Deep Learning aplicado a classificação de patologias da voz

Victor de Oliveira Guedes

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Prof. Dr. João Paulo Teixeira

Prof. Dr. Arnaldo Cândido Junior

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

Maio; 2019

Deep Learning aplicado a classificação de patologias da voz

Victor de Oliveira Guedes

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Sistemas de Informação.

Trabalho orientado por:

Prof. Dr. João Paulo Teixeira

Prof. Dr. Arnaldo Cândido Junior

Esta dissertação não inclui as críticas e sugestões feitas pelo Júri.

Bragança

Maio; 2019

Agradecimentos

Agradeço a minha família e amigos pela ajuda e por estarem sempre presentes no período em que morei em Portugal. Agradeço aos meus orientadores, Professor João Paulo Teixeira e Arnaldo Cândido Junior pelo conhecimento compartilhado e a inteira disponibilidade. Agradeço aos colegas de trabalho, juntos fizemos boas publicações. Agradeço ao Instituto Politécnico de Bragança e a Universidade Tecnológica Federal do Paraná por proporcionarem a oportunidade de realizar a dupla diplomação.

Obrigado.

Resumo

A classificação de patologias relacionadas a voz utilizando conceitos de *Deep Learning* vem crescendo consideravelmente nos últimos anos. Bons resultados já foram obtidos para a classificação em fala sustentada com vogais, mas ainda existem poucos trabalhos relacionadas a classificação deste problema utilizando fala contínua. Por isso, é foco desta dissertação realizar a implementação dos principais modelos de *Deep Learning* para a classificação de patologias da voz em fala contínua, utilizando a frase alemã "Guten Morgen, wie geht es Ihnen?" da base de dados *Saarbruecken Voice Database*. São utilizados as patologias de disfonia, laringite e paralisia das cordas vocais, além da classe dos saudáveis, para análises multi classe e binária. Além disso, também é realizado um estudo prévio para a classificação com vogais nas mesmas patologias. O melhor resultado para as vogais é de 99% de exatidão para a implementação de um modelo LSTM com parâmetros *Jitter*, *Shimmer* e Autocorrelação, na classificação binária entre laringite e saudável. Para as frases, é realizado um estudo comparativo entre modelos de redes neuronais, convolucionais e recorrentes para os parâmetros MFCCs e Espectrogramas na escala Mel obtendo resultados de 76% de medida-F para disfonia x saudável, 68% de medida-F para laringite x saudável, 80% de medida-F para paralisia x saudável. Para classificação multi classe é obtido 59% e 40% de medida-F para 3 classes e 4 classes, respectivamente.

Palavras-chave: *Long short-term memory*, Rede Neuronais Convolucional, Redes Neuronais Artificias, *Transfer Learning*.

Abstract

The classification of voice related pathologies using Deep Learning concepts has been increasing considerably in recent years. Good results have already been obtained for classification in sustained speech with vowels, but there are still few studies related to the classification of this problem using continuous speech. Therefore, the focus of this dissertation is to implement the main models of Deep Learning for the classification of voice pathologies in continuous speech, using the German phrase "Guten Morgen, wie geht es Ihnen?"From the Saarbruecken Voice Database. The pathologies of dysphonia, laryngitis and paralysis of the vocal cords, as well as the healthy class, are used for multi-class and binary analyzes. In addition, a previous study for the classification with vowels in the same pathologies is also carried out. The best result for the vowels is 99 % accuracy for the implementation of an LSTM model with parameters Jitter, Shimmer and Autocorrelation, in the binary classification between laryngitis and healthy. For the phrases, a comparative study between neural networks, convolutional and recurrent models with the parameter MFCCs and Spectrograms in the Mel scale, obtaining results of 76% F-measure for dysphonia x healthy, 68% F-measure for laryngitis x healthy, 80% F-measure for healthy x paralysis of the vocal cords. For multi-class classification is obtained 59% and 40% of F-measure for 3 classes and 4 classes, respectively.

Keywords: Long short-term memory, Convolutional Neural Networks, Artificial neural networks, Transfer Learning.

Conteúdo

Lista de Tabelas

Lista de Figuras

Siglas

API Application Programming Interface. 46

CNN *Convolutional Neural Network.* xv, 21, 35–37, 50–52, 54, 55

CNTK *The Microsoft Cognitive Toolkit.* 46

CPU *Central Process Unit.* 46

DCT *The Discrete Cosine Transform.* 17, 18

DNN *Deep Neural Network.* 6

EGG Eletroglotografia. 5

FEMH *FEMH Voice Data Challenge.* 7, 8, 46, 90, 93

GMM *Gaussian Mixture Model.* 4, 6

GPU *Graphics Processing Unit.* 46

LSTM *Long Short Term Memory.* xv, 2, 5, 7, 21, 32, 47, 48, 50–54, 60, 63–70, 74, 76, 78, 80, 81, 83, 87–89, 95

MEEI *Massachusetts Eye and Ear Infirmary.* 4, 6, 7, 46, 92

MFCCs *Mel Frequency Cepstral Coefficients.* xii, 2, 4, 6–8, 11, 17, 19, 45, 50, 51, 55–57, 62, 63, 67, 87, 88, 90, 92, 95

NFHE *Normalized First Harmonic Energy.* 8

ReLU *Rectified Linear Unit.* 28, 52, 54, 58

RNA Rede Neuronal Artificial. xv, 21, 47, 48, 50–53, 55, 63, 67–70, 73–76, 78, 87–89, 95

RNR Redes Neuronais Recorrentes. 30

ROC *Receiver operating characteristic.* 6

SHAC *Sequential Halving and Classification.* 8

STFT *Short-Time Fourier Transform.* 7, 15, 16

SVD *Saarbruecken Voice Database.* 5–7, 46, 47, 58

SVM *Support Vector Machine.* 4, 6–8

TNI *Turbulent Noise Index.* 8

Capítulo 1

Introdução

O Capítulo 1 é dedicado a uma introdução ao tema, seguido pelos objetivos do trabalho, as patologias da voz, a descrição de trabalhos relacionados (Estado da arte) e a estrutura desta dissertação.

1.1 Contextualização

Dentre os diversos meios de comunicação presentes, a fala é uma das mais utilizadas no cotidiano das pessoas. O processo de produção dos sons da fala é dado pelo aparelho fonador, que é composto pelos órgãos do sistema digestivo e respiratório.

O processo de produção dos sons da fala pode ser resumido como a passagem do ar (gerado pelos pulmões) pela laringe atingindo as pregas vocais (cordas vocais), que podem obstruir ou não este ar. Se existe a passagem do ar, há um movimento nas pregas vocais gerando um ligeiro afastamento, o que também pode ser denominado como vibrações das pregas vocais. Estas vibrações são responsáveis pela sonoridade produzida. Após a passagem pela laringe, o ar é bifurcado para a cavidade nasal e oral. Na cavidade oral há a presença da língua e dos lábios que em conjunto com os movimentos dos músculos da boca fazem a articulação na produção dos sons [1].

Entretanto, existem inúmeras doenças que afetam a laringe e que dificultam a realização da produção dos sons da fala, o que pode afetar o cotidiano dos indivíduos,

principalmente aqueles que trabalham com a voz como é o caso de professore, cantores, palestrantes, entre outros. Para diagnosticar essas doenças, o paciente deve-se submeter a exames que muitas das vezes podem causar desconforto.

Por esse motivo é que, com o advento da inteligência artificial, vêm sendo desenvolvidos inúmeros trabalhos relacionadas ao diagnostico de patologias da voz para auxiliar o médico na decisão final. Esta área de estudo está em constante crescimento.

Com isso, este trabalho tem como foco avaliar os principais modelos de *Deep Learning* na classificação das doenças patológicas, em específico as doenças disfonia, laringite crônica e paralisia das cordas vocais. São utilizadas como características da fala os parâmetros *Jitter*, *Shimmer*, Autocorrelação, MFCCs e espectrogramas na escala Mel.

1.2 Objetivos

O objetivo desta dissertação é realizar a implementação de modelos baseados em *Deep Learning* para a classificação de doenças patológicas através da voz utilizando fala contínua (frases), visto que na literatura é mais comum a utilização de vogais pronunciados de forma sustentada. Para isso, são definidos os seguintes objetivos específicos:

- Implementação de modelos baseados em Redes Neuronais Artificiais clássicas;
- Implementação de modelos baseados em Redes Recorrentes. Utilizando os conceitos da rede LSTM para a análise dos áudios ao longo do tempo;
- Implementação de modelos baseados em Redes Convolucionais;
- Aplicação dos modelos em problemas binários e multi classe para detecção de patologias na voz utilizando frases;
- Estudo comparativo das metodologias implementadas;
- Estudo comparativo com trabalhos relacionados.

1.3 Patologias da voz

Neste trabalho será abordada a classificação das seguintes doenças relacionadas com a voz: Disfonia, Laringite Crônica, Paralisia das Cordas Vocais.

1.3.1 Disfonia

Disfonia é caracterizado como um conjunto de distúrbios na produção vocal humana, ou seja, qualquer impedimento na produção normal da voz. Acontece bastante em pessoas que utilizam a voz com frequência e sem tomar as devidas precauções. Esta doença normalmente está associada a outros distúrbios, podendo ser diagnosticada como doença secundária ou primária (principal) [2].

As principais características da disfonia são normalmente associados à presença de dor na garganta, rouquidão, azia, dificuldade em manter a voz, cansaço ao falar, variações de frequência mais constantes, perda de volume, perda na projeção da voz, baixa resistência ao falar e gotejamento contínuo na parte de traz da garganta (catarro nasal) [2].

A disfonia pode ser caracterizada em duas classes principais, orgânica e funcional. A primeira está associado a alterações das pregas vocais (cordas vocais) através da presença de nódulos ou tumores. A segunda é assumida quando não é reconhecida a alteração anatômica da prega vocal [2].

1.3.2 Laringite crônica

A laringite crônica é associada a inflamação, agressão, irritação microbiana da mucosa laríngea [3], por outras palavras, é uma inflamação prolongada da laringe (localização das pregas vocais). A laringite crônica é uma evolução da laringite aguda. Quando a irritação da laringe persiste por mais do que duas ou três semanas, é recomendado a ida ao médico. A laringite crônica acontece principalmente em homens com idade entre 40 e 50 anos [3].

Os principais sintomas de uma pessoa com laringite aguda e crônica são irritações na garganta, voz rouca ou indetectável. A laringite aguda pode ser causada por infecções

por vírus e bactérias e uso excessivo da voz. A laringite crônica pode ser causada pela inalação de elementos químicos, pelos usos excessivo da voz, do álcool e do fumo [3].

1.3.3 Paralisia das cordas vocais

A paralisia das cordas vocais é definida como a perda de mobilidade dos músculos associados às pregas vocais. Esta perda na movimentação pode acontecer numa das cordas, denominada unilateral, ou nas duas cordas vocais, denominada bilateral. Esta doença pode ser causada por tumores, lesões, danos em nervos, infecções bacterianas, toxinas e complicações em processos cirúrgicos. Para o caso unilateral na maioria das vezes é causado por doenças neoplásticas (tumores). E para o caso bilateral é normalmente causado por problemas cirúrgicos [4].

Os principais sintomas para esta doença são a dificuldade de respirar, dificuldade de engolir, voz rouca, soprada, tosse fraca, [4].

1.4 Estado da arte

Nesta seção será descrita uma revisão da principal literatura publicada relacionada com o uso de inteligência artificial para o problema de classificação das patologias da voz, apresentando-as em ordem cronológica.

Na dissertação de doutoramento de Cordeiro [5], produzida em Outubro de 2016, são realizadas algumas abordagens para reconhecimento de patologias da voz. É proposto o reconhecimento através de fala contínua e através da vogal / a /. Nesta abordagem, são utilizadas três classes: saudáveis (36 pessoas), patologias laríngeas (edemas e nódulos com 59 pessoas) e patologias laríngeas neuromusculares (paralisia unilateral das pregas vocais 59 pessoas). Foram utilizados os coeficientes MFCCs extraídos da fala contínua da palavra "rainbow" e da vogal / a / da base de dados *Massachusetts Eye and Ear Infirmary* (MEEI) [6]. Os classificadores utilizados foram máquina de vetor suporte, do inglês *Support Vector Machine* (SVM), o modelo probabilístico de mistura gaussiana, do inglês *Gaussian Mixture Model* (GMM) e dois Discriminadores Lineares, avaliados com

validação cruzada. É concluído que a fala contínua obteve resultados superiores aos da vogal / a / resultando em 84% de exatidão (acurácia) em três classes.

Em relação ao trabalho de Harar, Alonso-Hernandezy, Mekyska et al. [7], publicado em Julho de 2017 é usada a base de dados alemã *Saarbruecken Voice Database* (SVD) [8] para fazer classificação entre pacientes saudáveis e patológicos. A base consiste em vozes coletadas de 2000 exemplos de sinais do formato Eletroglotografia (EGG) da vogal sustentada / a /, na qual 687 são de pessoas saudáveis e 1356 são patológicos divididos em 71 doenças. Harar, Alonso-Hernandezy, Mekyska et al. [7] fez a divisão de cada áudio em 64ms usando a janela de *hamming* com 30 ms de sobreposição, ao final é apresentada um vetor à rede com 3.200 amostras do sinal (0,064 s * 50 000 Hz = 3 200 amostras de sinal por cada segmento). Para fazer a classificação, foi dividida a base em 70% para treino, 15% validação e 15% teste, assegurando que a base fique balanceada com 480 pacientes saudáveis e 480 patológicos. O resto dos exemplos foi usado para teste. Harar, Alonso-Hernandezy, Mekyska et al. [7] diz ser o primeiro trabalho a apresentar um modelo de rede neuronal que utiliza uma combinação de redes convolucionais, LSTM e totalmente conectadas. Os resultados obtidos com esse modelos são 71,36% de exatidão, 65,04% de sensibilidade e 77,67% de especificidade em 206 dados de validação, e para o teste 68,08%, 66,75%, e 77,89% respectivamente em 874 áudios.

No trabalho de Teixeira, Fernandes e Alves [9], publicado em Novembro de 2017, é utilizado a base SVD para classificação binária entre saudável (41 mulheres e 29 homens) e patológico (41 mulheres e 29 homens) para grupos masculinos e femininos, separadamente. É analisada a doença Disfonia. Para isso, são extraídos 4 parâmetros *Jitter*, 4 parâmetros *Shimmer* e o *Harmonic to Noise Ratio* (HNR) de 3 vogais /a/,/i/,/u/ para três tons diferentes. O melhor modelo proposto é baseado na seleção de atributos e redução de dimensionalidade com Análise de regressão multi-linear e classificação com Rede Neuronal Artificial. É obtido 99% de exatidão para a classificação feminina e 90% de exatidão para a classificação masculina.

O artigo de Verde, De Pietro e Sannino [10], publicado em 15 de Março de 2018, utiliza da base de dados alemã SVD para fazer a classificação entre saudáveis e patológicos. Para

isso, é utilizado um total de 685 vozes patológicas contra 685 saudáveis, somente da vogal / a /. Como características, são utilizadas a frequência fundamental, *Jitter*, *Shimmer*, *Harmonic to Noise Ratio* (HNR), os MFCCs e a primeira e segunda derivada dos coeficientes mel-cepstrais ($\Delta MFCCs$ e $\Delta\Delta MFCCs$). Para fazer a classificação são realizados experimentos com os modelos SVM, *Decision Tree*, *Bayesian Classification*, *Logistic Model Tree*, *Istance-based Learning*. O melhor modelo foi o SVM, com a utilização de todos os parâmetros e 85,77% de exatidão, 87,59% de sensibilidade, 83,94% de especificidade e 85,80% de área *Receiver operating characteristic* (ROC), no português Característica de Operação do Receptor (COR).

O trabalho desenvolvido por Fang, Tsao, Hsiao et al. [11], publicado em 19 de Março de 2018, tem como proposta a classificação entre pacientes saudáveis e patológicos. Para isso foram extraídos os MFCCs de 3 segundos dos áudios da vogal sustentada / a / e a base utilizada no trabalho foi criada pelos autores com 60 exemplos de indivíduos saudáveis e 402 patológicos divididos em (Nódulos vocais, Pólipos e Cistos; Neoplasia glótica; Atrofia vocal; Distonia laríngea, Disfonia espasmódica e tremor vocal). Para avaliar o modelo treinado, foi utilizada a base de dados MEEI com 53 saudáveis e 173 patológicos. Para fazer a classificação foram utilizados os algoritmos *Deep Neural Network* (DNN), SVM, e GMM como modelo probabilístico, avaliados em validação cruzada com cinco *folds*, atingindo uma precisão na base masculina de 93,86% para MFCCs, 93,86% para MFCC+Delta ($\Delta MFCCs$) e 94,26% para MFCC(N)+Delta ($\Delta MFCCs$ normalizado) e na base feminina de 86,14%, 87,74% e 90,52% com os mesmos parâmetros anteriores respectivamente. Posteriormente atingindo 99.32% no modelo DNN.

O artigo publicado em Setembro de 2018 de Wu, Soraghan, Lowit et al. [12] também usa a base de dados alemã SVD para fazer classificação entre saudáveis e patológicos. Para isso são extraídos da base áudios da vogal sustentada / a / de 482 indivíduos de saudáveis e 482 patológicos que foram divididos em seis doenças (Laringite com 140 instancias, Leucoplasia com 41, Edema de Reinke com 68, Paralisia do nervo laríngeo recorrente com 213, Carcinoma de prega vocal com 22 e Pólipos de pregas vocais com 45). Para criar o modelo, a base é dividida em treinamento com 75% e teste com 25% das instâncias.

Os dados são convertidos para 25kHz de frequência de amostragem e transformados em domínio espectral com a Transformada de Fourier de Curto Termo (no inglês *Short-Time Fourier Transform* (STFT)). O autor apresenta um modelo baseado em Redes neuronais convolucionais desenvolvido em Python Tensorflow, usando o otimizador Adam, L2 para regularização e 100 épocas de treinamento. Os resultados obtidos são de 68% e 71% de exatidão na validação e no teste respectivamente e de 74% e 68% de sensibilidade e especificidade no teste.

O trabalho publicado em Outubro de 2018 de Teixeira, Fernandes, Guedes et al. [13] utiliza as doenças laringite crônica, paralisia das cordas vocais e disfonia para fazer uma classificação binária entre saudável e patológico. É utilizada neste trabalho a base alemã SVD para o conjunto de vogais /a/, /i/ e /u/ além da frase "Guten Morgen, wie geht es Ihnen?" ("Bom dia, Como esta você?"). Para a frase são utilizados os coeficientes MFCCs extraídos para uma janela com duração de 25ms e passo de 10ms, sendo, no final obtido um vetor de tamanho 1417, que é utilizado como entrada a um classificador SVM. O melhor resultado para a frase é de 69% de exatidão no teste.

No artigo de Gupta [14], publicado em 4 de Dezembro de 2018, é apresentada uma rede neuronal recorrente LSTM para classificação entre três doenças utilizando a base de dados do trabalho de [11] (*FEMH Voice Data Challenge* (FEMH) e MEEI). A motivação foi a competição "2018 FEMH Voice Data Challenge"[1]. Foram utilizadas 50 vozes normais e 150 exemplos de vozes patológicas, dividido em 60 de nódulos vocais, pólipos e cistos; 40 de neoplasma de glote e 50 de paralisia unilateral da voz para o treinamento e para o teste foi disponibilizado 400 exemplos. Como entrada da rede LSTM, o autor extraiu uma combinação de características utilizando MFCCs, *Spectral Centroid*, *Chroma* e *Spectral Contrast*, gerando 33 características para cada exemplo. Em seguida, no modelo existem duas camadas escondidas LSTM com 128 e 32 neurônios, respectivamente e uma camada de saída com 4 neurônios representando as classes. No final os resultados obtidos são divididos em duas fases, na primeira com sensibilidade e especificidade de 30% e 95,7%, respectivamente e na segunda com 22% e 97,1%, respectivamente. Quando

[1]Endereço para a competição: https://femh-challenge2018.weebly.com/

aplicados em média de Recall sem peso (unweighted average recall), que calcula a média não ponderada (considerando as classes igualmente importantes), os resultados são de 54% e 56%, respectivamente.

O artigo de Grzywalski, Maciaszek, Biniakowski et al. [15], publicado em 14 de Dezembro de 2018, também participou na competição "2018 FEMH Voice Data Challenge" para fazer a classificação entre as quatro classes com a vogal / a /. Entretanto, foi utilizada uma estratégia diferente do trabalho de [14], obtendo ao final da competição o segundo lugar. A estratégia utilizada foi a combinação de MFCC, e as características *Jitter* e início de ciclo representado por *Turbulent Noise Index* (TNI) e *Normalized First Harmonic Energy* (NFHE). O conceito de *Data augmentation* também foi utilizado, onde para cada áudio foi alterado a amplitude com uma multiplicação de um intervalo entre 0.4 e 1.2, uso de desvio de tonalidade aleatório e alongamento de tempo, utilizando uma rede neuronal com duas camadas escondidas. Ao final, os resultados obtidos para a competição foram de 89,4%, 66,0% e 71,20% em sensibilidade, especificidade e média de Recall sem peso (*unweighted average recall*), respectivamente.

O artigo publicado por Pishgar, Karim, Majumdar et al. [16] de 19 de Dezembro de 2018 também utilizou a base de dados da competição FEMH Voice Data Challenge de 2018. O objetivo também foi classificar entre as quatro classes patológicas disponíveis. Como características de áudio são utilizados os MFCCs e os respectivos Deltas. Ambos são concatenados. São então utilizados vários modelos de *Random Forests* para calcular as características mais importantes das 45 dimensões do vetor de entrada. Em seguida, é determinado um valor limiar para o algoritmo *Sequential Halving and Classification* (SHAC) que irá selecionar as características mais relevantes que estão acima deste valor. Por fim, é aplicado um classificador do tipo SVM. O resultado para as quatro classes apresenta um valor de 88,60% de Sensibilidade, 78,23% de Especificidade e 59% de *Recall* para os pesos de 40%, 20% e 40% respectivamente.

A publicação do primeiro lugar na competição FEMH não foi encontrada no levantamento deste estado da arte.

1.5 Estrutura do documento

Esta dissertação está estruturada da seguinte maneira.

No **Capítulo 2**, é apresentada a fundamentação teórica dos parâmetros do sinal de fala abordados na dissertação.

No **Capítulo 3**, é apresentada a fundamentação teórica das técnicas de aprendizagem computacional abordados na dissertação.

No **Capítulo 4**, são apresentadas as principais metologias desenvolvidas.

No **Capítulo 5**, são apresentados e discutidos os resultados obtidos das metologias desenvolvidas, bem como um estudo comparativo com os trabalhos relacionados.

No **Capítulo 6**, são apresentadas as conclusões da dissertação bem como os trabalhos futuros.

Capítulo 2

Parâmetros do Sinal de fala

Neste capítulo será descrito os principais parâmetros do sinal de fala que são utilizadas neste trabalho para o problema de patologias da voz, explicando inicialmente o *Jitter*, em seguida *Shimmer*, Autocorrelação, Espectrogramas, e por último os MFCCs.

2.1 *Jitter*

A característica de sinal *Jitter* tem como objetivo calcular se existe variação entre os períodos glotais das ondas de um sinal de fala sustentada, que também pode ser chamado de ciclos periódicos glotais. Essa variação acontece quando o paciente não consegue manter a frequência das vibrações das cordas vocais na produção de uma vogal de forma sustentada (constante). Pacientes com alguma doença costumam apresentar valores mais elevados [2] [17] [18].

Jitter pode ser medido de quatro formas diferentes, *Jitter* absoluto (jitta), *Jitter* relativo (local), *Jitter* Rap, *Jitter* ppq5.

Jitter absoluto (jitta) que representada a "variação da frequência fundamental de ciclo a ciclo ou a média absoluta da diferença entre períodos glotais" [2]. A Equação 2.1 mostra como calcular o *Jitter* absoluto, onde N é o número de períodos glotais e T.

$$jitta = \frac{1}{N-1} \sum_{i-1}^{N-1} |T_i - T_{i-1}| \tag{2.1}$$

Jitter relativo (local) representa a "média da diferença absoluta entre períodos glotais, dividido pelo período médio" [2]. A Equação 2.2 mostra o *Jitter* relativo.

$$jitter(local) = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} |T_i - T_{i-1}|}{\frac{1}{N} \sum_{i=1}^{N} T_i} * 100 \tag{2.2}$$

Jitter Rap representa a "pertubação média relativa à diferença absoluta média entre um período glotal e a media do mesmo e seus dois vizinhos, dividido pelo período glotal médio" [2]. A Equação 2.3 mostra o *Jitter* rap.

$$rap = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} \left| T_i - (\frac{1}{3} \sum_{n=i-1}^{i+1} T_n) \right|}{\frac{1}{N} \sum_{i=1}^{N} T_i} * 100 \tag{2.3}$$

Jitter (ppq5) é a representação do "quociente de pertubação de período glotal de cinco pontos, calculado como a média de ele e seus quatro vizinhos mais próximos, dividido pelo período médio" [2]. Pode se calculado com a Equação 2.4.

$$ppq5 = \frac{\frac{1}{N-1} \sum_{i=2}^{N-2} \left| T_i - (\frac{1}{5} \sum_{n=i-2}^{i+2} T_n) \right|}{\frac{1}{N} \sum_{i=1}^{N} T_i} * 100 \tag{2.4}$$

O *Jitter* foi obtido com o auxílio dos trabalhos de Fernandes, Teixeira, Fernandes et al. [19] e Teixeira e Gonçalves [17].

2.2 *Shimmer*

A característica de sinal *Shimmer* é a variação de uma quantidade e suposta e aparentemente constante e é obtida por cálculo da variação da amplitude entre os ciclos glotais do sinal de áudio, que também pode ser chamado de ciclos periódicos glotais. Essa variação acontece quando o paciente não consegue manter a elocução (volume de produção) de maneira constante. Pacientes com alguma doença costumam apresentar valores de *Shimmer* mais elevados [2] [17] [18].

Assim como *Jitter*, o parâmetro *Shimmer* também apresenta várias formas de ser medido, *Shimmer* (dB), *Shimmer* relativo (ShdB), *Shimmer* (apq3) *Shimmer* (apq5), são quatro as formas usadas frequentemente.

O *Shimmer* (dB) é expressa como a "variabilidade da amplitude de pico a pico em decibel, ou seja, a média absoluta da razão dos valores das amplitudes dos períodos consecutivos numa base logarítmica." [2]. A Equação 2.5 mostra o *Shimmer* (dB). Em que N é o número de períodos glotais, A a amplitude do período i.

$$ShdB = \frac{1}{N-1} \sum_{i=1}^{N-1} \left| 20 * log(\frac{A_{i+1}}{A_i}) \right| \tag{2.5}$$

O *Shimmer* relativo (Shim) é definido como "a diferença absoluta média entre as amplitudes de períodos consecutivos, dividido pela amplitude média, expressa em porcentagem" [2]. A Equação 2.6 mostra o caso.

$$Shim = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} |A_{i+1} - A_i|}{\frac{1}{N} \sum_{i=1}^{N} A_i} * 100 \tag{2.6}$$

O *Shimmer* (apq3) é o "quociente de perturbação de amplitude de três pontos, a diferença absoluta média entre a amplitude de um período e a média de amplitudes de seus vizinhos, dividida pela amplitude média. Expresso em porcentagem". A Equação 2.7 mostra o caso.

$$apq3 = \frac{\frac{1}{N-1} \sum_{i=2}^{N-1} \left| A_i - (\frac{1}{3} \sum_{n=i-1}^{i+1} A_n) \right|}{\frac{1}{N} \sum_{i=1}^{N} A_i} * 100 \tag{2.7}$$

O *Shimmer* (apq5) é definido como "o quociente de perturbação de amplitude de cinco pontos, a diferença absoluta média entre a amplitude de um período e a média das amplitudes de seus quatro vizinhos mais próximos, dividida pela amplitude média. Também é expressa em porcentagem". A Equação 2.8 mostra o caso.

$$apq5 = \frac{\frac{1}{N-1} \sum_{i=3}^{N-2} \left| A_i - (\frac{1}{5} \sum_{n=i-2}^{i+2} A_n) \right|}{\frac{1}{N} \sum_{i=1}^{N} A_i} * 100 \tag{2.8}$$

O *Shimmer* foi obtido com o auxílio dos trabalhos de Fernandes, Teixeira, Fernandes

et al. [19] e Teixeira e Gonçalves [17].

2.3 Autocorrelação

A autocorrelação tem como objetivo medir as partes semelhantes repetidas ao longo do sinal, ou seja, é feita uma comparação em partes ao longo de todo o sinal de áudio para encontrar suas respectivas semelhanças, assim quanto maior o número de semelhanças, maior é o valor da autocorrelação [20] [21] [22]. Matematicamente, a autocorrelação pode ser calculada em três passos.

No primeiro passo, demonstrado pela Equação 2.9, dado um sinal de áudio $x(t)$ é utilizado um segmento desse sinal com duração T e em determinado tempo t_{mid} para fazer a subtração da média μ_x do pedaço e multiplicar o resultado por uma janela $w(t)$, Assim é obtido uma janela do sinal $a(t)$ [22].

$$a(t) = (x(t_{mid} - \tfrac{1}{2}T + t) - \mu_x)w(t) \tag{2.9}$$

É recomendado por [22] a utilização de janela Senoidal dado pela Equação 2.10, ou a Janela de Hanning. Também se pode afirmar que a janela $w(t)$ é simétrica em $\frac{1}{2}T$ e zero e em todo o intervalo de tempo em $[0, T]$.

$$w(t) = \frac{1}{2} - \frac{1}{2}cos\frac{2\pi t}{T} \tag{2.10}$$

Com isso, é realizado o segundo passo que irá calcular a autocorrelação normalizada $r_a(\tau)$, onde τ é uma função de atraso. O cálculo é dado pela Equação 2.11.

$$r_a(\tau) = r_a(-\tau) = \frac{\int_0^{T-\tau} a(t)a(t+\tau)dt}{\int_0^T a^2(t)dt} \tag{2.11}$$

No terceiro passo é calculada a autocorrelação normalizada $r_w(\tau)$ janela anterior utilizando a agora a janela de Hanning. A Equação 2.12 mostra o cálculo.

$$r_w(\tau) = (1 - \frac{|\tau|}{T})(\frac{2}{3} + \frac{1}{3}cos\frac{2\pi\tau}{T}) + \frac{1}{2\pi}sin\frac{2\pi\,|\tau|}{T} \tag{2.12}$$

Por último é necessário calcular a autocorrelação $r_x(\tau)$ do segmento do áudio original. Para isso, é realizada a divisão de $r_a(\tau)$ por $r_w(\tau)$. A Figura 2.1 mostra um exemplo de todo o processo de extração autocorrelação.

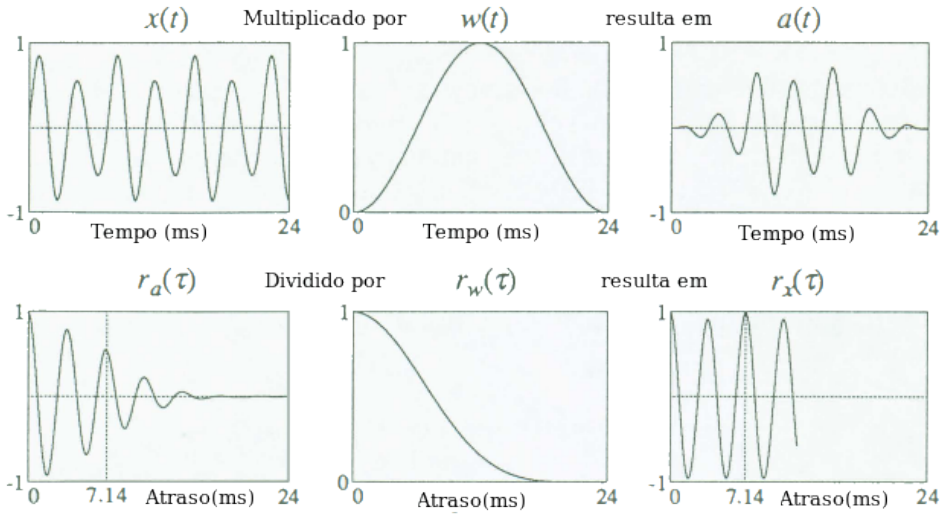

Figura 2.1: Exemplo calculo da Autocorrelação - adaptado e traduzido de [22]

2.4 Espectrograma

Outra forma de representar os sinais de áudio é através dos espectrogramas, estes que são matrizes tridimensionais representando a análise do sinal para a relação entre tempo, frequência e amplitude ao invés de tempo e amplitude originalmente [23]. Essa conversão de tempo-amplitude para tempo-frequência-amplitude pode ser dada pela Transformada de Fourier de curto tempo, do inglês *Short-time Fourier transform* (STFT) [24].

O STFT pode ser calculado em três etapas: na primeira são selecionados blocos do sinal (segmentos); a segunda multiplica-se este segmento por uma função de janela e então na terceira etapa é calculado a Transformada de Fourier de tempo discreto, do inglês *Discrete Time Fourier Transform* (DTFT). Os passos combinados formam a Equação 2.13, onde

$(x[n_0 - N], ..., x[n_0 + N - 1])^T$ é o segmento de tamanho 2N, $\omega[n]$ é a função de janela, ω é a frequência e n_0 o tempo [25].

$$X(\omega, n_0) = \sum_{n=-N}^{N-1} \omega[n]x[n + n_0]exp(-j\omega n) \tag{2.13}$$

Em resumo, o que acontece é que a Janela é deslocada no tempo analisando o respectivo segmento do sinal e os componentes do espectro são calculados como uma função de frequência [25]. Para ilustrar no formato de imagem, o STFT é então rotacionado em 90 graus e cada espectro do áudio é mapeado para um intervalo de [0,255] em três canais de cores (o mapeamento na imagem depende da implementação, pode ser para tons de cinza ou 3 canais). Ao final um sinal em domínio do tempo pode ser ilustrado em domínio da frequência como mostra a Figura 2.2.

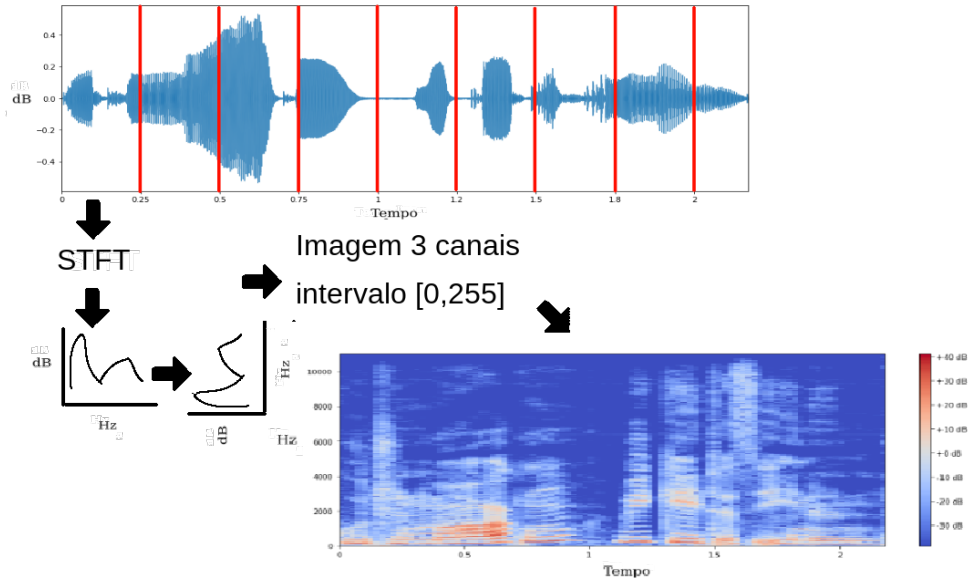

Figura 2.2: Exemplo Espectrograma

O espectrograma também pode ser convertido para espectrograma na escala Mel. As frequências são transformadas de uma escala linear para a escala Mel, utilizando a Equação 2.17.

2.5 Coeficientes Mel Cepstrais - MFCCs

Os Coeficientes Mel Cepstrais, do inglês *Mel Frequency Cepstral Coefficients*, são características muito utilizadas no processamento de sinal [26], frequentemente usado para o reconhecimento de fala e reconhecimento de orador [27]. Sua popularidade e importância é dada pelo potencial de representar os espectros de amplitudes do sinal de fala de uma forma compactada, com boas representações em casos de sinais sem a presença de ruídos [27], [28]. Também pode-se considerar o fato de que os MFCCs são baseados no sistema auditivo humano, representados na escala Mel [29].

O processo para a extração dos MFCCs pode ser dividido em sete passos: Pré-enfase, *Framing*, Janelamento, Transformada de Fourier discreta (DFT), Banco de filtros Mel, Transformada discreta do cosseno (*The Discrete Cosine Transform* (DCT)) e o calculo da energia [29], [30].

O primeiro passo (Pré-enfase) consiste em enfatizar as frequências mais altas, aumentando a energia no sinal nessas frequências. A Equação 2.14 mostra o cálculo para a Pré enfase, onde $x[n]$ é o sinal de áudio [30].

$$y[n] = x[n] - 0,95x[n-1] \tag{2.14}$$

O segundo passo denominado de *Framing* tem como objetivo fazer a segmentação do sinal de áudio em pequenos pedaços (*frame*), onde é recomendado que cada pedaço possa ter um tamanho dentro de um intervalo de 20 a 40 milissegundos [30].

O terceiro passo denominado de Janelamento tem como objetivo fazer a multiplicação de uma Função de Janela para cada *frame* do sinal de áudio. É recomendado usar a janela de *Hamming* para este processo [29], [30]. A Equação 2.15 mostra o janelamento.

$$y(n) = x(n)(0,54 - 0,46cos(\frac{2\pi n}{N-1})) \tag{2.15}$$

O quarto passo consiste em converter as N amostras de cada *frame* do domínio do tempo para o domínio da frequência, que também pode ser chamados de coeficientes

espectrais. É utilizada a Transformada discreta de Fourier (DFT) para tal operação [29], [30], e pode ser calculado com a Equação 2.16, onde $X(k)$ são os coeficientes espectrais, $x(n)$ o *frame* do sinal de áudio, e os valores de n, k devem ser maior ou igual a zero e menor ou igual a N-1.

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{i2\pi nk}{N}} \tag{2.16}$$

O quinto passo diz respeito à transformação para a escala Mel. A percepção humana das Frequências dos sinais de áudio não segue uma escala linear, ou seja, para cada tom representado em Hz deve-se representar este mesmo tom para a escala Mel. Esta escala é uma escala de frequência linear abaixo dos 1000Hz e com um espaçamento logarítmico acima de 1000Hz. Como exemplo, um tons de 1KHz (40 decibéis) que está na percepção humana, tem o valor de 1000 mels [29]. Para fazer tal cálculo é utilizada a Equação 2.17.

$$Mel(f) = 2595 * log_{10}(\frac{f}{700} + 1) \tag{2.17}$$

São aplicados filtros triangulares no espectro do sinal para fazer tal conversão. A Figural 2.3 mostra o banco de filtros.

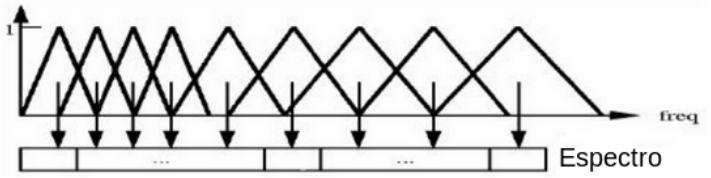

Figura 2.3: Escala Mel com filtro triangular - traduzido de [31]

O sexto passo é a Transformada discreta do cosseno (DCT) que consiste no processo de transformar o espectro Mel para o domínio do tempo. O resultado dessa transformação pode ser denominado de *Mel Frequency Cepstrum Coefficient*, onde os coeficientes de ordem superior representam a periodicidade na forma de onda, e os coeficientes de ordem mais baixa representam a forma do trato vocal. Em casos de reconhecimento de voz, somente os coeficientes de ordem mais baixas são utilizados, ordens menores do que vinte

[29], [30].

O sétimo e último passo é calcular a energia do *frame* de um sinal para um segmento no tempo $t1$ para o tempo $t2$ utilizando a Equação 2.18.

$$Energia = \sum x^2[t] \tag{2.18}$$

Com isso são obtidos os MFCCs por completo podendo ser utilizados para fazer as respectivas análises de voz.

Além dos parâmetros usados a outros reconhecidamente importantes como é o caso do *Harmonic to Noise Ratio* (HNR) que consiste na relação das componentes harmônicas e as componentes de ruído e é utilizado por alguns autores na detecção de patologias da voz Teixeira, Fernandes e Alves Verde, De Pietro e Sannino. Contudo, neste trabalho não foi usado pois já se estar a usar a autocorrelação que é um parâmetro muito relacionado ao HNR.

Capítulo 3

Aprendizagem Computacional

Neste capítulo será apresentada a fundamentação teórica referente às ferramentas de inteligência artificial usadas neste trabalho. Primeiramente serão introduzidos conceitos de Redes Neuronais artificiais, em seguida será abordado os modelos de Redes Neuronais Recorrentes com LSTM, Redes Neuronais Convolucionais, do inglês CNN, os conceitos de *Transfer learning* e as métricas de avaliação utilizadas.

3.1 Redes neuronais artificiais

A RNA foi desenvolvida baseada na forma como o cérebro humano processa a informação, diferente do computador, o cérebro tem a facilidade de aprendizagem e paralelismo muito mais eficiente. O cérebro é constituído por neurônios e é com esses neurônios em conjunto que o processo de comunicação de informação acontece. O neurônio é formado por núcleo, soma (corpo celular), dendritos e axônio. Na extremidade do prolongamento do axônio existe o terminal axonal, e é nessa região que existe a ligação com o soma ou com os dendritos de outro neurônio para transmitir a informação. Dendritos são prolongamentos que se estendem pelo corpo celular e recebem sinais de outros neurônios. A sinapse é o ponto de contato entre dois neurônios, e é composta por um lado pré-sináptico que é onde o sinal está sendo enviado, e um lado pós-sináptico que é onde o sinal está sendo recebido. A transmissão sináptica é a transferência de informação de um neurônio para

outro. Quando um impulso elétrico ou químico chega ao final do axônio, são liberadas substâncias químicas que são detectadas pelos dendritos ou corpo celular que se encontra do outro lado da sinapse. A partir dessa detecção são gerados nos dendritos impulsos elétricos para transmissão [32], a Figura 3.1 ilustra o caso.

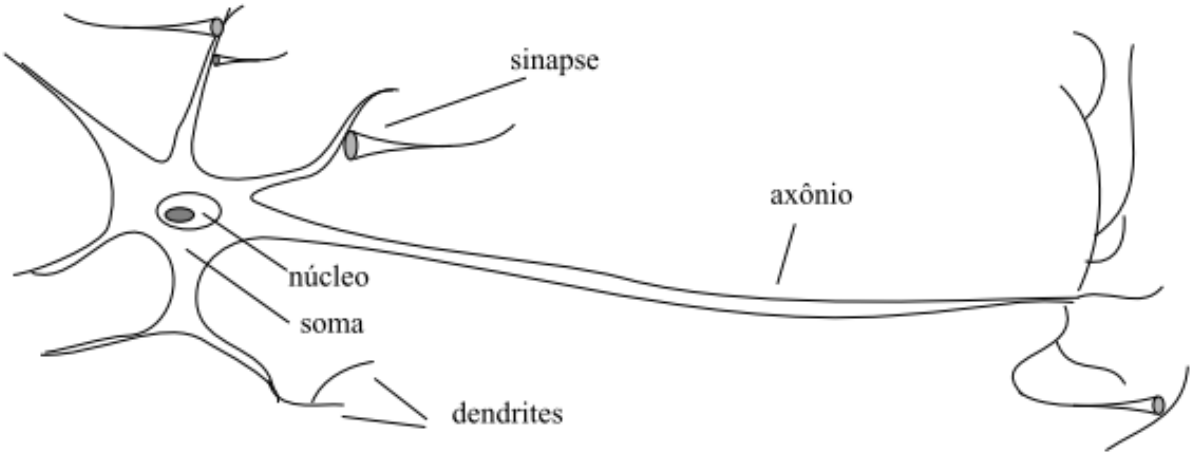

Figura 3.1: Neurônio [33]

A Rede Neuronal Artificial é constituída por conexões entre nós, os nós representam neurônios artificiais. Esses nós são organizados em camadas de entrada de dados, camadas intermediárias ou ocultas e camadas de saída, onde as conexões simulam uma sinapse de um neurônio biológico. Cada conexão apresenta um peso associado a um nó, e é com este peso que as operações matemáticas são realizadas para futuras conexões [32].

É importante apresentar os elementos da rede neuronal em notações, estas podem divergir dependendo do autor. Nesta seção será usada por convenção a notação abaixo:

- Atributo (dado de entrada): x_i, valores utilizados na entrada da Rede Neuronal Artificial;
- Peso: w_i, onde a informação é armazenada. Pode ter valores positivos ou negativos e é com o peso que é feita a ponderação com os valores de entrada;
- Liminar de Ativação: θ, pode ser chamado de *Bias* ou Viés. tem a função de aumentar ou diminuir o valor de entrada, limitando-o dentro do intervalo estabelecido

pela função de ativação;

- Sinal de saída: y_i, valor obtido na camada de saída após a passagem pela Rede;
- Potencial de Ativação: v, resultado obtido entre a ponderação dos pesos com os valores de entrada e a Função de ativação;
- Função de Ativação: φ, funcionalidade de limitação da saída do neurônio em um intervalo;
- Saída desejada: d_i, valor desejado no final do treinamento;
- Taxa de aprendizagem: n, valor que é constante dentro do intervalo da função de ativação para realizar o gradiente descendente.

Para determinadas entradas, é dado um peso associado a x_i e é realizada uma soma ponderada das entradas com os respectivos pesos. Este é o estimulo inicial de uma rede neuronal [34]. A Equação 3.1 apresenta o caso.

$$v = \sum_{i=1}^{n} w_i x_i \tag{3.1}$$

Na saída da rede neuronal é apresentado o sinal y_i que é o valor gerado pela Rede Neuronal (3.2). É atribuída uma função de ativação para restringir o valor final, normalmente o valor atribuído é um intervalo fechado de amplitude [0,1] ou [-1,1]. Também é apresentado um liminar de ativação (*bias*) que tem o efeito de aumentar ou diminuir a entrada liquida da função de ativação.

$$y_i = \varphi(v + \theta) \tag{3.2}$$

3.1.1 Perceptrão

O Perceptrão é uma rede neuronal simples para classificar padrões linearmente separáveis. Ele é construído levando em conta um neurônio não linear [34]. A Figura 3.2 representa uma rede Perceptrão com apenas um neurônio.

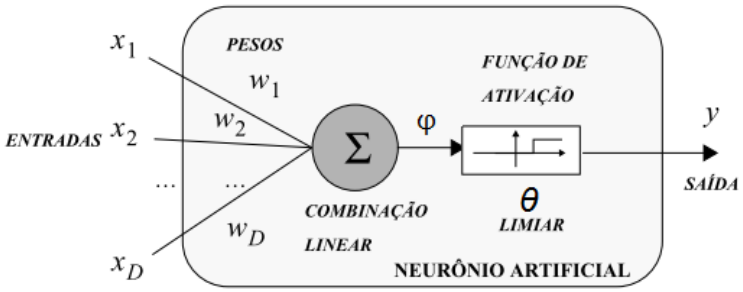

Figura 3.2: Modelo Neurônio McCulloch e Pitts [33]

A função de ativação representada para este modelo define a saída de um neurônio em termos da função de degrau. Representada pela equação 3.3.

$$y_i = \begin{cases} 1 \; se \; v \geq 0 \\ 0 \; se \; v < 0 \end{cases} \tag{3.3}$$

Para pesquisas atuais não é tão interessante, pois a dificuldade na aplicação do gradiente descendente não é facilitada, diminuindo o nível de aprendizagem do modelo. O interessante deste modelo é a possibilidade de representar funções primitivas AND, OR, NAND e NOR [35]. Infelizmente o XOR não é representado pelo Perceptrão, pois só pode lidar-se com padrões linearmente separáveis, a Figura 3.3 (b), ilustra o caso.

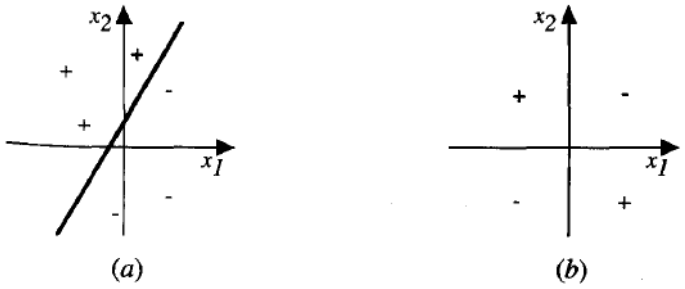

Figura 3.3: Superfície de decisão Perceptrão[35]

O treinamento através do Perceptrão é supervisionado por correção de erro. Um caminho de aprendizagem aceitável é começar com valores de pesos aleatórios e iterar para cada treino, ajustando o erro e modificando os pesos, caso o valor obtido seja diferente do

desejado. As Equações 3.4 e 3.5 representam este método.

$$w_i = w_i + \Delta w_i \tag{3.4}$$

$$\begin{aligned} \Delta w_i &= n w_i (d - y)(d \neq y) \\ \Delta w_i &= 0(d = 0y) \end{aligned} \tag{3.5}$$

3.1.2 *Multi-Layer Perceptron* - MLP

Diferentemente do Perceptrão, o *Multi-Layer Perceptron* é uma arquitetura mais utilizada no estudo de redes neuronais. Esse modelo normalmente utiliza até duas camadas escondidas. Os neurônios são conectados entre camadas vizinhas, sendo o treino realizado pelo algoritmo *Backpropagation*, dividido em duas fases (*forward*, *backward*). Como é visto na Figura 3.4, a informação é transmitida pela rede neuronal no sentido da esquerda para a direita, entretanto neste modelo existem dois sinais, sinais funcionais e sinais de erro [34]. O primeiro é um sinal de estímulo da rede. Ele começa na camada de entrada e é propagado até a camada de saída, tem esse nome pois acredita-se que no final a informação será útil. O Segundo é propagado da camada de saída até a entrada. Apresenta esse nome pois depende do cálculo de erro em cada nó percorrido. Este modelo de Rede Neuronal é denominado de *Feed-Forward*.

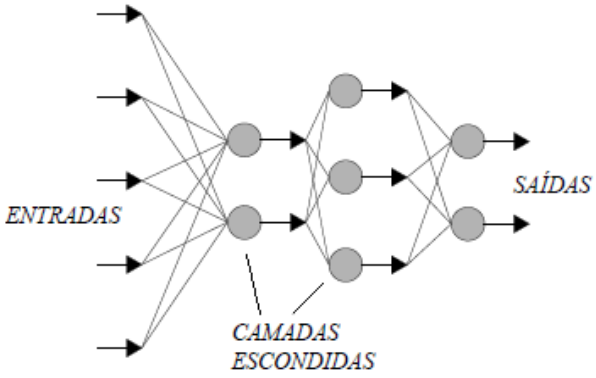

Figura 3.4: Modelo *Multi-Layer Perceptron* Feed-Forward. Adaptado de [33]

O problema apresentado pelo Perceptrão de não conseguir representar a função primitiva XOR é sanado com o Muti-Layer Perceptron. Na verdade, todas as funções primitivas AND, OR, NAND, NOR e XOR são representadas, pois agora é possível dividir o hiperplano em dois ou mais seguimentos [32]. Outro diferencial é que a função de ativação muda, neste caso é dada a possibilidade de escolha entre a função Sigmóide Logística ou a Tangente Hiperbólica.

A Sigmóide Logística aplica uma transformação com valores variando entre 0 a 1 [35]. Ela é utilizada pois a sua derivada é facilmente expressa em termos de produção, facilitando a aplicação do gradiente descendente. A Equação 3.6 representa a função Sigmóide.

$$\varphi(v) = \frac{1}{1 + e^{-ax}} \tag{3.6}$$

Graficamente a Sigmóide é representada pela Figura 3.5:

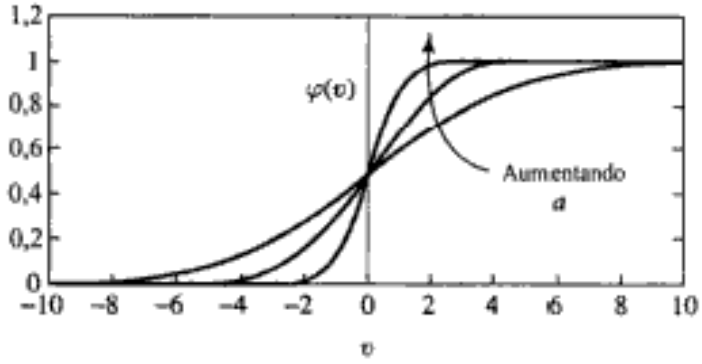

Figura 3.5: Gráfico Função Sigmóide [34]

Algumas vezes é desejado que a função de ativação varie entre -1 a +1, assumindo formas antissimétricas em relação à origem [34]. Isto é feito pela função Tangente Hiperbólica, representada pela Equação 3.7

$$\varphi(v) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3.7}$$

O treino do *Multi-Layer Perceptron* é realizado através do algoritmo *Backpropagation*.

O algoritmo aplica a descida de gradiente para tentar minimizar os erros quadráticos entre os valores de saída da rede e os valores desejados de saída e também é possível ajustar os pesos nas camadas intermediárias da rede neuronal [35].

A fase *forward* é normalmente iniciada na camada de entrada em sentido à camada de saída. Segue o mesmo princípio da Equação 3.2 para fazer a ativação dos nós, entretanto agora o processo é feito em todos os nós (Somatório) e a função de ativação pode ser a Sigmóide Logística ou a Tangente Hiperbólica.

A fase *backward* é o diferencial do algoritmo Back *Multi-Layer Perceptron*, e é nela que é calculado o erro sobre os valores de saída de cada nó. O cálculo do erro pode ser aplicado tanto na camada de saída, quanto nas camadas escondidas [34].

Na camada de saída é importante visualizar quais são os elementos que influenciam diretamente para o valor obtido naquele nó. Valor obtido e o valor desejado são entrada para a Equação 3.8, considerando que a função de ativação da rede é a Sigmóide Logística e que a função de custo são os Erros Quadráticos Médios.

$$\delta_j = (d_i - y_i)(y_i)(1 - y_i) \tag{3.8}$$

Com o resultado obtido da Equação 3.8 tem-se o erro do neurônio de saída. Ele permite ajustar o erro para os pesos que estão influenciando o nó de saída. Para isso utiliza-se o peso a ser ajustado, a taxa de aprendizagem, o valor obtido do nó anterior, o valor obtido do nó da última camada e os valores desejado e obtido. A Equação 3.9 representa o caso.

$$w_i = w_i + nh_i y_i (1 - y_i)\delta_j \tag{3.9}$$

Nas camadas intermediárias, o processo é um pouco diferente. Para ajustar os pesos dessas camadas é importante visualizar os pesos de entrada do nó e seus pesos de saída. Os pesos de saída são então relacionados com os erros dos neurônios da camada seguinte, produzindo assim um somatório de pesos a calcular que corresponde ao peso do neurônio sendo analisado. Com o resultado da Equação 3.10, basta aplicar na Equação 3.11 e o peso será ajustado.

$$\delta_k = \sum_{i=0}^{n}(w_i \delta_j) \tag{3.10}$$

$$w_i = w_i + n w_i h_i (1 - h_i) \delta_k \tag{3.11}$$

As equações minimizam a função de custo Erros Quadráticos Médios, pela pela Equação 3.12.

$$c(n) = \frac{\sum_{i=0}^{n}(y_i - d_i)^2}{2} \tag{3.12}$$

3.1.3 *Deep Learning*

Redes Neuronais Profundas ou *Deep Neural Network* (DNN) são geralmente modificações do *Mutilayer Perceptron* convencional, nas quais é possível a adição de várias camadas escondidas [36], [37]. Com isso, uma rede consegue usar um número menor de neurônios que uma rede rasa, o que permite sua utilização em base de dados de treinamento de grande porte (Big data). Também usam versões mais eficientes das funções de ativação para tornar o aprendizado da rede mais rápido.

Uma dessas funções de ativação é a função *Rectified Linear Unit* (ReLU) [37]. representada por $g(z) = max(0, z)$ onde z é o valor de entrada do neurônio, esta função tem por objetivo garantir erros menores para o treino da rede, pois o problema da desvanecimento do gradiente descendente (os gradientes da função de perda se aproximam de zero, tornando a rede difícil de treinar) [38] não é tão presente quanto comparado às função Sigmóide Logística e a Tangente Hiperbólica [39]. Além disso, diminui o tempo que o modelo deveria convergir. Graficamente é expressa pela Figura 3.6.

Para problemas de classificação, também é introduzida a função de custo *Softmax*, representada pela Equação 3.13. Onde z é o vetor das entradas para a camada de saída (10 classes, 10 elementos em z). Esta é frequentemente utilizada no final da rede e tem como objetivo fazer com que o resultado dos neurônios de saída possam ser interpretados como probabilidades. Além disso, também destaca o maior valor de entrada [39],[37].

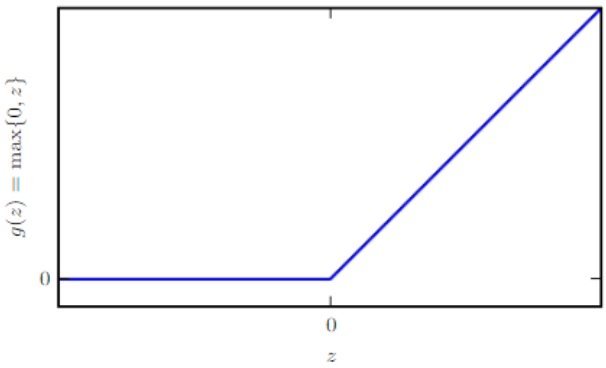

Figura 3.6: Gráfico da função ReLU [37]

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{j=1}^{k} e^{z_j}} \tag{3.13}$$

A seguir é apresentado o conceito de *Dropout* [37]. Esta técnica é normalmente usada para evitar *overfitting*, e tem como característica a remoção de neurônios (se aplicada à camada) da rede base. A Figura 3.7 mostra um exemplo de uma rede totalmente conectada com aplicação de *Dropout*.

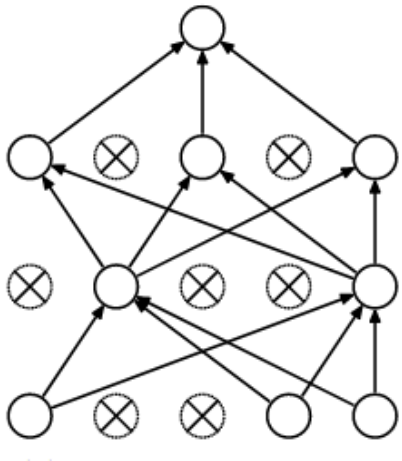

Figura 3.7: Exemplo de aplicação de *Dropout* [40]

3.2 Redes neuronais recorrentes

Outra topologia bastante utilizada em redes neuronais, são as Redes Neuronais Recorrentes (RNR). Diferentemente da Feed-Foward, com a RNR existe a possibilidade de o neurônio se auto alimentar, utilizando os próprios sinais obtido para se ajustar até ativar o liminar de ativação [39]. Um modelo mais utilizado neste contexto é a rede onde o neurônio de saída é conectado ao neurônio de entrada, criando ciclos e novas iterações com os valores finais obtidos [35]. A figura 3.8 ilustra o caso.

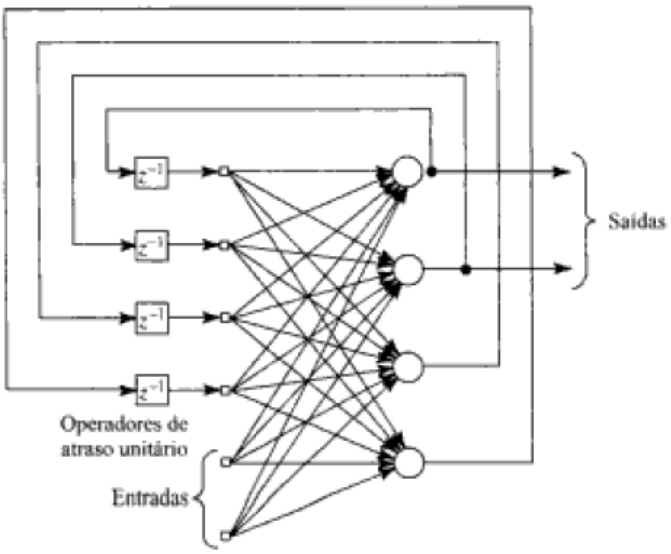

Figura 3.8: Rede Neuronal Recorrente com realimentação pela saída [34]

Esse tipo de rede é bastante utilizado para processamento de informações sequencias e temporais como textos, traduções, áudios e vídeos [39]. Na verdade, o principal objetivo é fazer com que a rede crie uma espécie de memória, fazendo com que quando um determinado dado (utilizado anteriormente) seja reconhecido com mais facilidade.

O uso de tempo nesta rede é importante, dados de um determinado tempo t podem ser utilizados como entradas para um posterior tempo $t+1$ [35]. Valores estes que podem ser modificados, alterando o comportamento da rede [39].

A Rede Neuronal Recorrente utiliza os mesmos conceitos de uma rede Feed-Forward

no que diz respeito as fases *Forward* e *Backward*. Entretanto agora para o treinamento é adicionado a variável de estado para o algoritmo de *Backpropagation*, criando o chamado *Backpropagation Through Time* (*Backpropagation* através do tempo) [39].

Antes de apresentar o algoritmo, é importante apresentar novos símbolos para a convenção [41].

- $x(t)$: dado de entrada em um determinado tempo t;
- $s(t-1)$: estado do neurônio oculto em um determinado tempo anterior;
- $s(t)$: estado do neurônio em um determinado tempo;
- $y(t)$: neurônio de saída num determinado tempo t;
- u: peso da conexão de realimentação.

Na fase *Foward* a propagação de valores entre as camadas também utiliza pesos, entretanto agora é importante incluir os pesos que fazem ciclos de propagação para o neurônio analisado. A Equação 3.14 define o caso para a saída do neurônio e esta é dependente da Equação 3.15, entretanto caso não exista um neurônio recorrente pode-se utilizar a Equação 3.15 sem o segundo termo mais o θ [42]. A Figura 3.9 representa a utilização como utilizar as Equações

$$y_i(t) = \varphi(v_i) \tag{3.14}$$

$$v_i(t) = \sum_i w_i x_i(t) + \sum_j u_j y_j(t-1) + \theta \tag{3.15}$$

3.2.1 *Long Short Term Memory*

A vantagem do uso da Rede Neuronal Recorrente é que ela permite a possibilidade de mapear informações em sequências de entrada e saída, infelizmente devido ao fato da desvanecimento do gradiente [38] a informação não é mantida por um longo período de

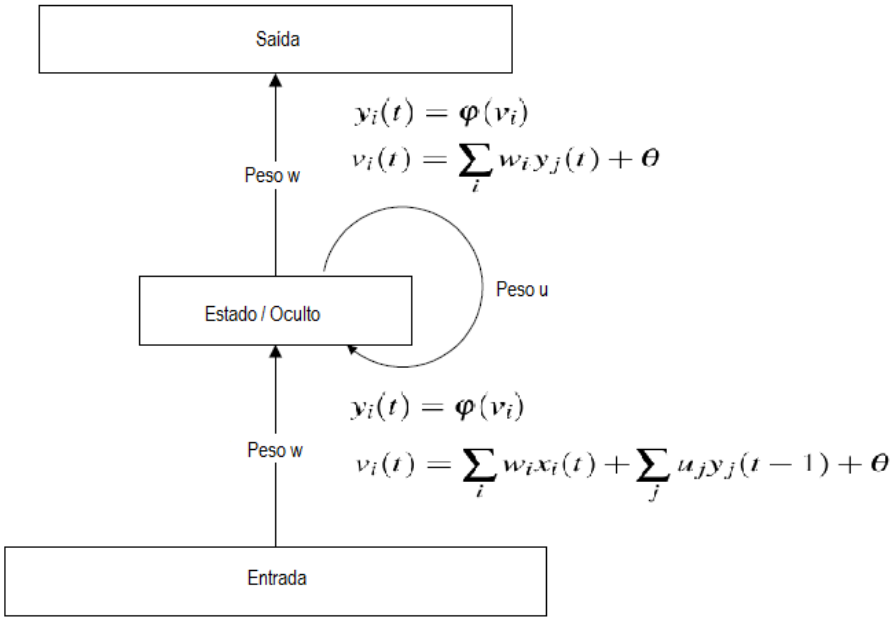

Figura 3.9: Rede Neuronal Recorrente com realimentação - adaptado de [42]

tempo. Isso acontece, pois conexões com dependências para longo prazo são difíceis de ser mantidas e aprendidas por causa da falta de troca de pesos, fazendo com que o sinal ou decaia ou cresça rapidamente dentro desse ciclo, a Figura 3.10 ilustra o caso.

Eis que surgem o *Long Short Term Memory* (LSTM) como alternativa. Este apresenta uma arquitetura que é constituído por um conjunto de sub-redes conectadas recorrentemente, que é conhecido como blocos de memória. Dentro do bloco existe a célula de memória. Também existem três unidades, conhecidas como porta de entrada, de saída e o de esquecimento com as respectivas funções de escrita, leitura e *reset* da célula [43]. As entradas das portas estão associadas com o vetor de entrada atual, saída do neurônio (bloco) anterior e o seu respectivo *bias* (viés).

Portas de uma célula de memória LSTM permitem o armazenamento e o acesso de informação por um longo período de tempo, resolvendo assim o problema da desvanecimento do gradiente. Se o potencial de ativação não for suficiente para ativar a porta de entrada, a informação na célula não será sobrescrita por novas entradas que a rede oferecer e poderá ser utilizada a longo prazo para futuras saídas [43]. A Figura 3.11 mostra um

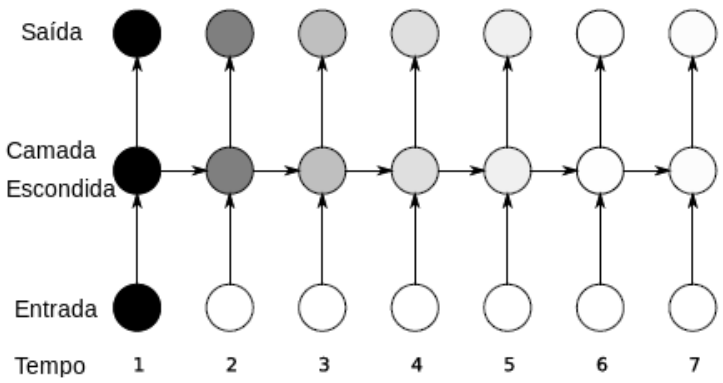

Figura 3.10: Dissipação (desvanecimento) do Gradiente - traduzido de [43]

bloco de memória.

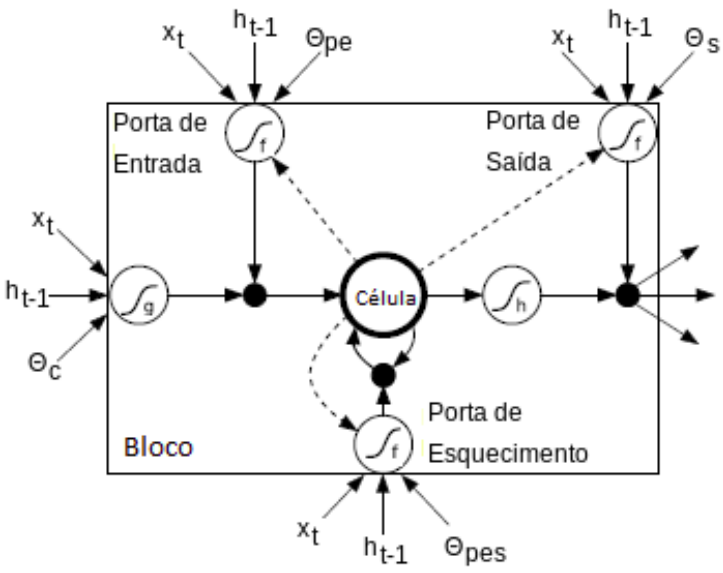

Figura 3.11: Bloco de Memória com uma célula [43]

A porta de entrada é responsável por atribuir informação na célula de memória, e é calculado o quanto do valor dado deve ser adicionado. É expressa pela Equação 3.16, onde x_t é o valor de entrada no tempo t, h_{t-1} é a saída do neurônio oculto anterior, c_{t-1} é o valor anterior da ativação da célula e φ é a função de ativação. Nesta porta normalmente aplica-se a Sigmóide Logística [36].

$$pe_t = \varphi(w^{xpe}x_t + w^{hpe}h_{t-1} + w^{cpe}c_{t-1} + \theta^{pe}) \tag{3.16}$$

A porta de esquecimento mantém ou esquece a informação da célula. De acordo com o valor dado é calculado quanta informação será lembrada e normalmente é aplicada a função Sigmóide Logística como função de ativação [36]. É expressa pela Equação 3.17.

$$pes_t = \varphi(w^{xpes}x_t + w^{hpes}h_{t-1} + w^{cpes}c_{t-1} + \theta^{pes}) \tag{3.17}$$

A célula de memória também apresenta um cálculo para a sua ativação, e esta é expressa pela Equação 3.18, onde $*$ são os pontos pretos representados na Figura 3.11, e a função de ativação normalmente usada é Tangente Hiperbólica [43].

$$c_t = pes_t * c_{t-1} + pe_t * \varphi(w^{xc}x_t + w^{hc}h_{t-1} + \theta^{c}) \tag{3.18}$$

A porta de saída lê a informação da célula de memória e envia de volta à Rede Recorrente. De acordo com o valor dado é armazenado o quanto de saída será desejado. A entrada no porta utiliza-se a Equação 3.19, e para efetivar a saída é utilizado a Equação 3.20. À função de ativação também associada a Sigmóide Logística [43]

$$s_t = \varphi(w^{xs}x_t + w^{hs}h_{t-1} + w^{cs}c_t + \theta^{s}) \tag{3.19}$$

$$h_t = s_t * \varphi(c_t) \tag{3.20}$$

O dado de estrada é diretamente conectado com todos as portas, assim cada porta recebe o mesmo valor.

3.3 Redes neuronais convolucionais

Redes neuronais convolucionais, do inglês, *Convolutional Neural Network* CNN são redes inspiradas na visão humana e que são frequentemente utilizadas para problemas de classificação com a utilização de imagens [44],[45]. Essa rede é basicamente composta por convoluções, *Kernel*, camadas de *pooling* e camadas totalmente conectadas.

As camadas convolucionais (convoluções) são operações entre imagens e o *Kernel* (quadrado deslizando sobre a imagem), onde é propagado na imagem desejada um *Kernel* de tamanho $N * N$, resultando em um filtro (ou *Feature Map*) e extraindo características que serão propagadas para as camadas seguintes da rede. O *Kernel* pode ser deslocado pela imagem sobrepondo a informação ou não (*overlap*). Uma camada convolucional é composta por vários filtros.

Um exemplo para uma imagem 5x5 por um *Kernel* 3x3 como pode ser visto com a Figura 3.12. É calculado o somatório entre o produto da imagem e o deslizamento do *Kernel*.

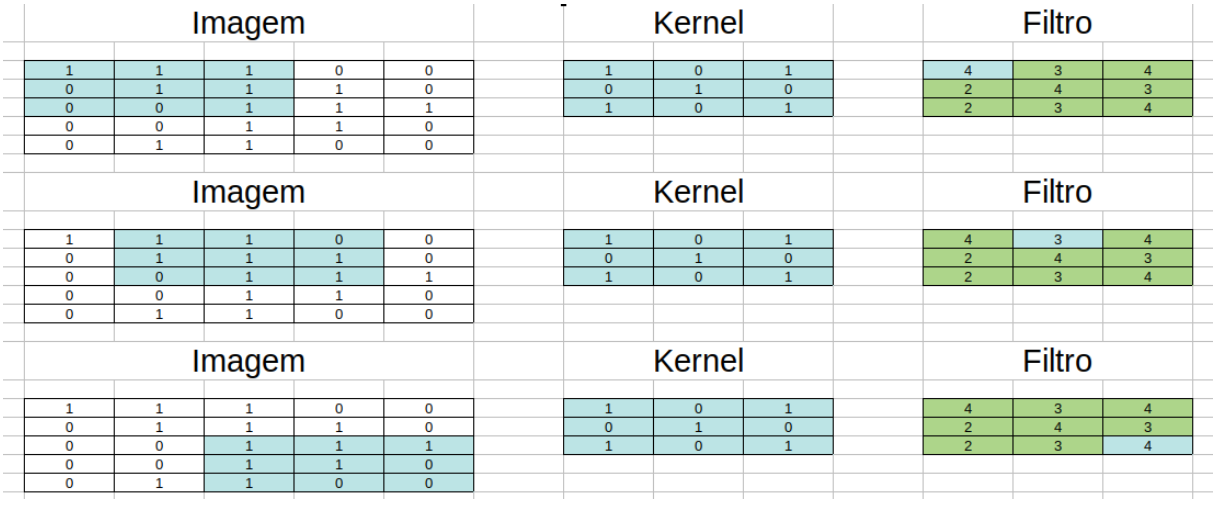

Figura 3.12: Exemplo aplicação *kernel*

Além da camada convolucional, também existe a camada de *pooling*. Esta camada tem como objetivo simplificar a matriz gerada pelas camadas convolucionais, através de duas abordagens: abordagem de maior (*Max pooling*) ou média (*Average pooling*). A Figura

3.13 mostra o caso do máximo. Para representar a média, ao invés de extrair o maior valor basta realizar $m = (1 + 1 + 5 + 6)/4$ por exemplo.

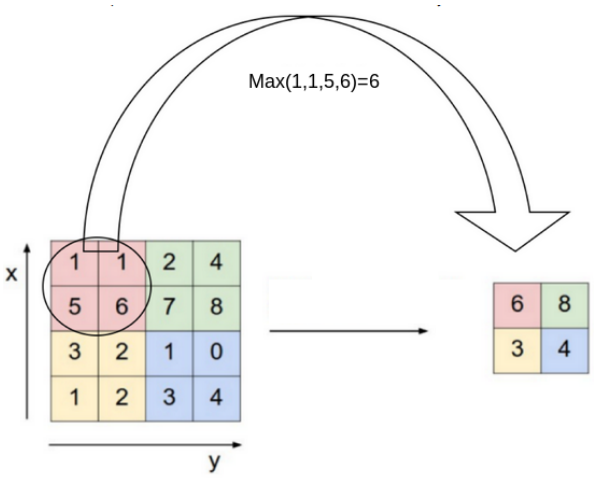

Figura 3.13: Exemplo *Max pooling* [46]

Um exemplo de aplicação de redes neuronais convolucionais é o caso do reconhecimento de caracteres com a arquitetura LeNet-5 [44], como mostra a Figura 3.14. Na Figura, *Subsampling* é o mesmo que aplicar *Max pooling*.

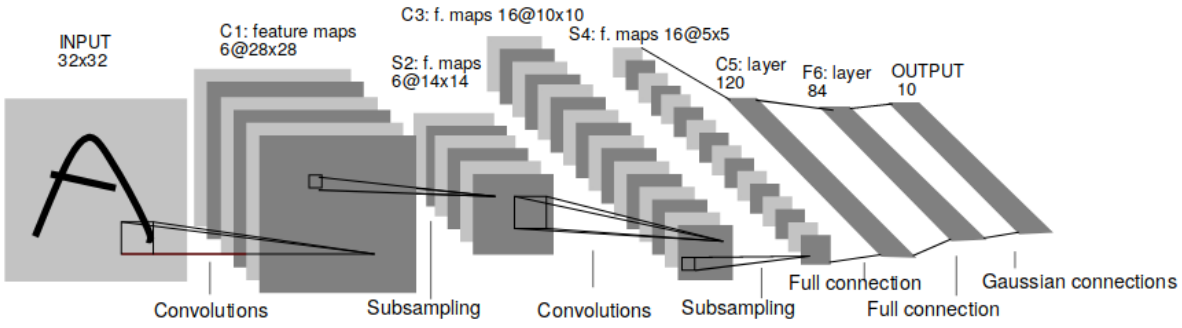

Figura 3.14: LeNet-5 [44]

Para problemas de classificação, ao final de uma CNN é as matrizes são normalmente transformadas em vetores unidimensionais para que sejam analisados por camada totalmente conectadas, profundas ou não. E por fim, é a camada de saída com sua respectiva função de ativação é usada por exemplo, para classificação. O Exemplo LeNet-5 utiliza deste processo.

Como visto, CNN são normalmente aplicadas em imagens, entretanto também é possível utilizar em áudio [47]. Ao invés de utilizar a matriz de pixels de 3 canais da imagem de um espectrograma, é sugerido binarizar as imagens ou passar as informações extraídas do espectrograma, ou seja, a matriz representando as frequências e os *frames*. Aplicando assim os conceitos dito anteriormente.

3.4 Transfer learning

Transfer learning é uma técnica de aprendizagem de máquina que diz que com base em uma rede treinada em um determinado problema, o aprendizado desta pode ser reutilizado para realizar a execução de problemas similares [48]. Para realizar tal técnica, é recomendado que a rede pré-treinada tenha excelente generalização no seu problema original.

Esta técnica também pode ser justificada principalmente quando não se apresenta uma base de dados grande para que seja treinada uma rede do zero, o que pode gerar *overfitting* na base.

A técnica vem sendo utilizada em problemas de *Deep Learning* principalmente para tarefas relacionadas a imagens, devido ao fato da facilidade de utilização de modelos pré-treinados com implementações em Python[1] e o contínuo desenvolvimento e melhorias para a classificação de imagens utilizando a base de dados ImageNet[2].

Um exemplo de aplicação desta técnica é a classificação de cães e gatos através de imagens. Ao invés de treinar uma rede inicialmente com pesos aleatórios, é possível reutilizar o aprendizado de redes treinadas na ImageNet para generalizar o novo classificador.

A utilização do aprendizado destes modelos pré-treinados pode ser realizado dentre duas abordagens principais: extração de características e *fine tunning*.

[1]Modelos pré-treinados para imagens: https://keras.io/applications/

[2]Desafio para classificação dentre as 1000 classes de imagens disponíveis. Site ImageNet: http://www.image-net.org/

3.4.1 Extração de características

A primeira abordagem é a reutilização das redes pré-treinadas para fazer o processo chamado extração de características, do inglês *Off-the-shelf feature extraction* [49]. Neste processo, a entrada desejada é ajustada para que esteja de acordo com a entrada da rede pré-treinada e propagada até uma determinada camada oculta, ou seja, é como se estivesse sendo feito o processo de predição da entrada, entretanto sem ir até a última camada para classificá-lo .

Deste modo, é gerado ao final da camada desejada da rede pré-treinada um tensor (também pode ser chamado *embedding*) de determinada dimensão que representa a informação da entrada para a base de dados treinada. Essa informação então pode ser retreinada ajustando um novo classificador para que seja possível a resolução de um novo problema. A Figura 3.15 mostra um exemplo onde são extraídas matrizes na terceira camada e reutilizadas em um novo classificador.

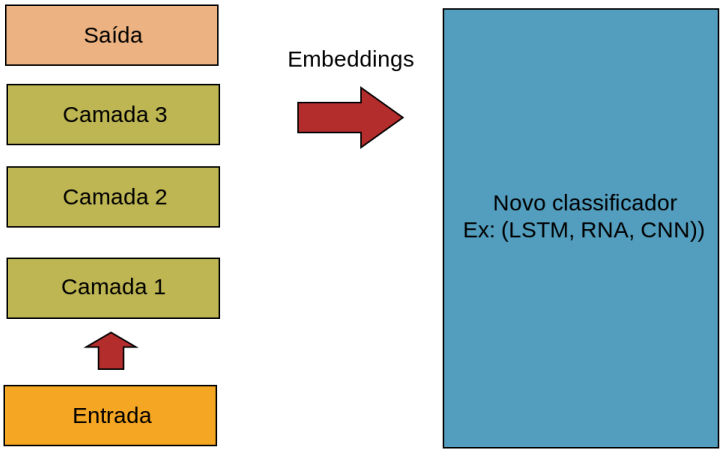

Figura 3.15: Exemplo extração de características com *transfer learning*

3.4.2 *Fine tuning* de modelo

Além da extração de características também existe a técnica *Fine tuning* de modelo [49]. Nesta abordagem é feito o processo de extrair a última camada do classificador pré-treinado e adicionar novas camadas para resolver um respectivo problema. Nota-se que toda a rede ou partes dela são retreinadas.

A vantagem desta abordagem é que facilita o processo de convergência da rede diminuindo o número de épocas a serem treinadas no novo problema. A desvantagem é que o custo de retreinar toda a rede é muito alto, exigindo um computador com processador e memória de melhor qualidade.

3.5 Métricas de avaliação e generalização de modelos

A avaliação de modelos de redes neuronais é feita com base em medidas de desempenhos. As medidas apresentadas são baseadas em quatro medidas mais, Verdadeiros Positivos (VP), Verdadeiros Negativos (VN), Falso Positivo (FP), Falso Negativo (FN) [50]. Para entender a diferença entre os conceitos, imagina-se o caso binário entre pessoas com patologia e pessoas saudáveis:

- **Verdadeiro Positivo (VP)**: os verdadeiros positivos são as pessoas que realmente tem patologia e que foram classificadas corretamente como doentes;
- **Verdadeiros Negativos (VN)**: os verdadeiros negativos é o oposto, são as pessoas saudáveis e que são classificadas corretamente como saudáveis;
- **Falso Positivo (FP)**: os falsos positivos são aquelas pessoas que são saudáveis e foram classificadas como doentes;
- **Falso Negativo (FN)**: os falsos negativos são aquelas pessoas que são doentes é que foram classificadas equivocadamente como saudáveis.

Estes conceitos podem ser melhor visualizados na Matriz de confusão, onde é disposta

a quantidade de elementos de cada classe, onde cada elemento foi classificado, e se este foi classificado corretamente. Um exemplo de Matriz de confusão é visto na Figura 3.16.

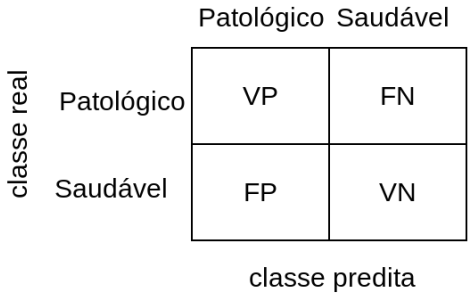

Figura 3.16: Exemplo Matriz de confusão para patológico e saudável

Com a matriz é possível avaliar o desempenho do modelo baseado em algumas métricas [50]. Dentre estas, é destaque para problemas de classificação as métricas de Exatidão (Acurácia), Precisão, Sensibilidade, Especificidade e Medida F.

3.5.1 Exatidão (Acurácia)

A métrica de exatidão calculada pela Equação 3.21 é a soma dos verdadeiros positivos e negativos dividido pela soma de todos os verdadeiros e falsos positivos e negativos. Esta é bastante utilizada em problemas de classificação, entretanto para teste com dados desbalanceados (desequilibrados) não é muito recomendada, pois acaba ignorando a classe com menor número, o que causa a impressão de que o modelo esta bom, entretanto apenas acertou mais da classe maioritária.

$$Exatidão = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.21}$$

3.5.2 Precisão

Esta métrica diz respeito à taxa de todas as instâncias positivas e que são realmente positivas, não é analisando os negativos, ou seja, a percentagem de todos os sujeitos

diagnosticados com patologia que são realmente doentes. Pode ser calculada pela Equação 3.22.

$$Precisao = \frac{VP}{VP + FP} \tag{3.22}$$

3.5.3 Sensibilidade e Especificidade

A métrica de Sensibilidade calculada pela Equação 3.23 e que também pode ser chamada de revocação, tem como objetivo identificar corretamente as instâncias positivas, por exemplo, para a separação entre patológico e saudável citado anteriormente será calculada a porcentagem de elementos que apresentam patologia e que foram classificados como positivos, ou seja, a porcentagem de patológicos corretamente diagnosticado.

$$Sensibilidade = \frac{VP}{VP + FN} \tag{3.23}$$

Em relação à métrica de Especificidade, calculada pela Equação 3.24 tem o objetivo inverso, irá identificar como falso todos os exemplos que foram classificados como falsos, ou seja, é calculada a percentagem de saudáveis corretamente diagnosticados.

$$Especificidade = \frac{VN}{VN + FP} \tag{3.24}$$

Ambas as medidas são bastante utilizadas em avaliação de classificadores com dados desbalanceados (desequilibrados) para garantir a qualidade de acertos de cada classe do modelo.

3.5.4 Medida F

A Medida F é definida pela média harmônica ponderada da relação de teste de Precisão e Sensibilidade. Pode ser calculada pela Equação 3.25, onde P significa Precisão e S Sensibilidade. Quanto mais próximo de 1 o valor da Medida F, mais confiável é o modelo.

$$MedidaF = 2 * \frac{P * S}{P + S} \tag{3.25}$$

Esta medida é comumente utilizada pois apresenta uma avaliação mais confiável dos modelos, principalmente em modelos desbalanceados.

3.5.5 Validação Cruzada *k-fold*

Quando é apresentado uma base de dados pequena para um classificador, normalmente este tem dificuldades para generalizar o modelo, devido ao fato de que não existe um conjunto suficiente de dados para que a base possa ser dividida entre treino, validação e teste com números fixos. Com isso, são utilizadas algumas técnicas que contornam este problema [51]. Uma dessas técnicas é a validação cruzada em *k-fold* (grupos).

Esta técnica tem como objetivo dividir aleatoriamente a base de dados em N grupos (*k-folds*) onde cada instância da base é treinada N-1 vezes e testada apenas uma vez [51]. Além disso, também é possível extrair parte da base de treino para ser utilizada como validação. A Figura 3.17 mostra a técnica.

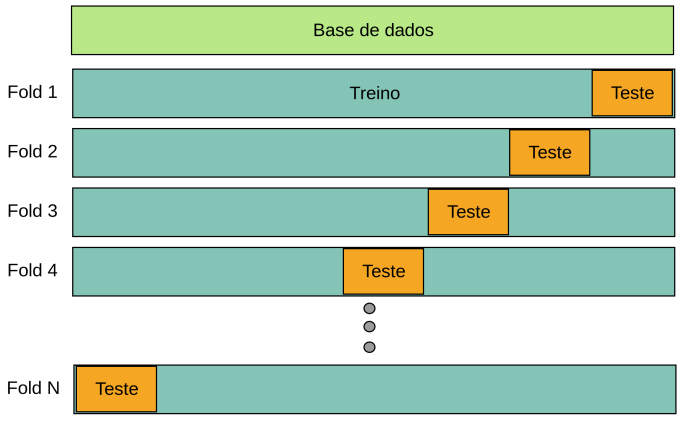

Figura 3.17: Validação Cruzada *k-fold*

A desvantagem de utilização desta técnica é de que o custo computacional e de tempo para treinar os "N *folds*" é muito maior do que dividir a base em números fixos.

Para avaliar a validação cruzada é possível gerar a matriz global representativa aos "N *folds*" do modelos e a partir desta matriz usar umas das métricas citadas anteriormente.

Capítulo 4

Metodologia e Desenvolvimento

Neste capítulo serão descritos os materiais e métodos desenvolvidos para detecção de patologias com base em três principais abordagens, a primeira utilizando parâmetros de *Jitter*, *Shimmer* e Autocorrelação, a segunda com MFCCs e a terceira com Espectrogramas. Antes disso serão descritos os materiais utilizados em todos as três abordagens.

4.1 Materiais

Foram utilizadas as seguintes ferramentas e bibliotecas para o desenvolvimento dos experimentos:

- **Python**[1]: é uma linguagem de programação de alto nível que permite o desenvolvimento de diversos paradigmas de programação, como orientação por objetos, procedural, imperativo e funcional. A linguagem é bastante utilizada para desenvolvimento de *scripts* para tarefas de Inteligência Artificial e Análise de dados. A linguagem também é usada como base para implementar as demais análises apresentadas neste trabalho;

- **Tensorflow**[2]: é uma biblioteca de código aberto desenvolvido pela Google que tem por objetivo facilitar a implementação de aplicações para Inteligência Artificial. Tem

[1]https://www.python.org/
[2]https://www.tensorflow.org/

suporte para implementação em *Central Process Unit* (CPU)s e *Graphics Processing Unit* (GPU)s, acelerando os cálculos de matrizes;

- **Keras**[3]: é uma Interface de programação de aplicações (em inglês API) de alto nível escrita em Python que facilita a implementação de redes neuronais. Tem suporte para os *backends* TensorFlow, *The Microsoft Cognitive Toolkit* (CNTK)[4] e Theano[5];

- **Scikit-learn**[6]: *framework* de código aberto de Python que auxilia no desenvolvimento de análise e mineração de dados.

A base de dados utilizada foi a base alemã SVD para indivíduos saudáveis e as doenças Disfonia, Laringite Crônica e Paralisia das Cordas Vocais que são as que apresentam a maior quantidade de áudios disponível. Além disso, a base alemã é a mais acessível dentre as três citadas no Estado da arte. A base MEEI e a FEMH não estão abertas a público. Mais detalhes de como será utilizada a base alemã nos experimentos é explicado nas metodologias a seguir.

As redes implementadas foram treinandos em um computador Inspiron 3421, Dell. Processador CPU Intel i5-3337U 1.80GHz. Sem GPU.

4.2 Metodologia para as vogais com *Jitter*, *Shimmer* e Autocorrelação

O primeiro experimento desenvolvido foi uma classificação binária entre pacientes com laringite crônica (pacientes diagnosticados com laringite crônica, mas que apresentam doenças secundárias) e pessoas saudáveis. Foram utilizados parâmetros de interesse extraídos conforme o trabalho desenvolvido por [19] com o software Praat. Tais parâmetros foram extraídos de três vogais /a/, /i/ e /u/ em três tonalidades diferentes, baixo, normal

[3]https://keras.io/
[4]Documentação CNTK: https://docs.microsoft.com/en-us/cognitive-toolkit/
[5]Documentação Theano: http://deeplearning.net/software/theano/
[6]https://scikit-learn.org/stable/

e alto. A Tabela 4.1 mostra como está disposta a base de dados criada, onde a primeira coluna representa o *Jitter* relativo que é medido em percentagem, o jitta que é mensurado em micro-segundos representa o *Jitter* absoluto, as próximas duas colunas contém as mesmas informações, mas para o *Shimmer*, e a última coluna apresenta a Autocorrelação. Os áudios pertencem à base de dados SVD, [8] e estão divididos em masculinos (33 saudáveis e 40 patológicos) e femininos (59 saudáveis e 30 patológicos). Os sinais têm 50 kHz de frequência de amostragem com 16 bits de resolução.

Valores Especificos intercalados									
Jitter				Shimmer				Harmonicidade	
jitter (%)		jitta (μs)		Shim		SHdB		Autocorrelação	
pat.	sau.	pat.	sau.	pat.	sau.	pat.	sau.	pat.	sau.
1,08	0,32	70,6	16,8	5,16	1,76	0,45	0,16	0,962	0,995
0,32	0,27	15,9	19,9	2,71	2,38	0,24	0,22	0,992	0,996
0,22	0,16	7,8	5,4	2,49	1,69	0,22	0,15	0,995	0,998
0,34	0,18	18,4	9,3	3,02	3,20	0,26	0,28	0,994	0,994

Tabela 4.1: Valores específicos intercalados - adaptado de [19].

Para fazer a classificação foi utilizada uma rede recorrente LSTM com uma camada de entrada totalmente conectada com formato (Número de passos de tempo, parâmetros) no qual é analisado um "time-step" com três parâmetros. Os parâmetros são analisados um por vez, ou seja, trata-se de uma sequência de 3 valores unitários. A seguir, existem duas camadas intermediárias LSTM com 3 neurônios e função de ativação Sigmóide Logística, seguido por um uma camada de normalização em lote [52]. No final é adicionada uma camada totalmente conectada com dois neurônios representando as duas classes e usando função Sigmóide Logística. O modelo é representado pela Figura 4.1. Para o cálculo do erro e otimização são utilizados respectivamente as funções de entropia cruzada categórica e descida do gradiente descendente estocástica [37]. Para efeito de comparação, também foi criada uma RNA semelhante, porém alterando as duas camadas intermediárias da rede LSTM para camadas totalmente conectada (Dense) sem recorrência.

O total da base de dados disponibilizado por [19] é de 1.454 instâncias e esses dados foram divididos em treino, validação e teste para quatro experimentos com rede LSTM,

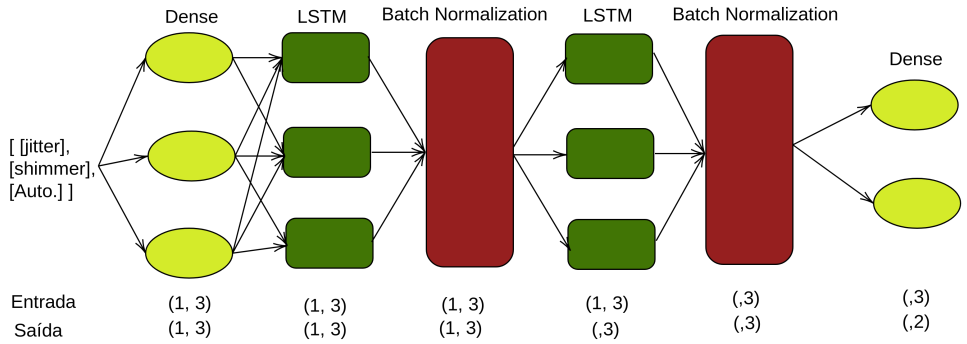

Figura 4.1: Modelo LSTM para *Jitter*, *Shimmer* e Autocorrelação

e um experimento com rede neuronal artificial clássica RNA. A divisão da base para os experimentos pode ser encontrada na Tabela 4.2. A RNA clássica foi implementada para fins de comparação, para verificar se é necessária a utilização de uma rede robusta como a LSTM.

Experimentos	**Treino**		**Validação**		**Teste**	
-	**total**	%	**total**	%	**total**	%
LSTM 1	1084	74,5	270	18,6	100	6,9
LSTM 2	818	56,3	273	18,8	363	25,0
LSTM 3	1050	72,2	186	12,8	218	15,0
LSTM 4	1049	72,1	186	12,8	219	15,0
RNA	1084	74,6	270	18,6	100	6,9

Tabela 4.2: Organização dos dados Laringite x Saudáveis dos experimentos iniciais - adaptado de [53]

Como a extração das características através do software Praat é um processo trabalhoso, é interessante a criação de algoritmos para extrair tais parâmetros automaticamente. Com isso, os trabalhos desenvolvidos por [17] para extração de *Jitter* e *Shimmer* e o trabalho de [20] para extração da autocorrelação foram utilizados como base para fazer a classificação das classes Disfonia, Laringite, Paralisia das cordas vocais e Saudáveis, gerando classificadores binários (2 classes) e multi classes (4 classes). Dois tipos de classificadores binários foram criados. No primeiro tipo, pacientes saudáveis foram considerados a classe negativa e uma patologia foi selecionada para ser a classe positiva. No segundo tipo, saudáveis também compuseram a classe negativa e pacientes com qualquer

uma das três patologias compuseram a classe positiva.

Para isso, foram realizados experimentos com o número de instâncias aproximadamente balanceado e não balanceado, onde cada indivíduo apresenta nove áudios (3 vogais * 3 tons). A Tabela 4.3 mostra como foram organizados os dados dos experimentos binários, sempre separando 15% para validação e teste (primeiramente é retirado 15% para teste da quantidade total e depois para o restante, são extraído os 15% para validação).

nº	**Experimento**	**Treino**		**Validação**		**Teste**	
	Não Balanceado	**total**	**%**	**total**	**%**	**total**	**%**
1	Disfonia x Saudáveis	1710	70	301	15	356	15
2	Laringite x Saudáveis	1527	70	270	15	318	15
	Aproximadamente Balanceado	**total**	**%**	**total**	**%**	**total**	**%**
3	Disfonia x Saudáveis	892	70	156	15	185	15
4	Laringite x Saudáveis	524	70	93	15	110	15
5	Paralisia x Saudáveis	2360	70	416	15	491	15
6	Patologias x Saudáveis	3075	70	543	15	639	15

Tabela 4.3: Organização dos dados dos experimentos binários com métodos clássicos

Os experimentos 1 e 3 possuem o mesmo foco. O Experimento 1 apresenta um total de 621 e 1.746 instâncias para disfonia e saudáveis respectivamente, sendo 529 instâncias de disfonia e 1482 de saudáveis para treino e validação, e 92 de disfonia e 264 de saudáveis para teste. No caso balanceado (Experimento 3) primeiramente é removido 1134 instâncias de saudável para ficar aproximadamente balanceado com um total de 621 (disfonia) e 612 (saudáveis), e é apresentado um total de 1048 para treino e validação, 95 (disfonia) e 90 (saudáveis) para teste.

Os experimentos 2 e 4 também são relacionados. A classificação binária entre Laringite e Saudável apresenta um total de 369 e 1.746 instâncias, respectivamente, onde para o caso não balanceado (Experimento 2) obtém-se de 320 instâncias de laringite e 1.477 de saudáveis para treino e validação, e 49 de laringite e 269 de saudáveis para teste. Para o caso balanceado (Experimento 4) primeiramente são removidas 1.388 instâncias de saudável, obtendo os valores de 316 e 301 para treino e validação, 53 e 57 para teste.

Para o Experimento 5, representado a classificação entre Paralisia e Saudável, as instâncias já estão aproximadamente balanceadas. Logo, só foi realizado um experimento,

com um total de 1.521 e 1.746 instâncias, onde 1.291 (paralisia) e 1.485 (saudável) são para treino e validação, e 230 (paralisia) e 261 (saudável) para teste.

No caso da classificação binária entre patológico e saudável representada pelo Experimento 6, foi feita a união das 3 patologias resultando em um total de 2.511 instâncias contra 1.746 dos saudáveis. Assim, a base foi dividida em 2.119 de patológicos e 1.499 de saudáveis para treino e validação, e 392, 247 para teste.

Para a classificação com as quatro classes são atribuído 529, 303, 1285 e 1501 instâncias de Disfonia, laringite, paralisia, saudável, respectivamente, para treino e validação. Para teste são atribuídas 92, 66, 236 e 245 instâncias, respectivamente.

4.3 Metodologia para as frases com MFCC

Nesta secção serão descritos os métodos desenvolvidos para detecção de patologias utilizando as frases em alemão com base em valores MFCCs.

Para os experimentos com os MFCCs e as frases foi realizada a implementação de três principais modelos baseados em redes RNA clássica, LSTM e CNN, fazendo a classificação entre as quatro patologias e a relação binária entre cada patologia e os saudáveis.

As frases fazem parte da base de dados Saarbrücken Voice Database [8] e são a representação da sentença em alemão *"Guten Morgen, wie geht es Ihnen?"* ("Bom dia, Como você está?"). Todos os arquivos estão com o formato WAVEform (WAV).

A extração dos MFCCs se deu pelo algoritmo em MATLAB desenvolvido no trabalho de [20], para um total de 69 instâncias de Disfonia, 41 de Laringite Crônica, 169 de Paralisia das Cordas Vocais e 194 de Saudáveis. Antes disso, foi necessário um pré-processamento no áudio para a remoção de silêncio no inicio e fim de cada um, utilizando-se uma média deslizante com janela de Hanning com comprimento de 35ms [20].

O algoritmo em MATLAB exige alguns parâmetros de entrada, que são a duração do *frame* de análise com o valor de duração de 35ms, o deslocamento do *frame* para um total de 50 janelas (com sobreposição), o coeficiente de pré-ênfase com valor de 0.97, o gama no intervalo de [300, 3700]Hz, o número de canais do banco de filtros igual a 20 e por fim o

número de coeficientes cepstrais igual a 13. Esses MFCCs foram exportados em formato *Comma-Separated Values* (CSV) para utilização.

Com os dados no arquivo CSV, foi preciso separá-los e adicionar um preprocessamento para utilizá-los como entrada nas redes escolhidas. O formato inicial de um arquivo em MFCCs é representado por uma matriz de 13 linhas por 50 colunas, assim para cada linha desta matriz deve ser aplicada a normalização dos dados, fazendo a normalização dos coeficientes ao longo dos 50 *frames* ao invés dos 13 para cada *frame* (coluna), assim a energia do sinal é normalizada ao longo do tempo. Isso faz com que todos os coeficientes de uma linha ao longo dos 50 *frames* estejam dentro de um intervalo entre -1 e 1.

Esta normalização é implementada com o auxilio do método *min-max-scaler.fit-transform* do scikit-learn no qual é passado como parâmetro o vetor desejado e é retornado o vetor normalizado. A Figura 4.2 mostra um exemplo de um arquivo de áudio onde cada linha é normalizada entre -1 e 1.

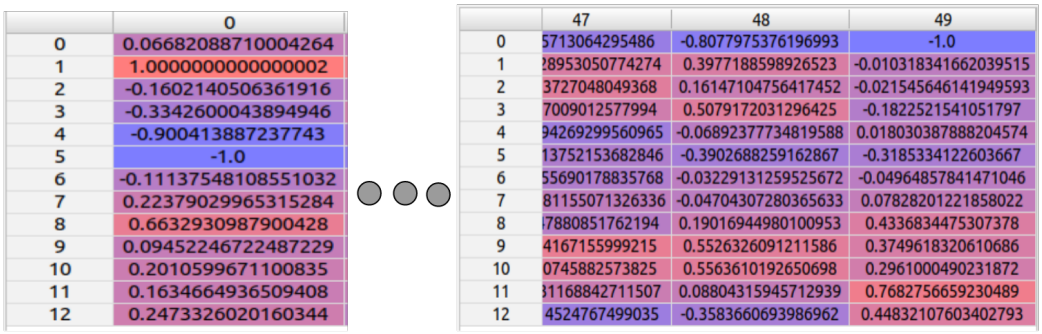

Figura 4.2: MFCC normalizado

Com a matriz normalizada é extraída sua a matriz transposta, assim o formato inicial de 13 linhas por 50 colunas, fica com 50 linhas por 13 colunas. Esta transformação simplifica o processo de ativação das redes LSTM e CNN.

Cada rede tem o seu formato de entrada. No caso da RNA clássica, a entrada deve ser um vetor unidimensional, os coeficientes no formato 50 por 13 são representados como um vetor contendo 650 elementos de entrada.

Para o caso do LSTM é definido o padrão de [instâncias, passos, parâmetros](N, 50, 13) onde o primeiro representa o número de exemplos de áudio, o segundo, representa os tempos a analisados para cada exemplo, e o terceiro o número de características analisadas em cada tempo.

Como existe uma relação de tempo nos dados, para o caso da CNN o formato de entrada segue o mesmo padrão da LSTM descrito anteriormente, entretanto é utilizado o método Conv1D do Keras ao invés de Conv2D. Isso foi implementado analisando-se uma coluna por vez para aplicar a convolução 1D, isto é, analisando apenas coeficientes de um mesmo passo no tempo.

Assim, foi feita uma busca em grade (*grid search*) para encontrar um bom modelo com as melhores configurações da RNA clássica, entretanto como o vetor de entrada é um vetor unitário, o número de neurônios de entrada foi testado com valores de [300, 350, 400, 450, 500, 550, 600, 650, 700]. O número de camadas da rede testado em [1, 2, 3, 4], o número de neurônios em cada camada foi de [300, 350, 400, 450, 500, 550, 600, 650, 700] e a taxa de aprendizagem de [0.1, 0.01, 0.001, 0.0001]. Com isso foram treinados 1296 modelos RNA onde os dez melhores podem ser vistos na Figura 4.3.

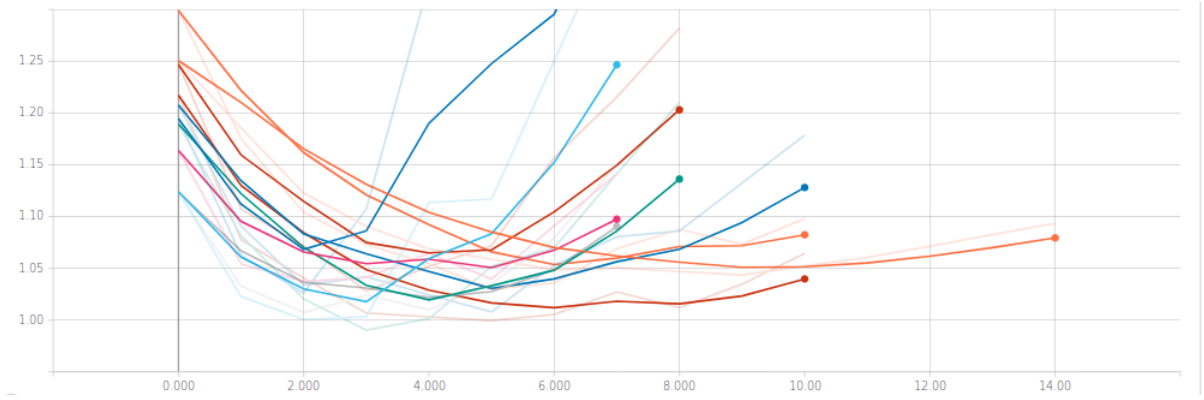

Figura 4.3: Erro validação dos melhores modelos RNA

O melhor erro de validação da Figura 4.3 foi de 1.012, cujo modelo usa 700 neurônios como entrada e função de ativação ReLU, apenas mais uma camada escondida de 350 neurônios com função ReLU e taxa de aprendizagem de 0.0001. Além da camada de saída com 4 neurônios com a função de ativação Softmax [37].

Como é possível observar na Figura 4.3, existe um aumento do erro em uma determinada época do treinamento, o que pode ser considerado como *Overfitting*, com isso foi adicionado *Dropout* [37] para a camada de entrada da rede. O modelo final da rede RNA Clássica pode ser visto na Figura 4.4.

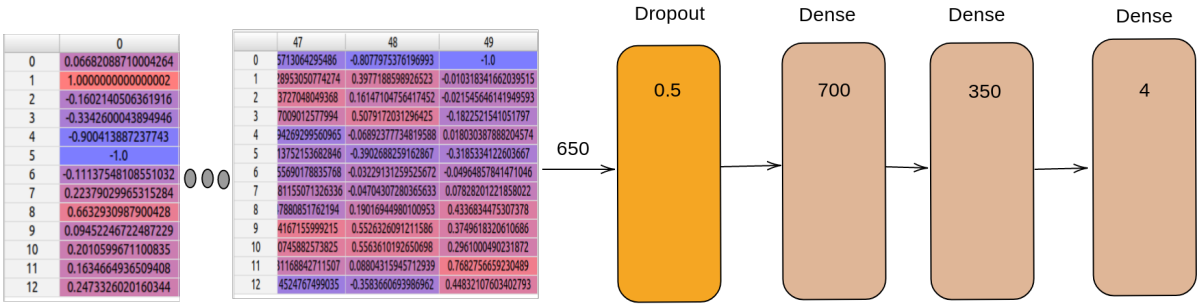

Figura 4.4: Modelo RNA clássica

O mesmo processo com as mesmas configurações de treino foi realizado para encontrar um melhor modelo LSTM, na qual é realizada uma análise do erro de validação do treino de 896 modelos diferentes, testando o número de neurônios da primeira camada em [2, 4, 8, 14, 32, 64, 128, 256], o número de camadas a serem utilizadas em [1, 2, 3, 4], o número de neurônios de cada camada restante em [2, 3, 8, 16, 32, 64, 128] e a taxa de aprendizagem em [0.1, 0.01, 0.001, 0.0001]. As redes foram treinadas extraindo 10% de cada classe para o grupo de teste, com um *batch size* de 32, 100 épocas de treinamento, entropia cruzada categórica como função de erro, otimização com Adam [54], e monitoramento através de EarlyStopping para interromper o treino quando a rede obtiver cinco maiores erros de validação em comparação com o menor erro salvo.

Dentre os 896 modelos, foram selecionados dez menores erros de validação, onde o menor entre eles obteve um valor de 0.8873. Esses modelos podem ser visualizados na Figura 4.5, onde o eixo X é o número de épocas e o eixo Y o erro. O menor erro de validação ocorreu próximo da época 35.

O modelo correspondente escolhido ao melhor erro é caracterizado com um LSTM de 4 neurônios na primeira camada, mais 3 LSTM com 32 neurônios em cada camada, e uma camada de saída totalmente conectada com 4 neurônios representando as 4 classes, com

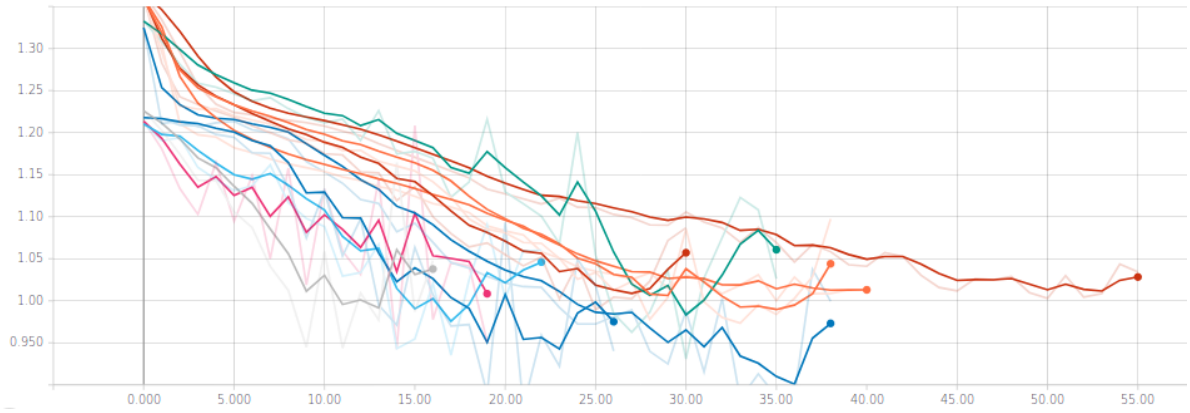

Figura 4.5: Erro validação dos melhores modelos LSTM

função de ativação Softmax [37] e uma taxa de aprendizagem de 0.001. O modelo pode ser visto na Figura 4.6

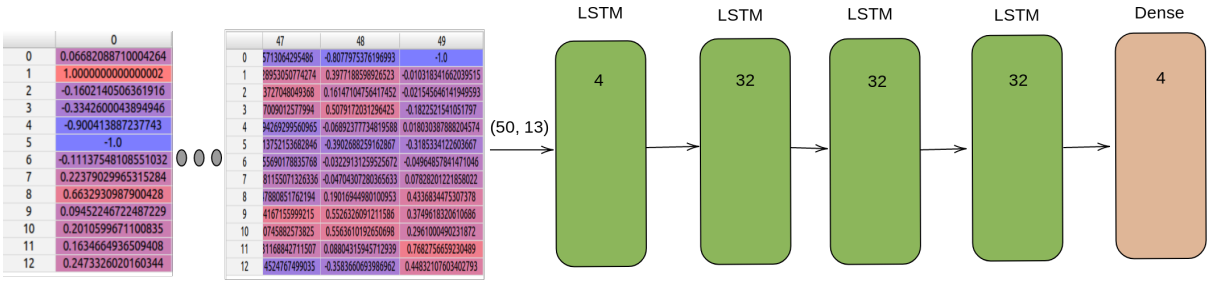

Figura 4.6: Modelo LSTM para MFCC

No caso da CNN, não foi realizado o treino de vários modelos pois o modelo mais básico foi considerado suficientemente bom, utilizando *Dropout* [55] para evitar *Overfitting*. A Figura 4.7 mostra a arquitetura escolhida para a Convolução unidimensional. A entrada é uma Conv1D de formato de entrada de (50, 13) com 32 filtros de dimensionalidade e *kernel* de tamanho 3, em seguida uma camada de *MaxPooling1D*, seguido por mais uma camada de Conv1D com 16 filtros e *Kernel* de 3, e uma camada de *GlobalAveragePooling1D* que transforma a matriz de três dimensões para duas. Por fim, é utilizada uma camada de *Dropout* com valor de 0.5, e duas camadas totalmente conectadas, a primeira com 16 neurônios e função de ativação ReLU [37] e a ultima sendo a saída da rede com 4 neurônios e função Softmax. As Conv1D também utilizam de função de ativação ReLU.

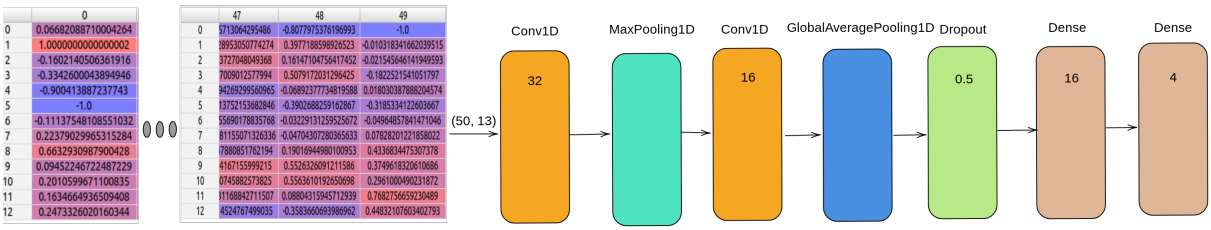

Figura 4.7: Modelo CNN Conv1D

Com os respectivos modelos selecionados, a base foi dividida em treino e teste, sem validação, pois o número de instâncias é pequena o que pode prejudicar o treinamento. A base foi dividida com o auxilio do método *train-test-split* da ferramenta scikit-learn extraindo para todos os casos 15% do total para teste.

Com isso, para os experimentos balanceados e aproximadamente balanceados entre Disfonia e Saudável obteve-se a organização de instâncias para cada classes descrita na Tabela 4.4, onde é apresentado um total de 223 instâncias para o treino e 40 para teste para o caso não balanceado, e no aproximadamente balanceado com 116 instâncias em RNA e Conv1D e 113 em LSTM, e 21 dados de teste.

nº	Disfonia x Saudável	Treino		Teste	
-	**Não Balanceado**	**dis.**	**sau.**	**dis.**	**sau.**
1	RNA	57	166	12	28
2	LSTM	60	163	9	31
3	Conv1D	59	164	10	30
-	**Aprox. Balanceado**	**dis.**	**sau.**	**dis.**	**sau.**
4	RNA	61	55	8	13
5	LSTM	58	55	11	10
6	Conv1D	60	56	9	12

Tabela 4.4: Organização dos dados para as Disfonia x Saudáveis com o MFCCs e frases

Para o caso entre Laringite e Saudável é obtida a Tabela 4.5, com uma quantidade de 199 instâncias para o treino e 36 para teste para o caso não balanceado, e no aproximadamente balanceado com 68 instâncias, e 12 dados de teste.

Para o caso entre Paralisia e Saudável é obtida a Tabela 4.6, com uma quantidade de 308 instâncias para o treino e 55 para teste para o caso aproximadamente balanceado.

Para o caso entre a união das três classes para a classificação binária em patológico

nº	Laringite x Saudável	Treino		Teste	
-	**Não Balanceado**	**lar.**	**sau.**	**lar.**	**sau.**
1	RNA	32	167	9	27
2	LSTM	35	164	6	30
3	Conv1D	35	164	6	30
-	**Aprox. Balanceado**	**lar.**	**sau.**	**lar.**	**sau.**
4	RNA	34	34	7	5
5	LSTM	34	34	7	5
6	Conv1D	36	32	5	7

Tabela 4.5: Organização dos dados para as Laringite x Saudáveis com o MFCCs e frases

nº	Paralisia x Saudável	Treino		Teste	
-	**Aprox. Balanceado**	**Par.**	**sau.**	**Par.**	**sau.**
1	RNA	142	166	27	28
2	LSTM	136	172	33	22
3	Conv1D	143	165	26	29

Tabela 4.6: Organização dos dados para as Paralisia x Saudáveis com o MFCCs e frases

e saudável é obtida a Tabela 4.7, com uma quantidade de 402 instâncias para o treino e 71 para teste. Para este caso foi criado apenas o experimento não balanceado, assumindo que as quantidades não influenciariam no desempenho no modelos.

nº	Experimento	Treino		Teste	
-	**Patologia x Saudável**	**pat.**	**sau.**	**pat.**	**sau.**
1	RNA	242	160	37	34
2	LSTM	240	162	39	32
3	Conv1D	234	168	45	26

Tabela 4.7: Organização dos dados para as Patologia x Saudáveis com o MFCCs e frases

Para a classificação das 4 classes é obtido a Tabela 4.8, com uma quantidade de 402 instâncias para o treino e 71 para teste. Como a quantidade de instâncias para cada classes é considerada pequena foi realizado apenas o teste desbalanceado.

Os experimentos foram desenvolvidos com um tamanho *batch* 32 em 100 épocas de treinamento. É aplicado ao final de cada época um embaralhamento dos dados de treinamento e sempre está sendo monitorado o menor erro do teste para ser utilizado como modelo escolhido. No final são feitos os experimentos em validação cruzada com 10 *Folds*

nº	Experimento	Treino				Teste			
-	**4 classes**	dis.	lar.	par.	sau.	dis.	lar.	par.	sau.
1	RNA	61	36	143	162	8	5	26	32
2	LSTM	61	37	139	165	8	4	30	29
3	Conv1D	59	36	144	163	10	5	25	31

Tabela 4.8: Organização dos dados para as quatro classes com o MFCCs e frases

[51] para os dados balanceados utilizando 10% do total do treino para validação.

Os resultados de validação cruzada são calculados através da função *classification report*[7] do scikit-learn. Nesta função é calculado a precisão, sensibilidade e medida F de cada classe, no final é apresentada a média desses resultados. Essa média será apresentada para análise dos resultados aqui abordados.

4.4 Metodologia com *transfer learning* nas frases

O ideal para fazer um *transfer learning* é que se tenha um modelo já treinado e que este tenha um bom desempenho no problema a ser resolvido. Neste sentido, como este trabalho está utilizando frases para detecção de patologia da voz, seria interessante que existisse um modelo treinado em várias classes patológicas e que tenha relação a voz, entretanto este problema ainda não foi resolvido. Com isso, foi necessário adotar as abordagens que serão explicadas nesta seção.

A hipótese deste experimento é de que um modelo pré-treinado pode extrair parâmetros relacionadas que auxiliam na resposta de um novo problema. Assim, o modelo pré-treinado utilizado foi o VGGish com AudioSet da Google [47], [56].

AudioSet é uma base de dados com 2,1 milhões de áudios catalogados, o que é equivalente a 5,8 mil horas de áudio divididos em 527 classes. Dentre essas classes, estão presente áudios de musica, voz, veículos, instrumentos musicais, entre outros [56]. O número de algumas dessas classes pode ser visto na Figura 4.8.

Inicialmente treinado para a classificação de imagens no desafio ImageNet[8], o modelo

[7]Documentação scikit-learn: https://scikit-learn.org/stable/documentation.html

[8]ImageNet é uma base que contem mais 14 milhões de imagens divididas em 1000 classes. Site oficial:

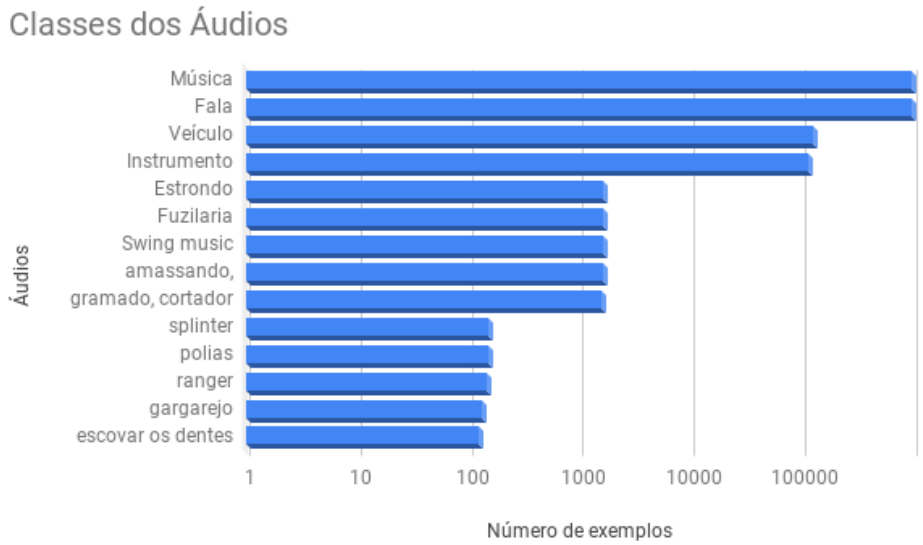

Figura 4.8: Classes da base AudioSet - traduzido e adaptado de [56]

VGGnet foi adaptado e treinado na base AudioSet. Este é composto por quatro camadas convolucionais e de *max polling*, com funções de ativação ReLU, seguidas por duas camadas totalmente conectadas com ReLU e a camada de *embedding* que é a camada utilizada para extrair matrizes de características.

Como dito no Capítulo 3 existem algumas técnicas para aplicar *transfer learning* e que podem ser aplicadas neste modelo. Retreinar a rede para novas classes adicionando mais camadas totalmente conectadas no topo do modelo é uma opção. Também é possível usar o modelo como extrator de parâmetros e posteriormente usar essas extrações como entrada de um novo classificador para determinar a classe desejada. A segunda abordagem (extração de características) foi escolhida.

Em relação à base patológica utilizada, a base alemã SVD [8] com a frase *"Guten Morgen, wie geht es Ihnen?"* ("Bom dia, Como você está?") foi utilizada. Entretanto foi feito novamente o *download* dos áudios, excluindo as vogais e usando apenas as frases, o que gerou um aumento no número de pacientes. Ficando com 70 pacientes de disfonia, 82 de laringite crônica, 197 de paralisia das cordas vocais e 632 para pessoas saudáveis.

http://www.image-net.org/

Para fazer a extração das características usando o modelo VGGish foi preciso fazer um pré-processamento. Inicialmente foi feito a remoção de silêncio do inicio e do fim de cada áudio para evitar que este influencie no aprendizado do modelo. Para isso, primeiramente foi preciso estabelecer um limite de níveis para ser considerado como silêncio, esse número foi definido calculando a média em dB de todos os primeiros níveis dos áudios. Em seguida foi desenvolvido um método que percorre o áudio em fatias de 10 milissegundos e analisa se o valor do decibel deste fragmento é menor do que -41dB (limite de silêncio) estabelecido, se sim, reconhece o silêncio. Isso é feito até encontrá um valor maior que o estabelecido (exemplo -20dB).

É realizado o mesmo processo para o silêncio do fim do áudio (usando -41dB), passando o vetor do áudio de trás para a frente. Os silêncios no meio do sinal não foram analisados nem removidos, pois isso pode ser uma dificuldade de uma determinada patologia.

Após a remoção do silêncio é realizado o pré-processamento para que seja compatível com os dados treinados na base da Audioset e que seja possível a utilização do modelo VGGish.

Primeiramente, todos os áudios são convertidos para 16kHz de frequência de amostragem de estéreo para mono, depois são extraídos os espectrogramas através da Transformada de Fourier de curto termo com um tamanho de janela de 25 ms, saltos de 10ms e janela de Hanning. Em seguida é feito a extração dos mel espectrogramas para um valor de 64 mels cobrindo um intervalo de 125-7500Hz. Estes dados extraídos são então agrupados em *frames* de 0.01 segundos (diferente do autor que usa 0.96 segundos) sem sobreposição, onde cada exemplo apresenta um formato de 64 bandas de mel por 96 *frames* [47]. É utilizado o tamanho de 0.01s ao invés de 0.96s para o número de exemplos, pois o tamanhos dos áudios é pequeno e com o valor de 0.96s acaba agrupando muita informação do sinal em apenas um conjunto, o que diminui a precisão para analisar a informação.

Os espectrogramas na escala mel no formato 96x64, são passados a rede VGGish para extrair os *embeddings* que retorna um formato de Nx128, ou seja, o tempo por 128 parâmetros. Como o valor de N pode variar dado o tamanho do áudio, foram feitos vários testes com os classificadores para se encontrar o melhor a ser utilizado. foram testados os

valor de 10, 50, 80, 100, 200 e o maior valor de N do maior áudio é de 363. O ideal seria utilizar o valor de 363, entretanto o melhor valor foi o de 80.

Nestes experimentos para se limitar o valor de 80, é utilizado o conceito de preenchimento com zeros, onde os áudios que apresentam N menores de 80 são preenchidos com zeros e os que tem mais de 80 são truncados. Ao final, o formato de entrada que serão utilizados nos classificadores é de 80x128, e pode ser visto na Figura 4.9 onde N inicialmente era de 68.

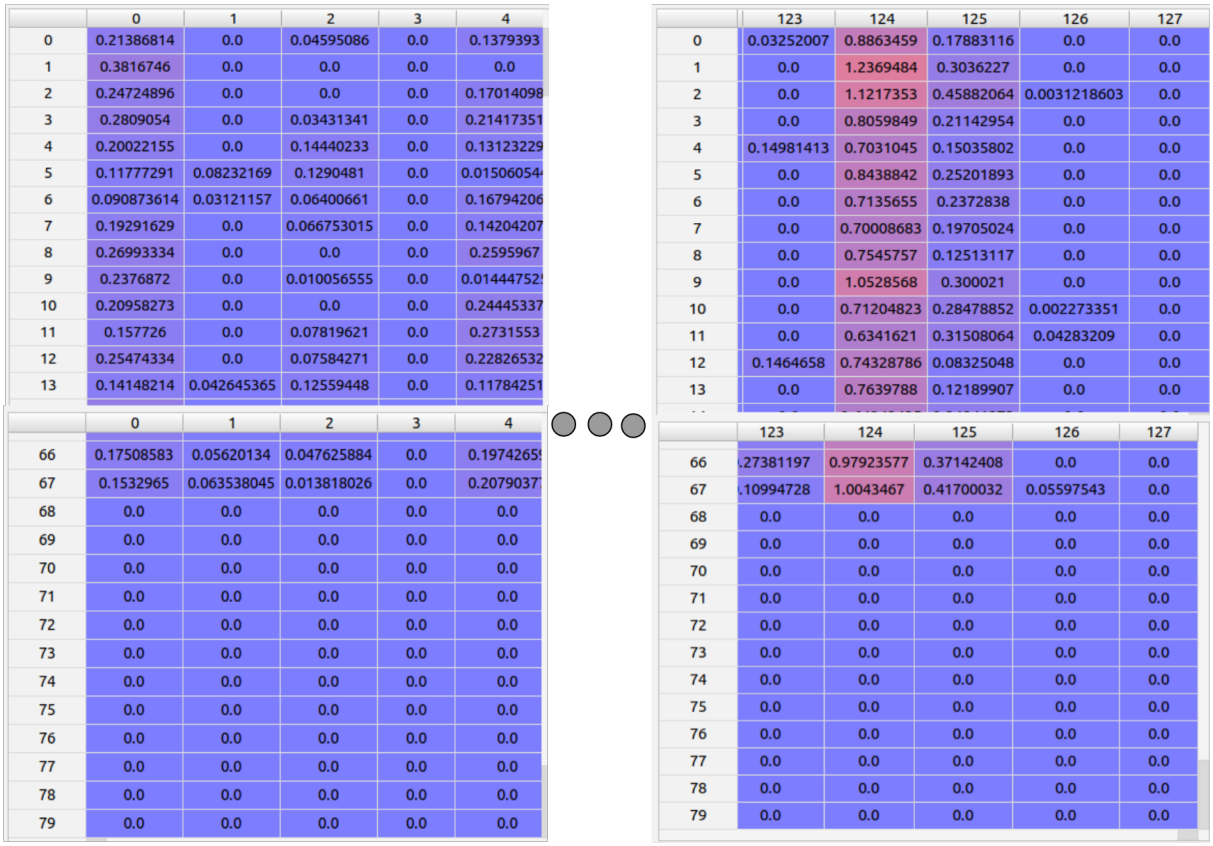

Figura 4.9: *Embedding* no formato 80x128 de um áudio de N igual a 68

Com essas matrizes agora é possível utilizar os *embeddings* extraídos dentre as quatro classes disponíveis. Para isso, foram desenvolvidos dois modelos, um baseado em LSTM e outro em Conv1D.

O modelo LSTM, na Figura 4.10, é composto por uma camada de entrada de normalização em lote com formato de (80, 128), seguido por duas camadas LSTM com 64 e

32 neurônios cada e função de ativação tangente hiperbólica. Por último, uma camada totalmente conectada com os quatro neurônios e função Softmax.

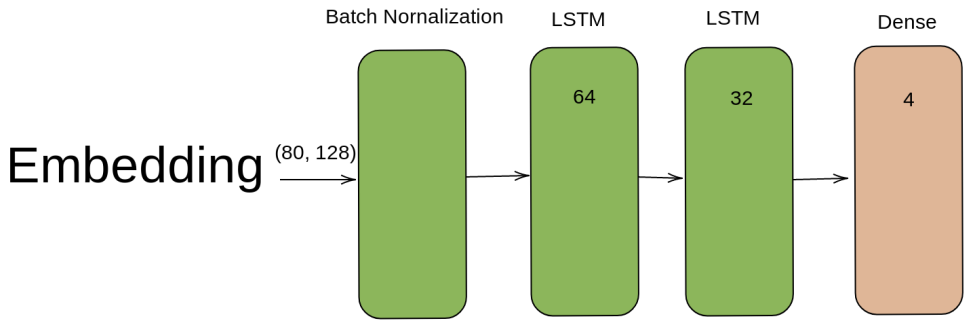

Figura 4.10: Modelo LSTM com *transfer learning*

O modelo Convolucional, representado na Figura 4.11, é composto inicialmente também por uma camada de normalização em lote, seguido por uma camada de convolução 1D com 32 neurônios, valor de *Kernel* 3 e tangente hiperbólica como função de ativação, seguido por uma camada de *Max pooling* com valor 2, outra convolução 1D com a mesma configuração anterior e 16 neurônios, uma camada de *Global Avarage pooling 1D* para diminuir a dimensão, uma camada de *Dropout* com valor de 0.2, uma camada totalmente conectada com 16 neurônios e função tangente hiperbólica, e por fim, a camada de saída.

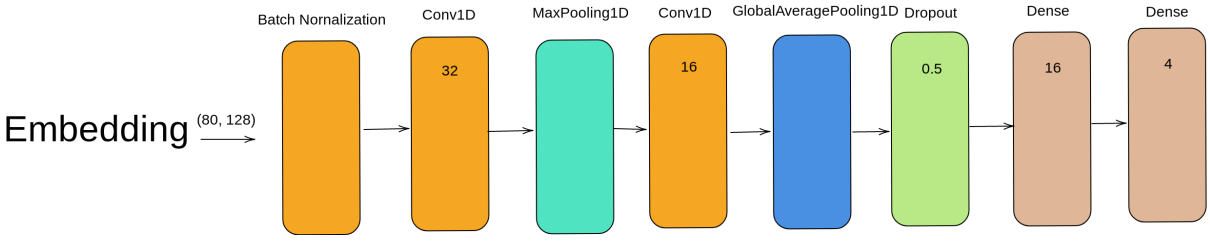

Figura 4.11: Modelo Conv1D com *transfer learning*

Do total de 70 de disfonia, 82 de laringite, 197 de paralisa e 632 de saudáveis, foi utilizado apenas 70 de disfonia, 82 de laringite, 76 de paralisia, 78 de saudáveis para que os modelos estivessem com dados aproximadamente balanceados para a classificação em quatro classes.

Para as classificações binárias foi definido em disfonia o balanceamento de 70 instâncias para disfonia e 78 para saudáveis, laringite com 82 e saudáveis com 78, paralisia com 197

instâncias e saudáveis com 197.

Os modelos foram treinados em validação cruzada de 10 *folds*, extraindo 10% para validação, em 1000 épocas com condição de parada analisando o menor erro da validação, bach size no valor de 32, e embaralhamento dos dados ao final de cada época. Além disso, foi utilizada a função de erro entropia cruzada categórica e otimização Adam com taxa de aprendizado de 0.0001.

Os resultados de validação cruzada são calculados através da função *classification report*[9] do scikit-learn, igual à metodologia dos MFCCs com frase.

Em resumo, esta metodologia é definida pela Figura 4.12.

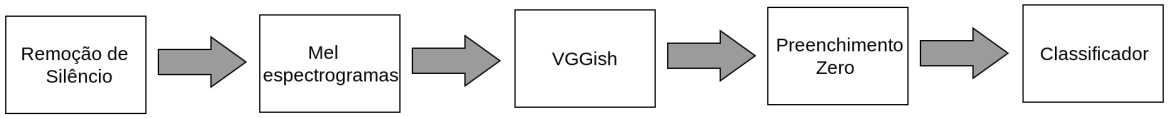

Figura 4.12: Metodologia *transfer learning*

[9]Documentação scikit-learn: https://scikit-learn.org/stable/documentation.html

Capítulo 5

Resultados e discussão

Neste capítulo serão descritos os resultados e a discussão das metodologias abordadas em três partes, a primeira referente aos parâmetros de *Jitter*, *Shimmer* e Autocorrelação, a segunda análise para as RNA baseadas em MFCCs e a terceira considerando os espectrogramas com *Transfer Learning*. Em seguida serão comparados os resultados com trabalhos relacionados.

5.1 Resultados para as vogais com *Jitter*, *Shimmer* e Autocorrelação

A Tabela 5.1 mostra os resultados dos experimentos iniciais. Pode-se observar que aqueles são contrários à hipótese inicial de a rede clássica ser poderosa o suficiente para a análise, considerando que seu desempenho não foi bom comparado aos outros experimentos iniciais, alcançando apenas 85% de exatidão na base de teste. Para os outros experimentos (LSTM 1,2,3) a exatidão foi de 100%. Os testes para os experimentos LSTM 1, 2 e 3 levantaram uma preocupação em relação a uma possível presença de *overfitting*. A rede poderia estar memorizando a sequência na qual os padrões são apresentados. Para investigar essa questão, o experimento LSTM 4 foi desenvolvido dividindo a base em 72% para treino, 12% para validação e 15% para teste onde os dados são embaralhados em cada

época do treino, evitando que seja treinado em uma época somente dados de uma única classe. Assim o experimento LSTM 4 obteve uma acurácia de 99%.

Experimentos	Validação		Teste	
x	Acurácia	erro	Acurácia	erro
LSTM 1	0.988	0.132	1.0	0.131
LSTM 2	0.996	0.024	1.0	0.344
LSTM 3	1.0	0.061	1.0	0.575
LSTM 4	1.0	0.006	0.990	0.117
RNA	0.969	0.505	0.850	0.464

Tabela 5.1: Resultados Laringite x Saudáveis - experimentos iniciais [53]

O gráfico e a matriz de confusão do Experimento LSTM 4 pode ser visto na Figura 5.1 e a Tabela 5.2 respectivamente. No gráfico é possível observar a acurácia da validação após cem épocas de treinamento, chegando a uma acurácia quase constante de 100%. Na matriz de confusão é exposto como as instâncias de teste estão classificadas. Como a quantidade de elementos não é igual é interessante usar a métrica de Sensibilidade e Especificidade para saber se há *bias* favorecendo uma das classes.

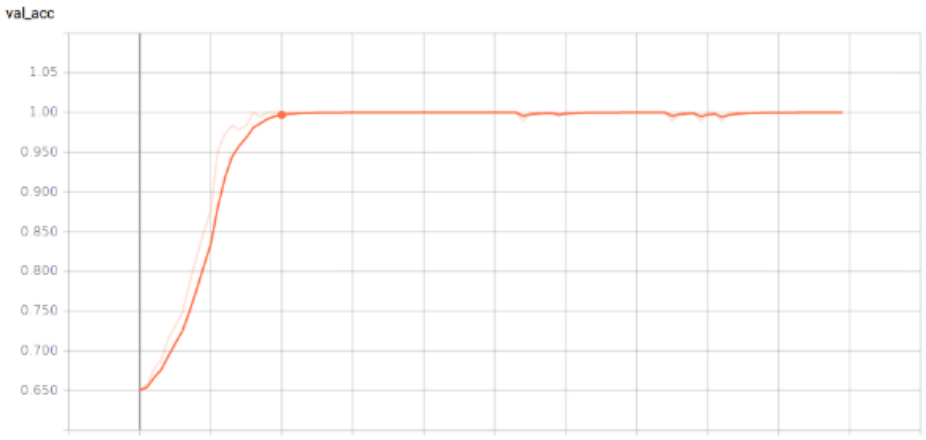

Figura 5.1: Acurácia da validação Experimento 4 [53]

Essas métricas podem ser calculadas segundo as Equações 3.23 e 3.24. Assim analisando a matriz do Experimento LSTM 4 obtém-se os valores de 99% de sensibilidade e 100% de especificidade [53].

Classe Real	Laringite	95	1
	Saudável	0	123
	x	Laringite	Saudável
		Classe predita	

Tabela 5.2: Matriz de confusão Laringite x Saudáveis - adaptado de [53]

Com esses resultados foi utilizada a mesma configuração do Experimento LSTM 4 para fazer a classificação com as outras classes, entretanto, agora usando os parâmetros extraídos e desenvolvidos por [17] e [20] (base diferente). A Tabela 5.3 mostra os resultados da exatidão para os conjuntos de treino, validação e teste, e a sensibilidade e especificidade no conjunto de teste.

-	**Treino**	**Validação**	**Teste**	**Sensibilidade**	**Especificidade**
Não Balanceado	%	%	%	%	%
Disfonia x Saudáveis	76,3	76,5	77,2	73,9	77,5
Laringite x Saudáveis	83,1	81,1	85,5	71,1	85,9
Aprox. Balanceado	%	%	%	%	%
Disfonia x Saudáveis	65,3	64,7	58,9	59,0	60,9
Laringite x Saudáveis	56,7	58,0	63,1	63,8	60,6
Paralisia x Saudáveis	69,4	68,3	67,8	68,0	68,9
Patologias x Saudáveis	63,5	64,1	64,5	66,6	61,1

Tabela 5.3: Resultados experimentos binários não balanceado e balanceado

Analisando a Tabela 5.3 é possível observar que a relação de sensibilidade para disfonia entre os experimentos balanceado e desbalanceado é bem discrepante, uma diferença de 14,9%, onde a quantidade de dados removido pode estar influenciando no desempenho do modelo.

Os resultados obtidos para os experimentos com a Laringite também são discrepantes, no caso balanceado obteve-se 63,8% e 60,6% de sensibilidade e especificidade comparado à 71,1% e 85,9% para não balanceado. Destaca-se que, nesta nova base extraída, os resultados não foram tão bons comparados com o experimento em [53], entretanto esta base é menor.

Para a paralisia a sensibilidade e especificidade representam 68,0% e 68,9% respectivamente, o que mesmo com um percentual baixo o modelo está conseguindo lidar com as

diferenças das classes.

Para o confronto de patológico e saudável obteve-se a Matriz de confusão representada pela Figura 5.2. Onde o modelo respondeu com 63,5% no treino, 64,1% na validação e 64,5% no teste, com 66,6% de sensibilidade e 61,1% de especificidade.

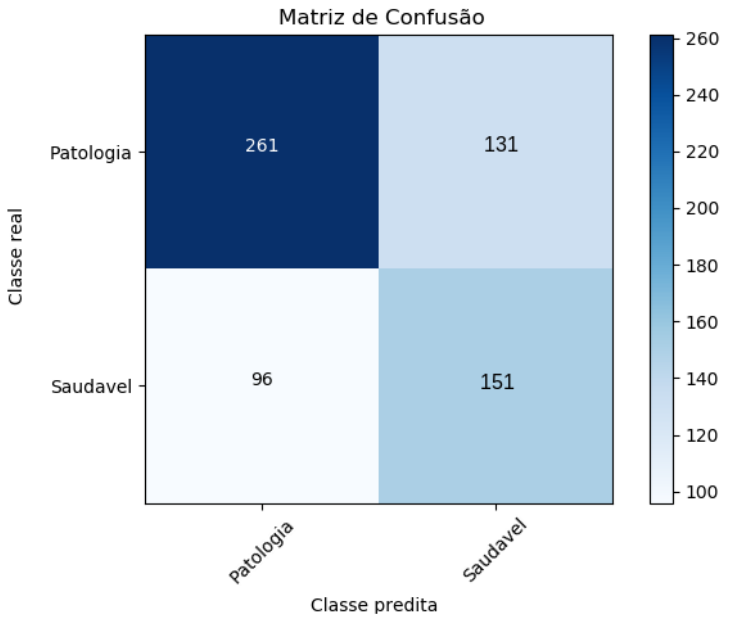

Figura 5.2: Matriz de confusão para classificação entre patológico e saudável

Em relação à classificação para as quatro classes, a exatidão obtida foi de 53,98% no treinamento, 56,72% na validação e 51,01% no teste, a matriz de confusão deste experimento pode ser visto na Figura 5.3.

Entretanto o modelo LSTM utilizado está aprendendo as classes maioritárias Paralisia e Saudável o que mostra que o modelo não identifica a disfonia nem laringite, ignorando-os totalmente, como pode ser visto na Figura 5.3. Com mais dados para as outras classes a rede poderia aprender melhor estas diferenças.

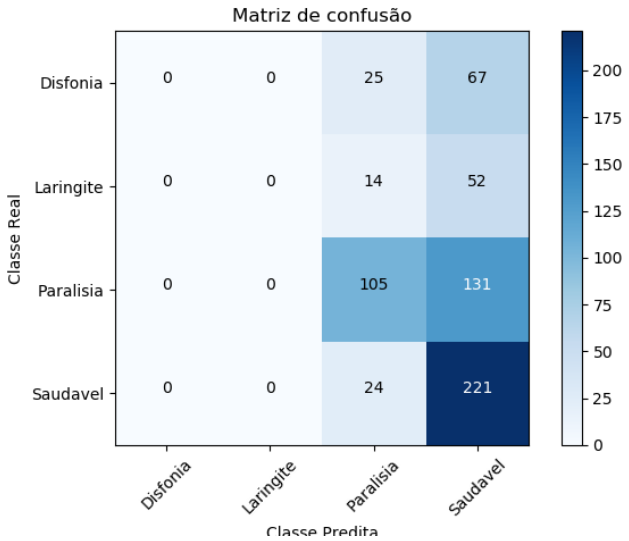

Figura 5.3: Matriz de confusão para as quatro classes para vogais

5.2 Resultados para as frases com MFCC

Esta seção irá mostrar os resultados dos MFCCs com as frases. Está dividido em duas subseções, sem a utilização de validação cruzada e com validação cruzada.

5.2.1 Sem validação cruzada de 10 *folds*

O experimento binário para disfonia e saudável representado pela Tabela 5.4 (referentes ao conjunto de teste) e referente à organização de dados da Tabela 4.4 mostra uma grande diferença ente os dados aproximadamente balanceados e não-balanceados. Para os dados não balanceados a Sensibilidade vale 14,28% na RNA clássica e 0% para LSTM e Conv1D, o que significa que o modelo praticamente classificou todos os pacientes como saudáveis. Apenas no modelo RNA clássica é que foi classificado apenas 1 paciente. Entretanto, quando o modelo é balanceado, são obtidos resultados mais indicativos. Neste caso, a RNA clássica se saiu melhor entre as três abordagens, atingindo um valor de 87,50% de sensibilidade e 92,30% de especificidade, errando apenas um paciente para cada classe. Em relação ao LSTM e Conv1D, o segundo modelo teve um melhor desempenho.

nº	Disfonia x Saudável	Exatidão	Sensibilidade	Especificidade
-	Não Balanceado	%	%	%
1	RNA	73,40	14,28	66,66
2	LSTM	77,70	0,00	77,50
3	Conv1D	75,00	0,00	75,00
-	Aprox. Balanceado	%	%	%
4	RNA	90,47	87,50	92,30
5	LSTM	66,66	66,66	66,66
6	Conv1D	76,19	75,00	76,92

Tabela 5.4: Resultados Disfonia x Saudável com MFCC das frases

Apesar dos bons resultados para a RNA clássica, a base para os experimentos é considerada pequena. O modelo se comportou bem, sem a presença de *overfitting*.

Em relação aos experimentos para laringite e saudável, apresentados na Tabela 5.5, relativas à Tabela 4.5, mais uma vez o caso aproximadamente balanceado mostrou melhores resultados, onde ocorre um empate de 75% de exatidão entre os três modelos. O que diferencia é a sensibilidade do caso LSTM atingindo 100%, e indicando que nenhum paciente com disfonia foi classificado como saudável e que o inverso também acontece. Entretanto a quantidade de instâncias no teste é pequena.

nº	Laringite x Saudável	Exatidão	Sensibilidade	Especificidade
-	Não Balanceado	%	%	%
1	RNA	69,44	25,00	75,00
2	LSTM	83,33	0,00	83,33
3	Conv1D	86,11	100,00	85,57
-	Aprox. Balanceado	%	%	%
4	RNA	75,00	83,33	66,66
5	LSTM	75,00	100,00	62,50
6	Conv1D	75,00	66,66	83,33

Tabela 5.5: Resultados Laringite x Saudável com MFCC das frases

Para os casos não balanceados, o melhor resultado obtido foi a Conv1D com 86,11% de teste, 100% de sensibilidade e 85% de especificidade. De 6 pacientes com laringite, o modelo só acertou 1 e, de 30 pacientes saudáveis, o modelo acertou todos. Apesar dos bons resultados, é possível que o modelo esteja favorecendo na classe maioritária.

Para a classificação binária entre paralisia e saudável apresentada na Tabela 5.6, relativo à Tabela 4.6, o melhor modelo acabou sendo o Conv1D com 72,72% de taxa de acerto, 73,91% de sensibilidade e 71,87% de especificidade. Apesar do LSTM ter obtido 80% na sensibilidade, este não foi melhor, pois acabou errando mais pacientes com paralisia, 20 acertos de 33 contra 17 acertos de 26 da Conv1D.

nº	**Paralisia x Saudável**	**Exatidão**	**Sensibilidade**	**Especificidade**
-	Aprox. Balanceado	%	%	%
1	RNA	65,45	65,53	65,51
2	LSTM	67,27	80,00	56,66
3	Conv1D	72,72	73,91	71,87

Tabela 5.6: Resultados Paralisia x Saudável com MFCC das frases

Na classificação binária entre a união das três classes patológicas e saudável apresentada na Tabela 5.7 e relativo à Tabela 4.7, os resultados foram muito próximos um do outro, tendo neste caso a RNA clássica um melhor desempenho com 76,05% de exatidão, 73,80% de sensibilidade e 79,31% de especificidade. A Conv1D apresentou um resultado melhor para sensibilidade, entretanto esse apresentava 8 instâncias a mais de patologia no teste. Apenas a LSTM ficou um pouco distante dos demais modelos. Os resultados sugerem que a RNA clássica é mais eficiente, até mesmo por ser um modelo mais simples, porém não foram realizados testes estatísticos para confirmar essa hipótese.

nº	**Patologia x Saudável**	**Exatidão**	**Sensibilidade**	**Especificidade**
-	-	%	%	%
1	RNA	76,05	73,80	79,31
2	LSTM	69,01	68,08	70,83
3	Conv1D	73,23	77,08	65,21

Tabela 5.7: Resultados Patologia x Saudável com MFCC das frases

Os resultados para a classificação entre as quatro classes (calculados através da função *classification report* do scikit-learn) apresentados na Tabela 5.8 e relativo à Tabela 4.8 mostram que os modelos LSTM e Conv1D foram melhor, tendo o Conv1D apresentado uma leve vantagem. O primeiro com 60,56% de exatidão, 61% de sensibilidade na média total das classes e 55% de medida F, e o segundo com 66,20% de exatidão no teste, 66%

de sensibilidade na média das classes e 60% para a medida F.

nº	4 classes	Exatidão	Sensibilidade	Medida-F
-	-	%	Média Total	Média Total
1	RNA	50,70	0,51	0,48
2	LSTM	60,56	0,61	0,55
3	Conv1D	66,20	0,66	0,60

Tabela 5.8: Resultados 4 classes com MFCC das frases

Apesar dos resultados, ambos os modelos tiveram classificação boas para as classes saudáveis e paralisia que são as classes majoritárias, ignorando completamente as classes disfonia e laringite. As matrizes de confusão da LSTM e Conv1D podem ser vistas nas figuras 5.4 e 5.5, respectivamente. Nessas matrizes, é possível observar que a Conv1D acaba classificando um pouco melhor que a LSTM, acertando mais pacientes em suas respectivas classes.

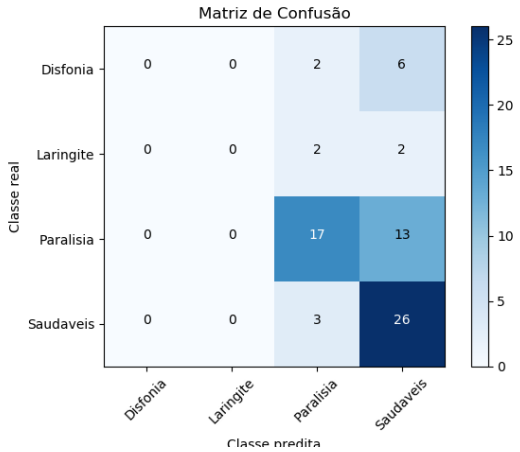

Figura 5.4: Matriz de Confusão LSTM para as 4 classes

O que também chama atenção é o erro do teste da classificação das quatro classes, representado pela Figura 5.6. Neste é possível observar que o erro da Conv1D diminui com o passar das épocas de treinamento, o que é esperado, entretanto para os casos da RNA e LSTM os erros diminuem até uma determinada época e depois aumentam consideravelmente, o que é um sintoma de *overfitting*. Neste sentido a utilização da Conv1D para estas comparações parece a mais promissora.

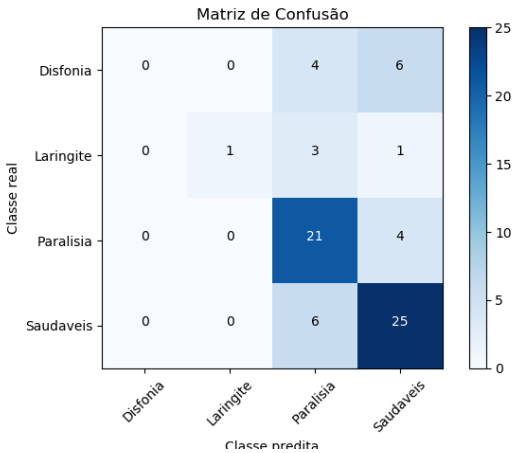

Figura 5.5: Matriz de Confusão Conv1D para as 4 classes

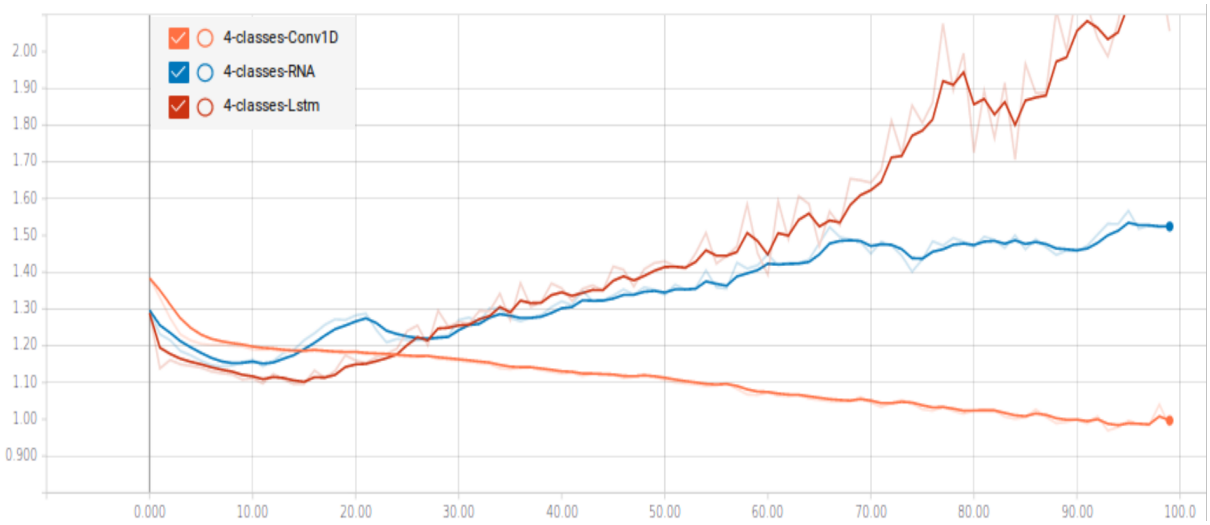

Figura 5.6: Erro do Teste para as 4 classes

Como os experimentos anteriores para as quatro classes foram realizados de maneira desbalanceada, este fator deve ser levado em consideração. Assim deve-se avaliar o comportamento dos dois melhores modelos balanceando os dados.

Para isso, como a classe de Laringite apresenta um total 41 instâncias, foram feitos sete experimentos aumentando balanceadamente os dados até o total de 140, nos quais o número de instâncias por classe variou entre 5 e 35, com um aumento gradual de 5 instâncias por classe. Além disso é extraído 10% de cada classe para utilizar como teste e como validação. Os resultados podem ser vistos nas Tabelas 5.7 e 5.8.

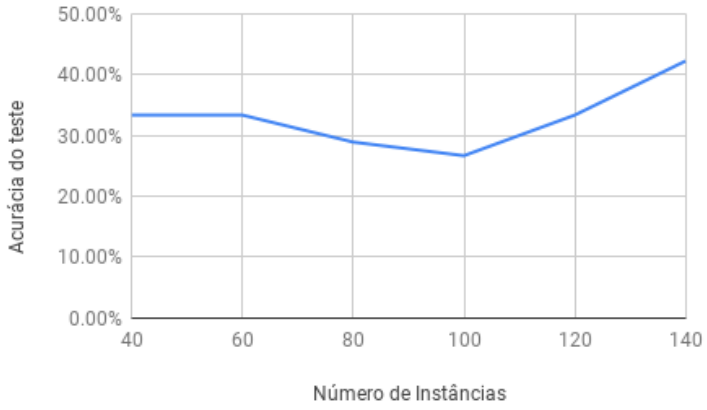

Figura 5.7: Acurácia de teste LSTM para 4 classes balanceado

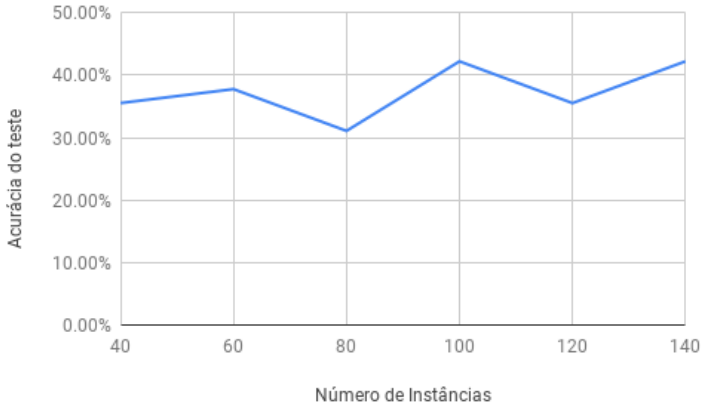

Figura 5.8: Acurácia de teste Conv1D para 4 classes balanceado

Na Figura 5.7 é possível observar que, com o aumento de instâncias, o modelo perde um pouco de desempenho, mas por volta de 100 instâncias a exatidão volta a crescer. Isso mostra que, com dados balanceados, e mais dados comparados a base atual, a tendência é que a acurácia também aumente. No caso Conv1D da Figura 5.8 também segue a mesma tendência, entretanto este acaba oscilando um pouco mais com menos dados.

5.2.2 Com validação cruzada de 10 *folds*

Efetuando-se os experimentos com validação cruzada 10 *folds* para a classificação binária de um total de 69 instâncias para disfonia e 68 para saudáveis, são obtidos os resultados apresentados na Tabela 5.9. Esta Tabela mostra que o modelo RNA clássico se saiu melhor obtendo resultados de 76,0% de sensibilidade e 86,8% de especificidade ao final dos 10 *folds* de validação cruzada. A Figura 5.9 mostra a matriz de confusão global respectiva à RNA clássica.

Disfonia x Saudável 10 Folds	Precisão	Medida-F	Sensibilidade	Especificidade
-	%	%	%	%
RNA	**76,0**	**76,0**	**76,0**	**86,8**
LSTM	52,0	55,0	58,0	64,7
Conv1D	48,0	49,0	50,0	49,1

Tabela 5.9: Resultados Disfonia x Saudável para validação cruzada com 10 *folds*

Para a classificação binária em 41 instâncias de laringite e 39 de saudável são obtidos os resultados da Tabela 5.10, que mostra uma vantagem do modelo RNA clássica com 59,0% de sensibilidade e 57,1% de especificidade.

Laringite x Saudável 10 Folds	Precisão	Medida-F	Sensibilidade	Especificidade
-	%	%	%	%
RNA	**59,0**	**59,0**	**59,0**	**57,1**
LSTM	38,3	43,0	49,0	42,9
Conv1D	49,1	51,0	53,0	52,0

Tabela 5.10: Resultados Laringite x Saudável para validação cruzada com 10 *folds*

Para a classificação binária com os dados balanceados em 169 instâncias de paralisia e 165 de saudáveis, a RNA clássica acaba sendo a melhor, com valores de 65,0% de

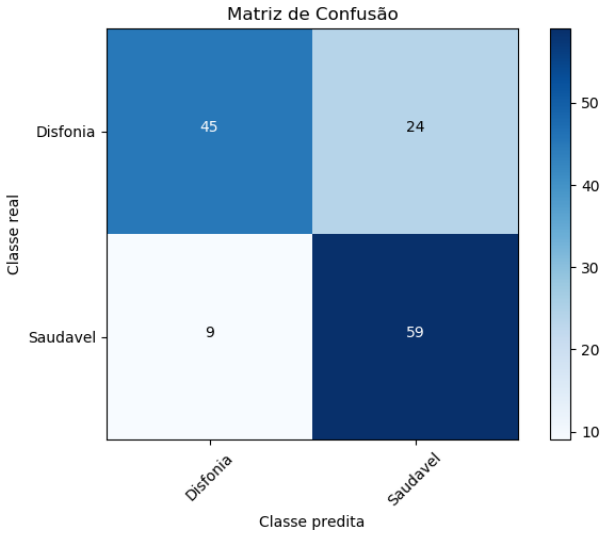

Figura 5.9: Matriz Global de Disfonia x Saudável para validação cruzada com MFCC

sensibilidade e 62,9% de especificidade, seguido pela LSTM e a Conv1D sendo a última. Os resultados são representados na Tabela 5.11.

Paralisia x Saudável 10 Folds	Precisão	Medida-F	Sensibilidade	Especificidade
-	%	%	%	%
RNA	**65,0**	**65,0**	**65,0**	**62,9**
LSTM	48,5	52,0	56,0	63,0
Conv1D	44,6	48,0	52,0	53,0

Tabela 5.11: Resultados Paralisia x Saudável para validação cruzada com 10 *folds*

Para a classificação com quatro classes com 35 instâncias em cada classe e retirando 10% para validação é obtido 27% de sensibilidade para RNA clássica, 28% para LSTM e 23% para Conv1D. Como a base para esse experimento é muito pequena não é possível obter resultados razoáveis.

O recomendado para utilização de modelos de *Deep Learning* é que se tenha uma quantidade grande de dados, a base utilizada não apresenta isso. Entretanto, acredita-se que os modelos utilizados podem ser capazes de fazer tais classificações corretamente, pois apresentam resultados satisfatórios para a classificação binária, sendo necessário apenas mais dados.

Neste sentido, foi feito novamente o *Download* da base de dados das frases para as quatro classes, e ficou visível o aumento de número de elementos para as classes de Laringite e Paralisia, agora com 82 e 197 (antes era de 41 e 169). Com os novos dados foi feito novamente a extração dos parâmetros com o algoritmo de [20] e aplicado novamente validação cruzada para os experimentos binários de laringite (balanceando em 82 para laringite, 82 para saudável), Paralisia (balanceando em 197 de paralisia, 196 para saudável) com uma leve vantagem para a RNA clássica. A Tabela 5.12 mostra os resultados e a Figura 5.10 as matrizes globais de laringite e paralisia

Experimentos	Precição	Medida-F	Sensibilidade	Especificidade
Laringite	%	%	%	%
RNA	**68,0**	**68,0**	**68,0**	**77,0**
Conv1D	57,0	57,0	57,0	56,0
LSTM	51,0	49,0	51,0	49,0
Paralisia	%	%	%	%
RNA	**78,0**	**77,0**	**77,0**	**82,0**
Conv1D	74,0	74,0	74,0	74,0
LSTM	63,0	63,0	63,0	63,0

Tabela 5.12: Resultados Laringite x Saudável / Paralisia x Saudável para validação cruzada com 10 *folds*

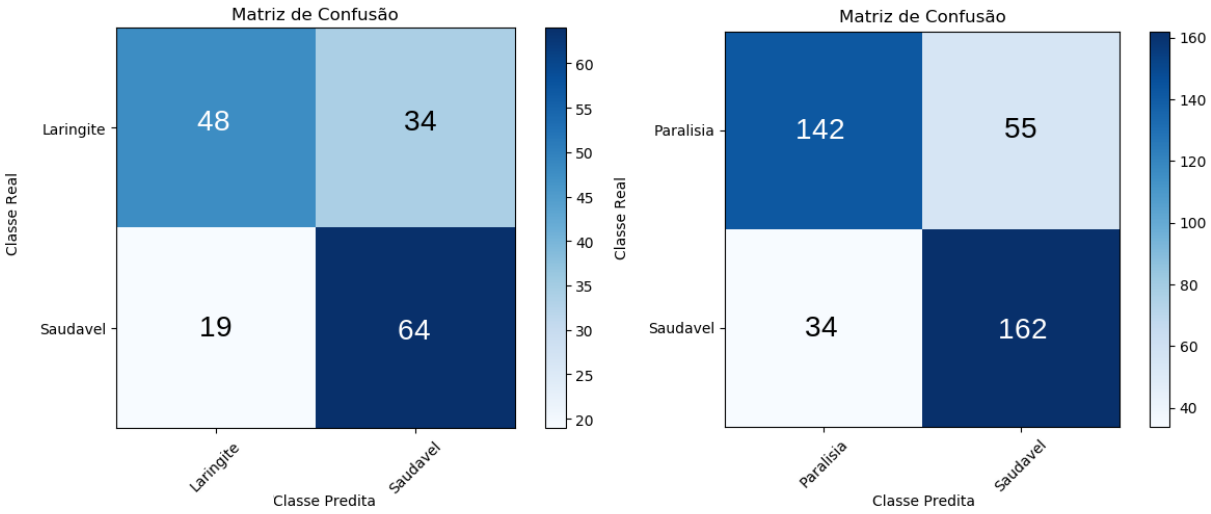

Figura 5.10: Matriz Global de Laringite x Saudável e Paralisa x Saudável para validação cruzada com MFCC

Além disso, também é realizado o experimento para a união das três classes patológicas contra os saudáveis, balanceando em 349 instâncias de patológicos e 348 de saudáveis, onde a RNA clássica se saiu melhor. Tabela 5.13 mostra esses resultados, e a Figura 5.11 a matriz global.

Experimentos	Precição	Medida-F	Sensibilidade	Especificidade
Patológico	%	%	%	%
RNA	**73,0**	**72,0**	**72,0**	**82,0**
Conv1D	61,0	60,0	61,0	57,0
LSTM	60,0	58,0	59,0	55,0

Tabela 5.13: Resultados Patológico x Saudável para validação cruzada com 10 *folds*

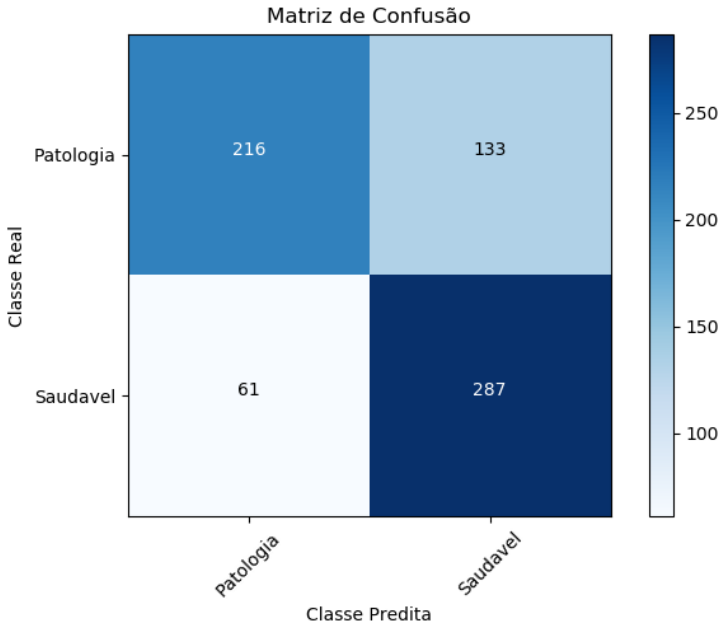

Figura 5.11: Matriz Global de Patológico x Saudável para validação cruzada com MFCC

Para o caso das quatro classes (balanceando em 70 para cada classe), que pode ser visto na Tabela 5.14, também houve um aumento, com destaque para a RNA clássica que aumentou de 27,0% para 35,0%. A Conv1D e LSTM não tiveram melhoras. Também foi realizado um teste balanceado os dados em 70 para disfonia, 82 para Laringite, 78 para Paralisia e 76 para Saudáveis, obtendo resultados de 38,0%. A matriz de confusão do segundo balanceamento para RNA clássica pode ser visto na Figura 5.12.

Experimentos	Medida-F	Sensibilidade	Precisão
4 classes (Balanceado em 70)	%	%	%
RNA	**35,0**	**35,0**	**36,0**
Conv1D	25,0	26,0	16,0
LSTM	20,0	24,0	18,0
4 classes - 2º Balanceamento	%	%	%
RNA	**38,0**	**38,0**	**38,0**

Tabela 5.14: Resultados 4 classes para validação cruzada com 10 *folds* e MFCC

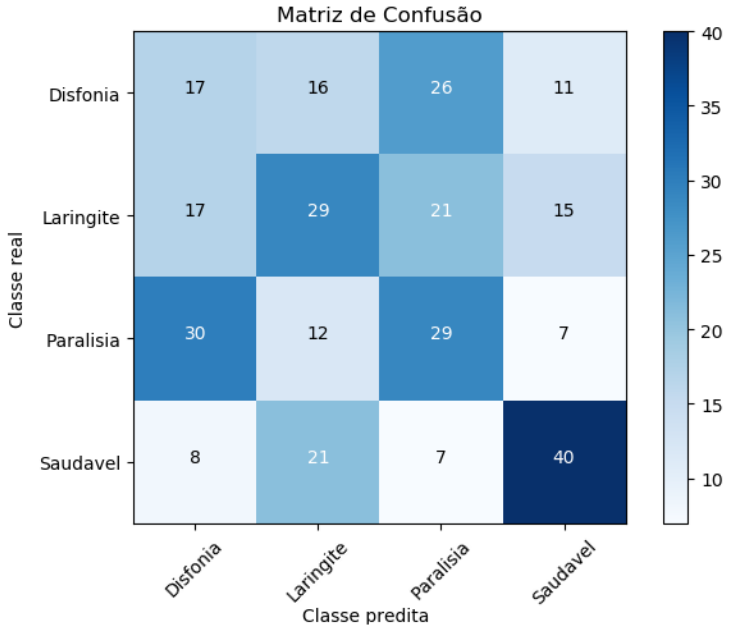

Figura 5.12: Matriz Global RNA clássica 4 classes

O que chama atenção é que houve uma melhora significativa tanto em laringite quanto em Paralisia, o que mostra que com mais variabilidade dos dados os modelos melhoram, entretanto para o caso de quatro classes, o número de dados ainda não é suficiente.

Como a classificação para a quatro classes não obteve resultados satisfatórios foram realizados os respectivos experimentos para três classes com o modelo RNA clássica: Disfonia x Laringite x Saudáveis; Laringite x Paralisia x Saudáveis; Disfonia x Paralisia x Saudáveis; Disfonia x Laringite x Paralisia. A quantidade de instancias para cada classe é de 70 em disfonia, 82 em laringite, 78 em paralisia, 76 em saudáveis. A Tabela 5.15 mostra os resultados.

3 classes - RNA	Precição %	Medida-F %	Sensibilidade %
Dis-Lar-Sau	51,0	51,0	51,0
Lar-Par-Sau	**59,0**	**59,0**	**59,0**
Dis-Par-Sau	56,0	55,0	55,0
Dir-Lar-Par	40,0	38,0	37,0

Tabela 5.15: Resultados 3 classes para validação cruzada com 10 *folds* e MFCC

Na Tabela 5.15 é possível observar que dentre as classificações a que se saiu melhor foi entre Laringite x Paralisia x Saudável com 59,0% na Medida F, o que faz sentido pois dados os sintomas as classes são bem distintas. O segundo melhor foi Disfonia x Paralisa x Saudável com 55,0%. A pior classificação foi a da relação entre as três classes. A Figura 5.13 mostra o melhor classificador de três classes, neste é possível visualizar que a classe que mais é classificado incorretamente é Laringite

5.3 Resultados para *transfer learning* nas frases

Para a classificação binária entre disfonia e saudável utilizando os conceitos de *transfer learning* e validação cruzada em 10 *folds* é possível observar na Tabela 5.16 que o modelo de Conv1D teve um desempenho levemente melhor que o modelo LSTM, obtendo um valor de 66,0% de sensibilidade, 71,0% de especificidade e 66,0% em medida-F, contra os 63,0% de medida-F. A respectiva classe predita pelas redes pode ser vista na Figura 5.14. A esquerda representando o modelo LSTM e a direita o modelo Conv1D.

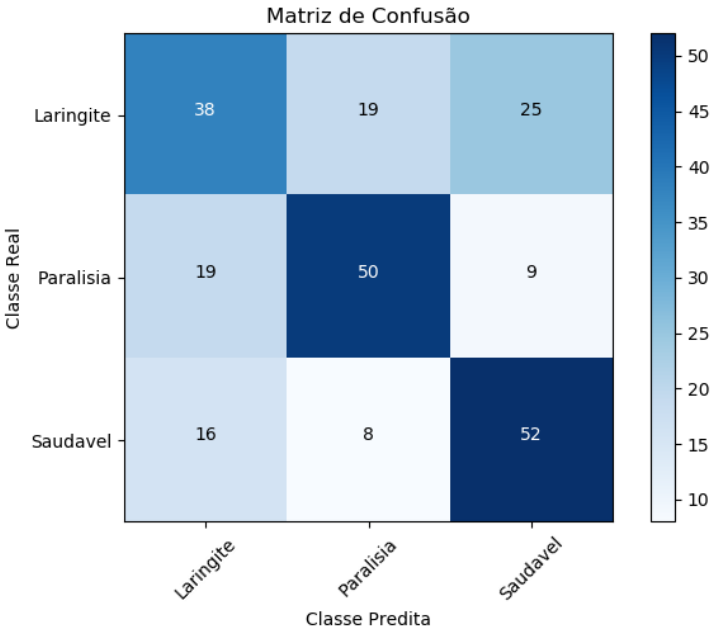

Figura 5.13: Matriz Global RNA clássica 3 classes com MFCC

Disfonia x Saudável 10 Folds	Precisão	Medida-F	Sensibilidade	Especificidade
-	%	%	%	%
LSTM	63,0	63,0	63,0	69
Conv1D	**66,0**	**66,0**	**66,0**	**71,0**

Tabela 5.16: Resultados Disfonia x Saudável 10 *folds* com *transfer learning*

Nestas matrizes globais, a diferença entre o acerto de saudáveis e de disfonia é de apenas duas instâncias, o mesmo para as classes que foram classificadas incorretamente. Com isso, apesar dos resultados da Tabela 5.16, pode-se dizer que ambos os modelos tiveram resultados próximos, podendo ser utilizado ambos para futuros experimentos.

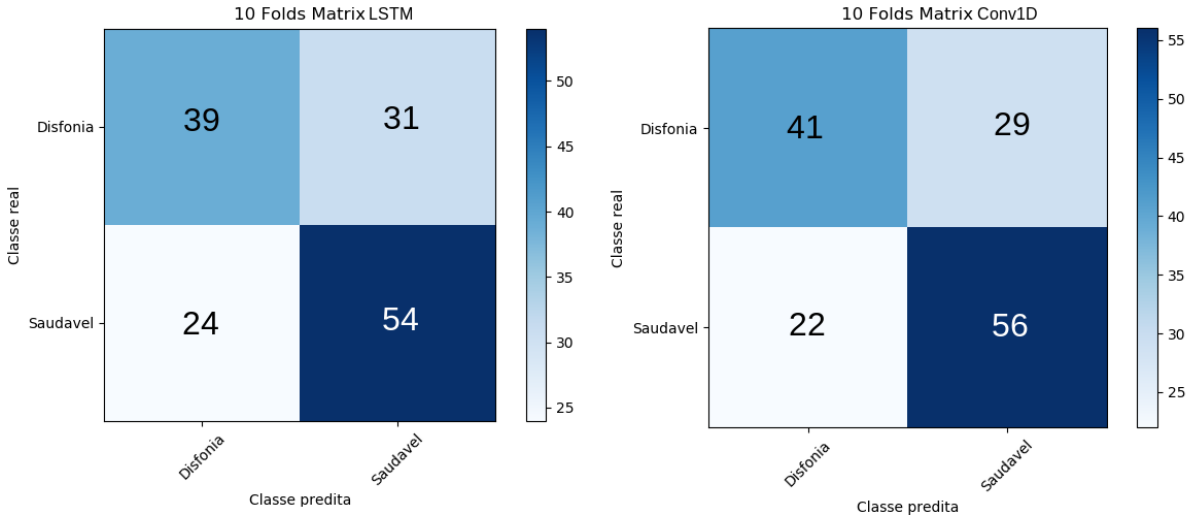

Figura 5.14: Matriz Global Disfonia x Saudável dos modelos LSTM e Conv1D *transfer learning*

Em relação à classificação entre laringite e saudável, representado pela Tabela 5.17, as diferenças entre os resultados é mais próxima, com uma leve vantagem para o modelo Conv1D, com valores de 67,0% de sensibilidade, 67,0% de especificidade e 67,0% de medida-F, contra 66,0% de sensibilidade, 67,0% de especificidade e 66,0% de medida-F.

Laringite x Saudável 10 Folds	Precisão	Medida-F	Sensibilidade	Especificidade
-	%	%	%	%
LSTM	66,0	66,0	66,0	67,0
Conv1D	**67,0**	**67,0**	**67,0**	**67,0**

Tabela 5.17: Resultados Laringite x Saudável 10 *folds* com *transfer learning*

O que tornou o modelo Conv1D um pouco melhor é que este acertou 54 instancias de disfonia contra os 52 do modelo LSTM. Também classificou menos dados errados de disfonia, equivalente a dois dados a menos classificados incorretamente. Esta diferença é apresentada na Figura 5.15. Neste sentido, não há um modelo que se saiu claramente

melhor.

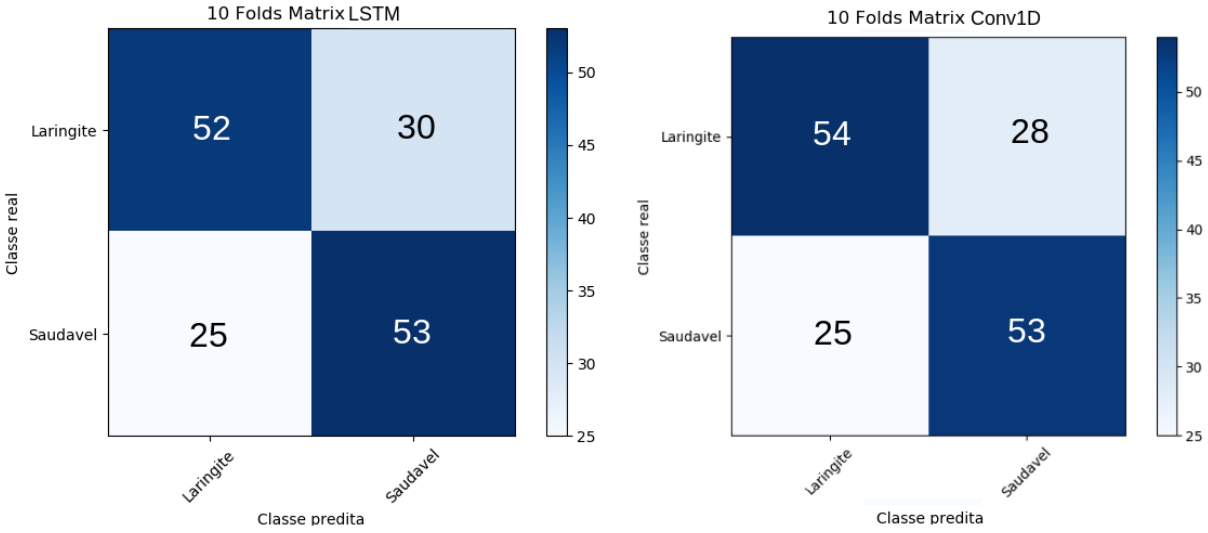

Figura 5.15: Matriz Global Laringite x Saudável dos modelos LSTM e Conv1D *transfer learning*

Para a análise entre paralisia das cordas vocais e as pessoas saudáveis, representada pela Tabela 5.18, é possível observar uma leve vantagem para o modelo LSTM, obtendo resultados de 80,0% para sensibilidade, especificidade e medida-F, contra os 78,0% e 80,0% do modelo Conv1D. Isto se deve ao fato de que o modelo LSTM acertou mais dados em paralisia, o equivalente a 159 contra 150. Também acabou errando menos quando a classe era paralisia e foi classificado como saudável. Essas relações são vistas na Figura 5.16.

Paralisia x Saudável 10 Folds	Precisão	Medida-F	Sensibilidade	Especificidade
-	%	%	%	%
LSTM	**80,0**	**80,0**	**80,0**	**80,0**
Conv1D	78,0	78,0	78,0	80,0

Tabela 5.18: Resultados Paralisia x Saudável 10 *folds* com *transfer learning*

Para a classificação binária entre a união das três classes patológicas contra saudáveis, balanceando os dados em 349 é obtido os resultados da Tabela 5.19, onde o modelo LSTM se saiu melhor, com 78,0% de Medida-F, contra 75,0% da Conv1D.

O que fez este modelo um pouco melhor é que este classificou corretamente mais dados patológicos, equivalente a 268 contra 250 da Conv1D. Também classificou 81 instâncias

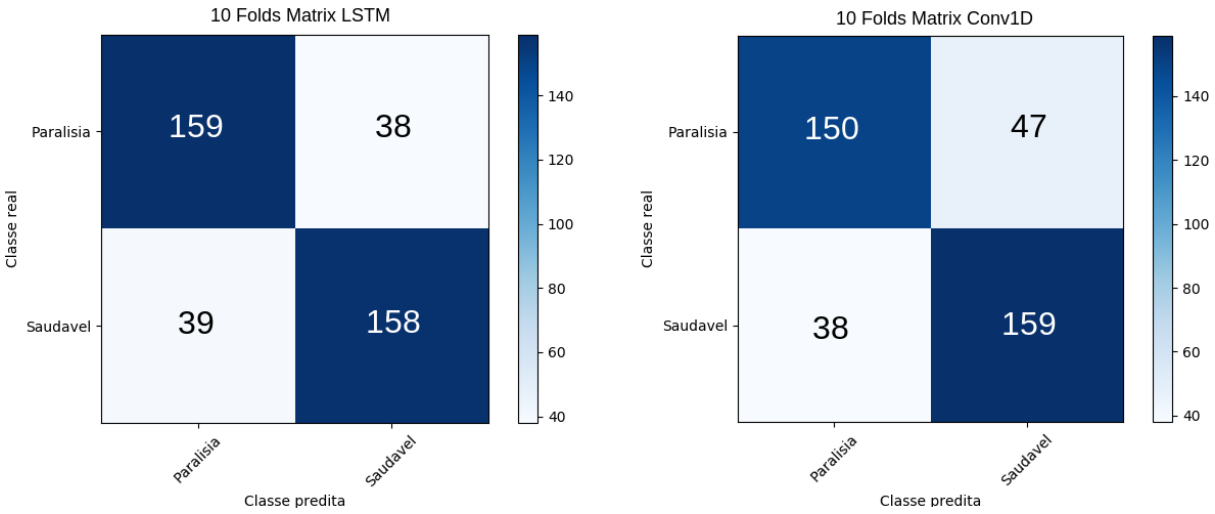

Figura 5.16: Matriz Global Paralisia x Saudável dos modelos LSTM e Conv1D *transfer learning*

Patológico x Saudável	Precição	Medida-F	Sensibilidade	Especificidade
-	%	%	%	%
LSTM	**78,0**	**78,0**	**78,0**	**79,0**
Conv1D	75,0	75,0	75,0	79,0

Tabela 5.19: Resultados Patológico x Saudável 10 *folds* com *transfer learning*

incorretamente, contra 99. Estas relações podem ser vistas na Figura 5.17.

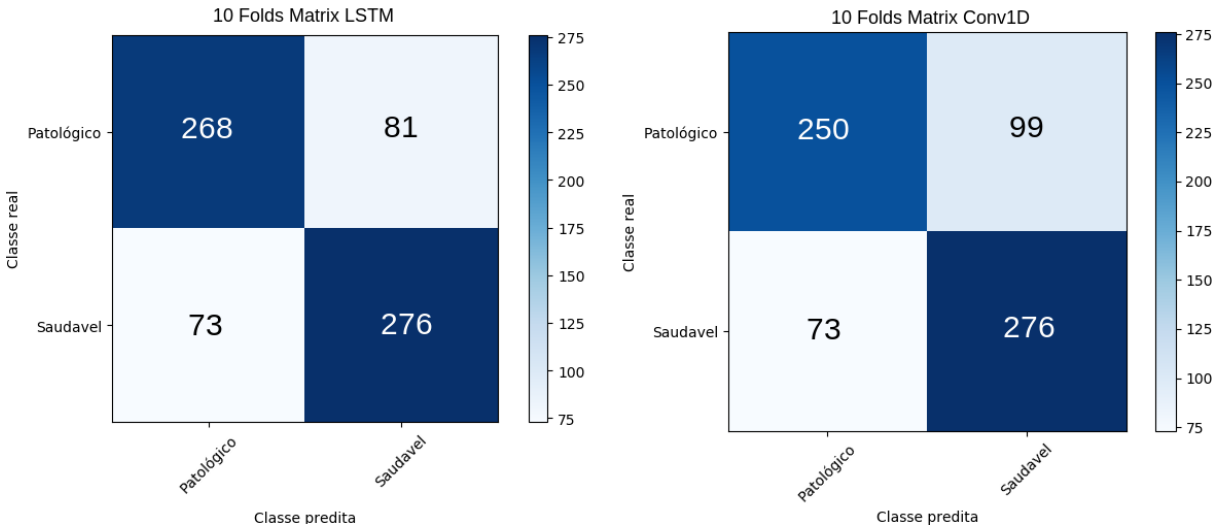

Figura 5.17: Matriz Global Patológico x Saudável dos modelos LSTM e Conv1D *transfer learning*

Para a classificação multi classe em validação cruzada em 10 *folds* é possível visualizar na Tabela 5.20 que existe uma leve vantagem para o modelo de Conv1D, com 41,0% de sensibilidade e 40,0% de medida-F. Entretanto, o modelo LSTM obteve resultados de 40,0% e 39,0% de sensibilidade e medida-F, respectivamente, o que pode é muito próximo dado.

4 classes 10 Folds	Precisão	Medida-F	Sensibilidade
-	%	%	%
LSTM	39,0	39,0	40,0
Conv1D	**40,0**	**40,0**	**41,0**

Tabela 5.20: Resultados 4 classes com 10 *folds* e *transfer learning*

Apesar dos resultados, as matrizes globas das Figuras 5.18 e 5.19 mostram que nas diagonais principais existe um aprendizado, com uma leve vantagem para o modelo Conv1D, entretanto ainda existe uma certa confusão entre as doenças patológicas. O pior caso para disfonia, onde o número de instâncias classificadas incorretamente é muito alto. O segundo pior caso é o de Laringite Crônica, cuja classificação também apresenta muitos erros, principalmente em Saudáveis e Paralisia. A classe dos saudáveis foi a que mais teve

instancias classificadas corretamente, entretanto houve bastante confusão com a classe laringite.

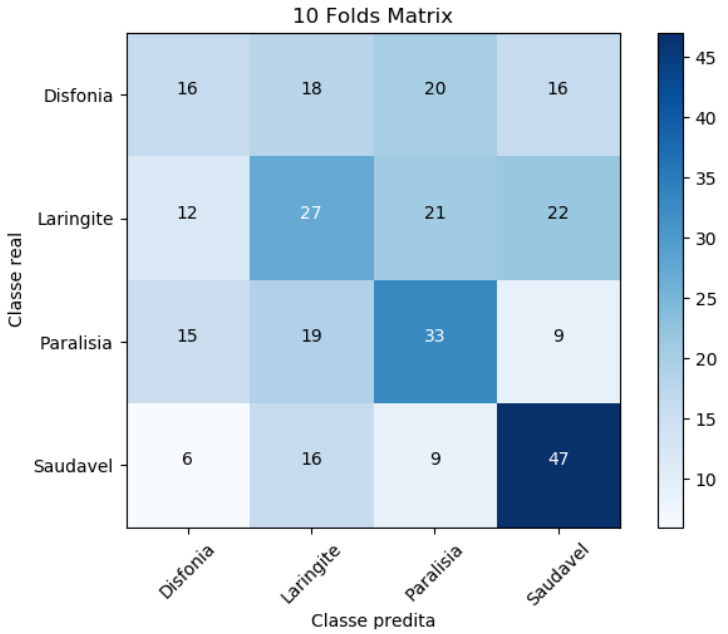

Figura 5.18: Matriz Global 4 classes do modelo LSTM com *transfer learning*

Em relação ao desempenho da rede nos treinamentos, a Figura 5.20 mostra a acurácia da validação dos 10 *folds* da Conv1D. É possível visualizar que existiram casos que obteve-se acurácia próximos dos 55%, dentre 40%. Considerando a quantidade de dados para um problema multi classe, esse resultado é relativamente bom, considerando que um classificador aleatório é aproximadamente de 25%.

Nas figuras 5.21 e 5.22 é mostrado o erro do treino e validação, respectivamente, dos 10 *folds* da Conv1D. Como foi feita a condição de parada analisando o erro da validação (*Early stopping*) as linhas dos gráficos não são do mesmo tamanho. Os modelos em validação apresentam casos de erro com valores bem menores e outros maiores, com destaque a linha azul escura que foi o modelo (*fold*) com menor erro dentre os dez. O bom é que com esta condição de parada quando começa a ter a presença de *overfitting* no modelo é interrompido o treinamento.

Estes resultados sugerem que a metodologia com a utilização de *transfer learning* realmente é válida. Entretanto, como a base de dados é consideravelmente pequena, a

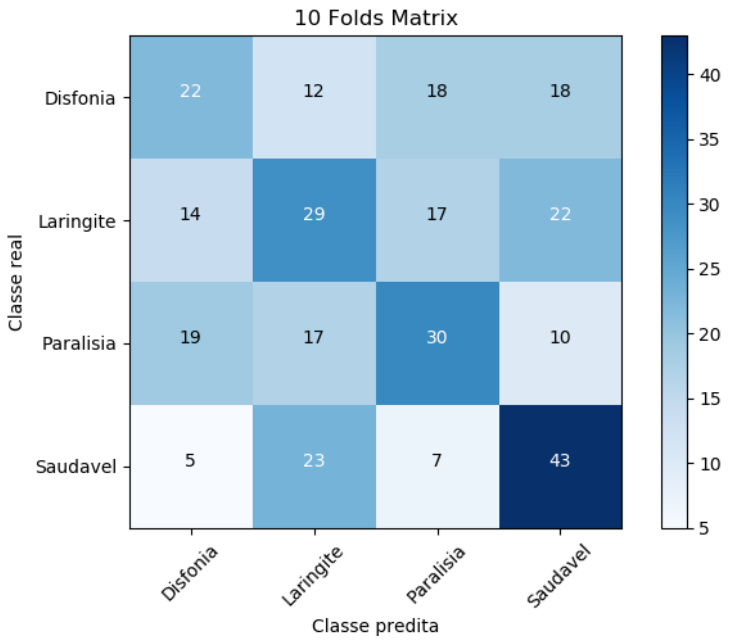

Figura 5.19: Matriz Global 4 classes do modelo Conv1D com *transfer learning*

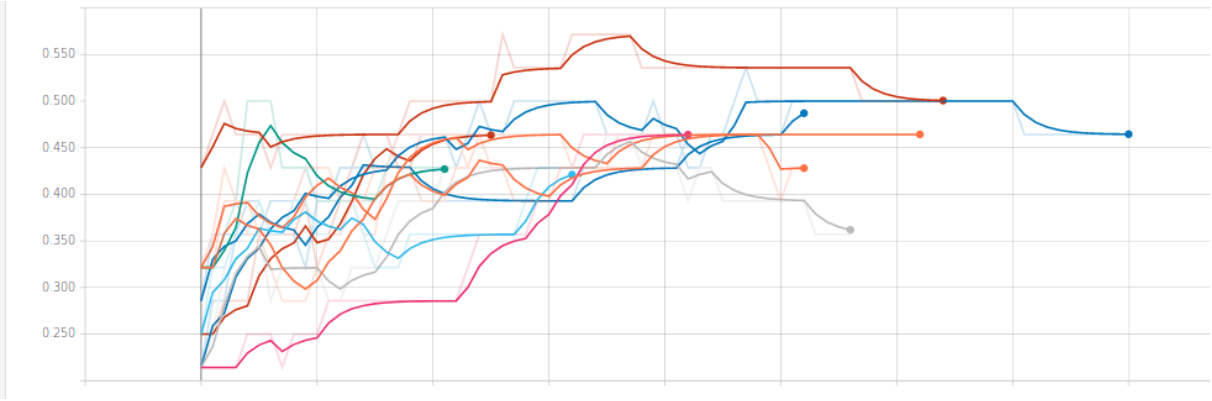

Figura 5.20: Acurácia da validação de 10 *folds* da Conv1D

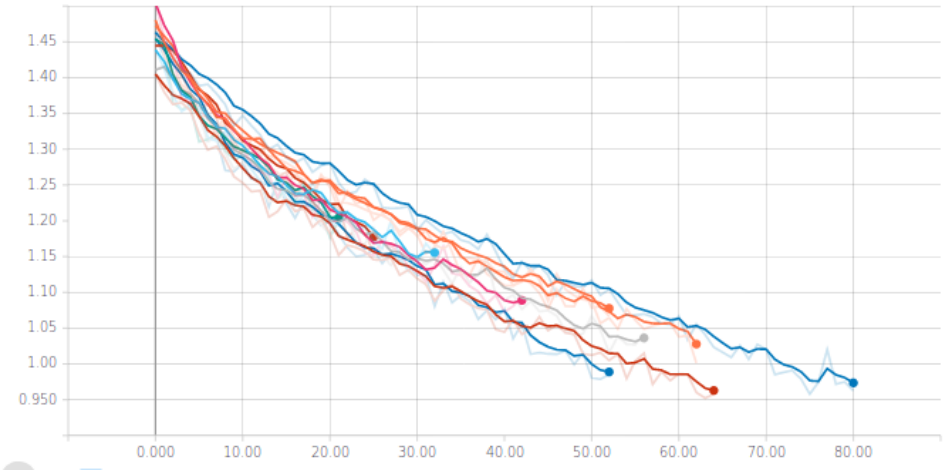

Figura 5.21: Erro do treino dos 10 *folds* da Conv1D

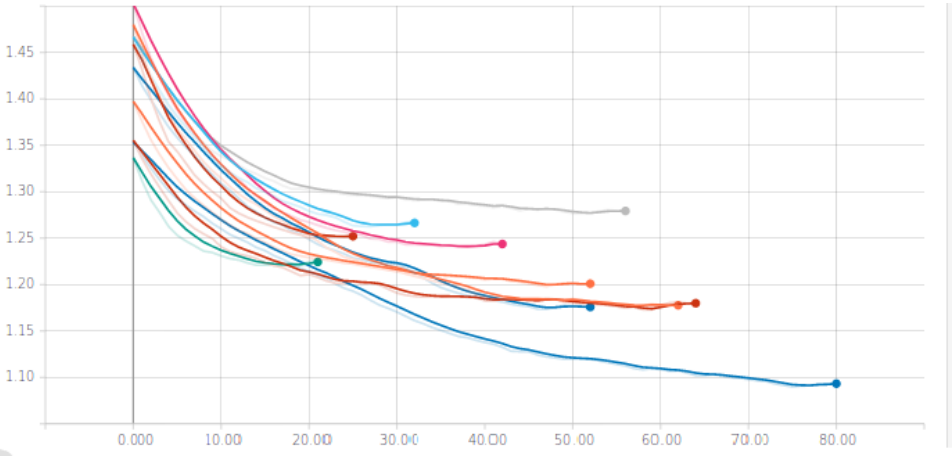

Figura 5.22: Erro da validação dos 10 *folds* da Conv1D

classificação se deu bem em casos binários. Para o caso multi-classe, ainda seria preciso uma variabilidade muito maior de dados para que se tenha resultados mais satisfatórios.

Como a classificação para a quatro classes não obteve resultados satisfatórios, foi realizado os respectivos experimentos para três classes com os modelos LSTM e Conv1D: Disfonia x Laringite x Saudáveis; Laringite x Paralisia x Saudáveis; Disfonia x Paralisia x Saudáveis; Disfonia x Laringite x Paralisia. A quantidade de instancias para cada classe é de 70 em disfonia, 82 em laringite, 78 em paralisia, 76 em saudáveis. A Tabela 5.21 mostra os resultados. É possível visualizar que a melhor classificação foi para laringite x paralisia x saudável com 53,0%.

3 classes	Modelo	Precisão %	Medida-F %	Sensibilidade %
Dis-Lar-Sau	LSTM	48,0	49,0	48,0
	Conv1D	**51,0**	**49,0**	**50,0**
Lar-Par-Sau	**LSTM**	**53,0**	**53,0**	**53,0**
	Conv1D	52,0	53,0	52,0
Dis-Par-Sau	LSTM	49,0	50,0	48,0
	Conv1D	**48,0**	**49,0**	**49,0**
Dir-Lar-Par	LSTM	38,0	40,0	39,0
	Conv1D	**40,0**	**41,0**	**41,0**

Tabela 5.21: Resultados 3 classes com 10 *folds* e *transfer learning*

5.4 Comparação dos resultados

A comparação dos resultados para as vogais utilizando a base de [19] e algoritmos de [17] [20] mostra uma importante informação de que com o mesmo modelo LSTM 4 não foi possível atingir os 99% de sensibilidade, isso pode ser justificado devido às diferenças das bases, onde na base de [19] é utilizada pacientes diagnosticados com laringite crônica mas que também apresentam outras patologias, o que mostra uma diferença significativa.

A comparação dos resultados para os métodos relacionados à frase e à classificação binária entre disfonia e saudável pode ser vista na Tabela 5.22. Nesta tabela, é possível observar que dentre os experimentos, o que obteve o melhor desempenho foi a RNA clássica utilizando MFCCs com 76,0% de medida-F, seguido pelos dois métodos de *Transfer*

Learning.

Disfonia x Saudável	Medida-F
Experimento	%
RNA MFCC	**76,0**
LSTM MFCC	55,0
Conv1D MFCC	49,0
LSTM Transfer	63,0
Conv1D Transfer	66,0

Tabela 5.22: Comparação dos resultados das frases para disfonia x saudável

A comparação dos resultados para os métodos relacionados à frase e à classificação binária entre laringite e saudável pode ser vista na Tabela 5.23. Nesta é possível observar que dentre os experimentos o que obteve o melhor desempenho foi novamente a RNA clássica utilizando MFCCs com 68,0% de Medida-F, entretanto o modelo Conv1D utilizando *Transfer Learning* obteve 67,0%.

Laringite x Saudável	Medida-F
Experimento	%
RNA MFCC	**68,0**
LSTM MFCC	49,0
Conv1D MFCC	57,0
LSTM Transfer	66,0
Conv1D Transfer	67,0

Tabela 5.23: Comparação dos resultados das frases para laringite x saudável

A comparação dos resultados para os métodos relacionados à frase e à classificação binária entre paralisia e saudável pode ser vista na Tabela 5.24. Nesta o melhor modelo foi o LSTM utilizando *Transfer Learning* obtendo 80,0% de Medida-F, entretanto o modelo RNA clássico usando MFCCs obteve 77,0%, ficando em terceiro lugar.

Para a classificação entre patológico e saudável unindo as três classes de patologia em uma só 5.30, houve uma melhora quando utilizado *Transfer Learning* e o modelo LSTM Transfer, entretanto a RNA com MFCC também apresenta resultado próximo.

Para a classificação em quatro classes das frases é possível observar na Tabela 5.26 que houve praticamente um empate entre os modelos, com uma leve vantagem para o modelo

Paralisia x Saudável	Medida-F
Experimento	%
RNA MFCC	77,0
LSTM MFCC	63,0
Conv1D MFCC	74,0
LSTM Transfer	**80,0**
Conv1D Transfer	78,0

Tabela 5.24: Comparação dos resultados das frases para paralisia x saudável

Patologico x Saudável	Medida-F
Experimento	%
RNA MFCC	72,0
LSTM MFCC	58,0
Conv1D MFCC	60,0
LSTM Transfer	**78,0**
Conv1D Transfer	75,0

Tabela 5.25: Comparação dos resultados das frases para patológico x saudável

Conv1D utilizando *Transfer Learning*, obtendo 40,0% de Medida-F.

4 classes	Medida-F
Experimento	%
RNA MFCC	38,0
LSTM Transfer	39,0
Conv1D Transfer	**40,0**

Tabela 5.26: Comparação dos resultados das frases para 4 classes

Com isso, é possível afirmar que para os experimentos binários a influência da utilização de *Transfer Learning* não foi expressivo comparado ao modelo RNA clássica, entretanto se comparado aos modelos LSTM e Conv1D sempre houve melhoras nestes modelos, o que sugere que o treinamento do zero para modelos considerados *Deep Learning* exigem uma quantidade de dados maior.

Entretanto o desempate acontece quando comparada a classificação entre as três classes Laringite x Paralisia x Saudável, onde a RNA clássica teve uma leve vantagem com 59,0%. Isso pode ser justificado, pois, mesmo com o uso do *Transfer Learning*, a quantidade de instâncias para cada classe ainda não é suficiente para os modelos LSTM e Conv1D. Os

resultados são vistos na Tabela 5.27.

3 classes	Medida-F
Lar-Par-Sau	%
RNA MFCC	**59,0**
LSTM Transfer	53,0
Conv1D Transfer	53,0

Tabela 5.27: Comparação dos resultados das frases para 3 classes

Fazendo o comparativo entre as metodologias das vogais e das frases pode-se concluir que as frases apresentaram resultados melhores que as vogais, como mostra a Tabela 5.28. Entretanto, o trabalho de [53], mostra que o potencial das vogais pode ser melhor explorado. Neste caso é utilizado a Sensibilidade para comparação.

Experimentos	Sensibilidade
Vogais	%
Disfonia x Saudável LSTM 4	59,0
Laringite x Saudável LSTM 4	63,8
Paralisia x Saudável LSTM 4	68,0
Frase	%
Disfonia x Saudável RNA MFCC	**76,0**
Laringite x Saudável RNA MFCC	**68,0**
Paralisia x Saudável Transfer	**80,0**

Tabela 5.28: Comparação dos resultados das entre vogais e frases

5.4.1 Comparação dos resultados com trabalhos relacionados

A Tabela 5.29 mostra dados avaliados em sensibilidade e acurácia, é demostrado a relação dos experimentos das vogais desenvolvidos nesta dissertação e os trabalhos relacionados para classificação binária entre patológico x saudável. Como pode ser visto a maioria dos trabalhos desenvolvidos utiliza somente a vogal / a / para fazer a sua classificação, incluindo o trabalho desenvolvido por [11] e que deu origem ao desafio FEMH [1]. Neste são utilizados MFCCs como parâmetros. Também é valido lembrar que apesar da base da tabela ser praticamente a mesma, as patologias de outras abordagens são diferentes.

[1]Site do desafio: https://femh-challenge2018.weebly.com/

As doenças utilizadas pelos trabalhos relacionados e por essa dissertação são:

- **DNN [11]**: Nódulos vocais, Pólipos, Cistos, Neoplasia glótica, Atrofia vocal, Distonia laríngea, Disfonia espasmódica e tremor vocal;
- **SVM [10]**: 71 doenças da base SVD (todas).
- **CNN [12]**: Laringite, Leucoplasia, Edema de Reinke, Paralisia do nervo laríngeo recorrente, Carcinoma de prega vocal e Pólipos de pregas vocais;
- **CNN+LSTM [7]**: 71 doenças da base SVD (todas).
- **RNA [9]**: Somente Mulheres com Disfonia;
- **RNA [9]**: Somente Homens com Disfonia;
- **LSTM [53]**: Pessoas diagnosticas com Laringite Crônica e outras doenças. Disfonia, Edema de Reinke, Leucoplasia, Disfonia hiper funcional, Pólipo, Laringes paquidermia, Carcinoma na epiglote, Paralisia do nervo laríngeo e Carcinoma.
- **Paralisia x Saudável**: Somente pessoas com Paralisia das Cordas Vocais;
- **Disfonia x Saudável**: Somente pessoas com Disfonia;
- **Laringite x Saudável**: Somente pessoas com Laringite Crônica;

Entretanto, o trabalho desenvolvido em [53] com a base [19] mostra que com as características de *Jitter*, *Shimmer* e Autocorrelação também é possível de alcançar-se resultados próximos. O que chama a atenção é que somente nos caso paralisia x saudável extraídos com [17] [20] é que se obteve resultados que superaram outras abordagens. Nos outros casos, todos tiveram desempenho inferior, o que sugere que com a adição de mais doenças relacionas ou de pacientes com mais de uma doença pode haver melhora para disfonia e laringite.

Em relação a utilização de fala contínua para a classificação de patologias, é citado a tese de [5], onde são classificadas três classes para a palavra “rainbow” da base de dado

Patológico x Saudável	**Base**	**Vogal**	**Sensibilidade**	**Exatidão**
DNN \[11]	Autoral+MEEI	a	-	99,3
RNA \[9]	SVD	a,i,u	-	99,0
LSTM \[53]	**SVD**	**a,i,u**	**99,0**	**99,0**
RNA \[9]	SVD	a,i,u	-	90,0
SVM \[10]	SVD	a	87,6	87,6
CNN \[12]	SVD	a	74,0	71,0
Paralisia x Saudável	**SVD**	**a,i,u**	**68,0**	**67,8**
CNN+LSTM \[7]	SVD	a	66,7	68,1
Dis+Lar+Par x Saudável	**SVD**	**a,i,u**	**66,6**	**64,5**
Laringite x Saudável	**SVD**	**a,i,u**	**63,8**	**63,1**
Disfonia x Saudável	**SVD**	**a,i,u**	**59,0**	**58,9**

Tabela 5.29: Comparação dos resultados das vogais com trabalhos relacionados

MEEI, obtendo 84% com MFCCs. Comparado ao obtido nesta dissertação para a frase "Guten Morgen, wie geht es Ihnen?" é obtido apenas 40% na base alemã Saarbrücken Voice Database em 4 classes, e 59% em três classes. As patologias em destaque são totalmente diferentes, entretanto, tanto na tese de [5] quanto nesta dissertação, foi possível obter resultados melhores para fala contínua do que para as vogais como pode ser visto na Tabela 5.28. Além disso, se comparado o resultado da frase alemã com os trabalhos relacionados das vogais é obtido os resultados da Tabela 5.30, onde o que obteve o melhor desempenho foi a de Paralisia contra os saudáveis.

Patológico x Saudável	Base	Vogal	Sensibilidade	Exatidão	Medida-F
DNN \[11]	Autoral+MEEI	a	-	99,3	-
RNA \[9]	SVD	a,i,u	-	99,0	-
RNA \[9]	SVD	a,i,u	-	90,0	-
SVM \[10]	SVD	a	87,6	87,6	-
Par x Sau	**SVD**	**Frase**	**80,0**	**-**	**80,0**
Pat x Sau	**SVD**	**Frase**	**78,0**	**-**	**78,0**
Dis x Sau	**SVD**	**Frase**	**76,0**	**-**	**76,0**
CNN \[12]	SVD	a	74,0	71,0	-
SVM \[13]	SVD	Frase	-	71,0	-
Lar x Sau	**SVD**	**Frase**	**68,0**	**-**	**68,0**
CNN+LSTM \[7]	SVD	a	66,7	68,1	-

Tabela 5.30: Comparação dos resultados das frases com trabalhos relacionados

O estudo multi-classe para detecção de patologias é relativamente recente, sendo a tese de doutorado de [5] o trabalho mais antigo (2016) levantado. No ano de 2018, com o desafio de FEMH, foram desenvolvidas novas abordagens [14], [15], [16] para classificação em quatro patologias, entretanto todas utilizando a vogal / a /, com destaque para a abordagem de [16] que obteve o segundo lugar na competição.

Capítulo 6

Conclusões e trabalhos futuros

6.1 Conclusões

O foco desta dissertação foi o estudo das abordagens com a utilização dos parâmetros MFCCs e espectrogramas aplicadas a frase para a classificação de patologias da voz, o que levou a resultados superiores aos das vogais.

O modelo que apresentou os melhores resultados nos experimentos para a frase é a RNA clássica em comparativo com os modelos da abordagem utilizando *Transfer learning*, pois esta, além de apresentar uma arquitetura simples, não demanda um grande processamento para sua utilização em sistemas embarcados. Nos experimentos de disfonia, laringite e classificação de três classes, o modelo RNA clássica se saiu melhor, estando muito próximo do melhor modelo nos casos de paralisia e a classificação das quatro classes. Isso sugere que, com o aumente do número de instâncias nas patologias, a RNA clássica ainda se comportaria bem para tal tarefa.

Apesar disso, não se pode ignorar a abordagem utilizando *Transfer learning*, pois esta mostra que realmente os modelos LSTM e Conv1D que por natureza exigem uma grande quantidade de dados, conseguiram comparar-se à RNA clássica.

Como foi dito antes, o ideal seria utilizar os conceitos de *Transfer learning* numa base de patologias pré-treinada, entretanto a base disposta nesta dissertação não possibilitou

esta abordagem.

Assim, a utilização de fala contínua para a classificação de doenças patológicas mostrou-se viável, necessitando apenas de mais dados para criar modelos finais, e as metodologias desenvolvidas nesta dissertação tornam-se promissoras para futuras implementações.

Outro ponto importante é o estudo com as vogais, no qual foi verificado que a utilização das classes de disfonia e laringite com pacientes com apenas uma doença não obtém resultados tão satisfatórios em comparativo com o estudo feito em [53], onde há a presença de pacientes com mais doenças relacionadas.

Nesta dissertação foi possível a publicação do artigo "Long Short Term Memory on Chronic Laryngitis Classification" [53] para o evento HCist (International Conference on Health and Social Care Information Systems and Technologies) de 2018, a publicação de "Classification of Control/Pathologic Subjects with Support Vector Machines" [13] também para o HCist, e "Harmonic to Noise Ratio Measurement - Selection of Window and Length"[57] estes como coautor, e a abordagem desenvolvida com a utilização do *Transfer learning* com a base AudioSet da Google foi submetida para o HCist de 2019.

6.2 Trabalhos futuros

Como trabalhos futuros é sugerida a criação de um software *Web* ou *Mobile* para aquisição de novos áudios catalogados, com frases em diferentes idiomas, associando médicos através de autenticação na aplicação para garantir a confiabilidade do que será gravado. Além disso, é importante que seja *Web* ou *Mobile* para que tais gravações dos áudios estejam em ambientes não controlados, apresentando ruído e tornando-os mais próximos da realidade.

Enquanto isso, uma solução para a melhora dos modelos é o agrupamento de novas doenças relacionadas aos grupos já existentes, ou seja, mais doenças relacionadas a laringite, disfonia e paralisia das cordas vocais ou pacientes com mais de uma doença, criando uma maior variabilidade nos dados. Com esses grupos mais abrangentes, caberia a possibilidade da criação de um *Transfer learning* na própria base para classificação de doenças específicas. Também poderia ser feito um estudo para remoção de *outliers* e a utilização

de outras características, como o HNR e os deltas MFCCs.

Além disso, é sugerido o estudo de viabilidade para o conceito de *One-shot learning* na base de dados *Saarbruecken Voice Database*. Esta abordagem utiliza do conceito de aprendizado por similaridade baseado em como os humanos aprendem e é bastante utilizada para *datasets* pequenos.

Bibliografia

[1] W. Ferreira Netto, *Introdução à fonologia da língua portuguesa*. jan. de 2011, ISBN: 9788599829394. DOI: 10.4322/978-85-99829-39-4.

[2] J. Teixeira e P. Fernandes, "Acoustic Analysis of Vocal Dysphonia", *Procedia Computer Science*, vol. 64, pp. 466–473, dez. de 2015. DOI: 10.1016/j.procs.2015.08.544.

[3] F. Huche e A. Allali, *A Voz - Vol.3: Patologia Vocal de Origem Orgânica*. Artmed Editora, 2016, ISBN: 9788536317915.

[4] A. Costa Pereira da Silva, S. Sena Esteves, S. Vaz Freitas, T. Feliciano e C. Almeida e Sousa, "Paralisia das cordas vocais - 8 anos de experiência no Centro Hospitalar do Porto", *Revista Portuguesa de Otorrinolaringologia e Cirurgia de Cabeça e Pescoço*, vol. 54, n.º 3, pp. 169–173, Mai. de 2017. URL: https://www.journalsporl.com/index.php/sporl/article/view/363.

[5] H. T. Cordeiro, "Reconhecimento de Patologias da Voz usando Técnicas de Processamento da Fala", tese de doutoramento, Universidade Nova de Lisboa, 2016.

[6] M. E. e E. Infirmary, "Voice disorders database", *Kay Elemetrics Corp., Lincoln Park, NJ*, vol. (Version 1.03 cd-rom), 1994.

[7] P. Harar, J. B. Alonso-Hernandezy, J. Mekyska, Z. Galaz, R. Burget e Z. Smekal, "Voice Pathology Detection Using Deep Learning: a Preliminary Study", em *2017 International Conference and Workshop on Bioinspired Intelligence (IWOBI)*, jul. de 2017, pp. 1–4.

[8] W. J. Barry e M. Pützer, "Saarbrücken Voice Database", Institute of Phonetics, Univ. of Saarland. URL: http://www.stimmdatenbank.coli.uni-saarland.de/help_en.php4.

[9] J. P. Teixeira, P. O. Fernandes e N. Alves, "Vocal Acoustic Analysis – Classification of Dysphonic Voices with Artificial Neural Networks", *Procedia Computer Science*, vol. 121, pp. 19–26, 2017, CENTERIS 2017 - International Conference on ENTERprise Information Systems / ProjMAN 2017 - International Conference on Project MANagement / HCist 2017 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2017, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.11.004. URL: http://www.sciencedirect.com/science/article/pii/S1877050917321956.

[10] L. Verde, G. De Pietro e G. Sannino, "Voice Disorder Identification by using Machine Learning Techniques", *IEEE Access*, vol. PP, pp. 1–1, mar. de 2018. DOI: 10.1109/ACCESS.2018.2816338.

[11] S.-H. Fang, Y. Tsao, M.-J. Hsiao, J.-Y. Chen, Y.-H. Lai, F.-C. Lin e C.-T. Wang, "Detection of Pathological Voice Using Cepstrum Vectors: A Deep Learning Approach", *Journal of Voice*, 2018, ISSN: 0892-1997. DOI: https://doi.org/10.1016/j.jvoice.2018.02.003. URL: http://www.sciencedirect.com/science/article/pii/S089219971730509X.

[12] H. Wu, J. Soraghan, A. Lowit e G. Di-Caterina, "A Deep Learning Method for Pathological Voice Detection Using Convolutional Deep Belief Networks", set. de 2018, pp. 446–450. DOI: 10.21437/Interspeech.2018-1351.

[13] F. Teixeira, J. Fernandes, V. Guedes, A. Junior e J. P. Teixeira, "Classification of Control/Pathologic Subjects with Support Vector Machines", *Procedia Computer Science*, vol. 138, pp. 272–279, 2018, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.10.039.

[14] V. Gupta, "Voice Disorder Detection Using Long Short Term Memory (LSTM) Model", *CoRR*, vol. abs/1812.01779, 2018.

[15] T. Grzywalski, A. Maciaszek, A. Biniakowski, J. Orwat, S. Drgas, M. Piecuch, R. Belluzzo, K. Joachimiak, D. Niemiec, J. Ptaszynski e K. Szarzynski, *Parameterization of Sequence of MFCCs for DNN-based voice disorder detection*, dez. de 2018.

[16] M. Pishgar, F. Karim, S. Majumdar e H. Darabi, "Pathological Voice Classification Using Mel-Cepstrum Vectors and Support Vector Machine", *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5267–5271, 2018.

[17] J. P. Teixeira e A. Gonçalves, "Algorithm for Jitter and Shimmer Measurement in Pathologic Voices", *Procedia Computer Science*, vol. 100, pp. 271–279, 2016, International Conference on ENTERprise Information Systems/International Conference on Project MANagement/International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN / HCist 2016, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2016.09.155. URL: http://www.sciencedirect.com/science/article/pii/S1877050916323237.

[18] J. Teixeira e A. Gonçalves, "Accuracy of Jitter and Shimmer Measurements", *Procedia Technology*, vol. 16, pp. 1190–1199, dez. de 2014. DOI: 10.1016/j.protcy.2014.10.134.

[19] J. Fernandes, F. Teixeira, P. O. Fernandes e J. P. Teixeira, "Cured Database of Sustained Speech Parameters for Chronic Laryngitis Pathology", *Proceedings of 31 st IBIMA Conference, Milan*, 2018.

[20] J. F. T. Fernandes, "Determinação da Autocorrelação, HNR e NHR para Análise Acústica Vocal", tese de mestrado, Instituto Politécnico de Bragança, 2019.

[21] H. Kale e D. S. S. Limaye, "Autocorrelation of a sound signal", *IOSR Journal of Electrical and Electronics Engineering*, p. 4, 2014.

[22] P. Boersma, "Accurate short term analysis of the fundamental frequency and the harmonics to noise ratio of a sampled sound", em *IFA Proceedings 17*, 1993, pp. 97–110.

[23] J. Teixeira, "Modelização paramétrica de sinais para aplicação em sistemas de conversão texto-fala", tese de mestrado, FEUP, 1996.

[24] A. Mallawaarachchi, S. Ong, M. Chitre e E. Taylor, "Spectrogram denoising and automated extraction of the fundamental frequency variation of dolphin whistles", *The Journal of the Acoustical Society of America*, vol. 124, pp. 1159–70, set. de 2008. DOI: 10.1121/1.2945711.

[25] D. Y.-W. Liu, *The Short-Time Fourier Transform*, 2015.

[26] S. Gupta, J. Jaafar, W. F. Wan Ahmad e A. Bansal, "Feature Extraction Using Mfcc", *Signal and Image Processing : An International Journal (SIPIJ)*, vol. 4, pp. 101–108, ago. de 2013. DOI: 10.5121/sipij.2013.4408.

[27] S. Majeed, H. HUSAIN, S. Samad e T. Idbeaa, "Mel frequency cepstral coefficients (Mfcc) feature extraction enhancement in the application of speech recognition: A comparison study", *Journal of Theoretical and Applied Information Technology*, vol. 79, pp. 38–56, set. de 2015.

[28] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling", *Proc. 1st Int. Symposium Music Information Retrieval*, nov. de 2000.

[29] V. Tiwari, "MFCC and its applications in speaker recognition", *Int. J. Emerg. Technol.*, vol. 1, jan. de 2010.

[30] L. Muda, M. Begam e I. Elamvazuthi, "Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques", *J Comput*, vol. 2, mar. de 2010.

[31] V. Sharma, "Discrete and Continuous Mouse Motion Using Vocal and Non Vocal Characteristics of Human Voice.", *International Journal of Computer and Information Sciences*, vol. 4, pp. 826–837, jun. de 2013.

[32] S. J. Russell e P. Norvig, *Artificial Intelligence: A Modern Approach*, 2. 1995, vol. 9, pp. 215–218, ISBN: 9780131038059. DOI: 10.1016/0925-2312(95)90020-9. arXiv: 9809069v1 [arXiv:gr-qc]. URL: http://portal.acm.org/citation.cfm?id=773294.

[33] T. W. Rauber, "Redes neurais artificiais", *Universidade Federal do Espírito Santo*, 2005.

[34] S. Haykin, *Redes Neurais Princípios e práticas*, 2ª ed., A. E. S.A, ed. São Paulo, 2008.

[35] T. M. Mitchell, *Machine Learning*, 1ª ed. New York, NY, USA: McGraw-Hill, Inc., 1997, ISBN: 0070428077, 9780070428072.

[36] D. Yu e L. Deng, *Automatic Speech Recognition: A Deep Learning Approach*. Springer Publishing Company, Incorporated, 2014, ISBN: 1447157788, 9781447157786.

[37] I. Goodfellow, Y. Bengio e A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[38] S. Hochreiter e J. Schmidhuber, "Long Short-term Memory", *Neural computation*, vol. 9, pp. 1735–80, dez. de 1997. DOI: 10.1162/neco.1997.9.8.1735.

[39] E. Bezerra, "Introdução à Aprendizagem Profunda", em, November, Rio de Janeiro, 2016, cap. 3, p. 30.

[40] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever e R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[41] J. Guo, "BackPropagation Through Time", *Unpubl. ms., Harbin Institute of Technology*, n.º 1, pp. 1–6, 2013.

[42] M. Boden, "A guide to recurrent neural networks and backpropagation", *the Dallas project*, 2002.

[43] A. Graves, "Supervised Sequence Labelling with Recurrent Neural Networks", *Image Rochester NY*, p. 124, 2008, ISSN: 01406736. DOI: 10.1007/978-3-642-24797-2. arXiv: arXiv:1308.0850v1.

[44] Y. Lecun, L. Bottou, Y. Bengio e P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, n.º 11, pp. 2278–2324, nov. de 1998, ISSN: 0018-9219. DOI: 10.1109/5.726791.

[45] Y. Bengio, "Learning Deep Architectures for AI", *Foundations*, vol. 2, pp. 1–55, jan. de 2009. DOI: 10.1561/2200000006.

[46] Brilliant.org. (). Convolutional Neural Network, URL: https://brilliant.org/wiki/convolutional-neural-network/ (acedido em 02/05/2019).

[47] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. Weiss e K. Wilson, "CNN Architectures for Large-Scale Audio Classification", em *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. URL: https://arxiv.org/abs/1609.09430.

[48] L. Torrey e J. Shavlik, "Transfer learning", em *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, IGI Global, 2010, pp. 242–264.

[49] R. Mormont, P. Geurts e R. Marée, "Comparison of Deep Transfer Learning Strategies for Digital Pathology", em *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, jun. de 2018, pp. 2343–234 309.

[50] M. Hossin e S. M.N, "A Review on Evaluation Metrics for Data Classification Evaluations", *International Journal of Data Mining and Knowledge Management Process*, vol. 5, pp. 01–11, mar. de 2015. DOI: 10.5121/ijdkp.2015.5201.

[51] P. Refaeilzadeh, L. Tang e H. Liu, "Cross-Validation", em *Encyclopedia of Database Systems*, L. LIU e M. T. ÖZSU, eds. Boston, MA: Springer US, 2009, pp. 532–

538, ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_565. URL: https://doi.org/10.1007/978-0-387-39940-9_565.

[52] S. Ioffe e C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", em *ICML*, 2015.

[53] V. Guedes, A. Junior, J. Fernandes, F. Teixeira e J. P. Teixeira, "Long Short Term Memory on Chronic Laryngitis Classification", *Procedia Computer Science*, vol. 138, pp. 250–257, 2018. URL: http://www.sciencedirect.com/science/article/pii/S1877050918316697.

[54] D. Kingma e J. Ba, "Adam: a method for stochastic optimization (2014)", *arXiv preprint arXiv:1412.6980*, vol. 15, 2015.

[55] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever e R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *The Journal of Machine Learning Research*, vol. 15, n.º 1, pp. 1929–1958, 2014.

[56] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal e M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events", em *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[57] J. Fernandes, F. Teixeira, V. Guedes, A. Junior e J. P. Teixeira, "Harmonic to Noise Ratio Measurement - Selection of Window and Length", *Procedia Computer Science*, vol. 138, pp. 280–285, 2018, CENTERIS 2018 - International Conference on ENTERprise Information Systems / ProjMAN 2018 - International Conference on Project MANagement / HCist 2018 - International Conference on Health and Social Care Information Systems and Technologies, CENTERIS/ProjMAN/HCist 2018, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.10.040. URL: http://www.sciencedirect.com/science/article/pii/S1877050918316739.