UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

BAUKE ALFREDO DIJKSTRA

RECONHECIMENTO DE FONEMAS UTILIZANDO REDES NEURAIS CONVOLUCIONAIS PARA TRANSCRIÇÃO FONÉTICA AUTOMÁTICA

DISSERTAÇÃO

PONTA GROSSA
2021

BAUKE ALFREDO DIJKSTRA

RECONHECIMENTO DE FONEMAS UTILIZANDO REDES NEURAIS CONVOLUCIONAIS PARA TRANSCRIÇÃO FONÉTICA AUTOMÁTICA

Phoneme Recognition using Convolutional Neural Networks for Automatic Phonetic Transcription

Dissertação apresentada como requisito para obtenção do título de Mestre em Ciência da Computação da Universidade Tecnológica Federal do Paraná (UTFPR).
Orientador: Dr. Ionildo José Sanches

PONTA GROSSA

2021

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Ponta Grossa

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

BAUKE ALFREDO DIJKSTRA

RECONHECIMENTO DE FONEMAS UTILIZANDO REDES NEURAIS CONVOLUCIONAIS PARA TRANSCRIÇÃO FONÉTICA AUTOMÁTICA

Trabalho de pesquisa de mestrado apresentado como requisito para obtenção do título de Mestre Em Ciência Da Computação da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Sistemas E Métodos De Computação.

Data de aprovação: 28 de Janeiro de 2021

Prof Ionildo Jose Sanches, - Universidade Tecnológica Federal do Paraná

Prof Hugo Valadares Siqueira, Doutorado - Universidade Tecnológica Federal do Paraná

Prof.a Rosane Falate, Doutorado - Universidade Estadual de Ponta Grossa (Uepg)

Dedico esse trabalho aos meus pais, Lolke e Johanna.

AGRADECIMENTOS

Agradeço aos meus pais, Lolke e Johanna, aos meus irmãos Albert e Christiane e ainda a Helen por serem meus maiores incentivadores. Por estarem sempre ao meu lado, acreditando que sou capaz de alcançar meus objetivos.

A Deus, por me dar força e saúde para ir atrás dos meus sonhos e conseguir concluir esse desafio.

Ao Prof. Dr. Ionildo pela dedicação em me orientar neste trabalho e pelo aprendizado que me proporcionou.

Aos professores Dr. Hugo e Dr. Marcelo, por participarem da banca da qualificação e pelas valiosas sugestões apresentadas.

Aos amigos que fiz durante o mestrado. Aleffer, Eduardo, Everton, Fábia, Lin e Luís obrigado pela amizade, diversão e ajuda sempre que precisei durante esses anos.

Aos professores do PPGCC, pelos ensinamentos oferecidos durante o curso.

Também gostaria de agradecer a Universidade Tecnológica Federal do Paraná (UTFPR) pelo apoio financeiro, que tornou possível o desenvolvimento desta pesquisa.

RESUMO

DIJKSTRA, Bauke Alfredo. *Reconhecimento de Fonemas utilizando Redes Neurais Convolucionais para Transcrição Fonética Automática.* 2021. 75 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Tecnológica Federal do Paraná. Ponta Grossa, 2021.

O reconhecimento de fonemas é a capacidade de extrair características para reconhecer as unidades sonoras das palavras e transcrevê-las. As aplicações do reconhecimento de fonemas são auxiliares no reconhecimento de fala, identificação de locutores, identificação de erros de pronúncia e reconhecimento de emoções. Para realizar esta tarefa aplica-se inicialmente uma etapa de pré-processamento nos áudios, denominado processamento acústico, que permite extrair as características, minimizar ruídos e as diferenças entre locutores. Em seguida, é realizada uma etapa de treinamento e classificação, utilizando algoritmos de aprendizagem de máquina com o objetivo de identificar os fonemas. Este trabalho tem como objetivo desenvolver uma técnica de reconhecimento automático de fonemas de fala contínua. No desenvolvimento, o treinamento e os testes foram realizados com dados extraídos das bases de áudios TIMIT *Acoustic-Phonetic Continuous Speech Corpus* que possui fala em inglês e possui transcrições ortográficas, fonéticas e de palavras alinhadas com o tempo, e as bases com fala em português brasileiro Sid e LaPS Benchmark 16k. As bases na língua portuguesa do Brasil são apenas transcritas na forma ortográfica, portanto, tornou-se necessário adicionar a transcrição fonética em relação aos áudios. Para isso, utilizou-se o software Praat com o *plugin* EasyAlign e foi desenvolvido um *script* para formatar as saídas do programa, alinhando os fonemas no tempo em relação aos *frames*. No processamento acústico, para extrair os coeficientes cepstrais de frequência de Mel (MFCC) e os *filter banks*, utilizou-se o *Kaldi Speech Recognition Toolkit*. Para o treinamento e classificação, das bases citadas, foi implementado uma rede neural convolucional juntamente com uma rede de memória de longo e curto prazo usando o *framework* Pytorch. O resultado obtido na base TIMIT apresentou uma taxa de erro de fonemas no *core test* de $18,11\%$ utilizando *filter banks* e uma taxa de erro de $19,04\%$ usando MFCCs. Na união das bases em português LaPS Benchmark 16k e Sid, obteve-se uma taxa de erro de $24,96\%$ usando *filter banks* e $25,54\%$ usando MFCC nos conjuntos de testes.

Palavras-chaves: Reconhecimento de fonemas. Processamento acústico. Aprendizagem de máquina. Rede neural convolucional.

ABSTRACT

DIJKSTRA, Bauke Alfredo. *Phoneme Recognition using Convolutional Neural Networks for Automatic Phonetic Transcription.* 2021. 75 p. Dissertation (Master's Degree in Computer Science), Federal University of Technology - Paraná. Ponta Grossa, 2021.

Phoneme recognition is the ability to extract features to recognize the sound units of words and transcribe them. Phoneme recognition applications provides assistance towards automatic speech recognition, speaker identification, pronunciation error identification, and emotion recognition. In order to carry out the recognition of the phonemes, a pre-processing stage is initially applied in the audios, called acoustic processing, which allows the characteristics to be extracted, noise and differences between speakers to be reduced. Then, a training and classification stage using machine learning algorithms in order to identify the phonemes. This work aims to develop a technique for automatic recognition of continuous speech phonemes. In the development of this project the tests were performed with phonemes extracted from the audio datasets such as TIMIT Acoustic-Phonetic Continuous Speech Corpus, which is an English-speaking dataset with time-aligned orthographic, phonetic and word transcripts, and Brazilian Portuguese-speaking datsets such as Sid and LaPS Benchmark 16k.The Brazillian Portuguese datasets are only transcribed in orthographic form, so it was required to make changes in these datasets to form the phonemes in regards to the audio recordings.The Praat software was used along with the EasyAlign plugin and a script was developed to format the program's outputs, aligning the phonemes in time with the frames. In acoustic processing, the Kaldi Speech Recognition Toolkit was applied to extract the MFCC and filter banks. For the training and classification, of the bases cited, a convolutional neural network was implemented in addition to a long short-term memory network using the Pytorch framework. The result obtained on the TIMIT base presented a phoneme error rate in the core test of 18,11% using filter banks and an error rate of 19,04% using MFCC. On the merged Portuguese bases LaPS Benchmark 16k and Sid, an error rate of 24,96% was obtained using filter banks and 25,54% using MFCC.

Keywords: Phoneme recognition. Acoustic processing. Machine learning. Convolutional neural network.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ASR	*Automatic Speech Recognition*
CAPES	Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CNN	*Convolutional Neural Network*
CPU	*Central Processing Unit*
CTC	*Connectionist temporal classification*
CUDA	*Compute Unified Device Architecture*
dB	Decibel
DCT	*Discrete Cosine Transform*
DFT	*Discrete Fourier Transform*
DNN	*Deep Neural Network*
FD	Filtro de Descarte
FFT	*Fast Fourier Transform*
FI	Filtro de Inclusão
GDDR	Graphics Double Data Rate
GMM	*Gaussian Mixture Models*
GPU	*Graphics Processing Unit*
HMM	*Hidden Markov Model*
Hz	Hertz
IA	Inteligência Artificial
JCR	*Journal Citation Reports*
KNN	*K-nearest Neighbor*
LPC	*Linear Predictive Coefficients*
LSTM	*Long-Short Term Memmory*
MFCC	*Mel-Frequency Cepstral Coefficients*
MLP	*Multilayer Perceptron*
PAS	Processamento Acústico do Sinal
PER	*Phoneme Error Rate*
PETRUS	*PhonEtic TRanscriber for User Support*

PLP	*Perceptual Linear Prediction*
SAMPA	*Speech Assessment Methods Phonetic Alphabet*
SDI	Sinal Digital Isolado
SVM	*Support Vector Machine*

SUMÁRIO

1 INTRODUÇÃO

O reconhecimento de fonemas consiste em identificar os fonemas de sinais digitais, permitindo realizar as transcrições fonéticas dos áudios. Essa tarefa pode ser executada utilizando conceitos de aprendizagem de máquina, processamento acústico do sinal digital e características linguísticas. Portanto, o reconhecimento de fonemas é um conjunto de três grandes áreas de estudos: ciência da computação responsável pelas técnicas de treinamento e testes de redes neurais artificias; engenharia elétrica para o processamento dos sinais digitais dos áudios; e linguística para análise dos dados assim como a utilização do reconhecimento de fonemas para auxiliar outras pesquisas.

As transcrições fonéticas podem ser obtidas de três formas: manual, semiautomática ou automática. As transcrições fonéticas manuais dependem exclusivamente do esforço e empenho de transcritores humanos, os procedimentos semiautomáticos e automáticos requerem o uso de um sistema computacional de transcrição. Considera-se transcrição semiautomática quando é realizado uma verificação e correção manual por transcritores humanos e a automática quando é tomado como correto utilizando exatamente o resultado obtido pelo sistema computacional (SERRANI, 2015).

Existem três modelos principais de reconhecimento de fala de áudios: palavras isoladas, palavras concatenadas e fala contínua. O primeiro caso trata do reconhecimento de palavras como pequenas unidades ou padrões simples. Normalmente quando se utiliza uma base de dados para realizar o reconhecimento da palavra, irá existir amostras para cada palavra que o sistema reconhece; o segundo caso se dá por uma sequência de palavras previamente especificadas, formando um vocabulário específico para uma base de dados. A diferença em relação a palavra isolada é que não é necessário que o sistema identifique pausas, podendo reconhecer, por exemplo uma sequência de código completa se falado de forma clara; o terceiro caso é uma tarefa mais complexa pois tem o objetivo de reconhecer qualquer frase de um locutor independente da velocidade de locução, tornando necessário o tratamento de unidades básicas da fala, como os fonemas ou sílabas (DINIZ; THOMé, 1997). Os modelos de fala contínua apresentam resultados bem inferiores a de palavras isoladas de acordo com Kshirsagar *et al.* (2012) portanto, ainda existem pesquisas em desenvolvimento para melhorar a acurácia desses sistemas.

Com isso pode-se observar que para auxiliar sistemas de reconhecimento automático de fala (ASR - *Automatic Speech Recognition*) pode-se utilizar o reconhecimento de fonemas. Outras aplicações são a identificação de dialetos e locutores, identificação de erros de pronúncia e reconhecimento de emoções (SEARA; NUNES; LAZZAROTTO-VOLCÃO, 2011). Essas aplicações se dão pelo fato que no sinal digital estão contidas múltiplas informações como fala, emoção e ruídos. Para o reconhecimento de fonemas é necessário realizar um processamento acústico com a intenção de remover a emoção e os

ruídos do sinal digital, para então realizar a tarefa.

De acordo com Deng, Yu *et al.* (2014) o reconhecimento de fala foi dominado pela técnica modelos de misturas de Gaussiana/modelo ocultos de Markov (GMM/HMM - *Gaussian Mixture Model/Hidden Markov Model*), mas a partir de 2010, técnicas de aprendizagem profunda (*deep learning*) começaram a ter um impacto no reconhecimento de fala de forma competitiva.

O trabalho visa o desenvolvimento de um sistema computacional de reconhecimento automático de fonemas em palavras contínuas aplicando o coeficientes cepstrais de frequência de Mel (MFCC - *Mel-Frequency Cepstral Coefficients*) e *filter banks* (banco de filtros) para processamento acústico e classificação usando redes neurais convolucionais juntamente com redes de memória de longo e curto prazo e utilizando uma camada de saída de classificação temporal conexionista. As bases utilizadas nesse trabalho são a TIMIT *Acoustic-Phonetic Continuous Speech Corpus* com áudios de fala na língua inglesa e a as bases LaPS Benchmark 16k, SID com áudios de falas em português brasileiro sendo que essas bases devem ser transcritas de forma fonética para se realizar o reconhecimento de fonemas.

1.1 JUSTIFICATIVA

Em uma atualidade motivada para o uso de várias aplicações tecnológicas ativadas pela voz, tanto a indústria quanto pesquisadores da área de tecnologia de fala têm se dedicado à pesquisas na área de transcrições de forma automática. Sistemas de transcrição fonética automática podem ser utilizados em muitas outras áreas tais como fonética, fonologia, ensino-aprendizagem de língua, e assim por diante (SERRANI, 2015).

Como não existem muitas bases de dados de áudio em português voltadas para a tarefa de reconhecimento de fonemas, torna-se mais difícil o desenvolvimento de novos modelos para serem utilizados em aplicações no português brasileiro. Com isso, foi identificada a necessidade de adicionar transcrições fonética a essas bases. Isto possibilita realizar pesquisas com maior facilidade para criar modelos de reconhecimento de fonemas para serem aplicados em sistemas ASR, sistemas de diagnóstico de fala, identificação locutores, entre outros.

Também é possível observar que muitos dos artigos relacionados apresentam taxas de erros de reconhecimentos automática de fonemas superiores a 18%, tais como: Ager, Cvetković e Sollich (2010) Lohrenz, Li e Fingscheidt (2018) e Seltzer e Droppo (2013) os quais apresentaram uma taxa de erro de reconhecimento de fonemas no *core test* da base TIMIT *Acoustic-Phonetic Continuous Speech Corpus* de 18,50%, 19,46% e 20,28% respectivamente. Assim, aplicações que necessitam do modelo como uma das etapas da aplicação, podem gerar uma propagação de erro, influenciando no resultado final. Outra

dificuldade do reconhecimento de fonemas se dá pelo fato de existirem fonemas com características muito similares entre eles (RABINER; JUANG, 1993).

1.2 OBJETIVO GERAL

Este trabalho tem como objetivo realizar o reconhecimento de fonemas em bases na língua inglesa e português do Brasil utilizando técnicas de aprendizagem profunda (*deep learning*), de tal modo que possam replicar esse trabalho para aplicações que necessitam de soluções neste campo como uma etapa de desenvolvimento.

1.2.1 Objetivos Específicos

Os objetivos específicos para o desenvolvimento desse trabalho são:

- Identificar bases de dados de fala existentes para o reconhecimento automático de fonemas;
- Adicionar transcrições fonéticas nas bases selecionadas em português brasileiro que possuem apenas os arquivos de áudio de fala com as transcrições ortográficas;
- Aplicar técnicas de processamento acústico tais como MFCC e *filter banks* para extração das características dos áudios;
- Implementar uma técnica de *deep learning* para o reconhecimento de fonemas utilizando redes neurais convolucionais;
- Realizar o treinamento do modelo de classificações das características dos áudios;
- Avaliar as taxas de erros obtidas com o modelo proposto e comparar com o estado da arte considerando a base TIMIT.

1.3 ORGANIZAÇÃO DO TRABALHO

Este documento está organizado em 5 capítulos. O Capítulo 2 apresenta a fundamentação teórica necessária para auxiliar o entendimento do trabalho, tais como fonética, processamento acústico, aprendizado de máquina e bases de dados de áudios de fala. O capítulo 3 apresenta o estado da arte e uma análise de trabalhos selecionados utilizando uma técnica de mapeamento sistemático sobre reconhecimento de fonemas. O Capítulo 4 apresenta os materiais e métodos para o desenvolvimento do trabalho. O Capítulo 5

apresenta os resultados e uma análise com a base TIMIT, a LaPS Benchmark 16k e a Sid. Por fim, o Capítulo 6 apresenta as conclusões do trabalho e sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma breve discussão sobre a fonética, dando ênfase aos detalhes da fonética acústica, pois é ela que realiza os estudos das ondas sonoras, possibilitando realizar uma transcrição fonética utilizando processamento digital.

Com isso, também são definidos processos para realizar um processamento acústico, uma explicação do que é uma janela de Hamming e o que são os coeficientes cepstrais de frequência de Mel utilizados para a extração das características do áudio.

Por fim tem-se uma seção para explicar sobre o que é o aprendizado de máquina, que tem como objetivo descobrir quais são os fonemas para realizar a transcrição fonética automática.

2.1 FONÉTICA

O estudo da fonética dá-se por base do aparelho fonador e os vários segmentos por ele produzido. São distribuídos em duas grandes classes de segmentos, sendo eles as vogais e as consoantes (SEARA; NUNES; LAZZAROTTO-VOLCÃO, 2011).

O aparelho fonador é composto por vários órgãos, como os pulmões, laringe, lábios, nariz, entre outros. Estes órgãos podem influenciar no som gerado na fala no modo acústico, mecânico e geométrico (SEARA; NUNES; LAZZAROTTO-VOLCÃO, 2011). A Figura 1 apresenta o aparelho fonador humano.

A unidade de estudo da fonologia é o fonema, estudo da Fonética é o fone. O termo "fone" é utilizado em Fonética para designar o menor segmento discreto perceptível de som em um fluxo da fala (continuum fônico ou substância fônica). Do ponto de vista da fonologia segmental, os fones são a realização física dos fonemas (CRYSTAL, 2011).

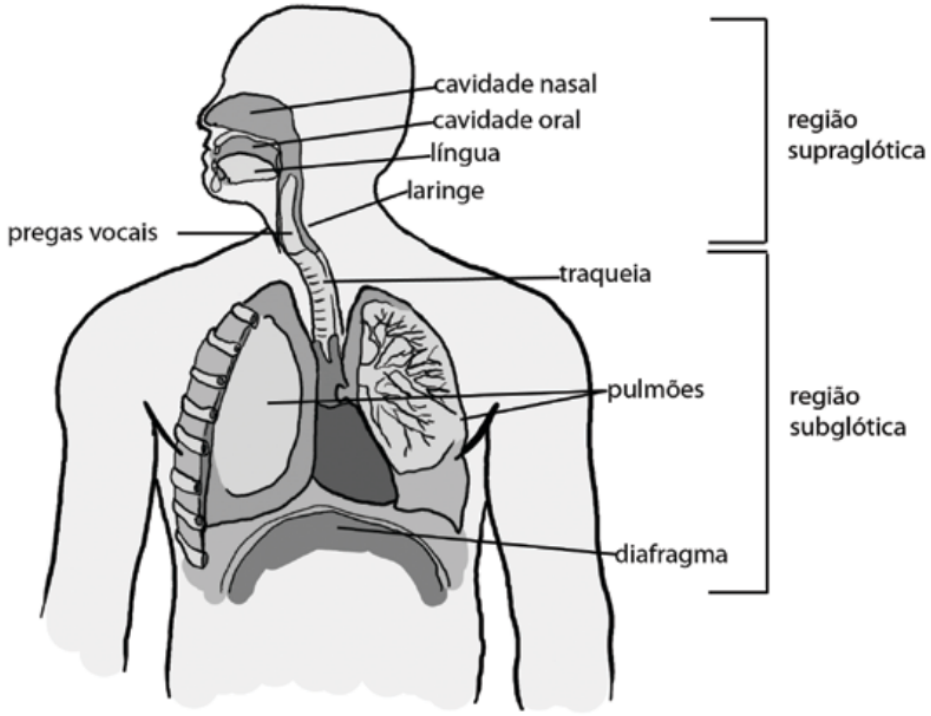
Como pode ser observado, existe uma divisão a partir da glote, sendo elas a região supraglótica, ou seja, acima da glote, é responsável pela criação das ressonâncias vocais e a região subglótica, em que se encontra os órgãos responsáveis pelo suprimento de energia para gerar os sons (SEARA; NUNES; LAZZAROTTO-VOLCÃO, 2011).

2.1.1 Segmentos Fonéticos

Existem dois segmentos fonéticos: as vogais e as consoantes. A diferença dá-se a partir da liberação do fluxo de ar dos pulmões.

As vogais são produzidas pelas pregas vocais quando não há nenhum impedimento

Figura 1 – Aparelho fonador



Fonte: (PARKER, 2007)

de passagem de ar. Elas ainda podem ser classificadas como vogais orais e nasais. Na produção das orais, o véu do palato fecha a passagem à cavidade nasal, fazendo com que o ar saia somente pelo trato oral. Nas vogais nasais, o véu palatino encontra-se abaixado, permitindo que o ar passe também pelas cavidades ressoadoras nasais (SEARA; NUNES; LAZZAROTTO-VOLCÃO, 2011).

Para a formação das consoantes devem ser realizadas obstruções no trato vocal, utilizando as características de ponto articulatório, modo articulatório e sonoridade. As consoantes ainda possuem duas classificações, sendo uma delas quando se utiliza das cordas vocais sendo as consoantes sonoras, e quando não utiliza das cordas vocais, conhecida como consoantes surdas ou não-vozeadas (SEARA; NUNES; LAZZAROTTO-VOLCÃO, 2011).

2.1.2 Transcrição Fonética

A transcrição fonética é a representação em forma de texto do som realizado por um falante para pronunciar as palavras em que é representada entre colchetes, por exemplo, a palavra "quilo" fica [’kilʊ] na transcrição ampla (SEARA; NUNES; LAZZAROTTO-VOLCÃO, 2011). Existem algumas variâncias de alfabetos fonéticos como o alfabeto fonético internacional.

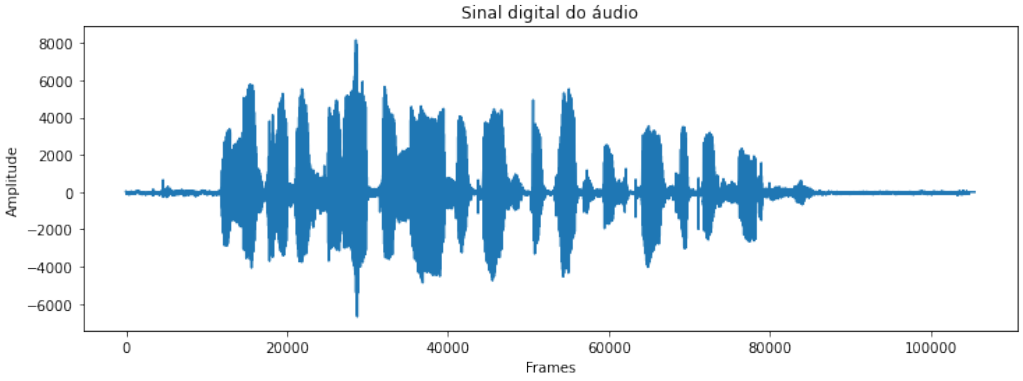
Outro alfabeto é o SAMPA (*Speech Assessment Methods Phonetic Alphabet*), que

é um alfabeto baseado no alfabeto fonético internacional com o objetivo de que uma máquina consiga ler, tendo variações para cada língua (WELLS *et al.*, 1997).

2.2 PROCESSAMENTO ACÚSTICO

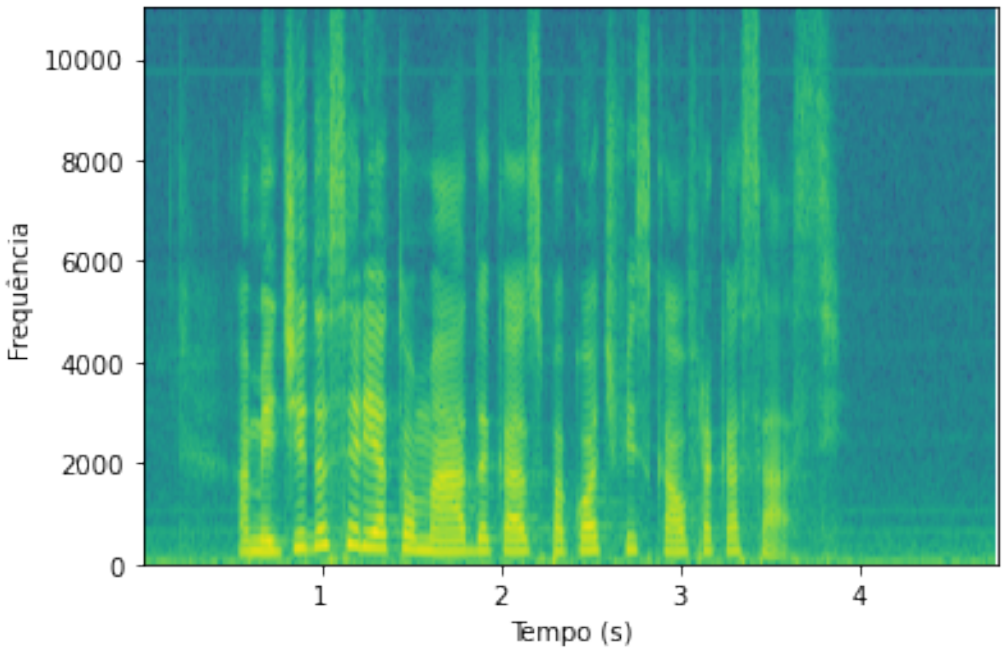
O processamento acústico ocorre em cima de um sinal digital, como mostrada na Figura 2, em que nesse caso possui informações da fala, ruídos e emoções do locutor da frase "A medida seria tomada caso o pacote fiscal fracassasse". A Figura 3 representa o espectrograma dessa frase que são gráficos que analisam dinamicamente a densidade espectral de energia, sendo que cada cor diferente representa intensidade diferente da densidade.

Figura 2 – Representação do sinal digital



Fonte: Autoria Própria.

Figura 3 – Representação do espectrograma



Fonte: Autoria Própria.

O processamento acústico tem o objetivo de reduzir os ruídos e conseguir extrair características dos áudios no domínio da frequência. Uma técnica muito utilizada é a extração dos coeficientes cepstrais de frequência de Mel (MFCC -*Mel-frequency Cepstral Coefficients*) (DAVE, 2013; KSHIRSAGAR *et al.*, 2012).

2.2.1 Filtros de janelamento

Os filtros em sinais têm como objetivo selecionar frequências e eliminar outras. Os filtros de janelamentos geralmente são usados em análise espectral e são baseados em um projeto de filtro de resposta ao impulso finita. Alguns exemplos de janelas são: Janela retangular representada pela Equação 1; Janela de Bartlett representada pela Equação 2 sendo que ela é triangular; Janela de Hann representada pela Equação 3; Janela Hamming representada pela Equação 4; Janela de Blackman representada pela Equação 5. Tal que M é o tamanho da janela, l é o local da janela para ser aplicada uma janela de duração finita $w[l]$ (OPPENHEIM; SCHAFER, 2012).

$$w[l] = \begin{cases} 1, & 0 \leqslant l \leqslant M, \\ 0, & \text{caso contrário} \end{cases} \tag{1}$$

$$w[l] = \begin{cases} \frac{2l}{M}, & 0 \leqslant l \leqslant \frac{M}{2}, \text{M par} \\ 2 - \frac{2l}{M}, & \frac{M}{2} \leqslant l \leqslant M, \\ 0, & \text{caso contrário} \end{cases} \tag{2}$$

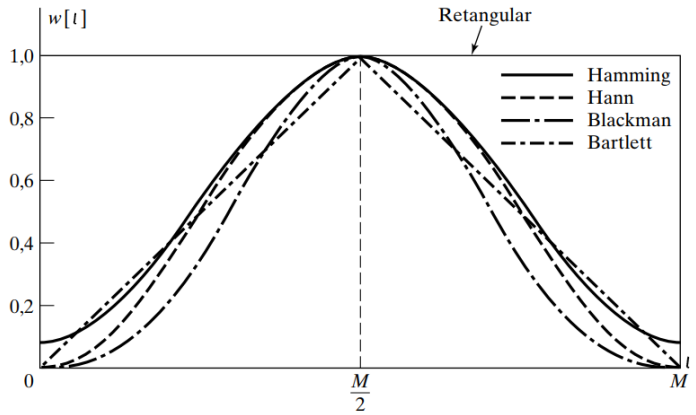
$$w[l] = \begin{cases} 0,5 - 0,5cos(\frac{2\pi}{M}), & 0 \leqslant l \leqslant M, \\ 0, & \text{caso contrário} \end{cases} \tag{3}$$

$$w[l] = \begin{cases} 0,54 - 0,46cos(\frac{2\pi}{M}), & 0 \leqslant l \leqslant M, \\ 0, & \text{caso contrário} \end{cases} \tag{4}$$

$$w[l] = \begin{cases} 0,42 - 0,5cos(\frac{2\pi}{M}) + 0,08cos(\frac{4\pi l}{M}), & 0 \leqslant l \leqslant M, \\ 0, & \text{caso contrário} \end{cases} \tag{5}$$

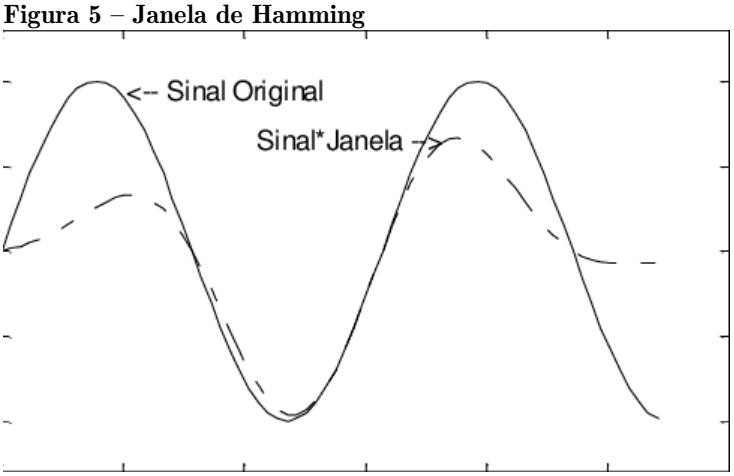
A Figura 4 exemplifica um comparativo dos filtros de janelamento como funções de uma variável contínua. Podemos observar que a Hamming, diferentemente das outras, não se aproxima de zero, ficando um pouco acima das outras nas laterais. Um exemplo da janela de Hamming pode ser vista na Figura 5.

Figura 4 – Filtros de janelamento utilizados no processamento digital



Fonte: (OPPENHEIM; SCHAFER, 2012)

Figura 5 – Janela de Hamming



Fonte: (MIRANDA *et al.*, 2003)

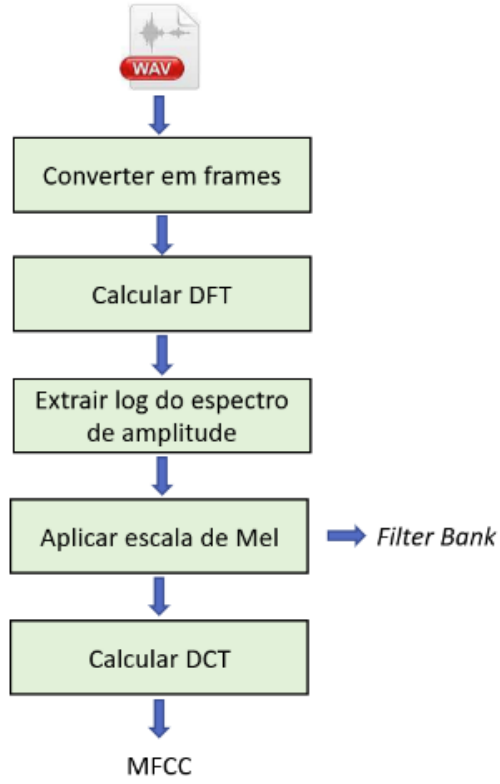
2.2.2 Coeficientes Cepstrais de Frequência de Mel e filter banks

A Figura 6 apresenta o passo a passo da extração das características do áudio, sendo a primeira etapa dividir o sinal digital da fala em *frames*, aplicando uma técnica de janelamento. O próximo passo é aplicar a transformada de Fourier em cada *frame* para conseguir um número de componentes espectrais. Então extrai-se apenas o logaritmo do espectro de amplitude. Após extrair os logaritmos, diminui-se o total de componentes do espectro utilizando a escala de Mel, nesse passo obtemos os *filter banks*. Para obter o MFCC, aplica-se a transformada discreta do cosseno (DCT - *Discrete Cosine Transform*) com o objetivo de reduzir a quantidade de parâmetros do sistema. Após esses passos obtém-se as características com as amplitudes do espectro (LOGAN *et al.*, 2000).

Como apresentado na Figura 6, a primeira etapa é converter o sinal digital s de um tamanho n, portanto, $s(n)$. Ao converter o sinal digital em *frames* obtém-se $s_i(n)$ em que i representa os *frames* do sinal digital.

O segundo passo para obter o MFCC é aplicar a transformada discreta de Fourier

Figura 6 – Processo para extrair o vetor de características dos áudios



Fonte: Autoria própria.

(DFT - *Discrete Fourier Transform*) em cada *frame* do áudio que tem como função a análise do conteúdo de frequências de sinais de tempo contínuo, para obter essa etapa aplica-se na Equação 6.

$$S_i(k) = \sum_{n=1}^{N} s_i(n) w(l) e^{\frac{-j2\pi kn}{N}} \qquad 1 \leqslant k \leqslant K \tag{6}$$

em que $S_i(k)$ é DFT do sinal digital nos frames, e $w(l)$ é uma janela de Hamming da amostra de tamanho l, K é o tamanho da DFT (HUANG *et al.*, 2001).

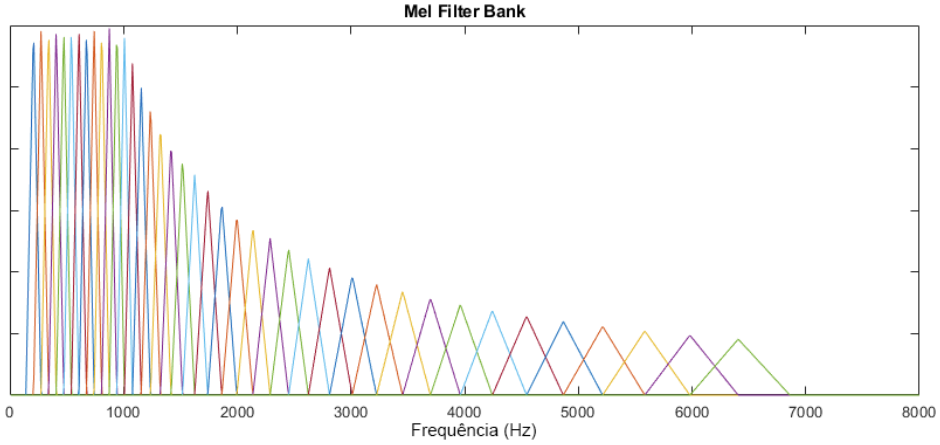
O terceiro passo é extrair o espectro de amplitude do *frame* i, onde $P_i(k)$ é a estimativa do periodograma do espectro de amplitude. Por fim aplica-se o log em cima dos resultados (HUANG *et al.*, 2001).

$$P_i(k) = \frac{1}{N} |S_i(k)|^2 \tag{7}$$

Então o quarto passo é calcular os *filter banks*. Que é um conjunto de filtros triangulares que aplicamos à estimativa espectral de potência do periodograma. Para calcular as energias do banco de filtros multiplicamos cada banco de filtros com o espectro de energia, depois somamos os coeficientes. Indicando o quanto de energia havia em cada *filter bank* (HUANG *et al.*, 2001).

A Escala de Mel M_{el} emula a resolução auditiva variável utilizando os *filter banks*, que divide o espectro em *bins* não uniformes e as somam dando uma ideia de quanta energia existe em diferentes regiões de frequência. Conforme mostrado na Figura 7, os primeiros filtros da escala de mel são muito estreitos e fornecem uma indicação de quanta energia existe em baixas frequências. Conforme as frequências ficam mais altas, os filtros ficam mais largos (STEVENS; VOLKMANN; NEWMAN, 1937; PICONE, 1993).

Figura 7 – 10 *filter banks* na escala de Mel



Fonte: Autoria própria.

A Equação 8 tem como o objetivo converter a frequência em escala de Mel M_{el} dado uma frequência f em Hz e a Equação 9 é a inversa, fazendo com que a escala de Mel volte a ser representada em frequência (PICONE, 1993).

$$M_{el}(f) = 1125ln(1 + f/700) \tag{8}$$

$$M_{el}{}^{-1}(m) = 700(e^{\frac{m}{1125}} - 1) \tag{9}$$

Por fim, para obter o MFCC é calculado a DCT dos *filter banks*, com o objetivo de reduzir a quantidade de dados.

2.3 APRENDIZADO DE MÁQUINA

Aprendizado de Máquina ou *Machine learning* é uma área que foi derivada da Inteligência Artificial (IA) e tem como objetivo gerar modelos ou regras analisando os dados (RASCHKA, 2015). O aprendizado de máquina pode ser subdividido em três tipos: aprendizado supervisionado, aprendizado não supervisionado e aprendizagem por reforço como podemos ver na Figura 8.

Sendo os três métodos de aprendizagem apresentados a seguir:

Figura 8 – Tipos de Aprendizado de Máquina

Fonte: (MEDIAVALET, 2019)

- **Aprendizado supervisionado:** Para se realizar o aprendizado supervisionado a base de dados possui um atributo classe que tem como objetivo representar o que a instância representa, com isso, o computador pode gerar um modelo para generalizar as características de cada classe.
- **Aprendizado não supervisionado:** Nesse caso a base de dados não possui um atributo classe para auxiliar o aprendizado, com isso os algoritmos não supervisionados tentam aprender características que identificam um grupo de dados, com isso realizando um agrupamento entre os dados.
- **Aprendizagem por reforço:** Esse tipo de aprendizagem é diferente das anteriores pois a aprendizagem por reforço possui um objetivo. Para isso agentes são gerados e melhorados com base nos acontecimentos do ambiente, uma forma bem comum de se realizar isso é com base de recompensas, quando um agente chega no objetivo ele é recompensado positivamente, caso contrário recebe uma recompensa negativa, e tenta melhorar a cada iteração, daí o nome aprendizagem por reforço.

De acordo com Carvalho *et al.* (2011) ter um comportamento inteligente é possuir a capacidade de aprender. A capacidade de uma máquina aprender refere-se a sua capacidade de induzir algo a partir de eventos passados que são informações que uma máquina deve analisar para realizar classificações ou predição de eventos futuros (CARVALHO *et al.*, 2011).

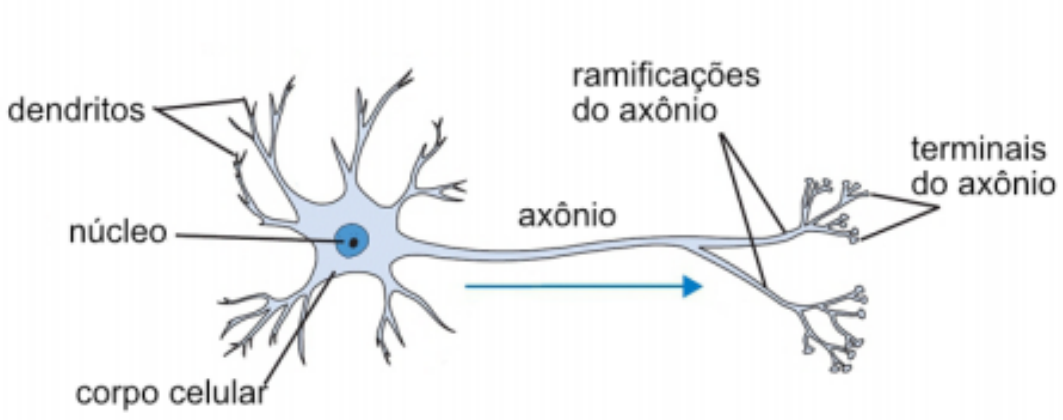
Uma das técnicas de aprendizagem de máquina são as redes neurais.

2.3.1 Redes Neurais Artificiais

As redes neurais artificiais são baseadas no funcionamento do sistema nervoso dos organismos superiores. Uma rede neural biológica é formada por células nervosas, também conhecidas como neurônios, ilustrado na Figura 9. O axônio é responsável por transmitir as informações por meio dos sinais elétricos que passam de uma célula a outra, que são ligadas aos outros neurônios por meio de terminais. Após isso, a informação é recebida

pelos dendritos e processadas no corpo celular para que sejam enviadas novamente a outros neurônios. Esse processo de comunicação entre as células nervosas é conhecido como sinapses (BEZERRA, 2016).

Figura 9 – Exemplo de neurônio biológico



Fonte: (BEZERRA, 2016)

De acordo com Haykin *et al.* (2009) as redes neurais artificiais possuem a capacidade de adquirir conhecimento por meio de um processo de aprendizagem. Os pesos sinápticos são conexões internas dos neurônios, que são aprendidos durante a fase de treinamento. De forma simplificada, uma rede é formada de neurônios que estão conectados uns aos outros na rede e possuem pesos nas conexões

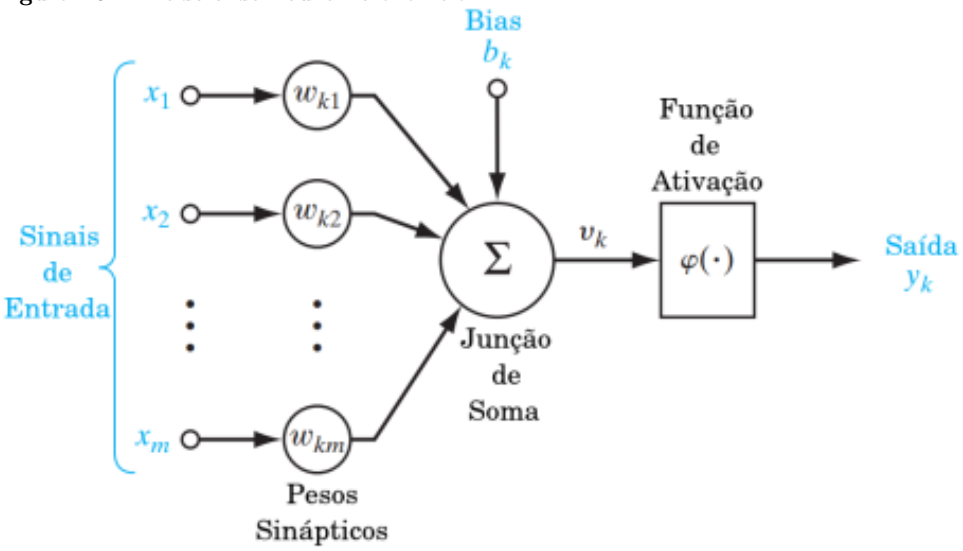
A Figura 10 apresenta um modelo de neurônio artificial, em que recebe vários sinais de entrada e apenas um de saída. As sinapses são caracterizadas por pesos sinápticos. Portanto os sinais de entrada são calculados em uma somatória para cada sinal de entrada de uma sinapse conectada a um neurônio, um peso sináptico é multiplicado, assim produzindo uma única saída. O modelo proposto também possui um bias de entrada que é uma variável incluída na função de ativação com o objetivo de aumentar ou diminuir a entrada, o valor do peso do bias é variável possibilitando ajustes durante o processo de treinamento da rede. A função de ativação é utilizada para limitar o valor da saída do neurônio a partir de um intervalo, por exemplo, mapeando os valores de saída para 0 ou 1 (HAYKIN *et al.*, 2009; SHALEV-SHWARTZ; BEN-DAVID, 2014).

Em termos matemáticos pode-se definir o neurônio artificial k pela Equação 10 e Equação 11, tal que x_j representa os sinais de entrada do neurônio, j as sinapses, w_{kj} representa os pesos sinápticos de um neurônio, φ é a função de ativação, em que u_k é o somatório dos sinais de entrada, b_k é valor do bias y_k é a saída do neurônio (HAYKIN *et al.*, 2009).

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{10}$$

$$y_k = \varphi(u_k + b_k) \tag{11}$$

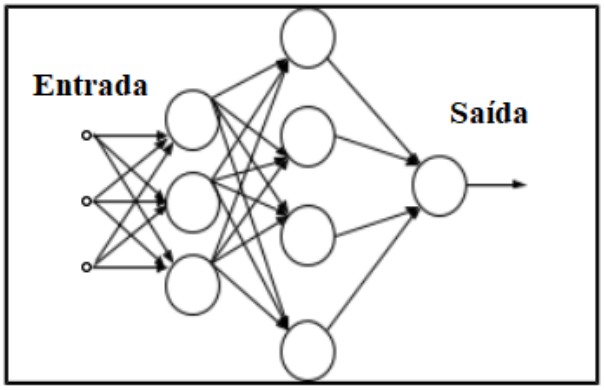
Figura 10 – Modelo de neurônio artificial



Fonte: Adaptado de (HAYKIN *et al.*, 2009)

Existem também redes neurais com camadas ocultas, um exemplo seria a rede *perceptron* multicamada ou *Multilayer perceptron* (MLP) do tipo *feed-foward* ilustrada na Figura 11, em que cada neurônio na primeira camada recebe uma entrada e dispara de acordo com os limites de decisão local predefinidos. Então a saída da primeira camada é passada para a segunda camada, cujos resultados são passados para a camada final de saída que consiste em um único neurônio (GULLI; PAL, 2017).

Figura 11 – *Multilayer Perceptron*



Fonte: Adaptado de (GULLI; PAL, 2017)

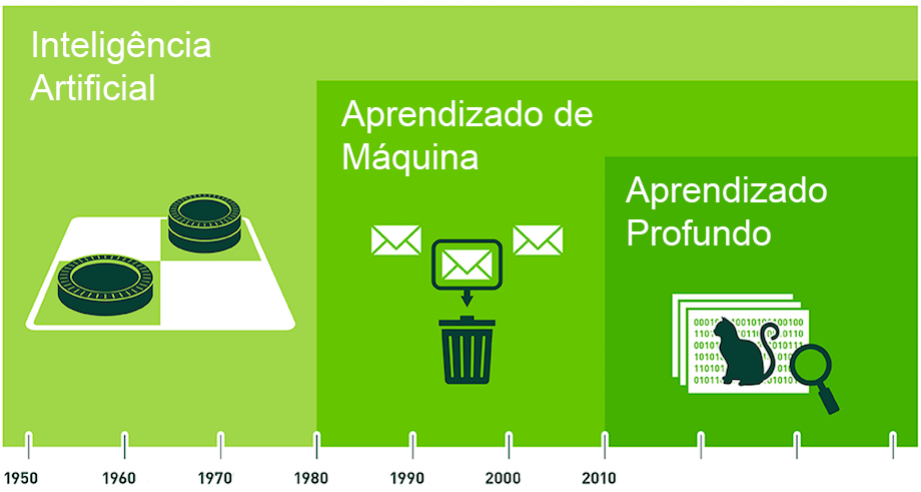
2.3.2 Redes Neurais Profundas

Redes neurais profundas são redes que possuem muitas camadas ocultas e neurônios treinados com uma grande quantidade de dados. Isso se tornou possível devido ao aumento da capacidade de processamento dos atuais computadores, assim surgindo o conceito de aprendizagem profunda ou *deep learning*. As redes neurais profundas são uteis para prever estruturas complexas em dados de alta dimensão, possuindo aplicações tanto

na visão computacional como em aplicações de reconhecimento de fala (LECUN; BENGIO; HINTON, 2015).

Deep learning pode ser definida com uma classe de técnicas de aprendizagem de máquina que explora várias camadas de processamento de informação para extração e transformação de recursos supervisionados ou não supervisionados (DENG; YU *et al.*, 2014). A Figura 12 apresenta o começo da inteligência artificial (IA) iniciando por volta de 1950, então da IA surge o aprendizado de máquina iniciando nos anos 80 e contendo o4 aprendizado profundo (*deep learning*) que passou a ser mais utilizada a partir de 2010.

Figura 12 – Linha do tempo da inteligência artificial até *deep learning*



Fonte: Adaptado de (BRIGADE, 2016)

Para a implementação de métodos de *deep learning*, pode-se fazer uso de bibliotecas desenvolvidas para essa finalidade como o Pytorch. O *Keras*[1] que é uma biblioteca de tensores otimizada para aprendizado profundo usando GPUs (*Graphics Processing Unit)* e CPUs (*Central Processing Unit*).

2.3.3 Redes Neurais Convolucionais

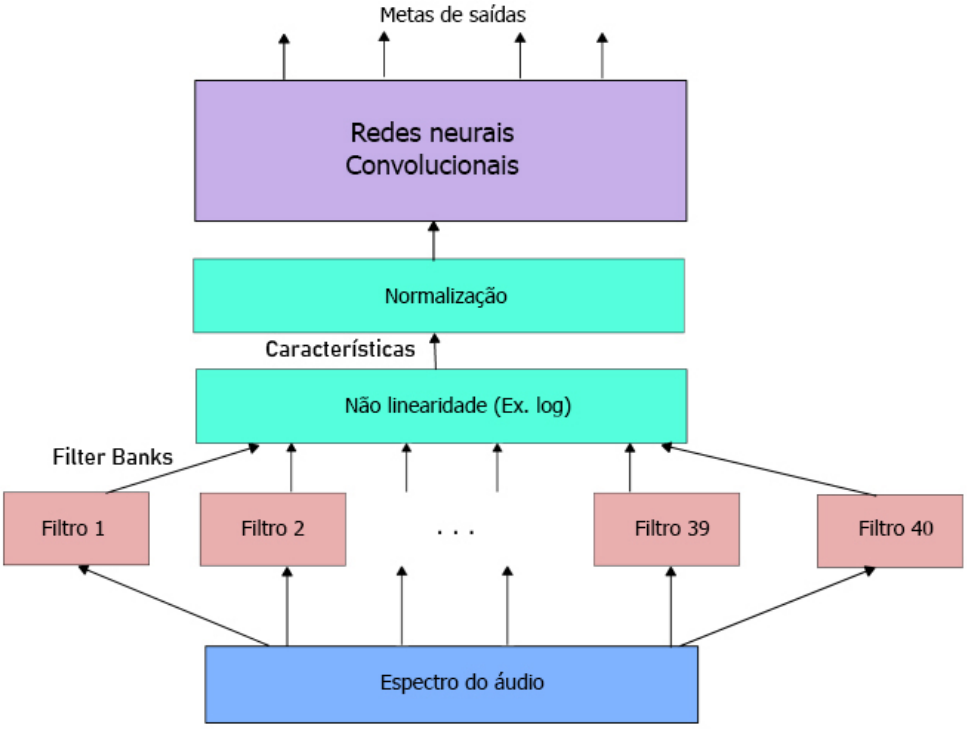
As Redes Neurais Convolucionais (CNNs) são algoritmos de inteligência artificial baseados em redes neurais multicamadas, que se inspiram no processo biológico do sistema nervoso. As CNNs consistem em várias camadas de convoluções e de agrupamento que tem a capacidade de aprender características relevantes com base nos dados de entrada (GOODFELLOW; BENGIO; COURVILLE, 2016).

A Figura 13 apresenta um modelo para reconhecimento de fala utilizando redes neurais convolucionais utilizando escala de Mel. A CNN tipicamente é estruturada com

[1] https://pytorch.org/

uma série de estágios, onde as primeiras etapas são compostas pela camada de convolução e camadas de *pooling* (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figura 13 – Modelo para reconhecimento de fala utilizando redes neurais convolucionais



Fonte: Adaptado de (DENG; YU *et al.*, 2014)

A saída das camadas convolucionais é um conjunto de combinações lineares de componentes de frequência do sinal de áudio original e as combinações dos coeficientes são determinadas pelas características dos filtros (GONG; POELLABAUER, 2018).

A camada de *pooling* tem como objetivo reduzir a saída produzida pelas camadas convolucionais, de tal forma a mesclar recursos semanticamente similares (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.3.4 Redes de Memória de Longo e Curto Prazo - LSTM

As Redes de Memória de longo e curto prazo, ou LSTM (*Long-Short Term Memory*), possuem a capacidade de aprender dependências de longo prazo criando pontes nos intervalos de tempo mesmo em caso onde há ruídos ou sequência de entradas incompreensíveis (HOCHREITER; SCHMIDHUBER, 1997).

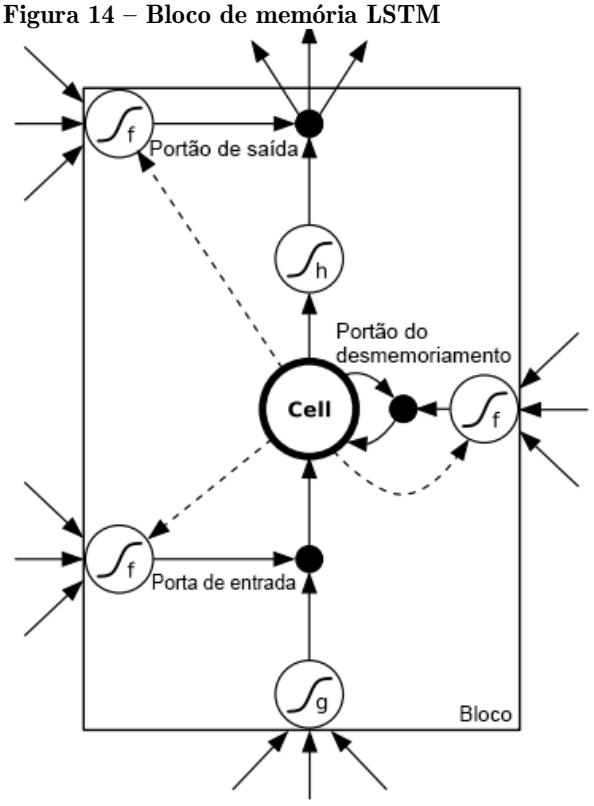
As células de memória são compostas por uma série de operações. Essas operações permitem que a LSTM esqueça ou mantenha informações. Cada célula possui 5 componentes principais: estado da célula (*cell state*), estado escondido (*hidden state*), porta de entrada (*input gate*), porta de desmemoriamento (*forget gate*) e a porta de saída (*output*

gate).

As portas (*gates*) possuem a função de ativação sigmóide (*sigmoid*), que vai projetar os valores recebidos numa escala de zero a um, sendo atribuído zero quando deve-se esquecer a informação e um para manter (memorizar) o valor (GERS; SCHMIDHUBER; CUMMINS, 1999). Dessa forma a rede consegue aprender quais dados devem ser esquecidos e quais dados devem ser guardados. Logo, as portas são responsáveis por regular o tráfego de informações nas células.

A Figura 14 apresenta um bloco de memória LSTM com uma célula. As três portas são unidades de soma não linear que coletam ativações de dentro e de fora do bloco e controlam a ativação da célula por meio de multiplicações (pequenos círculos pretos). As portas (*gates*) de entrada e saída multiplicam a entrada e a saída da célula, enquanto a porta de desmemoriamento multiplica o estado anterior da célula. Nenhuma função de ativação é aplicada dentro da célula. A função de ativação do portão 'f' é geralmente a sigmoide logística, de modo que as ativações do portão estão entre 0 (portão fechado) e 1 (portão aberto). As funções de ativação de entrada e saída da célula ('g' e 'h') são geralmente *tanh* ou sigmoide logística, embora em alguns casos 'h' seja a função de identidade.

Figura 14 – Bloco de memória LSTM



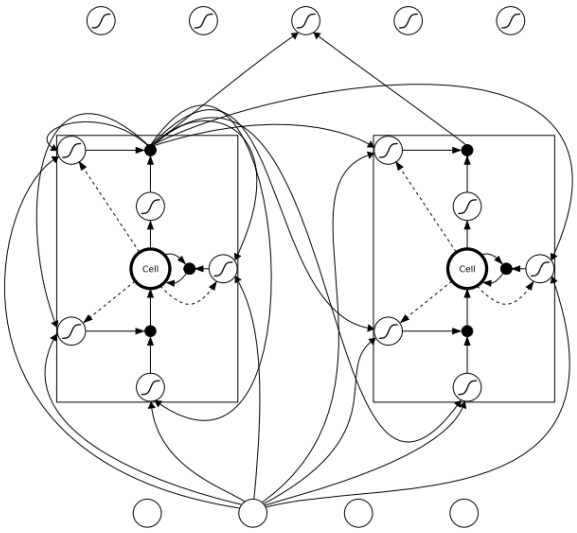
Fonte: Adaptado de (GRAVES, 2012)

As comportas são responsáveis por decidir quando manter, sobrescrever, acessar e prevenir perturbação de informações na memória da célula, possibilitando a rede LSTM extrair informações transmitidas pela ordem temporal das entradas que estão amplamente

separadas (GERS; SCHMIDHUBER; CUMMINS, 1999).

Um exemplo de uma rede LSTM pode ser visualizada na Figura 15, que consiste em quatro unidades de entrada, uma camada oculta de dois blocos de memória LSTM de célula única e cinco unidades de saída. Nem todas as conexões são mostradas. Observe que cada bloco possui quatro entradas, mas apenas uma saída.

Figura 15 – Bloco de memória LSTM



Fonte: (GRAVES, 2012)

2.3.5 Classificação Temporal Conexionista

A camada de classificação temporal conexionista (CTC - *Connectionist temporal classification*), é voltada para tarefas de classificação de modelos temporais, assim como a tarefa de classificação de áudio (GRAVES *et al.*, 2006).

De acordo com Bridle (1990) a CTC é basicamente uma camada de saída *softmax* com uma unidade a mais que os rótulos de um alfabeto, sendo esse alfabeto denotado **y** e **y'** é o conjunto do alfabeto mais o rótulo denotado "vazio" $\varnothing$ (GRAVES, 2012).

O treinamento consiste em conjunto de exemplos **X**, onde cada elemento é um par de sequências $(\mathbf{x} : \mathbf{y})$. A sequência $\mathbf{x} = (x_1, x_2, ..., x_T$ é uma sequência de entrada de tamanho T pertencente ao espaço de entrada **X**, enquanto $\mathbf{y} = (y_1, y_2, ..., y_U)$ é a sequência de saída esperada de tamanho U pertencente ao espaço de saída **y**, onde $U \leq T$. A rede gera uma distribuição de probabilidade sobre o espaço de todos os possíveis rótulos pertencentes a **y'**. Essas probabilidades estimam um caminho π sobre os elementos pertencente a **y'**.

A sequência de saída é representado por z, sendo que essa sequência de saída é

obtida pelo mapeamento β de caminhos π para o conjunto y de possíveis rótulos, ou seja $z = \beta(\pi)$, A Figura 16 apresenta um exemplo de como seria obtido a sequência dado o caminho (aaøatSiviidaaadZi), retirando os fonemas repetidos e os rótulos vazios obtendo (aatSividadZi).

Figura 16 – CTC identificando saída

Fonte: Autoria própria.

Nota-se que em casos de caracteres repetidos o ø possibilita identificar na sequência de saída ocorrências tais como "a a.." mesmo a rede identificando múltiplas vezes o mesmo carácter em um período de tempo.

3 ESTADO DA ARTE

Neste capitulo será discutido o estado da arte atual sobre reconhecimento de fonemas. Primeiramente será apresentado o método para realizar a seleção e leitura das pesquisas relevantes, levantar pontos importantes sobre o que os pesquisadores utilizam para realizar o reconhecimento de fonemas e por fim uma análise geral dos trabalhos.

3.1 SELEÇÃO E LEITURA DAS PESQUISAS RELEVANTES

Para realizar a seleção dos melhores artigos foi realizado um mapeamento com base na metodologia *Methodi Ordinatio* proposto por Pagani, Kovaleski e Resende (2015).

A intenção da pesquisa é identificar técnicas para o reconhecimento de fonemas e fazer um levantamento do que é usado para realizar tal tarefa. Para isso foi executada uma pesquisa exploratória preliminar em bases de buscas com a palavra chave "*phoneme recognition*", a tradução é reconhecimento de fonema. As bases de buscas escolhidas foram o IEEE Xplore[1], Science Direct[2], Scielo[3] e portal da Capes[4].

3.1.1 Definição e Combinação das Palavras Chaves e Base de Dados

Verificou-se três possíveis palavras chaves de buscas que podem ter pesquisas relevantes para o estudo sendo elas: *"acoustic model" AND "phoneme"*, *"speech recognition" AND "acoustic"AND "phoneme"* e *"phoneme recognition" AND ("machine learning" OR "deep learning" OR "neural network")*. Também foi definido na busca avançada o período de busca de 2010 até a data atual da busca, que no caso foi de 16 de maio de 2019. Nesse dia foram consultados todos os resultados de todas as bases.

Os resultados obtidos pelas *strings* de buscas em cada base são demonstrados na Tabela 1, totalizando 445 trabalhos.

[1] https://ieeexplore.ieee.org/Xplore/home.jsp
[2] https://www.sciencedirect.com/
[3] https://www.scielo.org/
[4] http://www.periodicos.capes.gov.br/

Tabela 1 – Resultados das *strings* de buscas nas bases de dados

	IEEE	Science Direct	CAPES	Scielo
"acoustic model" AND "phoneme"	57	18	0	1
"speech recognition" AND "acoustic" AND "phoneme"	233	38	6	0
"phoneme recognition" AND ("machine learning" OR "deep learning" OR "neural network")	73	11	8	0

Fonte: Autoria própria.

3.1.2 Processo de Filtragem

O processo de filtragem consiste em analisar o título, as palavras chaves e o resumo, para verificar se o documento está de acordo com a pesquisa. Como ferramenta de auxílio para o processo de filtragem foi utilizado o *Mendeley Reference Manager*[5] que possibilita reunir todos os trabalhos. Inicialmente elimina-se os documentos repetidos, obtendo por um total de 349 documentos. Como é uma quantidade muito extensa, foram criados alguns filtros de inclusão (FI) que mantém o documento se a condição for satisfeita e filtros de descarte (FD) para selecionar trabalhos relevantes para a pesquisa. Abaixo segue uma lista especificando os FI e FD utilizados nesse trabalho:

- FI1: A pesquisa discute sobre o tema de reconhecimento de fonemas;
- FI2: A pesquisa relata uma técnica de reconhecimento de fonemas;
- FI3: A pesquisa realiza uma comparação de uma ou mais abordagens para reconhecimento de fonemas;
- FD1: A pesquisa não é sobre reconhecimento de fonemas;
- FD2: A pesquisa leva em conta outros dados além do sinal digital do áudio;
- FD3: O autor tem múltiplos artigos com a mesma técnica e não é o mais atual;
- FD4: O artigo está em uma língua diferente do inglês ou português.

A aplicação dos filtros resultaram num total de 49 documentos, reduzindo em aproximadamente 86% do número total de trabalhos obtidos.

[5] https://www.mendeley.com/reference-management/reference-manager

3.1.3 Fator de Impacto, Ano da Publicação e Número de Citações

Como fator de impacto foi escolhido o índice H5[6] que é um indexador dos artigos publicados nos últimos cinco anos. O motivo da escolha do H5 em vez do fator JCR (*Journal Citation Reports*) deve-se ao fato do H5 avaliar também publicações de conferências. O ano da publicação e o número de citações é baseada nas informações oferecidas pelo Google Scholar.

3.1.4 Classificação com InOrdinatio

A Equação 12, adaptada de Pagani, Kovaleski e Resende (2015) é utilizado para classificar os documentos.

$$InOrdinatio = \left(\frac{H5}{10}\right) + \alpha[10 - (AnoPesq - AnoPub)] + \sum C_i \tag{12}$$

em que o $H5$ representa o fator de impacto comentado na seção 3.1.3. O valor de α é um valor que varia de um a dez e indica a importância do critério do ano que vai influenciar do resultado final da equação (PAGANI; KOVALESKI; RESENDE, 2015), *AnoPesq* é o ano da pesquisa, *AnoPub* é o ano da publicação e por fim, a soma das citações da pesquisa é a variável C_i.

Portanto, para o α foi escolhido o valor 5 de forma que o tempo não fosse um fator influente, porém, não prejudica tanto as técnicas mais antigas. Se o objetivo da pesquisa for mais teórico, recomenda-se utilizar valores inferiores para não diminuir a pontuação de publicações antigas.

3.2 ANÁLISE DOS TRABALHOS SELECIONADOS

A partir do conjunto de documentos selecionados e classificados com o método Methodi InOrdinatio foi realizada uma leitura para identificar os pontos importantes para o reconhecimento dos fonemas e fazer uma análise sobre o que é usado atualmente no estado da arte.

Ao realizar as leituras identificou-se alguns artigos que passaram despercebidos e que deveriam ser retirados pelos filtros de descarte apresentados na 3.1.2, Por fim 41 artigos foram selecionados que podem ser visualizados no apêndice A.

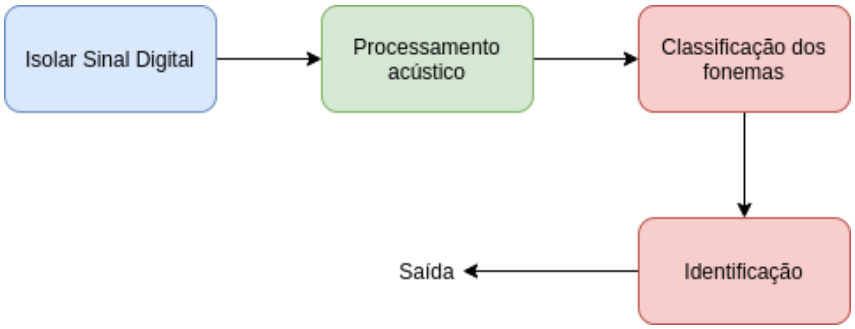
Com base na leitura preliminar desses artigos, identificou-se alguns aspectos importantes sobre o tema:

[6] <https://scholar.google.com.br/citations?view_op=top_venues&hl=pt-BR>

- Utilização de bases de dados para testes;
- Técnica de processamento acústico para extrair as características do áudio;
- Técnica de classificação dos fonemas;
- Discussão da acurácia do modelo proposto.

Kshirsagar *et al.* (2012) fazem uma comparação entre vários trabalhos de reconhecimento de fonemas mais antigos e apresentam um diagrama de blocos sobre alguns aspectos principais para realizar o reconhecimento de fonemas, como pode ser observado na Figura 17, sendo muito similar aos itens levantados anteriormente. Primeiramente o sinal digital é isolado. O próximo passo é realizar o processamento acústico do sinal, em que o objetivo é deixar mais claro para o reconhecimento de fonemas. Feito esse pré-processamento é realizada a classificação comparando as características obtidas e a identificação dos fonemas e por fim trazendo a saída com os fonemas resultantes.

Figura 17 – Diagrama de blocos para a técnica de reconhecimento de fonemas



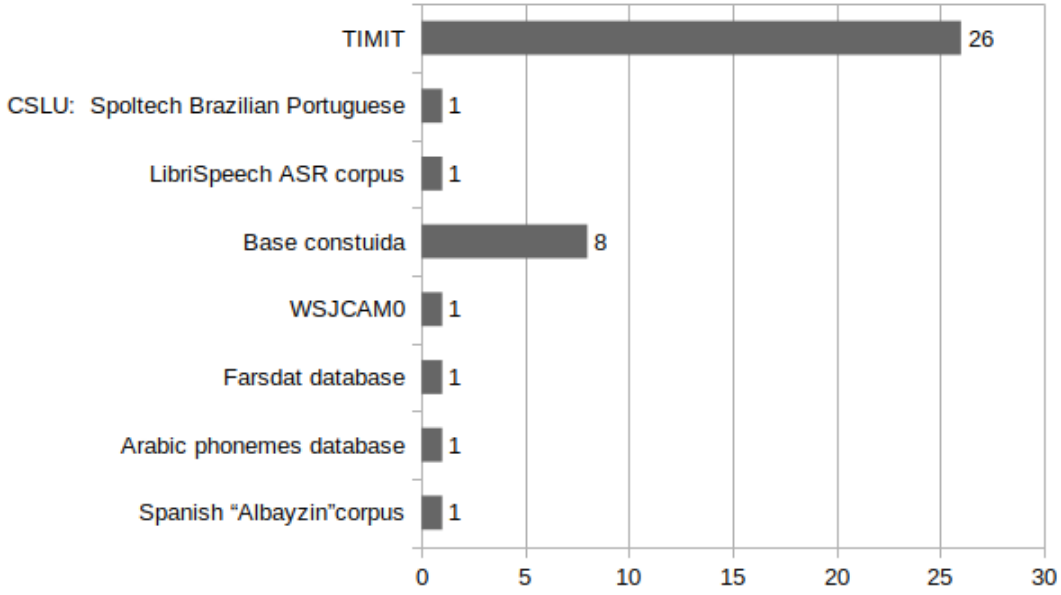
Adaptada de (KSHIRSAGAR *et al.*, 2012)

3.2.1 Bases de Dados

Com relação a base de dados, observou-se que a grande maioria utiliza a base em inglês TIMIT *Acoustic-Phonetic Continuous Speech Corpus* (GAROFOLO, 1993). Também foi possível observar a existência de várias outras bases de outras línguas ou os autores construíram a própria base de dados. Apenas um trabalho com base de dados em português brasileiro, para o reconhecimento de fonemas, foi listada na consulta realizada, em que Cardona, Nedjah e Mourelle (2017) utilizam a *CSLU: Spoltech Brazilian Portuguese Version 1.0* (SCHRAMM *et al.*, 2006). A Figura 18 apresenta um gráfico com uma análise quantitativa das bases utilizadas nos artigos selecionados.

Alguns artigos também apresentam testes utilizando ruídos da base NOISEX-92 junto com a TIMIT.

Figura 18 – Quantidade de trabalhos com as bases de áudios utilizadas



Fonte: Autoria própria.

3.2.2 Processamento Acústico

O processamento acústico nada mais é do que preparar os áudios para treina-los com algoritmos de classificação, ou seja, extrair as características dos áudios.

A janela de Hamming, inicialmente é utilizada para segmentar em *frames* o sinal digital e aplicar esses *frames* em técnicas de extração de características, porém, em geral ocorre uma grande variedade entre essas técnicas. A maioria dos trabalhos utilizam os coeficientes de Mel (MFCC), sendo que existem diferenças na aplicação do método, como a quantidade de dimensões utilizadas, a utilização do delta ou delta-delta ou ainda a adição de outras técnicas. Por exemplo, Khwaja *et al.* (2016) utilizam o MFCC juntamente com a função da área do trato vocal. Outras técnicas são o *perceptual linear prediction* (PLP), *Tandem features*, *Linear Predictive Coefficients* (LPC), *Time-Frequency Features*, *frequency domain linear prediction* e *hierarchical bottleneck features*.

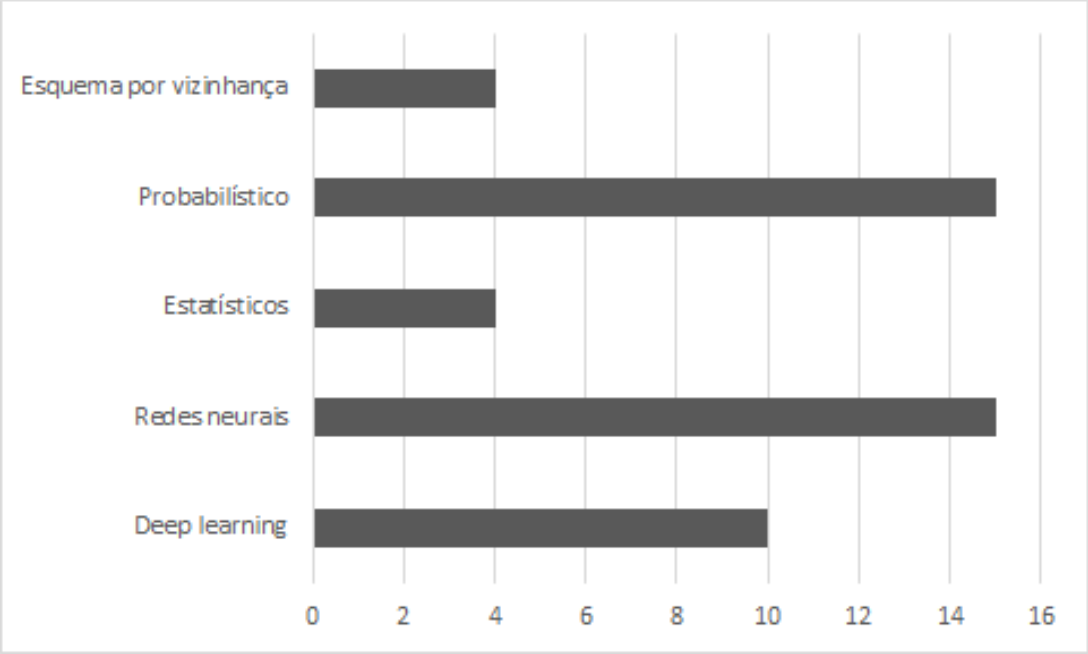
Alguns trabalhos têm como foco melhorar os métodos de processamento acústico já existentes, como Kazemi e Sobhanmanesh (2011) propõem uma melhora nos métodos *tandem features* e PLP conseguindo uma melhora de 2,2% e 2,7% respectivamente.

3.2.3 Técnicas de Classificação

Nos trabalhos selecionados, foram utilizadas múltiplas técnicas, dentre elas algoritmos de *deep learning*, redes neurais, métodos probabilísticos e estatísticos como o modelo oculto de Markov (HMM - *Hidden Markov Model*) e máquina de vetores de su-

porte (SVM - *Support Vector Machines*). A Figura 19 apresenta um gráfico com a relação das técnicas utilizadas nos artigos: *deep learning* sendo as técnicas provenientes do aprendizado profundo, por exemplo CNN e DNN; redes neurais como *multilayer perceptron* (MLP); algoritmos estatísticos como o SVM; probabilístico como o HMM; e esquema de vizinhança como o *k-Nearest Neighboors* (KNN) e florestas de decisão aleatória (*random forest trees*). Pode-se observar que a soma de todas as técnicas é maior que o total de trabalhos, isso se dá pelo motivo de que alguns artigos fazem comparações entre mais de uma técnica.

Figura 19 – Técnicas utilizadas no estudo do estado da arte



Fonte: Autoria própria.

3.2.4 Acurácia

Para medir a efetividade das abordagens utilizadas para resolver problemas de aprendizado de máquina, são utilizadas as seguintes métricas de avaliação: acurácia (*accuracy*), precisão (*precision*), revocação (*recall*) e F-*measure* (F1). Acurácia mede a soma total de acertos em todas as categorias. Precisão é a razão dos elementos classificados corretamente na categoria dividido pelo número de elementos classificados nessa categoria, enquanto a revocação é o número de acertos corretamente classificadas na categoria dividido pelo número de elementos da categoria. Já F1 sumariza as métricas precisão e revocação (SERRANI, 2015).

Observando os resultados de cada trabalho realizou-se uma relação para cada forma de avaliar a acurácia da técnica proposta, sendo a mais utilizada para a avaliação a taxa de erros de fonemas (PER -*Phoneme Error Rate*), representada na Equação 13 e

a acurácia, representada na Equação 14.

$$PER = 1 - \frac{N - D - I - S}{N} \tag{13}$$

$$Acurácia = \frac{N - D - I - S}{N} \tag{14}$$

em que N é o número total de amostras, S, D e I representam a contagem dos erros dos fonemas substituídos, deletados e inseridos respectivamente.

A Tabela 2 apresenta os sete trabalhos com os melhores resultados, foi estabelecido como o padrão para o PER. Um detalhe importante a se ressaltar é que para essa comparação considerou-se o reconhecimento de fonemas de múltiplos locutores, pois em Kannadaguli e Bhat (2015) conseguiu uma taxa de acerto de 100% para os fonemas testados em uma base própria. Isso mostra que essa tarefa é um pouco mais fácil, porém ainda relevante por se tratar de reconhecimento de fonemas.

Tabela 2 – Taxas de erros de fonemas (PER) no conjunto de testes dos melhores trabalhos pesquisados

Referências dos trabalhos	PER	Base
(CARDONA; NEDJAH; MOURELLE, 2017)	15,90%	CSLU
(KHWAJA *et al.*, 2016)	16,05%	WSJCAM0
(DANESHVAR; VEISI, 2016)	17,55%	Farsdat database
(ZHANG *et al.*, 2016)	18,30%	LibriSpeech corpus
(AGER; CVETKOVIĆ; SOLLICH, 2010)	18,50%	TIMIT
(LOHRENZ; LI; FINGSCHEIDT, 2018)	19,46%	TIMIT
(SELTZER; DROPPO, 2013)	20,25%	TIMIT

Fonte: Autoria própria.

3.3 TRABALHOS RELACIONADOS

Com a Seção 3.2 observou-se alguns pontos chaves para o reconhecimento automático de fonemas. Nessa seção serão apresentados alguns trabalhos principais de forma detalhada.

Lohrenz, Li e Fingscheidt (2018) apresentam um artigo utilizando uma técnica de turbo *fusion* entre DNN e CNN. A base de áudios utilizada foi a TIMIT usando o conjunto de treinamento padrão com um total de 3696 frases (os áudios com prefixo SA foram removidos) e o conjunto do *core test* com 192 frases. Para os resultados utilizou-se um mapeamento dos 61 fonemas para 39. O processamento acústico utiliza uma janela de Hamming de 25 ms com uma taxa de *frames* de 10 ms. A técnica de extração de características aplicada foi o *filter banks*, com 40 coeficientes de Mel baseados na transformada discreta de Fourier e um coeficiente de log-energia adicional. Derivadas temporais comuns

de primeira e segunda ordem foram anexadas ao vetor de características, resultando em um vetor de característica com 123 dimensões por *frame*. Com essa técnica apresentada, o melhor resultado apresentado no *core test* foi de 18,80% de PER.

Seltzer e Droppo (2013) utilizam uma técnica de aprendizagem multitarefa com DNN para o reconhecimento de fonemas utilizando a base TIMIT onde os 61 rótulos de fonemas foram mapeados em um conjunto de 39 rótulos de fonemas. A aprendizagem multitarefa é uma técnica em que uma tarefa de aprendizagem primária é resolvida juntamente com outras relacionadas usando uma representação de entrada compartilhada. Se essas tarefas forem bem escolhidas, a estrutura compartilhada serve para melhorar a generalização do modelo e sua precisão em um conjunto de teste invisível. (CARUANA, 1997). Para o processamento acústico, Seltzer e Droppo (2013) utilizaram uma janela de contexto de 11 *frames* de dados acústicos formados a partir do *frame* de destino no tempo t e 5 *frames* anteriores e subsequentes. Cada quadro foi representado por *filter banks* com 40 coeficientes de Mel e suas derivadas de primeira e segunda ordem. O melhor resultado obtido no *core test* foi de 20,25% de PER.

Além dos trabalhos encontrados pelo *Methodi Ordinatio*, também foram pesquisados outros artigos e dissertações que não apareceram no levantamento. Foi encontrada uma dissertação utilizando bases de áudios em português cujo o foco é o reconhecimento de palavras.

Abdel-Hamid *et al.* (2014) apresentam no artigo uma técnica para o reconhecimento de fonemas utilizando CNN juntamente com CTC. Os autores aplicaram o método de redução de fonemas de 61 para 39 na base de áudios TIMIT para a avaliação do método proposto e foram excluídos os áudios com prefixo SA. Para o processamento acústico utilizou-se a janela de Hamming de 25 ms com uma taxa de *frames* fixa de 10 ms. Os vetores de características dos áudios são gerados pela análise de banco de filtros baseada na transformada de Fourier, que inclui coeficientes de energia de 40 *log* distribuídos em uma escala de Mel, junto com suas primeiras e segundas derivadas temporais. Os autores obtiveram uma PER no core test de 20.39% no melhor caso.

Quintanilha (2017) apresenta em sua dissertação uma técnica de reconhecimento de fala utilizando CNN com LSTM e como camada de saída o CTC nas bases em português brasileiro LaPS benchmark 16k, Sid, VoxForge e CSLU: Spoltech Brazilian Portuguese. O melhor resultado obtido foi uma taxa de erro de caractere de 25,13%.

4 MATERIAIS E MÉTODOS

Neste capítulo são apresentados os elementos necessários para o desenvolvimento do trabalho proposto, incluindo os recursos de hardware e software, método para a transcrição fonética das bases de áudios em português, as técnicas para o pré-processamento, o modelo da rede neural e uma descrição das bases utilizadas para o treinamento, validação e testes.

A Figura 20 apresenta as etapas do trabalho, sendo que a primeira consiste em pesquisar bases de áudios de fala para o reconhecimento de fonemas. No caso das bases de áudio em português brasileiro selecionadas só as transcrições ortográficas estão disponíveis, então, foi necessário adicionar as transcrições fonéticas alinhadas no tempo. Em seguida, com as bases selecionadas, preparou-se o ambiente experimental para em seguida realizar o processamento acústico com objetivo de extrair as características dos áudios. Por fim, a última etapa do trabalho consiste em aplicar a rede neural proposta, classificar as características extraídas e verificar as acurácias obtidas no reconhecimento de fonemas para ter uma visão geral do desempenho do método.

As bases escolhidas foram a TIMIT, LaPS Benchmark 16k e Sid. Como as duas últimas bases não possuem a transcrição fonética disponível nos arquivos, foi necessário gerar as transcrições fonéticas e o alinhamento dos *frames* no tempo.

As etapas do método proposto são descritas nas seções seguintes. A Seção 4.1 apresenta o ambiente experimental utilizado, assim como uma descrição detalhada de cada base de áudio. A Seção 4.2 explica como é feita a extração das características dos áudios e a seção 4.3 apresenta o modelo de rede utilizada para a classificação dos fonemas, obtendo a transcrição fonética.

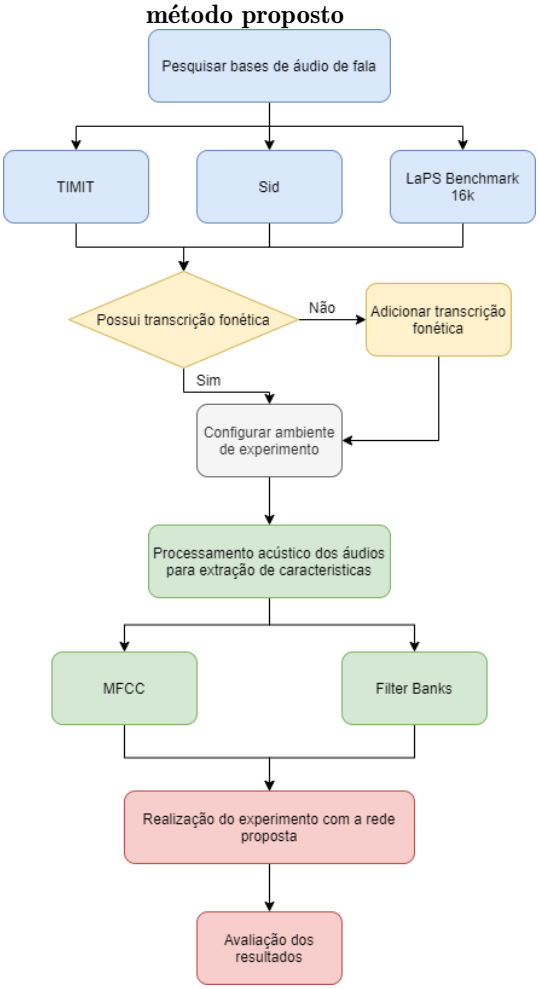
4.1 SETUP EXPERIMENTAL

Nesta seção serão apresentados detalhes do ambiente experimental como configuração do computador e as versões dos softwares utilizados. Na sequência é apresentada uma descrição das bases de áudios selecionadas para realizar os experimentos e como foi realizada a métrica de avaliação da rede proposta.

4.1.1 Ambiente Experimental

Para os experimentos utilizou-se um computador com as especificações apresentadas na Tabela 3. O principal recurso é a utilização de uma placa de vídeo da NVIDIA, para

Figura 20 – Fluxograma com as etapas do método proposto



Fonte: Autoria própria.

a utilização do CUDA (*Compute Unified Device Architecture*) para auxiliar o treinamento das redes neurais.

Tabela 3 – Configuração da máquina utilizada nos experimentos

Item	Característica
Processador	AMD Ryzen 3600 6 Cores - 3600 MHz
Memória RAM	16 GB
Placa de vídeo	NVIDIA GTX 1660 6GB (GDDR5 192 GB/seg); 1408 NVIDIA CUDA cores; arquitetura Turing

Fonte: Autoria própria.

Para a implementação do experimento configurou-se um ambiente de trabalho utilizando o Conda. A Tabela 4 apresenta as ferramentas utilizadas para o experimento e suas versões correspondentes.

Conda um sistema de gerenciamento de pacote de código aberto e sistema de gerenciamento de ambiente. O Pytorch é um pacote do Python que da suporte a cálculos de tensores (como NumPy) com aceleração de GPU e redes neurais profundas. O *Kaldi Speech Recognition Toolkit* é uma ferramenta *open source* para o reconhecimento de fala.

Tabela 4 – Ferramentas utilizadas nos experimentos

Ferramenta	Versão
Conda	4.9.2
Python	3.8.6
Pytorch	1.7.0
NumPy	1.19.2
Kaldi Speech Recognition Toolkit	5.5
CUDA	11.1

Fonte: Autoria própria.

CUDA é uma plataforma de computação paralela e modelo de programação desenvolvida pela NVIDIA para computação geral em unidades de processamento gráfico.

4.1.2 TIMIT *Acoustic-Phonetic Continuous Speech Corpus*

A TIMIT *Acoustic-Phonetic Continuous Speech Corpus* é uma base que já possui uma divisão entre áudios para treino e teste, sendo que o teste possui uma subdivisão de *core test*. O conjunto de treino possui um total de 4620 áudios, sendo que normalmente excluem-se os áudios com o prefixo SA, utilizando somente os SI e SX, totalizando um total de 3696 áudios de 462 locutores. O conjunto de testes possuem um total de 592 áudios já desconsiderando os áudios com prefixo SA. O *core test* possui 192 áudios com 32 locutores conforme apresentados na Tabela 5.

Tabela 5 – Locutores no conjunto do *Core Test*

Dialeto	Masculino	Feminino	Áudios/Locutor	Total de Áudios
1	DAB0, WBT0	ELC0	8	24
2	TAS1, WEW0	PAS0	8	24
3	JMP0, LNT0	PKT0	8	24
4	LLL0, TLS0	JLM0	8	24
5	BPM0, KLT0	NLP0	8	24
6	CMJ0, JDH0	MGD0	8	24
7	GRT0, NJM0	DHC0	8	24
8	JLN0, PAM0	MLD0	8	24
Total	16	8		192

Fonte: (GAROFOLO, 1993)

Para o treinamento das redes neurais foram utilizados os áudios do conjunto de treino. O conjunto do *core test* para os testes, e para validação o restante dos áudios do conjunto de teste totalizando 400 áudios assim como Lohrenz, Li e Fingscheidt (2018). A base foi rotulada utilizando 61 fonemas, porém muitos artigos reduzem para 48 ou 39 fonemas utilizando um mapeamento apresentado por Lee e Hon (1989).

A Tabela 6 apresenta a quantidade de amostras de cada fonema para a base

completa (Total) e para a parte selecionada para o Treino. A coluna (61) apresenta o somatório de todas ocorrências do fonema e na coluna (39) há duas possibilidades, sendo o primeiro caso em que apresenta o somatório de ocorrências do fonema e o segundo caso mostra o fonema em que ele é mapeado. Como por exemplo: O fonema *aa* possui um Total de 2623 amostras para 61 fonemas no alfabeto e o fonema *ao* é mapeado para *aa* com o alfabeto de 31 fonemas. Por fim, é feito uma somatória entre os fonemas mapeados, como no exemplo $aa + ao$ obtendo 4808 amostras para o fonema *aa* com um alfabeto de 39 fonemas.

Tabela 6 – Quantidade de fonemas no conjunto de áudios para o treino

	Total		**Treino**			**Total**		**Treino**	
	Quantidade		**Quantidade**			**Quantidade**		**Quantidade**	
Fonema	**61**	**39**	**61**	**39**	**Fonema**	**61**	**39**	**61**	**39**
aa	2623	4808	2256	4121	**ix**	8504	ih	7370	ih
ae	2634	2634	2292	2292	**iy**	5433	5433	4626	4626
ah	2649	7130	2266	6158	**jh**	1141	1141	1013	1013
ao	2185	aa	1865	aa	**k**	4335	4335	3794	3794
aw	819	819	728	728	**kcl**	4750	sil	4149	sil
ax	4073	ah	3535	ah	**l**	5205	6314	4425	5376
ax-h	408	ah	357	ah	**m**	4062	4205	3442	3566
axr	2907	er	2445	er	**n**	7208	8718	6219	7526
ay	2237	2237	1934	1934	**ng**	1343	1370	1194	1220
b	2562	2562	2181	2181	**nx**	790	n	677	n
bcl	2233	sil	1909	sil	**ow**	1910	1910	1653	1653
ch	924	924	820	820	**oy**	350	350	304	304
d	2784	2784	2432	2432	**p**	3005	3005	2588	2588
dcl	4554	sil	3998	sil	**pau**	1038	sil	891	sil
dh	2765	2765	2376	2376	**pcl**	3056	sil	2644	sil
dx	2116	2116	1864	1864	**q**	3080	3080	2685	2685
eh	3857	3857	3277	3277	**r**	5494	5494	4681	4681
el	1109	l	951	l	**s**	7158	7158	6176	6176
em	143	m	124	m	**sh**	1518	1696	1317	1466
en	720	n	630	n	**t**	4548	4548	3948	3948
eng	27	ng	26	ng	**tcl**	6601	sil	5725	sil
epi	1063	sil	908	sil	**th**	859	859	745	745
er	1979	4886	1693	4138	**uh**	583	583	500	500
ey	2615	2615	2271	2271	**uw**	599	2190	529	1952
f	2626	2626	2215	2215	**ux**	1591	uw	1423	uw
g	1384	1384	1191	1191	**v**	2310	2310	1994	1994
gcl	1526	sil	1312	sil	**w**	2639	2639	2216	2216
h#	8576	sil	7392	sil	**y**	1160	1160	995	995
hh	1094	1911	937	1660	**z**	4225	4225	3682	3682
hv	817	hh	723	hh	**zh**	178	sh	149	sh
ih	4895	13399	4248	11618	**sil**	-	32797	-	28928

Fonte: Autoria própria.

4.1.3 LaPS Benchmark 16k

É um corpus que contem 700 frases de 35 locutores sendo 25 homens e 10 mulheres. As gravações foram realizadas em ambientes não controlados existindo ruídos nas gravações, como conversas ao fundo dificultando o reconhecimento de fonemas. Como essa base não possui divisões pra treino e teste, decidiu-se separar a base em conjuntos de treino (60%), validação (20%) e teste (20%), sendo que um locutor não pode estar presente em diferentes conjuntos.

Para garantir que o resultado mostre uma generalização, decidiu-se dividir a base em quatro diferentes organizações selecionando locutores distintos para realizar o treinamento, validação e teste. A técnica assemelhando-se ao *k-fold cross validation* com um k=4 (HAYKIN *et al.*, 2009). A Tabela 7 apresenta a distribuição de locutores presentes em cada conjunto.

Tabela 7 – Distribuição de locutores para treino, validação e teste

Treino					Validação		Teste	
Masculino			Feminino		Masculino	Feminino	Masculino	Feminino
M033	M022	M032	F025	F004	M024	F006	M009	F012
M021	M016	M029	F026		M002	F015	M007	F035
M023	M018	M028	F019		M027		M011	
M003	M031	M020	F013		M001		M010	
M017	M008	M005	F014		M034		M030	
M022	M032	M020	F019	F015	M009	F012	M033	F025
M016	M029	M005	F013		M007	F035	M021	F026
M018	M028	M024	F014		M011		M023	
M031	M001	M002	F004		M010		M003	
M008	M034	M027	F006		M030		M017	
M032	M024	M001	F014	F035	M033	F025	M022	F019
M029	M002	M034	F004		M021	F026	M016	F013
M028	M027	M009	F006		M023		M018	
M020	M010	M007	F015		M003		M031	
M005	M030	M011	F012		M017		M008	
M024	M009	M010	F006	F026	M022	F019	M032	F014
M002	M007	M030	F015		M016	F013	M029	F004
M027	M011	M033	F012		M018		M028	
M001	M003	M021	F035		M031		M020	
M034	M017	M023	F025		M008		M005	

Fonte: Autoria própria.

A Tabela 8 apresenta a quantidade de amostras de cada fonema para a base completa, sendo que os fonemas são representados utilizando o formato SAMPA (*Speech*

Assessment Methods Phonetic Alphabet), que é baseado no alfabeto fonético internacional (WELLS *et al.*, 1997).

Tabela 8 – Quantidade de fonemas no *dataset* LaPS Benchmark 16k

Fonema	Total de amostras	Fonema	Total de amostras
_	1792	L	73
a	4232	m	887
a~	763	n	777
b	340	o	1220
d	1729	O	233
e	1940	o~	323
E	328	p	1026
e~	732	4	1426
f	393	s	3072
g	340	S	500
h/	981	t	1770
i	2804	u	2369
i~	432	u~	240
j	818	v	527
J	74	w	679
j~	284	w~	393
k	1137	z	388
l	660	Z	794

Fonte: Autoria própria.

4.1.4 Base Sid

Essa base possui arquivos de áudios com as transcrições ortográficas de 72 locutores. Foram selecionados um total de 700 áudios de 51 locutores diferentes, sendo 21 com falas femininas e 30 com falas masculinas. O critério para a seleção foi de que as frases deveriam possuir mais de uma palavra, pois na base possui áudios com palavras isoladas e de fala continua. Os áudios dessas bases foram gravados em ambientes não controlados, apresentando ruídos de fundo. A Tabela 9 apresenta um detalhamento da quantidade de áudios e o conjunto referente a cada locutor utilizado.

A Tabela 10 apresenta a quantidade de amostras selecionadas de cada fonema para a base Sid, sendo que para a representação dos fonemas também é utilizado o formato SAMPA.

Tabela 9 – Distribuição da base Sid com transcrições fonéticas adicionadas

Locutor	Quantidade de áudios	Conjunto	Locutor	Quantidade de áudios	Comjunto
F0001	61	Treino	M0006	19	Treino
F0002	32	Treino	M0007	10	Treino
F0003	12	Treino	M0008	10	Validação
F0004	40	Treino	M0009	10	Validação
F0005	19	Treino	M0010	10	Validação
F0006	10	Treino	M0011	10	Validação
F0007	10	Validação	M0012	10	Validação
F0008	10	Validação	M0013	10	Validação
F0009	10	Validação	M0014	10	Treino
F0010	10	Validação	M0015	10	Treino
F0011	10	Treino	M0016	10	Treino
F0012	10	Treino	M0017	10	Treino
F0013	10	Treino	M0018	10	Treino
F0014	10	Treino	M0019	10	Treino
F0015	10	Treino	M0020	10	Treino
F0016	10	Treino	M0021	10	Treino
F0017	10	Treino	M0022	10	Treino
F0018	10	Teste	M0023	10	Treino
F0019	10	Teste	M0024	10	Treino
F0020	10	Teste	M0025	10	Teste
F0021	10	Teste	M0026	10	Teste
M0001	30	Treino	M0027	10	Teste
M0002	33	Treino	M0028	10	Teste
M0003	12	Treino	M0029	10	Teste
M0004	21	Treino	M0030	10	Teste
M0005	21	Treino			

Fonte: Autoria própria.

4.1.5 Outras Bases de dados de áudio

As bases de dados de áudio são utilizadas para o estudo no desenvolvimento de sistemas para o ASR. Essas bases possuem pelo menos a transcrição ortográfica, que é uma transcrição em forma de texto do que foi falado no áudio. Outras bases também possuem transcrições em nível fonético, explicada na sessão 2.1.2, junto com alinhamento de tempo, isso é, para cada ocorrência de um fonema, possui dados do tempo inicial e final. A lista a seguir descreve algumas características de cada base de dados:

- **CSLU: *Spoltech Brazilian Portuguese Version 1.0*:** Essa base contém áudios de fala de várias regiões do Brasil. Contém transcrições ortográficas e fonéticas. Essa base possui arquivos de áudios de 477 locutores e um total de 8080 enunciados. Onde 2540 enunciados possuem transcrições de palavras sem alinhamento de tempo e 5479 possuem transcrições em nível fonético com alinhamento de tempo (SCHRAMM *et*

Tabela 10 – Quantidade de fonemas no dataset Sid

Fonema	Total de amostras	Fonema	Total de amostras
_	1518	L	83
a	3523	m	894
a∼	610	n	626
b	272	O	196
d	1445	o	1063
E	272	o∼	312
e	1797	p	807
e∼	711	4	1269
f	289	S	433
g	251	s	2916
h/	829	t	1782
i	2491	u	2156
i∼	428	u∼	259
J	77	v	487
j	829	w	620
j∼	256	w∼	333
k	1124	z	359
l	470	Z	651

Fonte: Autoria própria.

al., 2006).

- **LibriSpeech ASR corpus:** É um corpus de livre acesso, contendo 1000 horas de áudios extraídos de audiolivros em inglês. Foram realizados três subconjuntos de treino com aproximadamente 100, 360 e 500 horas respectivamente. O subconjunto de treino de 500 horas possui 1166 locutores, onde 564 são mulheres e 602 são homens (PANAYOTOV *et al.*, 2015).

- **WSJCAM0 *Cambridge Read News*:** É uma base de dados de fala em inglês britânico que foi desenvolvida com o objetivo de reconhecimento de fala contínua de grande vocabulário. Ela possui 140 locutores no total, onde 94 locutores gravaram 90 enunciados cada e 48 locutores gravaram 40 sentenças com apenas palavras. A base possui transcrições ortográficas, fonéticas e de palavras alinhadas (ROBINSON *et al.*, 1995).

- **Farsdat *database*:** É um banco de dados de fala persa que compreende gravações de 300 locutores nativos, de 10 regiões dialetais diferentes do Irã. As 6000 frases foram segmentadas e rotuladas, incluindo 386 sentenças foneticamente equilibradas (BIJANKHAN; SHEIKHZADEGAN; ROOHANI, 1994).

- ***Arabic letters corpus*:** Essa base de dados tem como objetivo avaliar a fala da língua arábica para não nativos, ela consiste de 50 locutores, com 1400 gravações da fala (ALMISREB; ABIDIN; TAHIR, 2013).

- ***Albayzin speech dabase*:** É uma base em espanhol dividida em três partes: base fonética, base de aplicação e banco de fala Lombard. A base fonética consiste em dois subconjuntos, a designada para treinamento que possui 200 sentenças foneticamente balanceadas de 160 locutores e a outra é designada para testes possuindo 500 sentenças foneticamente balanceadas de 40 locutores (BILBAO *et al.*, 1993).
- **VoxForge**[1]: Criado para coletar transcrições de fala para uso com programas de reconhecimento de voz livres e baseados em código aberto. Qualquer pessoa pode ajudar a aumentar essa base lendo algum texto, portanto seu tamanho é relativo.

4.1.6 Transcrição Fonética das Bases em Português

Uma das formas de se obter uma transcrição fonética é com base na técnica de grafema para fonema, em que tem como entrada o texto da transcrição ortográfica e como saída uma sequência de fones. Um exemplo dessa técnica é o sistema web PETRUS (*PhonEtic TRanscriber for User Support*) que realiza a transcrição fonética automática de lemas em verbetes de dicionários do português do Brasil apresentado por Serrani (2015), que realiza a transcrição fonética automática do português brasileiro. A técnica apresentada não realiza a segmentação em relação ao tempo do áudio.

Um processo para obter uma segmentação em relação ao tempo foi apresentada por Dijkstra e Sanches (2020), utilizando o pacote de software para fonética Praat. O Praat é uma ferramenta de análise de áudio que tem como objetivo facilitar os estudos em pesquisas fonéticas juntamente com o plugin EasyAlign (ferramenta de alinhamento fonético automático para fala contínua sob Praat), que recebe como entrada um áudio e sua transcrição ortográfica tendo como saída a transcrição fonética, silábica, lexical e de fala (BOERSMA, 2020; GOLDMAN, 2011).

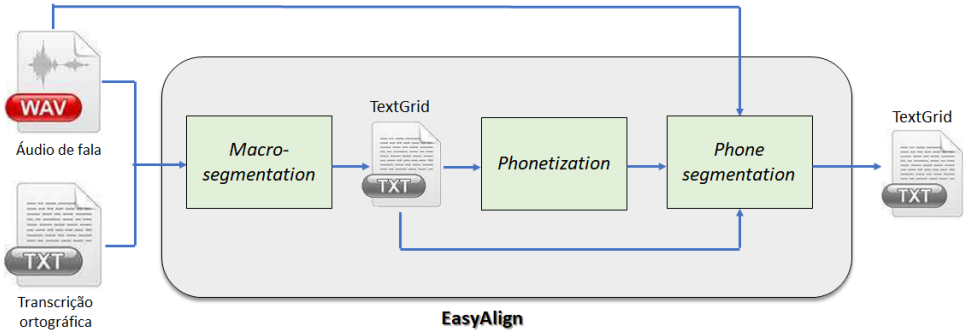
O processo de segmentação de um arquivo de fala é o seguinte: a partir de um arquivo de áudio de fala (.WAV) e sua correspondente transcrição ortográfica em um arquivo de texto (.TXT), o usuário deve passar por 3 etapas automáticas; verificações e ajustes manuais podem ser feitas para garantir uma qualidade ainda melhor. O resultado é um TextGrid multicamadas com fones, sílabas, palavras e segmentação de enunciados como na Figura 21.

Mais precisamente, essas três etapas são (GOLDMAN, 2011):

1. Macro-segmentação (*Macro-segmentation*) no nível de enunciado: torna o nível orto;
2. Conversão de grafema em fonema (*Phonetization*): torna a camada fono;

[1] http://www.voxforge.org/pt

Figura 21 – Processo de segmentação de um arquivo de fala no EasyAlign

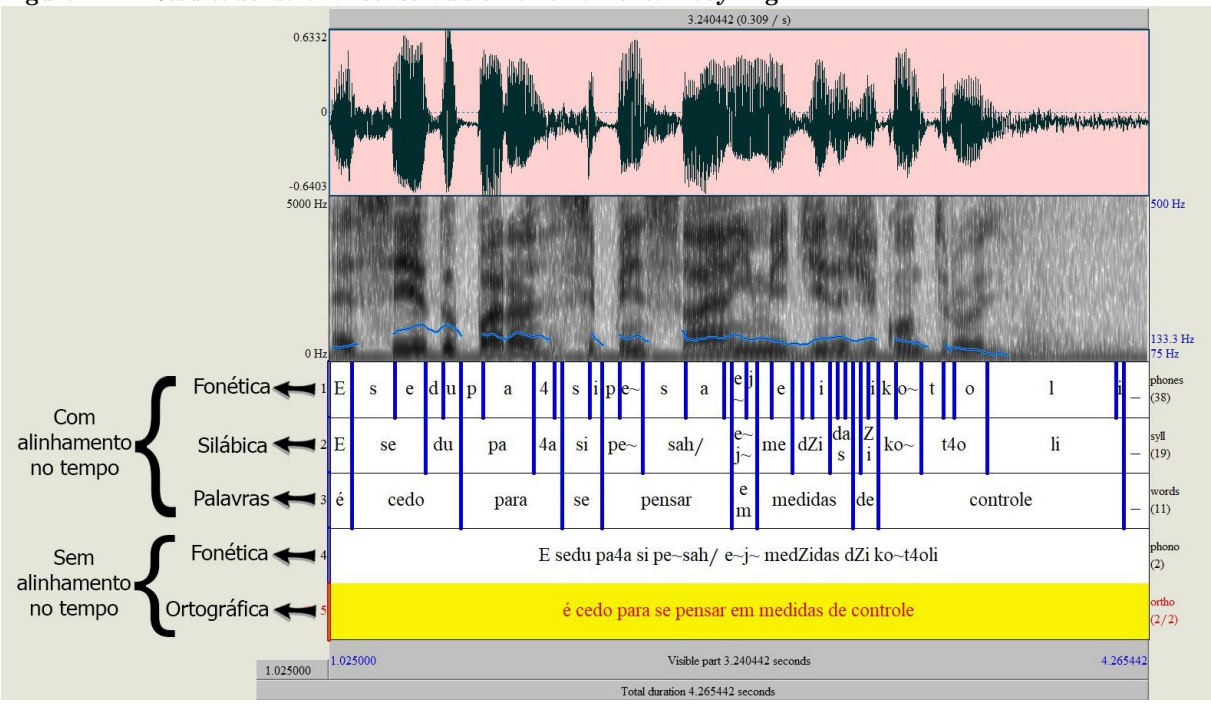


Fonte: Autoria própria.

3. Segmentação de fone (*Phone segmentation*): criar camadas de palavras, sílabas e fones.

A Figura 22 apresenta o sinal digital e as transcrições da frase "é cedo para se pensar em medidas de controle". Pode-se observar que existem 3 formas de transcrições do áudio alinhadas no tempo: 1) fonética; 2) silábica; 3) palavras. As últimas duas formas (4 e 5) são as transcrições fonéticas sem alinhamento e a transcrição ortográfica respectivamente.

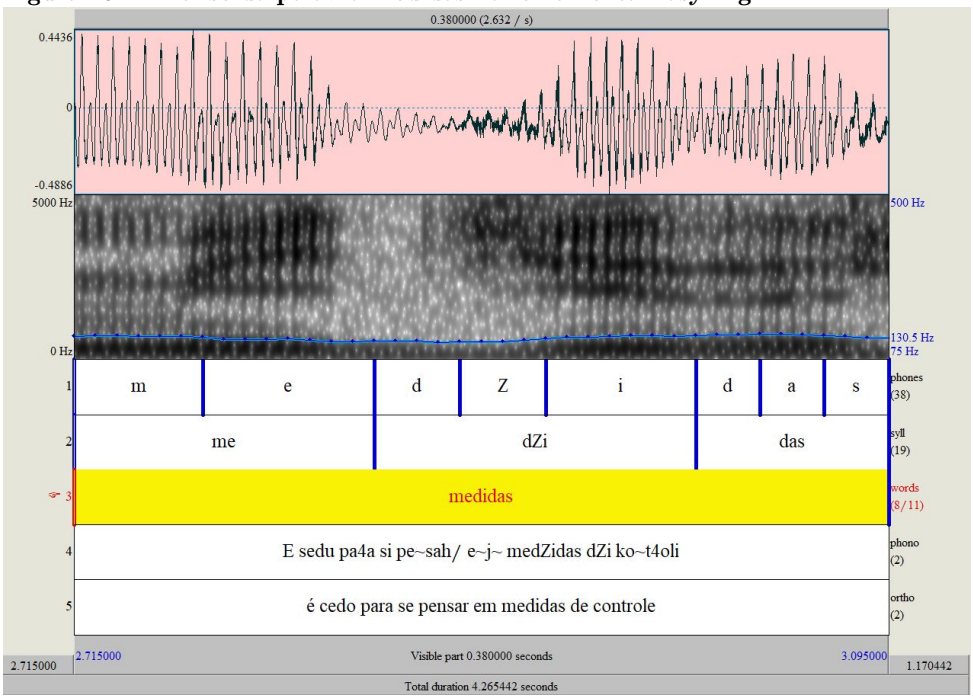
Figura 22 – Resultado da análise do áudio na ferramenta *EasyAlign*



Fonte: Autoria própria.

A Figura 23 apresenta de forma mais detalhada apenas a palavra "medidas". Dessa forma, pode-se observar as silabas e os fonemas no tempo. A Tabela 11 apresenta a transcrição fonética da frase da Figura 22 onde cada linha representa uma palavra do texto.

Figura 23 – Análise da palavra medidas na ferramenta *EasyAlign*



Fonte: Autoria própria.

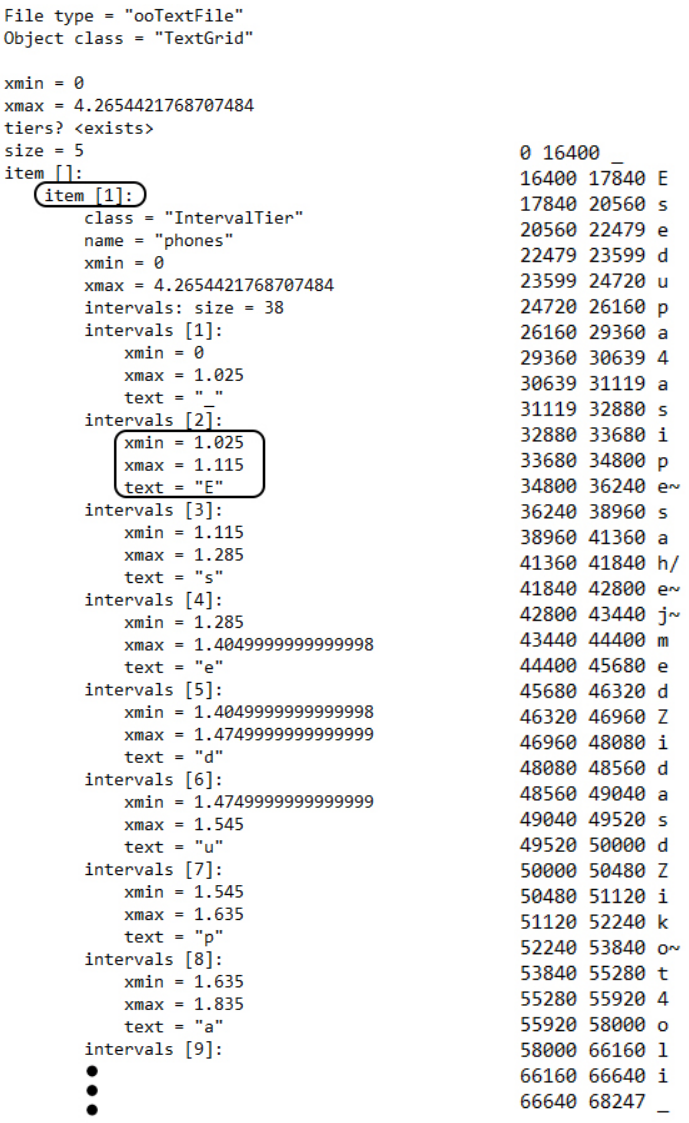
Tabela 11 – Transcrição fonética de cada palavra da frase

Palavra	Transcrição fonética (SAMPA)
é	E
cedo	sedu
para	pa4a
se	si
pensar	pe∼sah/
em	e∼j∼
medidas	medZidas
de	dZi
controle	ko∼t4oli

Fonte: Autoria própria.

O arquivo de saída com a transcrição fonética, gerado pelo *EasyAlign*, é diferente do modelo da base TIMIT, cujo alinhamento dos fonemas é por *frames*. A Figura 24(a) exemplifica a saída do *EasyAlign*, possuindo múltiplos itens que são as transcrições 1, 2, 3, 4 e 5. A parte desejada se encontra no item [1], que representa a transcrição fonética alinhada com o tempo. Pode-se observar que cada intervalo do fonema é indicado pelas variáveis *xmin* e *xmax* representando inicio e fim e o fonema desse intervalo está representado em *text*. Já na Figura 24(b) tem-se a conversão da saída para o modelo utilizado na base TIMIT, onde cada linha representa *frame* inicial, *frame* final e o fonema.

Figura 24 – Modificações da saída da ferramenta *EasyAlign*



```
File type = "ooTextFile"
Object class = "TextGrid"

xmin = 0
xmax = 4.2654421768707484
tiers? <exists>
size = 5
item []:
    item [1]:
        class = "IntervalTier"
        name = "phones"
        xmin = 0
        xmax = 4.2654421768707484
        intervals: size = 38
        intervals [1]:
            xmin = 0
            xmax = 1.025
            text = "_"
        intervals [2]:
            xmin = 1.025
            xmax = 1.115
            text = "E"
        intervals [3]:
            xmin = 1.115
            xmax = 1.285
            text = "s"
        intervals [4]:
            xmin = 1.285
            xmax = 1.4049999999999998
            text = "e"
        intervals [5]:
            xmin = 1.4049999999999998
            xmax = 1.4749999999999999
            text = "d"
        intervals [6]:
            xmin = 1.4749999999999999
            xmax = 1.545
            text = "u"
        intervals [7]:
            xmin = 1.545
            xmax = 1.635
            text = "p"
        intervals [8]:
            xmin = 1.635
            xmax = 1.835
            text = "a"
        intervals [9]:
        ●
        ●
        ●
```

(a)

```
0 16400 _
16400 17840 E
17840 20560 s
20560 22479 e
22479 23599 d
23599 24720 u
24720 26160 p
26160 29360 a
29360 30639 4
30639 31119 a
31119 32880 s
32880 33680 i
33680 34800 p
34800 36240 e~
36240 38960 s
38960 41360 a
41360 41840 h/
41840 42800 e~
42800 43440 j~
43440 44400 m
44400 45680 e
45680 46320 d
46320 46960 Z
46960 48080 i
48080 48560 d
48560 49040 a
49040 49520 s
49520 50000 d
50000 50480 Z
50480 51120 i
51120 52240 k
52240 53840 o~
53840 55280 t
55280 55920 4
55920 58000 o
58000 66160 l
66160 66640 i
66640 68247 _
```

(b)

Fonte: Autoria própria.

4.2 PROCESSAMENTO ACÚSTICO OU EXTRAÇÃO DE CARACTERÍSTICAS

Para o processamento acústico utilizou-se o *Kaldi Speech Recognition Toolkit*, uma ferramenta *open source* para o reconhecimento de fala feita em C++. O Kaldi possui alguns modelos de classificação de fala tal como modelo acústico baseado em GMM e extração de características de áudios como o MFCC e PLP (POVEY *et al.*, 2011). Como a implementação foi feita em Python, utilizou-se o Pytorch-Kaldi, que é pacote do Kaldi com o Python para se trabalhar com redes neurais utilizando o Pytorch. As funções como extração de características, processamento de rótulo e decodificação são realizadas pelo *Kaldi toolkit*.

Antes de aplicar essas técnicas de extração de características, foi padronizado

utilizar sempre áudios com uma frequência de 16000 Hz, quando necessário realizou-se a conversão.

Utilizou-se dois tipos diferentes de processamento dos áudios, sendo eles o MFCC e *filter banks*. O MFCC foi explicado na seção 2.2.2. A diferença entre os dois modelos de extração de características é que o *Filter Bank* não aplica a DCT.

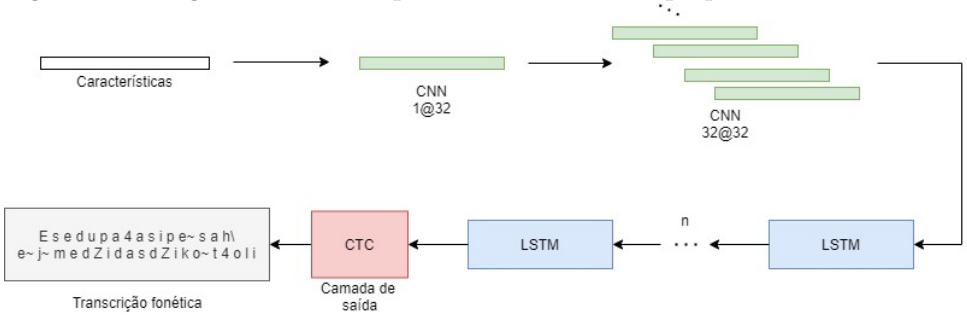
Como parâmetros para o *Filter Bank* utilizou-se a janela de Hamming com um tamanho de 25ms com uma taxa de *frames* de 10ms assim como em (LOHRENZ; LI; FINGSCHEIDT, 2018) aplicando 80 filtros de Mel.

Para o MFCC também utilizou-se a janela de Hamming com um tamanho de 25ms com uma taxa de *frames* de 10ms e 23 filtros de Mel sendo esse o padrão da ferramenta Kaldi.

4.3 MODELO DE REDE PROPOSTO

Este trabalho propõe um modelo de rede para identificar fonemas de acordo com as características extraídas de um áudio. Um grande desafio na identificação de fonemas é o fato de possuírem diferentes tamanhos (períodos de tempos). A Figura 25 apresenta o modelo de rede proposto, que possui como entrada de rede as características dos áudios. Na sequência ocorre o treinamento por uma sequência de duas redes convolucionais e então a saída dessas redes são processadas por um número de camadas LSTM. Nesse trabalho empregou-se dois, quatro e seis camadas LSTM em sequência (n). Por fim tem-se uma camada de saída CTC para obter a transcrição fonética final. Uma aplicação similar dessa técnica foi apresentada em Quintanilha (2017), porém em reconhecimento de fala.

Figura 25 – Diagrama com as etapas do modelo de rede proposto



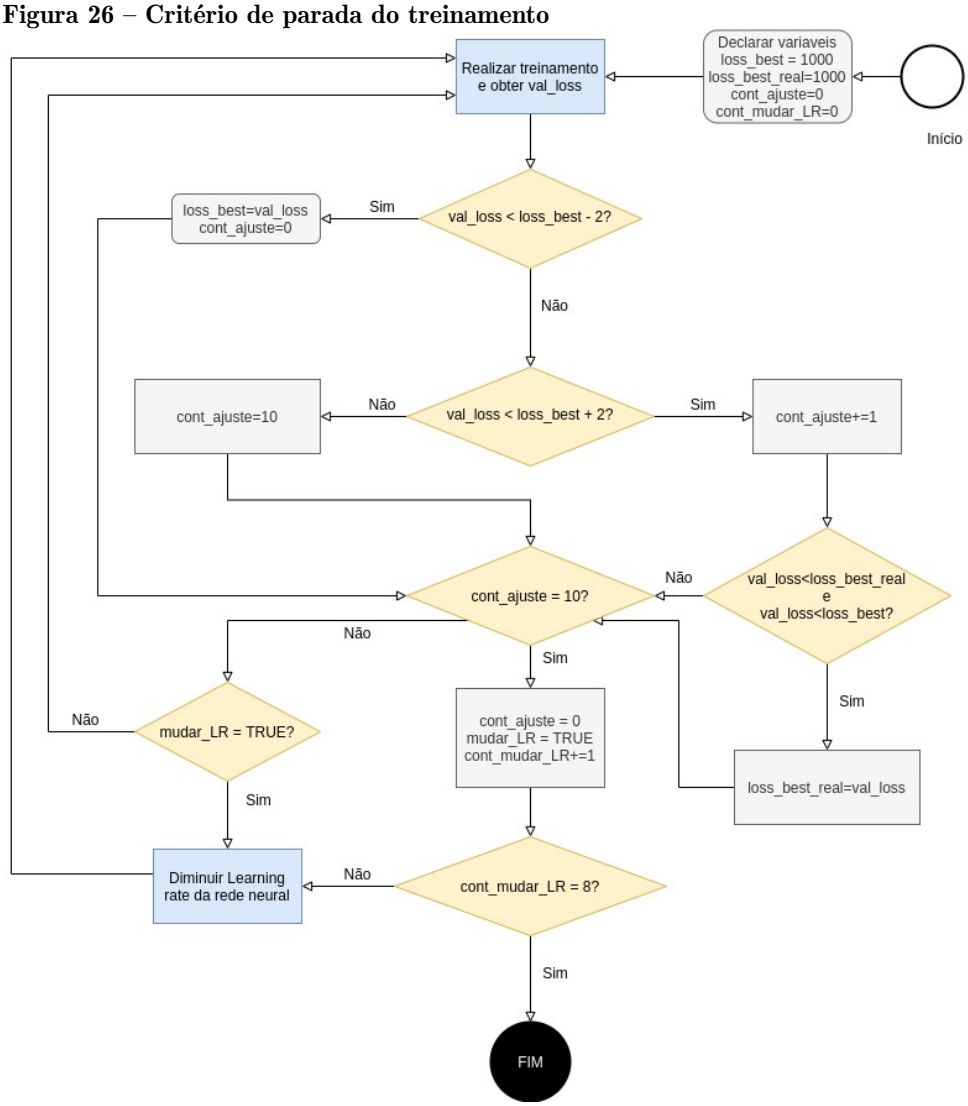
Fonte: Autoria própria.

Utilizou-se a CNN com o objetivo de reduzir a variedade do espectro dos vetores de características. Para isso empregou-se uma rede de CNN com duas camadas. Em seguida aplicou-se uma sequência de LSTMs para processar essas características e gerar uma sequência de saída utilizando a camada CTC, que tem como o objetivo decodificar cada saída das LSTM em uma sequência de fonemas sem que haja a repetição do mesmo, ao

menos que ela realmente exista. Isso se deve ao fato que o treinamento da rede é realizado por pequenos *frames* de áudios, obtendo uma possível saída para cada *frame*.

O critério de parada do treinamento ocorre de acordo com o valor da função de perda (*val_loss*) da camada de saída CTC. A Figura 26 ilustra o fluxograma mostrando como é feito o critério de parada do treinamento comparando os valores de *val_loss*. Caso ele comece a subir no meio do treinamento dado um valor de folga, nesse caso 2, é atualizado o parâmetro de contagem de alteração da taxa de aprendizagem da rede (*cont_mudar_LR*) para um valor menor. Quando o *cont_mudar_LR* chegar a 8 é finalizado o treinamento. O valor de taxa de aprendizagem utilizado foi de 0,001, decaindo 0,0005.

Figura 26 – Critério de parada do treinamento



Fonte: Autoria própria.

5 RESULTADOS

Neste capítulo serão apresentados os resultados obtidos através dos experimentos utilizando o modelo proposto para o reconhecimento de fonemas. As Seções 5.1, 5.2, 5.3 e 5.4 apresentam os resultados obtidos nas bases de áudios TIMIT, LaPS Benchmark 16k, Sid e a união das bases LaPS Benchmark 16k e Sid respectivamente. Na Seção 5.5 é apresentada uma análise e discussão dos resultados.

Foram executados 10 vezes o treinamento para cada uma das bases de áudios, calculando a média e o desvio padrão dos resultados obtidos.

5.1 TIMIT

Para o treinamento da base TIMIT, utilizou-se a divisão nativa da base de treino e testes, sendo que o conjunto de testes foi dividido entre conjunto do *Core test* e o conjunto de validação, como é apresentado na Seção 4.1.2.

Optou-se por mudar a configuração da rede proposta adicionando mais camadas LSTM, assim como alterar o método de extração de características entre *Filter Banks* e MFCCs.

A Tabela 12 apresenta as médias dos tempos para treinamento, número de épocas, taxas de erros de fonemas (PER - *phoneme error rate*) e seus desvios padrão obtidos no *core test* e no conjunto de validação (Val set) utilizando a rede proposta neste trabalho, considerando 10 execuções. Foram utilizadas duas, quatro e seis camadas LSTM juntamente com duas CNNs para a classificação das características dos áudios. Os melhores resultados médios foram obtidos utilizando seis LSTMs, porém o tempo de processamento foi maior pois a rede possui mais parâmetros para treinamento.

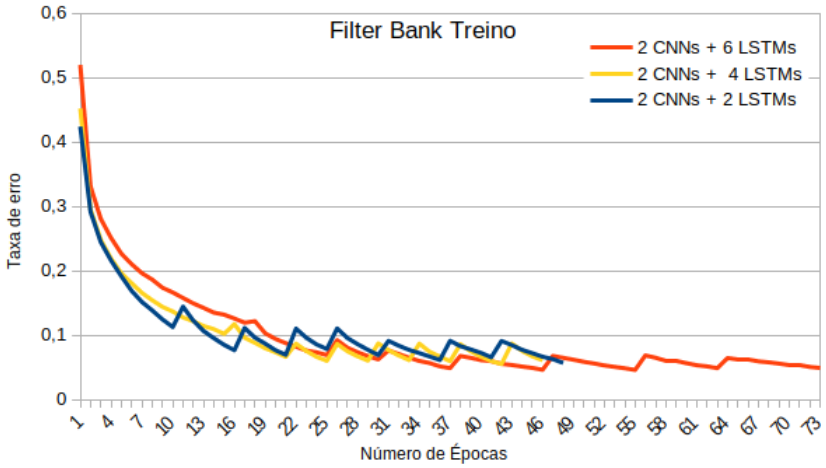
Tabela 12 – Taxas de erros utilizando *Filter Banks* na base TIMIT

Rede	Tempo para treinamento (min)	Número de Épocas	PER (%) Core test	PER (%) Val set
2 CNNs + 2 LSTMs	20,48 ± 3,54	46 ± 8	20.62 ± 0,18	18,40 ± 0,18
2 CNNs + 4 LSTMs	38,76 ± 4,86	52 ± 6	18,97 ± 0,23	16,70 ± 0,18
2 CNNs + 6 LSTMs	67,93 ± 8,71	62 ± 8	18,61 ± 0,37	16,44 ± 0,11

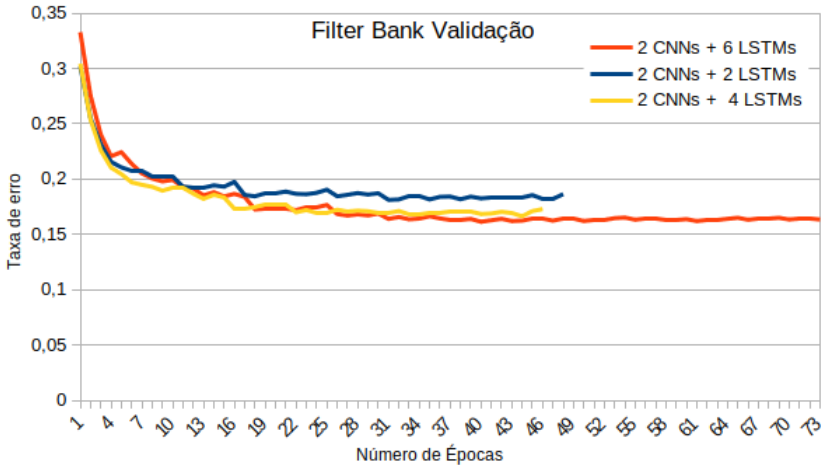
Fonte: Autoria própria.

A Figura 27 exemplifica os gráficos obtidos na execução de um dos treinamentos, apresentando as taxa de erros dos conjuntos de treinamento e validação. O objetivo é relatar o comportamento do treinamento durante as épocas. Os valores de épocas diferem entre eles pois o critério de parada de treinamento foram obtidas em tempos diferentes.

Figura 27 – Resultados da base TIMIT utilizando *filter banks*. (a) Gráfico da taxa de erro do conjunto de treino. (b) Gráfico da taxa de erro do conjunto de validação.



(a)



(b)

Fonte: Autoria própria.

Os resultados utilizando o MFCC apresentaram uma taxa de erro maior do que com os *filter banks*, como pode ser observado na Tabela 13.

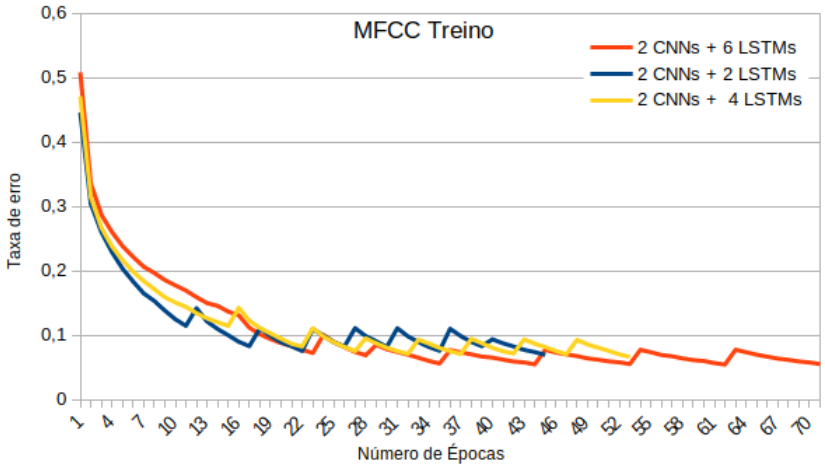
Tabela 13 – Taxa de erro utilizando MFCC na base TIMIT

Rede	Tempo para treinamento (min)	Número de Épocas	PER (%) Core test	PER (%) Val set
2 CNNs + 2 LSTMs	16,70 ± 3,19	56 ± 10	20,65 ± 1,90	18,67 ± 1,62
2 CNNs + 4 LSTMs	33,97 ± 4,13	55 ± 6	19,69 ± 0,34	17,47 ± 0,18
2 CNNs + 6 LSTMs	57,54 ± 8,57	60 ± 9	19,59 ± 0,33	17,43 ± 0,19

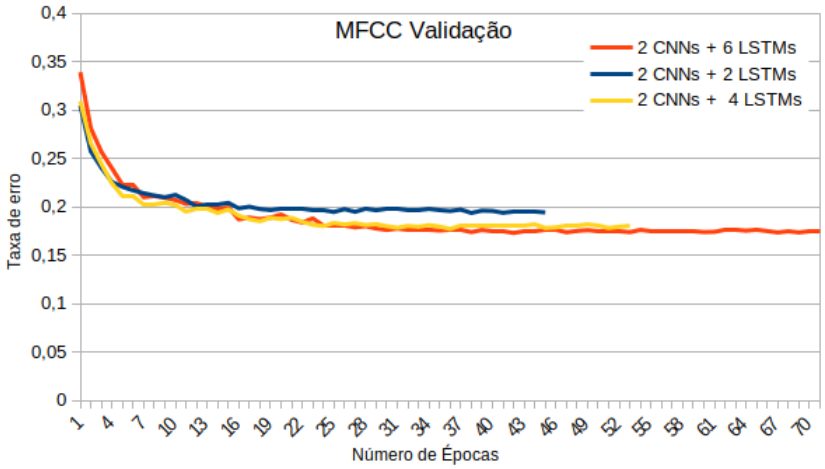
Fonte: Autoria própria.

A Figura 28(a) apresenta o gráfico de taxa de erro do conjunto de treino em uma das execuções e a Figura 28(b) do conjunto de validação utilizando o MFCC na base TIMIT.

Figura 28 – Resultados da base TIMIT utilizando MFCC. (a) Gráfico da taxa de erro do conjunto de treino. (b) Gráfico da taxa de erro do conjunto de validação.



(a)



(b)

Fonte: Autoria própria.

5.2 LAPS BENCHMARK 16K

Para o treinamento da base LaPS Benchmark 16K, realizou-se uma divisão entre os locutores de 60% para o conjunto de treino, 20% para o conjunto de validação e 20% para o conjunto de teste. Foram realizadas quatro distribuições diferentes para esses conjuntos, distribuídos de forma aleatória como foi previamente definido na Seção 4.1.3.

Como no experimento com a base TIMIT a melhor rede foi utilizando duas CNNs com seis camadas LSTM optou-se por utilizar esse modelo para realizar os experimentos com as demais bases. Foi realizado o treinamento das quatro distribuições obtendo uma da taxa de erro de fonemas média entre os conjuntos de $31,55\%$ utilizando *filter banks* e $33,04\%$ utilizando MFCC.

A Tabela 14 apresenta as médias dos tempos para treinamento, número de épocas, taxas de erros de fonemas (PER - *phoneme error rate*) e seus desvios padrão obtidos no conjunto de teste (*core test*) e no conjunto de validação (Val set) utilizando a rede proposta neste trabalho, considerando 10 execuções na base LaPS Benchmark 16k. Também é realizado uma média da taxa de erro entre os conjuntos por não haver uma distribuição de conjuntos predefinidos para a base, permitindo a obtenção de um resultado mais generalista do modelo. O melhor resultado entre todas as execuções foi do conjunto 4 com uma PER média de 30,62% no *core test*.

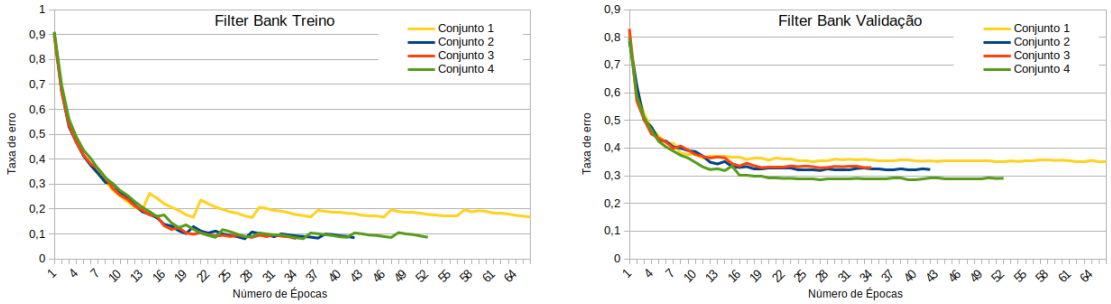
Tabela 14 – Taxa de erro utilizando *filter banks* nos conjuntos da base LaPS Benchmark 16k

Rede	Tempo para treinamento (min)	Número de Épocas	PER (%) Core test	PER (%) Val set
Conjunto 1	6,64 ± 1,73	40 ± 12	31,74 ± 0,71	33,62 ± 0,66
Conjunto 2	7,09 ± 1,52	42 ± 9	32,94 ± 0,52	32,15 ± 0,34
Conjunto 3	5,49 ± 0,62	34 ± 5	30,96 ± 0,87	34,53 ± 1,32
Conjunto 4	8,01 ± 1,76	47 ± 11	30,62 ± 0,41	28,57 ± 0,71
Média	6,81 ± 1,69	41 ± 10	31,56 ± 1,10	32,22 ± 2,43

Fonte: Autoria própria.

A Figura 29(a) apresenta o gráfico da PER nos conjuntos de treino e a Figura 29(b) nos conjuntos de validação.

Figura 29 – Resultados da base LaPS utilizando *filter banks*. (a) Gráfico da taxa de erro dos conjuntos de treino. (b) Gráfico da taxa de erro dos conjuntos de validação



(a) (b)

Fonte: Autoria própria.

A Tabela 15 apresenta as médias e os desvios padrão obtidos utilizando MFCC para extrair as características dos áudios. O melhor resultado no teste (*Core test*) foi uma PER de 31,60% obtido no conjunto 4. A média da taxa de erro de fonemas dos quatro conjuntos foi de 33,04%.

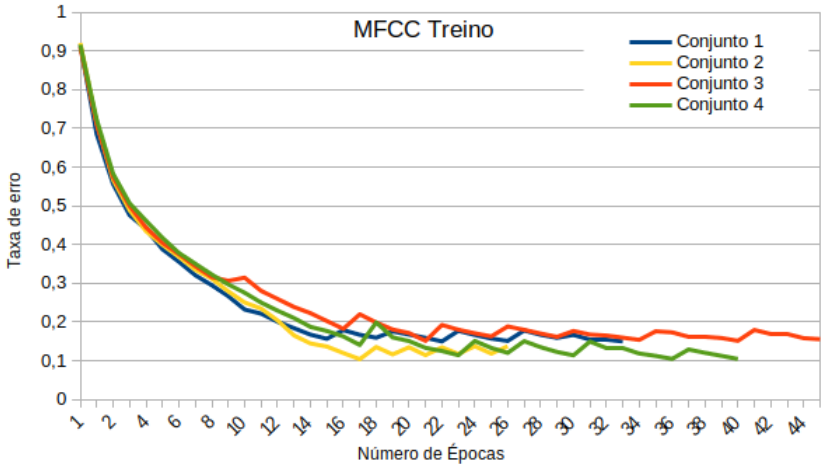
A Figura 30(a) apresenta o gráfico da PER de treino e a Figura 30(b) do conjunto de validação da base LaPS benchmark 16k utilizando MFCC.

Tabela 15 – Taxa de erro utilizando MFCC nos conjuntos da base LaPS Benchmark 16k

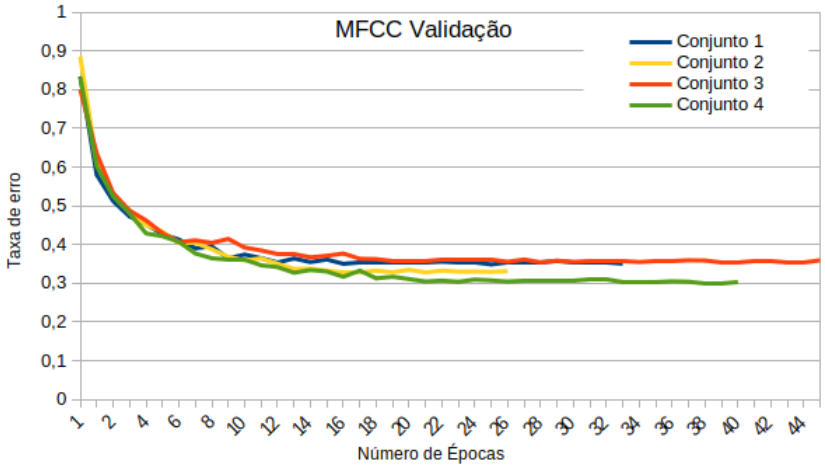
Rede	Tempo para treinamento (min)	Número de Épocas	PER (%) Core test	PER (%) Val set
Conjunto 1	5,88 ± 0,84	40 ± 6	33,96 ± 0,40	34,49 ± 0,64
Conjunto 2	6,41 ± 1,74	44 ± 12	34,47 ± 0,65	32,91 ± 0,90
Conjunto 3	6,04 ± 1,15	42 ± 9	31,86 ± 0,48	35,79 ± 0,39
Conjunto 4	8,01 ± 1,54	55 ± 12	31,87 ± 0,29	29,56 ± 0,50
Média	6,59 ± 1,56	45 ± 11	33,04 ± 1,28	32,19 ± 1,60

Fonte: Autoria própria.

Figura 30 – Resultados da base LaPS utilizando MFCC. (a) Gráfico da taxa de erro do conjunto de treino. (b) Gráfico da taxa de erro do conjunto de validação



(a)



(b)

Fonte: Autoria própria.

5.3 BASE SID

Para o treinamento da base Sid, realizou-se uma divisão de 60% para o conjunto de treino, 20% para o conjunto validação e 20% para o conjunto de teste. Sendo que para os conjuntos de testes e validação utilizou-se locutores com 10 áudios. A divisão é apresentada mais detalhadamente na Seção 4.1.4.

Os experimentos foram realizados utilizando a rede de duas CNNs com seis camadas LSTM. A Tabela 16 apresenta as médias dos tempos para treinamento, número de épocas, taxas de erros de fonemas (PER - *phoneme error rate*) e seus desvios padrão obtidos no *core test* e no conjunto de validação (Val set) utilizando a rede proposta neste trabalho, considerando 10 execuções. Para a extração de características dos áudios, foi utilizado *filter banks* e MFCC. O melhor resultado obtido foi uma média de 28,81% de taxa de erro para os conjuntos de testes.

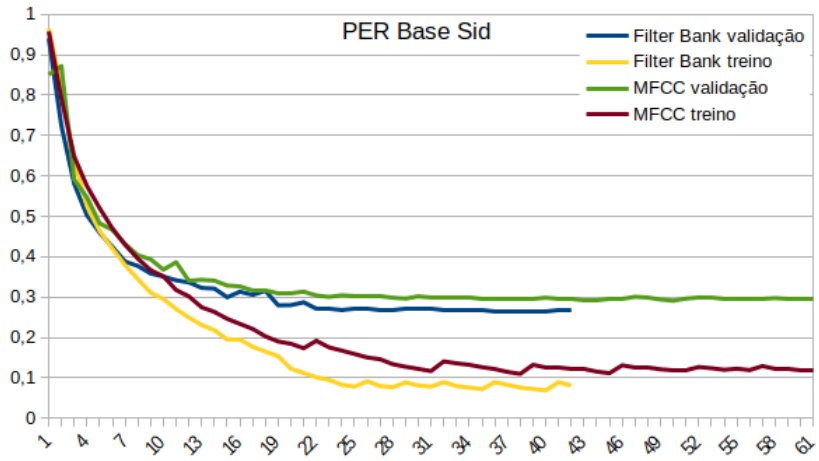
Tabela 16 – Taxa de erro de fonemas na base Sid

	Tempo para treinamento (min)	**Número de Épocas**	**PER (%) Core test**	**PER (%) Val set**
Filter banks	14,32 ± 2,93	57 ± 13	28,81 ± 0,70	27,60 ± 0,77
MFCC	11,10 ± 2,10	52 ± 10	29,67 ± 0,25	28,72 ± 0,49

Fonte: Autoria própria.

A Figura 31 apresenta o gráfico de taxa de erro dos conjunto de validação e treino obtido durante o treinamento da rede em uma das execuções.

Figura 31 – Resultados da base Sid usando *filter banks* e MFCC



Fonte: Autoria própria.

5.4 UNIÃO DAS BASES LAPS BENCHMARK 16K COM SID

Da base LaPS Benchmark 16k foram utilizados os 700 áudios e da base Sid utilizou-se os 700 áudios transcritos, totalizando 1400 arquivos de áudio de fala.

Para os treinamentos e testes optou-se por utilizar o conjunto 4 da base LaPS Benchmark 16k e unir com a base Sid. Foi mantido a organização dos locutores para os conjuntos de treino, validação e teste, ou seja, o conjunto de treino da base sid foi unida com o conjunto de treino da base LaPS Benchmark 16k.

As médias dos tempos para treinamento, número de épocas, taxas de erros de fonemas (PER - *phoneme error rate*) e seus desvios padrão obtidos no *core test* e no conjunto de validação (Val set) utilizando a rede proposta neste trabalhos são apresentadas na Tabela 17. Foram realizadas 10 execuções da rede utilizando duas CNNs com seis camadas LSTMs. Para a extração de características dos áudios, foi utilizado *filter banks* e MFCC, sendo que com *filter banks* também obteve uma média de taxa de erro menor em relação ao MFCC.

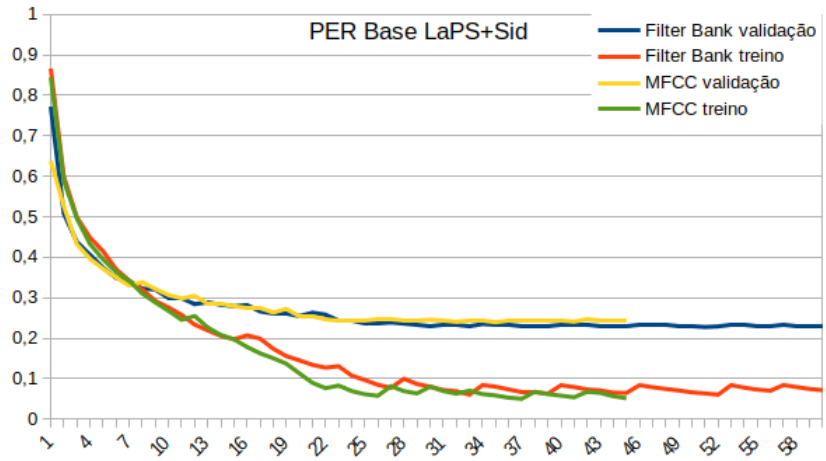
Tabela 17 – Taxa de erro de fonemas na união das bases LaPS Benchmark 16k com Sid

	Tempo para treinamento (min)	**Número de Épocas**	**PER (%) Core test**	**PER (%) Val set**
Filter banks	27,70 ± 4,02	65 ± 10	25,49 ± 0,44	22,90 ± 1,74
MFCC	20,54 ± 3,62	54 ± 10	26,20 ± 0,31	24,69 ± 0,80

Fonte: Autoria própria.

A Figura 32 apresenta o gráfico de taxa de erro dos conjunto de validação e treino obtido durante o treinamento da rede em uma das execuções.

Figura 32 – Resultado da base LaPS Benchmark 16k mesclada com a Sid

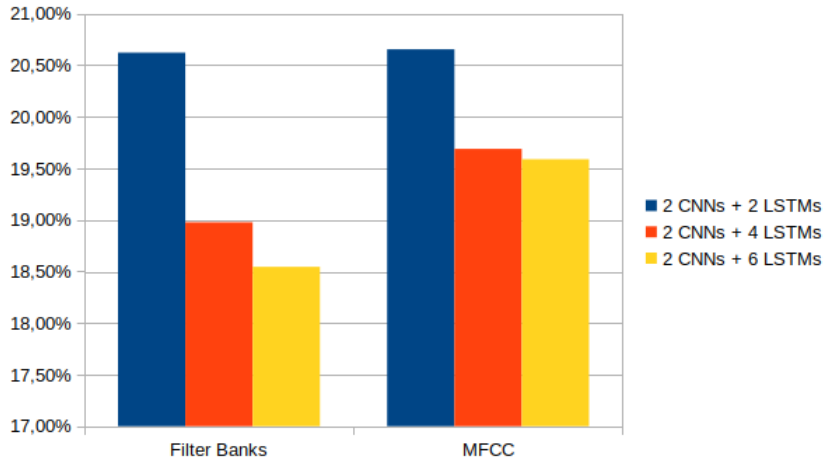


Fonte: Autoria própria.

5.5 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Com a base TIMIT foram realizados testes com diferentes combinações de rede, alterando o número de camadas LSTM, sendo que o melhor resultado foi obtido utilizando seis camadas. A Figura 33 apresenta um gráfico comparando as médias obtidas de taxas de erros no *core test*. Pode-se observar que o melhor resultado foi obtido utilizando *filter banks* para extrair as características dos áudios com a rede de 2 camadas CNNs e 6 LSTMs. Utilizando o MFCC a média entre as redes que utilizaram 4 e 6 camadas LSTM, apresentaram uma diferença de apenas 0,10%, sendo que a menor taxa de erros obtidas também foi utilizando a rede com 2 camadas CNNs e 6 LSTMs.

Figura 33 – Comparação entre os resultados obtidos utilizando diferentes combinações de redes na base TIMIT



Fonte: Autoria própria.

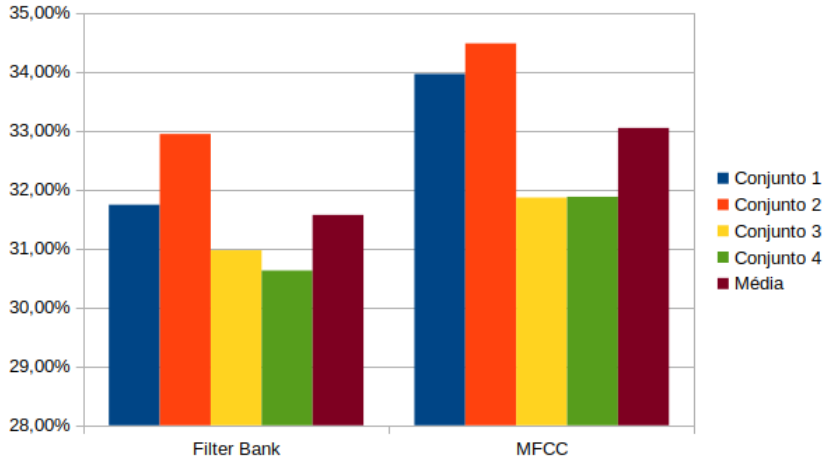
O melhor resultado obtido entre todas as execuções com *filter banks* no *core test* da base TIMIT foi de $18,11\%$ utilizando seis camadas LSTM, com o treinamento finalizando em 58 épocas. No conjunto de validação (Val set) o melhor resultado obtido foi uma PER de 16,36%. Utilizando o MFCC o melhor resultado obtido de taxa de erro no conjunto *core test* foi de 19,04%.

Outro detalhe observado com os experimentos da base TIMIT, foi a relação entre os tempos de execução. A técnica de extração de característica *filter banks* fez com que a rede demorasse mais para finalizar o treinamento em relação ao MFCC. Essa observação se manteve para todas as bases, porém com uma diferença menor devido ao fato das bases possuírem menos áudios.

A base LAPS Benchmark 16k foi a base que apresentou os piores resultados dos testes realizado nesse trabalho. Ela possui apenas 700 áudios e sem ambiente controlado, tornando mais difícil o processo de treinamento para a generalização da rede. A Figura 34 apresenta uma comparação das médias obtidas das execuções da rede entre as quatro distribuições dos conjuntos e a média entre essas distribuições. Pode-se observar que os

melhores resultados foram apresentados nos conjuntos 3 e 4, porém todos com uma PER média acima de 30%.

Figura 34 – Comparação entre os resultados obtidos na base LaPS Benchmark 16k



Fonte: Autoria própria.

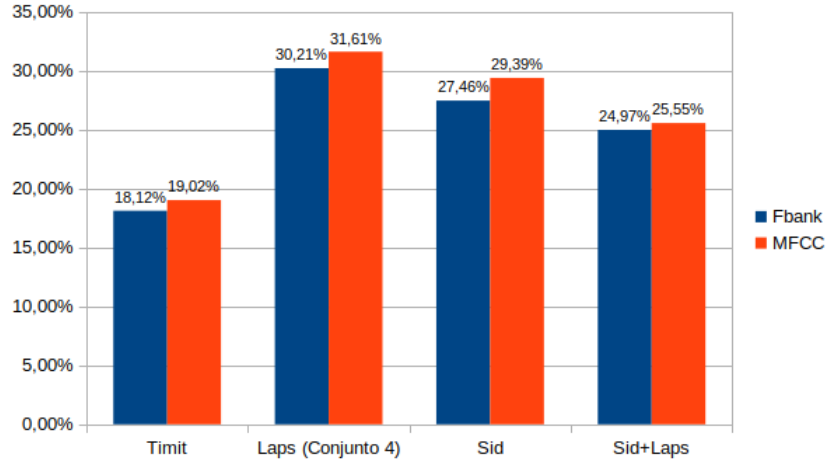
O melhor resultado da LaPS Benchmark 16k foi utilizando o conjunto 4 com a técnica de extração de características *filter banks*, apresentando uma PER de 30,20% no conjunto de testes e de 28,08% no conjunto de validação.

A base Sid apresentou uma PER média de 28,81% para o conjunto de testes e 27,60% no conjunto de validação, utilizando os *filter banks*. Essa taxa se mostrou melhor comparado com a base LaPS, porém cometeu mais erros comparados com a base TIMIT. Os treinamentos também foram realizados com 700 áudios, sendo que o melhor resultado entre todas as execuções foi uma PER de 27,45% e 26,62% nos conjuntos de teste e validação respectivamente, utilizando *filter banks*.

Unindo as bases Sid e LaPS Benchmark 16k, foi possível melhorar os resultados. A base ficou com um total de 1400 áudios apresentando uma PER média de 25,49% para o conjunto de testes e 22,90% no conjunto de validação utilizando os *filter banks*. Pode-se observar que os resultados foram menores que de ambas bases separadamente, mostrando que a adição de mais áudios e locutores nesse caso auxiliou a obter uma taxa de erros mais baixas. O melhor resultado entre todas as execuções foi uma PER de 24,96% e 22,76% nos conjuntos de testes e validação respectivamente, utilizando *filter banks*. Com o MFCC o melhor resultado foi uma PER de 25,54% e 23,93% para os conjuntos de teste e validação respectivamente.

A Figura 35 apresenta uma comparação dos melhores resultados obtidos em cada base de áudios nos conjuntos de testes (*core test*).

Figura 35 – Comparação entre os melhores resultados obtidos nas bases de áudios



Fonte: Autoria própria.

A Tabela 18 apresenta, como exemplo, cinco transcrições fonéticas automáticas obtidas utilizando o método proposto, sendo a frase 1 e 2 áudios da base TIMIT, a frase 3 da base LaPS Benchmark 16k e as frases 4 e 5 da base Sid. A linha Verdade é a transcrição fonética da base enquanto a Saída da Rede é a transcrição resultante. Os caracteres que foram classificados de forma incorreta estão marcados em negrito.

Tabela 18 – Detalhamento de resultados obtidos pela rede

	Transcrição
Frase 1	Books are for schnooks.
Verdade	sil b uh sil k s ah f ah sh sil n uh sil k s sil
Saída da Rede	sil b **ah** sil k s ah f ah sh n uh sil k s sil
Frase 2	He will allow a rare lie.
Verdade	sil hh iy w l ah l aw er r eh r l ay sil
Saída da Rede	sil hh iy w l ah l aw er r eh **er** l ay sil
Frase 3	é cedo para se pensar em medidas de controle
Verdade	_ E s e d u p a 4 a s i p e~ s a h/ e~ j~ m e d Z i d a s d Z i k o~ t 4 o l i _
Saída da Rede	_ E s **i** d u p a 4 a **h/ i** s i p e~ s a **i~** m **i** d Z i **4 u** d Z i k o~ t **S i** 4 o **j v u** _
Frase 4	O mar inventa macacos.
Verdade	_ u m a h/ i~ v e~ t a m a k a k u s _
Saída da Rede	_ u m a h/ **i** v e~ t a m a k a k u s _
Frase 5	zero oito oito sete dois cinco sete cinco
Verdade	_ z e 4 u o j t u o j t u s E t S i d o j s s i~ k u s E t S i s i~ k u _
Saída da Rede	_ **s** e 4 u o **a** j t u o j t u s E t S i d o j s s i~ k s E t S i s i~ k u _

Fonte: Autoria própria.

É importante notar que na frase 3, o fonema *e* é identificado como *i*, isso pode ser pelo fato de o locutor falar de forma menos clara o *e*, algo comum na fala de locutores, muitas vezes pronunciando eles de forma muito similar.

5.5.1 Comparação com o Estado da Arte

Essa comparação foi realizada apenas com a base TIMIT, pois não foram encontrados trabalhos utilizando a LaPS benchmark 16k ou Sid para o reconhecimento de fonemas. Para comparação dos resultados com o estado da arte utilizou a métrica de taxa de erro de reconhecimento de fonemas (PER), que é a mesma utilizada nos artigos que utilizam a base TIMIT. A Tabela 19 apresenta uma comparação entre o método proposto e os trabalhos identificados na seção 3.3. Observa-se que utilizando LSTMs juntamente com as CNNs melhoraram os resultados que foram apresentados por Abdel-Hamid *et al.* (2014) que também utilizaram uma técnica de classificação de fonemas utilizando CNN e CTC. Os resultados foram melhores que os apresentado por Lohrenz, Li e Fingscheidt (2018) onde foi utilizando uma técnica de *deep leaning* com o turbo *fusion* entre CNN e DNN e da técnica de reconhecimento apresentado pelos autores Ager, Cvetković e Sollich (2010) que foi a melhor taxa identificada entre os trabalhos selecionados na Seção 3.2.

Tabela 19 – Comparação da taxas de erros de fonemas (PER) com outros no *core test*

Referências dos trabalhos	**PER**
Método Proposto	18,11%
(AGER; CVETKOVIĆ; SOLLICH, 2010)	18,50%
(LOHRENZ; LI; FINGSCHEIDT, 2018)	19,46%
(SELTZER; DROPPO, 2013)	20,25%
(ABDEL-HAMID *et al.*, 2014)	20,39%

Fonte: Autoria própria.

6 CONCLUSÃO

Este trabalho apresentou um modelo para reconhecimento automático de fonemas. Foram realizados treinamentos e testes na base em inglês TIMIT *Acoustic-Phonetic Continuous Speech Corpus* e em bases em português brasileiro com o objetivo de possibilitar pesquisas que necessitam da utilização de fonemas, como a identificação de desvio de fala em crianças, identificação de dialetos e locutores, identificação de erros de pronúncia e reconhecimento de emoções.

As bases identificadas foram a TIMIT, CSLU: Spoltech Brazilian Portuguese Version 1.0, LaPS Benchmark 16k, Sid, Voxforge entre outras. Sendo que as utilizadas nesse trabalho foram a TIMIT, LaPS Benchmark 16k e Sid. Foi necessário realizar a adição de transcrições fonéticas nas bases Sid e LaPS Benchmark 16k pois elas possuíam apenas as transcrições ortográficas. Para isso, utilizou-se o software Praat com o plugin EasyAlign de forma semiautomática com o objetivo de facilitar esse processo.

Para extrair as características dos áudios utilizou-se as técnicas de MFCC e *filter banks*. Pode-se observar uma melhor taxa de acertos para o reconhecimento de fonemas utilizando os *filter banks*.

O modelo proposto consiste de duas camadas de CNNs utilizadas para reduzir a variedade do espectro dos vetores de características dos áudios e uma sequência de camadas LSTM para classificar esses fonemas com uma camada de saída CTC para obter um caminho de fonemas que representa o áudio.

A união das bases Sid e LaPS Benchmark 16k, possibilitou obter resultados melhores que em relação as bases separadas. A taxa de erro média no conjunto de teste foi de $24,96\%$ utilizando *filter banks* e $25,54\%$ utilizando MFCC.

O modelo também foi treinado e testado utilizando a base TIMIT que é muito utilizado para comparações de vários sistemas de reconhecimento de fala devido ao fato de possuir vários tipos de transcrições, como pode ser observado no estado da arte. Outra vantagem é que ela possui o alinhamento fonético dos fonemas, possibilitando estudos envolvendo reconhecimento de fonemas. Os resultados obtidos com a base TIMIT foram melhores em comparação a outros trabalhos pesquisados. Com o método proposto obteve uma taxa de erro de fonemas no *core test* de $18,11\%$ utilizando *filter banks* e de $19,04\%$ com MFCC.

6.1 TRABALHOS FUTUROS

A seguir, são apresentadas algumas sugestões para trabalhos futuros que permitirão expandir o trabalho realizado. São elas:

- Realizar otimizações com o objetivo de melhorar a taxa de acerto;
- Aumentar a base de transcrição fonética transcrevendo mais áudios da base Sid e outras bases existentes para diminuir a taxa de erro de fonemas;
- Realizar testes com bases adquiridas em ambientes controlado;
- Desenvolver um modelo de identificação de desvio de falas para auxiliar na área de fonoaudiologia;
- Realizar testes com profissionais da área de fonoaudiologia utilizando da transcrição fonética automática apenas recebendo o áudio como entrada;
- Fazer com que o reconhecimento dos fonemas seja feito em tempo real possibilitando acompanhar os fonemas enquanto o locutor fala.

REFERÊNCIAS

ABDEL-HAMID, Ossama *et al.* Convolutional neural networks for speech recognition. **IEEE/ACM Transactions on audio, speech, and language processing**, IEEE, v. 22, n. 10, p. 1533–1545, 2014.

AGER, Matthew; CVETKOVIĆ, Zoran; SOLLICH, Peter. High-dimensional linear representations for robust speech recognition. In: IEEE. **2010 Information Theory and Applications Workshop (ITA)**. [S.l.], 2010. p. 1–5.

AKHILA, KS; KUMARASWAMY, R. Comparative analysis of kannada phoneme recognition using different classifiers. In: IEEE. **2015 International Conference on Trends in Automation, Communications and Computing Technology (I-TACT-15)**. [S.l.], 2015. p. 1–6.

ALMISREB, Ali Abd; ABIDIN, Ahmad Farid; TAHIR, Nooritawati Md. Arabic phonemes recognition system based on malay speakers using neural network. In: IEEE. **2014 IEEE Symposium on Wireless Technology and Applications (ISWTA)**. [S.l.]. p. 188–192.

_____. Arabic letters corpus based malay speaker-independent. In: IEEE. **2013 IEEE 3rd International Conference on System Engineering and Technology**. [S.l.], 2013. p. 232–236.

AROUS, Jamil; AYED, Dorra Ben; ELLOUZE, Noureddine. Cooperative static and dynamic neural networks for phoneme recognition. In: IEEE. **2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)**. [S.l.], 2012. p. 847–851.

ATSONIOS, Ioannis. Speaker independent robust phoneme recognition using higher-order statistics and entropic-based features in adverse environments. In: IEEE. **2011 11th International Conference on Hybrid Intelligent Systems (HIS)**. [S.l.], 2011. p. 662–667.

BEZERRA, Sabrina Guimarães Tavares de Andrade. Reservoir computing com hierarquia para previsão de vazões médias diárias. 2016.

BIJANKHAN, M; SHEIKHZADEGAN, J; ROOHANI, MR. Farsdat-the speech database of farsi spoken language. In: PROCCEDINGS AUSTRALIAN CONFERENCE ON SPEECH SCIENCE AND TECHNOLOGY. [S.l.], 1994.

BILBAO, M Asunción Moreno *et al.* Albayzin speech database: Design of the phonetic corpus. In: . EUROSPEECH. **EUROSPEECH 1993: 3rd European Conference on Speech Communication and Technology: Berlin, Germany: September 22-25, 1993**. [S.l.], 1993. p. 175–178.

BOERSMA, Paul. Praat: doing phonetics by computer [computer program]. Versão 6.1.08, 2020. Disponível em: <http://www.praat.org/>. Acesso em: 20 de Outubro de 2020.

BRIDLE, John S. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: **Neurocomputing**. [S.l.]: Springer, 1990. p. 227–236.

BRIGADE, Data Science. **A Diferença Entre Inteligência Artificial, Machine Learning e Deep Learning**. 2016. Disponível em: <https://medium.com/data-science-brigade/a-diferen\%C3\%A7a-entre-intelig\%C3\%AAncia-artificial-machine-learning-e-deep-learning-930b5cc2aa42>. Acesso em: 03 de Outubro de 2019.

CARDONA, Diana A Bonilla; NEDJAH, Nadia; MOURELLE, Luiza M. Online phoneme recognition using multi-layer perceptron networks combined with recurrent non-linear autoregressive neural networks with exogenous inputs. **Neurocomputing**, Elsevier, v. 265, p. 78–90, 2017.

CARUANA, Rich. Multitask learning. **Machine learning**, Springer, v. 28, n. 1, p. 41–75, 1997.

CARVALHO, André *et al.* Inteligência artificial–uma abordagem de aprendizado de máquina. **Rio de Janeiro: LTC**, 2011.

CRYSTAL, David. **A dictionary of linguistics and phonetics**. [S.l.]: John Wiley & Sons, 2011. v. 30.

CUTAJAR, Michelle *et al.* Neural network architectures for speaker independent phoneme recognition. In: IEEE. **2011 7th International Symposium on Image and Signal Processing and Analysis (ISPA)**. [S.l.], 2011. p. 90–94.

DABBAGHCHIAN, Saeed *et al.* Robust phoneme recognition using mlp neural networks in various domains of mfcc features. In: IEEE. **2010 5th international symposium on telecommunications**. [S.l.], 2010. p. 755–759.

DANESHVAR, Mohammad; VEISI, Hadi. Persian phoneme recognition using long short-term memory neural network. In: IEEE. **2016 Eighth International Conference on Information and Knowledge Technology (IKT)**. [S.l.], 2016. p. 111–115.

DAVE, Namrata. Feature extraction methods lpc, plp and mfcc in speech recognition. **International journal for advance research in engineering and technology**, v. 1, n. 6, p. 1–4, 2013.

DELCROIX, Marc *et al.* Context adaptive deep neural networks for fast acoustic model adaptation. In: IEEE. **2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2015. p. 4535–4539.

DENG, Li; YU, Dong *et al.* Deep learning: methods and applications. **Foundations and Trends® in Signal Processing**, Now Publishers, Inc., v. 7, n. 3–4, p. 197–387, 2014.

DENNIS, Jonathan *et al.* A discriminatively trained hough transform for frame-level phoneme recognition. In: IEEE. **2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2014. p. 2514–2518.

DIJKSTRA, Bauke Alfredo; SANCHES, Ionildo José. Transcrição fonética de bases de áudio e reconhecimento de fonemas. In: **Anais do I Congresso Brasileiro Interdisciplinar em Ciência e Tecnologia. Anais...** Diamantina(MG) Online: [s.n.], 2020. Disponível em: <https//www.even3.com.br/anais/icobicet2020/268421-TRANSCRICAO-FONETICA-DE-BASES-DE-AUDIO-E-RECONHECIMENTO-DE-FONEMAS>. Acesso em: 01 de Outubro de 2019.

DINIZ, Suelaine S; THOMé, Antonio C G. Uso de técnicas neurais para o reconhecimento de comandos à voz. **Rio de Janeiro: IME**, 1997.

FRIKHA, Mondher *et al.* Advanced classification approach for neuronal phoneme recognition system based on efficient constructive training algorithm. **International Journal of Speech Technology**, Springer, v. 16, n. 3, p. 273–284, 2013.

GANAPATHY, Sriram; THOMAS, Samuel; HERMANSKY, Hynek. Comparison of modulation features for phoneme recognition. In: IEEE. **2010 IEEE International Conference on Acoustics, Speech and Signal Processing**. [S.l.], 2010. p. 5038–5041.

GAROFOLO, John S. Timit acoustic phonetic continuous speech corpus. **Linguistic Data Consortium, 1993**, 1993.

GERS, F. A.; SCHMIDHUBER, J.; CUMMINS, F. Learning to forget: Continual prediction with lstm. **Neural Computation**, IET, v. 10, n. 12, p. 2451—-2471, 1999. Disponível em: <https://doi.org/10.1162/089976600300015015>. Acesso em: 20 de março de 2019.

GOLDMAN, Jean-Philippe. Easyalign: an automatic phonetic alignment tool under praat. 2011.

GOLIPOUR, Ladan; O'SHAUGHNESSY, Douglas. A segmental non-parametric-based phoneme recognition approach at the acoustical level. **Computer Speech & Language**, Elsevier, v. 26, n. 4, p. 244–259, 2012.

GONG, Yuan; POELLABAUER, Christian. **How do deep convolutional neural networks learn from raw audio waveforms?** 2018. Disponível em: <https://openreview.net/forum?id=S1Ow_e-Rb>.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep learning**. [S.l.]: MIT press, 2016.

GRAVES, Alex. **Supervised Sequence Labelling with Recurrent Neural Networks**. [S.l.]: Springer-Verlag Berlin Heidelberg, 2012. ISBN 978-3-642-24797-2.

GRAVES, Alex *et al.* Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: **Proceedings of the 23rd international conference on Machine learning**. [S.l.: s.n.], 2006. p. 369–376.

GULLI, Antonio; PAL, Sujit. **Deep Learning with Keras**. [S.l.]: Packt Publishing Ltd, 2017.

HAYKIN, Simon S *et al.* **Neural networks and learning machines/Simon Haykin.** [S.l.]: New York: Prentice Hall„ 2009.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997.

HUANG, Xuedong *et al.* **Spoken language processing: A guide to theory, algorithm, and system development**. [S.l.]: Prentice hall PTR, 2001.

JLASSI, Chiraz; AROUS, Najet; ELLOUZE, Noureddine. Phoneme recognition by means of a growing hierarchical recurrent self-organizing model based on locally adapting neighborhood radii. **Cognitive Computation**, Springer, v. 2, n. 3, p. 142–150, 2010.

KANNADAGULI, Prashanth; BHAT, Vidya. A comparison of gaussian mixture modeling (gmm) and hidden markov modeling (hmm) based approaches for automatic phoneme recognition in kannada. In: IEEE. **2015 International Conference on Signal Processing and Communication (ICSC)**. [S.l.], 2015. p. 257–260.

KANNADAGULI, Prashanth; THALENGALA, Ananthakrishna. Phoneme modeling for speech recognition in kannada using hidden markov model. In: IEEE. **2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)**. [S.l.], 2015. p. 1–5.

KAZEMI, AR; SOBHANMANESH, F. Mlp refined posterior features for noise robust phoneme recognition. **Scientia Iranica**, Elsevier, v. 18, n. 6, p. 1443–1449, 2011.

KHWAJA, Mohammed Kamal *et al.* Robust phoneme classification for automatic speech recognition using hybrid features and an amalgamated learning model. **International Journal of Speech Technology**, Springer, v. 19, n. 4, p. 895–905, 2016.

KIM, Jonghong; HWANG, Kyuyeon; SUNG, Wonyong. X1000 real-time phoneme recognition vlsi using feed-forward deep neural networks. In: IEEE. **2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2014. p. 7510–7514.

KOSIC, Dino. Neural network for single phoneme recognition based on mel-frequency cepstral coefficients coding. In: IEEE. **10th Symposium on Neural Network Applications in Electrical Engineering**. [S.l.], 2010. p. 191–192.

KOTWAL, Mohammed Rokibul Alam *et al.* Bangla phoneme recognition for asr using multilayer neural network. In: IEEE. **2010 13th International Conference on Computer and Information Technology (ICCIT)**. [S.l.], 2010. p. 103–107.

_____. Dpf-based japanese phoneme recognition using tandem mlns. In: IEEE. **2010 10th International Conference on Hybrid Intelligent Systems**. [S.l.], 2010. p. 209–212.

KSHIRSAGAR, Ashwini *et al.* Comparative study of phoneme recognition techniques. In: IEEE. **2012 Third International Conference on Computer and Communication Technology**. [S.l.], 2012. p. 98–103.

LECUN, Yann; BENGIO, Yoshua; HINTON, Geoffrey. Deep learning. **nature**, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.

LEE, K-F; HON, H-W. Speaker-independent phone recognition using hidden markov models. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, IEEE, v. 37, n. 11, p. 1641–1648, 1989.

LOGAN, Beth *et al.* Mel frequency cepstral coefficients for music modeling. In: **ISMIR**. [S.l.: s.n.], 2000. v. 270, p. 1–11.

LOHRENZ, Timo; LI, Wei; FINGSCHEIDT, Tim. Dnn/cnn acoustic model turbo fusion for phoneme recognition. In: VDE. **Speech Communication; 13th ITG-Symposium**. [S.l.], 2018. p. 1–5.

MEDIAVALET. **Artificial Intelligence for Marketers: Types of Machine Learning**. 2019. Disponível em: <https://www.mediavalet.com/blog/artificial-intelligence-marketers-types-machine-learning/>. Acesso em: 03 de Outubro de 2019.

MIRANDA, André LL *et al.* Cálculo de fasores com taxas não múltiplas da freqüência fundamental. **VII Seminário Técnico de Proteção e Controle (VII STPC), Rio de Janeiro, RJ, Brasil**, p. 22–27, 2003.

MIRHASSANI, Seyed Mostafa; TING, Hua Nong; GHARAHBAGH, Abdorreza Alavi. Fuzzy decision fusion of complementary experts based on evolutionary cepstral coefficients for phoneme recognition. **Digital Signal Processing**, Elsevier, v. 49, p. 116–125, 2016.

OPPENHEIM, AV; SCHAFER, RW. **Processamento em tempo discreto de sinais**. [S.l.]: São Paulo: Pearson Education do Brasil, 2012.

PAGANI, Regina Negri; KOVALESKI, João Luiz; RESENDE, Luis Mauricio. Methodi ordinatio: a proposed methodology to select and rank relevant scientific papers encompassing the impact factor, number of citation, and year of publication. **Scientometrics**, Springer, v. 105, n. 3, p. 2109–2135, 2015.

PALAZ, Dimitri; MAGIMAI-DOSS, Mathew; COLLOBERT, Ronan. Joint phoneme segmentation inference and classification using crfs. In: IEEE. **2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)**. [S.l.], 2014. p. 587–591.

PANAYOTOV, Vassil *et al.* Librispeech: an asr corpus based on public domain audio books. In: IEEE. **2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2015. p. 5206–5210.

PARKER, Steve. O livro do corpo humano. **São Paulo: Ciranda Cultural**, 2007.

Patil, P. P.; Pardeshi, S. A. Devnagari phoneme recognition system. In: **2014 Fourth International Conference on Advances in Computing and Communications**. [S.l.: s.n.], 2014. p. 5–8.

PICONE, Joseph W. Signal modeling techniques in speech recognition. **Proceedings of the IEEE**, IEEE, v. 81, n. 9, p. 1215–1247, 1993.

POVEY, Daniel *et al.* The kaldi speech recognition toolkit. In: IEEE SIGNAL PROCESSING SOCIETY. **IEEE 2011 workshop on automatic speech recognition and understanding**. [S.l.], 2011.

PRADEEP, R; RAO, K Sreenivasa. Deep neural networks for kannada phoneme recognition. In: IEEE. **2016 Ninth International Conference on Contemporary Computing (IC3)**. [S.l.], 2016. p. 1–6.

_____. Split acoustic modeling in decoder for phoneme recognition. In: IEEE. **2017 14th IEEE India Council International Conference (INDICON)**. [S.l.], 2017. p. 1–5.

QUINTANILHA, Igor Macedo. End-to-end speech recognition applied to brazilian portuguese using deep learning. **Ph. D. dissertation, MSc dissertation**, PEE/COPPE, Federal University of Rio de Janeiro, 2017.

RABINER, Lawrence; JUANG, Biing-Hwang. Fundamentals of speech processing. **Prantice Hall**, 1993.

RASCHKA, Sebastian. **Python machine learning**. [S.l.]: Packt Publishing Ltd, 2015.

ROBINSON, Tony *et al.* Wsjcam0 cambridge read news. **Linguistic Data Consortium, Philadelphia**, 1995.

SCHRAMM, M *et al.* **CSLU: Spoltech Brazilian Portuguese version 1.0 LDC2006S16**. [S.l.]: Philadelphia, 2006.

SEARA, Izabel Christine; NUNES, Vanessa Gonzaga; LAZZAROTTO-VOLCÃO, Cristiane. **Fonética e fonologia do português brasileiro: 2º período**. [S.l.]: Florianópolis: LLV/CCE/UFSC, 2011.

SELTZER, Michael L; DROPPO, Jasha. Multi-task learning in deep neural networks for improved phoneme recognition. In: IEEE. **2013 IEEE International Conference on Acoustics, Speech and Signal Processing**. [S.l.], 2013. p. 6965–6969.

SERRANI, Vanessa Marquiafável. **Ambiente web de suporte à transcrição fonética automática de lemas em verbetes de dicionários do português do Brasil**. 2015. 202 f. Tese (doutorado) - Universidade Estadual Paulista Julio de Mesquita Filho, Instituto de Biociências, Letras e Ciências Exatas, 2015.

SHALEV-SHWARTZ, Shai; BEN-DAVID, Shai. **Understanding machine learning: From theory to algorithms**. [S.l.]: Cambridge university press, 2014.

SHARIFZADEH, Sara; SERRANO, Javier; CARRABINA, Jordi. Spectro-temporal analysis of speech for spanish phoneme recognition. In: IEEE. **2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP)**. [S.l.], 2012. p. 548–551.

SIVARAM, Garimella SVS; HERMANSKY, Hynek. Multilayer perceptron with sparse hidden outputs for phoneme recognition. In: IEEE. **2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2011. p. 5336–5339.

STEVENS, Stanley Smith; VOLKMANN, John; NEWMAN, Edwin B. A scale for the measurement of the psychological magnitude pitch. **The Journal of the Acoustical Society of America**, ASA, v. 8, n. 3, p. 185–190, 1937.

TANG, Hao; MENG, Chao-Hong; LEE, Lin-Shan. An initial attempt for phoneme recognition using structured support vector machine (svm). In: IEEE. **2010 IEEE International Conference on Acoustics, Speech and Signal Processing**. [S.l.], 2010. p. 4926–4929.

WEBER, Philip *et al.* Progress on phoneme recognition with a continuous-state hmm. In: IEEE. **2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2016. p. 5850–5854.

WELLS, John C *et al.* Sampa computer readable phonetic alphabet. **Handbook of standards and resources for spoken language systems**, Berlin and New York: Mouton de Gruyter. Part IV, section B, v. 4, 1997.

XIE, Yue *et al.* Phoneme recognition based on deep belief network. In: IEEE. **2016 International Conference on Information System and Artificial Intelligence (ISAI)**. [S.l.], 2016. p. 352–355.

YOUSAFZAI, Jibran *et al.* Combined features and kernel design for noise robust phoneme classification using support vector machines. **IEEE Transactions on Audio, Speech, and Language Processing**, IEEE, v. 19, n. 5, p. 1396–1407, 2010.

ZAKI, Mohammadi; SAILOR, Hardik B; PATIL, Hemant A. Analysis of hierarchical bottleneck framework for improved phoneme recognition. In: IEEE. **2016 International Conference on Signal Processing and Communications (SPCOM)**. [S.l.], 2016. p. 1–5.

ZHANG, Bo *et al.* Application of pronunciation knowledge on phoneme recognition by lstm neural network. In: IEEE. **2016 23rd International Conference on Pattern Recognition (ICPR)**. [S.l.], 2016. p. 2906–2911.

ZHENG, Xin *et al.* Contrastive auto-encoder for phoneme recognition. In: IEEE. **2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2014. p. 2529–2533.

_____. Learning dynamic features with neural networks for phoneme recognition. In: IEEE. **2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. [S.l.], 2014. p. 2524–2528.

APÊNDICE A - TABELA DE PONTUAÇÃO INORDINATIO

Ranking	Referências dos trabalhos	Ano Publicação	Indice H5 (Google metrics)	Citações	InOrdinatio
1	(SELTZER; DROPPO, 2013)	2013	79	179	199,079
2	(DELCROIX *et al.*, 2015)	2015	79	37	67,079
3	(KIM; HWANG; SUNG, 2014)	2014	79	42	67,079
4	(CARDONA; NEDJAH; MOURELLE, 2017)	2017	71	8	48,071
5	(LOHRENZ; LI; FINGSCHEIDT, 2018)	2018	7	1	46,007
6	(PRADEEP; RAO, 2017)	2017	0	1	41
7	(WEBER *et al.*, 2016)	2016	79	3	38,079
8	(ZHANG *et al.*, 2016)	2016	28	3	38,028
9	(KANNADAGULI; BHAT, 2015)	2015	8	8	38,008
10	(MIRHASSANI; TING; GHARAHBAGH, 2016)	2016	38	2	37,038
11	(KHWAJA *et al.*, 2016)	2016	14	2	37,014
12	(ZHENG *et al.*, 2014a)	2014	79	11	36,079
13	(PRADEEP; RAO, 2016)	2016	14	1	36,014
14	(XIE *et al.*, 2016)	2016	3	1	36,003
15	(ZAKI; SAILOR; PATIL, 2016)	2016	8	0	35,008
16	(DANESHVAR; VEISI, 2016)	2016	0	0	35
17	(KANNADAGULI; THALENGALA, 2015)	2015	9	3	33,009
18	(AKHILA; KUMARASWAMY, 2015)	2015	0	0	30
19	(PALAZ; MAGIMAI-DOSS; COLLOBERT, 2014)	2014	28	3	28,028
20	(ZHENG *et al.*, 2014b)	2014	79	2	27,079
21	(DENNIS *et al.*, 2014)	2014	79	1	26,079

Ranking	Referências dos trabalhos	Ano Publicação	Indice H5 (Google metrics)	Citações	InOrdinatio
22	(SIVARAM; HERMANSKY, 2011)	2011	79	16	26,079
23	(YOUSAFZAI *et al.*, 2010)	2011	42	16	26,042
24	(Patil; Pardeshi, 2014)	2014	10	1	26,01
25	(ALMISREB; ABIDIN; TAHIR,)	2014	10	0	25,01
26	(CUTAJAR *et al.*, 2011)	2011	12	11	21,012
27	(GOLIPOUR; O'SHAUGHNESSY, 2012)	2012	33	5	20,033
28	(FRIKHA *et al.*, 2013)	2013	14	0	20,014
29	(SHARIFZADEH; SERRANO; CARRABINA, 2012)	2012	11	5	20,011
30	(AROUS; AYED; ELLOUZE, 2012)	2012	0	3	18
31	(KSHIRSAGAR *et al.*, 2012)	2012	11	2	17,011
32	(TANG; MENG; LEE, 2010)	2010	79	10	15,079
33	(GANAPATHY; THOMAS; HERMANSKY, 2010)	2010	79	9	14,079
34	(KAZEMI; SOBHANMANESH, 2011)	2011	25	4	14,025
35	(KOTWAL *et al.*, 2010a)	2010	10	5	10,01
36	(ATSONIOS, 2011)	2011	9	0	10,009
37	(JLASSI; AROUS; ELLOUZE, 2010)	2010	314	4	9,314
38	(DABBAGHCHIAN *et al.*, 2010)	2010	8	3	8,008
39	(AGER; CVETKOVIĆ; SOLLICH, 2010)	2010	27	1	6,027
40	(KOSIC, 2010)	2010	0	1	6
41	(KOTWAL *et al.*, 2010b)	2010	11	0	5,011