A Comparison of Deep Learning Architectures for Automatic Gender Recognition from Audio Signals

Alef Iury S. Ferreira[1], **Frederico S. Oliveira**[2], **Nádia F. Felipe da Silva**[1], **Anderson S. Soares**[1]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
Goiânia – Goiás – Brasil

[2]Câmpus Várzea Grande – Universidade Federal de Mato Grosso (UFMT)
Cuiabá – Mato Grosso – Brasil

alefiury14@gmail.com, fredoliveira@ufmt.br, nadia@inf.ufg.br

anderson@inf.ufg.br

Abstract. *Automatic gender recognition from speech is a problem related to the area of speech analysis and has a variety of applications thats extends from the personalisation of product recommendation to forensics. Identifying the efficiency and costs of different approaches that deal with this problem is imperative. This work aims to investigate and compare the efficiency and costs of different deep learning architectures in the task of gender recognition from speech. The results show that the one-dimensional convolutional model achieves the best results. However, experiments conducted demonstrate that the fully connected model has similar results, using less memory and trained in much less time compared to the one-dimensional convolutional model.*

Resumo. *O reconhecimento de gênero a partir da fala é um problema relacionado à análise de fala humana, e possui diversas aplicações que vão desde a personalização na recomendação de produtos à ciência forense. A identificação da eficiência e custos de diferentes abordagens que lidam com esse problema é imprescindível. Este trabalho tem como foco investigar e comparar a eficiência e custos de diferentes arquiteturas de deep learning para o reconhecimento de gênero a partir da fala. Os resultados mostram que o modelo convolucional unidimensional consegue os melhores resultados. No entanto, constatou-se que o modelo fully connected apresentou resultados próximos com menor custo, tanto no uso de memória, quanto no tempo de treinamento.*

1. Introdução

O Reconhecimento Automático de Gênero (*Automatic Gender Recognition - AGR*) refere-se à identificação de gênero de um locutor a partir de um sinal de áudio [Kabil et al. 2018]. Sistemas de AGR são utilizados, por exemplo, para a melhora da recomendação de produtos em estratégias de marketing [Shepstone et al. 2013]; análise de voz em cenários de investigação de crimes [Nair and Savithri 2021]; melhora de sistemas interação humano-computador [La Mura and Lamberti 2020] e no auxílio de reconhecimento automático de voz [Parveen and Green 2003].

Algumas das estratégias mais populares para a resolução de tal tarefa incluem: *Suport Vector Machines* (SVM) [Bocklet et al. 2008, Nair and Vijayan 2019], *Random Forest* [Levitan et al. 2016, Nair and Vijayan 2019], Regressão Linear [Levitan et al. 2016], Regressão Logística [Levitan et al. 2016] e Redes Neurais Profundas (em inglês *Deep Neural Networks* - DNNs) [Alkhawaldeh 2019, Kabil et al. 2018]. Este último, tem substituído algoritmos clássicos de Aprendizado de Máquina (em inglês *Machine Learning*) em diversas áreas, devido à sua rápida evolução e aos seus resultados, que têm se mostrado melhores que seus antecessores [Pouyanfar et al. 2018].

Este trabalho tem como foco investigar e comparar o desempenho de três algoritmos baseados em DNNs para a realização da tarefa de classificação de gênero através da fala, apontando seus custos relacionados ao uso de memória e tempo de treinamento.

As arquiteturas escolhidas para os experimentos foram: *fully connected*, Redes Neurais Convolucionais Unidimensional e Redes Neurais Convolucionais Bidimensional. Essas arquiteturas foram selecionadas com base em seus resultados, que são considerados o estado da arte em aplicações de análise de áudio [Purwins et al. 2019]. Os experimentos conduzidos evidenciam que o modelo que obteve os melhores resultados em termos de acurácia, *recall*, *precision* e *f1-score* foi o modelo treinado com a arquitetura convolucional unidimensional, utilizando comprimentos de áudio de 3 segundos. No entanto, constatou-se que o modelo treinado em uma arquitetura *fully connected*, utilizando como entrada a concatenação das representações dadas por mel espectrograma, *Wav2Vec* e coeficiente de frequência mel-cepstral, apresenta resultados próximos ao modelo convolucional unidimensional, nos índices analisados, com menor custo, tanto no uso de memória, quanto no tempo de treinamento.

Para realizar a avaliação dos modelos, considerou-se diferentes durações de tempo para os arquivos de áudio, diferentes características (*features*)[1] de entrada e hiperparâmetros. Realiza-se uma análise do impacto que a quantidade de tempo utilizada em cada arquivo de áudio tem no desempenho de cada modelo, assim como as diferenças de uso de memória e tempo de treinamento entre os modelos testados para a realização da tarefa proposta.

Este trabalho está organizado da seguinte forma: A seção 2 introduz cada uma das *features* utilizadas nos testes do modelo *fully connected* e uma breve introdução a cada uma delas; a seção 3 apresenta e compara brevemente trabalhos relacionados; a seção 4 detalha os experimentos realizados em cada uma das arquiteturas propostas; a seção 5 apresenta os resultados obtidos e na seção 6 conclui-se o trabalho.

2. Extração de *Features*

Sistemas que realizam análises ou decisões utilizando sinais de áudio necessitam aproveitar ao máximo as características inerentes deste tipo de sinal. Essa tarefa pode ser realizada por meio da extração de *features*, que por sua vez, proveem uma representação compacta e descritiva do sinal de áudio, destacando as características que são mais pertinentes para a resolução da tarefa proposta. Usualmente, para a extração de *features* de áudio, é comum a utilização de representações no domínio da frequência [Sharma et al. 2020]. A seguir, um resumo das principais *features* referentes à áudio utilizadas neste trabalho, as

[1]Neste trabalho, as palavras ”*features*”e ”características”são utilizadas indistintamente.

quais, com exceção da Frequência Fundamental e *Wav2vec*, foram extraídas utilizando a biblioteca librosa [McFee et al. 2015].

2.1. Espectrograma

Um espectrograma é uma representação visual do espectro de frequências de um sinal, variando com o tempo. Uma das formas de transformar o sinal de áudio em um espectrograma é aplicando *Short-Term Fourier Transform* - STFT, a qual, é uma transformação que decompõe um sinal, geralmente uma função no domínio do tempo, em suas frequências constituintes [Kanatani 2018].

2.1.1. Mel Espectrograma

O Mel Espectrograma é um espectrograma representado na Escala de Mel. A Escala de Mel é o resultado de uma transformação em escala logarítmica que tem por finalidade manter os tons de frequência equidistantes para a audição humana [Picone 1993].

A escala mel é definida da seguinte forma:

$$mel = 2595 * \log 10(1 + \frac{heartz}{700})$$

2.1.2. Coeficiente de Frequência Mel-Cepstral

O Coeficiente de Frequência Mel-Cepstral (em inglês *Mel Frequency Cepstral Coefficients* - MFCC) é um mel espectrograma que passa, posteriormente, por uma transformação discreta de cosseno [Sharma et al. 2020]. A diferença entre o Cepstral e o Cepstral de frequência de mel é que no MFCC as bandas de frequência são igualmente espaçadas na escala mel, que emula o sistema auditivo humano [Sharma et al. 2020], diferentemente do que ocorre no Cepstral, em que o espaçamento é linear.

2.2. Frequência Fundamental

Frequência Fundamental (em inglês *Fundamental Frequency* - F0) é a frequência de vibração dos ligamentos ao pronunciar sons sonoros, não considerando sussurros ou assobios. Ela está diretamente ligado com o *pitch*, que é a entonação da voz. Para este trabalho, considerou-se três algoritmos para a estimação do F0: *Rapt* [Yamamoto et al. 2020], *Yaapt* [Cheveigné and Kawahara 2002] e *Praat* [Boersma and Weenink 2018].

2.3. Contraste Espectral

O Contraste Espectral (em inglês *Spectral Contrast* - SC) considera a diferença entre o pico espectral e o vale espectral em cada sub-banda de frequência para que seja possível determinar as *features* espectrais relativas. Essas *features* representam a distribuição de componentes harmônicos e não harmônicos [Jiang et al. 2002].

2.4. Cromagrama

O Cromagrama, também chamado Chroma-STFT, estima quanta energia possui cada uma da sequência de todos os 12 semitons, que correspondem às notas Dó, Dó#, Ré, Ré#,

Mi, Fá, Fá#, Sol, Sol#, Lá, Lá# e Si. A partir dessa estimativa é possível capturar características harmônicas e melódicas em um sinal de áudio. O *Chorma-STFT* é obtido aplicando-se STFT [Ellis 2007].

2.5. *Features* de Centróide Tonal

Features de Centróide Tonal (em inglês *Tonal Centroid Features* - Tonnetz), detecta as mudanças no conteúdo da harmônica de sinais de áudios musicais, utilizando o método em [Harte et al. 2006].

2.6. *Wav2vec Features*

Wav2vec [Schneider et al. 2019] é um *framework* que tem por finalidade obter representações gerais, a partir de sinais de áudios em seu estado bruto, utilizando os benefícios do pré-treinamento e o regime de treinamento *self-supervised*. Foi desenvolvido com a finalidade de ser utilizado como um extrator de *features* em cenários em que há pouca disponibilidade de dados anotados, melhorando a performance de modelos de reconhecimento de fala, principalmente, em tarefas em que a aquisição de dados anotados é custosa. Para a extração de *features*, utilizou-se a versão *wav2vec large* pré-treinada.

3. Trabalhos Relacionados

Em meados da década de 1990 surgiram os primeiros trabalhos que utilizavam *features* para realizar a classificação de gênero de forma automática a partir da fala [Levitan et al. 2016]. Desde então, têm surgido inúmeros trabalhos que buscam estabelecer técnicas cada vez melhores de extração de *features* e algoritmos para AGR.

Em [Bocklet et al. 2008] é realizada uma comparação entre duas abordagens distintas para a predição de 7 classes que representam diferentes faixas etárias e diferentes gêneros, a partir de sinais de áudio. A primeira abordagem utiliza uma combinação entre um Modelo de Misturas de Gaussianas (em inglês *Gaussian Mixture Models* - GMM) e Modelos Universais (em inglês *Universal Background Models*). Na segunda abordagem, uma Máquina de Vetores de Suporte (em inglês *Support Vectors Machine* - SVM) é treinada com super vetores, os quais, são o resultado da concatenação da média das saídas de um modelo do tipo GMM, que é treinado a partir de cada locutor do subconjunto de dados de teste e treinamento, sendo que cada super vetor é rotulado com uma das 7 classes. Em seus experimentos, concluíram que a abordagem que utiliza o modelo SVM se saiu melhor em todos os experimentos realizados.

Em [Levitan et al. 2016] é realizada uma comparação entre F0 e MFCC, enquanto *features* usadas na representação de entrada com os seguintes modelos: Regressão Logística, Regressão Linear, *Random Forest*, *AdaBoost* e *LLAMA* (classificador categórico) para realizar a classificação de gênero de forma automática a partir da fala. Além disso, é feita uma investigação do impacto nos resultados com a utilização de áudios de comprimentos que variam de 0.5 a 2 segundos, e o impacto de se ter uma base de dados monolíngue ou multilíngue, assim como a adição de mais uma classe para ser classificada: a fala de crianças. Concluíram que o modelo de Regressão Logística utilizando dois segundos de áudio e a combinação das duas *features* propostas em uma base de dados multilíngue, conseguiu a melhor acurácia na classificação de gênero masculino e feminino. Nos testes realizados com a classe de crianças, o modelo que obteve o melhor

resultado foi o *Random Forest* utilizando 2 segundos de áudio e a combinação de ambas as *features*.

Em [Kabil et al. 2018] é realizada uma comparação entre variações de modelos de convolução unidimensional, que recebem como entrada áudios em seu estado bruto e dois modelos *fully connected*, cujas entradas eram MFCC e F0, respectivamente. Em seus experimentos, concluíram que o modelo convolucional obteve os melhores resultados.

Em [Mamyrbayev et al. 2020] os autores atém-se a comparar modelos de redes neurais convolucionais bidimensionais e modelos *fully connected*, ambos com diferentes hiper-parâmetros, tendo como entrada o MFCC, a qual foi normalizada utilizando *z-score*, e em seguida, sendo transformada em uma matriz de Gram. Em seus experimentos concluíram que o modelo *fully connected* obteve uma melhor generalização e melhor avaliação nos índices de *recall*, *precision* e *F1 score* para ambos os gêneros.

Em [Bocklet et al. 2008] e [Kabil et al. 2018] são utilizados algoritmos clássicos de Aprendizado de Máquina para a classificação de gênero a partir da fala. Entretanto, tais trabalhos já não são representantes da maioria das pesquisas realizadas atualmente na área de análise de áudio [Pouyanfar et al. 2018]. Em [Kabil et al. 2018] e [Mamyrbayev et al. 2020] são apresentados algoritmos baseados em DNNs, no entanto, há uma limitação quanto à utilização de extratores de *features*, os quais, se limitaram somente ao MFCC e o F0. Além disso, nenhum dos trabalhos citados realiza uma análise dos custos envolvidos na utilização de tais algoritmos para a solução do problema de reconhecimento de gênero a partir da fala.

Portanto, este trabalho tem o intuito de preencher tais lacunas, apresentando uma análise da eficiência de arquiteturas de DNNs na resolução da tarefa proposta, em cenários que consideram a utilização de diferentes *features* em áudios de diferentes comprimentos, evidenciando os custos relacionados à utilização de memória e tempo de treinamento daqueles modelos que apresentam os melhores resultados em termos de acurácia, *recall*, *precision* e *f1-score*.

4. Experimentos

Os experimentos foram executados com base nos treinamentos realizados com as arquiteturas *Fully Connected* (FC), Redes Neurais Convolucionais Unidimensional (CNN1D) e Redes Neurais Convolucionais Bidimensional (CNN2D). Tais experimentos foram executados em uma placa de vídeo GTX 1660 com 6GB de memória disponível, utilizando o *framework* Pytorch. A fim de permitir a replicação dos experimentos, o código-fonte referente aos modelos pode ser acessado em: `https://github.com/alefiury/Automatic-Gender-Classification`.

4.1. Base de Dados

Os treinamentos foram efetuados com os dados da base de dados *Librispeech* [Panayotov et al. 2015], utilizando-se 100 horas de áudios que foram gravados em um ambiente sem ruídos, a partir da versão *train-clean-100*. Cada arquivo de áudio passou por um pré-processamento, no qual, apenas os R, com $R = \{1, 2, 3\}$, primeiros segundos foram considerados. Os arquivos de áudio foram convertidos para 16khz, PCM 16 bits.

Por fim, a base de dados foi dividida em três subconjuntos: treinamento, validação e teste; sendo que, 80% dos dados foram utilizados para treinamento, 10% para validação

e 10% para teste. Tal divisão foi realizada de forma estratificada, mantendo uma proporção equilibrada das classes em cada um dos subconjuntos de dados.

4.2. Modelo *Fully Connected* (FC)

Os experimentos focaram em realizar uma comparação entre as *features* mencionadas na seção 2, considerando diferentes tamanhos de áudio. As *features* que obtiveram os melhores resultados individuais foram combinadas para serem utilizadas como entrada para o modelo FC final (Figura 1). Esse modelo foi obtido utilizando-se a configuração dos hiper-parâmetros como detalhado na seção 4.2.2.

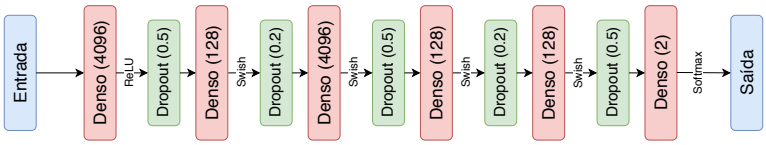

Figura 1. Modelo final encontrado através da otimização hiper-paramétrica.

4.2.1. Seleção de *Features*

As seguintes *features* foram extraídas dos áudios: Mel Spectrograma, MFCC, F0, SC, Cromagrama, Tonnetz e Wav2Vec. Tais *features* foram utilizadas como entrada em um modelo preliminar (Figura 2), com 4 camadas densas, a fim de atestar suas relevâncias para a tarefa proposta.

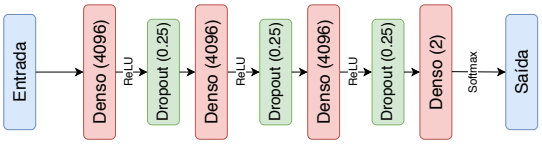

Figura 2. Arquitetura do modelo FC preliminar.

O critério de escolha de um subconjunto de *features* foi baseado no desempenho do modelo preliminar considerando os seguintes índices para cada gênero: Acurácia, *recall*, *precision* e *f1-score*. As três *features* com os melhores resultados foram selecionadas, combinadas e alimentadas à uma rede mais complexa (Figura 1).

Na Tabela 1 pode-se verificar os resultados individuais de cada *feature* considerando-se diferentes comprimentos de áudio, que variam de 1 a 3 segundos, utilizando o modelo FC preliminar. Os resultados exibem os índices de Acurácia de ambos os gêneros, e valores de *recall*, *precision* e *f1-score* para cada um dos gêneros, separadamente. Os melhores resultados foram obtidos, para cada *feature*, utilizando-se comprimento de 3 segundos, com exceção do *Rapt* e *Praat* os quais obtiveram melhores resultados com 2 e 1 segundos, respectivamente. Além disso, as *features* que obtiveram os melhores resultados em todos os índices analisados foram: MFCC, Wav2Vec e Mel Espectrograma, com 3 segundos de comprimento de áudio.

4.2.2. Otimização Hiper-paramétrica

Hiper-parâmetros são parâmetros cujos valores são escolhidos antes do treinamento. Tais valores dependem, muitas vezes, de resultados experimentais. Assim, uma abordagem

Tabela 1. Detalhamento do desempenho de cada *feature* utilizando o modelo FC preliminar. Utilizando os dados do subconjunto de teste.

		Male			Female		
Feature	**Acurácia**	**Recall**	**Precision**	**F1**	**Recall**	**Precision**	**F1**
MFCC (1 seg)	96.6%	95.9	97.3	96.6	97.4	96.0	96.7
MFCC (2 seg)	97.8%	97.3	98.3	97.8	98.3	97.3	97.8
MFCC (3 seg)	**98.4%**	**98.3**	**98.5**	**98.4**	**98.5**	**98.3**	**98.4**
Yappt (1 seg)	89.7%	88.2	91.0	89.6	91.3	88.6	89.9
Yappt (2 seg)	91.6%	89.6	93.5	91.5	93.7	89.9	91.8
Yappt (3 seg)	92.2%	91.2	92.8	92.0	93.1	91.7	92.4
Wav2Vec (1 seg)	96.1%	95.7	96.4	96.1	96.5	95.7	96.1
Wav2Vec (2 seg)	98.2%	98.3	98.2	98.2	98.2	98.3	98.2
Wav2Vec (3 seg)	**98.6%**	**98.6**	**98.5**	**98.6**	**98.6**	**98.6**	**98.6**
Mel Espectrograma (1 seg)	92.4%	92.5	92.1	92.3	92.3	92.6	92.4
Mel Espectrograma (2 seg)	93.0%	92.7	93.2	93.0	93.3	97.3	93.1
Mel Espectrograma (3 seg)	**95.1%**	**94.0**	**96.1**	**95.0**	**96.2**	**94.1**	**95.2**
Rapt (1 seg)	87.3%	86.8	87.8	87.3	87.8	86.8	87.3
Rapt (2 seg)	88.4%	87.8	88.1	87.9	89.0	88.7	88.9
Rapt (3 seg)	87.0%	86.1	86.8	86.5	87.8	87.2	87.5
Praat (1 seg)	85.5%	82.3	88.4	85.2	88.8	89.9	85.7
Praat (2 seg)	83.0%	81.5	83.5	82.5	84.5	82.6	83.5
Praat (3 seg)	82.3%	84.9	80.8	82.8	79.8	84.1	81.9
SC (1 seg)	76.3%	75.0	73.1	77.3	71.0	80.3	75.4
SC (2 seg)	80.1%	83.5	77.7	80.5	76.8	82.8	79.7
SC (3 seg)	81.6%	82.4	81.2	81.8	80.8	82.0	81.4
Cromagrama (1 seg)	71.7%	75.0	70.7	72.8	68.2	72.8	70.4
Cromagrama (2 seg)	75.9%	76.7	74.8	75.8	75.1	76.9	76.0
Cromagrama (3 seg)	76.9%	78.4	75.7	77.0	75.4	78.1	76.7
Tonnetz (1 seg)	57.4%	56.8	57.4	57.1	58.0	57.5	57.7
Tonnetz (2 seg)	58.2%	60.0	87.6	58.9	86.1	58.8	57.5
Tonnetz (3 seg)	60.3%	66.8	89.3	62.8	53.6	61.5	57.3

preferível para determinar um subconjunto ótimo de hiper-parâmetros é explorar diferentes combinações e avaliar seus impactos em modelos distintos. Frequentemente, tais explorações exigem tempo, recursos e um senso afinado resultado de experiência e intuição do indivíduo realizando os experimentos. Consequentemente, em cenários nos quais tempo e recursos são escassos, é preferível a utilização de estratégias sistemáticas para a escolha de hiper-parâmetros, utilizando uma abordagem de busca automática [Bergstra and Bengio 2012, Wu et al. 2019].

A escolha do método de otimização hiper-paramétrica levou em consideração a estratégia que realizaria a escolha de um subconjunto ótimo ou próximo de ótimo com o menor custo de memória e tempo possível. A abordagem que mais se adequou a tais parâmetros de escolha foi o *Bayesian Optimization* [Betrò 1991]. Na Tabela 2 pode-se verificar o subespaço de busca para a quantidade de neurônios ocultos, camadas ocultas e

a taxa de *dropout*, em que a coluna Valor Mínimo representa o menor valor a ser testado, que é iterado com os valores contidos na coluna Iteração, até que se chegue em um valor máximo, representado por Valor Máximo. A Tabela 3 apresenta o subespaço de busca para as funções de ativação e optimizadores, em que as possíveis escolhas são categóricos.

Tabela 2. Tabela de hiper-parâmetros numéricos utilizados na otimização hiper-paramétrica.

Hiper-parâmetro	Valor Mínimo	Valor Máximo	Iteração
Neurônios Ocultos	128	4028	32
Camadas Ocultas	2	5	1
Taxa de *Dropout*	0.2	0.5	0.05

Tabela 3. Tabela de hiper-parâmetros categóricos utilizados na otimização hiper-paramétrica.

Hiper-parâmetro	Categorias
Funções de Ativação	Relu, Swish, Mish, Tanh
Optimizadores	Adam, Nadam, SGD, Adadelta

A busca no espaço de possíveis configurações de hiper-parâmetros foi baseado na acurácia dos dados de validação, cada subconjunto foi testado duas vezes, e o algoritmo de otimização foi iterado 100 vezes.

4.2.3. Treinamento

As *features* de áudio escolhidas para a realização dos experimentos, assim como detalhado na seção 4.2.1, foram: Mel Espectrograma, MFCC e Wav2vec. Houve a concatenação da média da saída de cada método, os quais geraram: 128, 40 e 512 *features*, respectivamente. Produzindo assim, um *array* unidimensional com 680 *features*. Tal redução de dimensionalidade foi realizada com o intuito de acelerar o treinamento e reduzir a utilização de memória. Por fim, a *array* unidimensional resultante foi utilizada como entrada para o modelo FC (Figura 1).

Nesta arquitetura, utilizou-se 7 camadas *fully connected*, em que cada camada possui as seguintes quantidades de neurônios: 4096, 128, 4096, 128, 128 e 2, respectivamente. Cada camada densa é seguida de uma função de ativação do tipo *Swish* [Ramachandran et al. 2017], com exceção da primeira, cuja função de ativação é do tipo *ReLU* e da camada de saída, cuja função de ativação é do tipo *softmax*. Logo após a função de ativação há uma camada de *dropout* cujas probabilidades se alternam, de uma camada para outra, entre 0.5 e 0.2.

Durante o treinamento, utilizou-se um *mini batch* de tamanho 4096 com o otimizador Adam com $\beta_1 = 0.9$, $\beta_2 = 0.999$ e $\varepsilon = 1^{e-07}$. A função *Categorical Cross Entropy* [Murphy 2012] foi utilizada como função de *loss*. Os treinamentos foram realizados considerando $R = 1$, $R = 2$ e $R = 3$. Utilizou-se *early stopping* com *patience* $= 25$.

4.3. Modelo CNN1D

Utilizou-se 5 blocos convolucionais, começando com 64 filtros, que são repetidos a cada camada subsequente, *stride* $S = 1$ e *kernel* $K = 5$. Cada bloco convolucional é seguido

de uma função de ativação do tipo *ReLU* e como estratégia de *downsampling* uma camada de *max pooling* com *batch normalization*. A função *Categorical Cross Entropy* foi utilizada como função de *loss*.

Para o treinamento do modelo CNN1D (Figura 3) utilizou-se sinais de áudio em seu estado bruto (em inglês *raw audios*) em *mini batch* de tamanho 16 com o otimizador Adam com $\beta_1 = 0.9$, $\beta_2 = 0.999$ e $\varepsilon = 1^{e-07}$. Os treinamentos foram realizados considerando $R = 1$, $R = 2$ e $R = 3$. Utilizou-se *early stopping* com *patience* $= 25$.

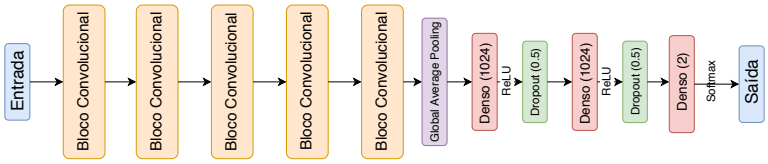

Figura 3. Arquitetura do modelo de Redes Neurais Convolucionais Unidimensional.

4.4. Modelo CNN2D

Utilizou-se 4 blocos convolucionais, começando com 32 filtros, que são dobrados a cada camadas subsequente, *stride* $S = 1$ e *kernel* $K = 3$. Cada bloco convolucional é seguido de uma função de ativação do tipo *ReLU* e como estratégia de *downsampling* uma camada de *max pooling* com *batch normalization*. A função *Categorical Cross Entropy* foi utilizada como função de *loss*. Utilizou-se *mini batch* de tamanho 32 com o otimizador Adam com $\beta_1 = 0.9$, $\beta_2 = 0.999$ e $\varepsilon = 1^{e-07}$. O modelo CNN2D (Figura 4) recebeu como entrada espectrogramas na Escala de Mel.

Os treinamentos foram realizados considerando $R = 1$, $R = 2$ e $R = 3$. Utilizou-se *early stopping* com *patience* $= 25$.

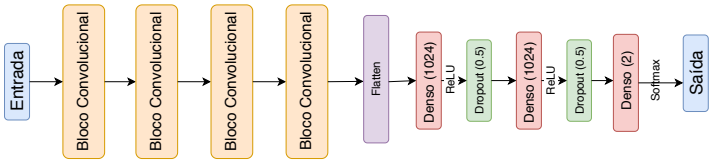

Figura 4. Arquitetura do modelo de Redes Neurais Convolucionais Bidimensional.

5. Resultados

Na Figura 5 pode-se verificar os resultados dos três modelos considerando diferentes comprimentos de amostras de áudio. Constata-se que os resultados do modelo CNN1D se sobressaem considerando os índices de acurácia, *recall*, *precision* e *F1 score* para ambos os gêneros. As diferenças nos resultados de tais índices utilizando diferentes comprimentos das amostras de áudio, em todos os três modelos, permite afirmar que existe uma melhora substancial de desempenho à medida que o tempo de áudio sendo passado como entrada também aumenta. Ao realizar uma comparação de resultados entre o modelo FC e o modelo CNN2D, verifica-se que o modelo FC conseguiu beneficiar-se melhor à medida que

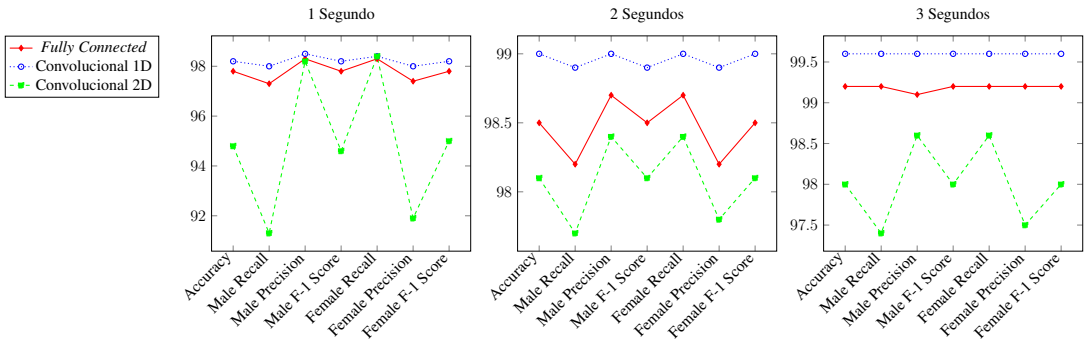

Figura 5. Resultados por modelo e tamanho de áudios.

o comprimento das amostras de áudio aumentavam, apresentando uma ampla margem de vantagem com 3 segundos de áudio.

Na Figura 6 tem-se o uso de memória em GPU e CPU, assim como a quantidade de tempo, em segundos, que cada modelo levou para treinar. Analisando essa figura pode-se constatar que o modelo CNN1D aumenta o seu tempo de treinamento, uso de memória de GPU e uso de memória RAM consideravelmente à medida que o comprimento de tempo das amostras de áudios alimentadas à rede também aumenta. Esse aumento de uso considerável de memória de GPU e RAM não é observado com tamanha acentuação nos modelos FC e CNN2D.

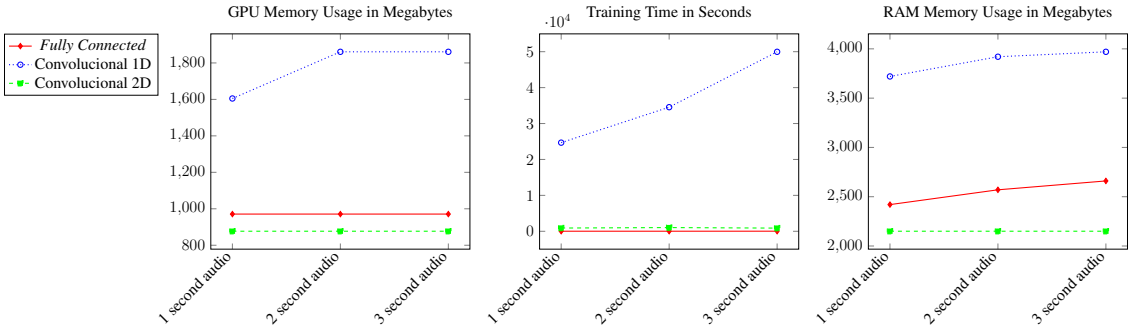

Figura 6. Gráficos de custo de cada modelo, considerando o tempo de áudio dado como entrada.

6. Conclusão

Neste trabalho, foram realizados experimentos a fim de encontrar as melhores *features* para classificação de gênero utilizando sinal da voz. Para isso, foram testadas três arquiteturas de redes neurais: FC, CNN1D e CNN2D. Também foram realizados experimentos, considerando diferentes *features* e hiper-parâmetros, para encontrar a melhor combinação dentre os subespaços analisados.

Outro aspecto relevante, também analisado nesse trabalho, refere-se ao tamanho da amostra a ser analisada. Foram realizados experimentos com amostras de 1, 2 e 3 segundos a fim de descobrir o tamanho ideal.

A partir dos resultados dos experimentos observa-se que o tempo de áudio alimentado às redes teve substancial contribuição para os resultados. Além disso, pode-se concluir que o modelo CNN1D utilizando áudios de 3 segundos de comprimento obteve

o melhor resultado nos índices de acurácia, *recall*, *precision* e *F1 score* para ambos os gêneros; seguido pelo modelo FC e CNN2D. Assim, o modelo CNN1D se coloca como ótimo candidato em aplicações que necessitam da mais alta precisão, que não se preocupam, ou não estão restritas, quanto ao custo relacionado ao uso de memória e poder de processamento.

Constatou-se, também, a pertinência do modelo FC na resolução da tarefa de classificação de gênero através da análise da fala. Mesmo obtendo um desempenho inferior ao modelo CNN1D, o modelo FC apresentou resultados próximos, com menor custo, tanto no uso de memória, quanto no tempo de treinamento. Assim, o modelo FC se coloca como um ótimo candidato em aplicações que não necessitem de alta precisão e estão restritas quanto ao uso de memória e poder de processamento, evidenciando o seu ótimo custo-benefício.

Como trabalhos futuros, recomenda-se uma análise mais profunda em relação ao tamanho das amostras de áudio sendo alimentadas às redes em seu desempenho, encontrando um intervalo de tempo que maximize-o. Além disso, a análise de tais modelos em áudios que emulam ambientes reais (áudios com adição de ruídos) também pode ser oportuno. Adicionalmente, o processo de escolha das *features* é baseada na hipótese de que elas são independentes. Entretanto, a combinação de dois conjuntos de *features* com desempenho ruim poderia superar outras *features*. Por fim, uma comparação entre os resultados apresentados e trabalhos correlatos se faz oportuna.

7. Agradecimentos

Agradecemos ao CEIA (Centro de Excelência em Inteligência Artificial) e ao Grupo CyberLabs pelo apoio financeiro proporcionado para o desenvolvimento deste trabalho.

Referências

Alkhawaldeh, R. S. (2019). Dgr: Gender recognition of human speech using one-dimensional conventional neural network. *Scientific Programming*, 2019:7213717.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(null):281–305.

Betrò, B. (1991). Bayesian methods in global optimization. *Journal of Global Optimization*, 1(1):1–14.

Bocklet, T., Maier, A., Bauer, J., Burkhardt, F., and Nöth, E. (2008). Age and gender recognition for telephone applications based on gmm supervectors and support vector machines. *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1605–1608.

Boersma, P. and Weenink, D. (2018). Praat: doing phonetics by computer [Computer program]. Version 6.0.37, retrieved 3 February 2018 `http://www.praat.org/`.

Cheveigné, A. and Kawahara, H. (2002). Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111:1917–30.

Ellis, D. P. (2007). Chroma feature analysis and synthesis.

Harte, C., Sandler, M., and Gasser, M. (2006). Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, AMCMM '06, page 21–26, New York, NY, USA. Association for Computing Machinery.

Jiang, D.-N., Lu, L., Zhang, H., Tao, J., and Cai, L. (2002). Music type classification by spectral contrast feature. *Proceedings. IEEE International Conference on Multimedia and Expo*, 1:113–116 vol.1.

Kabil, S., Muckenhirn, H., and Magimai-Doss, M. (2018). On learning to identify genders from raw speech signal using cnns. pages 287–291.

Kanatani, K.-i. (2018). Fast fourier transform. In *Particle characterization in technology*, pages 31–50. CRC Press.

La Mura, M. and Lamberti, P. (2020). Human-machine interaction personalization: a review on gender and emotion recognition through speech analysis. In *2020 IEEE International Workshop on Metrology for Industry 4.0 IoT*, pages 319–323.

Levitan, S., Mishra, T., and Bangalore, S. (2016). Automatic identification of gender from speech. pages 84–88.

Mamyrbayev, O., Toleu, A., Tolegen, G., and Mekebayev, N. (2020). Neural architectures for gender detection and speaker identification. *Cogent Engineering*, 7(1).

McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. (2015). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.

Nair, A. M. S. U. and Savithri, S. P. (2021). Classification of pitch and gender of speakers for forensic speaker recognition from disguised voices using novel features learned by deep convolutional neural networks. *Traitement du Signal*, 38(1):221–230.

Nair, R. R. and Vijayan, B. (2019). Voice based gender recognition. *International Research Journal of Engineering and Technology*, 6.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Parveen, S. and Green, P. (2003). Multitask learning in connectionist robust asr using recurrent neural networks. In *INTERSPEECH*.

Picone, J. (1993). Signal modeling techniques in speech recognition. *Proceedings of the IEEE*, 81(9):1215–1247.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., and Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Comput. Surv.*, 51(5).

Purwins, H., Li, B., Virtanen, T., Schlüter, J., Chang, S.-Y., and Sainath, T. (2019). Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219.

Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions.

Schneider, S., Baevski, A., Collobert, R., and Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. *CoRR*, abs/1904.05862.

Sharma, G., Umapathy, K., and Krishnan, S. (2020). Trends in audio signal feature extraction methods. *Applied Acoustics*, 158:107020.

Shepstone, S. E., Tan, Z.-H., and Jensen, S. H. (2013). Audio-based age and gender identification to enhance the recommendation of tv content. *IEEE Transactions on Consumer Electronics*, 59(3):721–729.

Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., and Deng, S.-H. (2019). Hyperparameter optimization for machine learning models based on bayesian optimizationb. *Journal of Electronic Science and Technology*, 17(1):26 – 40.

Yamamoto, R., Santos, J. F., and Blaauw, M. (2020). r9y9/pysptk: v0.1.18 release. Zenodo.