análise de UT, e então pode fazer processamentos para extrair coeficientes da FFT ou DWT, por exemplo.

Para treinar o modelo, fez 100 ensaios de UT em CFRP com defeitos inseridos artificialmente e sem defeitos. Cada exemplo foi processado por DWT e oito características foram extraídas para cada sinal: valor médio, desvio padrão, máxima e mínima amplitude, máxima e média energia, frequência de amostras de energia mínima e ponto médio (frequência que divide todo o espectro em duas partes de mesma área). Além disso, com o sinal original, no domínio do tempo, foram extraídos valor médio, valor RMS, desvio padrão e valor absoluto (módulo). Somando todas as características extraídas obteve 12 diferentes indicadores. Devido à grande dimensionalidade do vetor de características a ser alimentado no classificador, foi utilizada a técnica de PCA para reduzir o número de componentes; o resultado obtido foi de 6 componentes principais a serem alimentadas no classificador.

Em busca do melhor classificador, usou 3 diferentes funções núcleo para o SVM, escolhendo a polinomial como melhor. Obteve taxa de acerto de 92.5% em 81.4s de treinamento. Para fins de comparação, utilizou a *toolbox* de RN do MATLAB para treinar uma rede classificadora. Usou as mesmas 6 características de entrada, função de ativação tangente hiperbólica. A rede em questão tinha apenas 1 camada escondida com 13 neurônios. A eficiência obtida foi de 86,25% com 85s de treinamento.

Kesharaju e Nagarajah [47], estudaram a extração de características para treinamento de um classificador para ser usado com UT em componentes cerâmicos. A ideia é obter-se um sistema automatizado que avise ao operador quando for encontrada uma falha. Para seleção das características a serem utilizadas, processam o sinal com DWT, fazem uso de PCA e algoritmos genéticos para escolher as entradas mais relevantes e então alimentam uma rede neural classificadora. Duas placas cerâmicas foram utilizadas com regiões com defeito e sem defeito, posição dos defeitos foi determinada por testes de raios-X. Realizou-se então a aquisição dos dados com o ensaio de UT, totalizando 132 sinais. Em seguida, aplicou-se a DWT para extração de algumas características provenientes de seus coeficientes, essas foram adicionadas à outras características obtidas no domínio temporal, totalizando 12 elementos que foram reduzidos a apenas 5 após a PCA. Concluem que a utilização de PCA e algoritmos genéticos melhoraram a performance de classificação da rede de 91% para 96%, além de reduzir os custos computacionais já que, ao invés de treinar e processar 12 entradas, a rede lidou apenas com 5 características.

Mironovova e Bíla [48], apresentam uma nova abordagem para classificação de anomalias cardíacas registradas por eletrocardiograma (ECG). Exames de ECG podem apontar diversas anomalias cardíacas, não necessariamente ligadas a doenças ou problemas de saúde. Vários aparelhos de ECG processam dados do ECG e disparam alertas para possíveis problemas, porém esses sistemas não têm capacidade de levar em conta muitas variáveis e, portanto, não são tão úteis assim e produzem resultados falsos. O objetivo do trabalho [48] é desenvolver um sistema complexo para uma classificação mais acurada, levando em conta um maior número de fatores.

O primeiro passo desse estudo foi detectar os picos R, provenientes da curva QRS, uma curva específica de exames ECG. Detectado o pico foi possível determinar diferentes fatores como, taxa de batimento cardíaco e arritmias. O sinal obtido pelo ECG é processado com uso de FFT e comparado a um valor limite que, se ultrapassado, indica um pico R e, portanto, um batimento cardíaco; registra-se então as coordenadas daquele ponto como um pico R.

Foi treinada uma rede MLP com 1 camada de entrada e 1 camada escondida, alimentada com diferentes quantidades de amostras que representavam condições normais e de anomalias. A função de erro utilizada foi a erro métrico quadrático.

No final, concluem que a rede funciona, porém quando se apresenta um novo conjunto de dados a classificação é ruim. Assim, os autores sugerem que serão feitos novos testes com outros conjuntos de dados, maiores e mais variados.

Lee [25], faz uma análise crítica de alguns métodos de extração de características de sinais de UT, principalmente FFT e DWT. O autor comenta que é esperado que dados provenientes do domínio das frequências, após o uso de FFT, sejam mais confiáveis e implicam num menor vetor de características.

O autor faz uma revisão sobre a técnica de FFT, e comenta sobre a entrada da função ser um número real e a saída ser um número complexo. Mostra que é importante utilizar os coeficientes de FFT normalizados (quando multiplica pelo conjugado dos números complexos), pois estes representam características físicas do sinal obtido. Ao mesmo tempo, relata a importância de utilizar as informações sobre fase que podem ser obtidas também pelo FFT, que geralmente são ignoradas por fazerem mais sentido apenas em sinais longos. Assim, cita um outro trabalho de sua autoria onde faz a comparação entre os dois vetores de características e conclui que obteve melhores resultados de classificação ao utilizar o vetor de dados obtido pela FFT que tem informações tanto a magnitude quanto a fase do sinal.

Lee [25] menciona de uma grande desvantagem da técnica de FFT, que é esta não guardar nenhuma informação sobre o tempo, então sugere a aplicação de outra técnica denominada *Shor-Time FT* (STFT). O STFT é uma representação tempo-frequência, divide o sinal inicial em pequenos pedaços (janelamento) e aplica a FFT nessa seção. Dessa forma, a técnica fornece informação sobre qual e quando um evento do sinal acontece. A desvantagem é a limitação da resolução que será imposta pelo tamanho da janela usada. Grande janela de tempo fornece uma boa resolução das frequências, mas pobre em tempo e vice-versa. Dessa limitação desenvolve-se a transformada de Wavelet (cujas algoritmo mais comum é a DWT).

O autor cita ainda outros trabalhos que comparam o uso de FFT e DWT para extração de características de sinais e posterior classificação. Todas as comparações mostram que o uso de DWT foi mais eficiente, porém faz a ressalva que nenhuma das comparações fez uso da parte da FFT que leva em conta a fase dos sinais.

Simas Filho *et al.* [49], estudam quais as características resultam numa melhor rede neural classificadora para um sistema automatizado de apoio à tomada de decisão em ensaios de UT de juntas soldadas. Utiliza FFT, DCT e DWT para processamento dos sinais e também o uso de PCA.

Para os ensaios utiliza uma chapa de aço carbono 1020 soldada, no cordão de solda foram inseridos alguns defeitos, localizados por inspeção de raios-X. Faz-se então o ensaio de UT em imersão, varrendo duas linhas paralelas ao cordão de solda, com diferentes afastamentos. São obtidos 100 sinais para cada classe de interesse (3 com defeito e 1 sem defeitos).

Os dados são alimentados num MLP de três camadas (uma camada escondida), porém o número de neurônios variava a cada entrada de dados buscando a melhor classificação, que ocorreu ao utilizar 12 neurônios. A função de ativação utilizada foi tanh e treinadas com o algoritmo de Levenberg-Marquadt.

Os resultados obtidos mostraram que sem o uso de PCA, a melhor classificação resultou numa taxa de acerto de 94,8% utilizando apenas a FFT, seguido de 93,5% ao usar DCT. Com o uso de PCA, o que resultou numa diminuição de 80% do conjunto de dados, o melhor resultado continuou sendo o que utilizou apenas FFT, 97,5%, seguido do DCT com 96,2% e DWT com 89,93%. Concluiu que a os testes com DWT não foram afetados pelo uso de PCA, ao mesmo tempo, afirmou obter eficiência de 89,5% quando usou apenas 25% dos dados disponíveis.

Cruz *et al.* [50], revisitam o que foi publicado em [49], utilizando o mesmo material e com o mesmo objetivo. As diferenças são: a utilização de outra técnica estatística para seleção de características, no caso a Wilcoxon-Mann-Whitney (WMW), a alimentação da RN com os sinais do *A-Scan* sem processamento e uma nova metodologia para treinamento da RN.

O conjunto de dados é o mesmo, isto é, sinais de UT obtidos na varredura de um cordão de solda de um aço carbono 1020, com defeitos artificiais inseridos. No treinamento da RN, duas metodologias são implementadas: a primeira, semelhante ao primeiro artigo, faz a classificação em 4 diferentes classes (3 com defeito e 1 sem defeito). A segunda metodologia propõe uma classificação em duas etapas: na etapa 1 será feita a separação em dois grupos – “com defeito” e “sem defeito”. Em seguida, na etapa 2, o grupo dos “com defeito” será então reclassificado, resultando nos 3 diferentes tipos de defeito inseridos no cordão de solda.

Como resultado, a metodologia 1 apresentou melhores resultados, sendo o melhor a combinação da FFT + PCA com eficiência de 97,5% usando apenas 20 características. O pior foi o resultado obtido com o treinamento apenas do *A-Scan*, com 2500 características e eficiência de 53,1%. O melhor resultado da metodologia 2 foi bastante semelhante à de 1, mas também contou com o uso da WMW, que teve resultados muito próximos aos obtidos com uso de PCA.

A seguir a Tabela 2.2 apresenta um resumo dos artigos descritos acima e de outros que não foram aqui apresentados de forma extensiva por sua semelhança em relação aos demais.

Tabela 2.2: Resumo de artigos sobre UT e RNs.

Autor	Material	Técnicas	N.º Ca-ract.	Efic.	Observações
Wang [46]	CFRP	DWT e PCA	6	92,5%	Usa SVM, mas compara o resultado com uma RN.
Kesharaju e Nagarajah [47]	Cerâm.	DWT e PCA	5	96%	-
Mironovova e Bíla [48]	-	FFT	-	-	Classifica sinais de ECG.
Lee [25]	-	FFT, STFT e DWT	-	-	Faz uma análise crítica das técnicas de extração de características.
Simas Filho <i>et al.</i> [49]	Solda em aço	FFT, DCT, DWT e PCA	-	97,5%	Melhor resultado da combinação FFT + PCA.
Cruz <i>et al.</i> [50]	Solda em aço	FFT, DCT, DWT, PCA e WMW	20	97,5%	Melhor resultado foi combinação FFT + PCA.
Barry <i>et al.</i> [51]	CFRP	-	28	98%	Usa como características vários dados estatísticos retirados do sinal e consegue uma RN com excelente classificação e bastante generalizada.
Sambath <i>et al.</i> [52]	Aço inox	DWT	8	94%	Das 12 características obtidas com DWT, faz a seleção das 8 mais relevantes gerando diversas RNs.

# Capítulo 3

## Materiais e Métodos

### 3.1 Metodologia

Este estudo iniciou-se com a produção dos corpos de prova de PVDF, então submetidos a ensaios de UT, ensaios de tração e novamente inspeção por UT, para que se obtivesse sinais com e sem a presença de *whitening*. Em seguida esses dados foram tratados e alimentados para o treinamento de uma rede neural artificial classificadora do tipo perceptron multi-camadas (MLP). O esquema é mostrado no fluxograma da Figura 3.1 a seguir.

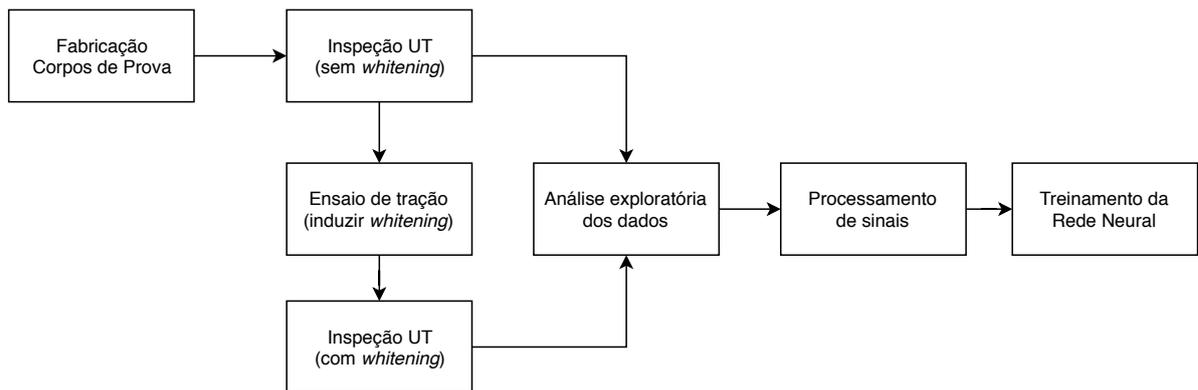
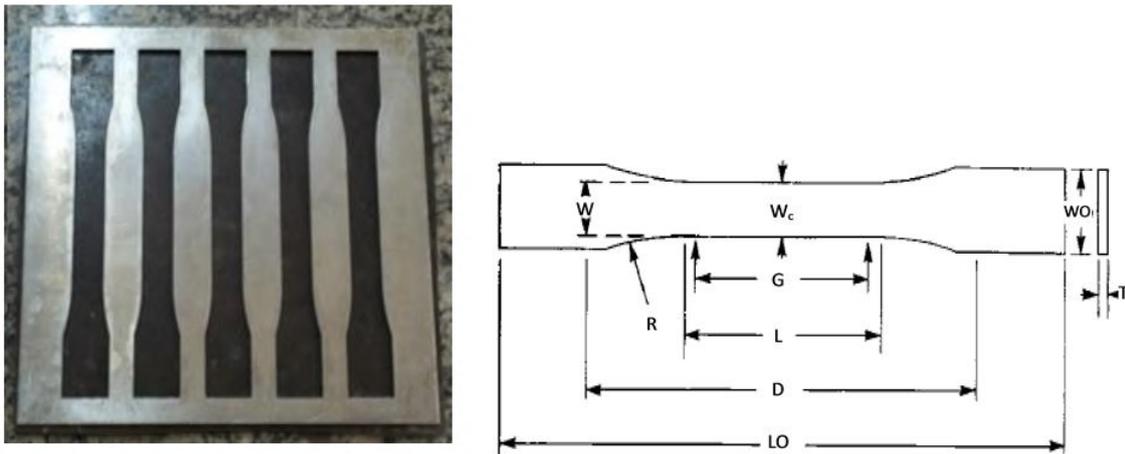


Figura 3.1: Fluxograma das etapas do projeto.

## 3.2 Fabricação dos Corpos de Prova - PVDF

Os corpos de prova foram produzidos a partir do material e auxílio fornecidos pelo Laboratório de Processamento e Caracterização de Materiais (LPCM) da COPPE/UFRJ, a partir de *pellets* do copolímero de PVDF processados pela técnica de moldagem por compressão em uma prensa hidráulica MARCONI MA 098/A do Laboratório de Análise Química e Processamento Cerâmico da COPPE/UFRJ.



(a) Molde em aço inox. Retirado de [3]. (b) Dimensões dos corpos de prova. Adaptado de [5].

Figura 3.2: Molde e dimensões dos corpos de prova, segundo ASTM-D638 [11]. Dimensões:  $w_c = 13\text{mm}$ ,  $w_0 = 19\text{mm}$ ,  $T = 3,3\text{mm}$ ,  $L_0 = 165\text{mm}$ ,  $D = 115\text{mm}$  e  $R = 38,4^\circ$ .

Na primeira etapa, o molde dos corpos de prova (Figura 3.2), que obedecem à norma ASTM D638 [11], são preenchidos com os *pellets* de PVDF e levados por 15 minutos à estufa Marconi MA030 com vácuo de 200mmHg e aquecida a 150°C. Em seguida, o molde preenchido é colocado na prensa hidráulica aquecida a 220°C onde é submetido a uma sequência de carregamentos e degasagens, iniciando em 1 tonelada até alcançar 6 toneladas, conforme ilustrado no fluxograma da Figura 3.3. A degasagem serve para liberar gases e umidade que possam estar aprisionados no molde, dessa forma evitando a formação de vazios ou defeitos durante o processo de fabricação.

Após atingir a carga de 6 toneladas e realizada a última degasagem, espera-se 15 minutos e, em seguida, retira-se o molde para resfriar em temperatura ambiente. Os corpos de prova são desmoldados e separados, sendo necessário lixar suas laterais

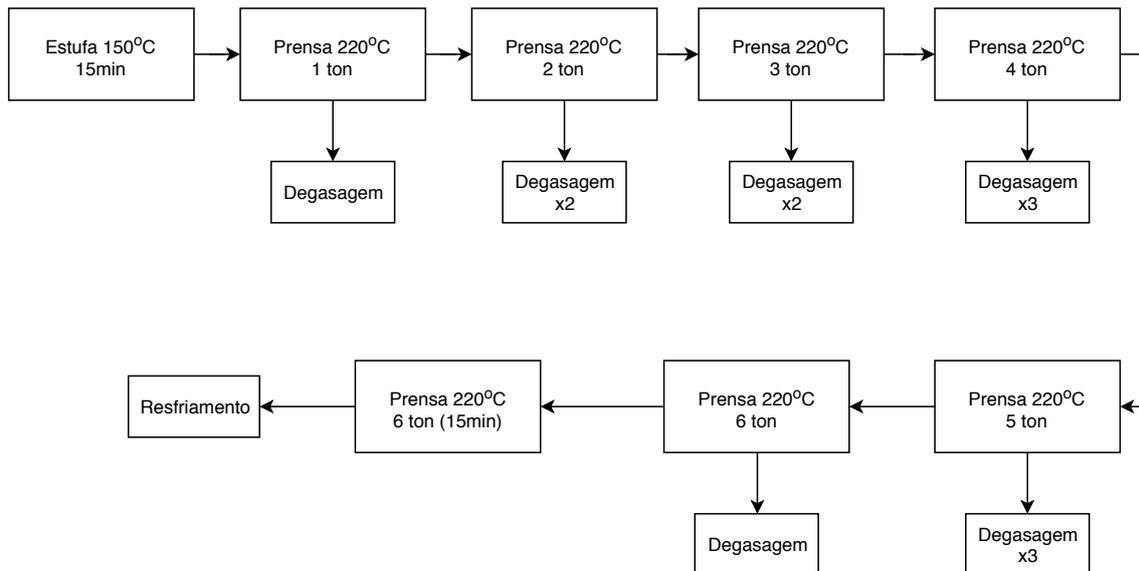


Figura 3.3: Fluxograma das etapas de degasagem e carregamento.

com uma lixa de granulometria 220 afim de retirar pequenas rebarbas, finalizando com uma lixa de granulometria 600. Essas rebarbas apesar de serem um fino filme do polímero, no momento do ensaio de tração podem funcionar como concentradores de tensão e induzir naquele ponto um defeito.

Foram produzidos 15 corpos de prova por moldagem por compressão. Devido ao tamanho do molde, que tem espaço para 5 peças, foram necessárias 3 bateladas então nomeadas A, B e C.

A primeira batelada, denominada A, foi a primeira a ser produzida. As outras duas bateladas, B e C, foram produzidas na sequência. As amostras do grupo B e C apresentaram coloração amarelada, mais pronunciada no grupo C, vide o exemplo na Figura 3.4. O amarelamento presente nas amostras é um sinal de envelhecimento.

de acordo com cada uma das 3 bateladas produzidas.

### 3.3 Ensaio de Tração

Para induzir o efeito *whitening* nas amostras, os corpos de prova foram submetidos a um ensaio de tração realizado no Laboratório de Propriedades Mecânicas (PROP-MEC) da COPPE/UFRJ. O equipamento utilizado foi uma Instron 9632, com velocidade transversal de 50mm/min. Inicialmente utilizou-se um extensômetro para

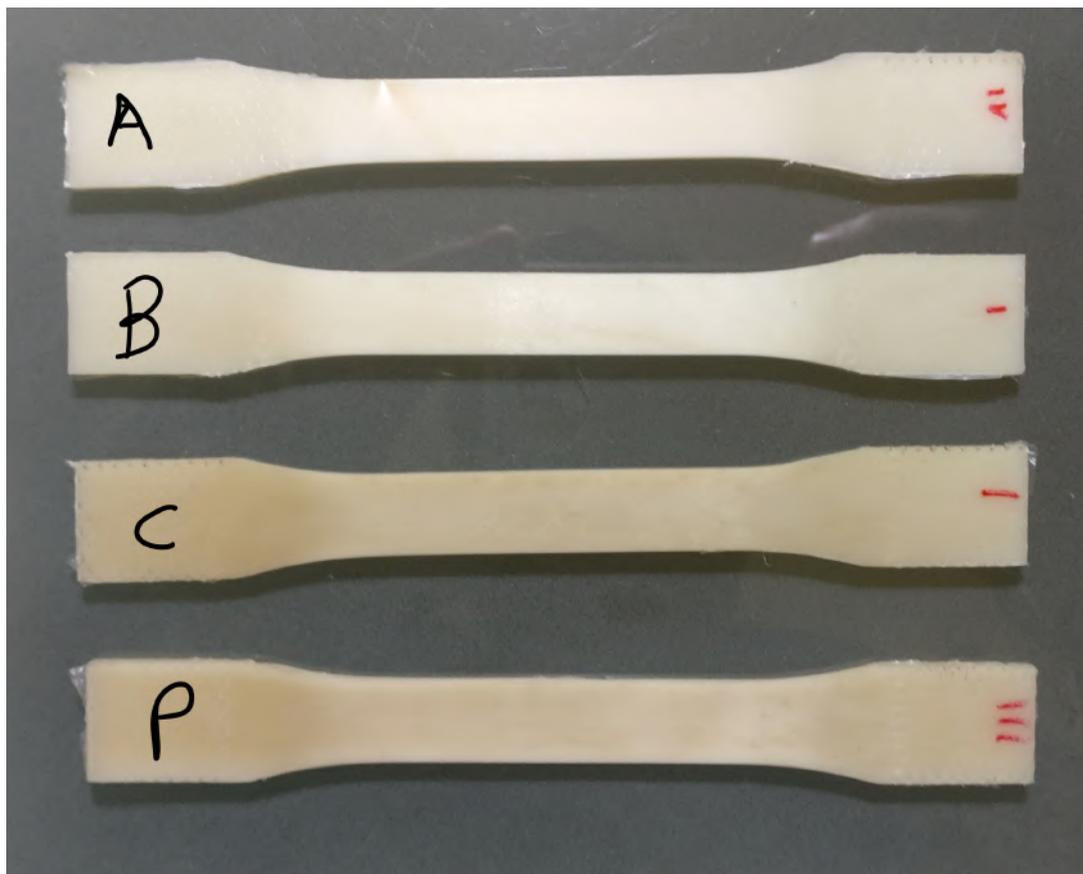


Figura 3.4: Corpos de prova de PVDF.

que fosse medida a deformação de forma mais acurada. Porém, o extensômetro disponível não tinha as dimensões adequadas para o comprimento útil do corpo de prova utilizado, o que terminou por romper uma das amostras. Em seguida, optou-se por medir o deslocamento pelo próprio sensor que controla a máquina, no caso, o deslocamento do travessão.

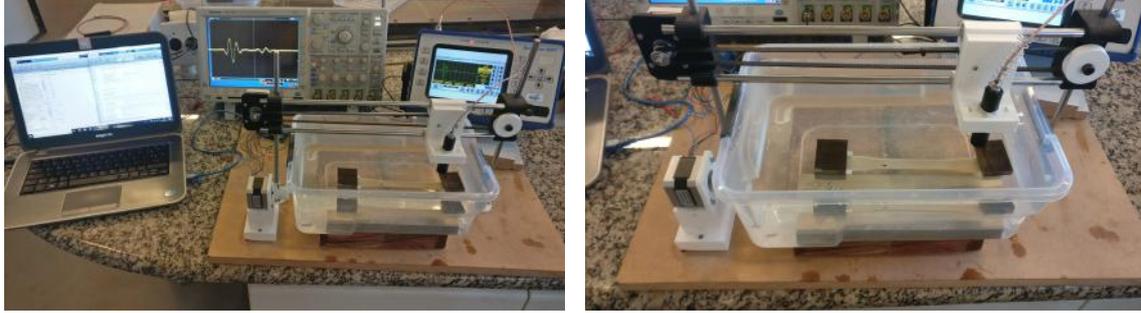
As amostras foram subdividas em 3 grupos, submetidos à deformação de 5%, 7,5% e 10%, conforme apresentado na Tabela 3.1.

Tabela 3.1: Relação dos corpos de prova.

CP	Def. (%)	
A 01	10	
A 02	10	
A 03	7,5	
A 04	7,5	
B 01	10	
B 02	10	
B 03	10	
C 01	5	
C 02	5	
C 03	5	
C 04	5	
C 05	5	Observação:
P 01	5	Polida
P 02	5	Polida e Fraturada
P 03	5	Polida

### 3.4 Ensaios de UT

Os ensaios de UT foram realizados no Laboratório de Ensaios Não Destrutivos e Soldagem (LNDC) da COPPE/UFRJ, fazendo uso de um sistema automatizado de inspeção desenvolvido no próprio laboratório, apresentado na Figura 3.5.



(a) Sistema completo

(b) Detalhe

Figura 3.5: Sistema automatizado para ensaio UT em imersão.

O aparato é composto por dois motores de passo com 1W de potência cada, o controle é feito por uma plataforma Arduino UNO. Os sinais de UT foram captados em imersão por um transdutor IMASONIC 1779 de 15mm diâmetro, não focalizado, com frequência nominal de 1,6MHz e largura de banda de 100%. Para a geração dos pulsos ultrassônicos foi utilizado um SONOTRON NDT ISONIC 2005, conectado a um osciloscópio digital Tektronix MSO 4043 com taxa de amostragem de 1,25 giga amostras por segundo e frequência de 200MHz. Este conecta-se a um computador que rodava um *script* para sincronizar o sinal obtido com a posição do transdutor no corpo de prova, armazená-lo e controlar o Arduino.

A altura da coluna d'água foi calculada em 70mm, de acordo com a (eq. 3.1)[53].

$$CD = \frac{V_{agua}}{V_{material}} \times e + 0,006m \quad (3.1)$$

Onde  $CD$  é o tamanho da coluna d'água em metros,  $V_{agua}$  e  $V_{material}$  são a velocidade do som em (m/s) na água e no material respectivamente, e  $e$  a espessura do corpo de prova em metros.

O sinal adquirido era composto por 10.000 pontos. Foram inspecionados 15 corpos de prova sem *whitening* e 14 com a presença do efeito (uma foi rompida no ensaio de tração devido a um mal-funcionamento do equipamento). Cada amostra foi avaliada em 3 varreduras consecutivas (3 leituras para cada ponto), nos 16 diferentes pontos, distantes 3mm entre si, exibidos na Figura 3.6. Dessa forma, obteve-se 1408 sinais *A-Scan* para serem tratados e alimentados na rede neural.

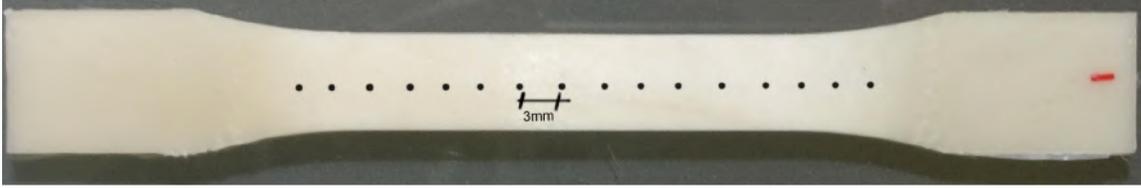


Figura 3.6: Pontos de inspeção do UT no corpo de prova.

Para saber a distância mínima entre um ponto e outro, utilizou-se a seguinte equação (eq. 3.2)[54].

$$\sin \gamma = K1 \times \frac{V}{D_{ef} \times f} \quad (3.2)$$

Onde  $\gamma$  é o ângulo entre o eixo principal de propagação do feixe sônico e sua lateral,  $K1$  é um parâmetro adimensional que varia de acordo com o ganho do sinal,  $V$  a velocidade do som no material (m/s),  $D_{ef}$  o diâmetro efetivo do transdutor e  $f$  a frequência (Hz). Feito o cálculo, concluiu-se que os pontos deveriam estar 0,1mm distantes um do outro para que se pudessem ser diferenciados. Como adotou-se uma distância de 3mm, sabe-se com certeza que cada medida é única.

### 3.5 Processamento de Sinais

Como a rede neural seria treinada utilizando o ambiente de desenvolvimento do *Python*, este também foi empregado para o tratamento de sinais beneficiando-se das bibliotecas *numpy* e *scipy*.

Preliminarmente foi feita a plotagem e análise dos dados, procurando a presença de ruídos. Em seguida, aplicou-se dois métodos para filtrar os sinais, primeiro

fazendo a média aritmética dos dados, seguido do filtro de Savitzky-Golay.

Como é esperado que o efeito *whitening* ocorra devido ao aparecimento de microcavidades, decidiu-se analisar o trecho do sinal de UT correspondente à interação entre a onda do UT e o material que corresponde à região de retroespalhamento, esta fica localizada entre os dois ecos consecutivos do sinal *A-scan* (Figura 3.7). A partir deste momento, trabalhou-se apenas com essa fração do espectro.

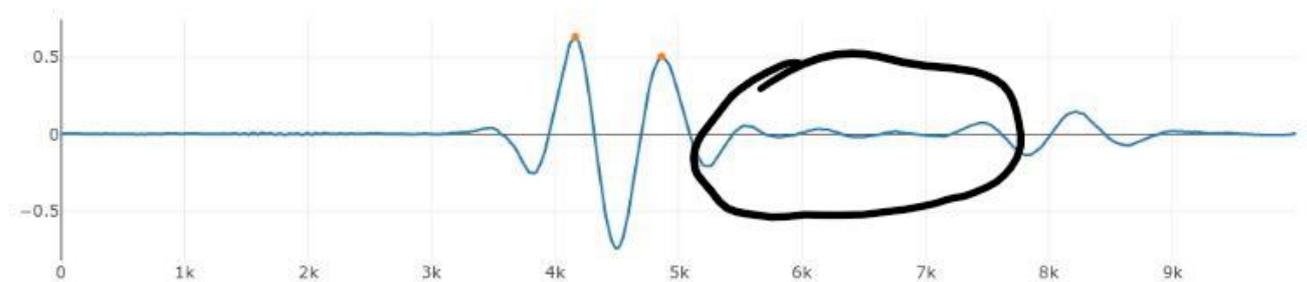


Figura 3.7: Sinal retroespalhado.

Conforme pôde ser visto na Tabela 2.2 da seção 2.7 (pág. 32), não há consenso sobre a melhor estratégia ou técnica para extração das características para treinamento da rede. Alguns autores obtiveram melhores resultados usando DWT, outros com a combinação de técnicas como FFT + PCA. Indicando que talvez essas análises sejam empíricas.

Por ser o método mais observado na pesquisa bibliográfica, utilizou-se inicialmente a técnica da transformada discreta de Fourier, no caso, seu algoritmo mais eficaz, o FFT. A segunda transformada aplicada foi a discreta dos cossenos, DCT.

É importante frisar que essa etapa de tratamento de sinais não ocorreu da forma linear descrita aqui, a cada nova filtragem ou transformada aplicada, treinava-se a rede neural para avaliar a eficácia do método utilizado.

## 3.6 Desenvolvimento da RNA

A rede foi desenvolvida utilizando a linguagem *Python* 3.6 utilizando a biblioteca de aprendizado de máquina *TensorFlow* 1.9.0, implementado pela interface *Keras* 2.2.2.

A escolha deste ambiente deve-se primeiramente à sua capacidade e disponibilidade, devido às bibliotecas, no desenvolvimento de redes neurais artificiais, permitindo o total dimensionamento e configuração das redes; à sua natureza de código aberto, ou seja, de uso livre e totalmente gratuito; e por ser praticamente o ferramental padrão na indústria de inteligência artificial.

O tipo de rede neural (RN) treinada é um perceptron multi-camada (MLP). A topologia e os parâmetros da rede como funções de ativação, algoritmo de treinamento e avaliação foram determinados no decorrer do estudo, devido ao fato de serem ajustados conforme os resultados obtidos. Trata-se de um processo empírico [34].

Treinava-se uma nova RN a cada nova etapa concluída no processamento de sinais, como mencionado, trata-se de um desenvolvimento paralelo. A cada novo resultado obtido na etapa anterior, verificava-se sua influência numa nova rede, sendo um processo cíclico.

# Capítulo 4

## Resultados e Discussão

Neste capítulo são apresentados os resultados obtidos neste trabalho. Da seção 4.2.6 até 4.2.9, com o objetivo de simplificar a compreensão, os dados obtidos com a classificação pelas redes neurais estão divididos por tipo de processamento de sinais realizado. Na seção 4.3, são realizados alguns estudos de casos com os resultados obtidos no trabalho.

### 4.1 Ensaios de UT e Tração

O conjunto de amostras disponíveis A, B e C foi inspecionado por UT em 16 pontos distintos ao longo do corpo para aquisição dos sinais sem a presença de *whitening*, submetidos a ensaio de tração e novamente inspecionados por UT - nesse momento para registrar os sinais com a presença do efeito *whitening*.

Os ensaios de UT foram realizados com o sistema automatizado desenvolvido no LNDC. Na primeira inspeção, com o conjunto de corpos de prova A, B e C, obteve-se um total de 240 sinais *A-scan*.

Na tentativa de estudar a porosidade proveniente do processo de moldagem do material, separou-se 3 corpos de prova para a inspeção por microscopia eletrônica de varredura (MEV). O MEV devia ser executado em modo de baixo vácuo para que não fosse necessário fazer recobrimento na peça e, portanto, danificar a amostra. Mesmo sem recobrimento foi necessário realizar um polimento na superfície da peça.

Para isso, designou-se uma amostra do grupo A e duas do grupo B, nomeados a partir de então grupo P. Para manter a uniformidade da nomenclatura, deste ponto em diante os grupos A e B passam a ter 4 e 3 corpos de prova respectivamente, mesmo quando houver referência ao primeiro ensaio de UT (quando não havia a divisão P).

Note que o resultado do MEV e mesmo sua descrição no capítulo 3 - Materiais e Métodos, não são reportados por se tratar apenas de um teste rápido que não teve resultados, isto é, não foi possível a observar estruturas relevantes da peça, mesmo com a técnica de baixo vácuo. O único motivo de mencionar-se aqui sua ocorrência é devido ao polimento das amostras, conforme será esclarecido a seguir.

Mesmo que o polimento provoque um leve dano à superfície do material, suspeitou-se que esse pequeno dano pudesse implicar numa má leitura do sinal de UT na segunda inspeção. Portanto as amostras do grupo P foram as primeiras a serem utilizadas no ensaio de tração, como precaução caso ocorresse algum problema nos testes iniciais, que são sempre mais críticos.

No decorrer do ensaio de tração, devido à má adequação do extensômetro utilizado, uma das amostras foi rompida. Consequentemente, o conjunto dos corpos de prova para ensaio de UT com a presença de *whitening* foi reduzida para 14. Foram realizadas deformações de 5%, 7,5% e 10%, conforme mostrado na Tabela 4.1 e pode-se observar na curva do ensaio na Figura 4.1.



Figura 4.1: Curva de Tensão x Deformação.

Tabela 4.1: Relação dos corpos de prova.

CP	Def. (%)	
A 01	10	
A 02	10	
A 03	7,5	
A 04	7,5	
B 01	10	
B 02	10	
B 03	10	
C 01	5	
C 02	5	
C 03	5	
C 04	5	
C 05	5	Observação:
P 01	5	Polida
P 02	x	Polida e Fraturada
P 03	5	Polida

Deformadas as amostras fez-se a segunda inspeção de UT para a caracterização dos corpos de prova com a presença do efeito *whitening*. Novamente utilizou-se o sistema automatizado de inspeção, capturando os sinais em 16 pontos distintos, distantes 3 mm entre si, em 3 varreduras. Conforme pressuposto, o polimento teve grande influência no ensaio de UT, tornando as amostras do grupo P inviáveis para inspeção. Isso acontece pois, devido às ranhuras criadas na superfície do material, as ondas ultrassônicas tendem a sofrer desvios para diferentes direções.

Calculou-se a velocidade do som no PVDF antes e depois do ensaio de tração, de forma a confirmar que o efeito *whitening* havia sido induzido no material. Conforme reportado em [3], era esperado que a velocidade diminui-se com o aparecimento do fenômeno, e isso foi confirmado, conforme pode ser observado na Tabela 4.2.

Tabela 4.2: Velocidade do som calculada.

CP	Velocidade (m/s)	Variação (%)
A2	2250,92	-2,52%
A2 - 10%	2194,24	
C2	2305,76	-3,27%
C2 - 5%	2230,30	

## 4.2 Redes Neurais

### 4.2.1 Análise Exploratória dos Dados

Para realizar o treinamento da rede neural, o primeiro passo é analisar o conjunto de dados. Nesse sentido, dividiu-se os sinais *A-scan* dos resultados de UT de duas formas: sem *whitening* e com *whitening*. O conjunto de dados relativos aos corpos de prova não deformados e deformados serão doravante denominados original (ou originais) e *whitening*, respectivamente. No caso da *whitening* convencionou-se adicionar o percentual de deformação ao nome do corpo de prova. Todos os dados podem ser vistos separados na Tabela 4.3.

Tabela 4.3: Divisão do conjunto de dados.

Original	<i>whitening</i>
P1	X
P2	X
P3	X
A1	A1 - 10%
A2	A2 - 10%
A3	A3 - 7,5%
A4	A4 - 7,5%
B1	B1 - 10%
B2	B2 - 10%
B3	B3 - 10%
C1	C1 - 5%
C2	C2 - 5%
C3	C3 - 5%
C4	C4 - 5%
C5	C5 - 5%

A primeira classe, original, tinha um total de 720 dados enquanto a segunda, *whitening* possuía 528. Como esse é o resultado das 3 varreduras, onde a aquisição é feita no mesmo ponto, o conjunto de dados distintos possui, de fato, 240 sinais originais e 176 *whitening*. O menor número do segundo conjunto são devido aos corpos de prova do grupo P *whitening*, que convencionou-se não utilizar, conforme já explanado.

O processo de revisão dos conjuntos é iniciado pela análise exploratória dos dados. Traçando-se as curvas *A-scan* de cada conjunto de forma superposta sendo possível rapidamente ter uma boa noção da uniformidade dos sinais. Destarte, observou-se que os sinais referentes à primeira análise de UT, isto é, da classe original, poderiam ser utilizados.

Da mesma forma, revisando os sinais da classe *whitening* descartou-se:

- Grupo A
  - todo corpo de prova A1 - 10%
  - primeiro ponto de A3 - 7,5%
- Grupo B
  - todos os dados - 10%
- Grupo C
  - primeiro e segundo pontos de C1 - 5%

Em A3-7,5% e C1-5%, ocorreu provavelmente o mau posicionamento do corpo de prova na hora da inspeção de UT. Dessa forma, fazendo com que esses primeiros pontos não pertencessem à área útil, mas aos extremos do corpo que fica bastante danificado por ser a região de contato com a garra do ensaio de tração. Em A1-10% e em todo grupo B notou-se que os formatos da curva que não apresentavam qualquer semelhança quando comparados com os dos outros corpos de prova (vide Figura 4.2). É esperado que haja diferença nas curvas de duas peças distintas, porém, essa alteração deve ocorrer na amplitude do sinal ou no deslocamento e mudança na posição dos picos.

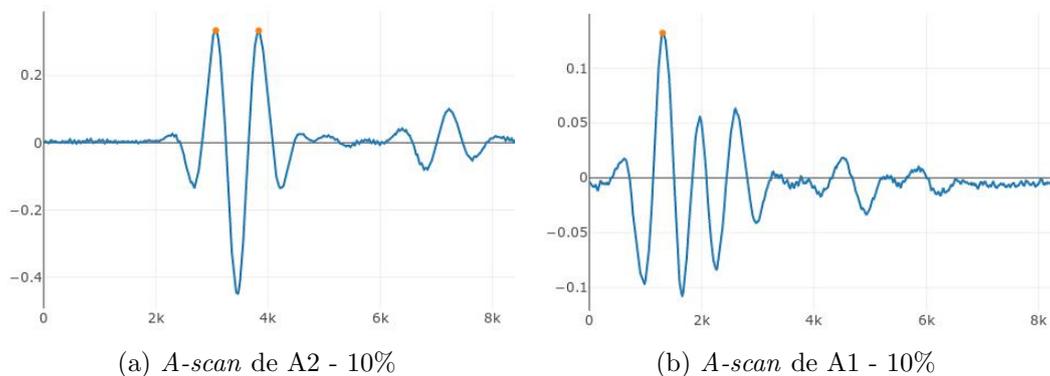


Figura 4.2: Comparação entre um *A-scan* (a) íntegro e um (b) deteriorado.

O descarte dos corpos de prova acima mencionados acaba de alguma forma prejudicando o treinamento da rede neural e, dessa forma, sua robustez. Isto é, no

caso onde foram eliminadas muitas amostras com 7,5% de deformação, tende-se a ter uma rede menos especializada na detecção desse percentual de deformação. Porém, como o objetivo principal deste trabalho é conseguir fazer a separação entre a ocorrência ou não do efeito *whitening*, a redução desses dados não teve impacto negativo significativo no treinamento da rede.

### 4.2.2 Divisão das Classes para Treinamento

Com o conjunto de dados analisado e limpo, pode-se então fazer a divisão das classes que irão ser utilizadas para o treinamento da rede. Estabeleceu-se dois tipos de divisões, uma com 2 classes e outro com 4 classes distintas. A primeira classifica os dados em original ou *whitening*, enquanto a segunda é mais específica, não só separando amostras em relação à presença ou não do efeito, mas, quando do último caso, divide os dados quanto à um percentual de deformação sendo 10%, 7,5% e 5%. Tanto a divisão e número de dados disponível para treinamento em cada classe podem ser observados na Tabela 4.4 a seguir.

Tabela 4.4: Divisão das classes para treinamento e quantidade de dados disponíveis.

	2 Classes		4 Classes			
	original	<i>whitening</i>	original	<i>whitening</i>		
				10%	7,5%	5%
Qtd. Dados Disp.	239	125	239	16	31	78
Qtd. Total Dados	364		364			

### 4.2.3 Processamento de Sinais

A análise exploratória dos dados tem como objetivo eliminar dados que não correspondam ao problema estudado. Isso é diferente da etapa a seguir, do processamento de sinais, onde o objetivo é eliminar ruídos e manipular os conjuntos de dados disponíveis de forma que seja possível a extração de características que sejam relevantes no trabalho.

Inicialmente foi feita a plotagem dos sinais *A-scan* para verificar a presença de ruídos e a necessidade da aplicação de filtros. De forma a garantir que os sinais captados no ensaio de UT sejam válidos, foram realizadas 3 varreduras. Em seguida, calcula-se a média aritmética para cada ponto obtido em cada varredura. Além disso, esse recurso funciona como um filtro de ruído, um exemplo é mostrado na Figura 4.3. Note que há diminuição no ruído, porém de forma suave.

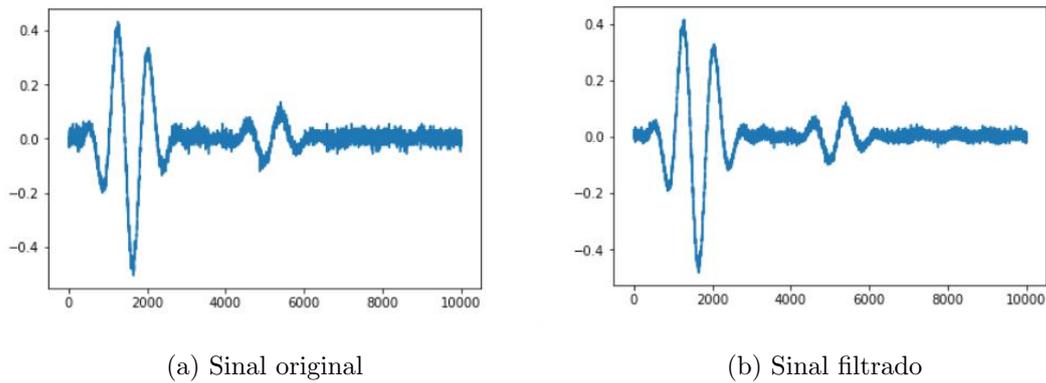
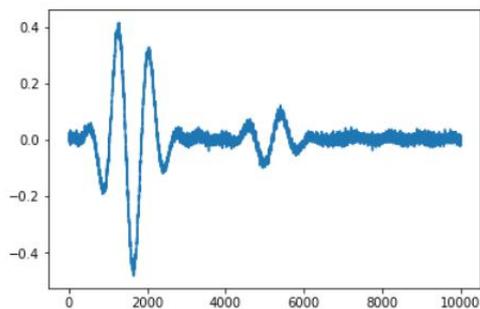


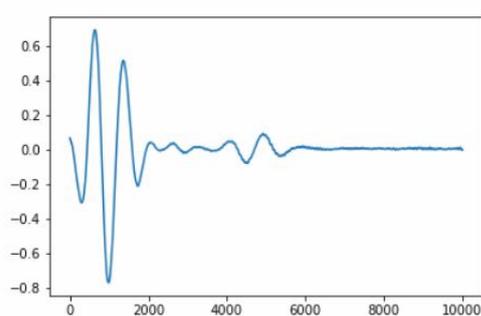
Figura 4.3: Comparação de sinais filtrados.

Para fazer o corte na porção retroespalhada do sinal, na técnica utilizada, era necessário encontrar os ecos de superfície e de fundo para então fazer a extração da parte de interesse. Entretanto, embora já aplicada a filtragem de inicial, pela média aritmética, ainda havia uma quantidade relevante de ruídos. O fato tornou-se evidente quando da utilização de algoritmos para encontrar os picos do sinal, que não alcançava performance satisfatória. Portanto, foi necessária a aplicação de outro filtro de ruídos.

O filtro escolhido foi o de Savitzky-Golay, que apresentou excelente resultado conforme pode ser observado na Figura 4.4.



(a) Sinal filtrado por média aritmética



(b) Sinal filtrado por Savitzky-Golay

Figura 4.4: Comparação de sinais filtrados.

Com o sinal filtrado foi possível encontrar os picos e fazer o corte, ou janelamento dos sinais, extraíndo a região de interesse a seção retroespalhada.

A partir dessa etapa, com os sinais limpos pela análise exploratória de dados, filtrados e particionados no processamento de sinais, dá-se seguimento para o treinamento das redes neurais.

#### 4.2.4 Topologia das Redes

De modo geral, utilizou-se a mesma topologia de rede para os 3 casos deste trabalho: *A-scan*, FFT e DCT. Essa decisão inicialmente visava manter um sistema consistente de teste e ao se variar as topologias durante os treinamentos notou-se que não havia diferenças relevantes.

Portanto, o número de camadas, funções de ativação, algoritmo de treinamento, partição de classes de teste e treino, método de acurácia são os mesmos para todas as redes. A variação ocorreu no número de neurônios das duas primeiras camadas, o que era imperativo devido à diferença no tamanho do vetores de entrada de cada tipo de alimentação. O detalhamento dessas informações é dado nas Tabelas 4.5 e 4.6.

Tabela 4.5: Propriedades das Redes Neurais.

Propriedades		
Nº camadas	3, sendo 1 escondida	
Otimizador	Adam	
Taxa aprendizagem	0,001	
Função erro	Entropia cruzada categórica	
Métrica	Acuracidade	
Função de ativação	1ª/2ª Camadas	3ª Camada
	ReLU	<i>Softmax</i>
Partição dados	Treino	Teste
	60%	40%

Tabela 4.6: Número de neurônios nas redes.

		<i>A-scan</i>	FFT	DCT
Camada	1ª	2590	10	25
	2ª	50	10	10
	3ª	2 Classes	4 Classes	
2		4		

A topologia das redes é mostrada na Figura 4.5 a seguir:

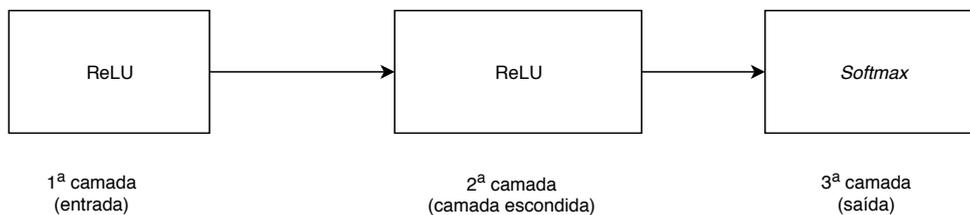


Figura 4.5: Topologia das Redes.

## 4.2.5 Resultados das Classificações

A partir desta seção, são apresentados os resultados das classificações. De forma a manter-se a coesão na referência à qualidade dos resultados, estabeleceu-se a escala

exposta na Tabela 4.7.

Tabela 4.7: Escala da qualidade de classificação.

Faixa de Acerto (%)	Escala
95,0% - 100%	Excelente
90,0% - 94,9%	Muito Bom
85,0% - 89,9%	Bom
80,0% - 84,9%	Regular
75,0% - 79,9%	Ruim
<75,0%	Muito Ruim

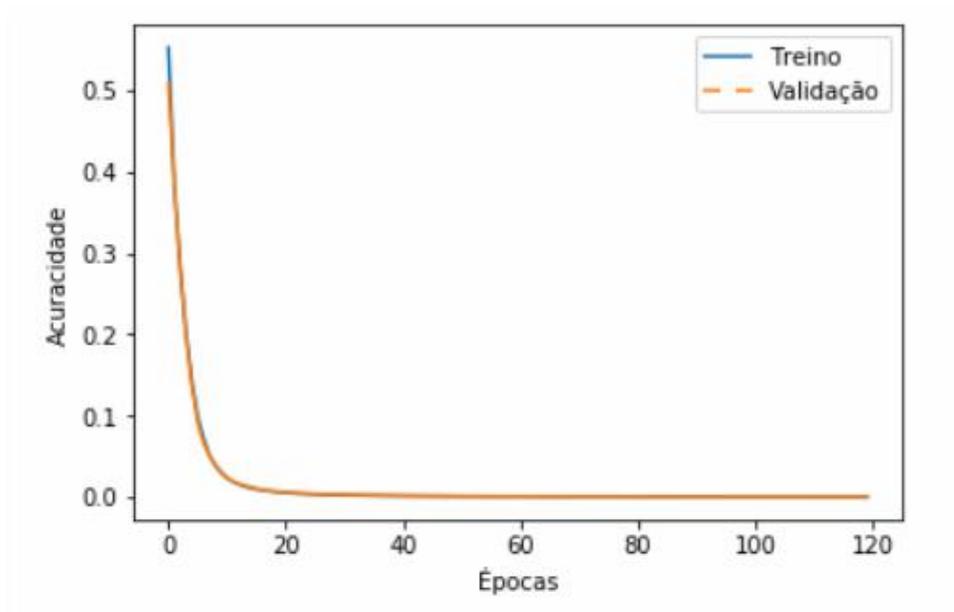
#### 4.2.6 Classificação com *A-Scan*

Embora note-se na pesquisa bibliográfica para este trabalho (seção 2.7), que os autores que utilizaram o *A-scan* sem processamento obtiveram resultados ruins, decidiu-se por testar inicialmente essa condição. Porém, como apresentado, apenas foi utilizada a porção referente à região retroespalhada do sinal. Além disso, na bibliografia não é claro se os sinais usados para alimentar os modelos haviam sido filtrados ou não.

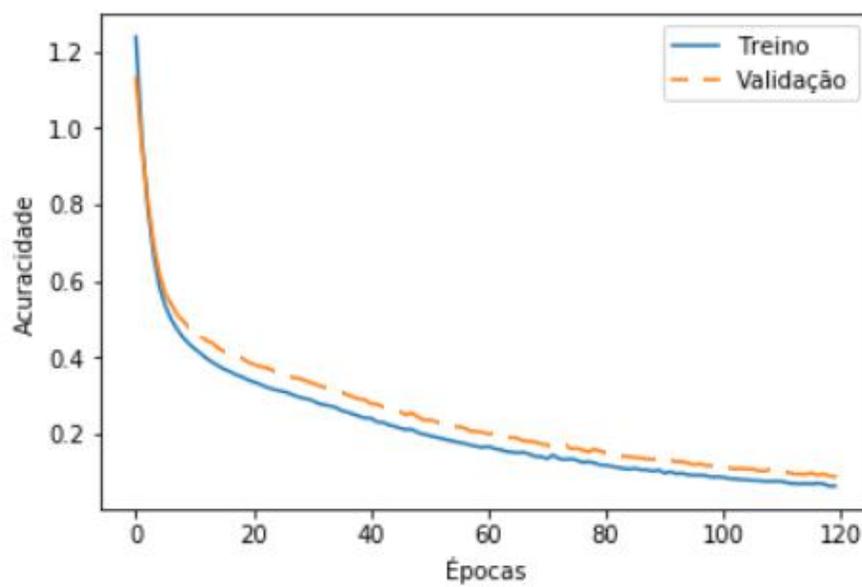
**Estatísticas do Treinamento** Na Figura 4.6, são exibidas as curvas relativas ao treinamento desta rede.

Nota-se que na classificação em 2 classes a convergência é muito rápida, ocorrendo em torno de 20 épocas de treinamento. Isto é, o treino da rede aconteceu de forma veloz.

Quando da classificação em 4 classes, a convergência é um pouco mais lenta, ainda não ocorrendo totalmente em 120 épocas. Apesar de não ter atingido a convergência total, o erro já era suficientemente pequeno, assim, não requerendo novo treinamento com maior número de épocas.



(a) 2 classes



(b) 4 classes

Figura 4.6: Estatísticas do treinamento da rede.

**Resultados da classificação** Apresenta-se aqui a matriz confusão com os percentuais de classificação desta rede para 2 e 4 classes.

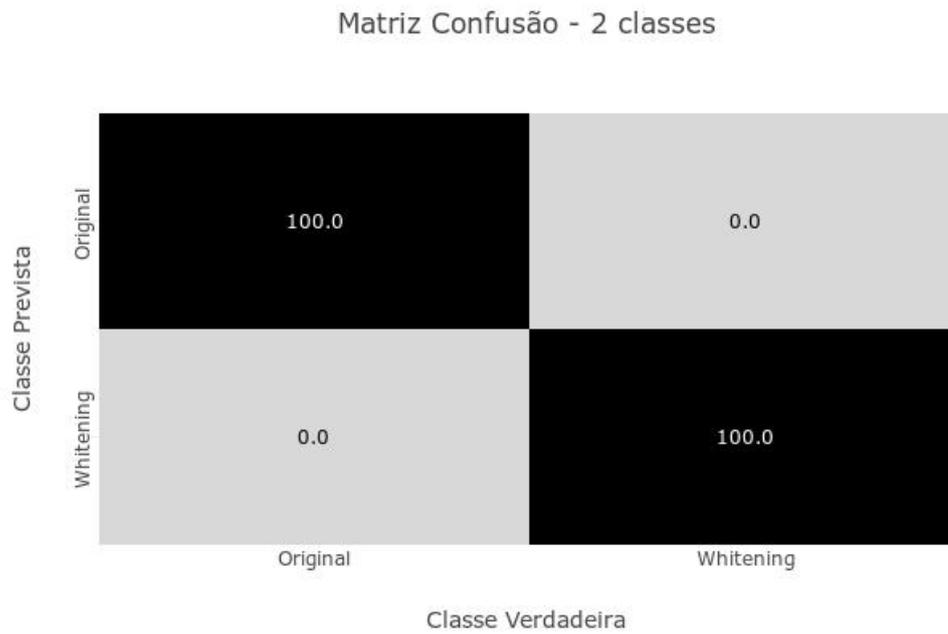


Figura 4.7: Matriz confusão - 2 classes.

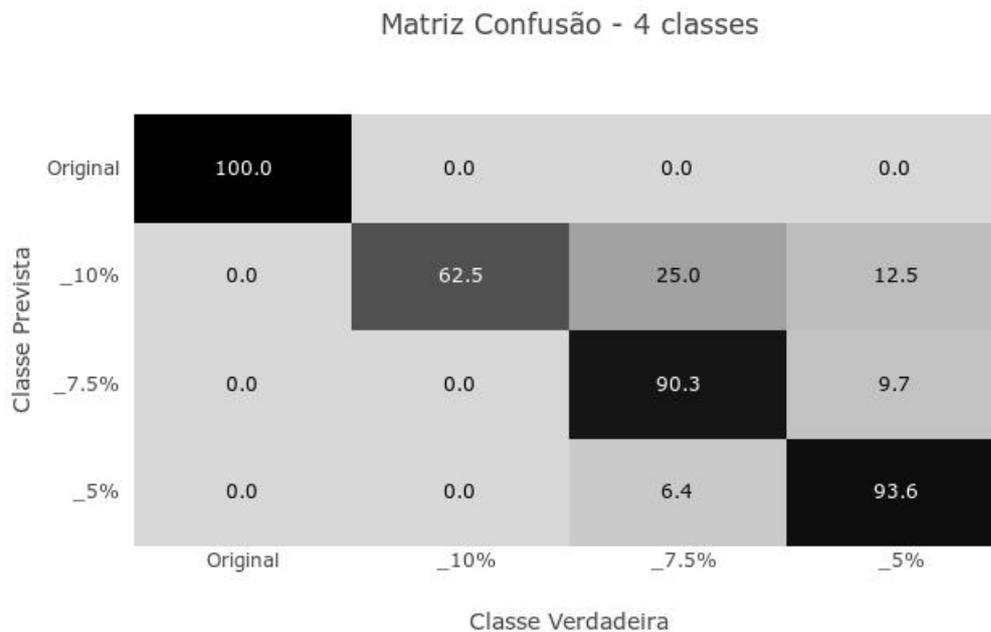


Figura 4.8: Matriz confusão - 4 classes.

Nota-se que esse modelo de rede tem performance excelente quando da classificação tanto em 2 classes quanto em 4 classes. Ao mesmo tempo, pode-se perceber que o resultado de 2 classes, 100% de acerto, aponta fortemente para um caso de

*overfitting*. Entretanto, ao se estudar o conjunto de dados e os recursos da rede, pode-se afirmar que este problema não ocorre.

Analisando o resultado de classificação para 4 classes, observa-se um decaimento na taxa de acerto, apesar de ainda se manter na faixa de classificação muito boa, de acordo com a escala adotada, com 90% de sucesso. Ainda, é importante salientar dois pontos de grande relevância: Primeiro, pode-se inferir que, a separação entre as duas classes principais - original e *whitening* acontece sem erros. O que corrobora para a afirmação de que a classificação em 2 classes apresentado na Figura 4.7 de fato não está apresentando *overfitting*. Além disso, onde atinge-se os maiores percentuais de acerto, como em 5% e 7,5% de deformação, uma maior quantidade de dados para treinamento desses corpos de prova estava disponível.

#### **4.2.7 Classificação com FFT**

Os trabalhos vistos na seção 2.7, apontam quase unanimemente a bons resultados de classificação quando utilizado FFT para alimentar uma RN. Alguns autores [46, 49, 50] usam ainda a técnica de PCA para reduzir a quantidade de dados de entrada, e obtém resultados ainda melhores.

Embora a classificação obtida pela rede alimentada apenas pelo *A-scan* tenha apresentado resultados excelentes, era de grande interesse a verificação de uma rede treinada com os coeficientes de FFT. Não se fez necessário o uso de uma técnica de redução de dados, como o PCA, pois o vetor de entrada já era suficientemente reduzido, com apenas 10 valores. Isso aconteceu pois, no momento do cálculo desta transformada, observou-se que os picos mais relevantes concentravam-se no início do espectro, nos 10 primeiros pontos. Dessa forma, extraiu-se os 10 primeiros pontos dos demais, conforme pode ser observado na Figura 4.9.

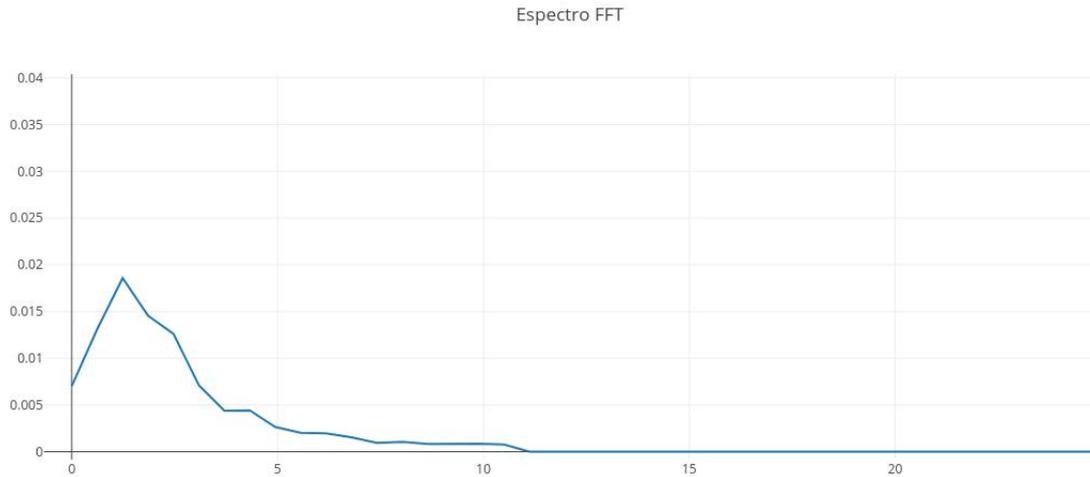
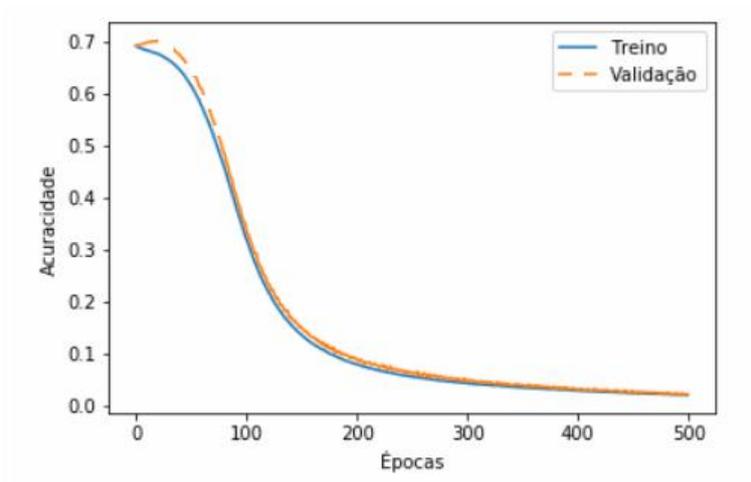


Figura 4.9: Espectro FFT.

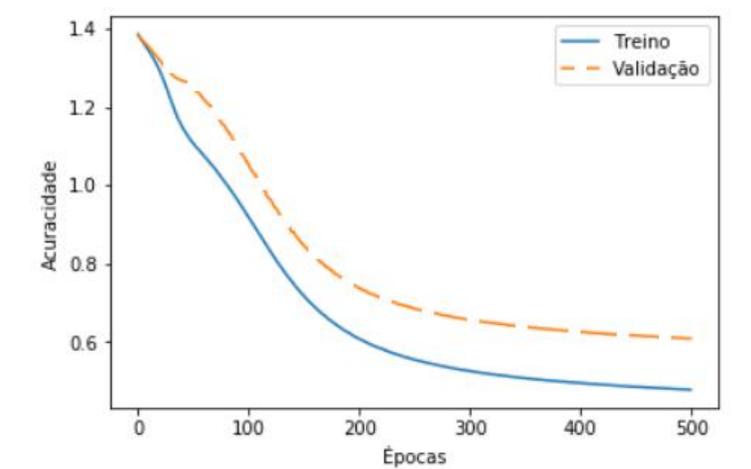
**Estatísticas do Treinamento** Na Figura 4.10, são exibidas as curvas relativas ao treinamento desta rede.

Na classificação em 2 classes, diferente do que foi observado na rede alimentada com o sinal *A-scan* apenas, nota-se que a convergência ocorre, próximo de 400 épocas.

Ao analisar a estatística de treinamento para 4 classes, é possível notar a diferença do formato da curva em relação aos outros gráficos de treinamento. Pode-se perceber facilmente que a velocidade de convergência é menor, e o erro do conjunto de validação termina em torno de 60%, ainda bastante elevado.



(a) 2 classes



(b) 4 classes

Figura 4.10: Estatísticas do treinamento da rede.

**Resultados da classificação** Apresenta-se aqui a matriz confusão com os percentuais de classificação desta rede para 2 e 4 classes.

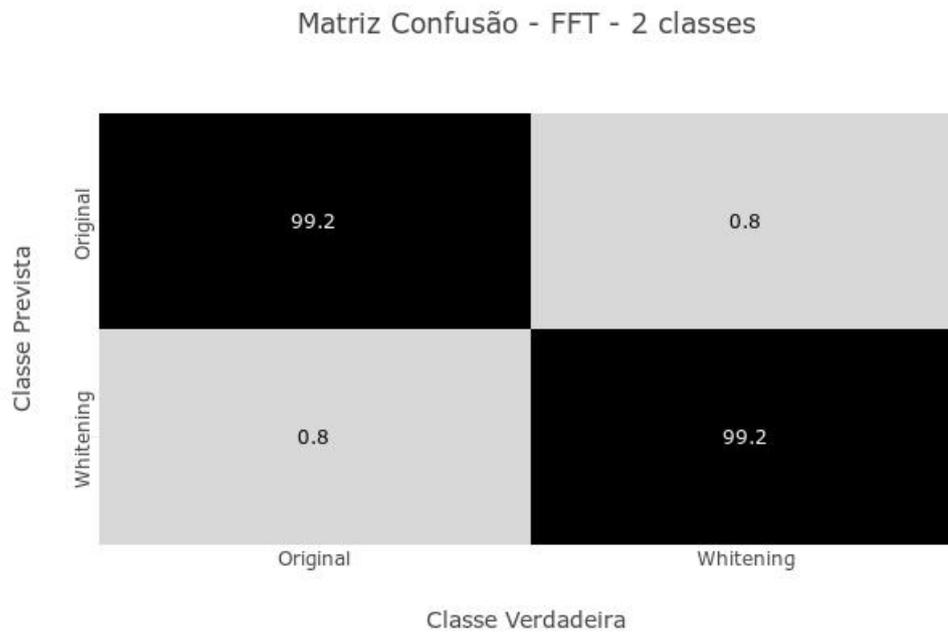


Figura 4.11: Matriz confusão - 2 classes.

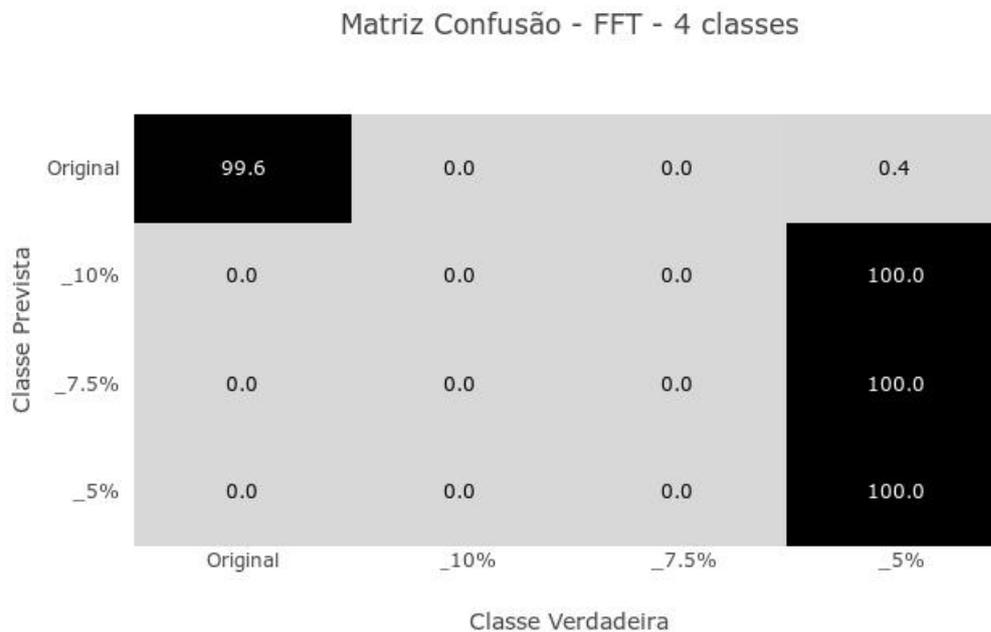


Figura 4.12: Matriz confusão - 4 classes.

Em concordância com o resultado obtido pela rede alimentada com o sinal *A-scan*, a rede alimentada com os coeficientes de FFT também apresentou elevada taxa de acertos, 99,2% na classificação de 2 classes. Novamente, a hipótese da ocorrência

de *overfitting* foi levada em consideração. Porém, de forma semelhante à rede do *A-scan*, na separação em 4 classes, fica evidente que a rede separou sem erros os corpos originais dos com *whitening*. Ao mesmo tempo, seu objetivo de classificar em 4 classes foi significativamente inferior, com 50% de acerto, muito ruim, de acordo com a escala adotada.

## 4.2.8 Classificação com DCT

Como a rede alimentada com FFT não apresentou resultados satisfatórios na classificação em 4 classes, optou-se por fazer um novo teste, alimentando a rede com coeficientes de DCT.

De forma semelhante ao que foi feito com a curva do FFT, observou-se que os picos mais pronunciados estavam concentrados no início do espectro. Assim, foi feito um seccionamento e utilizou-se os 25 primeiros pontos (Figura 4.13) como vetor de entrada da rede neural.

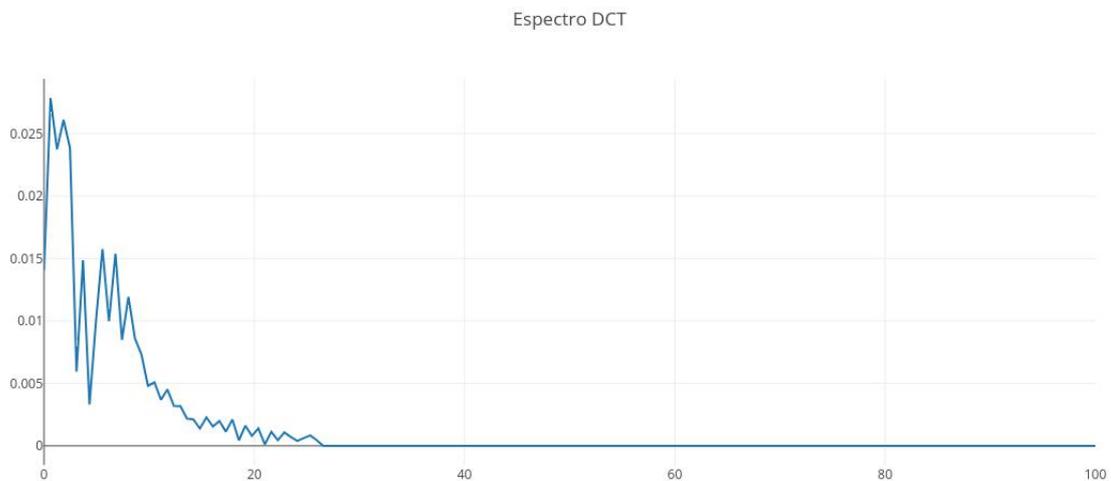


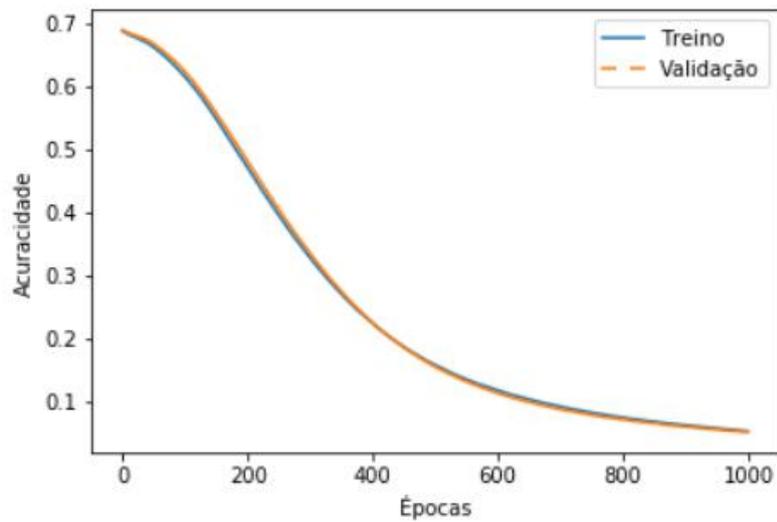
Figura 4.13: Espectro DCT.

**Estatísticas do Treinamento** Na Figura 4.14, são exibidas as curvas relativas ao treinamento desta rede.

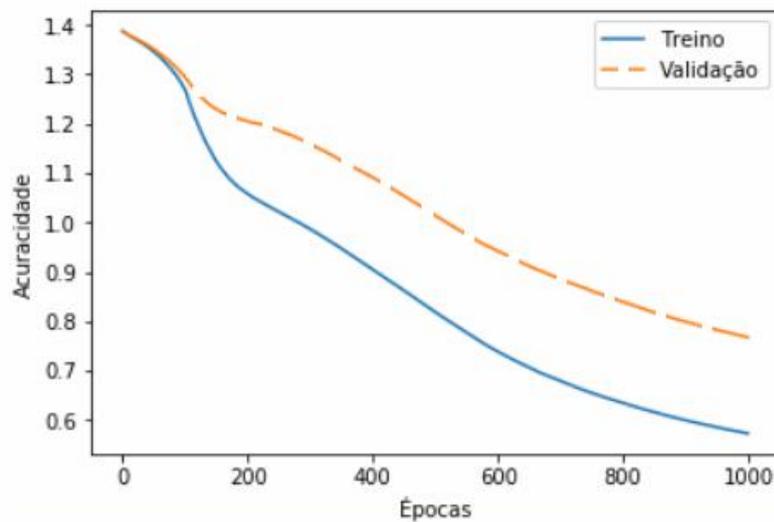
Na classificação em 2 classes, em concordância com os resultados obtidos com as outras configurações de rede, nota-se que ocorre a convergência, embora dessa vez

mais lenta, acontecendo em torno das 900 épocas.

Na análise da estatística de classificação em 4 classes, de forma semelhante ao que pode-se observar no resultado da rede treinada com o sinal tratado com FFT, nota-se que a convergência é extremamente lenta. Além disso, mesmo com 1000 épocas os sinais não se aproximavam da estabilização.



(a) 2 classes



(b) 4 classes

Figura 4.14: Estatísticas do treinamento da rede.

**Resultados da classificação** Apresenta-se aqui a matriz confusão com os percentuais de classificação desta rede para 2 e 4 classes.

Matriz Confusão - DCT - 2 classes

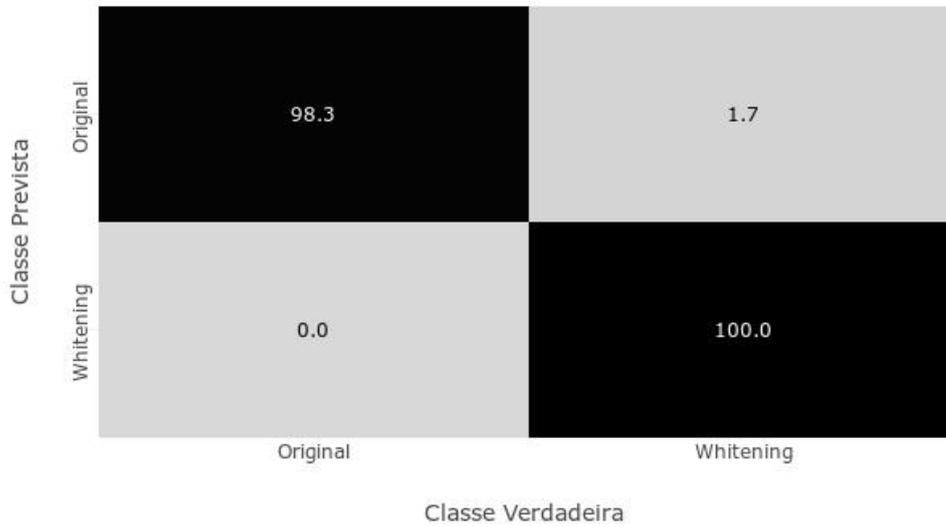


Figura 4.15: Matriz confusão - 2 classes.

Matriz Confusão - DCT- 4 classes

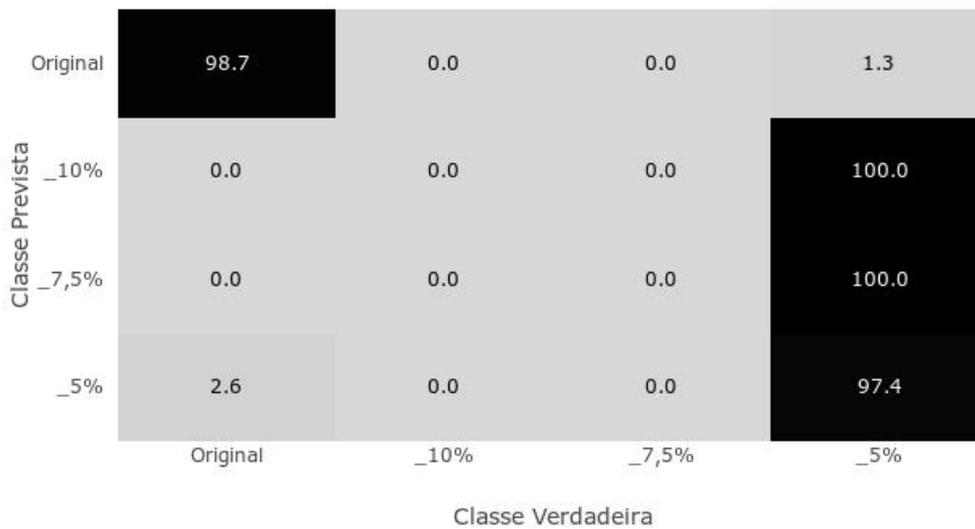


Figura 4.16: Matriz confusão - 4 classes.

Novamente, de forma muito semelhante aos resultados obtidos pela rede FFT, pode-se observar que a classificação em 2 classes é excelente, com acerto de 99%. Contudo, a classificação em 4 classes falha severamente. E mais uma vez, nota-se na

classificação em 4 classes que ocorre uma boa separação entre as amostras originais e com *whitening*.

#### 4.2.9 Comparação das Redes

Após a análise dos resultados das redes alimentadas com três conjuntos distintos de dados, sendo esses *A-scan*, FFT e DCT, é importante fazer uma comparação do desempenho das classificações, conforme pode-se observar na Tabela 4.8.

Tabela 4.8: Comparação da classificação das redes.

	<i>A-Scan</i>	FFT	DCT
2 classes	100%	99,2%	99,2%
4 classes	96,1%	86,7%	85,5%

A rede com melhor resultado foi a alimentada com o sinal de *A-scan* somente. O que vai contra ao que foi encontrado na literatura. Certamente esse resultado deve-se ao fato da seleção apenas da porção correspondente ao sinal retroespalhado dos ensaios de UT. Em seguida, as redes alimentadas tanto com os coeficientes de FFT e DCT, na classificação em 2 classes atingem um rendimento excelente, de acordo com a escala adotada. Ou seja, a separação proposta inicial no trabalho, entre corpos de prova originais contra os com efeito *whitening*, é alcançada com sucesso.

Ao se analisar o desempenho das redes nas classificações em 4 classes, onde separa-se não somente a ocorrência ou não do efeito *whitening*, mas também o percentual de deformação a qual foi submetido o corpo de prova, a rede alimentada com *A-scan* foi capaz de classificar com um desempenho excelente. Ao mesmo tempo, as redes alimentadas com os coeficientes de FFT e DCT, apresentam desempenho bom, de acordo com a escala.

## 4.3 Estudos de Casos

Nesta seção são feitos alguns estudos de casos utilizando as redes treinadas no desenvolvimento deste trabalho.

### 4.3.1 Caso 01 - Particionamento Forçado

No primeiro estudo de caso, decidiu-se fazer uma nova separação nos grupos dos corpos de prova a serem utilizados no treinamento das redes neurais alimentadas por *A-scan* para 2 e 4 classes. Nas redes alimentadas por FFT e DCT, classificou-se apenas com 2 classes.

Esse caso foi denominado particionamento forçado pois foram retirados de maneira arbitrária alguns corpos de prova com e sem efeito *whitening*. Os corpos retirados são exibidos na Tabela 4.9:

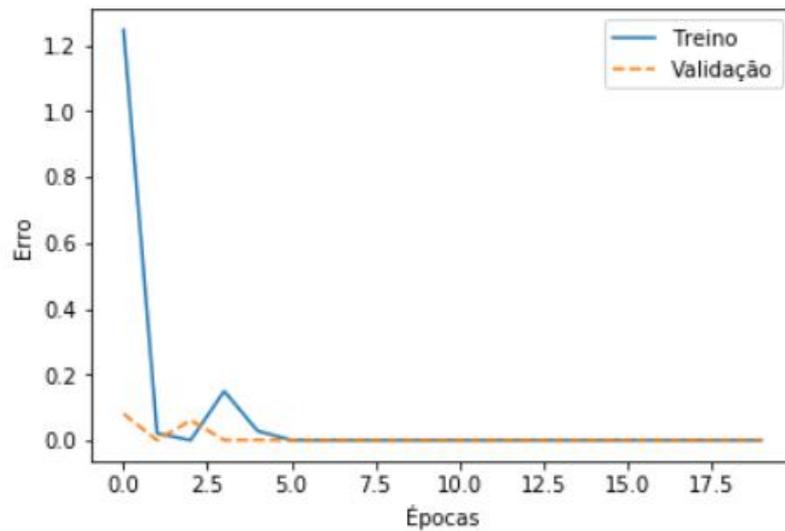
Tabela 4.9: Corpos de Prova Retirados

Original	<i>Whitening</i>
P3	C4 - 5%
A4	C5 - 5%
C4	
C5	

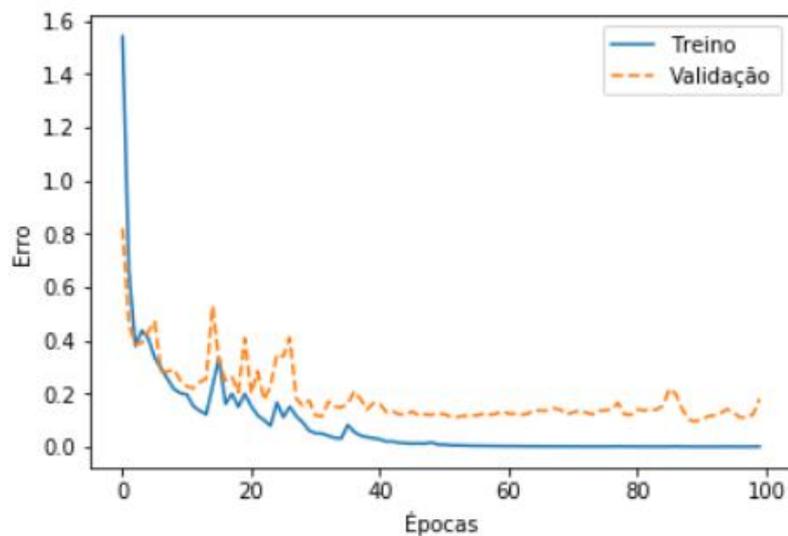
Naturalmente, quando do treinamento da RN, conforme exposto na seção 2.6.2, os dados são particionados em diferentes conjuntos, sendo que os conjuntos de validação e teste nunca são apresentados à rede no momento do treinamento. Porém, retirando corpos de prova inteiros, como feito nesse estudo de caso, garante-se que nenhum ponto sequer daquele corpo será apresentado à rede durante seu treinamento e validação. É importante lembrar que, embora o conjunto de validação não seja usado para treinamento da rede, ele serve para reiniciar o treinamento caso seu resultado de acertos seja baixo.

Dessa forma são apresentados na Figura 4.17 os dados de treinamento da nova

RN.



(a) 2 classes



(b) 4 classes

Figura 4.17: Estatísticas do treinamento da rede.

Nota-se que, assim como nos dados relativos ao treinamento da outra RN alimentada por *A-scan*, a convergência ocorre de forma muito acelerada, com 5 épocas para 2 classes e 40 épocas para 4 classes. O que pode ter ocasionado essa velocidade, principalmente no caso para 2 classes, é ter aumentado o passo de treinamento para 0,01.

**Resultados de Classificação** A seguir são apresentadas as matrizes confusão das novas redes. Primeiro nas Figuras 4.18 e 4.19, a classificação em 2 e 4 classes respectivamente do conjunto total de dados.

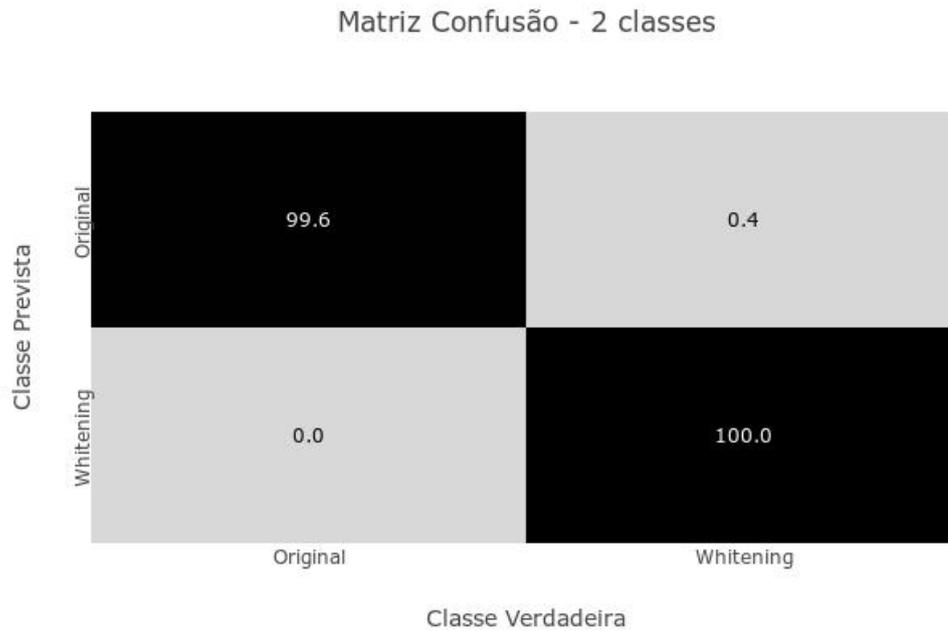


Figura 4.18: Matriz confusão 2 classes.

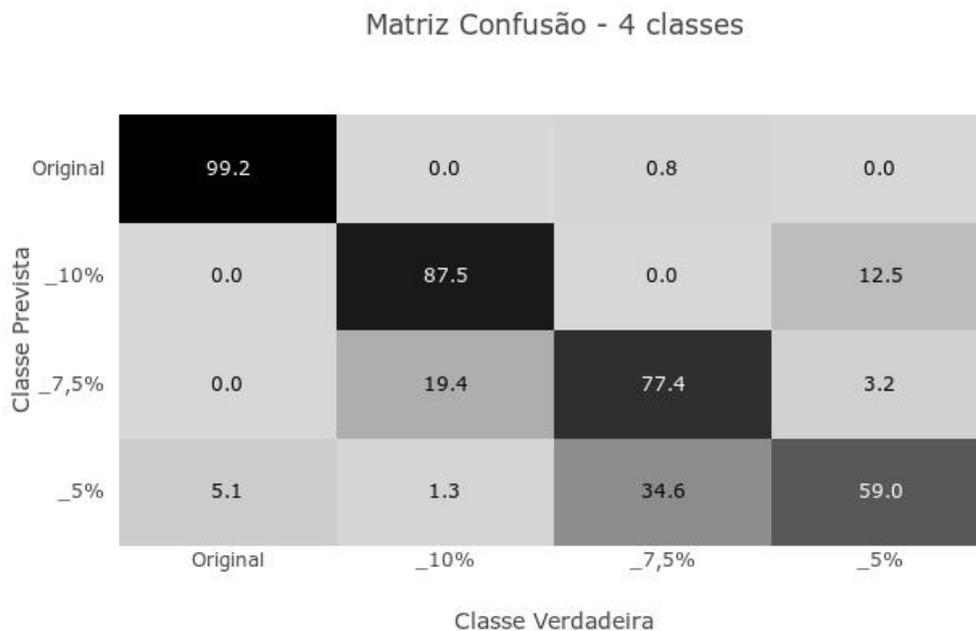


Figura 4.19: Matriz confusão 4 classes.

Nota-se de forma rápida que os resultados são ligeiramente piores, no caso com 4 classes, quando comparados aos das redes neurais treinadas pelos conjuntos de dados completos.

Agora, são apresentados os dados de classificação, na forma de matriz confusão, dos dados que foram excluídos do treinamento, em 2 e 4 classes respectivamente.

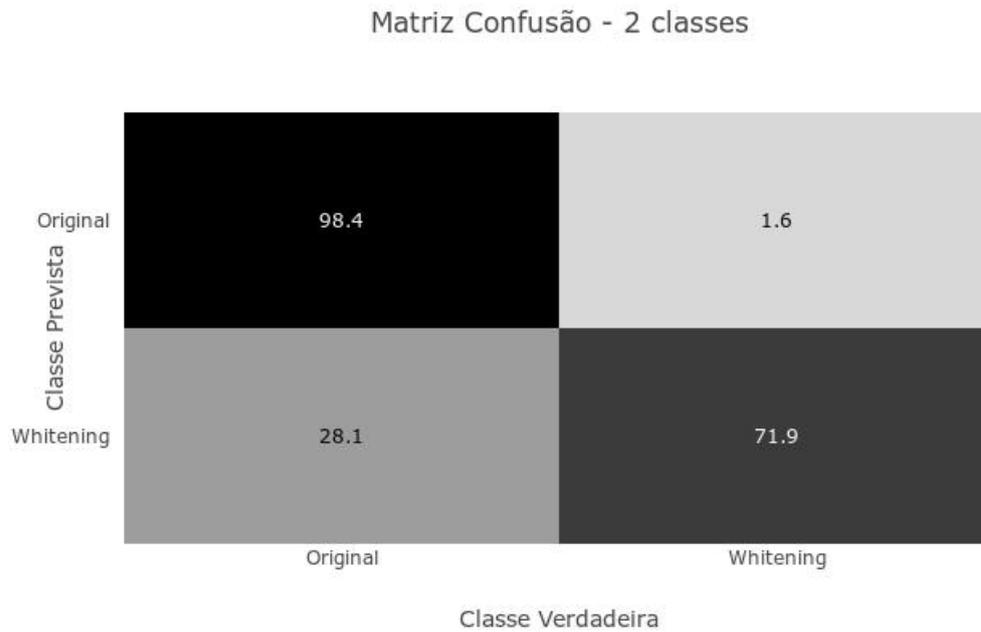


Figura 4.20: Matriz confusão - Excluídos - 2 classes.

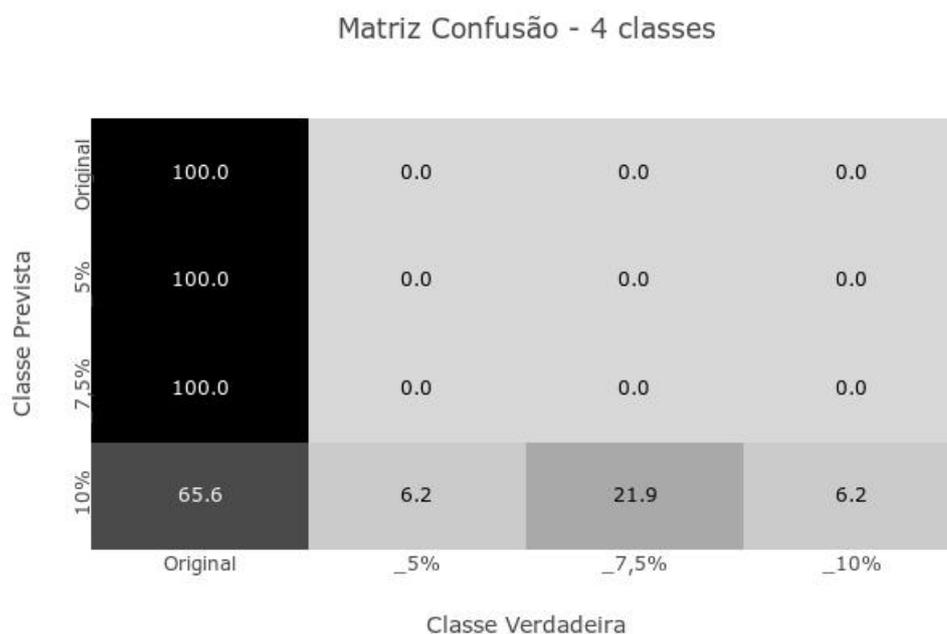


Figura 4.21: Matriz confusão - Excluídos - 4 classes.

Quando faz-se a análise dos resultados dos corpos de prova excluídos o resultado é muito ruim e difere do que era esperado.

### 4.3.2 Caso 02 - Outros Corpos de PVDF

Conforme mencionado em 1, este trabalho é uma sequência da dissertação de mestrado de Nascimento Filho, E. S. [3], que também utilizou corpos de prova PVDF. Os resultados dos ensaios de UT de algumas dessas amostras foram inseridas nas redes aqui treinadas de maneira a ser mais uma forma para avaliar sua capacidade de generalização.

Esses dados foram analisados, filtrados por média e pelo filtro de Savitzky-Golay, recortados na porção do sinal retroespalhado e calculadas as transformadas FFT e DCT.

De início já foram feitas algumas observações, quando da análise exploratória dos dados. Os sinais *A-scan* dessas amostras eram um pouco diferentes das utilizadas no treinamento das redes. Nota-se de forma mais pronunciada essa propriedade ao avaliar o tamanho do intervalo de sinal retroespalhado. Conclui-se então que as

amostras utilizadas em [3], deveria ser menos espessas.

Devido à essa diferença de comprimento do sinal retroespalhado, descartou-se a possibilidade de classificar essa porção do UT com a rede treinada apenas por *A-scan*. Dessa maneira, utilizou-se os dados relativos aos coeficientes calculados pelas transformadas FFT e DCT, já que esses independem do comprimento do sinal *A-scan*.

Os corpos de prova de [3] estavam divididos em NR1 e NR2, cuja diferença era a rota de processamento do PVDF. NR1 segue a mesma rota utilizada neste trabalho, isto é, com resfriamento ao ar e foi exposto a deformações de 2%, 3,5%, 5% e 10%. NR2 tem o resfriamento controlado e dividido em cinco deformações: 0%, 2%, 3,5%, 5% e 10%. Foram medidos 3 diferentes pontos em cada corpo de prova, totalizando 12 e 15 amostras para NR1 e NR2 respectivamente.

**Resultados de Classificação** O resultado referente à classificação de NR1 com a rede alimentada com os sinais FFT é apresentado na matriz confusão, conforme pode ser visto na Figura 4.22 a seguir.

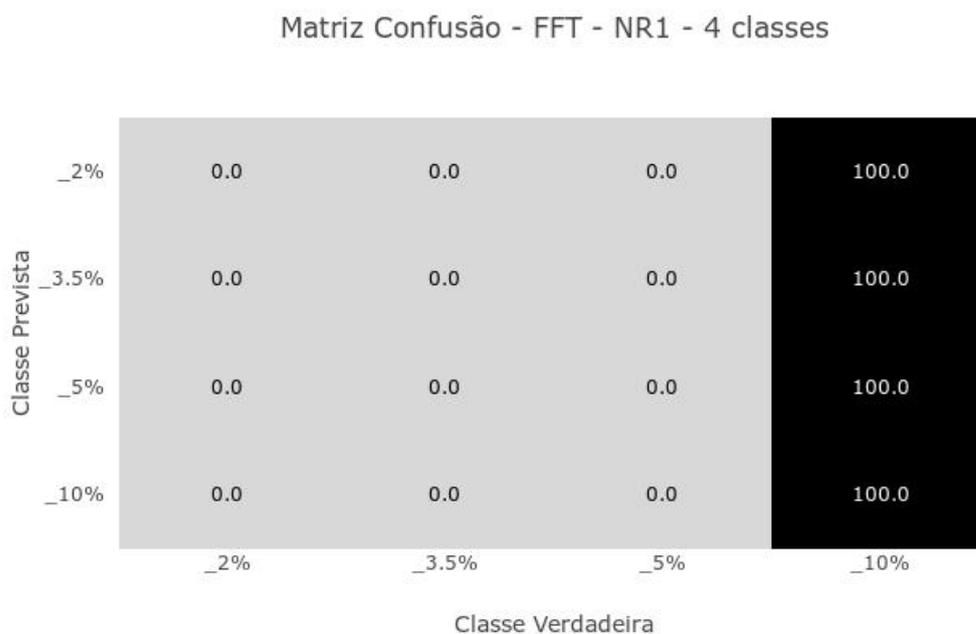


Figura 4.22: Matriz confusão NR1 - 4 classes.

Os resultado referentes à classificação de NR2 com a rede alimentada com os sinais FFT são apresentados nas matrizes confusão, conforme pode ser visto nas Figuras 4.23, 4.24 a seguir.

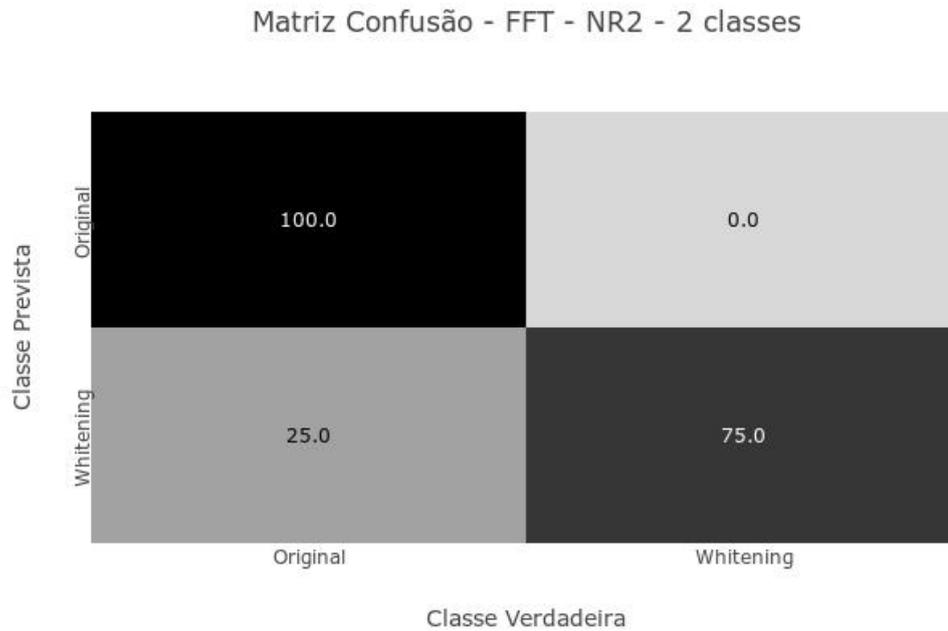


Figura 4.23: Matriz confusão NR2 - 2 classes.

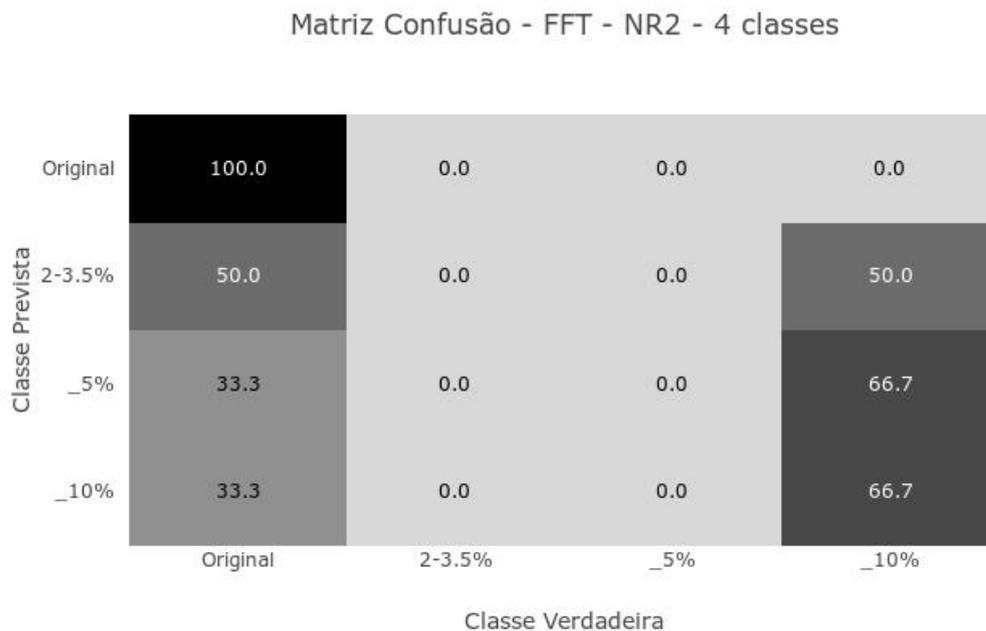


Figura 4.24: Matriz confusão NR2 - 4 classes.

Conforme visto nas figuras, a classificação entre as amostras com e sem *whitening* ocorre e é boa de acordo com a escala, com 87,5% de acertos. Quanto a classificação mais específica, quanto às deformações de cada corpo de prova, os resultados não são bons, conforme já havia sido visto nas seções anteriores.

Dessa maneira, pode-se afirmar que a classificação em 2 classes dessas amostras, mesmo com espessura diferente, pode acontecer de forma bastante satisfatória quanto à presença ou não de efeito *whitening*.

A seguir, são apresentados os resultados das classificações de NR1 em 4 classes pela rede alimentada por DCT.

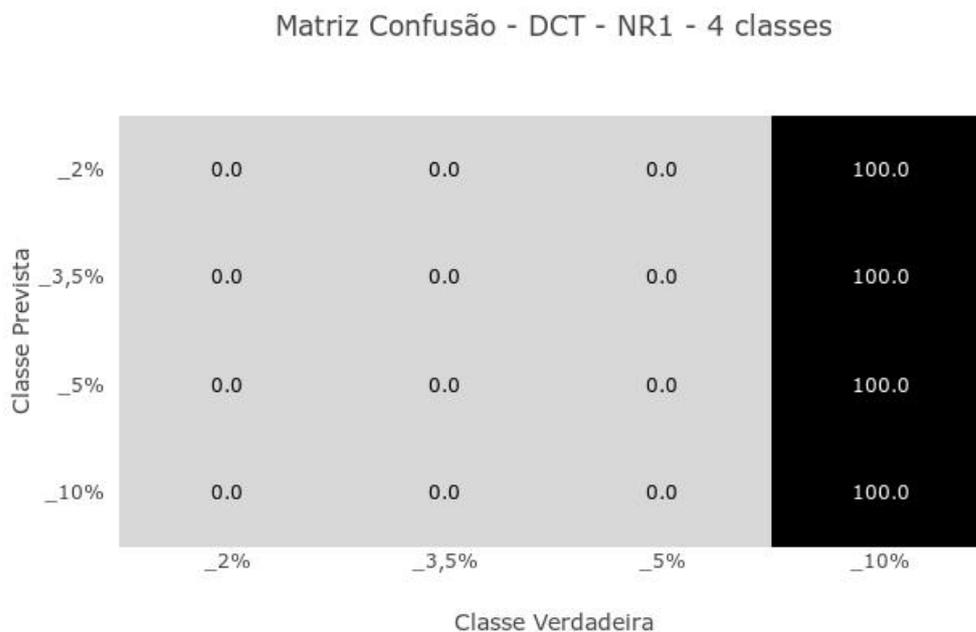


Figura 4.25: Matriz confusão - DCT - NR1 - 4 classes.

Os resultados referentes à classificação de NR2 com a rede alimentada com os sinais DCT são apresentados nas matrizes confusão, conforme pode ser visto nas Figuras 4.26, 4.27 a seguir.

Matriz Confusão - DCT - NR2 - 2 classes



Figura 4.26: Matriz confusão - DCT NR2 - 2 classes.

Matriz Confusão - DCT - NR1 - 4 classes

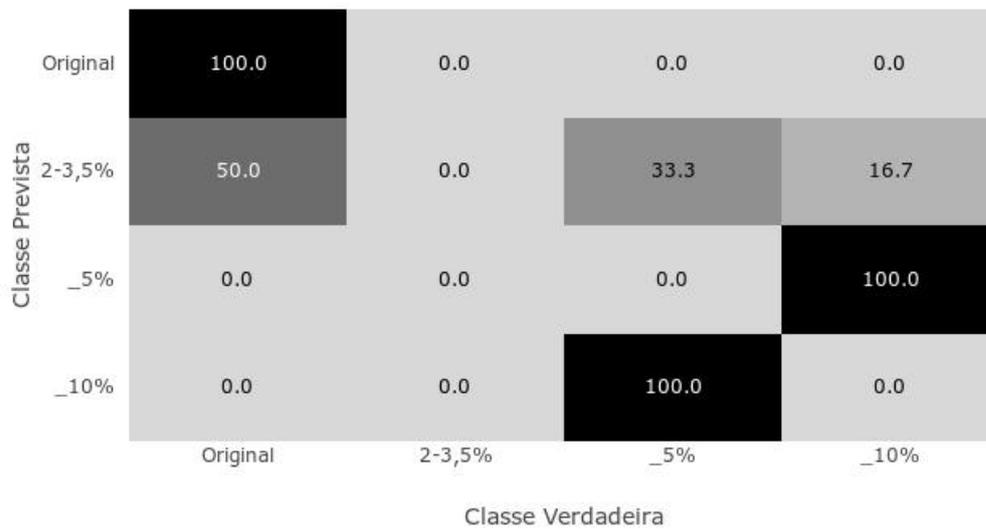


Figura 4.27: Matriz confusão - DCT - NR2 - 4 classes.

Novamente, observa-se a boa classificação entre os corpos de prova com e sem a presença do efeito *whitening*. Porém, a classificação mais específica, em 4 classes é muito ruim.

### 4.3.3 Caso 03 - *Polyfit*

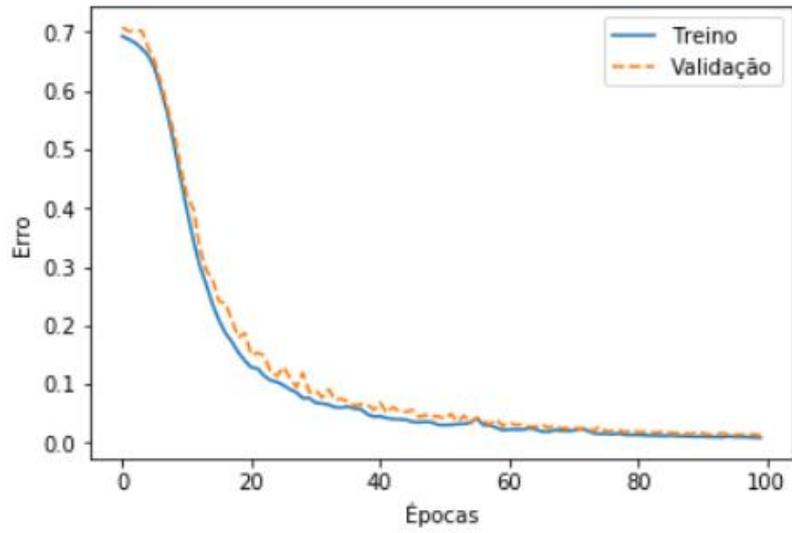
Como a classificação específica quanto ao percentual de deformação aplicado a cada amostra NR não foi satisfatória, elaborou-se então uma nova estratégia para aproveitamento do sinal *A-scan* referente à porção retroespalhada do UT, de forma que não fosse dependente da comprimento deste recorte.

A solução encontrada foi o ajuste da curva desta fração do sinal, por uma regressão polinomial, já que dessa maneira teria-se uma descrição da curva utilizando apenas os coeficientes do polinômio aplicado. A melhor configuração foi atingida ao utilizar um polinômio de grau 9.

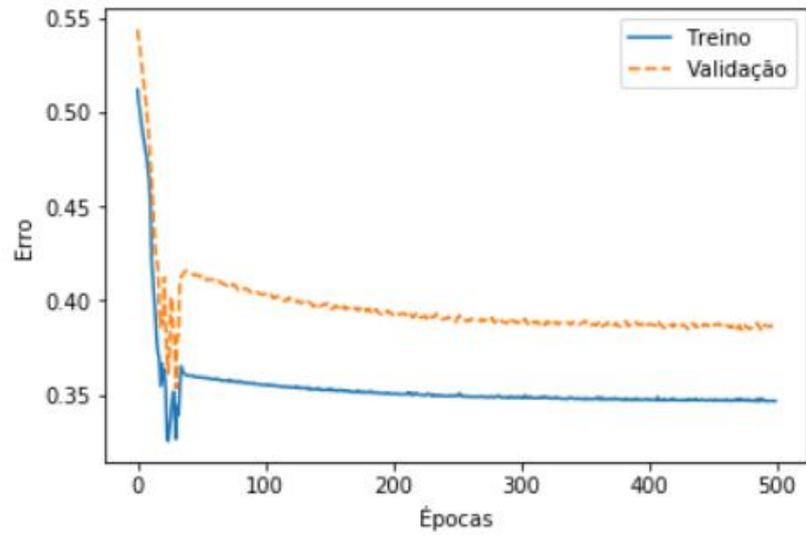
Ao mesmo tempo, devido aos resultados obtidos nesse treinamento, optou-se por testar a mudança no tipo de função de ativação na camada de entrada da rede. Utilizou-se duas estruturas: a primeira sendo igual à utilizada previamente neste trabalho e a segunda, trocando a função de ativação ReLU por *tanh* na primeira camada.

O intuito dessa troca justifica-se no fato de que ao utilizar *tanh*, embora haja saturação rápida para os extremos da função em  $-1$  e  $+1$ , há uma melhor definição de coeficientes negativos, o que não acontece quando é utilizada ReLU, que, ao ligar com valores menores do que zero, iguala-os a zero [35].

Dessa forma, os dados relativos ao treinamento são apresentados a seguir para classificação em 2 classes:



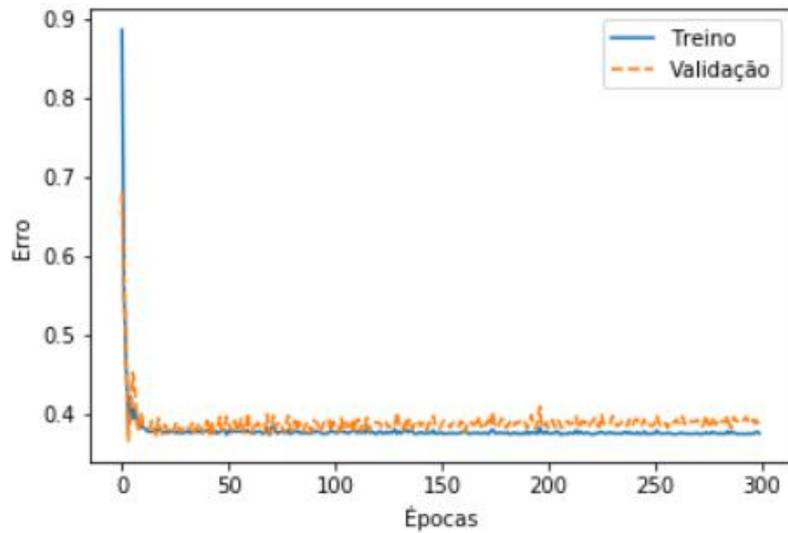
(a) 2 classes - ReLU



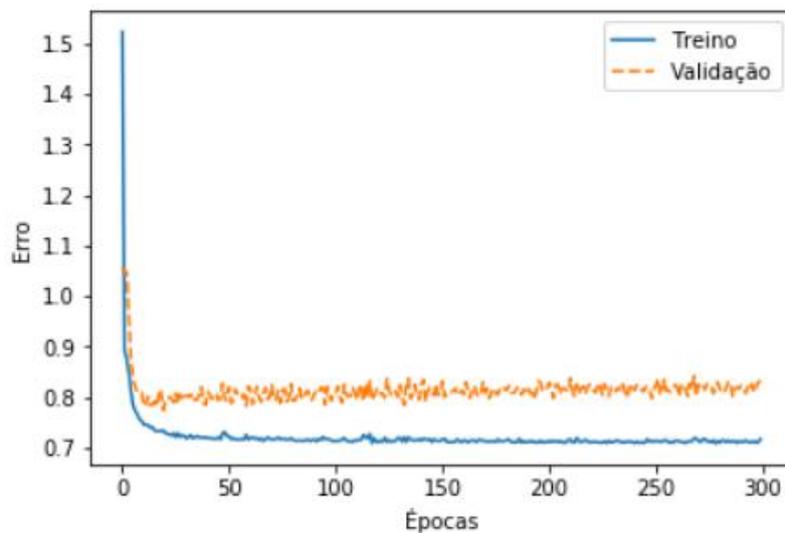
(b) 2 classes - *tanh*

Figura 4.28: Estatísticas do treinamento da rede - ReLU x *tanh*.

Agora a classificação em 4 classes:



(a) 4 classes - ReLU



(b) 4 classes -  $\tanh$

Figura 4.29: Estatísticas do treinamento da rede - ReLU x  $\tanh$ .

Analisando os gráficos das estatísticas de treinamento duas observações podem ser prontamente feitas: primeiro, a convergência ocorre de forma rápida, mas em diferentes patamares. Nota-se que quando feita a classificação em 2 classes, o treinamento converge em torno de 90 épocas tanto quando utilizada a função ReLU quanto com a função  $\tanh$ . Já na classificação em 4 classes, a convergência é muito mais rápida, ocorrendo em torno de 30 épocas para os dois tipos de ativação, porém, observa-se que o erro é bastante elevado, permanecendo na faixa de 0,4 no caso com

ReLU e em 0,7 quando se usa *tanh*.

Para confirmar as observações feitas pelos gráficos das estatísticas de treinamento, são exibidas a seguir as matrizes confusão relativas à cada uma das 4 situações expostas.

Primeiro as classificações em 2 classes:

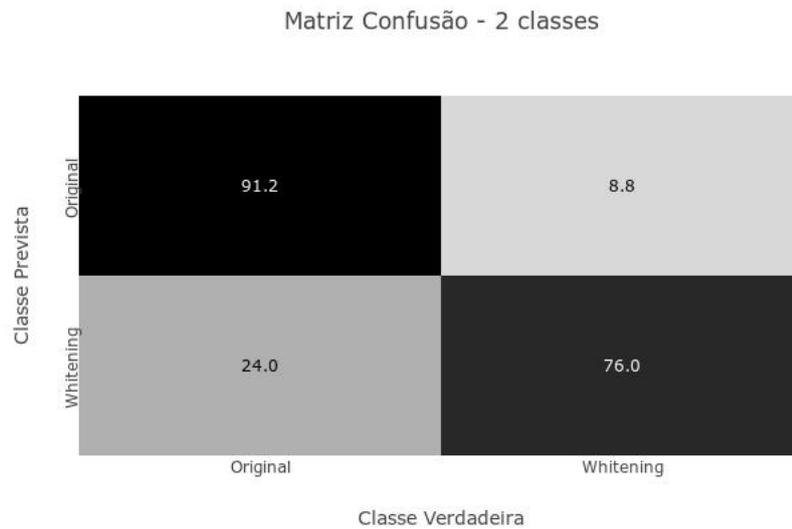


Figura 4.30: Matriz confusão - ReLU - 2 classes.

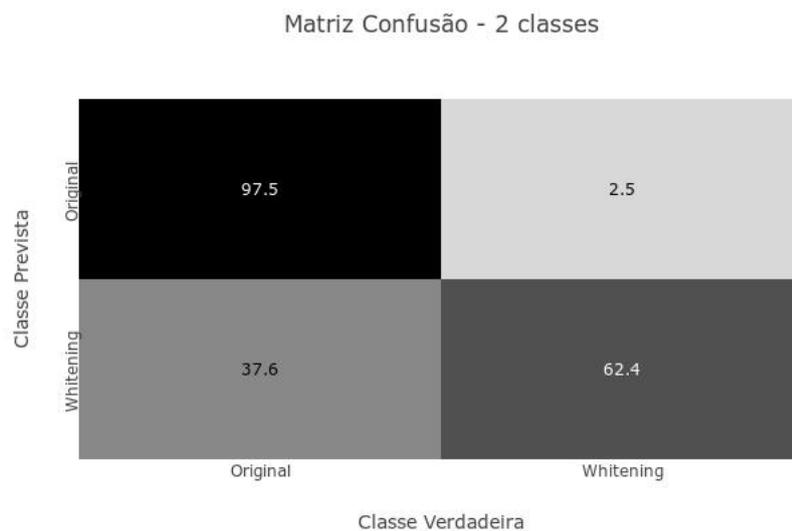


Figura 4.31: Matriz confusão - *tanh* - 2 classes.

Abaixo as matrizes confusão da classificação em 4 classes:

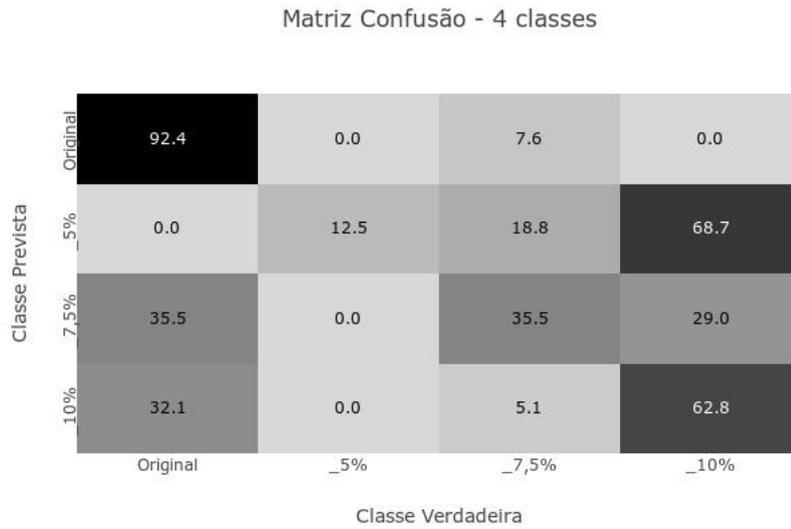


Figura 4.32: Matriz confusão - ReLU - 4 classes.

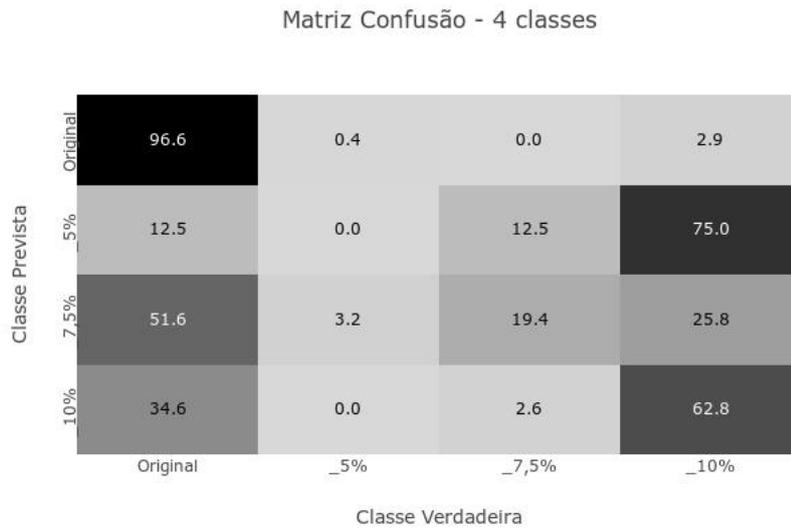


Figura 4.33: Matriz confusão - *tanh* - 4 classes.

Nota-se que a classificação em 2 classes é regular, de acordo com a escala, atingindo 83,6%. Como o objetivo desse estudo de caso era o aproveitamento da porção retroespalhada do UT sem depender de sua dimensão, deve-se então comparar essas classificações com os resultados obtidos por FFT e DCT, conforme visto na Tabela 4.10:

Tabela 4.10: Comparação de classificações

	FFT	DCT	<i>Polyfit</i>	
			ReLU	tanh
2 classes	99,2%	99,3%	83,6%	80%
4 classes	86,7%	85,5%	77,5%	78,3%

Ao analisar a os resultados das matrizes confusão tanto das classificações em 2 classes, quanto das em 4 classes percebe-se que há uma piora nos resultados. Segundo a escala adotada, esses índices são considerados regulares para o caso de 2 classes e ruins quando do caso de 4 classes.

## 4.4 Considerações Finais

O objetivo do trabalho foi alcançado. Fez-se a separação entre as classes com e sem *whitening* conforme o planejado.

De maneira mais detalhada, observa-se que a rede alimentada com o sinal *A-scan* apresentou os melhores resultados de classificação, tanto para 2 quanto para 4 classes. Esse resultado vai no caminho contrário ao que foi visto na literatura, era esperado que essa rede fosse a com pior percentual de acerto.

Possivelmente esse resultado positivo é decorrente do tratamento que foi dado ao sinal *A-scan*, isto é, talvez não tanto devido à filtragem e suavização da curva, mas principalmente por utilizar-se apenas a porção referente à região retrospalhada.

Em relação aos estudos de caso realizados, o primeiro, que utiliza os corpos de prova ensaiados por UT no trabalho de [3], talvez seja o mais relevante da perspectiva da aplicação prática do que foi desenvolvido. Nesse ponto, pode-se confirmar que as redes FFT e DCT têm capacidade de classificar corpos de prova do mesmo material quanto à presença ou não do efeito *whitening*, mesmo com dimensões relativamente diferentes.

No caso 02 é realizado um teste de generalização da rede, enquanto no caso 03 procura-se aumentar a eficiência da classificação. Ambos falham de alguma forma,

em 02 um resultado inesperado aparece e deve ser revisto de forma cuidadosa, ao passo que em 03 a ideia parece ser promissora, mas talvez seja necessário um pouco mais de refino para que seja eficaz.

Do ponto de vista acadêmico, ainda é interessante continuar o estudo para que se atinja a classificação do percentual exato de deformação, como sugerido na próxima seção. Ao mesmo tempo, pode ser que da perspectiva da indústria e suas necessidades de evolução e inovação, esse resultado seja muito útil e de fácil e pronta implementação.

# Capítulo 5

## Conclusão

Ao analisar os resultados obtidos pela classificação das redes desenvolvidas nesse trabalho, pode-se concluir que seu objetivo foi alcançado, isto é, criou-se uma rede neural capaz de classificar de forma eficaz a presença de efeito *whitening* em corpos de prova PVDF.

A rede alimentada com o sinal *A-Scan* obteve os melhores resultados. Porém, notou-se que esse resultado é fortemente dependente das dimensões do corpo de prova.

As outras duas redes, que independem das dimensões do material, alimentadas com FFT e DCT não apresentaram resultados satisfatórios do ponto de vista da precisão proporcionada pela divisão em 4 classes. Entretanto, ambas performam muito bem para separação em 2 classes.

# Capítulo 6

## Sugestões para Trabalhos Futuros

Como sugestão para trabalhos futuros, pode-se citar:

- Fazer a integração desta rede com o equipamento de aquisição de sinais de UT, de forma a criar um sistema integrado que possa apontar em tempo real, durante a inspeção a presença ou não do efeito *whitening*;
- Produzir mais corpos de prova de PVDF, do mesmo formato, mas submetê-los à deformações muito pequenas, desde 1% até 10%, porém de forma gradual. Isto é, fazer pequenos incrementos nessas faixas de deformação. Desse modo, realizando o ensaio de UT e modificando um pouco a rede, pode ser que se consiga uma classificação que determina o percentual exato de deformação do material;
- Tornar essa rede cada vez mais robusta e generalizada, fazendo a alimentação desta com outros corpos de prova, ainda de PVDF, porém de formatos diversos, como um tubo, por exemplo.

# Referências Bibliográficas

- [1] “Driving cost efficiency in deeper water: Next-generation flexible pipe”. Disponível em: <[https://www.geoilandgas.com/sites/geog/files/ge\\_ss\\_compositeflexibles-042516\\_final.pdf](https://www.geoilandgas.com/sites/geog/files/ge_ss_compositeflexibles-042516_final.pdf)>. Acesso em: 2018-abr-02.
- [2] DNV-RP-F206. *Riser Integrity Management*. Standard, Det Norske Veritas, Høvik, Norway, abr. 2008. Disponível em: <<http://rules.dnvgl.com/docs/pdf/DNV/codes/docs/2008-04/RP-F206.pdf>>.
- [3] NASCIMENTO FILHO, E. S. *Caracterização do efeito Whitening em Polifluoreto de vinilideno (PVDF) por ensaio ultrassônico*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2017.
- [4] SHOULBERG, R. H., LANG, E. R. “Stress whitening of polymer blends as a tool for experimental stress analysis: Observations outlined in this technical note indicate that, although the initial experimental results are of low accuracy, the method has some merits”, *Experimental Mechanics*, v. 2, n. 9, pp. 271–273, set. 1962. ISSN: 0014-4851, 1741-2765. doi: 10.1007/BF02324373. Disponível em: <<http://link.springer.com/10.1007/BF02324373>>. Acesso em: 2018-mar-12.
- [5] SILVA, M. D. C. *Geração do efeito whitening em polifluoreto de vinilideno (PVDF) sob carregamento cíclico*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2014.
- [6] SILVA, A. B. D. *Processamento, caracterização e determinação do perfil de polarização do Poli(fluoreto de vinilideno)*. Tese de Mestrado, mar. 2009. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/774>>.
- [7] MANO, E. *Introdução a Polímeros*. 2 ed. São Paulo, Edgar Blücher, 2004. ISBN: 978-85-212-0247-9.
- [8] THIRÉ, R. “Notas de aula da disciplina COT-795 - Princípio dos Materiais Poliméricos”. mar. 2017.

- [9] LAURENTINO, D. A. *Avaliação mecânica do polifluoreto de vinilideno (PVDF) exposto ao dióxido de carbono supercrítico: efeito da temperatura, pressão e histórico fractográfico*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2014.
- [10] MURREN, J. *Failure modes, inspection, testing and monitoring*. Report, Seaflex, Norway, 2007. Disponível em: <[http://www.ptil.no/getfile.php/135780/PDF/P5996RPT01REV02cSeaflex\\_JanMuren.pdf](http://www.ptil.no/getfile.php/135780/PDF/P5996RPT01REV02cSeaflex_JanMuren.pdf)>. Acesso em: 2018-abr-01.
- [11] ASTM D638-14. *Standard Test Method for Tensile Properties of Plastics*. Standard, ASTM International, West Conshohocken, PA - USA, 2014. Disponível em: <<http://www.astm.org/cgi-bin/resolver.cgi?D638>>.
- [12] CANEVAROLO JR., S. V. *Ciência dos polímeros: um texto básico para tecnólogos e engenheiros*. São Paulo, Artliber, 2002. ISBN: 978-85-88098-10-7.
- [13] CHERRY, B. W., HIN, T. S. “Stress whitening in polyethylene”, *Polymer communications*, p. 3, dez. 1981.
- [14] LAIARINANDRASANA, L., BESSON, J., LAFARGE, M., et al. “Temperature dependent mechanical behaviour of PVDF: Experiments and numerical modelling”, *International Journal of Plasticity*, v. 25, n. 7, pp. 1301–1324, jul. 2009. ISSN: 07496419. doi: 10.1016/j.ijplas.2008.09.008. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0749641908001514>>.
- [15] BRUSSELLE-DUPEND, N., ROSENBERG, E., ADRIEN, J. “Characterization of cavitation development while tensile testing PVF2 using 3D X-ray microtomography”, *Materials Science and Engineering: A*, v. 530, pp. 36–50, dez. 2011. ISSN: 09215093. doi: 10.1016/j.msea.2011.09.009. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0921509311009695>>.
- [16] KOCH, T., SEIDLER, S. “Correlations Between Indentation Hardness and Yield Stress in Thermoplastic Polymers”, *Strain*, v. 45, n. 1, pp. 26–33, fev. 2009. ISSN: 00392103, 14751305. doi: 10.1111/j.1475-1305.2008.00468.x. Disponível em: <<http://doi.wiley.com/10.1111/j.1475-1305.2008.00468.x>>.
- [17] LIU, Y., KENNARD, C. H. L., CALOS, N. J. “Characterization of stress-whitening of tensile yielded isotactic polypropylene”, p. 9.

- [18] LIU, X., GUO, M., WEI, W. “Stress-Whitening of High-Impact Poly(propylene): Characterization and Analysis”, *Macromolecular Symposia*, v. 312, n. 1, pp. 130–138, fev. 2012. ISSN: 10221360. doi: 10.1002/masy.201100008. Disponível em: <<http://doi.wiley.com/10.1002/masy.201100008>>.
- [19] “Introduction to Ultrasonic Testing”. Disponível em: <<https://www.nde-ed.org/EducationResources/CommunityCollege/Ultrasonics/Introduction/description.php>>. Acesso em: 2018-mar-13.
- [20] PEREIRA, G. R. “Notas de aula da disciplina COT-729 - Ensaios Não Destrutivos Avançados”. 2017.
- [21] SMITH, S. W. *The Scientist and Engineer’s Guide to Digital Signal Processing*. 1 ed. California, California Technical Pub, 1997. Disponível em: <<http://www.dspguide.com>>.
- [22] SILVA NETO, R. J. *Avaliação de técnica de medida do tempo de percurso de ondas ultrassônicas baseado no espectro de frequência*. Dissertação de M.Sc., CNEN/IEN, Rio de Janeiro, RJ, Brasil, 2014.
- [23] DE SIQUEIRA, N. C., ARENAS, M. P., DE ALMEIDA, P. D., et al. “Characterization of the aging state of modified HP steels by ultrasonic signal processing and the effect of creep voids in the interdendritic region”, *Journal of Materials Research and Technology*, 2018. ISSN: 2238-7854. doi: <https://doi.org/10.1016/j.jmrt.2018.05.014>. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2238785417307536>>.
- [24] “SURFING THE WAVELETS”. Disponível em: <<http://www.wavelet.org/tutorial/whistory.htm>>. Acesso em: 2018-mar-14.
- [25] LEE, K. “Feature extraction schemes for ultrasonic signal processing”. In: *5th International Conference on Computer Sciences and Convergence Information Technology*, pp. 366–372, nov. 2010. doi: 10.1109/ICCIT.2010.5711085.
- [26] DUHAMEL, P., VETTERLI, M. “Fast fourier transforms: A tutorial review and a state of the art”, *Signal Processing*, v. 19, n. 4, pp. 259 – 299, 1990. ISSN: 0165-1684. doi: [https://doi.org/10.1016/0165-1684\(90\)90158-U](https://doi.org/10.1016/0165-1684(90)90158-U).
- [27] SAVITZKY, A., GOLAY, M. J. E. “Smoothing and Differentiation of Data by Simplified Least Squares Procedures.” *Analytical Chemistry*, v. 36, n. 8, pp. 1627–1639, jul. 1964. ISSN: 0003-2700. doi: 10.1021/ac60214a047. Disponível em: <<https://doi.org/10.1021/ac60214a047>>.

- [28] NISHIDA, E. N. *Propriedades da Filtragem de Savitzky-Golay Aplicadas na Identificação de Complexos QRS em Sinais de Eletrocardiograma*. Dissertação de M.Sc., UNIFEI, Itajubá, MG, Brasil, 2017.
- [29] Y. SHOHAM, E. BRYNJOLFSSON, J. L. *2017 AI Index Report*. Relatório técnico, Artificial Intelligence Index, 2017. Disponível em: <<https://aiindex.org>>. Acesso em: 2018-mar-12.
- [30] THOMSEN, M. “Microsoft’s Deep Learning Project Outperforms Humans In Image Recognition”, *Forbes*, fev. 2015. Disponível em: <<https://www.forbes.com/sites/michaelthomsen/2015/02/19/microsofts-deep-learning-project-outperforms-humans-in-image-recognition/>>. Acesso em: 2018-mar-13.
- [31] HERN, A. “Computers are now better than humans at recognising images”, *The Guardian*, maio 2015. Disponível em: <<http://www.theguardian.com/global/2015/may/13/baidu-minwa-supercomputer-better-than-humans-recognising-images>>. Acesso em: 2018-mar-13.
- [32] MOURA JUNIOR, N. N. *Modelo Neural para Correção da Estimção de energia para um detector de partículas de propósito geral*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2014.
- [33] MCCULLOCH, W., PITTS, W. “A logical calculus of the ideas immanent in nervous activity”, *The bulletin of mathematical biophysicss*, v. 5, n. 4, pp. 115–133, 1943.
- [34] DA SILVA, I. N. *Redes Neurais Artificiais Para Engenharia e Ciencias Aplicadas: Fundamentos Teoricos e Aspectos Praticos*. São Paulo, ARTLIBER, 2016. ISBN: 8588098873.
- [35] HAYKIN, S. *Neural Networks and Learning Machines*. N. 10. 3 ed. Ontario, Canada, Prentice Hall, 2009. ISBN: 978-0-13-147139-9.
- [36] LECUN, Y., BENGIO, Y., HINTON, G. “Deep learning”, *Nature*, v. 521, pp. 436, maio 2015. Disponível em: <<http://dx.doi.org/10.1038/nature14539>>.
- [37] “TensorFlow Guide”. Disponível em: <<https://www.tensorflow.org/guide/>>. Acesso em: 2018-aug-18.
- [38] *Keras Documentation*, 2018. Disponível em: <<https://keras.io/>>. Acesso em: 2018-Ago-15.

- [39] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., et al. “Scikit-learn: Machine Learning in Python”, *Journal of Machine Learning Research*, v. 12, pp. 2825–2830, 2011.
- [40] “ReLU : Not a Differentiable Function: Why used in Gradient Based Optimization? and Other Generalizations of ReLU.” Disponível em: <https://medium.com/@kanchansarkar/relu-not-a-differentiable-function-why-used-in-gradient-based-optimization>. Acesso em: 2018-aug-18.
- [41] CALÔBA, L. P. “Redes Neurais em Modelagem de Sistemas”. In: Aguirre, L. A. (Ed.), *Enciclopédia de Automática*, 1 ed., v. 3, pp. 325–344, São Paulo, Brasil, Edgar Blucher, 2007.
- [42] DIPIETRO, R. “A Friendly Introduction to Cross-Entropy Loss”. Disponível em: <https://rdipietro.github.io/friendly-intro-to-cross-entropy-loss/>. Acesso em: 2018-aug-18.
- [43] CALÔBA, L. P. “Notas de aula da disciplina CPE-721 - Redes Neurais Feedforward”. mar. 2017. Disponível em: <http://www.lps.ufrj.br/~caloba/CPE721/>.
- [44] KINGMA, D. P., BA, J. “Adam: A Method for Stochastic Optimization”, *CoRR*, v. abs/1412.6980, 2014. Disponível em: <http://arxiv.org/abs/1412.6980>.
- [45] BROWNLEE, J. “Gentle Introduction to the Adam Optimization Algorithm for Deep Learning”. 2017. Disponível em: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>. Acesso em: 2018-aug-18.
- [46] WANG, Y. “Ultrasonic Flaw Signal Classification using Wavelet Transform and Support Vector Machine”, v. 11, n. 12, pp. 9.
- [47] KESHARAJU, M., NAGARAJAH, R. “Feature selection for neural network based defect classification of ceramic components using high frequency ultrasound”, *Ultrasonics*, v. 62, pp. 271–277, set. 2015. ISSN: 0041624X. doi: 10.1016/j.ultras.2015.05.027. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0041624X15001468>.
- [48] MIRONOVOVA, M., BÍLA, J. “Fast fourier transform for feature extraction and neural network for classification of electrocardiogram signals”. pp. 1–6. IEEE, jul. 2015. ISBN: 978-1-4799-8267-7. doi: 10.1109/FGCT.2015.

7300244. Disponível em: <<http://ieeexplore.ieee.org/document/7300244/>>.

- [49] SIMAS FILHO, E. F., MARTINEZ, L., ALBUQUERQUE, M., et al. “Efficient feature extraction for an automatic ultrasound testing decision support system”. In: *Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, 2014 IEEE International*, pp. 1279–1284. IEEE, 2014.
- [50] CRUZ, F., SIMAS FILHO, E., ALBUQUERQUE, M., et al. “Efficient feature selection for neural network based detection of flaws in steel welded joints using ultrasound testing”, *Ultrasonics*, v. 73, pp. 1–8, jan. 2017. ISSN: 0041624X. doi: 10.1016/j.ultras.2016.08.017. Disponível em: <<http://linkinghub.elsevier.com/retrieve/pii/S0041624X16301500>>.
- [51] BARRY, T., KESHARAJU, M., NAGARAJAH, C., et al. “Defect characterisation in laminar composite structures using ultrasonic techniques and artificial neural networks”, *Journal of Composite Materials*, v. 50, n. 7, pp. 861–871, mar. 2016. ISSN: 0021-9983, 1530-793X. doi: 10.1177/0021998315584651. Disponível em: <<http://journals.sagepub.com/doi/10.1177/0021998315584651>>.
- [52] SAMBATH, S., NAGARAJ, P., SELVAKUMAR, N. “Automatic Defect Classification in Ultrasonic NDT Using Artificial Intelligence”, *Journal of Nondestructive Evaluation*, v. 30, n. 1, pp. 20–28, mar. 2011. ISSN: 0195-9298, 1573-4862. doi: 10.1007/s10921-010-0086-0. Disponível em: <<http://link.springer.com/10.1007/s10921-010-0086-0>>.
- [53] SANTIN, J. L. *Ultra-Som: Técnica e Aplicação*. 2 ed. Curitiba, Paraná, Artes Gráficas e Editora Unificado, 2003.
- [54] “Tutorial sobre Phased Array: Dinâmica do Declive da Onda”. 2018. Disponível em: <<https://www.olympus-ims.com/pt/ndt-tutorials/transducers/wave-front/>>. Acesso em: 2018-aug-28.

# Apêndice A

## Códigos

```
# # Importing Packages

###Delete after drafting
import matplotlib.pyplot as plt
# plt.style.use(['dark_background'])

import scipy.io
import numpy as np
import plotly
from plotly.graph_objs import *
plotly.offline.init_notebook_mode()    #allows Plotly to work in Offline mode

from scipy.signal import find_peaks, savgol_filter
from scipy.fftpack import fft, dct
import pywt

class ImportFile:
    def __init__(self, group, specimen, sweepNumber = None):
```

```

self.group = group
self.specimen = specimen
if sweepNumber is None:
    self.sweepNumber = 1

self.fileName = str('/home/guilherme/Documents/Net_03/UT_JUL_18/UT_PVDF_Gr_
self.sourceFile = scipy.io.loadmat(self.fileName, variable_names = ('Y_tota
self.timeAxis = self.sourceFile['X_total']
self.amplitudeAxis = self.sourceFile['Y_total']

class Denoise:
    def __init__(self, signal, indexNumb):
        #Divide scan list into sweep order
        self.indexNumb = indexNumb
        self.amplitudeAxis = signal.flatten()
        self.scan_01 = self.amplitudeAxis[0:16]
        self.scan_02 = self.amplitudeAxis[16:32]
        self.scan_03 = self.amplitudeAxis[32:48]

        #Noise reduction
        self.signalMean = (self.scan_01+self.scan_02+self.scan_03)/3
        #Savitzky-Golai filter
        self.denoisedSignal = savgol_filter(self.signalMean[self.indexNumb], window
#         self.denoisedSignal = self.signalMean

class FindMaxCoord:
    def __init__(self, signal):
#         self.indexNumb = indexNumb

```

```

#         self.signal = signal.denoisedSignal[self.indexNumb]    #Tranform the array
self.signal = signal.denoisedSignal[0]
self.x_axis = self.signal.argmax()    #Return the X coordinate of the maxi
self.y_axis = max(self.signal)    #Return the Y coordinate for the maximum v
self.coord = [self.x_axis+0, self.y_axis]

def checkPeak (self):
    self.peak, _ = find_peaks(self.signal, height=0.07, distance=120)
    self.lista = []
    for i in self.peak:
        self.lista.append(self.signal[i])

    return self.peak, self.lista

def trimSignal(self):
    self.minPoint = np.amin(self.signal)    #Find signal min value
    self.auxList = self.signal[self.signal.argmin():]    #Auxiliary trimmed list
    self.trimmedList = self.auxList[(self.auxList.argmax()+222):(self.auxList.a

#         return self.minPoint, self.signal, self.auxList, self.trimmedList
return self.trimmedList

def coordinate(self):
#         return '{} {}'.format(self.x_axis, self.y_axis)    #Return the X,Y co
return self.coord

class PolyFit:
    def __init__(self, signal):
        self.signal = signal

```

```
self.fit = np.polyfit(np.linspace(0,1,len(self.signal)), self.signal, 9)
```

```
class PlotSignal:
```

```
    def __init__(self, cleanSignal):
```

```
        self.cleanSignal = cleanSignal
```

```
        self.aux = self.cleanSignal.denoisedSignal[0] #where indexNumb should be c
```

```
        self.peak, self.lista = FindMaxCoord(self.cleanSignal).checkPeak()
```

```
        self.trimmedList = FindMaxCoord(self.cleanSignal).trimSignal()
```

```
        trace1 = {"y": self.aux, "mode": "lines", "type": "scatter"}
```

```
        trace2 = {"y": self.lista, "x": self.peak, "mode": "markers", "type": "scatter"}
```

```
#        trace3 = {"y": self.trimmedList, "mode": "lines", "type": "scatter"}
```

```
        data = Data([trace1, trace2])
```

```
        fig = Figure(data=data)
```

```
        plotly.offline.iplot(fig)
```

```
class Filters:
```

```
    def __init__(self, cleanSignal, samplingFrequency=None):
```

```
        self.trimmedSignal = FindMaxCoord(cleanSignal).trimSignal()
```

```
        self.length = len(self.trimmedSignal)
```

```
        if samplingFrequency == None:
```

```
            self.samplingFrequency = 1600
```

```
        else: self.samplingFrequency = samplingFrequency
```

```
        self.timeVector = np.linspace(0, 7.9992e-6, num = self.length)
```

```
        self.frequencyVector = self.samplingFrequency * np.arange(0, (self.length//
```

```

self.fft = fft(self.trimmedSignal)
self.fft_abs = np.abs(self.fft[0:self.length//2])/self.length
self.fft_abs = self.fft_abs[0:10]
self.dct = dct(self.trimmedSignal)
self.dct_abs = np.abs(self.dct[0:self.length//2])/self.length
self.dct_abs = self.dct_abs[0:25]

def FFT(self):
    self.fft = fft(self.trimmedSignal)
    self.fft_abs = np.abs(self.fft[0:self.length//2])/self.length
    self.fft_abs = self.fft_abs[0:10]
    return self.fft_abs

def DCT(self):
    self.dct = dct(self.trimmedSignal)
    self.dct_abs = np.abs(self.dct[0:self.length//2])/self.length
    self.dct_abs = self.dct_abs[0:25]
    return self.dct_abs

def DWT(self):
    self.cA, self.cD = pywt.dwt(self.trimmedSignal, 'sym5')
    self.frequencies = [10000, 50000, 33000, 1000, 100]
    self.coefs, self.freq = pywt.cwt(self.trimmedSignal, self.frequencies, 'morl')
    return self.coefs, self.freq

def plotFFT(self):
    trace_fft = {"y": self.fft_abs, "x": self.frequencyVector, "mode": "lines",
    trace_dct = {"y": self.dct_abs, "x": self.frequencyVector, "mode": "lines",

```

```

data = Data([trace_fft, trace_dct])

layout = Layout(xaxis = dict(range=[0,30]))

fig_fft = Figure(data=data, layout=layout)

plotly.offline.iplot(fig_fft)

# Begins list

fft_list_02=np.zeros([10,])
dct_list_02=np.zeros([25,])
signal_list_02 = np.zeros([2590,])
polyfit_list_02 = np.zeros([10,])

# bons = [('A',2)],('A',4),('C',1),('C',2),('C',3),('C',4),('C',5)]
group = ['C']
for g in group:
    for CP in range(4,6):
        for ind in range(16):
            cleanSignal = Denoise(ImportFile(g, CP).amplitudeAxis, ind)
#             plt.plot(FindMaxCoord(cleanSignal).trimSignal())
            signal_list_02=np.vstack((signal_list_02, FindMaxCoord(cleanSignal).trimSignal()))
            fft_list_02=np.vstack((fft_list_02, Filters(cleanSignal).FFT()))
            dct_list_02=np.vstack((dct_list_02, Filters(cleanSignal).DCT()))
            polyfit_list_02=np.vstack((polyfit_list_02, PolyFit(FindMaxCoord(cleanSignal).trimSignal())))
#             PlotSignal(cleanSignal)
#             Filters(cleanSignal).plotFFT()
            print(g, CP, ind)
#             plt.show()

# lista_teste = np.delete(lista_teste,0,0)

```

```

import keras
from tensorflow.keras.utils import to_categorical
import numpy as np

class ShuffleData:
    def __init__(self, Data, Labels):
        self.Data = Data
        self.Labels = Labels

        self.combinedList = list(zip(self.Data, self.Labels))
        np.random.shuffle(self.combinedList)
        self.Data[:,], self.Labels[:,] = zip(*self.combinedList)

    def labelMatrix(self):
        return to_categorical(self.Labels)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()

### Building stack model

model = keras.Sequential()
# Adds a densely-connected layer with n OUTPUT units to the model
# in this first case, won't specify n. Only doing it at the last layer
# model.add(keras.layers.Dense(n, activation='relu'))
# Add a layer

### For signal use 2590 neurons in 1st layer

```

```

### For FFT use 10 neurons in 1st layer
### For DCT use 25 neurons in 1st layer
# model.add(keras.layers.BatchNormalization())
model.add(keras.layers.Dense(25, activation='tanh'))
# model.add(keras.layers.Dense(2000, activation='relu'))
# model.add(keras.layers.Dense(1500, activation='tanh'))
model.add(keras.layers.Dense(10, activation='relu'))
# model.add(keras.layers.Dense(200, activation='relu'))
# model.add(keras.layers.Dense(10, activation='relu'))
# model.add(keras.layers.Dense(5, activation='relu'))
# model.add(keras.layers.Dense(20, activation='relu'))
# Add another layer (last one)
model.add(keras.layers.Dense(4, activation='softmax'))

### Load data

# signal_list_01 = np.load('signal_list_99_out.npy')
# signal_list_02 = np.load('signal_list_99_POST_out.npy')

signal_list_01 = np.load('signal_list_01_out.npy')
signal_list_02 = np.load('signal_list_02_out.npy')
fft_list_01 = np.load('fft_list_01_out.npy')
fft_list_02 = np.load('fft_list_02_out.npy')
dct_list_01 = np.load('dct_list_01_out.npy')
dct_list_02 = np.load('dct_list_02_out.npy')
polyfit_list_01 = np.load('polyfit_list_01_out.npy')
polyfit_list_02 = np.load('polyfit_list_02_out.npy')

```

```

# Stack loaded data

signal_list = np.vstack((signal_list_01, signal_list_02))
fft_list = np.vstack((fft_list_01, fft_list_02))
dct_list = np.vstack((dct_list_01, dct_list_02))
polyfit_list = np.vstack((polyfit_list_01, polyfit_list_02))

### Two-class labeling

signal_labels = np.hstack((np.zeros((1,176)), np.ones((1,93)))) .T
fft_labels = np.hstack((np.zeros((1,238)), np.ones((1,125))) ) .T
dct_labels = np.hstack((np.zeros((1,238)), np.ones((1,125)))) .T
polyfit_labels = np.hstack((np.zeros((1,238)), np.ones((1,125)))) .T

# fft_labels=to_categorical(fft_labels)
# dct_labels=to_categorical(dct_labels)
# signal_labels=to_categorical(signal_labels)

### Four-class labeling

signal_labels = np.hstack((np.zeros((1,238)), 1*np.ones((1,16)), 2*np.ones((1,31)),
fft_labels = np.hstack((np.zeros((1,238)), 1*np.ones((1,16)), 2*np.ones((1,31)), 3*
dct_labels = np.hstack((np.zeros((1,238)), 1*np.ones((1,16)), 2*np.ones((1,31)), 3*
polyfit_labels = np.hstack((np.zeros((1,238)), 1*np.ones((1,16)), 2*np.ones((1,31))

# Shuffle Data

# ShuffleData(signal_list, signal_labels)
# signal_labels = to_categorical(signal_labels)
# ShuffleData(fft_list, fft_labels)

```

```

# fft_labels = to_categorical(fft_labels)
ShuffleData(dct_list, dct_labels)
dct_labels = to_categorical(dct_labels)
# ShuffleData(polyfit_list, polyfit_labels)
# polyfit_labels = to_categorical(polyfit_labels)

###Scale data
# scaler.fit(fft_list)
# polyfit_list = scaler.transform(fft_list)

scaler.fit(dct_list)
polyfit_list = scaler.transform(dct_list)

# scaler.fit(signal_list)
# polyfit_list = scaler.transform(signal_list)

# scaler.fit(polyfit_list)
# polyfit_list = scaler.transform(polyfit_list)

## Training setup
model.compile(optimizer=keras.optimizers.Adam(lr=0.01),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
# model.compile(optimizer=keras.optimizers.Adam(lr=0.01),
#               loss='mean_squared_error',
#               metrics=['accuracy'])

# Training data

```

```

# data = np.random.random((1000,32))
# labels = np.random.random((1000, 3))
# # validation data - to easily monitor the training
# val_data = np.random.random((100,32))
# val_labels = np.random.random((100,3))
# fit the data:

history = model.fit(dct_list, dct_labels, validation_split=0.4, epochs=200, verbose=1)

# history = model.fit(polyfit_list, polyfit_labels, validation_split=0.4, epochs=200, verbose=1)

# history = model.fit(fft_list, fft_labels, validation_split=0.4, epochs=300, verbose=1)
# history = model.fit(signal_list, signal_labels, validation_split=0.4, epochs=30, verbose=1)

import matplotlib.pyplot as plt
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'], '--')
plt.ylabel('Erro')
plt.xlabel('Épocas')
plt.legend(['Treino', 'Validação'])
plt.show()

### Evaluation and prediction

# test_error, test_acc = model.evaluate(signal_list, signal_labels)
test_error, test_acc = model.evaluate(dct_list, dct_labels)
# test_error, test_acc = model.evaluate(fft_list, fft_labels)

```

```

# test_error, test_acc = model.evaluate(polyfit_list, polyfit_labels)
# model.predict(fft_list, batch_size=32)
# test_error, test_acc = model.evaluate(signal_A0203, signal_A0203_labels)
print('Validation accuracy', test_acc)
# prediction = model.predict(signal_C02)

prediction = model.predict(fft_list)
np.set_printoptions(precision=4, suppress=True)
for i in range(len(prediction)): print(prediction[i])

### Save the trained model
keras.models.save_model(model, 'Net_03_FFT_STD_4class_3layers.h5') ###Save with th
# model.save_weights('PESO_REDE.h5')

### For reloading the model
# # Recreate the exact same model, including weights and optimizer.
# new_model = keras.models.load_model('my_model.h5')
# new_model.summary()

model = keras.models.load_model('Net_03_DCT_STD_4class_3layers.h5')
model.summary()

### Load data

signal_list_01 = np.load('signal_list_01_out.npy')
signal_list_02 = np.load('signal_list_02_out.npy')
fft_list_01 = np.load('fft_list_01_out.npy')
fft_list_02 = np.load('fft_list_02_out.npy')
dct_list_01 = np.load('dct_list_01_out.npy')
dct_list_02 = np.load('dct_list_02_out.npy')

```

```

polyfit_list_01 = np.load('polyfit_list_01_out.npy')
polyfit_list_02 = np.load('polyfit_list_02_out.npy')

signal_99 = np.load('signal_list_99_TEST.npy')
signal_99_post = np.load('signal_list_99_POST_TEST.npy')

# Stack loaded data
signal_list = np.vstack((signal_list_01, signal_list_02))
fft_list = np.vstack((fft_list_01, fft_list_02))
dct_list = np.vstack((dct_list_01, dct_list_02))
polyfit_list = np.vstack((polyfit_list_01, polyfit_list_02))

signal_99_list = np.vstack((signal_99, signal_99_post))

# prediction = model.predict(polyfit_list)
prediction = model.predict(dct_list)
# prediction = model.predict(signal_list)
# prediction = model.predict(fft_NR_stack)
# prediction = model.predict(signal_99_list)

# np.set_printoptions(precision=4, suppress=True)
# for i in range(len(prediction)): print(prediction[i])

from sklearn.metrics import confusion_matrix, classification_report
y_true = np.argmax(dct_labels, axis=1)
y_pred = np.argmax(prediction, axis=1)
labels_7class = ['P', 'A', 'B', 'C', 'A_10%', 'A_7.5%', 'C_5%']
labels_6class = ['P', 'A', 'B', 'C', 'A_white', 'C_white']

```

```

labels_2class = ['Original', 'Whitening']
# labels_4class = ['Original', '10%', '7,5%', '5%']
labels_old_4class = ['Original', '_5%', '_7,5%', '_10%']
# labels_old_4class = ['_2%', '_3,5%', '_5%', '_10%']
# labels_old_4class = ['Original', '2-3,5%', '_5%', '_10%']
z = confusion_matrix(y_true, y_pred)
z

import plotly.plotly
import plotly.figure_factory as ff
plotly.offline.init_notebook_mode()    #allows Plotly to work in Offline mode

#     Normalized Z
z = np.round(100*z.astype('float') / z.sum(axis=1)[:, np.newaxis],1)

colorscale = [[0, '#d8d8d8'], [1, '#000000']]

fig = ff.create_annotated_heatmap(z[::-1], x=labels_old_4class, y=labels_old_4class)
fig.layout.title = 'Matriz Confusão - DCT - 4 classes'
fig['layout']['xaxis'].update(side='bottom', title='Classe Verdadeira')
fig['layout']['yaxis'].update(title='Classe Prevista', tickangle = 0, tick0=0.5)
# fig['layout'].update(width = )
plotly.offline.iplot(fig, filename = 'Confusao_2_class')

```