Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

DEEP: Uma arquitetura para reconhecer emoção com base no espectro sonoro da voz de falantes da língua portuguesa

Gabriel A. Campos
Lucas da S. Moutinho

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Geraldo P. Rocha Filho

Brasília
2020

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

DEEP: Uma arquitetura para reconhecer emoção com base no espectro sonoro da voz de falantes da língua portuguesa

Gabriel A. Campos
Lucas da S. Moutinho

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Geraldo P. Rocha Filho (Orientador)
CIC/UnB

Prof. Dra. Roberta Barbosa Oliveira
CIC/UnB

Prof. Dr. Thiago Paulo Faleiros
CIC/UnB

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 24 de novembro de 2020

Dedicatória

Dedicamos este trabalho a todos nossos amigos e parentes que nos apoiaram nesta fase final de conclusão do curso. Também dedicamos este trabalho para a comunidade *open source* brasileira.

Agradecimentos

O presente trabalho foi realizado com o apoio do Departamento de Ciência da Computação (CIC) da Universidade de Brasília (UNB) por meio da disciplina de trabalho de graduação ofertada pela grade curricular do curso.

Agradecemos ao apoio do Prof. Dr. Geraldo P. Rocha Filho que foi nosso orientador para a realização da pesquisa. Agradecemos também aos nossos demais apoiadores, como amigos e parentes que auxiliaram, em diversos aspectos, para a elaboração deste trabalho e para conclusão do curso.

Agradecemos explicitamente a revisora e amiga, Luísa de Lumiére, pela imensa ajuda na revisão do trabalho.

Agradecimentos finais para os membros da empresa júnior de computação (CJR) da universidade de brasília por fornecerem, além do espaço físico, apoio moral na realização deste trabalho.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

O reconhecimento de emoção em fala é uma linha de pesquisa dentro da Inteligência Artificial (IA) que exige arquiteturas robustas de modelos de *Deep Learning* (DL) para a correta distinção das emoções percebidas na voz. Para responder a essa exigência, trabalhos recentes da literatura sugerem arquiteturas cada vez mais robustas, como a de modelos híbridos. No entanto, a utilização de múltiplas redes neurais de maneira sequencial pode ocasionar a propagação de erros entre os modelos. Além desse problema, ressalta-se que não foram encontrados outros trabalhos que treinam modelos em língua portuguesa. Dessa forma, a fim de lidar com as referidas limitações da literatura relacionada, neste trabalho é desenvolvida uma arquitetura para o reconhecimento de emoções com base em padrões presentes no espectro sonoro gerado pela voz de falantes da língua portuguesa: DEEP - ***DE****tection of voice* ***E****motion in* ***P****ortuguese language* (Detecção de Emoção na Voz na Linguagem Portuguesa). O DEEP é composto por um conjunto de modelos especialistas de redes neurais convolucionais, do inglês *Convolutional Neural Networks* (CNNs), modelos de DL treinados em língua portuguesa, cujo intuito visa à especialização da detecção de emoções. Para treinamento do modelo, foi utilizada a base de dados de voz em língua portuguesa VERBO, o que permite que esta tecnologia seja aplicada em diversas áreas nos países que têm esse idioma como oficial. Para avaliar os resultados da performance alcançada com a arquitetura proposta, em um primeiro momento, os modelos especialistas foram hiper parametrizados, o que permitiu o descobrimento de configurações otimizadas na detecção de cada emoção. Em seguida, as acurácias obtidas foram comparadas com as alcançadas por um modelo CNN classificador tradicionalmente apresentado na literatura relacionada, denominado neste trabalho por modelo *baseline*, em que foram observados ganhos de performance para todas as 7 emoções presentes no VERBO, com uma diferença média de 12.39%, tendo o maior ganho com a emoção Medo, esta que foi 24.42% maior quando comparado com a CNN.

Palavras-chave: Reconhecimento de emoção na voz, modelos especialistas, DEEP, VERBO, CNN, *deep learning*, língua portuguesa

Abstract

Speech emotion recognition is a line of research within Artificial Intelligence (AI) that requires complex architectures of Deep Learning (DL) models to distinguish the perceived emotions in voice. To fulfill this requirement, recent works suggest increasingly complex architectures, such as hybrid models. However, these models can propitiate error propagation among the sequentially placed models, increasing false positives. In addition to this problem, it is noteworthy that no other studies that train models in Portuguese were found. Thus, in order to deal with the referred limitations of the related literature, this work presents an architecture for emotion recognition based on patterns present in the sound spectrum generated by the voice of Portuguese speakers: DEEP - ***DE****tection of voice* ***E****motion in* ***P****ortuguese language*. DEEP is composed of specialist models of convolutional neural networks (CNN) whose aim is to specialize in detecting emotions. The Portuguese voice database VERBO was used for training the model, which allows this technology to be applied in several areas in countries that have this language as an official language. To evaluate the proposed architecture, the specialist models were hyper parameterized, which allowed the discovery of optimized configurations to detect each emotion. Then, the DEEP was compared with a CNN model, in which performance gains were observed for all seven emotions present in VERBO, with an average difference of 12.39%, having the highest gain with the Fear emotion, which was 24.42% higher when compared to CNN.

Keywords: Speech emotion recognition, specialist models, DEEP, VERBO, CNN, deep learning, portuguese language

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O reconhecimento de emoção em fala é uma linha de pesquisa dentro da Inteligência Artificial (IA), que consiste nas tarefas de reconhecimento e classificação da reação afetiva de um indivíduo [1]. O estudo das emoções e a maneira pelas quais as entendemos e as representamos no contexto computacional formam a área de conhecimento denominada Computação Afetiva [2]. Historicamente, o campo de estudo de IA foi construído com base em conceitos de como a aprendizagem humana funciona [3]. Analogamente, a exemplo de estudos como de Picard (2000) [2], existem evidências que apontam que o processo de aprendizagem humana é diretamente ligado à emoção. Consequentemente, tarefas que estão contidas no conjunto de áreas estudadas pela computação afetiva, tais como a análise de sentimentos e reconhecimento de emoção, são importantes para o desenvolvimento da área da IA como um todo. Os sistemas de reconhecimento de emoção em fala demonstram importante utilidade em aplicações que se baseiam primordialmente na interação humano-computador. Essas aplicações têm como intuito fundamental a compreensão das respostas emocionais do usuário a partir de uma determinada interação. Neste sentido, vale destacar a aplicação de sistemas de reconhecimento de voz em *call centers*, nos quais se torna possível avaliar qualitativamente o sucesso de uma chamada ao medir a expressão de determinadas emoções, tais como raiva, alegria e tristeza; em *smart assistants*, como a *Amazon Alexa* [4], em que o assistente de voz pode ter comportamentos diferentes, dependendo da emoção do usuário; e em aplicativos de música, os quais podem adequar as recomendações musicais ao usuário com base na emoção captada [5].

A aplicação de modelos estatísticos e de algoritmos de aprendizagem de máquina, do inglês *machine learning* (ML), vem sendo um dos possíveis caminhos para realizar tarefas de reconhecimento de emoção na fala desde o final do século XX [6] [7], com o intuito de permitir a realização desta tarefa de forma automática em aplicações de IA. Entretanto, conseguir identificar o estado emocional de um sujeito não é uma tarefa trivial, visto que demanda uma apurada capacidade de percepção. Em seu contexto original, os interlo-

cutores utilizam várias informações visuais, auditivas, semânticas e metalinguísticas [8] para determinar qual emoção a fala de uma pessoa invoca, o que torna a tarefa bastante complexa e propensa a erros para o contexto da IA. Diversos estudos foram realizados para o entendimento de quais fatores são relevantes para o reconhecimento de emoção [8] [9]. Dentre essas pesquisas, cabe citar o trabalho de Scherer (1995) [8], que apresenta evidências de que emoções são expressas diferentemente pela fala humana e que ouvintes são capazes de corretamente inferir o estado emocional de um interlocutor apenas com a informação da voz. Ainda assim, existem problemáticas relacionadas à tarefa de reconhecimento de emoção em voz dentro da IA. Uma das principais consiste na inexistência de um consenso sobre a definição teórica das emoções [10], de modo que as características mais importantes para a distinção entre emoções ainda não são claras. Além disso, aspectos como a variabilidade das frases faladas por pessoas diferentes adiciona outro nível de dificuldade na distinção de emoções, pois características específicas da voz de um indivíduo, como sotaque e ritmo da fala, alteram os resultados das *features* comumente extraídas da voz [9], tais como as prosódicas e MFCC (*Mel-Frequency Cepstral Coefficients*), utilizadas em trabalhos da literatura relacionada [6] [11] [12]. Adicionalmente, existem trabalhos [13] que apontam que a linguagem falada é um dos aspectos que influenciam diretamente no processo de reconhecimento de emoção.

O aumento da capacidade de processamento dos computadores e o avanço das tecnologias de *cloud computing* mitigaram o custo computacional de técnicas mais complexas de ML, como as de *Deep Learning* (DL) [14], esta que engloba técnicas de aprendizagem de máquina que exploram uma grande quantidade de dados ao identificar padrões de maneira profunda [15]. Esse fator favorece a utilização de técnicas de *Deep Leaning* em problemas cujas características relevantes não são facilmente reconhecíveis, como visão computacional, processamento de linguagem natural e reconhecimento automático de emoção na fala.

A literatura relacionada apresenta diversas propostas de modelos para a realização da tarefa de reconhecimento de emoção na fala. Trabalhos como Dellaert et al. (1996) [6], Kwon et al. (2003) [7] e Pan, Shen (2012) [11], utilizam modelos mais tradicionais de IA, tais como *K-nearest Neighbors* (KNN), *Support Vector Machines* (SVM) e *Hidden Markov Models* (HMM). Em Dellaert et al. (1996) [6], são utilizadas apenas *features* prosódicas para o modelo, com o intuito de reconhecer emoções somente pelas características fonéticas e linguísticas do som. Já em Kwon et al. (2003) [7] e Pan, Shen (2012) [11], são incorporadas as *features* MFCC aos modelos, que, a partir deste momento, tornou-se a *feature* predominante em trabalhos da área. Contudo, Dellaert et al. (1996) [6], Kwon et al. (2003) [7] e Pan, Shen (2012) [11] utilizam, ainda, modelos de ML mais simplórios, como o HMM, o KNN e o SVM. Tais modelos dificultam a distinção de um maior número de

emoções semelhantes as quais só poderiam ser melhor diferenciadas com a aprendizagem profunda de técnicas de DL, como as redes neurais convolucionais, do inglês *Convolutional Neural Networks* (CNN) [12] [16].

O próximo avanço das pesquisas analisadas é apresentado nos trabalhos de Han, Yu, Tanshev (2014) [12] e Abdel-Hamid et al. (2014) [16], no qual é introduzida a utilização de modelos de DL. A partir deste ponto, os trabalhos divergem entre si quanto à exploração de modelos com arquiteturas cada vez mais robustas, como no caso de [17] e [18], que utilizam modelos híbridos [1] para a classificação de emoções na voz. Porém, tais arquiteturas podem propagar erros nos modelos sequenciais aplicados, aumentando o número de falsos positivos. Além destas pesquisas, cita-se [19] e [20], que exploram novos grupos de *features* para a tarefa, tais como a combinação das *features* de voz com as de imagem e texto. Contudo, estes trabalhos aumentam a complexidade ao exigirem modelos adicionais para a conversão dos dados de voz em texto ou imagem. Adicionalmente, ressalta-se que nenhum dos trabalhos relacionados treina modelos com base em dados de voz na língua portuguesa do Brasil, e, tendo em vista as evidências [13] dos impactos da linguagem no processo de expressão de emoção em voz, em função, por exemplo, da presença de aspectos congruentes aos regionalismos e ritmos de fala, evidencia-se a importância de trabalhos que explorem reconhecimento de emoção em voz com enfoque na língua portuguesa.

1.1 Objetivos

Este trabalho tem como propósito desenvolver e validar o DEEP - ***DE****tection of voice* ***E****motion in* ***P****ortuguese language* (Detecção de Emoção na Voz na Linguagem Portuguesa), uma arquitetura para o reconhecimento de emoções em voz com base em padrões presentes no espectro sonoro gerado pela voz de falantes da língua portuguesa. O DEEP será composto por um conjunto de modelos CNN especialistas, modelos de DL treinados em língua portuguesa, com a utilização da base de dados de emoção na voz em língua portuguesa VERBO [21], cujo intuito visa à especialização da detecção de emoções. Para validar o DEEP, tornam-se necessárias duas etapas, aqui colocadas como objetivos específicos: (i) otimizar os hiper parâmetros dos modelos especialistas, com o objetivo de se encontrar a configuração de maior performance para cada um destes; e (ii) comparar os resultados obtidos com os de um único modelo CNN, a fim de se comparar os ganhos de performance da arquitetura proposta em relação a um modelo tradicional. Com a validação do DEEP, este se apresentará como uma alternativa a outras arquiteturas propostas em trabalhos atuais para a tarefa de reconhecimento de emoção em voz.

[1] Por modelos híbridos, entende-se como arquiteturas que contêm mais de um modelo, colocados sequencialmente, para a classificação, por exemplo, dos CNN-LSTM [17] e [18], em que uma CNN, utilizada para extração de *features*, é ligada diretamente a uma LSTM para classificação

1.2 Organização do Trabalho

O restante deste trabalho está organizado em cinco capítulos, apresentados a seguir:

- **Fundamentação Teórica**: Neste capítulo, são apresentados a base teórica e os principais conceitos importantes para o entendimento do trabalho.
- **Trabalhos Relacionados**: Neste capítulo, são descritos os trabalhos analisados e utilizados como inspiração para a criação deste, citando os resultados importantes e possíveis limitações.
- **Uma arquitetura para o reconhecimento de emoção na voz em língua portuguesa**: Neste capítulo, será descrito o processo de desenvolvimento do DEEP, detalhando os processos de organização dos dados, fluxos de desenvolvimento dos modelos e apresentando sua arquitetura com detalhes.
- **Resultado**: Neste capítulo, são apresentados os resultados obtidos sob determinadas perspectivas diferentes e é iniciada uma discussão sobre os resultados alcançados e se os objetivos determinados no capítulo de introdução foram cumpridos.
- **Conclusão**: Neste capítulo, apresentamos algumas considerações finais sobre o trabalho desenvolvido, resultados, objetivos e possíveis aspectos a serem explorados em trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Neste capítulo, será apresentada a base teórica utilizada para o desenvolvimento do projeto e de conceitos sobre voz e emoção, mais especificamente dos aspectos referentes às *features* utilizadas.

2.1 Voz e Emoção

A voz é um dos instrumentos de comunicação mais utilizados pelo ser humano. Ela pode ser usada para transmitir informações, comunicar perigo, ou, o ponto de interesse deste trabalho, expressar emoções.

A voz é composta por ondas acústicas que se propagam em diferentes frequências, transmitidas pelas cordas vocais de uma pessoa. Para os termos deste trabalho, também é importante fazer a distinção entre o som emitido pelo falante e as suas representações. Neste trabalho, é necessário lidar com a representação digital dos sons, tendo em vista que os áudios da base de dados utilizada foram gravados e convertidos digitalmente. Em sua forma digital, o sinal de áudio é codificado como amostras numéricas em uma sequência contínua. Em especial, a base de dados de voz VERBO [21] utiliza o formato *.wav*, que não realiza compressão do som digital, sendo, dessa forma, o formato digital mais próximo da expressão natural do som.

As emoções também têm papel importante na comunicação humana. No contexto natural, utilizamos várias informações do ambiente para conseguirmos detectar emoções em suas expressões. Assim, é seguro afirmar que a fala é um dos elementos relevantes ao se tentar observar quais emoções estão sendo expressadas por um interlocutor. Nesse contexto, existem trabalhos que afirmam que emoções são expressas diferentemente pela fala humana e que ouvintes são capazes de corretamente inferir o estado emocional de um interlocutor apenas com a informação da voz [8]. Adicionalmente, existem trabalhos que demonstram que é possível inferir a emoção expressada por uma representação digital

de uma fala, com diferentes técnicas de inteligência artificial [11] [12] [17]. Assim, são confirmadas duas suposições importantes para este trabalho: (i) É possível inferir o estado emocional de um interlocutor apenas com a informação da fala; e (ii) é possível solucionar a tarefa de reconhecimento de emoção na voz por técnicas de IA.

Uma das formas de se organizar o conjunto de emoções é observada no Modelo Circumplexo, introduzido por Russel em [22] e ilustrado na Figura 2.1. Esta Figura apresenta um conjunto de emoções dispostas em um círculo, com valores de prazer expressos no eixo horizontal e valores de excitação expressos no eixo vertical, sendo que o centro do círculo representa um estado de neutralidade quanto à excitação e prazer. Dessa forma, as emoções que se encontram próximas umas das outras, na disposição circular, apresentam maiores correlações, seja pela proximidade dos valores de excitação ou de prazer. Este modelo aponta que as emoções apresentam diversas semelhanças quanto às suas características e que uma completa distinção entre elas não é possível. Essa noção é essencial para o trabalho, pois evidencia um padrão importante em relação à expressão de emoções.

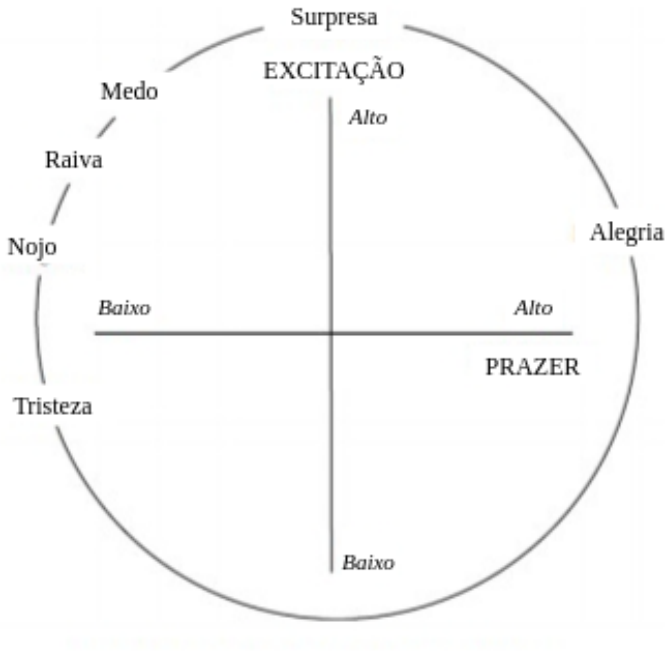


Figura 2.1: Modelo Circunflexo de Russel para emoções apresentado em [22], adaptado para língua portuguesa

2.2 Aprendizagem de Máquina

Aprendizagem de Máquina (ML) é uma subárea de estudos de IA, definida [23] como o campo de estudos de agentes inteligentes. Os agentes inteligentes são módulos que recebem

alguma informação do ambiente e tomam ações baseadas nas informações recebidas.

Atualmente, a ML é tratada como um conjunto de ferramentas e técnicas que podem ser utilizadas no reconhecimento de padrões para a solução de problemas. Para esses tipos de tarefa, ao utilizar algoritmos de ML, tentamos construir uma aproximação boa o suficiente dos padrões de entrada, mas que, ao serem expostos a novos exemplos, ainda retornem os resultados desejados. Logo, o funcionamento desses algoritmos não deve simplesmente construir uma função que mapeie os pares de entrada-saída conhecidos, mas que o faça de forma a manter um certo grau de generalização.

Em síntese, a aplicação de ML tem como objetivo construir algoritmos que otimizem uma métrica de performance definida, pelo uso de dados de exemplos ou experiências prévias [24]. No seu núcleo, técnicas de ML utilizam métodos estatísticos e modelos matemáticos para construir inferências ou generalizações baseadas nos exemplos de treinamento.

ML pode ser utilizada para realizar uma vasta gama de tarefas. Para este trabalho, utilizamos a ML para realizar a tarefa de **classificação**. Classificação é a tarefa de predição dos respectivos rótulos das instâncias de entrada, sendo tais rótulos denominados de dados de saída. No contexto deste trabalho, visamos a realizar a classificação das entradas (áudios do conjunto de dados) com suas respectivas saídas (rótulos de emoção).

Os modelos de ML, denominados como classificadores, são modelos que, dado um conjunto de rótulos de tamanho n $\{y_1, y_2, y_3, \ldots, y_n\}$ e um conjunto de dados de entrada de tamanho m $\{X_1, X_2, X_3, \ldots, X_m\}$, têm como finalidade reproduzir um algoritmo que seja capaz de conectar uma entrada qualquer X_k com seu respectivo rótulo. No contexto deste trabalho, o conjunto de rótulos $\{y_1, y_2, y_3, \ldots, y_n\}$ seria as emoções abrangidas pelo conjunto de dados VERBO [21], sendo elas: (i) alegria, (ii) nojo, (iii) medo, (iv) neutro, (v) raiva, (vi) surpresa e (vii) tristeza. O conjunto de entrada $X = \{X_1, X_2, X_3, \ldots, X_m\}$ requer mais explicações. Como citado anteriormente, a ML utiliza-se de métodos estatísticos e modelos matemáticos para reproduzir um algoritmo que mapeia o conjunto de entrada com o conjunto de saída. Desta forma, é necessário que os dados de uma instância $X_i | X_i \in X$ sejam valores numéricos. Consequentemente, todos os modelos de ML necessitam de uma representação numérica dos dados de entrada. Além disso, no caso de um modelo classificador, é necessário que essa representação consiga descrever as características que distinguam as instâncias de uma classe das instâncias de outra. Logo, para os dados de entrada, é necessário realizar a escolha de quais informações dentro dos dados é importante para a resolução do problema para compor o conjunto de dados de entrada. Os valores que compõem uma instância X_i de X são chamados de *features*.

As ***features*** são as características que compõem uma instância do conjunto de entrada $\{X_1, X_2, X_3, \ldots, X_m\}$, julgadas relevantes para a distinção dentre classes possíveis

do conjunto de saída do problema, por exemplo, para um modelo que tenha como objetivo fazer a previsão do tempo do dia atual. Suponha que as informações necessárias para determinar se vai chover ou não sejam a temperatura atual, a pressão atmosférica e a umidade do ar. Logo, o conjunto de informações de temperatura atual, pressão atmosférica e umidade do ar são as *features* do conjunto. Para outras tarefas, não é tão simples determinar quais informações são relevantes para entrar no conjunto de dados. As *features* utilizadas para o desenvolvimento deste trabalho serão exploradas em seções posteriores.

2.2.1 Aprendizagem Supervisionada

Existem múltiplas abordagens apresentadas na literatura para construir um modelo classificador. Para o desenvolvimento deste trabalho, foi utilizada uma abordagem denominada de Aprendizagem Supervisionada.

A Aprendizagem Supervisionada é um dos tipos de aprendizagem de IA. Para essa abordagem de aprendizagem, é necessário possuir conhecimento prévio de exemplos de entrada-saída. Por exemplo, a base de dados utilizada para o desenvolvimento deste trabalho possui exemplos de arquivos de áudios e suas respectivas emoções representadas, que representam respectivamente, as possíveis entradas e saídas desta abordagem. O objetivo deste método de aprendizagem é conseguir mapear os dados de entrada com saídas desejadas, com base em um conjunto de exemplos, denominado de conjunto de treinamento, ou seja, de entrada-saída já conhecidos [3]. Os algoritmos de Aprendizagem Supervisionada analisam o conjunto de treinamento para inferir uma função que, em seguida, é utilizada para mapear um outro conjunto, denominado de conjunto de validação, para, consequentemente, avaliar a função inferida.

2.2.2 Métricas de avaliação

Um detalhe importante para a construção deste trabalho é detalhar algumas métricas da avaliação para a performance de algoritmos de ML. Estas métricas são importantes pois permitirão a comparação da arquitetura do DEEP com outros modelos para a classificação de emoção na voz. Na lista abaixo, temos uma lista de variáveis que serão utilizadas para calcular as métricas de avaliação:

- **Verdadeiros Positivos (VP)**: Classificação correta das classes positivas
- **Verdadeiros Negativos (VN)**: Classificação correta das classes negativas
- **Falsos Positivos (FP)**: Erro em que o modelo previu uma classe positiva, quando o valor real pertencia a classe negativa

- **Falsos Negativos (FN)**: Erro em que o modelo previu uma classe negativa, quando o valor real pertencia a classe positiva

Tendo-se a lista de variáveis, podemos agora detalhar as fórmulas para o cálculo das principais métricas de avaliação.

- **Acurácia**: Indica a performance geral do modelo

$$acurácia = \frac{VP + VN}{VP + FP + VN + FN} \tag{2.1}$$

- **Precisão**: Dentre as classificações positivas que o modelo fez, quais foram corretas

$$precisão = \frac{\text{VP}}{\text{VP + FP}} \tag{2.2}$$

- **Sensibilidade**: Dentre todas as classificações positivas esperadas, quantas foram corretas

$$sensibilidade = \frac{\text{VP}}{\text{VP + FN}} \tag{2.3}$$

- **F1-Score**: Média harmônica entre precisão e sensibilidade

$$F1 - Score = \frac{\text{2*precisão*sensibilidade}}{\text{precisão + sensibilidade}} \tag{2.4}$$

2.3 Redes Neurais

O termo Rede Neural deriva historicamente das tentativas de modelar e representar o processo de aquisição de conhecimento por computador, baseando-se no processo biológico de aprendizagem [25].

A base das redes neurais é composta pelos chamados *perceptrons*. As redes neurais são uma forma de ML, logo, também possuem a mesma finalidade: mapear um conjunto de entrada $\{X_1, X_2, X_3, \ldots, X_m\}$ a um conjunto de saída $\{y_1, y_2, y_3, ..., y_n\}$ através de uma função. Mais especificamente, redes neurais como as *Multi-Layer Perceptrons*, estas pertencem ao ramo de DL e realizam o reconhecimento de padrões à níveis bem mais profundos. Os *Perceptrons* realizam esse processo nas redes neurais, como demonstra a Figura 2.2.

A rede neural é representada como um conjunto de nós e arestas. Cada nó possui um valor, que é denominado **valor de ativação**, e arestas que conectam cada nó com todos os outros nós da camada seguinte, chamadas de **pesos**. Os **valores de ativação**

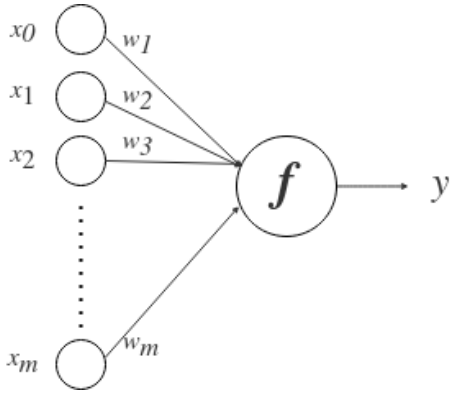


Figura 2.2: Arquitetura básica de um *Perceptron*

estão contidos entre os valores 0 e 1 e são análogos à representação de valores de *bits*: 0 representa um nó *"totalmente desativado"* e 1 representa um nó *"totalmente ativado"*.

Contudo, as redes neurais podem possuir uma grande quantidade de *perceptrons*. Para as chamadas *Multi-Layer Networks*, elas agrupam os *perceptrons* em organizações denominadas de camadas, de modo que utilizam múltiplas camadas intermediárias entre a entrada e a saída. Essa forma, apresentada na Figura 2.4, é a apresentação tradicional de uma rede neural, na qual existe uma série de nós interconectados por arestas (pesos) em múltiplas camadas subsequentes.

Será utilizada a arquitetura ilustrada na Figura 2.2 para explicar como funciona o processo de treinamento de uma rede neural. Assim, considere que esse *perceptron* está sendo utilizado para determinar se um áudio expressa ou não a emoção *alegria*. A entrada $X = [x_0, x_1, x_2, \ldots, x_m]$ sendo passada para este *perceptron* corresponde às m *features* de uma instância do conjunto de dados, e o valor de saída $y \in \{0, 1\}$ representa se a instância X pertence ou não à classe *alegria*, sendo 1 o valor positivo e 0 o valor negativo. Na camada de entrada, não existe transformação ou cálculo dos dados, sua função é transmitir as m *features* para o nó seguinte. Os pesos, representados na imagem como o conjunto $\{w_0, w_1, \ldots, w_m\}$, são os responsáveis por transmitir as *features* relevantes para as camadas seguintes. Se, por exemplo, é sabido que a *feature* x_1 não é relevante para a classificação, o peso da aresta w_1 provavelmente teria um valor próximo de 0. Isso acontece, pois, para transmitir os valores de *features* para os nós subsequentes, são multiplicados o valor da *feature* com seus pesos respectivos. No caso apresentado, o cálculo de transmissão para um nó se dá pela Equação 2.5, ou a forma genérica de transmissão dos valores de ativações para k nós da camada seguinte pela Equação 2.6.

$$x_0 w_0 + x_1 w_1 + \cdots + x_m w_m \tag{2.5}$$

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \dots \\ x_m \end{bmatrix} \begin{bmatrix} w_{00} w_{01} \dots w_{0m} \\ w_{10} w_{11} \dots w_{1m} \\ w_{20} w_{21} \dots w_{2m} \\ \dots \\ w_{k0} w_{k1} \dots w_{km} \end{bmatrix} \tag{2.6}$$

Entretanto, como citado anteriormente, os valores de ativação desse exemplo são contidos no intervalo $\{0, 1\}$. Esse é um problema comum em algoritmos de ML, pois, em modelos classificadores, os valores de saídas são limitados pela quantidade de classes possíveis do problema. Para solucionar tais problemas, existe uma gama de funções que podem ser aplicadas nos valores de ativação para limitá-los dentre os valores desejados de saída. Para o exemplo detalhado anteriormente, em que os valores de saída estão contidos em $\{0, 1\}$, é possível aplicar funções de ativações, como a função *softmax* para delimitar os valores de saída.

Após o processo de transmissão das *features* entre as k camadas de uma Rede Neural, é obtido no final um valor predito y. Isso significa que esse classificador fez a predição da instância do conjunto de dados X_i, representado pelas *features* $\{x_0, x_1, \dots, x_m\}$, pertencente à classe y_i. No processo de treinamento supervisionado de modelos de Redes Neurais, temos um conjunto previamente conhecido de pares entrada-saída, então, no final do cálculo de transmissão, também é conhecido o valor esperado de y_i. Assim, após cada instância, é calculado o erro de treinamento, subtraindo o valor esperado de y_i do valor obtido. Ao final do treinamento de todas as instâncias do conjunto de treinamento, é calculada uma média de todos os erros. O objetivo final do treinamento das redes neurais é minimizar ao máximo o valor da métrica de erro, ajustando os valores dos pesos como uma função do valor do erro da etapa anterior do treinamento.

Um dos motivos do porquê Redes Neurais se apresentam como uma interessante forma de resolução de problemas de IA é a capacidade de extração de informações a partir dessa arquitetura básica. Ao adicionar mais camadas a uma arquitetura de Redes Neurais, é esperado que os nós de camadas intermediárias, chamadas *hidden layers*, sejam capazes de identificar padrões nos valores das *features* que são relevantes à classificação e consigam minimizar o erro geral da arquitetura a partir da manipulação dos valores dos pesos da rede. Por conta dessa abordagem, o treinamento de Redes Neurais se beneficia da grande quantidade de dados, por isso se torna uma alternativa interessante para lidar com dados de alta dimensionalidade.

2.3.1 Redes Neurais Convolucionais

As Redes Neurais Convolucionais, ou do inglês *Convolutional Neural Networks* (CNNs), são uma abordagem dentro das *Multi-Layer Networks*. Nessa abordagem, as camadas de nós possuem uma representação tridimensional, normalmente representada como *altura* × *largura* × *profundidade*, na qual a altura e largura são valores escolhidos para arquitetura e a profundidade está relacionada com a quantidade de *features*, o que as torna próprias para tarefas como análise de imagens ou de séries temporais, como no caso deste trabalho. Entretanto, o que define as CNNs é a operação de convolução. A convolução é um filtro usado para mapear os valores de ativação de uma camada para a próxima. A operação de convolução utiliza um filtro de pesos tridimensional com a mesma profundidade da camada atual, mas com altura e largura menores. Os valores da camada seguinte serão o produto dentre este filtro de pesos aplicados ao bloco de *altura* × *largura* da camada seguinte. O processo de convolução está exemplificado na Figura 2.3. Na Figura 2.3, é ilustrada uma camada de entrada com altura 100, largura 100 e 128 de profundidade, e um filtro de pesos com altura 20 e largura 20. A convolução será aplicada em todo espaço na camada seguinte que possuir as dimensões do filtro de pesos.

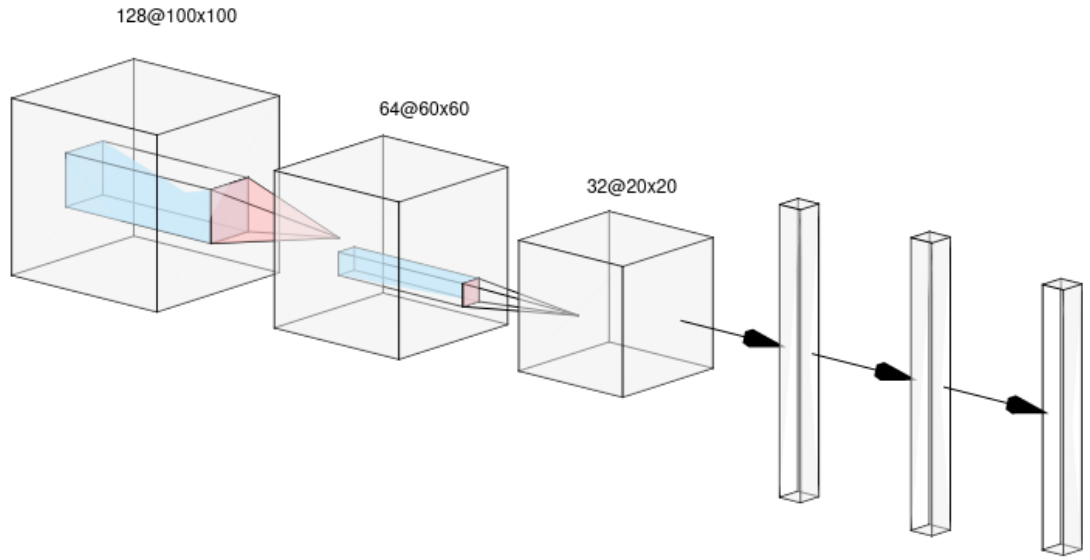


Figura 2.3: Ilustração do processo convolucional, seguido por camadas de classificação.

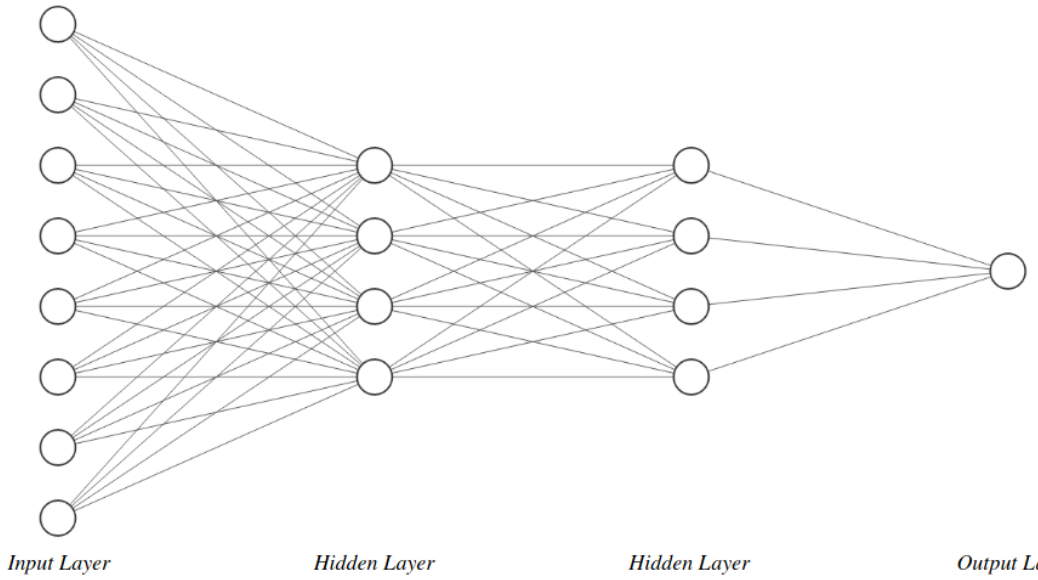


Figura 2.4: Exemplo de Rede Neural

2.4 Hiper Parametrização e *Hyperopt*

Os algoritmos de Aprendizagem de Máquina comumente apresentam uma gama de variáveis configuráveis que podem alterar significativamente os resultados de um modelo. Essas variáveis configuráveis são chamadas de **hiper parâmetros**.

Uma tarefa essencial ao se construir modelos de ML é encontrar os melhores hiper parâmetros para seu respectivo modelo. Uma problemática na utilização de modelos de Redes Neurais é que, o reconhecimento de quais hiper parâmetros são os mais importantes para o treinamento é complicado, devido à quantidade de possíveis hiper parâmetros e configurações de camadas. Outra problemática em relação à escolha dos hiper parâmetros é que esse é um processo que deve ser feito antes do treinamento dos modelos. Os algoritmos de ML, assim como quaisquer algoritmos computacionais, requerem algum custo computacional para operar, tanto de utilização de CPU quanto de tempo de processamento, sendo estes algoritmos notoriamente custosos. Logo, uma pessoa que deseja avaliar diversas configurações necessita esperar o tempo do processo de treinamento para testar configurações de hiper parâmetros diferentes.

Além disso, as configurações de hiper parâmetros são altamente dependentes do domínio do problema a ser resolvido, do algoritmo de ML escolhido, dos dados a serem utilizados, da abordagem do treinamento e de outras variáveis do ambiente do experimento. Logo, não existe uma solução simples para determinar quais são os valores *"ótimos"* para

qualquer tarefa de ML. Considerando a importância do processo de otimização de hiper parâmetros, a literatura relacionada apresenta várias abordagens e ferramentas para a realização desta tarefa.

O *Hyperopt* [26] é uma biblioteca de otimização de hiper parametrização para a linguagem *python*. O objetivo do processo de hiper parametrização é procurar a melhor configuração de hiper parâmetros para alguma determinada métrica de avaliação dentro do espaço das possíveis configurações que podem ser construídas com os valores de hiper parâmetros determinados. Os valores dos hiper parâmetros podem ser determinados como um valor dentro de um intervalo ou um valor dentre uma lista fixa de escolhas. O *hyperopt* oferece três escolhas de algoritmos para o processo de otimização dos hiper parâmetros, dentre eles o *Tree-structured Parzen Estimator*, que foi o algoritmo usado neste trabalho. O *Tree-structured Parzen Estimator* (TPE) [1] é uma abordagem de otimização sequencial de modelos que consiste em sequencialmente construir modelos e, com base no histórico das medidas de performance, consequentemente, escolher novos hiper parâmetros para serem testados no modelo.

[1] https://optunity.readthedocs.io/en/latest/user/solvers/TPE.html

Capítulo 3

Trabalhos Relacionados

Neste capítulo serão apresentados os trabalhos relacionados a esta pesquisa e que foram utilizados como referência para a construção deste. Em uma primeira instância, tais trabalhos serão apresentados em ordem cronológica de suas publicações, evidenciando seus objetivos, metodologias e limitações. Em seguida, é feita uma análise comparativa entre o DEEP e os trabalhos relacionados, ressaltando semelhanças e diferenças, além de destacar como este se propõe para lidar com as limitações da literatura atual.

A aplicação de modelos estatísticos e de algoritmos de aprendizagem de máquina para o reconhecimento da fala vem sendo discutida em trabalhos acadêmicos desde o final do século XX [27] [28]. A utilização destes para o reconhecimento de emoção na fala também tem trabalhos datados do mesmo período [6] [7], porém ganharam mais espaço apenas nos últimos anos, em especial de 2012 a 2020 [11] [12] [17] [18] [19] [20]. Apesar da popularização recente do tema, pouco foi publicado acerca do assunto em âmbito nacional, de forma que a pesquisa de Neto et al.(2018) [21] é tomada como a única referência brasileira, sendo esta utilizada para adquirir a base de dados de voz em português. Desse modo, o DEEP se destaca de todos os outros trabalhos citados por ser o único que aborda uma base de dados de voz em língua portuguesa.

No início da década de 1990, alguns trabalhos já consolidavam a possibilidade de reconhecimento da fala por máquinas. Juang e Rabiner (1991) [27] apresentam uma gama de aplicações da tecnologia em seu artigo expositivo, a exemplo dos discadores de voz para telefone, como consequência direta do grande avanço obtido por meio da utilização de modelos estatísticos em sistemas de reconhecimento de fala. Dentre os sistemas apresentados na pesquisa, os que utilizavam HMM, considerado estado da arte da época, já alcançavam um alto grau de acurácia em tarefas como o reconhecimento de palavras. Shannon, Zeng, Kamath, Wygonski e Ekelid (1995) [28] também estudam, em seu trabalho, metodologias para alcançar melhores performances na tarefa de reconhecimento de

palavras por voz, observando resultados quase perfeitos sob condições de grande redução da informações espectrais. Pouco tempo depois, são publicadas as primeiras pesquisas que exploram um novo viés, o reconhecimento de emoção na fala. Dellaert, Polzin e Waibel (1996) [6], em seu trabalho pioneiro, se propõem a criar um modelo capaz de reconhecer a emoção a partir de áudios de expressões gravadas. Para treinamento, utilizaram uma base de mil expressões gravadas entre 4 possíveis emoções (alegria, tristeza, raiva e medo), da qual realizaram a extração de *features* prosódicas dos áudios e as alimentaram a vários algoritmos de aprendizagem supervisionada, obtendo um melhor resultado, em termos de acurácia, com o KNN. Apesar de uma acurácia menor, em comparação com a tarefa inicial de reconhecimento de palavras de [27] e [28], o resultado obtido reafirmava a possibilidade de reconhecimento de emoção na voz com ML, porém, por se tratar de um estudo piloto, a falta de conhecimento de *features* e de outros algoritmos de ML mais apropriados, como os utilizados por trabalhos mais recentes e pelo DEEP, impactaram diretamente nos resultados.

Outras pesquisas, além de utilizarem as *features* prosódicas, começaram a explorar as *features* MFCC [7] [11]. Nesse cenário, Kwon, Chan, Hao e Lee (2003) [7] propõem uma nova metodologia de reconhecimento de emoção na voz que melhor diferenciasse um maior número de emoções de valor correlato, positivas ou negativas. Para isso, realizam a extração de *features* MFCC e prosódicas de dois *databases* de voz, o SUSAS [1] e o AIBO [2], alimentando-as a um classificador SVM. A utilização dessas *features* apresentou bons resultados para o *database* SUSAS, que possui 4 classes de emoção de usuários sob condições de stress (raiva, neutro, barulhento e *lombard*). Contudo, a pesquisa não obteve resultados satisfatórios com o AIBO, que possui 5 classes de emoção (raiva, tédio, alegria, tristeza e neutro), pois, apesar de terem encontrado *features* mais apropriadas para a tarefa, o modelo para classificação por regressão utilizado, o SVM, não era robusto o suficiente para distinguir o maior número de emoções correlatas, neste caso negativas: tédio, tristeza e raiva, o que diminui a acurácia final. Outra pesquisa bem similar a esta foi a de Pan, Shei e Shei (2012) [11] que também propôs a exploração de *features* MFCC e prosódicas na tarefa de reconhecimento de emoção na voz. Nesta, foram utilizadas 2 bases de voz, uma alemã e outra chinesa, e foram obtidos ótimos resultados com um classificador SVM, ainda que ambas as bases utilizadas possuam apenas 3 classes de emoção (alegria, tristeza e neutro). Assim, observa-se em ambos [7] [11] que a utilização de *features* prosódicas e de MFCC com algoritmos de aprendizagem de máquina apresentou melhores acurácias para até 4 classes de emoção, porém, para o reconhecimento de um maior número de emoções e, consequentemente, de emoções mais correlatas, o SVM utilizado por ambas

[1] https://catalog.ldc.upenn.edu/LDC99S78

[2] http://www-gth.die.upm.es/research/documentation/AI-76Emo-02.pdf

ainda não abarca a complexidade exigida para maiores distinções, o que seria possível com a aplicação de modelos de redes neurais, como as presentes no DEEP.

Dada a necessidade de exploração de modelos mais robustos, na bibliografia mais recente, alguns trabalhos optam por estudar o impacto de algoritmos de DL no reconhecimento de emoção na voz [12] [16]. Na pesquisa de Han, Yu e Tashev (2014) [12], objetiva a procura de modelos mais apropriados para a tarefa de reconhecimento de emoção na fala, que possam superar os considerados estado-da-arte da época, entre estes os SVM. Nesta busca, foram obtidos resultados que sugerem que a utilização de redes neurais, como as DNNs (*Deep Neural Networks*), alcança acurácias 20% melhores em comparação aos demais modelos de aprendizagem de máquina, como as SVM e as HMM. Em concordância com o objeto de estudo da linha de pesquisa em questão, cabe citar também o trabalho de Abdel-Hamid, Mohamed, Jiang, Deng, Penn e Yu (2014) [16], que tem como objetivo a análise comparativa entre CNNs (Convolutional Neural Networks) e DNNs como classificadores para sistemas de reconhecimento de voz, observando uma redução na taxa de erro de até 10% com a utilização das CNNs. Ambas as pesquisas [12] [16] sugerem ganhos de performance com a utilização de redes neurais, seja em sistemas de reconhecimento de voz ou de emoção na voz, entretanto, ainda que sejam mais apropriados para a tarefa de reconhecimento de emoção na voz, a aplicação de apenas um modelo de rede neural para distinguir as nuances de um maior número de emoções correlatas, como 7 ou 8 emoções, ainda é insuficiente. Assim, neste caso, são exigidas arquiteturas ainda mais robustas, como a de modelos híbridos ou a de modelos especialistas do DEEP.

O âmbito de reconhecimento de emoção na voz continua sendo bastante visado em pesquisas atuais. No último ano (2019), foram publicadas diversas pesquisas sugerindo a aplicação de modelos DL para a tarefa de classificação de emoção na voz com arquiteturas mais robustas [17] [18]. Neste contexto, cita-se, em primeira instância, a pesquisa de Huang, Wu, Hong, Su e Chen (2019) [17] cuja proposta refere-se a um modelo híbrido para o reconhecimento de emoção na voz, o qual visou superar os resultados estado-da-arte da base de voz chinesa NMINE [3]. Nesta base, os pesquisadores se deparam com o desafio de classificação de 7 emoções (alegria, desgosto, medo, neutro, raiva, surpresa, tristeza), sendo necessária a diferenciação de um maior número de emoções correlatas, como, por exemplo, tristeza, medo, raiva e desgosto, as quais estão posicionadas na polaridade negativa de acordo com o Modelo Circumplexo de Russel [22]. Assim, para superar o estado-da-arte de algoritmos tradicionais de DL nesta base, como um único modelo LSTM (*Long-Short Term Memory*) ou CNN, os pesquisadores sugerem a extração de ambos os aspectos verbais e não verbais da fala para classificação, o que seria possível

[3] https://www.researchgate.net/publication/322876066$_{N}NIME_{T}he_{N}THU-NTUA_{C}hinese_{i}nteractive_{m}ultimodal_{e}motion_{c}orpus$

utilizando uma arquitetura híbrida: uma CNN para extração das *features* e uma LSTM (*Long-Short Term Memory*) para classificação. O modelo híbrido superou o resultado dos modelos tradicionais de DL. Outra pesquisa neste viés é a de Zhao, Mao e Chen (2019) [18], essa que estuda, também, ganhos de performance na aplicação de um modelo de arquitetura híbrida para a classificação. Nesta pesquisa, duas bases são utilizadas, a *Berlin EmoDB* [4], uma base de dados alemã com 7 classificações de emoção (alegria, desgosto, medo, neutro, raiva, ansiedade, tédio), e o IEMOCAP [5], uma base de dados em inglês com 8 classificações de emoção (alegria, surpresa, desgosto, medo, neutro, raiva, frustração, tédio). O ganho de performance é observado ao se utilizar uma arquitetura híbrida 2D CNN-LSTM ao invés de somente uma DBN (*Deep Belief Network*) ou uma CNN, em ambas as bases. Percebe-se em ambas pesquisas [17] [18] que a aplicação de arquiteturas mais robustas, como as híbridas, pode acarretar em ganhos de performance quando comparados a um único modelo de DL. Todavia, deve-se colocar que a aplicação destes modelos sequencialmente pode propagar os erros do primeiro modelo para os próximos. O DEEP ataca diretamente este problema ao utilizar de uma arquitetura de modelos especialistas para cada emoção, de modo a estabelecer uma conexão não sequencial, e, sim, paralela, visando diminuir o número de falsos positivos dos resultados.

Pesquisas deste ano também estão procurando novas metologias que superem a performance de modelos únicos, dando foco não somente em arquiteturas mais robustas como em conjuntos de entrada diferentes dos tradicionais, que utilizam somente de *features* prosódicas e MFCC [19] [20]. Ho, Yang, Kin e Lee (2020) [19] em sua pesquisa se propõem a criar uma metodologia que alcance melhores resultados do que a alcançada por modelos únicos em 3 bases: a IEMOCAP, a MELD [6] e a CMU-MOSEI [7]. Para isso, os pesquisadores utilizam duas modalidades de *input*, texto e áudio. Para o áudio, realizam a extração de *features* MFCC utilizando a ferramenta *opensmile*, enquanto que as informações textuais são extraídas utilizando o modelo pré-treinado BERT (*Bidirectional Encoder Representations from Transformers*). Ambas as representações de áudio e texto são alimentadas, através de um mecanismo MLMHFA (*Multi-Level Multi-Head Fusion Attention*) em dois fluxos de RNNs (*Recurrent Neural Networks*), garantindo, assim, os ganhos de performance desejados. Outra pesquisa que se propõem a avaliar ganhos de performance com conjuntos de entrada distintos dos tradicionais é a de Kwon e Mustaqeem (2020) [20]. Nesta, os pesquisadores alimentam uma variante de CNN, uma DSCNN (Deep Stride Convolutional Neural Network), com *inputs* de imagem de espectogramas gerados a partir dos áudios das bases de dados IEMOCAP e RAVDESS

[4] http://emodb.bilderbar.info/start.html
[5] https://sail.usc.edu/iemocap/
[6] https://affective-meld.github.io/
[7] https://github.com/A2Zadeh/CMU-MultimodalSDK

[8], conquistando acurácias um pouco mais altas do que outros trabalhos estado-da-arte. Percebe-se, assim, com [19] e [20], que a busca de melhores performances também envolve a exploração de novas *features*, contudo, ambos os trabalhos restringem a utilização das metodologias sugeridas para *databases* que possuam informações além das de áudio, exigindo a aplicação de modelos adicionais, e igualmente complexos, para converter a voz em imagem ou texto, caso estas informações não existam. Esta maior complexidade dificulta que tais metodologias sejam replicadas para aplicações de áudio somente. Desse modo, o DEEP facilita a resolução da problemática ao, simplesmente, adicionar uma nova *feature* de voz, as *features* cromáticas, visando a otimização.

3.1 Discussão dos Trabalhos Relacionados

A partir da leitura dos trabalhos relacionados, observa-se que o tema de reconhecimento de emoção na fala, apesar de recente, teve grandes avanços na descoberta de *features* e algoritmos mais apropriados para a tarefa, mas que, como evidenciados na literatura mais atual, ainda necessitam de mais estudo para a obtenção de melhores resultados.

Percebe-se, pelos trabalhos mais recentes (2019-2020), o desafio de se encontrar arquiteturas mais robustas para a classificação, capazes de superar as atuais acurácias consideradas estado-da-arte para as diferentes bases de dados utilizadas. Neste sentido, o trabalho aqui proposto converge com os trabalhos atuais, porém, diferencia-se quanto ao *design* da arquitetura proposta e a linguagem em foco. Diferentemente de [17] e [18], que sugerem a aplicação de modelos híbridos CNN-LSTM, o DEEP propõe uma arquitetura composta por classificadores CNN especialistas para cada emoção, com o intuito de não propagar os erros, como pode ocorrer ao se colocar modelos sequencialmente.

Na busca de novas metodologias, o DEEP também converge com [19] e [20] na exploração de entradas diferentes das tradicionais, que contém somente *features* prosódicas e MFCC. Ao contrário de [19], que incorpora entradas de texto, e [20], que incorpora entradas de imagem dos espectogramas, o DEEP acrescenta uma nova *feature* de áudio, as cromáticas, estas que carregam consigo características tonais do som, visando otimizar o processo que exigiria a adição de modelos para a conversão de voz em áudio e imagem. Além dos fatores mencionados, o DEEP também se diferencia de [17], [18] [19] e [20], por ser o único modelo treinado para o reconhecimento de emoção na fala em língua portuguesa.

Na Tabela 3.1 são expostas as principais características das metodologias de reconhecimento de emoção na voz utilizadas nos trabalhos relacionados e no DEEP, separando arquiteturas tradicionais, que utilizam de um único modelo de ML, de novas arquitetu-

[8]https://www.kaggle.com/uwrfkaggler/ravdess-emotional-speech-audio

ras mais robustas, como arquiteturas híbridas, variantes ou especialistas. Também são separados conjuntos de entradas tradicionais para a tarefa, como as *features* prosódicas e MFCC extraídas do áudio, de novos conjuntos de entrada, como texto, imagens de espectograma e *features* cromáticas do áudio. Por fim, também diferencia-se a linguagem do *database* no qual os modelos foram treinados, separando língua portuguesa de línguas estrangeiras.

Tabela 3.1: Comparação das metodologias dos trabalhos relacionados e do DEEP

	Conjuntos de entrada tradicionais		**Novos conjuntos de entrada**			**Arquiteturas tradicionais**		**Novas arquiteturas**			**Linguagem do *Database***	
Trabalhos	Prosódicas	MFCC	Cromáticas	Texto	Imagens	ML (KNN, SVM)	DL (DNN, CNN)	Híbridas (CNN-LSTM, 2D CNN-LSTM)	Variantes (MLMHFA RNN, DSCNN)	CNNs especialistas	Línguas estrangeiras (inglês, chinês, alemão)	Língua portuguesa
Dellaert e outros (1996) [6]	X					X					X	
Kwon e outros (2003) [7]	X	X				X					X	
Pan e outros (2012) [11]	X	X				X					X	
Han e outros (2014) [12]	X	X					X				X	
Huang e outros (2019) [17]	X	X						X			X	
Zhao e outros (2019) [18]	X	X						X			X	
Ho e outros (2020) [19]		X		X					X		X	
Kwon e Mustaqeem (2020) [20]					X				X		X	
DEEP (2020)	X	X	X							X		X

Percebe-se que o DEEP se diferencia dos trabalhos mais recentes ao utilizar tanto uma arquitetura quanto um conjunto de entrada diferentes dos já abordados na literatura. Adicionalmente, adotamos como objetivo realizar a tarefa de reconhecimento de emoção na fala com o conjunto de dados na língua portuguesa, por não terem sido encontrados trabalhos na literatura que tenham realizado esta tarefa. Tal objetivo só é possível devido à recente publicação do trabalho de Neto, José e Filho (2018) [21] que contempla a construção de uma base de dados de emoção na voz em língua portuguesa, o VERBO. Por se tratar de uma base de dados nova, o estudo dos ganhos de performance da arquitetura proposta exige, também, a comparação dos resultados com o de arquiteturas tradicionais de um único modelo treinados na mesma base, no caso, utilizando-se de apenas um classificador CNN.

Capítulo 4

Uma arquitetura para o reconhecimento de emoção na voz em língua portuguesa

Neste capítulo é apresentada, em uma primeira instância, a estratégia adotada para o reconhecimento de emoções com base em padrões presentes no espectro sonoro gerado pela voz de falantes da língua portuguesa, o DEEP. Para tanto, o *dataset* escolhido, a extração das *features* e como o pré-processamento dos dados que foram utilizados para o desenvolvimento deste projeto serão descritos. Logo após, é apresentado o DEEP, detalhando-se a metodologia utilizada na sua modelagem. Ao final, especulamos sobre as possíveis aplicabilidades do DEEP em diversas áreas.

4.1 Visão geral da estratégia para desenvolvimento da proposta

Na Figura 4.1 é apresentada a estratégia adotada para a modelagem do DEEP. Como apresentado na Figura 4.1, três etapas principais são necessárias para o reconhecimento das emoções pelos padrões de fala: (i) aquisição de informações, (ii) extração de características e (iii) detecção das emoções.

A primeira etapa, Figura 4.1, Rótulo A - aquisição de informações, refere-se a necessidade da coleta de dados de voz devidamente categorizados para que seja possível a aplicação de métodos de aprendizagem de máquina supervisionado para o reconhecimento de emoção em fala de forma automática. Visando à isso, utilizamos a base de dados de emoção VERBO [21], a qual é composta por áudios falados na língua portuguesa do Brasil.

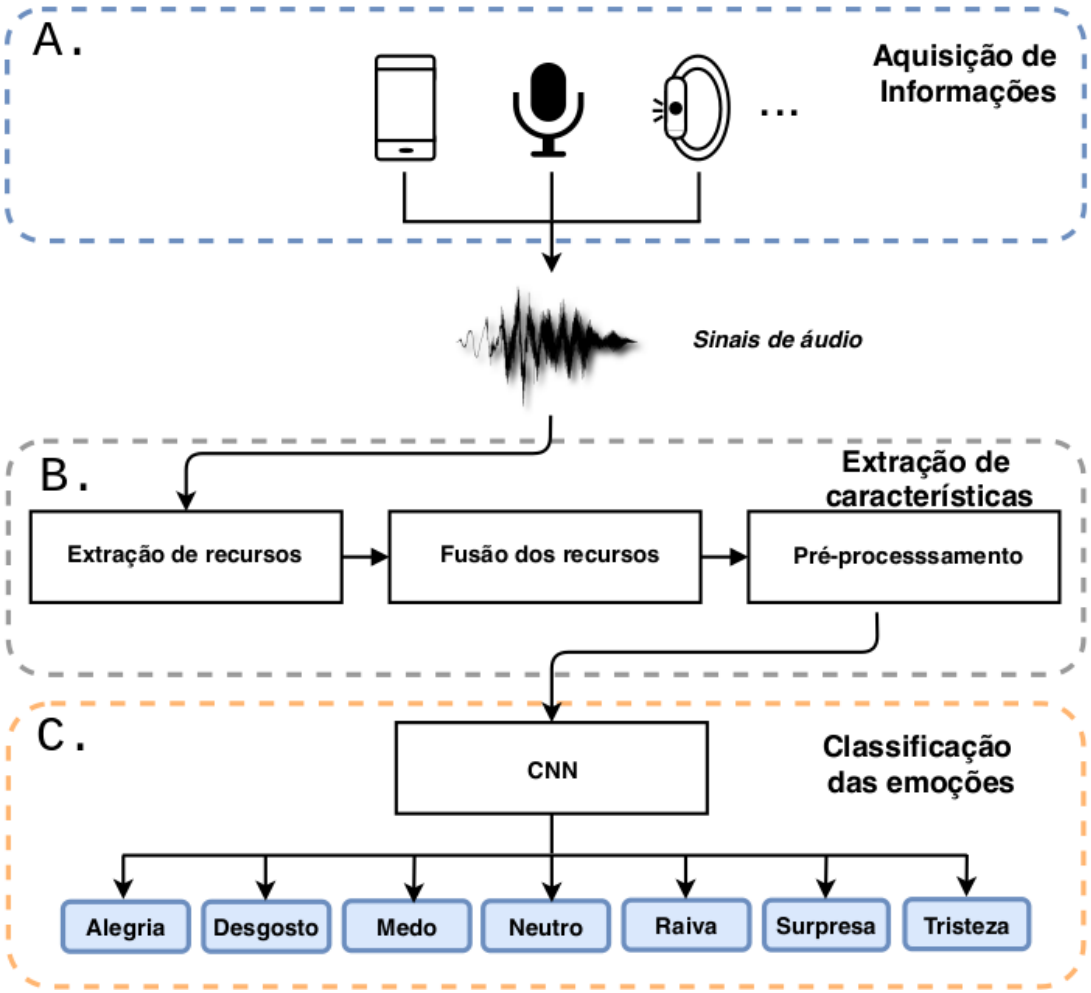


Figura 4.1: Visão geral da proposta

Na etapa seguinte, Figura 4.1, Rótulo B, refere-se a extração do conjunto de características (*features*) dos sinais de áudio presentes na base de dados. Tais *features* serão utilizadas como entrada para o DEEP. Pela leitura dos trabalhos relacionados, foram identificados as principais *features* aplicadas em modelos computacionais para reconhecimento de emoção e fala, entre elas: *features* MFCC, Prosódicas e Cromáticas. Assim, o primeiro passo para criar um conjunto de entrada adequado aos modelos da arquitetura proposta é a extração destas características e formação de um conjunto de dados reunindo essas informações, que será o *dataset* utilizado no desenvolvimento deste projeto. Em seguida, o conjunto destas *features* passa por uma etapa de pré-processamento, para enfim, ser adequada para utilização como entrada para os modelos.

Na etapa final, Figura 4.1, Rótulo C, o modelo proposto recebe as características como entrada e identifica as emoções com base em padrões do espectro sonoro da fala que foram utilizados como entrada. Neste trabalho optou-se pela arquitetura de CNNs para a criação dos modelos de classificação, seguindo uma metodologia de modelos especialistas

para cada emoção, ao invés de um modelo geral para o reconhecimento de todas as emoções analisadas. A seguir, cada etapa será apresentada.

4.2 Aquisição de informações da base de conhecimento

O primeiro passo no desenvolvimento de um modelo para o reconhecimento de emoções é a aquisição de uma base de dados de emoções. Para tanto, foi utilizado o *dataset Voice Emotion Recognition dataBase in Portuguese language* (VERBO) [21]. A decisão de utilizar o VERBO para o desenvolvimento deste trabalho se deu, principalmente, pela inexistência de classificadores de emoção para a voz em língua portuguesa, que, como levantado em [21], é a sexta língua mais falada do mundo, permitindo, assim, que países que possuam a língua portuguesa como idioma oficial possam ter um maior proveito desta tecnologia.

O VERBO é uma base de dados composta por 1176 arquivos de áudio com diferentes emoções na língua portuguesa do Brasil [21], que foi criada no Instituto de Matemática e Ciências da Computação da Universidade de São Paulo, ICMC-USP. O VERBO é composto por áudios balanceados foneticamente que variam de 2 à 5 segundos, gravados por doze atores brasileiros de diferentes idades e regiões, sendo seis homens e seis mulheres.

Os áudios gravados são compostos por quatorze frases que foram validadas por um profissional linguístico cobrindo todos os fonemas da língua portuguesa. As emoções representadas no VERBO são as seis emoções básicas proposto por Russel [22]: (i) alegria, (ii) nojo, (iii) medo, (iv) raiva, (v) surpresa e (vi) tristeza. Além destas, foi feita a adição de um sétimo estado emocional, denominado de (vii) neutro. O modelo circumplexo de Russel caracteriza e dispõe as emoções pelo seu nível de excitação e prazer [21], como apresentado na Figura 2.1. O modelo de Russel permite a compreensão das emoções de forma interconectada, e não somente individual, relacionando-as com base em suas proximidades apresentadas dentro do círculo.

4.3 Extração de características (*features*)

Nesta seção, é apresentada como foi realizada a extração das características do VERBO para o reconhecimento de emoções pelos padrões de fala da língua portuguesa. Inicialmente, é apresentada como a representação bidimensional do sinal foi implementada. Em seguida, é apresentada como a extração das *features* foi executada. Por fim, é apresentado o pré-processamento sobre as *features* extraídas dos áudios. Para realizar a extração das

features deste projeto, utilizamos a ferramenta OpenSMILE [29]. Esta ferramenta também foi utilizada na extração de *features* de áudios para reconhecimento de emoção na voz em trabalhos recentes como o de Ho, Yang, Kin e Lee (2020) [19].

4.3.1 Representação bidimensional do sinal

O modo como as frases são pronunciadas levam a inúmeras interpretações pelo ouvinte. As emoções expressas pela voz podem ser diferenciadas do sinal de voz e como o espectro de energia do sinal se comporta. Isso significa que emoções cuja expressão vocal é caracterizada por exaltação e choro tendem a carregar mais energia do que emoções cuja expressão é mais suave.

A arquitetura aqui proposta realiza a detecção de emoções por meio de métodos de DL, estes que necessitam de uma representação numérica que expresse as informações necessárias dos dados de entrada para que seja possível determinar padrões de aprendizagem para o modelo. Em razão disso, é necessário representar as informações dos áudios do conjunto de dados de entrada de forma que expressem informações relevantes às emoções de forma numérica; as *features* do nosso modelo.

Portanto, o principal objetivo da etapa de extração de características é transformar os arquivos de áudio, que estão no formato *.wav*, para uma representação bidimensional, expressando o tempo e outras informações relevantes de cada *feature* em específico. Isso segue verdade para todas as *features* selecionadas para o trabalho que neste caso são: *MFCC*, Cromáticas e Prosódicas. Para o MFCC, teremos um conjunto de coeficientes Cepstrais de Frequência Mel ao decorrer do tempo, para as *features* Cromáticas, teremos informação da classificação tonal do áudio ao decorrer do tempo e para as *features* Prosódicas temos informações de entonação, estresse, tremor e ritmo, ao decorrer do tempo. A Figura 4.2 apresenta o processo de extração das *features*, elucidando as três pré-processos de extração, (i) Enquadramento, do inglês *framing*, (ii) criação de janelas, do inglês *windowing* e (iii) transformada rápida de Fourier, don inglês *Fast Fourier Transform* (FFT), que são necessários para se obter a representação bidimensional dos áudios e, então, ser possível extrair as *features* desejadas: *MFCC*, Prosódicas e Cromáticas.

O processo começa com a etapa de Enquadramento, Figura 4.2, Rótulo A. Nesta primeira etapa, dividimos o sinal do áudio em diversos quadros, ou *frames*, com o objetivo de produzir unidades de frequências que podem ser processadas e analisadas individualmente, ao invés de analisar o áudio inteiro de uma só vez. A divisão dos quadros é feita por valores fixos do seu comprimento e do intervalo entre quadros. Normalmente, estes valores são determinados para que haja algum grau de sobreposição entre os quadros. Por meio de uma visão holística, escolhemos quadros com $50ms$ de comprimento e com intervalos de $10ms$, como apresentado na Figura 4.3.

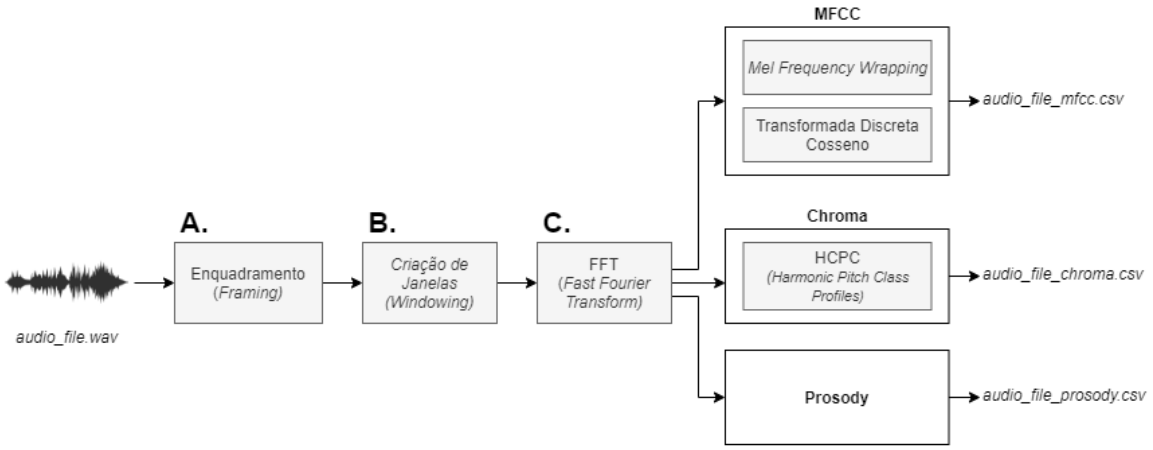


Figura 4.2: Processo de extração de *features*

A etapa seguinte é a de *Windowing*, Figura 4.2, Rótulo B. Nesta, para eliminar algum tipo de distorção ou descontinuidade presente no espectro do sinal, é aplicado uma função de janela de Hamming com ganho unitário para a obtenção de esquadrias com arestas atenuadas em relação ao centro para cada quadro. Este tipo de pré-processamento é bastante utilizado em processos de análises de sinal que envolvem transformadas discretas de Fourier para adaptar o sinal da janela ao padrão de sinal esperado como parâmetro para a função. Como consequência da aplicação da função de janela de *Hamming*, as informações que ficam nas extremidades de cada quadro são perdidas. Entretanto, isso foi remediado pela sobreposição causada no processo de enquadramento ao escolher um valor de intervalo entre quadros menor que o valor do comprimento do quadro.

Os processos supracitados são o pré-processamento necessário para a transformação do sinal de áudio na representação bidimensional desejada. Tal representação é alcançada na terceira etapa, com a aplicação da FFT, Figura 4.2, Rótulo C, que é uma implementação otimizada em termos de tempo de execução da transformada discreta de Fourier, do inglês *Discrete Fourier transform* (DFT).

4.3.2 Extração de *features* utilizando *OpenSmile*

Tendo-se, agora, a representação bidimensional do sinal, tem-se o necessário para os passos mais específicos inerentes à extração de cada uma das *features* selecionadas. Para tanto, estas *features* são detalhadas a seguir:

- *MFCC*: Além de ser uma das formas de representação espectral do som, é uma das *features* mais aplicada para tarefas de reconhecimento de fala [30] [31] e de emoção

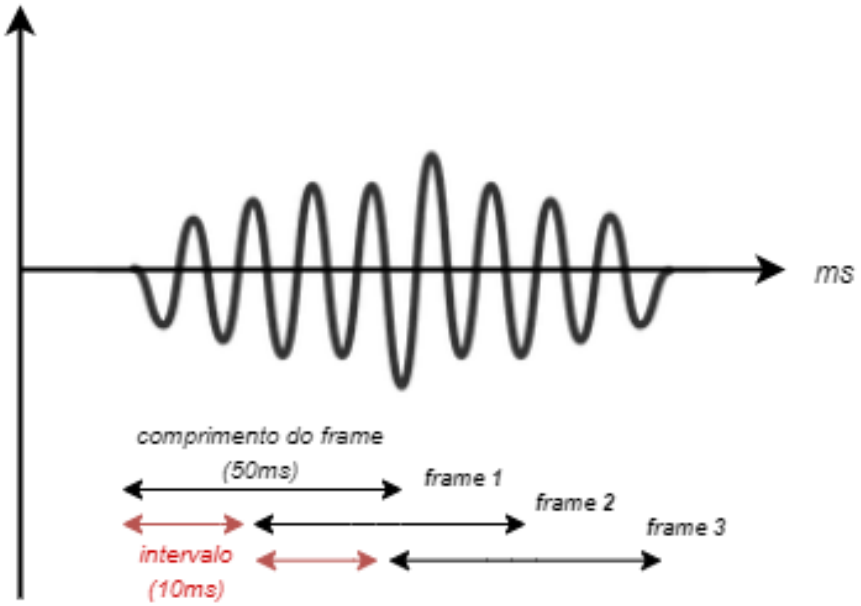


Figura 4.3: Processo de enquadramento

na fala [11] [7]. Os MFCCs são compostos por um conjunto de coeficientes, que coletivamente formam um MFC, *Mel-Frequency Cepstrum*, que é uma representação da densidade espectral a curto prazo de um som. Para obter estes coeficientes, é realizado o processo chamado de *Mel-Frequency Wrapping*. Sua ideia é de transformar a saída da FFT em uma instância de uma escala Mel [32], que é uma escala construída baseada em tons que são perceptivelmente equidistantes ao ouvido humano. Após o processo de *Mel-Frequency Wrapping*, aplicamos a Transformada Discreta de Cosseno nos resultados obtidos do passo anterior. O objetivo desse passo é obter o *Cepstrum* [33], que é utilizado para investigar estruturas e obter informação sobre fala em um espectro.

- *Cromáticas*: A *feature* cromática é composta por doze classes tonais. A combinação das informações relacionadas ao tom de um som é chamada de HPCP (*Harmonic Pitch Class Profiles*), que descreve as características tonais do som. Tal *feature* utilizada para análise musical, por exemplo, um método computacional para improvisação musical por meio da detecção automática da escala de acordes [34]. A decisão de aplicá-la para a tarefa de reconhecimento de voz foi pautada na tonalidade da voz que auxiliou a distinguir as emoções, principalmente considerando a diferença tonal da expressão de uma emoção por cada ator e atriz, diferente dos trabalhos da literatura que não a utilizam.

- *Prosódicas*: A *feature* prosódica relaciona-se às características fonéticas e linguísticas do som, as quais se associam com elementos interligados da fala, ao contrário de segmentos fonéticos individuais. Tal *feature* utilizada em sistemas de reconhecimento de voz [6] [11] para informar sobre aspectos tais como entonação, tremulação, estresse e ritmo. Com essa *feature* é possível compreender a sonoridade, a tremulação e a afinação do discurso humano, padrões presentes no espectro sonoro que auxiliaram no treinamento do modelo.

O resultado final do processo de extração são arquivos *csv* para cada uma das *features*. Para o MFCC, o *.csv* contém treze coeficientes mel-cepstrais para cada um dos quadros do arquivo de áudio de entrada. Para a Cromática, o *.csv* contém doze valores de peso referentes a cada uma das doze possíveis classes tonais para cada um dos quadros do arquivo de áudio de entrada. Para a Prosódica, o arquivo *.csv* contém valores de $F0$, *loudness*, *jitter* e *shimmer* para cada um dos quadros do arquivo de áudio de entrada, como apresentado na Tabela 4.1.

4.3.3 Pré-processamento

A Tabela 4.1 apresenta a descrição do *Dataset* obtido após a etapa de extração das *features*. É este *dataset* que é utilizado no treinamento e validação do DEEP, como será apresentado na Seção 5.1 no Capítulo 5 5.3.

Tabela 4.1: Descrição do *Dataset* construído com base no VERBO.

Descrição do Dataset	
Features	MFCC, Cromática e Prosódica
Número de quadros	540
Valores por quadro	13 MFCCs, 12 cromáticas e 6 prosódicas
Número de instâncias	1176
Valores Nulos	Nenhum
Tarefas Associadas	Classificação de emoção
Formato dos arquivos de áudio	.WAV
Emoção (*target*)	Neutro (0), Desgosto (1), Medo (2), Alegria (3), Raiva (4), Surpresa (5), Tristeza (6)

Como ilustrado na Figura 4.2, o resultado do processo de extração das *features* são arquivos *csv* separados para cada *feature*. Para compor o *Dataset* descrito na Tabela 4.1, foram combinados os valores dos arquivos de *csv* das *features* MFCCs, Prosódicas e Cromáticas para cada áudio do conjunto de dados do VERBO. Desta forma, uma linha do *Dataset* possui f quadros, e cada um destes quadros possuem 31 valores, sendo destes 13 valores de MFCC, 12 valores das *features* cromáticas e 6 valores das *features* prosódicas.

Para que este *dataset* fosse utilizado na etapa de treinamento, foi necessário um último tratamento. Como citado na Subseção 4.3.1, os áudios foram divididos em uma série de quadros menores e de tamanho fixo, sendo, em seguida, realizado a extração das *features* desejadas para cada um destes quadros. Entretanto, como os áudios possuem comprimento variável, a quantidade de quadros f e, consequentemente, a quantidade de *features* extraídas para cada áudio variam, o que se torna um problema. A irregularidade dos quadros não é uma tarefa trivial de ser resolvida, uma vez que para treinar o nosso modelo todas as entradas devem está no formato matricial. Em razão disso, selecionamos a entrada com o maior número de quadros $f_0 = 540$, e, para cada entrada com menos de 540 quadros, realizamos o processo de *padding*, completando a entrada com valores nulos até alcançar o número de quadros desejáveis. Como fizemos a extração de 31 características totais (somando as *features* MFCC, Cromática e Prosódica), para cada quadro, o valor nulo usado é uma matriz 1-dimensional com 31 valores 0.

4.4 DEEP - *DEtection of voice Emotion in Portuguese language*

Esta seção apresenta o DEEP, uma arquitetura para o reconhecimento de emoções com base no espectro sonoro da voz de falantes da língua portuguesa. Com este modelo, objetiva-se a aplicação de métodos de IA e aprendizagem supervisionadas para a tarefa de reconhecimento de que emoção que um áudio expressa, replicando as categorizações existente no *dataset* VERBO. Para tanto, o *DEEP* foi modelado com base em CNNs especialista para reconhecimento de cada emoção, com o objetivo de conseguir distinguir quais emoções um áudio pode ou não apresentar.

Uma vez apresentados, no Capítulo 3, os desafios de pesquisa e o estado da arte relacionados ao desenvolvimento deste trabalho, o DEEP se destaca por combinar diferentes *features* de Descritores de Baixo Nivel (MFCC, Cromáticas e Prosódicas) e pela sua arquitetura inovadora contendo modelos especializados para cada emoção. Portanto, o principal objetivo do DEEP é prover maior precisão para o reconhecimento de emoção na voz com base em CNNs especialistas para cada emoção e da combinação de *features* de Descritores de Baixo Nível, tendo com meta específica diferenciar de maneira significativa as emoções humanas por meio de voz.

Serão abordados, em uma primeira instância, o *design* da arquitetura proposta para o DEEP e, em seguida, a arquitetura e o processo de treinamento dos classificadores especialistas que o compõem.

4.4.1 Design da arquitetura proposta para o DEEP

A Figura 4.4 apresenta o *design* da arquitetura do DEEP. Os dados de entrada deste modelo são as *features* MFCC, Cromáticas e Prosódicas extraídas dos arquivos de áudio presentes no VERBO. As entrada de dados contendo as *features* selecionadas é, então alimentado aos modelos especialistas, sendo um modelo para cada emoção. Portanto, o principal objetivo do DEEP é prover maior precisão para o reconhecimento de emoção com base nas *features* analisadas, tendo com meta específica diferenciar de maneira significativa as emoções humanas por meio de voz.

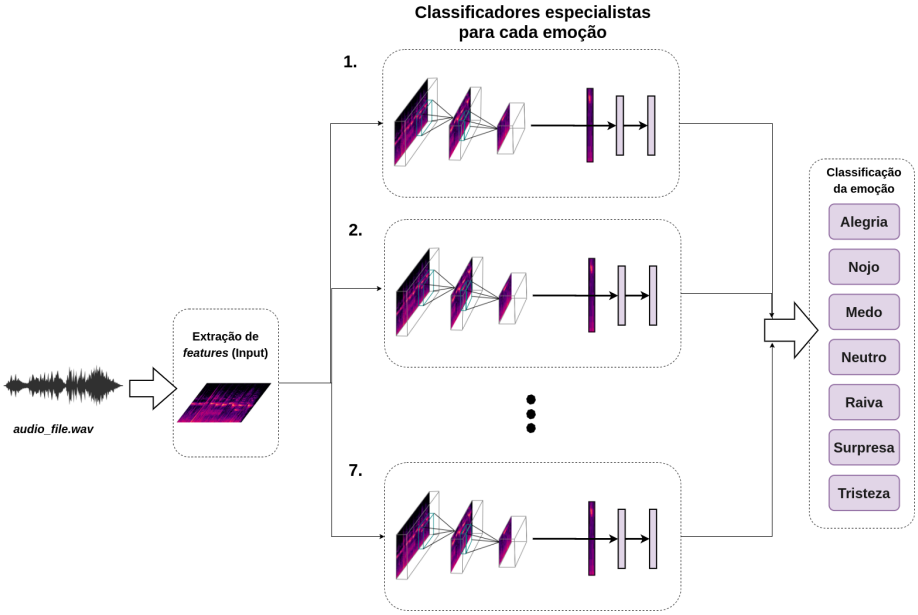


Figura 4.4: *Design* da arquitetura proposta para o DEEP

Cada modelo especialista no DEEP gera uma probabilidade para duas classes que indicam se a emoção identificada pertence à categoria de sua especialidade ou não por meio de uma camada de ativação final que utiliza de uma função *softmax*. Uma função *softmax* é uma generalização de uma função logística para múltiplas dimensões, como apresentada na Equação 4.1:

$$\phi i = \frac{e^{Z} i}{\sum_{j \in Group} e^{Z} j} \tag{4.1}$$

Onde i representa o índice do neurônio de saída, j representa os índices de todos os neurônios de um nível e Z designa o vetor de neurônios de saída. A função *softmax* é utilizada pela rede neural para normalizar os dados de saída em uma distribuição probabilística entre as possíveis classes da saída, que, neste caso, são duas para cada um dos

modelos especialistas. A partir dos dados de saída dos classificadores CNN, determina-se uma das 7 categorias da emoção possíveis.

Para um melhor entendimento do *design* da arquitetura proposta, na próxima seção são descritos, em mais detalhes, os classificadores especialistas.

4.4.2 Modelos especialistas para cada emoção

Para o desenvolvimento dos classificadores especialistas foram utilizadas CNNs, a qual foram implementadas por meio de uma *deep learning framework API* para *python* denominada *Keras*. Ressalta-se que este projeto foi todo desenvolvido em código aberto e está disponível no *Github* [1].

CNNs são compostas por uma série de camadas de convolução, *convolutional layers*, responsáveis pela extração e aprendizado de *features* adicionais ao filtrar as matrizes de entrada seguido pelo conjunto de camadas totalmente conectadas, *fully connected*, de classificação tradicional de redes neurais. CNNs, recentemente, vêm sido aplicadas em sistemas de reconhecimento de fala com taxas de erro reduzidas em comparação a outras redes neurais como as DNNs [16], característica que também foi observada neste trabalho durante a etapa de modelagem e que foi crucial na escolha desta arquitetura.

Para uma melhor visualização do *DEEP*, a Figura 4.5 apresenta a arquitetura das CNNs treinadas. Cada instância do *dataset X* utilizado de entrada, *input*, Figura 4.6 Rótulo A, que é o conjunto de *features* extraídas de um arquivo de áudio, está sendo representada pela imagem do espectrograma do áudio. Em seguida, as instâncias do *input* passam por uma série de camadas de convolução, *convolutional layers*, Figura 4.6 Rotulo B, sendo filtradas até chegarem nas camadas totalmente conectadas, *fully connected*, Figura 4.6, rotulo C, para que seja feita a classificação.

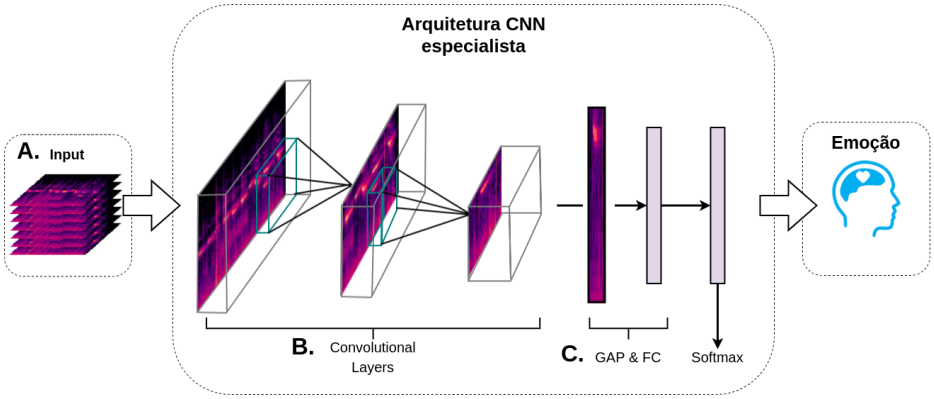


Figura 4.5: Arquitetura CNN utilizada nos classificadores especialistas.

[1] https://github.com/lucasmoutinho/emotion-recognition-by-voice

A Figura 4.6, apresenta as camadas presentes nos modelos CNN especialistas. O modelo é composto por uma camada de entrada, camadas de convolução e camadas totalmente conectadas. Uma explicação breve de cada uma destas camadas é feita em seguida.

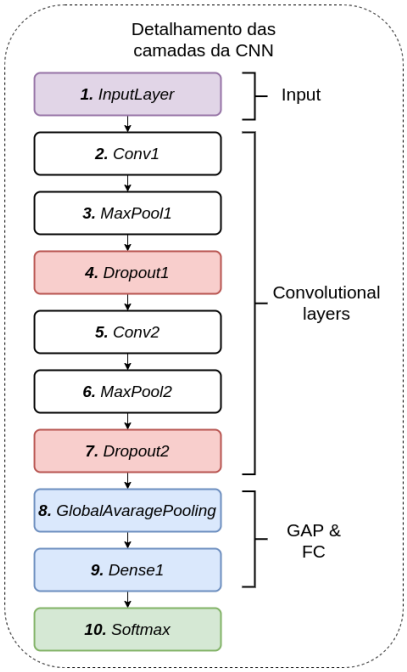


Figura 4.6: Detalhamento das camadas utilizadas na CNN dos classificadores especialistas.

1. **Input Layer:** Camada de entrada da CNN. Aceita dados no formato de entrada (A, B, C), em que A corresponde ao número de linhas do *dataset*, B ao número de colunas ou *features* e C ao número de canais na matriz de entrada. Os dados de entrada para treino e teste são ajustados para este formato de entrada de matrizes e padronizados para terem apenas 1 canal por se tratarem de dados de voz.

2. **Convolution layer 1:** Primeira camada de convolução da CNN. As camadas de convolução são a principal característica das CNNs. Nestas são aplicados filtros para ressaltar padrões encontrados nos dados de entrada, neste caso, enaltecendo algumas informações das *features* extraídas dos áudios. A extração de *features* a partir da matriz de *input* se dá por meio da aplicação de uma função matemática que utiliza um parâmetro *filter*. O valor deste parâmetro é ajustado na etapa de hiper-parametrização e varia para cada classificador.

3. **Max Pooling layer 1:** Primeira camada de mineração, *pooling*. Responsável por reduzir o tamanho espacial da matriz, reduzindo o poder computacional necessário

para processar a *feature* e extraindo *features* dominantes. Seu parâmetro, *pool size*, utilizado na redução foi determinado como 2 para todos os classificadores.

4. **Dropout Layer 1:** Primeira camada de *dropout*, que ignora uma quantidade de neurônios para prevenir *over-fitting*. O parâmetro que especifica a taxa de *dropout* é ajustado na etapa de hiper-parametrização e varia para cada classificador.

5. **Convolution layer 2:** Segunda camada de convolução da CNN. O valor do parâmetro *filter* desta camada é ajustado na etapa de hiper-parametrização e varia para cada classificador.

6. **Max Pooling layer 2:** Segunda camada de mineração, *pooling*. Seu parâmetro, *pool size*, utilizado na redução foi determinado como 2 para todos os classificadores.

7. **Dropout layer 2:** Segunda camada de *dropout*. O parâmetro que especifica a taxa de *dropout* é ajustado na etapa de hiper-parametrização e varia para cada classificador.

8. **Global Avarage Pooling Layer:** Camada de mineração que calcula a média de cada *feature* mapeada ao final das camadas de convolução e alimenta o resultado para uma camada totalmente conectada de classificação.

9. **Dense layer:** A única camada totalmente conectada da CNN. Alimenta a camada de ativação para a classificação final da rede.

10. **Softmax Activation layer:** A camada de ativação do final da rede neural. Utiliza uma função de ativação *softmax* para classificar entre as duas categorias possíveis de *outputs*: 0 para indicar que não é a emoção desejada e 1 indicando que é a emoção desejada.

4.5 Aplicabilidade da solução proposta

O *DEEP* foi desenvolvido com o intuito de poder ser aplicado em diversos contextos práticos. O grande foco deste trabalho e, consequentemente, de todas as etapas abordadas na seção de desenvolvimento é a implementação do *DEEP*, cuja arquitetura pode ser visualizado na Figura 4.4. É de interesse dos autores deste trabalho não somente a análise e a comparação das métricas obtidas com a arquitetura aqui proposta, como também ressaltar a relevância das possíveis aplicabilidades deste modelo.

Análise de sentimentos por meio da voz é uma tarefa que vêm sendo cada vez mais explorada no campo de IA, devido a sua vasta gama de possíveis aplicações. Dentre as aplicações que o DEEP pode ser utilizado, menciona-se o seu uso em *smart environments*

[35] e *smart assistants*, como a *Amazon Alexa* [4], em que a aplicação computacional pode ter comportamentos diferentes, dependendo da emoção do usuário.

A recomendação de serviços com base na emoção detectada na voz do usuário também é uma das possibilidades. Na pesquisa de Mano et al.(2019) [5], é observado um aumento de 60% na satisfação dos usuários ao se utilizarem de reconhecimento de emoção para recomendação da próxima música em um aplicativo tocador de músicas. Além dessa possibilidade, outra aplicação seria na utilização de serviços de *telemarketing* para a detecção automática do nível de satisfação do cliente com o atendimento ou em sistemas de voz interativos, como os utilizados em centrais de atendimento, para a geração de respostas customizadas, sendo que ambas opções são serviços que dependem de conversas por telefone.

Outra aplicação diz respeito no uso do *DEEP* na legenda automática de vídeos e vídeo-chamadas. Com a detecção de emoção é possível não somente transcrever o que foi dito, mas como colocar a emoção do que foi falado. Esta informação extra da emoção nas legendas é de muito valor, por exemplo, para grupos de pessoas com deficiência auditiva para que possam ter um maior entendimento da conversa abordada em vídeo.

O reconhecimento de emoção na voz também pode ser utilizado em tecnologias de fim lúdico, como videogames. Em jogos de tema musical, o jogador pode ter diferentes respostas da plateia dependendo da emoção colocado ao se cantar uma música, tornando a experiência proporcionada pelo jogo mais imersiva ao usuário. Um exemplo de detecção de emoção na voz já utilizado para aumentar a imersão em um jogo é na série de jogos de futebol *Fifa*, onde o juiz pode ter diferentes comportamento de acordo com o que o jogador expressa na fala.

Capítulo 5

Resultados Experimentais

Neste capítulo, são apresentados os resultados experimentais, as avaliações de desempenho e a metodologia utilizada na validação do *DEEP*. Para isso, o *DEEP* foi avaliado sob duas perspectivas:

- **Avaliação sob a perspectiva da otimização dos hiper parâmetros**: Nessa perspectiva, apresentada na seção 5.1, objetivamos apresentar os resultados alcançados a partir do processo de otimização dos hiper parâmetros dos modelos especialistas e, para isso, utilizamos a biblioteca *hyperopt* [26] na realização desse processo. Temos como finalidade, assim, encontrar a configuração ótima para cada um dos modelos.
- **Avaliação dos resultados em comparação com o modelo *baseline***: Nessa perspectiva, apresentada na seção 5.2, apresentamos os resultados obtidos com o *DEEP*, comparando-o com um modelo *baseline*. O modelo *baseline* foi construído com o intuito de representar a abordagem mais tradicional vista na literatura relacionada, sendo criado um único modelo para classificação [16] [17] [36].

5.1 Resultados sob a perspectiva da otimização dos hiper parâmetros do *DEEP*

Nesta seção, apresentaremos os resultados sob a perspectiva do processo de otimização dos hiper parâmetros, a fim de encontrar configurações ótimas para os modelos do *DEEP*.

5.1.1 Descrição do Cenário do Experimento

Na seção 4.4.1, definimos a arquitetura proposta para o DEEP, como também apresentamos os modelos CNN especialistas desenvolvidos para cada emoção presente no conjunto de dados VERBO [21].

Com o intuito de encontrar o melhor conjunto de configurações de hiper parâmetros para o DEEP, utilizou-se a biblioteca em *python* chamada *hyperas* [1]. O *Hyperas* é uma implementação do *Hyperopt* específica para modelos construídos utilizando o *framework Keras* [2]. Nesta seção, iremos apresentar a evolução dos resultados com base nas etapas de treinamento do *hyperopt* e nos melhores parâmetros. O processo de hiper parametrização tem como objetivo otimizar os parâmetros do modelo especialista, utilizando o algoritmo *Tree-structured Parzen Estimator* [37] para estimar a melhor configuração de parâmetros dentro de uma lista de possíveis combinações. A Tabela 5.1 apresenta os parâmetros que foram utilizados no processo de hiper parametrização por meio do *hyperas*.

Tabela 5.1: Valores possíveis para cada camada da CNN.

Nome da Camada	**Valores Possíveis**
Convolutional Layer 1	32, 64, 128, 256
Dropout Layer 1	0.2, 0.4, 0.6
Convolutional Layer 2	32, 64, 128, 256
Dropout Layer 2	0.2, 0.4, 0.6
Otimizadores	Adam, SGD, RMSProp

No processo de treinamento com o *hyperas*, é importante visualizar o treinamento de cada modelo especialista como independente um do outro. Temos sete modelos especialistas, um para cada emoção contida no conjunto de dados. Para cada um destes modelos, compomos o conjunto de treinamento por meio do processo de subamostragem, do inglês *undersampling*, de forma que são utilizadas todas as 167 amostras da emoção respectiva do modelo, e outras 168 amostras das demais seis emoções, divididas igualmente. Para exemplificar, considere que o conjunto de treinamento do modelo especialista da emoção *neutro* é composto por 167 instâncias de áudios que representam a emoção *neutro* e outras 168 instâncias das outras emoções: 28 instâncias de *alegria*, 28 instâncias de *tristeza*, 28 instâncias de *surpresa*, 28 instâncias de *raiva*, 28 instâncias de *nojo* e 28 instâncias de *medo*. Adicionalmente, por se tratarem de modelos especialistas, as 167 instâncias da emoção *neutro* compõem as instâncias *positivas* do modelo, e as instâncias das emoções restantes compõem as instâncias *negativas*. O *undersampling* é feito com o intuito de balancear o conjunto de dados utilizados para o treinamento, evitando, assim, que os modelos sejam enviesados em decorrência do balanceamento entre classes.

[1] *Hyperas*: https://github.com/maxpumperla/hyperas
[2] *Keras*: https://github.com/keras-team/keras

Para os modelos especialistas, transformamos as *labels* das classes, de forma que, caso a *label* seja a da emoção do modelo, seu valor será 1; caso contrário, seu valor será 0. O *hold-out* foi utilizado para gerar os modelos da DEEP, separando o *dataset* em 70% para treino e 30% para validação. O treinamento foi feito para 500 épocas, com o *batch size* fixado em 64. A função de perda utilizada foi a *categorical crossentropy*, esta que é empregada em classificações multi-classe para diferenciar duas distribuições de probabilidade discretas, evidenciando se um exemplo pertence à categoria da emoção desejada com probabilidade 1 e a outras com probabilidade 0. Para este trabalho, o treinamento é executado 10 vezes durante o processo de hiper parametrização. O *hyperas* na etapa i utiliza os resultados de acurácia das etapas anteriores $j < i$ para escolher uma nova configuração de parâmetros para o treinamento.

5.1.2 Análise dos hiper parâmetros selecionados para o DEEP

Tabela 5.2: Resultados de Acurácia e F1-Score ao decorrer das etapas do processo de hiper parametrização

	Acurácia							**F1 Score**						
Trials	Neutro	Nojo	Medo	Alegria	Raiva	Surpresa	Tristeza	Neutro	Nojo	Medo	Alegria	Raiva	Surpresa	Tristeza
1	0.6627	0.6766	0.5345	0.6377	0.6617	0.6976	0.6209	0.6565	0.6932	0.5141	0.7070	0.6686	0.7139	0.6116
2	0.6030	0.6796	0.5405	0.6228	0.5284	0.6617	0.6328	0.4784	0.7003	0.5591	0.6786	0.6776	0.5022	0.6120
3	0.6239	0.6916	0.5345	0.6168	0.6269	0.7156	0.6388	0.6062	0.7082	0.5401	0.6923	0.6313	0.7246	0.6084
4	0.5791	0.6407	0.5045	0.5060	0.5493	0.7335	0.5910	**0.7056**	0.5489	0.6323	0.6680	**0.6834**	0.7262	0.3116
5	0.6418	0.6467	**0.5616**	0.6377	0.5970	0.7036	0.6478	0.6129	0.6446	**0.6894**	0.7027	0.5970	0.7195	**0.6967**
6	0.6448	0.6946	0.5375	0.6437	0.6537	0.7096	0.6328	0.6293	0.7151	0.2936	0.7032	0.6420	0.7357	0.5941
7	**0.6866**	0.6707	0.5345	0.6467	0.6090	0.7156	0.6388	0.6729	0.6893	0.6709	0.6828	0.6246	0.7164	0.5724
8	0.6627	0.6796	0.5045	0.6587	**0.6739**	**0.7725**	0.5940	0.6586	0.6272	0.0000	0.6816	0.6250	0.7516	0.4472
9	0.6269	**0.7426**	0.5045	0.6587	0.6060	0.7695	0.6239	0.5645	**0.7640**	0.0000	**0.7164**	0.6071	**0.7674**	0.5532
10	0.6418	0.7096	0.5315	**0.6677**	0.6060	0.7305	**0.6626**	0.6273	0.7139	0.6763	0.6856	0.5976	0.7256	0.6093

A Tabela 5.2 apresenta os resultados de acurácia e *F1 Score* para cada emoção no decorrer das etapas do processo de otimização dos hiper parâmetros dos modelos. Os valores apresentados são as médias das 10 melhores épocas do processo de treinamento. Nesta Tabela, os resultados da *Trial 1* são correspondentes à avaliação do modelo, utilizando uma mesma configuração de hiper parâmetros inicial. Dessa forma, observando tanto os resultados de acurácia quanto os de *F1-Score* apresentados na Tabela 5.2 e comparando-os com os valores da *Trial 1*, que correspondem aos valores de avaliação dos modelos com a configuração inicial, é possível afirmar que, com o processo de otimização dos hiper parâmetros, conseguimos encontrar configurações melhores das que a configuração inicial para todas as emoções, pois existem *Trials* posteriores com resultados superiores aos da inicial.

Em consequência dos resultados apresentados, escolhemos os modelos que apresentaram maior acurácia para cada emoção para integrá-los na arquitetura proposta ao DEEP.

Tabela 5.3: Configuração final de parâmetros dos modelos especialistas

Emoção	*CL 1*	*DL 1*	*CL 2*	*DL 2*	Otimizador
Tristeza	64 filtros	0.2	64 filtros	0.4	SGD
Nojo	256 filtros	0.6	64 filtros	0.4	RMSprop
Alegria	128 filtros	0.6	256 filtros	0.6	RMSprop
Surpresa	32 filtros	0.4	256 filtros	0.6	Adam
Raiva	256 filtros	0.6	64 filtros	0.4	RMSprop
Medo	128 filtros	0.6	256 filtros	0.6	Adam
Neutro	128 filtros	0.4	256 filtros	0.6	RMSprop

A Tabela 5.3 apresenta as configurações ótimas de hiper parâmetros encontradas para cada modelo especialista e que permitiram o alcance de maior performance individual para cada um.

5.2 Avaliação de desempenho do *DEEP* com comparação ao modelo *Baseline*

Esta seção tem como objetivo avaliar o desempenho do DEEP para o reconhecimento das emoções dos falantes da língua portuguesa, comparando-o com um modelo *baseline*. O modelo *baseline* foi desenvolvido aos moldes de uma arquitetura de abordagem tradicional, que contém um único modelo para a classificação das emoções. Apresentaremos, inicialmente, a descrição do cenário, explicando como foi realizado o treinamento e, em sequência, apresentaremos e analisaremos os resultados obtidos.

5.2.1 Descrição do Cenário do Experimento

Para avaliar a eficiência do *DEEP*, desejamos comparar os resultados obtidos com os de uma arquitetura tradicional, seguindo a estratégia apresentada nos trabalhos relacionados [16] [17] [36]. Com essa finalidade, modelamos uma arquitetura tradicional que utiliza um único modelo CNN classificador para todas as emoções, denominado de **modelo *baseline***. Optamos por construir um modelo *baseline* ao invés de comparar diretamente com resultados de modelo apresentado na literatura relacionada devido à falta de trabalhos que utilizem o VERBO [21] ou *datasets* em português como um todo.

O modelo *baseline* é uma CNN com uma arquitetura semelhante à dos modelos especialistas. A principal diferença do modelo *baseline* com o *DEEP* é que os modelos especialistas do *DEEP* tiveram seus hiper parâmetros otimizados para cada uma das emoções, enquanto que o modelo *baseline* utiliza uma única configuração de hiper parâmetros para todas as emoções. Para realizar uma avaliação justa com o *DEEP*, o *baseline*

Tabela 5.4: Configuração de Hiper Parâmetros para o modelo *baseline*.

Nome da Camada	**Valor**
Convolutional Layer 1	128
Dropout Layer 1	0.2
Convolutional Layer 2	64
Dropout Layer 2	0.2

Tabela 5.5: Comparação dos resultados dos modelos especialistas com o modelo *baseline*

Emoção	Acurácia DEEP	F1-*Score* DEEP	Acurácia *Baseline*	F1-*Score Baseline*
Neutro	**0.7285**	0.3378	0.6568	0.3232
Nojo	**0.7482**	0.3858	0.6764	0.3727
Medo	**0.7843**	0.3046	0.5400	0.2630
Alegria	**0.7637**	0.3199	0.6372	0.3311
Raiva	**0.6374**	0.3371	0.6372	0.3301
Surpresa	**0.8556**	0.4900	0.7254	0.4511
Tristeza	**0.8505**	0.3326	0.6274	0.3141

passou pelo mesmo processo de hiper parametrização, envolvendo os mesmos parâmetros escolhidos para os modelos do *DEEP*. A configuração de hiper parâmetros alcançada está apresentada na Tabela 5.4.

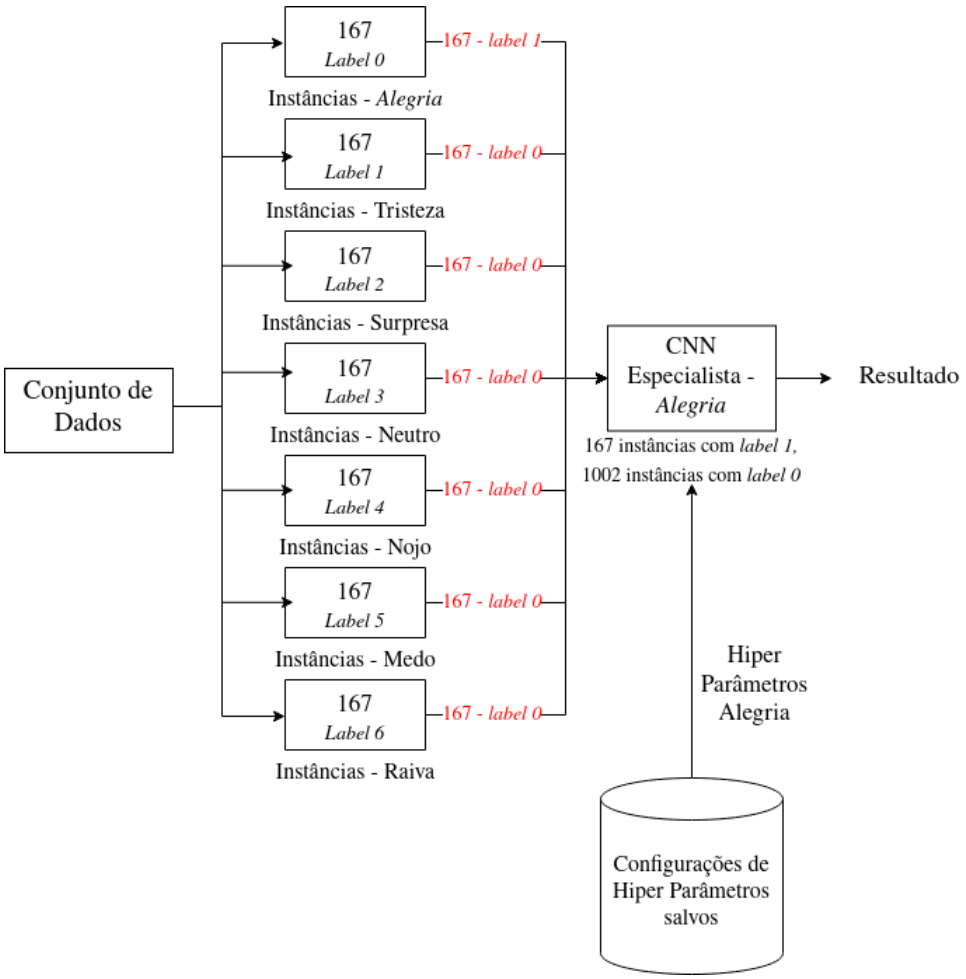
Para a comparação dos modelos, efetuamos o processo de avaliação utilizando o conjunto de dados completo do VERBO [21]. Para os modelos especialistas, o processo de avaliação está exemplificado para a emoção Alegria na Figura 5.1. Para cada uma das emoções, carregamos suas configurações de hiper parâmetros, obtidas no processo descrito na Seção 5.1, e fizemos as predições para todas as instâncias do conjunto de dados. O processo de avaliação do modelo *baseline* é bastante similar, com a diferença de que é usada a mesma configuração de hiper parâmetros para as 7 emoções.

5.2.2 Comparação dos resultados experimentais da arquitetura proposta com o modelo *baseline*

Uma das questões principais que planejávamos responder sobre a arquitetura proposta é: uma arquitetura composta com CNN especialistas para detecção de emoção nos retorna melhores resultados do que um único modelo de CNN?

A Tabela 5.5 apresenta os resultados de acurácia do *DEEP* quando comparados com o modelo *baseline* no experimento descrito na subseção anterior. Com base nos resultados, observa-se que o DEEP demonstrou desempenho melhor, em termos de acurácia, quando comparado com o modelo *baseline*. Para todas as emoções analisadas, a arquitetura do *DEEP* apresentou valores de acurácia com os modelos especialistas maiores

Figura 5.1: Processo de avaliação para a emoção *alegria*



do que os do modelo *baseline*. Observa-se que emoções que se encontram próximas no Modelo Circumplexo de Russel, ou seja, que possuem certo nível natural de confusão, como por exemplo tristeza e medo, foram as emoções que apresentaram menor acurácia no modelo *baseline*. No entanto, pode-se verificar que a arquitetura do DEEP obteve melhores resultados nestes casos, alcançando um acréscimo de 24.42% para a emoção Medo, e um acréscimo de 22.31% para a emoção Tristeza. Isso denota que, com a utilização dos modelos especialistas do DEEP, o distanciamento dessas emoções apresentou-se mais evidente, com a emoção Tristeza tendo um grande aumento de acurácia em comparação ao modelo *baseline*.

Adicionalmente, verifica-se um aumento geral em acurácia em todas as emoções. Baseado nestes resultados, infere-se que os modelos especialistas apresentam melhor capacidade de reconhecimento das emoções para as quais foram treinadas. A diferença média entre as acurácias dos modelos especialistas comparado com o modelo *baseline* foi de 12.39%, e as emoções com mais diferença de acurácias entre abordagens foram Medo, com 24.42% de aumento de acurácia e Tristeza, com o aumento de acurácia de 22.31%.

Em suma, com os resultados obtidos, inferimos que a utilização da abordagem tradicional apresentada na literatura relacionada com um único modelo pode acarretar a dificuldade de distinção de determinadas emoções. Nestas situações, os modelos especialistas do DEEP apresentam melhor resultado do que o da abordagem tradicional encontrada em trabalhos relacionados.

Outro detalhe de atenção é em relação ao *F1-Score*. Olhando a tabela 5.5 percebe-se que houve um aumento no *F1-Score* do DEEP em comparação ao modelo *baseline*, porém este não foi significativo. Este pequeno aumento se justifica ao olharmos para a fórmula do *F1-Score*, 2.4, no Capítulo 2. Como o *F1-Score* é uma média harmônica da precisão e da sensibilidade, caso haja um grande número de classes negativas em comparação as positivas, os valores de *F1-Score* podem ficar baixos. Como as classes positivas consideradas no cálculo, o número de registros da emoção em questão, são poucas em comparação ao número de registros de outras emoções presentes no *dataset* completo, que são as classes negativas, os cálculos de *F1-Score* ficaram baixos para ambos. Assim, considera-se que a acurácia foi a melhor métrica de avaliação para comparar as arquiteturas.

5.3 Discussão dos resultados

Nesta seção iremos iniciar uma breve discussão sobre os resultados alcançados e também apresentaremos algumas dificuldades encontradas, além das limitações gerais do projeto.

Com base nos resultados da Tabela 5.5, verificamos que a abordagem proposta de construção de modelos especialistas retornou um resultado melhor com as métricas escolhidas do que a abordagem *baseline*. Em situações em que naturalmente ocorreria uma confusão entre emoções próximas, os modelos especialistas demonstraram capacidade maior de reconhecer emoções do que o modelo *baseline*. Dessa forma, a abordagem de construção de modelos especialistas apresenta-se como uma alternativa válida para a tarefa de reconhecimento de emoções, apresentando melhor desempenho para reconhecimento de múltiplas emoções do que arquiteturas tradicionais de IA.

Em consequência dos resultados, confirmamos a principal questão de pesquisa para este trabalho: a abordagem com modelos especialistas da arquitetura do *DEEP* apresentou resultados melhores do que abordagens tradicionais com um modelo único. Consequentemente, verificamos que a abordagem de construção de modelos especialistas pode ser utilizada em tarefas de aprendizagem de máquina como uma alternativa aos modelos tradicionais apresentados na literatura relacionada.

Capítulo 6

Conclusão

Neste capítulo serão abordadas considerações finais sobre o projeto, levantando quais objetivos foram alcançados e contribuições que o projeto traz. Também serão levantadas as dificuldades encontradas e as limitações desta pesquisa. Por fim, serão elucidados os possíveis trabalhos futuros.

Por meio do objeto de estudo desta pesquisa, compreende-se que a recognição de emoção na voz é uma vertente de IA cuja tarefa baseia-se no reconhecimento e na classificação da reação emotiva de uma pessoa quanto a um estímulo [1]. Tal tarefa possui suma importância em aplicações que se pautam primordialmente na interação humano-computador, a exemplo dos *call centers*, de modo a possibilitar uma avaliação qualitativa do sucesso de uma chamada, neste caso, ao medir a expressão de determinadas emoções, como raiva, alegria e tristeza.

Neste trabalho, foi apresentado o DEEP, uma arquitetura de modelos especialistas para o reconhecimento de emoções, com base em padrões presentes no espectro sonoro gerado pela voz de falantes da língua portuguesa. A arquitetura de modelos especialistas do DEEP, construída com CNNs, é proposta de forma a contornar limitações encontradas nos trabalhos relacionados mais recentes, como a utilização de modelos híbridos, que podem propagar erros entre os modelos sequenciais colocados. Além disso, cabe ressaltar que os modelos foram treinados com base em dados de voz da língua portuguesa presentes no *database* VERBO [21], por não terem sido encontradas outras pesquisas que apresentaram modelos treinados nesta língua.

Em um primeiro momento, foi detalhada toda a estratégia adotada para a modelagem do DEEP, elucidando as *features* extraídas, a etapa de pré-processamento e a estrutura dos modelos especialistas, para, em seguida, validar a arquitetura proposta. Na etapa de validação, foi utilizada a biblioteca *hyperas* da linguagem *python*, o que possibilitou a descoberta de grupos de hiper parâmetros em configurações melhores do que as iniciais

para todos os modelos especialistas. Assim, tal resultado pode ser visualizado na Tabela 5.2 apresentada no Capítulo 5.

Por se tratar de uma base de dados nova, em que não existiam trabalhos prévios para a comparação de resultados obtidos, foi necessária, a fim de se avaliar o desempenho do DEEP, a construção de um modelo *baseline* cuja arquitetura é composta por um único modelo CNN para classificar as emoções. Após o detalhamento do modelo *baseline*, foi possível compará-lo com o DEEP, observando-se neste, assim, ganhos de performance, em termos de acurácia, para todas as 7 emoções presentes no VERBO. A diferença média entre as acurácias das arquiteturas comparadas foi de 12.39%, tendo o maior ganho com a emoção Medo, esta que foi 24.42% maior. As acurácias obtidas podem ser visualizadas na Tabela 5.5 no Capítulo 5. Tais resultados confirmam que, de fato, o DEEP distingue melhor as emoções do que um modelo único de DL.

Após alcançados os objetivos de apresentação e validação do DEEP, conclui-se que a arquitetura aqui proposta é robusta o suficiente para alcançar os ganhos de performance desejados e pode ser utilizada como uma alternativa a outras arquiteturas, como as de modelos híbridos, para as tarefas de classificação de emoção na voz. Além disso, o DEEP contribui para a literatura por ser o primeiro modelo treinado em língua portuguesa, o que propicia a sua aplicação, em diversas áreas, nos países cujo idioma oficial é a língua portuguesa.

A dificuldade mais proeminente no desenvolvimento deste projeto foi em decorrência do custo computacional dos processos desenvolvidos. A otimização dos hiper parâmetros dos modelos especialistas foi um processo difícil e bastante longo. Os treinamentos dos modelos CNN demoravam cerca de 20 segundos por época para serem executados. Considerando que os treinamos com 500 épocas e que o processo de hiper parametrização executa 10 etapas, então, a estimativa de tempo para a hiper parametrização de um único modelo CNN especialista é de 100.000 segundos, ou, aproximadamente, 28 horas. Como fizemos esse processo para cada uma das emoções analisadas, com o intuito de compor a arquitetura do *DEEP*, foram necessárias cerca de **196 horas**. Este custo de tempo para testar as iterações da arquitetura acarretou a diminuição do escopo do trabalho, o qual pretendemos explorar em trabalhos futuros.

A respeito das limitações desta pesquisa, identificamos principalmente a falta de referências de trabalhos para a comparação dos resultados. Acreditamos que a situação ideal para avaliar a arquitetura proposta seria comparando-a com algum modelo previamente desenvolvido e validado na literatura relacionada, porém, até o então momento da escrita deste trabalho, não existem outras pesquisas que utilizavam a base de dados VERBO [21] para treinamento de modelos. Essa validação pode ser refeita com algum conjunto de dados semelhante, porém requer que o processo de hiper parametrização seja

recriado, além de ser, como citado no parágrafo acima, um processo bastante custoso computacionalmente.

Trabalhos futuros a esta pesquisa envolvem, em primeira instância, o desenvolvimento de aplicações para *smartphones* que utilizem o DEEP para o reconhecimento de emoção a partir da voz captada do usuário. Essas aplicações podem ser utilizadas, por exemplo, para sugestão de músicas com base na emoção percebida do usuário. Outro ponto cabível de ser abordado no futuro é a construção de um módulo de votação para o DEEP. Com um módulo de votação, o modelo pode escolher, a partir das probabilidades apontadas por cada modelo especialista, à qual emoção um dado de entrada pertence. Por fim, cabe colocar, também, a replicação da arquitetura do DEEP para outras bases de dados de voz, validando os resultados obtidos com os de outros trabalhos estado-da-arte dessas *databases*.

Bibliografia

[1] Moataz El Ayadi, Mohamed S Kamel e Fakhri Karray. "Survey on speech emotion recognition: Features, classification schemes, and databases". Em: *Pattern Recognition* 44.3 (2011), pp. 572–587.

[2] Rosalind W Picard. *Affective computing*. 2000.

[3] Stuart Russell e Peter Norvig. "Artificial intelligence: a modern approach". Em: (2002).

[4] Amanda Purington et al. "" Alexa is my new BFF" Social Roles, User Satisfaction, and Personification of the Amazon Echo". Em: *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 2017, pp. 2853–2859.

[5] Leandro Y Mano et al. "An intelligent and generic approach for detecting human emotions: a case study with facial expressions". Em: *Soft Computing* (2019), pp. 1–13.

[6] Frank Dellaert, Thomas Polzin e Alex Waibel. "Recognizing emotion in speech". Em: *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*. Vol. 3. IEEE. 1996, pp. 1970–1973.

[7] Oh-Wook Kwon et al. "Emotion recognition by speech signals". Em: *Eighth European Conference on Speech Communication and Technology*. 2003.

[8] Klaus R Scherer. "Expression of emotion in voice and music". Em: *Journal of voice* 9.3 (1995), pp. 235–248.

[9] Rainer Banse e Klaus R Scherer. "Acoustic profiles in vocal emotion expression." Em: *Journal of personality and social psychology* 70.3 (1996), p. 614.

[10] Paul R Kleinginna e Anne M Kleinginna. "A categorized list of emotion definitions, with suggestions for a consensual definition". Em: *Motivation and emotion* 5.4 (1981), pp. 345–379.

[11] Yixiong Pan, Peipei Shen e Liping Shen. “Speech emotion recognition using support vector machine”. Em: *International Journal of Smart Home* 6.2 (2012), pp. 101–108.

[12] Kun Han, Dong Yu e Ivan Tashev. “Speech emotion recognition using deep neural network and extreme learning machine”. Em: *Fifteenth annual conference of the international speech communication association*. 2014.

[13] Lisa Feldman Barrett, Kristen A Lindquist e Maria Gendron. “Language as context for the perception of emotion”. Em: *Trends in cognitive sciences* 11.8 (2007), pp. 327–332.

[14] Haytham M Fayek, Margaret Lech e Lawrence Cavedon. “Evaluating deep learning architectures for Speech Emotion Recognition”. Em: *Neural Networks* 92 (2017), pp. 60–68.

[15] Ian Goodfellow et al. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.

[16] Ossama Abdel-Hamid et al. “Convolutional neural networks for speech recognition”. Em: *IEEE/ACM Transactions on audio, speech, and language processing* 22.10 (2014), pp. 1533–1545.

[17] Kun-Yi Huang et al. “Speech emotion recognition using deep neural network considering verbal and nonverbal speech sounds”. Em: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 5866–5870.

[18] Jianfeng Zhao, Xia Mao e Lijiang Chen. “Speech emotion recognition using deep 1D & 2D CNN LSTM networks”. Em: *Biomedical Signal Processing and Control* 47 (2019), pp. 312–323.

[19] Ngoc-Huynh Ho et al. “Multimodal Approach of Speech Emotion Recognition Using Multi-Level Multi-Head Fusion Attention-Based Recurrent Neural Network”. Em: *IEEE Access* 8 (2020), pp. 61672–61686.

[20] Soonil Kwon et al. “A CNN-assisted enhanced audio signal processing for speech emotion recognition”. Em: *Sensors* 20.1 (2020), p. 183.

[21] José Torres Neto et al. “VERBO: Voice Emotion Recognition dataBase in Portuguese language”. Em: *Journal of Computer Science* 14 (nov. de 2018), pp. 1420–1430. DOI: 10.3844/jcssp.2018.1420.1430.

[22] James A Russell. “A circumplex model of affect.” Em: *Journal of personality and social psychology* 39.6 (1980), p. 1161.

[23] David Poole, Alan Mackworth e Randy Goebel. “Computational Intelligence”. Em: (1998).

[24] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[25] Warren S McCulloch e Walter Pitts. "A logical calculus of the ideas immanent in nervous activity". Em: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[26] James Bergstra et al. "Hyperopt: a python library for model selection and hyperparameter optimization". Em: *Computational Science & Discovery* 8.1 (2015), p. 014008.

[27] Biing Hwang Juang e Laurence R Rabiner. "Hidden Markov models for speech recognition". Em: *Technometrics* 33.3 (1991), pp. 251–272.

[28] Robert V Shannon et al. "Speech recognition with primarily temporal cues". Em: *Science* 270.5234 (1995), pp. 303–304.

[29] Florian Eyben, Martin Wöllmer e Björn Schuller. "Opensmile: the munich versatile and fast open-source audio feature extractor". Em: *Proceedings of the 18th ACM international conference on Multimedia*. 2010, pp. 1459–1462.

[30] Md Sahidullah e Goutam Saha. "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition". Em: *Speech communication* 54.4 (2012), pp. 543–565.

[31] Vibha Tiwari. "MFCC and its applications in speaker recognition". Em: *International journal on emerging technologies* 1.1 (2010), pp. 19–22.

[32] Stanley Smith Stevens, John Volkmann e Edwin B Newman. "A scale for the measurement of the psychological magnitude pitch". Em: *The Journal of the Acoustical Society of America* 8.3 (1937), pp. 185–190.

[33] A Michael Noll. "Cepstrum pitch determination". Em: *The journal of the acoustical society of America* 41.2 (1967), pp. 293–309.

[34] Emir Demirel, Baris Bozkurt e Xavier Serra. "Automatic chord-scale recognition using harmonic pitch class profiles". Em: *Barbancho I, Tardón LJ, Peinado A, Barbancho AM, editors. Proceedings of the 16th Sound & Music Computing Conference; 2019 May 28-31; Málaga, Spain.[Málaga]: SMC; 2019.* Sound & Music Computing Conference. 2019.

[35] Diane Cook e Sajal Kumar Das. *Smart environments: technology, protocols, and applications*. Vol. 43. John Wiley & Sons, 2004.

[36] Zhi-Xuan Tan et al. "A multimodal LSTM for predicting listener empathic responses over time". Em: *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*. IEEE. 2019, pp. 1–4.

[37] James S Bergstra et al. “Algorithms for hyper-parameter optimization”. Em: *Advances in neural information processing systems*. 2011, pp. 2546–2554.