Gabriel Louzada Malfatti

Estudo, Desenvolvimento e Implementação de uma técnica de síntese de fala para língua portuguesa

Orientador: Prof. Dr. Kenji Nose Filho

Santo André - SP

2021

Gabriel Louzada Malfatti

Estudo, Desenvolvimento e Implementação de uma técnica de síntese de fala para língua portuguesa

Monografia apresentada ao programa de Graduação em Engenharia de Informação da Universidade Federal do ABC (UFABC), como requisito parcial à obtenção do título de Bacharel em Engenharia de Informação.

Orientador: Prof. Dr. Kenji Nose Filho

Santo André - SP

2021

Agradecimentos

Este trabalho só foi possível graças a todo apoio dado por amigos e familiares, que me ajudaram de forma direta e indireta ao longo de minha graduação.

Gostaria de agradecer todos meus colegas de IBM Research, onde estagiei por dois anos, e pude aprender diversos conceitos que puderam ser aplicados neste trabalho, em especial Ana Paula Appel e Renato Cunha, que foram importantes pilares nessa minha jornada. Deixo meus agradecimentos a todos os professores da Universidade Federal do ABC que fizeram parte de minha graduação. Gostaria de agradecer meus colegas de graduação Diego Senese, Gabriel Amaral, Marcos Romero e Thiago Silva pelas discussões produtivas e companheirismo ao longo do curso.

Quero deixar meus agradecimentos especiais para minha mãe, Silvia Helena Louzada, e minha irmã, Maria Gabriela Malfatti, fontes de apoio e inspiração para toda minha jornada. E dedico este trabalho a meu pai, David Justo Malfatti.

Por fim, deixo meus agradecimentos para meu orientador Kenji Nose Filho, um grande companheiro que, neste trabalho, acrescentou muito com seu conhecimento através de diversas conversas e suas opiniões sempre pertinentes ao longo deste trabalho de graduação.

Resumo

No presente trabalho foi realizado um estudo de recentes técnicas de síntese de fala a partir de um texto (TTS, do inglês *text-to-speech)*, por meio de técnicas orientadas a dados e modelos baseados em redes neurais. Em específico, foram escolhidas duas arquiteturas para elaboração de um sistema TTS para a língua portuguesa: O processamento de texto realizado pelo Tacotron 2, onde é gerado um espectrograma da escala mel a partir de um texto, e um modelo de inversão de espectrograma baseado em uma Rede Generativa Adversarial (GAN), o MelGAN. Os resultados obtidos mostraram que o modelo de processamento de texto pré-treinado na língua inglesa foi capaz de rapidamente se adaptar a sentenças em português após poucas iterações de treinamento numa base em português. Já o modelo utilizado para inversão de espectrograma, treinado do zero, pois não havia sua versão pré-trenada disponível, em 1250 épocas, não foi capaz de gerar áudios com naturalidade e inteligibilidade. Acredita-se que o modelo deveria ser treinado por mais épocas para que se alcançasse bons resultados. Os experimentos realizados ao longo deste trabalho mostraram as dificuldades a cerca da construção de um sistema TTS para língua portuguesa e vê o uso de modelos pré-treinados como uma boa alternativa para projetos com recursos limitados.

Palavras chave: *Text-to-speech*, Síntese de Fala, Modelos generativos, Redes Neurais Profundas.

Abstract

In the present work, we presented a study of recent text-to-speech (TTS) techniques, using data-driven techniques and neural networks based models. Specifically, two architectures were chosen for the TTS system development for the Brazilian Portuguese language: Text processing performed by Tacotron 2, where a mel scale spectrogram is generated from text and a Spectrogram Inversion model based on a Generative Adversarial Network (GAN), the MelGAN. The results showed that the text processing model pre-trained in English was able to quickly adapt to sentences in Brazilian Portuguese after few training iterations using a Portuguese database. The model used for spectrogram inversion, trained from scratch, as there was no pre-trained version available, for 1250 epochs, was not able to generate audio with good naturality and intelligibility. We believe that the model should be trained for more epochs to reach good results, as expected. The experiments presented throughout this work showed the difficulties surrounding the construction of a TTS system for the Portuguese language and see the use of pre-trained models as a good alternative for projects with limited resources.

Keywords: Text-to-speech, Speech Synthesis, Generative Models, Deep Neural Networks

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O processo de texto para fala, *text-to-speech* (TTS), tem como objetivo fazer um computador, a partir de uma mensagem textual, gerar sinteticamente um sinal de fala. Para isto, é necessário que ele cumpra duas tarefas essenciais: Análise de texto, onde se lê um texto, o transformando em algum código computacionalmente interpretável e, a partir deste, a etapa de síntese, em que se produz um sinal de áudio inteligível e natural para sua aplicação. A versão simplificada deste processo é apresentada na Figura 1.1.

Um sistema de TTS possui diversas aplicações. Inicialmente eles foram desenvolvidos no intuito de realizar leituras para deficientes visuais, onde o sistema iria ler um texto, por exemplo, de um livro, e converter para uma fala. Atualmente, tais sistemas têm sido bastante utilizados na automação de *call-centers*, *chatbots* e assistentes virtuais. Porém, para estas aplicações, em que a interação com o usuário é essencial, sistemas com falas robóticas, não naturais, são vistos de forma negativa pelos usuários [1]. Portanto, cada vez mais, é necessário criar um modelo em que a fala produzida pelo computador seja mais similar a de um humano.

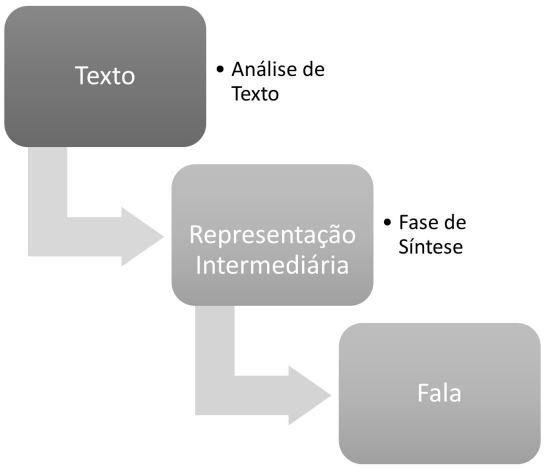

Figura 1.1: Diagrama de blocos simplificado de um sistema TTS (Adaptado de [2]).

A criação de um modelo de TTS que atenda as necessidades de suas aplicações possui diversos níveis de complexidade. Por exemplo, frases escritas com as mesmas palavras podem ter, dependendo da pontuação, sentidos diferentes ou o contexto pode fazer com que elas sejam pronunciadas com entonações e ritmos diferentes, no que chamamos

de estudo de prosódia [3]. Portanto, gravar um conjunto de palavras ou frases comuns e recombiná-las irá resultar em um modelo com baixa flexibilidade, afastando-se de seu objetivo principal, de ser capaz de falar qualquer coisa, independente da mensagem ter sido originalmente gravada ou não. Desta forma, a naturalidade da fala depende também do entendimento da sentença como um todo.

No estado da arte vemos, ao longo do tempo, diversos modelos para a síntese de voz a partir de um texto. Podemos citar três abordagens clássicas: síntese formante (*formant synthesis*), síntese articulatória (*articulatory synthesis*) e a síntese concatenativa (*concatenative synthesis*). A síntese formante é baseada num sistema de fonte e filtros, onde um sinal de excitação é gerado numa frequência fundamental e diversos filtros em série e em paralelo são conectados a esta fonte para gerar frequências formantes que se somam para gerar o sinal de fala. O modelo PARCAS, que segue a abordagem de síntese formante, conta com 16 parâmetros de controle e filtros em paralelo e série com o objetivo de simular o trato vocal [2]. A síntese articulatória sintetiza fala com base em modelos que simulam a fisiologia do trato vocal humano, como formato, posição e movimento dos orgãos ao longo do tempo. Esta abordagem é a mais precisa, porém é muito difícil realizar estas modelagens em sua totalidade [4]. A síntese concatenativa busca concatenar unidades pré-gravadas de fala, como fonemas, sílabas ou palavras inteiras. Esta última abordagem originou diversas técnicas, como Síntese baseada em Dífonos, Síntese por seleção de unidade e Síntese baseada em Modelo Oculto de Markov (HMM) [2].

Recentemente, o uso de redes neurais artificiais tem apresentado resultados bastante interessantes, como a *WaveNet* [5], um modelo auto-regressivo de síntese de fala, em que a rede aprende o mapeamento de um conjunto de características linguísticas, como informações relacionadas ao texto de uma sentença, para o domínio de amostras de sinal de áudio. Este modelo busca gerar a probabilidade condicional das amostras de sinal, ou seja, se infere a distribuição de probabilidade condicional do conjunto de amostras que formam o sinal, condicionadas pelas características linguísticas de um texto. Esta técnica tornou possível a geração de sinais comparáveis com áudios originais. Por ser auto-regressivo, ele sintetiza cada amostra à partir de suas anteriores, considerando a relação entre as variáveis ao longo do tempo. Desde então o uso de redes neurais profundas para geração de fala vem crescendo, buscando alternativas para algumas formulações propostas no WaveNet, como processamento paralelo, simplificação do mapeamento linguístico, velocidade de treinamento da rede, dentre outros.

Dentre novas arquiteturas podemos citar o Waveglow [6], que busca gerar a distribuição que mapeia o sinal à partir do espectrograma na escala mel, por meio de uma técnica de *normalizing flows* chamada Glow [7], o Tacotron 2 [8] que busca utilizar uma interpretação intermediária dos atributos textuais, o espectrograma na escala mel, por meio de um conjunto de dados no formato *Sentença, Áudio* e MelGAN [9], que utiliza-se de uma arquitetura baseada em GANs [10] para fazer uma rede que faça a inferência das amostras de áudio à partir de um espectrograma na escala mel.

Frente a estas novidades que a aplicação de redes neurais profundas nos sistemas de TTS vem gerando na maneira que estes são produzidos, este trabalho tem como objetivo

o estudo de recentes técnicas de *Text-to-speech* que se utilizam de redes neurais artificiais profundas. Além disso, este trabalho visa aplicar estas técnicas na produção de um modelo de TTS para a língua portuguesa, comparando a síntese gerada com áudios naturais de fala.

1.1 Custo de um sistema TTS

O avanço de técnicas na área de TTS permitiram a produção de áudios de maior naturalidade e inteligibilidade, diminuindo-se a diferença entre trechos de vozes humanas e sintéticas. Isto, em grande parte, foi graças às arquiteturas de redes neurais e modelos centrados em dados, que apesar de eficientes, requerem um grande volume de dados de alta qualidade, ou seja, que cubra todos os fonemas disponíveis na língua e suas variações linguísticas, além de gravações com alta inteligibilidade e desprovidas de ruídos, e centenas de horas para estes modelos aprenderem os parâmetros de sua arquitetura, necessários para o processamento de texto e síntese do áudio. Nesta seção, iremos explorar os diferentes desafios para se produzir um sistema TTS, além deste cenário para a língua portuguesa.

1.1.1 Uso comercial de um sistema TTS

Cada vez mais, os sistemas de síntese de fala por texto vêm sendo utilizados para diferentes usos. Dentre eles, podemos citar para usos de acessibilidade, onde pessoas com alguma deficiência visual necessitam de auxílio para leitura de textos, *Chatbots* para indústria financeira e afins, onde assistentes virtuais guiam seus usuários por meio de áudio, transmissões de rádio e notificações de tráfego e clima, como avisos de acidentes ou mal tempo no rádio; sistemas de rotas, automação de casas, sistemas de ensino remoto, dentre outros. Para que estas aplicações sejam produzidas, é necessário ter posse de um sistema TTS ou acesso a um serviço que o forneça.

Neste trabalho, foi feito um levantamento do custo de uso comercial dos principais sistemas de TTS no mercado e disponibilizado abaixo. É importante ressaltar que a maioria dos serviços possuem dois tipos distintos de sistema TTS: Voz padrão (não neural), que não utiliza-se de redes neurais, com abordagens mais clássicas, como a de síntese concatenativa, e o TTS Neural, que se utiliza de redes neurais profundas, como as que serão estudadas ao longo deste trabalho, para síntese de voz.

- **Google Cloud Text-to-Speech[1]:** Na tabela de preços, disponível no site da empresa, consta dois modelos: Voz padrão (não WaveNet), com o valor de 4 dólares a cada milhão de caracteres, e Voz WaveNet, baseada em redes neurais profundas, por 16 dólares a cada milhão de caracteres.

- **Amazon Polly[2]:** Neste serviço, é mostrado diversos modelos de pagamento conforme o uso, disponível no site da empresa, mas o valor padrão é de 4 dolares por milhão de caracteres para o TTS padrão e 16 dólares por milhão de caracteres pelo TTS neural.

[1] https://cloud.google.com/text-to-speech/pricing
[2] https://aws.amazon.com/pt/polly/pricing/

- **Watson Text to Speech**[3]**:** O plano padrão é no valor de 20 dólares a cada milhão de caracteres, com a disponibilidade de outros planos a consultar.
- **Microsoft Azure**[4]**:** 4 dólares por milhão de caracteres no modelo padrão e 16 dólares por milhão de caracteres no modelo TTS Neural.

Apesar de cada serviço utilizar um método específico de cobrança, pode-se ter uma noção do custo para uma empresa que deseja incluir um serviço baseado em conversão de texto para fala e que esteja disponível para diversos clientes simultaneamente.

De acordo com o que foi observado pelos serviços disponíveis, ter posse de um bom modelo de TTS, em que a voz sintetizada tenha alta naturalidade e inteligibilidade, é um negócio lucrativo. Mas o que deve ser questionado também é o custo para que estes modelos de alta qualidade sejam feitos, como a disponibilidade de um banco de dados e poder computacional suficientes para treinar as redes neurais que fazem isso possível.

1.1.2 Treinamento de um modelo

A etapa de treinamento de um modelo baseado em redes neurais profundas e ajuste de hiperparâmetros exige diversas horas e recursos computacionais para que bons resultados sejam atingidos. Principalmente na área de Processamento de Linguagem Natural, observamos modelos que possuem muitos parâmetros e necessitam de muitos recursos para serem treinados, como por exemplo o Google BERT, um algoritmo que aumentou a compreensão da linguagem humana pelos mecanismos de busca [11], que estima-se valores entre US$ 2.5 mil até US$ 1.6 milhão para que sejam treinados, a depender do tamanho do modelo, em número de parâmetros [12].

Dentre algumas arquiteturas utilizadas em sistemas de TTS, podemos verirficar os custos envolvidos no treinamento desses modelos. O WaveGlow [6] utilizou 8 placas Nvidia GV100 para treinar 580 mil iterações de uma base de dados de 16 mil clipes de áudio. O modelo apresentado em [13] relatou um treinamento de 877 mil iterações, totalizando cerca de 12 dias de treinamento. No Parallel WaveGAN [14] foi necessário cerca de 400 mil iterações, com uso de duas GPUs Tesla V100 para que os resultados fossem obtidos. Em MelGAN [9] os experimentos foram feitos à partir de 400 mil iterações na base LJ Speech [15], e o modelo utilizado de referência nos resultados precisou de 2,5 milhões de iterações para convergência. Na arquitetura ClariNet [16] foram necessárias 1 milhão de iterações para seu treinamento numa base de dados de 20 horas gravadas.

Com base nas arquiteturas utilizadas como referência, vemos que o processo de treinamento de um modelo precisa de muitas iterações para que se atinja a convergência desejada. Além disso, são necessários equipamentos com alta capacidade de processamento, como GPUs com centenas de Gb de memória e velocidade de processamento, para que trechos do *dataset* (*batches*[5]) sejam processados rapidamente a cada iteração e que um grande volume de dados seja usado em cada iteração.

[3]https://www.ibm.com/in-en/cloud/watson-text-to-speech/pricing

[4]https://azure.microsoft.com/pt-br/pricing/details/cognitive-services/speech-services/

[5]Agrupamento de n amostras da base de dados para treiná-las simultaneamente numa iteração de um modelo.

No Capítulo 2 serão abordados alguns conceitos necessários para o entendimento do trabalho, como a *pipeline* de um sistema de síntese de voz por texto e as arquiteturas utilizadas neste trabalho, assim como os conceitos que baseiam seu funcionamento.

Capítulo 2

Fundamentação Teórica

As aplicações de *text-to-speech* tiveram um grande salto de desempenho e naturalidade no som produzido à partir do WaveNet [5]. Desde então, o uso de redes neurais profundas para síntese de áudio acabou sendo mais explorado, buscando-se melhorar a qualidade do som produzido e a eficiência, em questão de recursos computacionais, dos métodos de treinamento. Neste capítulo, serão apresentados os conceitos essenciais para o entendimento do trabalho e das técnicas de TTS abordadas.

2.1 Sistemas TTS

O principal objetivo de um sistema TTS é criar um som natural de fala à partir de uma sequência de caracteres. O desenvolvimento desses sistemas é baseado em dois fatores principais, naturalidade e inteligibilidade, ou seja, que a fala se pareça originalmente vinda de um locutor humano e que a sentença seja de fácil compreensão. Ao longo do tempo, diversos modelos foram sugeridos para síntese de fala e sistemas de TTS, como as abordagens mais clássicas apresentadas no Capítulo 1.

Na última década, com o avanço da tecnologia computacional, novas técnicas orientadas a dados, que usam redes neurais artificias têm sido bastante utilizadas para aprender a síntese de fala à partir de um *Corpus*, uma coletânea de sentenças, composto de tuplas *(sentença, fala)*. Desta forma, as redes neurais desenvolvidas permitem extrair parâmetros estatísticos referentes ao mapeamento do texto em fala [2].

Dentro destas técnicas orientadas a dados, podemos separar a *pipeline* do processo em duas principais tarefas, que serão melhor explicadas nas seções seguintes:

- **Processamento de Texto**: Extrair as informações sequenciais do texto que compõe uma sentença, gerando uma representação intermediária. Para isto utilizam-se, geralmente, arquiteturas como as LSTMs (*Long short-term memory*) e as RNNs (*Recurrent neural networks*), que consigam processar uma sequência de caracteres e mapear para uma sequência em outro domínio, como o de espectrogramas na escala mel, como mostrado em [13] e [8].

- **Fase de Síntese**: Construir as amostras de áudio a partir de um conjunto de características, como frames de um espectrograma na escala mel [5], [6] e [9].

2.2 Processamento de texto

A tarefa do processamento de texto é extrair informações sequenciais do texto, gerando uma representação intermediária deste texto, para que estas sejam processadas nas etapas seguintes da *pipeline* do TTS. É de suma importância que a representação intermediária gerada pelo processamento do texto possa captar as características acústicas desta sequência de caracteres.

Os caracteres que compõem o texto muitas vezes podem não passar a informação correta da pronúncia das palavras. Para isto, é comum representar as palavras não só por seus conjuntos de caracteres como também por suas características acústicas, como seus fonemas. Desta forma, palavras que não existem no vocabulário do modelo terão sua representação dada pelos caracteres, enquanto que as que existirem no vocabulário terão a representação fonética e de caracteres. Assim, as palavras novas podem ter suas características acústicas inferidas [17]. Novas arquiteturas de redes neurais buscam identificar variações de pronúncia e significado à partir de mecanismos de atenção e processamento de sequências, como LSTMs e RNNs [8].

Com base nesta ideia, o Tacotron 2 [8] busca fazer o mapeamento sequencial do texto de forma a extrair outra sequência, o espectrograma na escala mel. Na sub-seção seguinte será apresentado os detalhes referentes a este componente de processamento de texto.

2.2.1 Tacotron 2

A Tacotron 2 é uma arquitetura de rede neural feita para síntese de fala diretamente do texto. O sistema proposto consiste de dois componentes: Uma rede recorrente *sequence-to-sequence*, que determina a sequência de *frames* de espectrograma na escala mel a partir da sequência de caracteres, e uma versão modificada do *WaveNet* [5], que gera sinais de áudio/fala à partir dos espectrograma na escala mel. Esta arquitetura é representada na Figura 2.1.

Modelos *seq2seq*, ou *sequence-to-sequence* são denominados assim pois a função deles é de converter sequências de um domínio para um outro. Neste caso, estamos convertendo uma sequência de caracteres para uma de frames de espectrograma na escala mel. O termo recorrente se deve ao fato de que a saída atual leva em consideração valores de saídas anteriores, ou seja, há recursão/realimentação. Os exemplos mais comuns de arquiteturas recorrentes são as RNNs e as LSTMs, sendo esta última utilizada na arquitetura do Tacotron 2.

Neste caso, a arquitetura utiliza-se de *frames* de espectrograma na escala mel como representação intermediária, isso pois esta representação é facilmente obtida à partir das formas de onda no domínio no tempo, ser mais compacta que amostras de forma de onda, e é mais fácil de se treinar usando o erro quadrático como métrica, devido sua invariância em relação à fase de cada *frame* [8].

Neste trabalho, iremos utilizar a parte de Processamento de Texto da arquitetura do Tacotron 2, para geração de espectrogramas na escala mel. Para síntese de voz (vocoder), será aproveitado o modelo pré-treinado do WaveGlow [6], disponível no repositório do

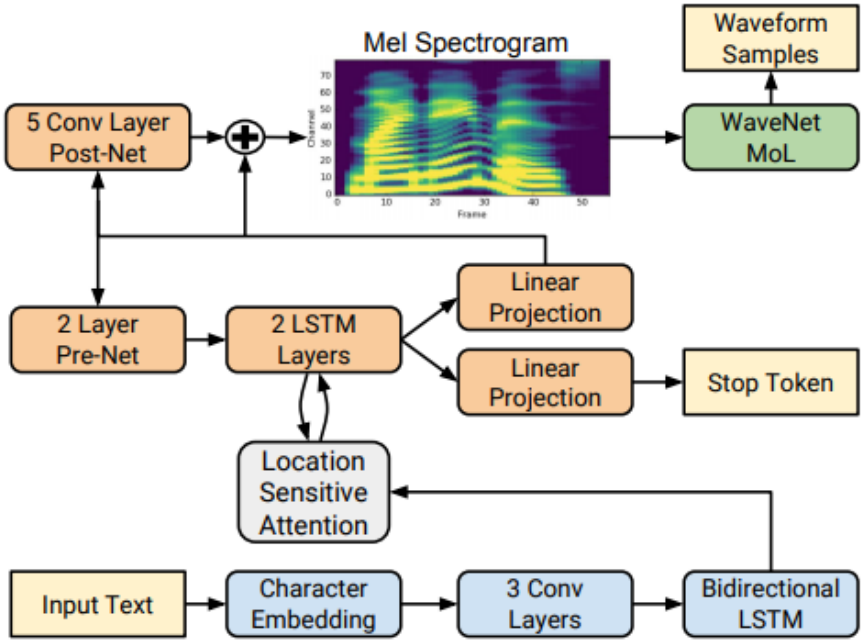

Figura 2.1: Arquitetura do Tacotron2 [8]

Tacotron 2 no Github [1].

2.3 Representação Intermediária de atributos

Como forma de separar os sistemas em duas etapas, de processamento de texto e de síntese de fala, é necessário uma representação que sirva de ponte entre elas. Durante muitas décadas, a representação utilizada é do espectrograma na escala mel: Uma versão não linear do espectrograma linear, obtido através de uma Transformada de Fourier de Tempo Curto (STFT, do inglês *Short-Time Fourier Transform*) do sinal, onde se aplica uma transformação não linear no eixo da frequência, como mostrado na figura 2.2, inspirado nas respostas do sistema auditório humano.

Sistema	MOS
Tacotron 2 (Linear + GL)	3.944 ± 0.0091
Tacotron 2 (Linear + WaveNet)	4.510 ± 0.054
Tacotron 2 (Mel + WaveNet)	**4.526 ± 0.066**

Tabela 2.1: Comparação de avaliação de MOS para Griffin-Lim vs. WaveNet como vocoders, usando 1,025 dimensões no espectrograma linear vs. 80 dimensões no espectrograma mel como entradas condicionais do WaveNet [6].

O espectrograma é uma representação do sinal pelo tempo com a adição da informação das frequências que formam cada trecho/período de tempo. Esta representação permite

[1] https://github.com/NVIDIA/tacotron2

observar parâmetros importantes da fala, como as frequências fundamentais e formantes. Para se obter o espectrograma através do sinal aplica-se uma transformação, como a Transformada de Fourier de tempo curto (STFT), que é uma Transformada Discreta de Fourier (DFT) sobre curtas janelas sobrepostas, que quando concatenadas sob todo o sinal, permitem a visualização das frequências ao longo do tempo. A DFT de um sinal discreto $x[n]$ é dada por:

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j(2\pi/N)kn} \tag{2.1}$$

onde $k = 0, ..., (N-1)$.

Quando a STFT é calculada com sobreposição entre as janelas, é possível recuperar a informação de fase do sinal e reconstruí-lo através de uma Transformada Inversa de Fourier de Tempo curto (iSTFT) [18]. Porém, os espectrogramas na escala mel apresentam maior perdas em relação ao espectrograma linear, além disso, sua representação é dada por um número discreto de canais, gerando uma representação mais simples do sinal mas dificultando a tarefa de inversão [8].

A escala mel enfatiza baixas frequências, que são críticas na inteligibilidade do sinal, enquanto possui menos canais para frequências maiores, que são dominadas por ruídos e fricativos sonoros [8]. Desta forma, é possível obter uma representação das frequências utilizando um menor número de coeficientes. Em [6] foi feita uma comparação da qualidade dos áudios gerados por um espectrograma linear e na escala mel. Os resultados, apresentados na Tabela 2.1 mostram que a qualidade dos áudios gerados, medida pelo *Mean Opinion Score*[2] (MOS), uma métrica subjetiva que depende de uma avaliação imparcial de um grupo de ouvintes, não diferiu muito à partir da escolha do espectrograma, sendo que o espectrograma linear utiliza 1025 dimensões para representar as faixas de frequência, enquanto que na escala mel são utilizadas apenas 80 dimensões. Este resultado, mostra que, apesar as perdas geradas esperadas à partir da diminuição da resolução do espectrograma, a forma que a escala mel é construída permite enfatizar faixas em que a informação é realmente necessária para reconstrução do sinal, resultando numa qualidade equivalente em menor resolução.

2.4 Modelagem do áudio

Um sinal de fala amostrado possui, geralmente, uma taxa de 16.000 amostras por segundo, e estas amostras possuem dependências temporais de curto e longo termo, isso pois estudos indicam que as frequências que influenciam na inteligibilidade da fala se limitam até as faixas de 8 kHz [19] e pelo Teorema de Nyquist, a taxa de amostragem de um sinal deve ser pelo menos duas vezes maior que a maior componente de frequência do sinal em tempo contínuo [20]. Certas abordagens buscam simplificar modelos de processamento de texto utilizando representações de menor resolução do sinal, como o seu espectrograma [9]. Para estes casos, é necessária a reconstrução do sinal a partir da inversão do espectrograma,

[2]Mean Opinion Score: Uma métrica subjetiva que avalia a qualidade de experiência, numa escala de 1 a 5

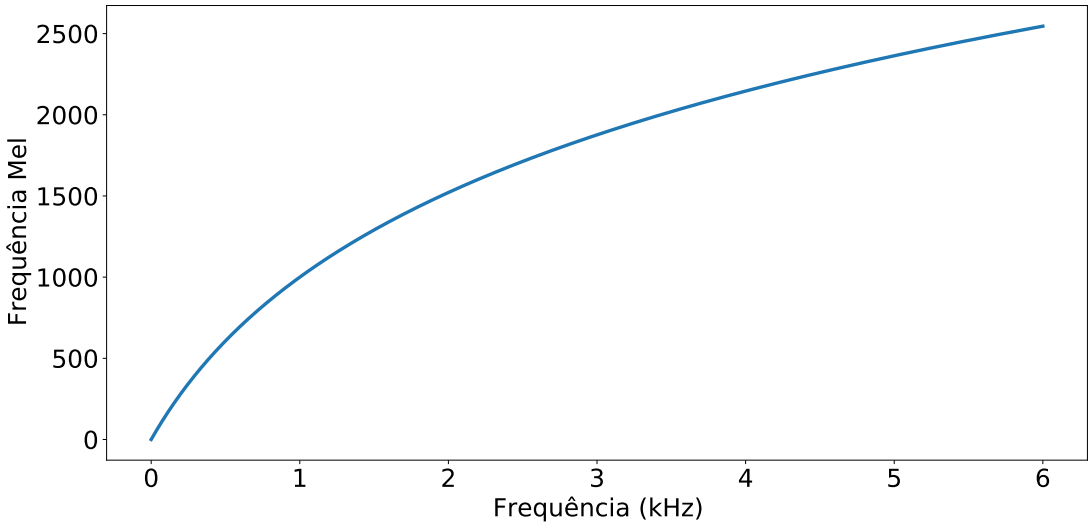

Figura 2.2: Escala mel em função da frequência. Fonte: Autor.

como o algoritmo *Griffin-Lim* [18], que realiza a inversão do espectrograma, reconstruindo a sua fase, necessária para realizar a inversão. A tarefa da síntese de áudio é, através desta representação intermediária e certos parâmetros. Tais como fonemas que compõem um texto e características específicas do locutor, como o timbre, gerar amostras temporais do sinal de áudio.

2.4.1 Modelos generativos

Um modelo generativo busca mapear a distribuição de probabilidade conjunta de um grupo de variáveis $P(X|Y)$. No contexto de síntese de áudio, esses modelos buscam identificar a distribuição conjunta entre amostras de áudio (X) e atributos linguísticos (Y), como o espectrograma da escala mel. Para que seja obtida essa distribuição, existem três abordagens mais comuns, que buscam modelar uma distribuição inicial, como uma distribuição normal, a fim de se estimar a distribuição do conjunto de estudo. As três abordagens mais comuns de modelos generativos são os *Variational Auto Encoders* (VAE), *Generative Adversarial Networks* (GAN) e os *Normalizing flow models*.

Os modelos baseados em *flow*, cujo o conceito é aplicar um série de transformações inversíveis numa distribuição gaussiana de entrada até que se consiga a distribuição dos dados de treino. Por serem transformações inversíveis, pode-se calcular sua matriz jacobiana [3], que é a derivada das funções de transformação de uma rede neural, permitindo a propagação do erro das transformações de cada variável parâmetro presente nestas arquiteturas. Consequentemente é possível otimizar estes parâmetros baseando-se no erro final e a parcela de cada parâmetro. Na Figura 2.3 podemos ver uma representação visual do modelo de transformações na distribuição de probabilidade, até atingir uma probabilidade similar à variável de estudo. O Waveglow [6], modelo de síntese de fala, utiliza-se de um modelo baseado em flow para a síntese das amostras de áudio.

[3] matriz formada pelas derivadas parciais de primeira ordem de uma função vetorial

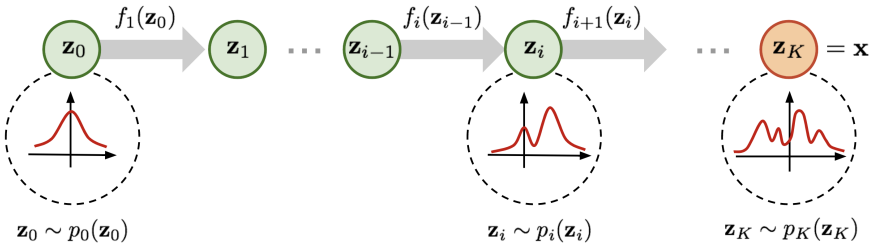

Figura 2.3: Esquema de transformações inversíveis dos modelos baseados em *normalizing flow* [21].

As Redes Adversárias Generativas, *Generative Adversarial Networks* em inglês, tem como princípio o uso de duas estruturas, um gerador e um discriminador. O gerador é um modelo que recebe variáveis no espaço latente, uma representação que contem informação codificada do espaço observável, e gera amostras no espaço observável, enquanto que o papel do discriminador é distinguir variáveis reais das geradas pelo gerador [10]. Esta dinâmica torna possível treinar as duas redes simultaneamente, fazendo com que o gerador seja capaz de gerar amostras sintéticas muito semelhantes com os dados originais. Com base nessa ideia, foi criada a MelGAN [9], uma arquitetura de rede neural para síntese de áudio baseada em GAN. No domínio de síntese de voz, a arquitetura das redes funciona da seguinte maneira: O objetivo do gerador é sintetizar a voz a partir de um conjunto de variáveis, o espectrograma na escala mel. O discriminador recebe amostras de sinais de fala reais e sinais sintetizados pelo gerador e seu objetivo é distinguir os sinais originais dos sintéticos. É nesta interação que uma rede busca aperfeiçoar a síntese de áudio e a outra melhorar sua habilidade de discriminar as amostras analisadas. Desta forma, ao final do treinamento, o gerador estará otimizado com a capacidade de gerar áudios com boa qualidade. Na Figura 2.4 é mostrado um esquema da arquitetura de uma GAN e a relação entre as duas redes na etapa de treinamento.

2.4.2 MelGAN

A MelGAN foi a primeira arquitetura a produzir áudios de alta qualidade por meio de GANs, Redes Adversárias Generativas, uma arquitetura que permite aprender a inversão do espectrograma por meio de um pequeno número de parâmetros, comparado a outros modelos de redes neurais mais comuns, como Wavenet, Clarinet e WaveGlow. Além disso, por não ser um modelo auto-regressivo, é possível realizar a sua paralelização, aumentando a eficiência no uso de GPUs e TPUs [4]

A arquitetura da MelGAN possui dois componentes: o *Generator*, que é efetivamente o responsável em inverter o espectrograma para a forma de onda de fala, e o *Discriminator*, responsável por distinguir amostras sintéticas das originais. O *Generator* é uma rede com diversas camadas de convolução que tem como entrada um espectrograma na escala mel s e as amostras da forma de onda x na saída. Como o espectrograma utilizado nos experimentos possui resolução menor que as amostras desejadas na saída, as camadas de

[4]Do inglês *Tensor Processing Unit*, são processadores que operam diretamente sobre tensores, um tipo de dado específico utilizado em aplicações de Inteligência Artificial.

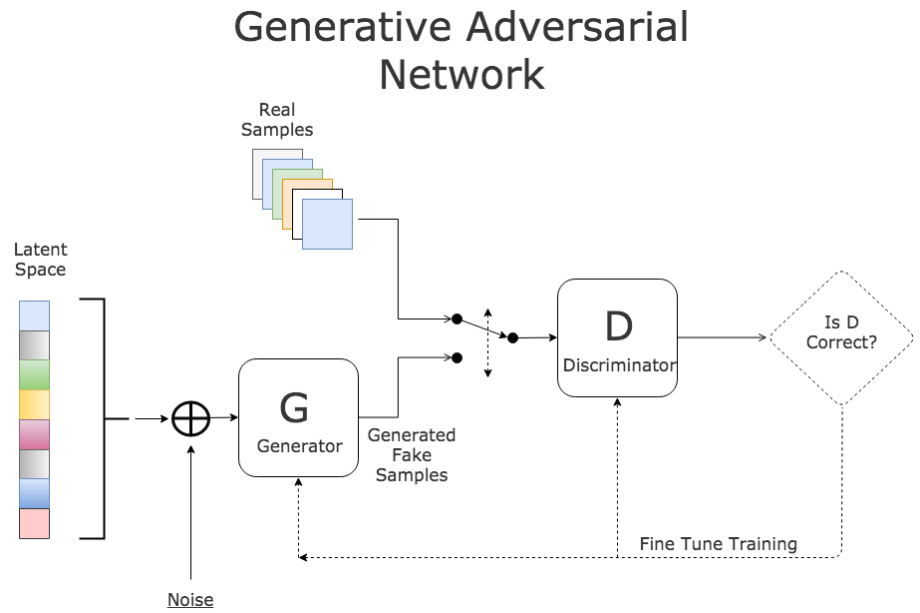

Figura 2.4: Arquitetura de uma Rede Adversária Generativa (GAN). Onde o Gerador tem a tarefa de gerar amostras à partir de um espaço latente e o Discriminador deve diferir uma amostra falsa da real. Essa iteração aumenta a capacidade de gerar amostras mais realistas do Gerador e o poder discriminante do Discriminador. Fonte [22].

convolução são utilizadas para o *Upsample* das amostras de entrada s. Depois de cada camada de convolução existe um bloco de convoluções dilatadas, uma convolução que leva em consideração as amostras de passos anteriores, importante visto a alta correlação entre as amostras ao longo do tempo. O *Discriminator* é composto de 3 discriminadores (D1, D2, D3) idênticos mas que operam em 3 diferentes escalas de áudio, D1 é na escala original, D2 e D3 são decimados pelo fator de 2 e 4, respectivamente. Desta forma, cada discriminador se especializa em diferentes frequências. Por exemplo, D1 será capaz de identificar maiores variações em frequências maiores, enquanto que D3, por trabalhar numa taxa menor, irá identificar as variações em frequências mais baixas. Na Figura 2.5 é apresentada a arquitetura da MelGAN, onde é possível identificar os componentes explicados acima. Por fim, a tarefa do *Discriminator* é diferenciar amostras sintéticas geradas pelo *Generator* de originais e é esta interação entre estes dois componentes que acaba por otimizá-los, onde o *Generator* será capaz de gerar áudios semelhantes ao original.

2.4.3 WaveGlow

O WaveGlow [6] é uma arquitetura de modelo generativo baseado em *flow*, que modela a distribuição das amostras de áudio através de uma gaussiana na mesma dimensão da saída desejada e esta é transformada por uma série de camadas até se obter a distribuição desejada. O modelo é treinado buscando minimizar a log-verossimilhança negativa entre o sinal sintetizado e o original, e isto é possível, pois as transformações aplicadas ao longo da rede são inversíveis. Na Figura 2.6 é mostrada a arquitetura do WaveGlow, composta por diversos blocos de convolução, que efetivamente modelam a distribuição normal até a distribuição desejada. Pode-se observar que a arquitetura possui um grande número

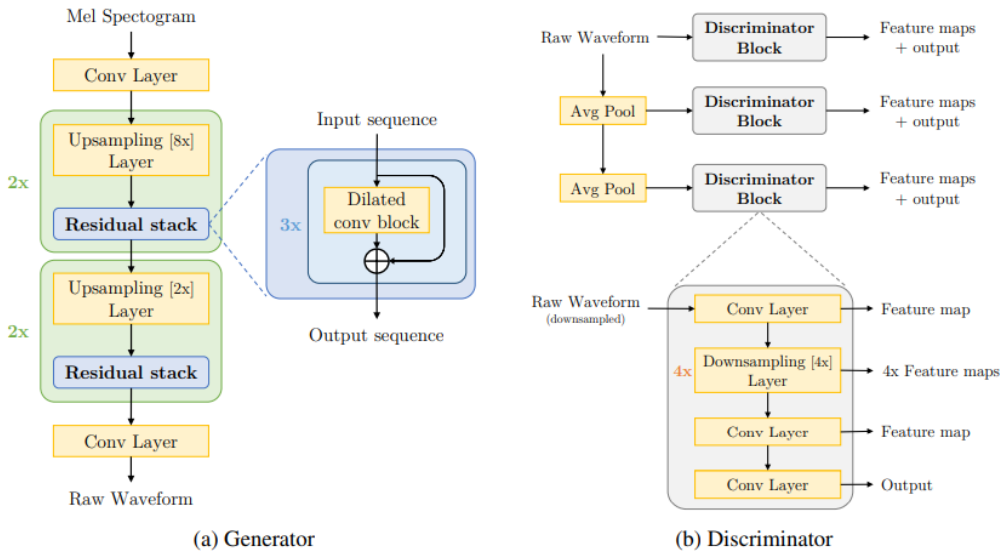

Figura 2.5: Arquitetura do MelGAN. Composta por duas redes, um Gerador, responsável pela síntese de áudio, e um Discriminador, responsável por discriminar amostras geradas pelo Gerador e amostras originais [9].

de parâmetros devido a repetição desses blocos. Desta forma, o WaveGlow torna-se um modelo com um alto custo computacional para treinar, pois as iterações levam maior tempo de processamento, em comparação com outros modelos de síntese de voz.

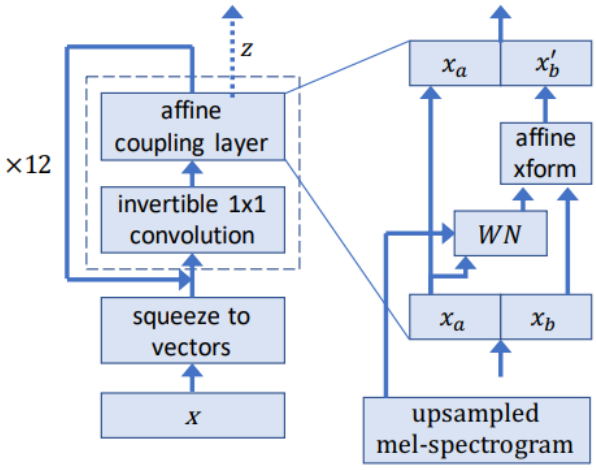

Figura 2.6: Arquitetura da Waveglow. Composta por 12 blocos de convolução e camadas de acoplamento [6].

2.5 Base de dados

A qualidade da base de dados utilizada para treinar a rede neural responsável pela síntese de fala se reflete na qualidade do sinal gerado, isso pois o *corpus* utilizado deve cobrir as diversas variações fonéticas da língua. A maioria dos sistemas TTS comercialmente vendidos são treinados em bancos de dados de código fechado, sendo necessário buscar

alternativas de código aberto. No estado da arte de TTS na língua inglesa, o banco de dados mais utilizado é o *LJ Speech Dataset* [15]: Uma base em domínio público composta de 13.100 sentenças curtas de áudio de um único locutor, variando entre 1 a 10 segundos, totalizando aproximadamente 24 horas de áudio.

Para língua portuguesa ainda não existe um *dataset* em domínio público na qualidade e quantidade equiparável ao *LJ Speech* e isto pode influenciar na capacidade de se refinar um modelo de TTS orientado a dados. Os modelos estudados neste projeto são projetados, em sua maioria, para um único locutor, e desta forma o banco de dados mais apropriado encontrado foi o *TTS-Portuguese Corpus* [23]: Uma base de aproximadamente 10 horas de áudio, contendo 3.632 sentenças, variando entre 0,67 e 50 segundos, num ambiente silencioso, porém não gravado num estúdio acústico.

No Capítulo 3 será apresentada a metodologia abordada neste trabalho, como as tecnologias utilizadas para cada etapa da síntese de fala, assim como o ambiente de desenvolvimento utilizado para elaboração dos testes realizados neste trabalho.

Capítulo 3

Metodologia

Como já dito anteriomente, o processo de síntese de fala pelo texto é, na maioria das vezes, dividido em duas etapas: Processamento de texto - onde o texto será codificado para uma linguagem que represente de maneira concisa o texto e intenção de fala, num processo que identifica a semântica das palavras e reduz o espaço para um comprimento limitado (*embedding*); A síntese de fala - que irá, através do texto codificado pela etapa do *encoder*, decodificar para o espaço das ondas de voz, com o intuito de modelar o áudio de fala. Na figura 3.1 é mostrado um exemplo do sistema TTS, onde o modelo Seq2Seq é um componente que processa o texto, gerando uma representação intermediária, o espectrograma na escala mel, e o MelGAN, uma componente que sintetiza a fala à partir do espectrograma na escala mel. O motivo de sua escolha foi, principalmente, devido o número de parâmetros a otimizar ser menor que os outros modelos propostos e, consequentemente, maior velocidade de treinamento. Pois, como este trabalho foi feito com limitações de recursos, deseja-se diminuir o custo computacional para que seja possível obter um resultado com os recursos disponíveis.

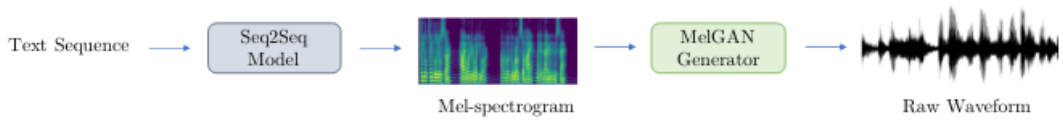

Figura 3.1: Exemplo de uma pipeline TTS. Onde a sequência de texto passa por um modelo que processa sequências para gerar um espectrograma na escala mel, que será utilizada como entrada do inversor de espectrograma, o MelGAN. [9].

3.1 Ambiente de Desenvolvimento

Os experimentos foram realizados através de uma máquina virtual no Microsoft Azure, da série NC6 — equipada com uma GPU Tesla K80 de 12 GB da NVIDIA [1], 6 núcleos e 56 GB de RAM.

[1] A placa possui 24 GB distribuídos em 2 chips GPU, onde somente um é disponibilizado na instância

Os códigos das arquiteturas utilizadas neste trabalho foram obtidos dos repositórios no *Github* apontados na Metodologia. A linguagem utilizada, tanto nos componentes, quanto nas manipulações dos resultados, foi a Python[2]. As bibliotecas utilizadas para visualização dos resultados dos experimentos foram Matplotlib[3], Librosa[4], SciPy[5], dentre outras. Mais informações a respeito do ambiente de desenvolvimento podem ser obtidas no repositório no *Github* deste trabalho[6].

3.2 Base de Dados

O banco de dados utilizado para o treinamento dos modelos foi o *TTS-Portuguese Corpus* [23], que possui 3.632 arquivos de áudio no formato WAV, variando entre 0,67 to 50,08 segundos de duração, gravado em 48 kHz e também disponível em 22 kHz, no formato PCM 32 Bits. Utilizou-se os áudios na taxa de 22 kHz e o formato foi convertido para PCM 16 Bits, com o auxilio da biblioteca SoundFile [7], que possui ferramentas de manipulação de áudio na linguagem Python. Estes arquivos de áudio, foram divididos em conjuntos de treino, validação e teste, definido conforme disponibilizado pelo próprio autor da base — 3084 para o conjunto de treino, 499 para o conjunto de validação e 34 de teste.

A variação da duração dos áudios era muito grande e isto pode causar problemas na hora do treinamento em *batches*, visto que o consumo de memória seria muito variado, a depender do conjunto de arquivos aleatoriamente selecionados por iteração e consequentemente o tamanho do conjunto de dados deve ser reduzido para não passar a capacidade da GPU. Como forma de amenizar este problema, criou-se uma base de dados variada onde cada arquivo de áudio foi dividido com comprimento máximo de 4 segundos. Para que os cortes dos áudios não fossem brutos, foi utilizado um detetor de atividade de voz (VAD) por limiar de amplitude, desta forma os cortes só ocorrem em momentos de silencio. Estas subamostras continuaram no conjunto de dados predeterminados e só foram utilizadas na etapa de Inversão do Espectrograma, onde não é necessário alinhar o texto com o áudio.

As mudanças no formato da base de dados utilizada foi para adaptar a entrada de acordo com modelos pré-treinados disponíveis para as arquiteturas utilizadas neste trabalho. Tinha-se disponível, em seus repositórios oficiais, versões pré-treinadas das arquiteturas do Tacotron 2 e WaveGlow, tornando possível o Aprendizado por Transferência, do inglês *Transfer Learning* [24]. Esta técnica diz que é possível se aproveitar de parte da arquitetura, já treinada numa tarefa específica, para uma tarefa secundária. Dessa forma, com auxílio deste artifício, os modelos pré-treinados podem permitir uma convergência mais rápida, à partir de um menor número de iterações, o treinamento para uma nova tarefa. No domínio deste trabalho, estamos utilizando modelos treinados para a língua inglesa, e a partir deles queremos treiná-los para um mapeamento na língua portuguesa.

[2] www.python.org
[3] www.matplotlib.org
[4] www.librosa.org
[5] www.scipy.org
[6] github.com/gabrielmalfatti/TTS-portuguese
[7] pysoundfile.readthedocs.io

Parâmetro	Valor
sampling rate	22050
n_mel_channels	80
filter_length	1024
hop_length	256
win_length	1024
symbols_embedding_dim	512
encoder_kernel_size	5
enconder_n_convolutions	3
encoder_embedding_dim	512
decoder_rnn_dim	1024
prenet_dim	256
max_decoder_steps	1000
batch_size	8

Tabela 3.1: Parâmetros utilizados na arquitetura do Tacotron 2

3.3 Texto para Espectrograma

Para a conversão de texto para espectrograma foi escolhido o modelo Tacotron 2 [8]. Uma arquitetura em código aberto que converte diretamente texto em fala, treinado à partir de uma base de dados no formato <*texto, fala*>. Sua escolha foi devido os resultados relatados no artigo e a premissa dele conseguir gerar um bom sistema de texto para fala com uma versão acústica compacta de atributos, o espectrograma na escala mel. O modelo conseguiu atingir um *Mean Opinion Score* (MOS) de 4,53, comparável com um MOS de 4,58 de uma fala gravada profissionalmente. O trabalho junta a arquitetura baseada em LSTM que converte texto em espectrograma com uma versão modificada do WaveNet [5] para síntese do sinal de fala, enquanto que sua implementação apresenta o WaveGlow [6] como *vocoder*.

Como o treinamento de um modelo destes pode levar semanas para que atinja uma convergência partindo-se de parâmetros iniciais, a estratégia para esta etapa foi de utilizar um modelo pré-treinado em inglês, disponível no repositório do projeto. Desta forma, são necessárias menos iterações para que este aprenda realizar o mapeamento em outra língua. Por se tratar de um alfabeto diferente, foi necessário alterar os símbolos utilizados para incluir alguns símbolos da língua portuguesa, como as vogais acompanhadas de acentos. Isso foi feito pela modificação do alfabeto definido no código desta arquitetura. Além disso, foi utilizado um limpador de texto básico, onde não há interpretação de algumas abreviações do inglês, como $drs = doctors$. Por fim, seguindo as diferenças de alfabeto, o argumento *warm-start* foi utilizado para iniciar o treinamento. Isso significa que as camadas de *embedding*, que são as que fazem o mapeamento do texto para um vetor numérico interpretável computacionalmente, são ignoradas e treinadas de acordo com o novo alfabeto. Os outros parâmetros foram utilizados de acordo com o padrão do repositório e são mostrados na tabela 3.1.

O modelo de processamento de texto Tacotron 2 foi treinado ao longo de 13500 iterações pela base de dados na língua portuguesa, o que equivale a cerca de 40 horas de treinamento na instância utilizada.

3.4 Inversão do Espectrograma

Uma vez que o texto é mapeado para sua forma em espectrograma na escala mel, é necessário um sistema que faça a inversão deste espectrograma para se obter o sinal de fala. Como o espectrograma na escala mel é uma versão compacta, e com perdas do sinal original, esta inversão não pode ser feita diretamente com o cálculo da Transformada Inversa de tempo curto (iSTFT) nas janelas do espectrograma, por exemplo. Então é necessário uma outra arquitetura de rede neural que faça essa inferência. Neste trabalho, treinou-se uma rede neural adversarial, a MelGAN [9], que devido seu menor número de parâmetros presentes acredita-se convergirem em menor tempo em comparação com outras redes neurais. Como forma de referência desta etapa, utilizou-se um modelo pré-treinado de Inversão de Espectrograma do WaveGlow [6], disponibilizado no repositório do Tacotron 2, que apesar de ter sido treinado na língua inglesa, acredita-se que a inversão do espectrograma não tenha tanta influência da língua usada na base de dados. Nas próximas seções estas duas arquiteturas serão explicadas, assim como o processo de treinamento da MelGAN.

3.4.1 MelGAN

Este modelo se destaca por ser um dos primeiros a sintetizar fala com boa qualidade, definida pelo índice MOS dos áudios gerados por ele, através de uma simples arquitetura GAN — uma arquitetura que conta com 2 redes neurais profundas: o gerador - que busca sintetizar o sinal de voz; e o discriminador - que busca discriminar os áudios sintetizados pelo gerador. O número de parâmetros utilizados em sua arquitetura é menor que nas arquiteturas utilizadas na literatura, como mostra a Tabela 3.2. Isso acaba por influenciar a velocidade de inferência e de treinamento. É importante destacar que a velocidade de inferência é influenciada por outros fatores além do número de parâmetros, como o fato do modelo ser auto-regressivo ou não, permitindo a geração de amostras de forma paralela, como mostra a diferença de inferência entre o Wavenet (auto-regressivo) e Waveglow (não auto-regressivo).

Como os recursos computacionais para este trabalho são limitados, acredita-se que a MelGAN pode fornecer um áudio natural e inteligível com um menor custo computacional de treinamento. Por fim, é extraído o espectrograma dos sinais de áudio e as redes são treinadas para reconstruir o sinal, utilizando-se como erro a *L1-loss*[8] em relação ao sinal original, que é uma espécie de Erro Médio Absoluto calculado ponto a ponto entre o sinal original e o sintetizado. Na Tabela 3.3 são mostrados os parâmetros utilizados no treinamento do modelo, para reprodutibilidade.

Como a escala utilizada no espectrograma na escala mel do modelo no repositório do MelGAN é diferente da utilizada no Tacotron 2, optou-se por usar um repositório Github [9] alternativo com a implementação do MelGAN em que o espectrograma calculado é igual ao gerado pelo Tacotron 2. Desta forma, não seria necessária nenhuma conversão entre

[8] https://pytorch.org/docs/stable/generated/torch.nn.L1Loss.html
[9] https://github.com/seungwonpark/melgan

Modelo	Nº parâmetros (milhões)	Velocidade em CPU (em kHz)	Velocidade em GPU (em kHz)
Wavenet	24,7	0,0627	0,0787
Clarinet	10,0	1,96	221
WaveGlow	87,9	1,58	223
MelGAN	**4,26**	**51,9**	**2500**

Tabela 3.2: Comparação do número de parâmetros e velocidade de inferência de vocoders. Velocidade em n kHz significa que o modelo consegue gerar n x 1000 amostras de áudio por segundo. Todos modelos foram avaliados utilizando o mesmo hardware. Fonte [9].

as duas etapas do sistema e então seria possível ter um sistema direto, onde a saída do Tacotron 2 pode ser utilizada como entrada do MelGAN.

A rede treinada para inversão de espectrograma não tinha disponível, no caso do MelGAN, modelos pré-treinados e, dessa forma, esperava-se um maior tempo para otimização da rede. Dessa forma, a etapa de treinamento passou por cerca de 240 mil iterações, equivalente a 1250 épocas. Este treinamento durou cerca de 96 horas na instância utilizada para os experimentos.

Parâmetro	**Valor**
n_mel_channels	80
ngf	32
n_residual_layers	3
ndf	16
num_D	3
n_layers_D	4
downsamp_factor	4
lambda_feat	10
batch_size	16
learning rate	0.0001
$\beta 1$	0.5
$\beta 2$	0.9

Tabela 3.3: Parâmetros utilizados no modelo MelGAN.

3.4.2 WaveGlow

Como a arquitetura do WaveGlow possui um elevado número de parâmetros, o treinamento dela seria muito lento e consumiria muitos recursos computacionais. Desta forma, o WaveGlow foi utilizado apenas como referência, através de um modelo pré-treinado na língua inglesa, para comparação com o modelo treinado do MelGAN.

Capítulo 4

Resultados e Discussão

Para avaliar a síntese de fala, é comum se utilizar o *Mean Opinion Score* (MOS), que necessita de um grupo de ouvintes imparciais avaliando a qualidade do áudio. Para isso ser possível, é necessário o uso de serviços de *crowdsourcing* [1], como o Amazon Mechanical Turk [2]. Como a utilização deste serviço está fora do escopo do trabalho, os sistemas foram comparados com base na forma de onda gerada, seu espectrograma na escala mel e a qualidade perceptível dos áudios gerados. As amostras de som geradas pelos experimentos apresentados nesta seção, assim como a forma de onda e espectrograma de todos arquivos do conjunto de teste, estão disponíveis no repositório do Github deste trabalho.

Para avaliação dos sistemas e modelos utilizados neste trabalho, foram realizados 4 experimentos para geração de fala:

1. **Mel Original + Waveglow:** Utilização do espectrograma na escala mel gerado diretamente dos áudios originais do conjunto de testes, como entrada do vocoder pré-treinado em ingles WaveGlow para síntese do áudio.

2. **Mel Original + MelGAN:** Utilização do espectrograma na escala mel gerado diretamente dos áudios originais do conjunto de testes, como entrada do vocoder treinado MelGAN para síntese do áudio.

3. **Tacotron 2 + Waveglow:** Utilização do espectrograma na escala mel gerado pelo Tacotron 2 à partir das sentenças do conjunto de testes, como entrada do vocoder pré-treinado em ingles WaveGlow para síntese do áudio.

4. **Tacotron 2 + MelGAN:** Utilização do espectrograma na escala mel gerado pelo Tacotron 2 à partir das sentenças do conjunto de testes, como entrada do vocoder treinado MelGAN para síntese do áudio.

A análise destes experimentos será apresentada sobre 4 sentenças selecionadas do conjunto de testes da base de dados utilizada para o treinamento dos modelos, dispostas abaixo.

[1] Solicitação de um serviço para um grupo de pessoas, de forma terceirizada, neste caso a avaliação da fala sintetizada

[2] Disponível em: www.mturk.com

(a) "Para as pessoas estranhas o panorama é desolador."

(b) "Em muitas cidades a população está diminuindo."

(c) "Nunca se deve ficar em cima do morro."

(d) "O ministério mudou demais com a eleição."

Nas Figuras 4.1 a 4.4 estão dispostas as formas de onda e espectrogramas das sentenças originalmente gravadas e dos resultados das ondas geradas nos Experimentos 1 a 4, respectivamente. Na Figura 4.5 é apresentado o mapa de cores dos espectrogramas expostos nesta seção.

4.1 Mel Original + Waveglow

Este experimento serve de referência para os outros, pois o espetrograma utilizado é uma conversão direta dos áudios presentes no conjunto de teste e em seguida eles foram utilizados como entrada do *vocoder* Waveglow, que foi treinado numa base de dados em inglês. Desta forma, o experimento avalia a capacidade do Waveglow de mapear os espectrogramas para o sinal de áudio.

Os sinais de áudio gerados neste experimento foram bem semelhantes aos áudios originais, comparando-se pelas formas de onda e espectrograma plotadas, além do áudio perceptivelmente possui alta inteligibilidade e naturalidade. Este resultado se explica pois o modelo pré-treinado disponível do waveglow é extremamente otimizado e, apesar de ser treinado na língua inglesa, a tarefa de mapear os espectrogramas para amostras do sinal mostra não ser influenciada por isso, desde que o mapeamento da rede neural consiga cobrir os fonemas da língua portuguesa.

4.2 Mel Original + MelGAN

O objetivo deste experimento é verificar a capacidade do Vocoder MelGAN realizar a inversão de espectrograma, através dos espectrogramas obtidos diretamente dos áudios do conjunto de testes, portanto não há grandes perdas no processo de geração do espectrograma a não ser pelas perdas da gravação da base de dados.

É possível observar pela comparação dos espectrogramas das Figuras 4.1 a 4.4 que o MelGAN falha no mapeamento de algumas frequências intermediárias e mais altas, causando consequentemente deformações na forma de onda gerada. Além disso, as faixas de frequências mais baixas não são bem definidas como no sinal original e como estas são determinantes para inteligibilidade do áudio, para frequências fundamentais e formantes, percebe-se este efeito nos áudios gerados. Os áudios gerados pelo *vocoder* também possuem baixa inteligibilidade e acredita-se que para obter um modelo mais refinado seria necessário realizar mais iterações na etapa de treinamento, visto que ao longo das épocas notou-se uma melhoria significativa no modelo, porém não o suficiente para gerar um áudio na qualidade esperada.

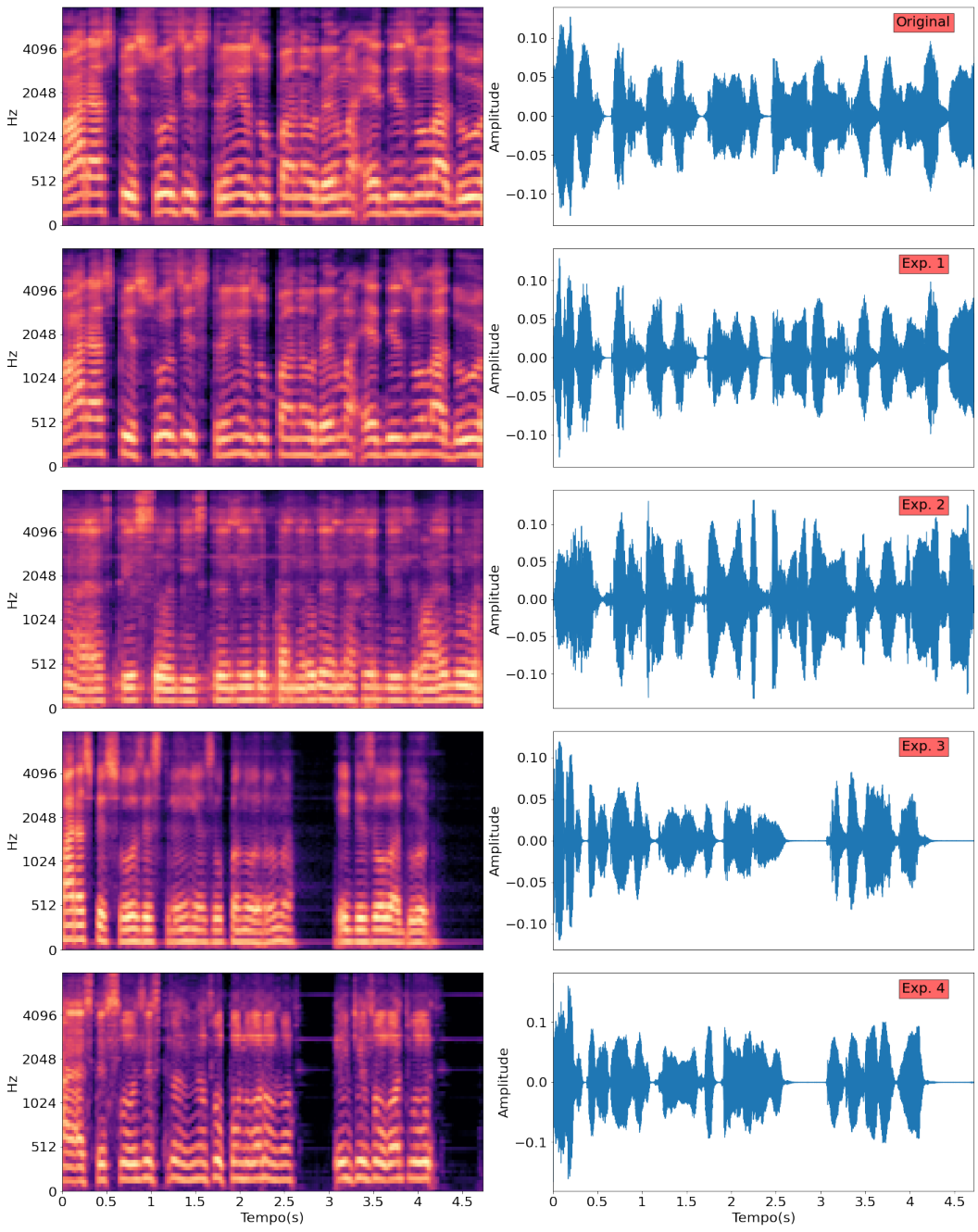

Figura 4.1: Espectrograma e Forma de onda dos sinais para a sentença (a) — "Para as pessoas estranhas o panorama é desolador". De cima para baixo, espectrogramas dos áudios originais e dos Experimentos de 1 a 4, respectivamente.

4.3 Tacotron 2 + Waveglow

Neste experimento podemos observar o funcionamento completo do sistema TTS, onde o texto descrito na entrada do Tacotron 2 terá um espectrograma na saída e este alimentando a entrada do *vocoder* Waveglow. Em comparação com os Experimentos 1 e 2, podemos observar as diferenças com a inclusão do Tacotron 2 no sistema.

Pelo fato do processo de geração do espectrograma ser diferente dos Experimentos 1 e 2, podemos observar que a forma de onda difere desses citados. Uma nova característica

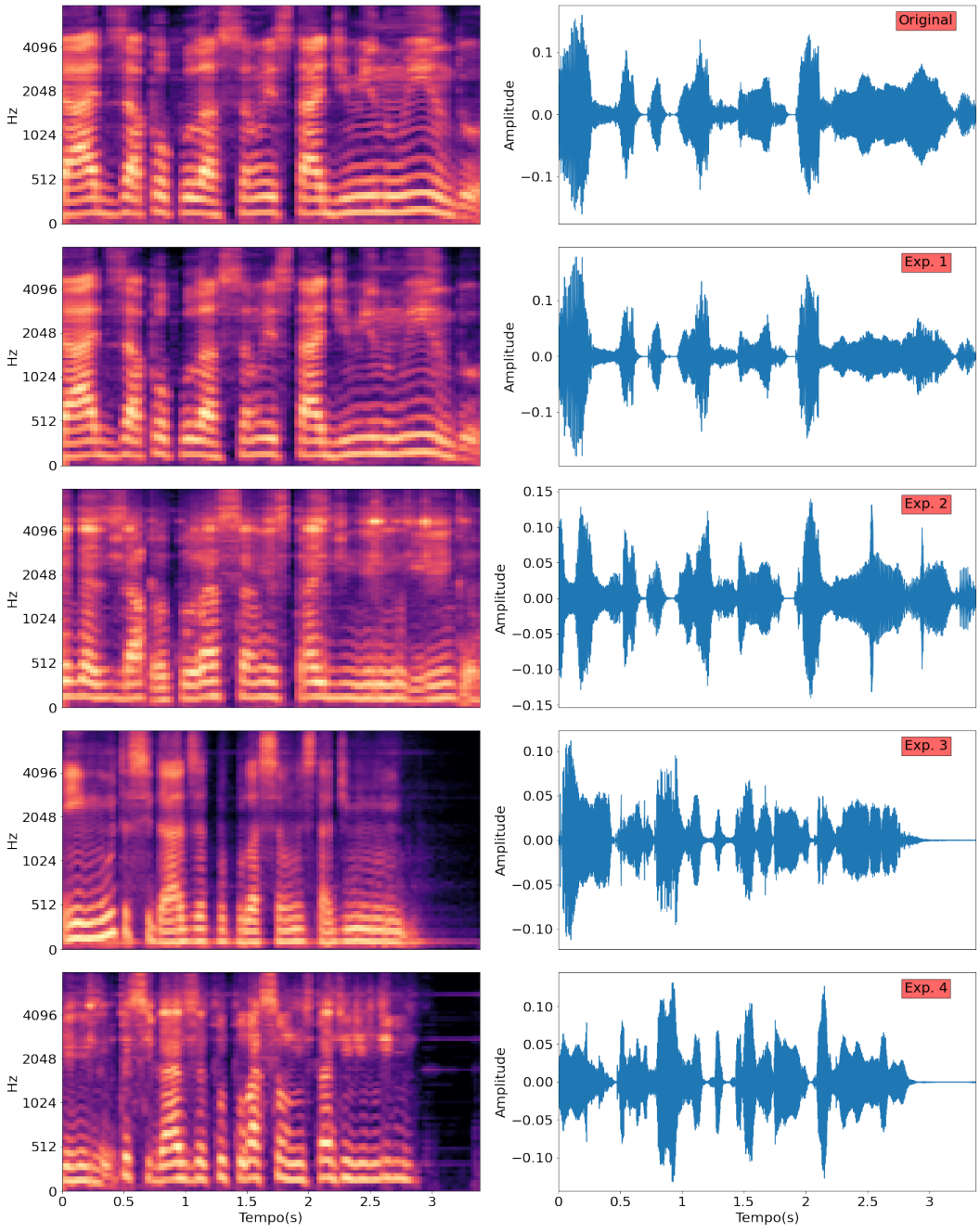

Figura 4.2: Espectrograma e Forma de onda dos sinais para a sentença (b) — "Em muitas cidades a população está diminuindo". De cima para baixo, espectrogramas dos áudios originais e dos Experimentos de 1 a 4, respectivamente.

apresentada aqui, é que o mapeamento de texto para espectrograma é feito de forma mais limpa, portanto vemos a menor presença de ruídos e, portanto, em momentos de silêncio a amplitude da onda chega a zero. Porém, em questão de inteligibilidade e naturalidade do áudio, podemos observar um áudio de boa qualidade, ao se ouvir as amostras geradas, para a maioria das sentenças utilizadas para comparação, demonstrando que a rede neural do Tacotron 2 consegue fazer um bom mapeamento das sentenças para o espectrograma e acredita-se que foi crucial a utilização de um modelo pré-treinado e, em seguida, uma

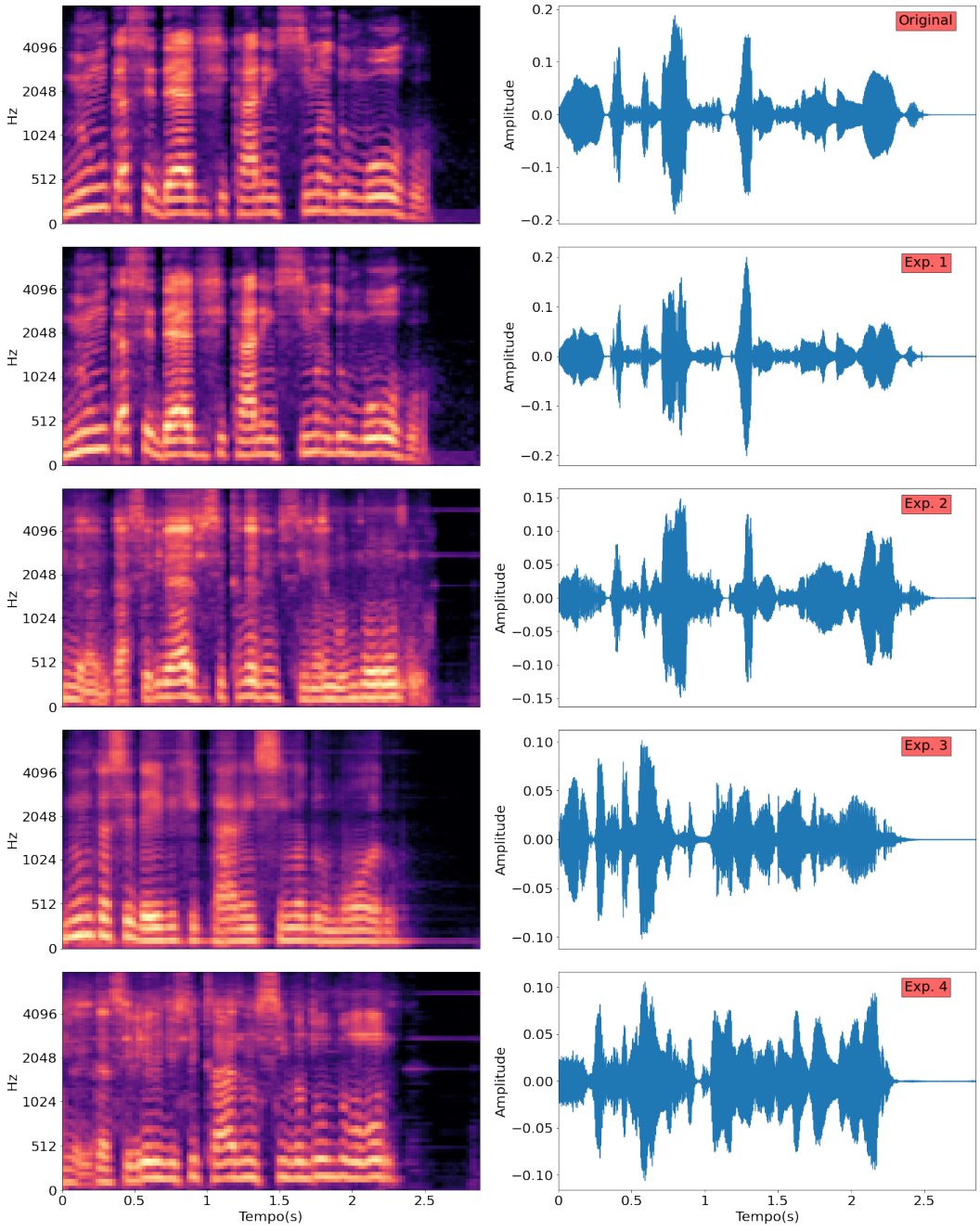

Figura 4.3: Espectrograma e Forma de onda dos sinais para a sentença (c) — "Nunca se deve ficar em cima do morro". De cima para baixo, espectrogramas dos áudios originais e dos Experimentos de 1 a 4, respectivamente.

afinação do modelo treinando na base de dados deste trabalho.

4.4 Tacotron 2 + MelGAN

Neste experimento podemos observar o funcionamento completo do sistema TTS com os modelos treinados neste trabalho, a partir da base de dados na língua portuguesa. Por se tratar de um sistema com dois modelos de inferência, o Tacotron 2 e o MelGAN, espera-

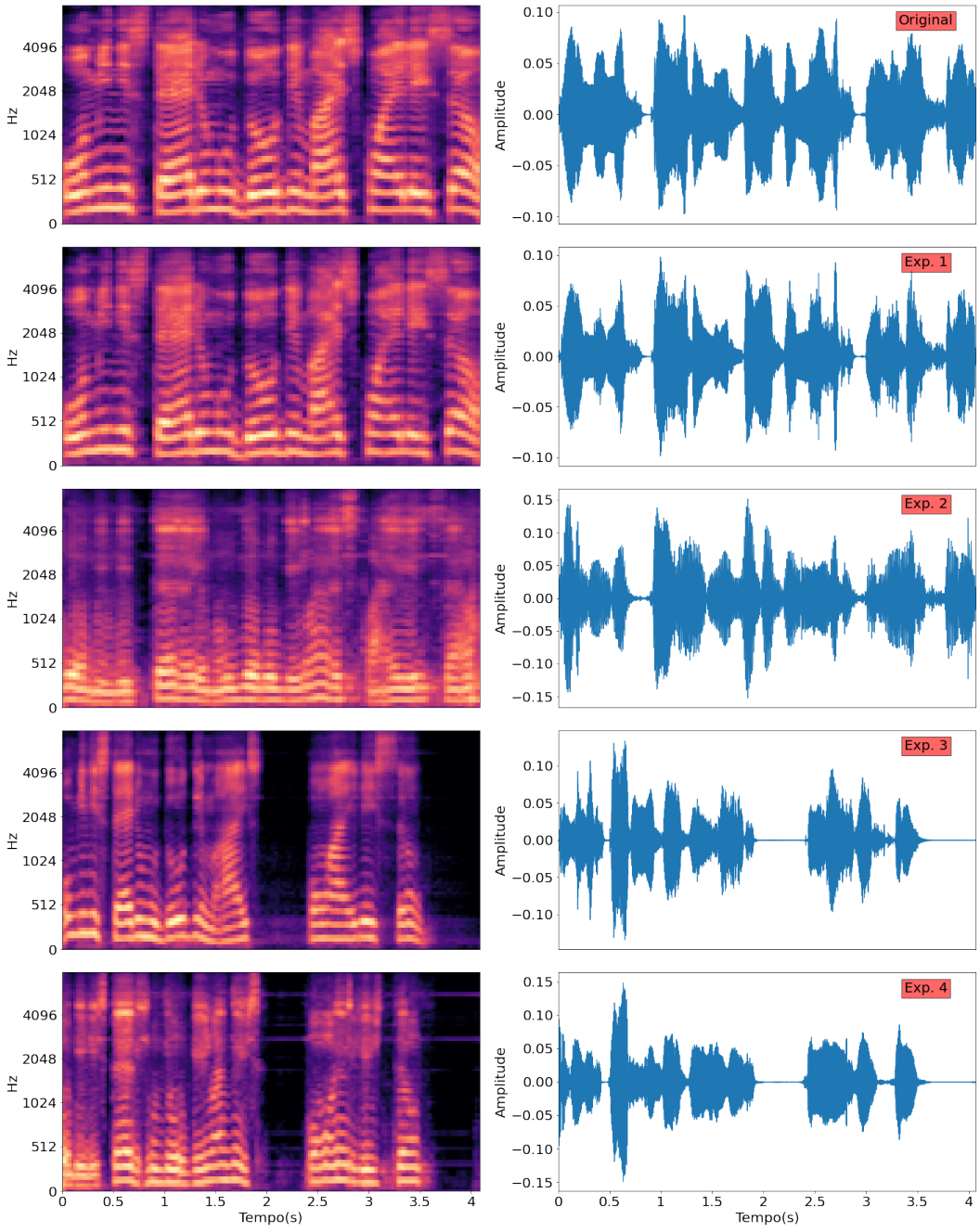

Figura 4.4: Espectrograma e Forma de onda dos sinais para a sentença (d) — "O ministério mudou demais com a eleição". De cima para baixo, espectrogramas dos áudios originais e dos Experimentos de 1 a 4, respectivamente.

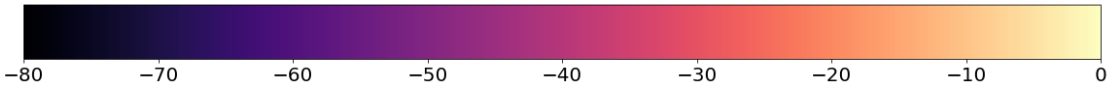

Figura 4.5: Mapa de Cores dos Espectrogramas apresentados, em decibéis (dB).

se que um erro no mapeamento do processamento de texto se propague na inversão do espectrograma.

Os resultados obtidos neste experimento foram áudios de baixa inteligibilidade e na-

turalidade, e que isto pode ser devido, em maior parte à inversão do espectrograma feita pelo MelGAN. Isso pois o Experimento 3 mostrou que um sistema com processamento de texto realizado pelo Tacotron 2 obteve bons resultados e no Experimento 2 observamos que, mesmo com espectrogramas gerados diretamente pelo áudio original, a inversão de espectrograma não foi capaz de gerar bons sinais de áudio.

4.5 Observações Finais

Ao fim dos experimentos, foi possível averiguar a robustez dos modelos pré-treinados disponíveis neste trabalho e que para economia computacional eles devem ser o caminho para treinar outros modelos.

É importante ressaltar que um sistema de TTS para textos em português foi testado à partir de apenas os modelos pré treinados disponíveis do Tacotron 2 e do Waveglow. Apesar de ser capaz de sintetizar as sentenças descritas, com exceção dos caracteres inválidos no alfabeto inglês, alguns fonemas e pronúncias de palavras tinham aspecto não natural, semelhante a quando uma pessoa não falante da língua portuguesa tenta pronunciar sentenças em português. Esse teste serviu para mostrar que, apesar dos modelos pré-treinados apresentarem boa qualidade de síntese, deve-se aproveitar de sua flexibilidade para, partindo destes modelos, treiná-lo com conjuntos de sentenças e áudios em português, para que ele componha um sistema de maior naturalidade.

O modelo MelGAN treinado neste trabalho mostrou que ainda não estava otimizado, gerando áudios com pouca qualidade. Apesar de ter sido o modelo com maior tempo de treinamento (96 horas), por ter sido treinado do zero, não foi o suficiente para atingir a convergência desejada. Este resultado enfatiza ainda mais a necessidade de muitos recursos computacionais para processar o número de iterações necessárias para obter a qualidade de síntese de áudio desejada. Com base nos resultados obtidos no MelGAN, a melhor versão do sistema TTS obtido neste trabalho foi a combinação do Tacotron 2, treinado à partir de um modelo pré-treinado, com o Waveglow, utilizado apenas como referência, pré-treinado na língua inglesa somente. Porém, não pode-se dar certeza de qual a melhor combinação para o sistema, visto que os resultados obtidos no MelGAN podem ser justificados pela falta de mais épocas de treinamento, devido as limitações de recursos computacionais no trabalho.

Capítulo 5

Considerações Finais

5.1 Conclusão

Ao longo deste trabalho, podemos identificar que a importância dos sistemas de TTS vem crescendo, com o aumento de suas aplicações ao longo de diversos setores da indústria, como educação, telecomunicações e questões de acessibilidade. Para acompanhar essa evolução, modelos mais robustos, de redes neurais e orientados a dados, surgiram na última década e hoje temos sistemas capazes de produzir fala à partir do texto com boa qualidade, comparável a áudios originais. Dado este cenário, este trabalho buscou explorar os seguintes tópicos:

- **Recursos necessários:** Com base nos experimentos realizados, vimos que para alcançar resultados próximos aos apresentados nos modelos de referência muito poder computacional e tempo de processamento são necessários, indicando que a tarefa de treinamento de um modelo pode ser muito custosa e limitada de acordo com os recursos disponíveis.

- **TTS na língua portuguesa:** A grande maioria dos sistemas TTS foram projetados e treinados para a língua inglesa, sendo que existem poucos trabalhos na língua portuguesa, além de base de dados menores e mais limitadas. Neste trabalho, procuramos explorar estes modelos desenvolvidos e treinados, inicialmente, para língua inglesa, na língua portuguesa.

- **Modelos pré-treinados:** Uma das soluções encontradas para as limitações de recursos, computacional e de dados, foi a utilização de modelos pré-treinados, e destes é realizado um treinamento mais curto para refinar o sistema. Este artifício permite economizar muito tempo computacional, visto que é necessário um menor número de iterações para obter um modelo bem treinado. Além disso, os modelos pré-treinados podem cobrir algumas falhas em bases de dados menores, tornando o resultado final mais robusto.

5.2 Trabalhos Futuros

Após a execução deste trabalho, foi possível identificar os principais desafios acerca de um sistema de síntese de fala à partir do texto. Com base nisto, uma série de tópicos podem ser abordados futuramente para melhoria e continuação do que foi abordado até agora:

- **Retomar treinamento dos modelos apresentados:** Como já destacado ao longo deste trabalho, os recursos computacionais utilizados eram limitados e isto impossibilitou de continuar a etapa de treinamento das redes neurais apresentadas, mesmo com indícios que estes ainda não atingiram a convergência. Para que isso seja possível, será necessário acesso a recursos mais acessíveis, como máquinas disponibilizadas pela faculdade ou alguma parceria com serviços de computação em nuvem.

- **Explorar o pré-processamento dos dados:** A base de dados utilizada foi gerada à partir de gravações não profissionais, portanto alguns ruídos foram identificados ao longo da base. Para minimizar estes problemas, o autor do banco de dados comenta a utilização de um filtro para remoção do ruído, que não foi abordado neste trabalho. Acredita-se que a utilização de um filtro que possibilite a melhoria da qualidade do áudio possa beneficiar o treinamento dos modelos.

- **Explorar outras arquiteturas para base de comparação:** Ao longo deste trabalho, apesar de se ter estudado outras arquiteturas de redes neurais utilizadas num sistema TTS, apenas 3 arquiteturas foram utilizadas nos experimentos, enquanto que apenas um conjunto de processamento de texto e inversão de espectrograma foram treinados. Para que o estudo consiga comparar de forma mais eficiente as arquiteturas apresentadas nos últimos tempos, seria necessário aplicar o sistema nestas outras redes.

- **Desenvolver uma aplicação de TTS:** As aplicações dos sistemas TTS vem crescendo cada vez mais e ao longo do trabalho apresentamos exemplos de onde pode ser usada, porém não foi desenvolvido nenhuma demonstração de um sistema que integre o TTS, limitando-se apenas a um snippet de inferência com base nos modelos treinados. Num trabalho futuro, para enfatizar a importância destes sistemas como um produto, espera-se desenvolver uma aplicação que se utilize da síntese de fala por texto.

Referências Bibliográficas

[1] P. Taylor, *Text-to-speech synthesis*. Cambridge university press, 2009.

[2] D. Siddhi, J. M. Verghese, and D. Bhavik, "Survey on various methods of text to speech synthesis," *International Journal of Computer Applications*, vol. 165, no. 6, pp. 26–30, 2017.

[3] J. Schroeter, A. Conkie, A. Syrdal, M. Beutnagel, M. Jilka, V. Strom, Y.-J. Kim, H.-G. Kang, and D. Kapilow, "A perspective on the next challenges for tts research," in *Proceedings of 2002 IEEE Workshop on Speech Synthesis, 2002.* IEEE, 2002, pp. 211–214.

[4] B. Kröger and P. Birkholz, "Articulatory synthesis of speech and singing: State of the art and suggestions for future research," vol. 5398, 01 2008, pp. 306–319.

[5] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[6] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 3617–3621.

[7] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," 2018.

[8] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 4779–4783.

[9] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," 2019.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[12] O. Sharir, B. Peleg, and Y. Shoham, "The cost of training nlp models: A concise overview," 2020.

[13] H. Tachibana, K. Uenoyama, and S. Aihara, "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr 2018. [Online]. Available: http://dx.doi.org/10.1109/ICASSP.2018.8461829

[14] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," 2020.

[15] K. Ito and L. Johnson, "The lj speech dataset," https://keithito.com/LJ-Speech-Dataset/, 2017.

[16] W. Ping, K. Peng, and J. Chen, "Clarinet: Parallel wave generation in end-to-end text-to-speech," *arXiv preprint arXiv:1807.07281*, 2018.

[17] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller, "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," *arXiv preprint arXiv:1710.07654*, 2017.

[18] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[19] B. B. Monson, E. J. Hunter, A. J. Lotto, and B. H. Story, "The perceptual significance of high-frequency energy in the human voice," *Frontiers in Psychology*, vol. 5, p. 587, 2014. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpsyg.2014.00587

[20] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, 3rd ed. USA: Prentice Hall Press, 2009.

[21] L. Weng, "Flow-based deep generative models," *lilianweng.github.io/lil-log*, 2018. [Online]. Available: http://lilianweng.github.io/lil-log/2018/10/13/flow-based-deep-generative-models.html

[22] KDNuggets, "Generative adversarial networks – hot topic in machine learning," 2017, [Online; Acessado em 15 de Julho, 2021]. [Online]. Available: https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html

[23] E. Casanova, A. C. Junior, F. S. de Oliveira, C. Shulby, J. P. Teixeira, M. A. Ponti, and S. M. Aluisio, "End-to-end speech synthesis applied to brazilian portuguese," *arXiv preprint arXiv:2005.05144*, 2020.

[24] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden

layers," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013. [Online]. Available: microsoft.com/en-us/research/publication/cross-language-knowledge-transfer-using-multilingual-deep-neural-network-with-shared-hidden-layers/