Universidade de São Paulo
Instituto de Matemática e Estatística
Bacharelado em Ciência da Computação

Uso de redes neurais generativas para síntese de voz

Estudo e revisão de literatura

Lucy Anne de Omena Evangelista

Monografia Final

mac 499 — Trabalho de Formatura Supervisionado

Supervisor: Carlos Eduardo Elmadjan

São Paulo
2022

O conteúdo deste trabalho é publicado sob a licença CC BY 4.0
(Creative Commons Attribution 4.0 International License)

Agradecimentos

Agradeço ao Cadu, meu orientador e amigo, pelo apoio e discussões ao longo deste trabalho; à Patrícia, companheira de discussões, pelas aprendizagens conjuntas e suporte em diversas etapas do trabalho; ao Thales, meu companheiro, por estar sempre comigo, acreditar em mim e incentivar minhas ideias; à Lara, minha amiga, por me motivar e estar presente, sempre; e à minha família por servir de apoio e inspiração.

Resumo

Lucy Anne de Omena Evangelista. **Uso de redes neurais generativas para síntese de voz: *Estudo e revisão de literatura***. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2022.

Resultante a partir de estudos de Interação Humano-Computador (IHC) do projeto ADA (Assistente Distribuída Avançada) do grupo de extensão USP Code Lab do Instituto de Matemática e Estatística da Universidade de São Paulo, o objetivo deste trabalho de conclusão de curso é documentar o processo de escolha, implementação e avaliação de uma rede neural para síntese de voz a partir de texto em linguagem natural para que seja possível a reprodução de testes de IHC para a formulação de uma assistente virtual de código aberto. Desta forma, o trabalho é dividido em introdução do ramo de estudo, aplicações e breve introdução teórica de redes neurais; estudo bibliográfico comparativo entre as arquiteturas para síntese de voz Char2Wav, ClariNet, Tacotron ,Tacotron 2 e Deep Voice 3, realizado a partir dos artigos das arquiteturas, de forma a sistematizar informações quanto a recursos e capacidade das arquitetura; testes de treinamento com a rede mais adequada, partindo-se de implementações em repositórios abertos e uso da técnica de *transfer learning*, de acordo com as limitações de quantidade de dados em português disponíveis para treino; e avaliação dos áudios gerados pela rede, utilizando-se a escala MOS de naturalidade e compreensão. Os áudios produzidos pela rede e o repositório da implementação são disponibilizados ao longo do trabalho. Ao final, são discutidas dificuldades e descobertas ao longo do estudo, e também são sugeridos alguns tipos de informações que podem ser relevantes ao se comparar arquiteturas para síntese de voz e medidas para melhoria no áudio sintetizado.

Palavras-chave: Redes neurais generativas. Síntese de Voz. Processamento de linguagem natural.

Abstract

Lucy Anne de Omena Evangelista. **The use of generative neural networks for voice synthesis: *Study/Research and literature review***. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2022.

Resulting from Human-Computer Interaction (HCI) studies of the ADA (Advanced Distributed Assistant) project of the USP Code Lab extension group of the Institute of Mathematics and Statistics of the University of São Paulo, the objective of this course conclusion work is to document the process of choosing, implementing and evaluating a neural network for speech synthesis from text in natural language so that it is possible to reproduce HCI tests for the formulation of an open source virtual assistant. In this way, the work is divided into an introduction of the branch of study, applications, and a brief theoretical introduction of neural networks; a comparative bibliographical study between the architectures for voice synthesis Char2Wav, ClariNet, Tacotron, Tacotron 2, and Deep Voice 3, carried out from the articles of the architectures, to systematize information regarding the resources and capacity of the architectures; training tests with the most suitable network, starting with implementations in open repositories and using the *transfer learning* technique, under the limitations of the amount of data available in Portuguese for training; and evaluation of the audio generated by the network, using the MOS scale of naturalness and understanding. The produced audios by the network and the implementation repository are made available throughout the work. At the end, difficulties and discoveries throughout the study are discussed, and some types of information that may be relevant when comparing architectures for speech synthesis and measures for improving synthesized audio are also suggested.

Keywords: Generative neural networks. voice synthesis. Natural language processing.

Lista de Abreviaturas

ADA	Assistente Distribuída Avançada
IHC	Interação Humano-Computador
GAN	*Generative Adversarial Network* (Rede Neural Adversarial)
RNN	*Recurrent Neural Network* (Rede Neural Recorrente)
GRU	*Gated Recurrent Unit* (Unidade Recorrente Fechada)
LSTM	*Long Short Term Memory* (Memória de Longo e Curto Prazo, em tradução livre)
GPT	*Generative Pre-Training Transformer* (Transformer Generativo de Pré-Treinamento)
BERT	*Bidirectional Encoder Representations from Transformers* (Representações de codificador bidirecional de Transformers)
MCV	Mozilla Common Voice
IME	Instituto de Matemática e Estatística
USP	Universidade de São Paulo

Lista de Figuras

1.1 Estrutura de um *Autoencoder*, onde os dados entram pelo Encoder, na camada de entrada, são simplificados como o código *h*, reconstruídos no Decoder e o resultado da recontrução sai pela camada de saída. Fonte: Wikimedia Commons 8

3.1 Arquitetura Char2Wav 14
3.2 Arquitetura Multi-Speaker Clarinet 14
3.3 Arquitetura Deepvoice3 15
3.4 Arquitetura Tacotron 16
3.5 Arquitetura Tacotron2 18
3.6 Tabela comparativa das arquiteturas estudadas 19

4.1 Espectrograma Mel da frase "*This is the first test on running synthesized audio for the project Ada*" 23
4.2 Alinhamento entre o decodificador e codificador no áudio gerado para a frase "*This is the first test on running synthesized audio for the project Ada*" 23
4.3 Espectrograma Mel da frase "*The first step on doing Portuguese synthesized audio*" 24
4.4 Alinhamento entre o decodificador e codificador no áudio gerado para a frase "*The first step on doing Portuguese synthesized audio*" 24
4.5 Tabela de conversão de letras com acentuação 26
4.6 Tabela com valores obtidos da escala MOS. 29
4.7 Tabela resultante da avaliação de naturalidade (Q1). 29
4.8 Tabela resultante da avaliação de compreensão (Q2). 29

Sumário

Introdução **1**

1 Breve introdução a redes neurais generativas **3**
1.1 Redes discriminativas e generativas . . . 3
1.2 Redes neurais generativas no mercado . . . 4
1.2.1 Texto escrito e falado . . . 4
1.2.2 Imagem . . . 5
1.3 Elementos de redes neurais generativas . . . 6
1.3.1 Elementos de arquiteturas de redes para síntese de voz . . . 6
1.3.2 Termos específicos de redes que sintetizam voz . . . 7

2 Métodos e Procedimentos **11**

3 Escolha de arquitetura para a síntese de voz **13**
3.1 Arquiteturas estudadas . . . 13
3.1.1 Char2wav . . . 13
3.1.2 Multi-Speaker Clarinet . . . 13
3.1.3 Deep Voice 3 . . . 15
3.1.4 Tacotron . . . 16
3.1.5 Tacotron 2 . . . 17
3.2 Comparação de arquiteturas . . . 17

4 Implementação e avaliação da arquitetura **21**
4.1 Experimentos de treinamento da rede neural . . . 21
4.1.1 Uso de transfer learning . . . 21
4.1.2 Primeiras sintetizações . . . 22
4.1.3 Desafios e testes de pré-processamento de entrada . . . 24
4.1.4 Modificações na entrada da arquitetura escolhida . . . 25
4.1.5 Experimentos do treinamento da rede . . . 26

4.2 Métodos de avaliação da qualidade do áudio gerado 28

5 Considerações e conclusões **31**

Referências **33**

Introdução

Os estudos mostrados neste trabalho começaram a ser desenvolvidos durante a iniciação científica com foco no estudo do ramo de interação humano-computador (IHC), financiada pelo CNPq, número do processo 120151/2019-7, que é resultante do projeto ADA (Assistente Distribuída Avançada) do grupo de extensão USP Code Lab, do IME-USP.

O Projeto ADA tem como objetivo o desenvolvimento de uma aplicação de assistente virtual de código aberto e que utilize ferramentas abertas e acessíveis à toda a sociedade e comunidade acadêmica. Esse objetivo por si só fomenta e incentiva o compartilhamento com a sociedade desse conhecimento específico, que é em sua maior parte desenvolvido por empresas fechadas, pois seu desenvolvimento e estudo é caro devido à necessidade de equipamentos específicos e capacidade de processamento.

A síntese de voz a partir de entradas de texto em linguagem natural é um tema típico de aprendizado de máquina e o conhecimento sobre esse processo e sua implementação é imprescindível na área de Interação Humano-Computador (IHC) para que haja a adaptação ou a criação de novos modelos de interação por voz.

Sendo a iniciação científica sob bolsa PIBITI (Programa Institucional de Bolsas de Iniciação em Desenvolvimento Tecnológico e Inovação), os estudos sobre Interação Humano-Computador foram direcionados a uma aplicação tecnológica prática. Com isso, foram realizadas tanto revisões bibliográficas e estudos sobre o tema como também foi proposta uma forma de elencar e implementar uma rede neural generativa para síntese de voz. A partir da caracterização do que consiste o treinamento de uma rede com essas características, que será abordada neste trabalho de forma mais detalhada, o trabalho objetiva facilitar o desenvolvimento de interfaces por voz para que, a partir dessas, estudos de interação humano-computador possam ser feitos por aspirantes no assunto.

Este trabalho de conclusão tem por objetivo realizar uma revisão mais aprofundada e sistemática de modelos generativos para síntese de voz, de modo a ser um facilitador do tema para todos os interessados, em particular estudantes do ramo de exatas e computação que são falantes de português. Assim, o trabalho se divide em seções que mostram: exemplos de redes neurais generativas populares na atualidade, o que é uma rede generativa e quais aplicações foram desenvolvidas a partir delas, o que é uma rede generativa para síntese de voz e, por fim, de forma mais extensa e com exemplos desenvolvidos durante a iniciação científica, tópicos de implementação e treinamento dessas redes.

Os estudos documentados neste trabalho de conclusão de curso foram gratificantes e

espero que de utilidade para leitores interessados no assunto.

Capítulo 1

Breve introdução a redes neurais generativas

1.1 Redes discriminativas e generativas

O ramo de aprendizagem de máquina é massivamente conhecido pelas suas capacidades discriminativas, ou seja, pela sua capacidade de conseguir classificar (ou discriminar) determinados conjuntos de dados de uma base de conhecimento construída. Tradicionalmente, nos referimos ao problema de aprendizado de máquina como aquele onde queremos aprender uma função específica (hipótese), desconhecida, com base em um conjunto de dados de treinamento e em um conjunto de hipóteses(representações do comportamento que queremos identificar) que julgamos representar o comportamento desejado (Abu-Mostafa, 2012). Dentro desse contexto, queremos encontrar um algoritmo de aprendizagem tal que consiga formular uma função hipótese com base nesses componentes.

Em termos didáticos, essa abordagem é muito efetiva quando pensamos em problemas discriminativos, ou seja, onde nosso espaço de hipóteses representa um comportamento específico a ser identificado pelo nosso algoritmo de aprendizagem dentro do nosso conjunto de exemplos para treino. Bons exemplos dessa aplicação seriam a aferição de um modelo de crédito para grupos populacionais ou ainda a escolha de qual música ou filme determinada população gostaria de ouvir, por exemplo. Problemas de classificação e de regressão são notavelmente o tipo de problema que o aprendizado de máquina sabe resolver, contanto que se use os algoritmos adequados e se possua uma base de dados representativa e genérica. Em termos teóricos, podemos nos referir ao problema como discriminativo quando queremos inferir a probabilidade condicional $p(S|D)$ de um evento acontecer, como por exemplo compreender qual será a probabilidade de uma pessoa sair para o parque (evento S) dado que o dia está ensolarado e a pessoa gosta de praticar atividades físicas (dados de entrada, ou característicos, D). O objetivo é encontrar disjunções nos dados de forma que seja possível identificar algum padrão específico.

Por outro lado, o aprendizado de máquina como conhecemos hoje também possui grandes capacidades generativas que fascinam até pessoas não envolvidas com o ramo da computação. Suas capacidades tomam destaque com a geração de imagens realistas a partir

de textos, com a replicação de estilos artísticos e também com os famosos "*deepfakes*", vídeos gerados por ferramentas que usam aprendizado de máquina para replicar padrões de alguma pessoa específica e recriá-la falando ou fazendo alguma tarefa específica. Discussões sociais sobre criatividade humana e proteções de propriedade intelectual sobre a saída dessas redes surgiram devido a sua proximidade com as capacidades humanas de geração.

Modelos generativos, ao invés dos discriminativos, têm como objetivo entender como um conjunto de dados foi gerado para que, então, possa criar uma saída que represente os dados originais. A função hipótese do problema de aprendizagem é, então, uma função de probabilidade conjunta $p(x, y)$ onde queremos aprender essa função de probabilidade e, para um conjunto de dados de entrada, sermos capazes de retornar a saída provável. De acordo com Patterson e Gibson, 2017, os modelos generativos são normalmente construídos como modelos gráficos probabilísticos que capturam as relações sutis nos dados.

Também, pelas redes generativas não possuírem uma resposta correta concreta para cada problema, ela se utiliza principalmente de algoritmos de aprendizagem não supervisionada. Nesse grupo, três importantes arquiteturas são a base dessas aplicações: Redes Generativas Adversariais (*Generative Adversarial Networks*, ou GANs); Redes de Crenças Profundas (*Deep Belief Networks*) e *Autoencoders* (que pode ser traduzido livremente como "codificadores automáticos").

1.2 Redes neurais generativas no mercado

Atualmente redes generativas têm se tornado cada vez comuns com aplicações sofisticadas em áreas relacionadas a texto, imagem e som. Embora os temas de inteligência artificial e redes neurais estejam mais presentes no nosso dia a dia, a compreensão do que esses termos significam e o que representam não é tão generalizadoTambém, muitas vezes, quando se estuda o campo teórico, é difícil ter a percepção de como é a materialização teórica do mesmo em ferramentas de acesso a empresas ou a população. Assim, esse capítulo tem como objetivo trazer algumas dessas ferramentas, em uso ou sendo desenvolvidas atualmente, para auxiliar na caracterização de redes neurais generativas em relação a redes neurais discriminativas.

Este capítulo traz exemplos sucintos de aplicações para que se possa ter um primeiro contato com ferramentas que utilizam esse tipo de rede. O objetivo é contextualizar o estudo de redes neurais generativas.

1.2.1 Texto escrito e falado

Atualmente as vozes de robôs com os quais interagimos no nosso dia a dia são geradas a partir de redes neurais profundas que são treinadas com muitos dados de frases faladas, quando geradas de processos considerados o estado da arte da área de síntese.

Exemplos de aplicações: No campo de texto temos por exemplo as seguintes redes:

- **GPT-3, do Chat GPT** (https://chat.openai.com): O modelo de linguagem natural da empresa OpenAI é capaz de executar muitas tarefas de geração de texto, incluindo

a geração de texto em idioma natural a partir de um terminal. É um modelo pré treinado e proprietário baseado em Transformers treinado em contextos de 2048 *tokens* contando com mais de 175 bilhões de parâmetros. Transformers são, de forma simples, redes baseadas em mecanismos de atenção (explicados mais a frente no trabalho) combinadas com processamento paralelo de forma eficiente que permite a identificação rápida de relações entre dados sequenciais.

- **GPT-Neo**: implementação similar porém de código livre do GPT-2 e 3 baseado em TensorFlow com diversas configurações e opções específicas de treinamento (https://github.com/EleutherAI/gpt-neo);
- **BERT**: um modelo baseado em Transformers de uma geração anterior para processamento de linguagem natural desenvolvido pela Google;
- **ELMo**: um modelo baseado em redes *Long short term memory* (LSTM) bi-direcionais.

Todos os modelos acima são do tipo *encoder-decoder* e baseiam-se na mesma proposta de conseguir gerar representações intermediária de texto e a partir delas, inferir qual deve ser a continuação das sentenças.

Como esses modelos são feitos para serem genéricos, a aplicação final precisa de uma etapa de treinamento específica e *tuning*(ajuste fino) de parâmetros para que possa ser utilizada com sucesso em outras aplicações.

Entre as aplicações comerciais de geração de texto encontramos por exemplo:

- Geração automática de respostas automáticas para reviews (Replier.ai - https://www.replier.ai/);
- Sistemas conversacionais multi-linguísticos(QuickChat - https://www.quickchat.ai/);
- Geração de conteúdo (Copysmith - https://copysmith.ai/, e copy.ai - https://www.copy.ai/);
- Tradução de código entre linguagens distintas (https://gpt3demo.com/apps/typist-codes);
- Geração de código (Github Copilot - https://github.com/features/copilot);
- Leitura de documentos, via síntese de voz, com muitas opções de voz (**Speechify** - https://speechify.com/)

1.2.2 Imagem

No campo de imagem temosm, por exemplo, as seguintes redes:

- **DALL-E e DALL-E 2**: modelos proprietários da OpenAI capazes de gerar imagens com base em texto. Ele usa uma combinação de modelos de texto (GPT-3) para capturar a semântica do texto passado e de modelos de difusão gaussiana para gerar imagens de acordo com a representação semântica passada.
- ***Stable diffusion***: modelo de código aberto concorrente do DALL-E criado por stability.ai, considerado estado da arte em geração de imagens (https://stability.ai/

blog/stable-diffusion-public-release);

- **DLSS (*Deep learning super sampling*)**: tecnologia da Nvidia capaz de aumentar a resolução de uma imagem em tempo real. Ela usa um rede convolucional auto-encoder sobre a mesma imagem em qualidades diferentes (https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/);
- ***Midjourney***: Companhia que fornece ferramenta para geração de imagens. Foi usada por Json Allen para ganhar um concurso artístico.

Além das aplicações diretas de geração de imagem e melhora de performance em jogos virtuais do DALL-E e do DLSS, diversos modelos baseados em GANs são utilizados para tarefas específicas. GANs, ou redes neurais generativas adversariais, de forma superficial, funcionam com duas redes concorrentes, uma que gera uma saída e outra que tenta classificar a saída como real ou artificial. A ideia é que elas sejam treinadas em conjunto, a ponto de o gerador conseguir gerar imagens cada vez mais realistas e o classificador conseguir distinguir cada vez melhor uma imagem real de uma artificial de forma tal que ambas tenham um desempenho equilibrado.

Entre as aplicações específicas para esse tipo de rede estão:

- Síntese de faces, cujo objetivo é gerar imagens de um mesmo rosto em diversos ângulos (Huang *et al.*, 2017)
- *Inpainting*, cujo objetivo é gerar um segmento de imagem dado os arredores (https://github.com/pathak22/context-encoder)
- Síntese de imagem de alta definição a partir de contornos (https://tcwang0509.github.io/pix2pixHD/)

Note, no entanto, que o DALL-E é capaz de realizar todas essas tarefas e não é baseado em GANs.

1.3 Elementos de redes neurais generativas

1.3.1 Elementos de arquiteturas de redes para síntese de voz

Uma arquitetura neural para síntese de voz é composta por um conjunto de módulos, neurais ou não, que, em conjunto, desempenham a função de aprender a representação sonora e fonética de palavras para que ela consiga, a partir de texto fornecido em linguagem natural, gerar sua representação em voz.

A arquitetura para síntese de voz tem dois módulos de atuação, que são: a etapa de *parsing* ("parseamento", ou análise), que é o mapeamento de voz para fonemas; e a etapa de *rendering* (renderização, ou ainda tradução), que é o mapeamento de fonemas em voz.

A etapa de *parsing* consiste em, dada um par de entrada de texto e voz (uma sentença ou conjunto de dados), extrair informações dessa entrada no processo de *character embedding*, e, a partir de um algoritmo não supervisionado, encontrar padrões de representação desse texto em fonemas. A saída dessa primeira etapa é qualquer representação escolhida

para o fonema aprendido. Em geral, para problemas de síntese de voz, essa saída é um espectrograma.

A etapa de *rendering*, por outro lado, consiste em renderizar uma representação fonética (a entrada dessa etapa) em uma onda sonora que se assemelhe à voz humana. Essa renderização pode ser feita tanto por um algoritmo linear quanto uma rede neural. O elemento que faz essa renderização é chamado de Vocoder no campo de estudos de síntese de voz.

Ambas as etapas são treinadas em conjunto quando se trata de uma arquitetura ponta-a-ponta para síntese de voz, onde todas as etapas são realizadas em conjunto para minimização de erros globais. Quando a arquitetura é treinada e queremos simplesmente realizar a função de sintetizar voz, o texto de entrada é convertido para sua representação fonética aprendida pela rede na etapa de *parsing* e é convertido em som na etapa de *rendering*. Normalmente, um *autoencoder* é utilizado como principal arquitetura para aprendizagem da representação de textos em áudio.

Autoencoder

Autoencoders são são uma classe de algoritmos de aprendizado de máquina do tipo não supervisionado, que funcionam codificando uma entrada em uma representação de baixa dimensão e, em seguida, decodificando a representação para reconstruir a entrada original. A ideia principal desse tipo de algoritmo é conseguir reproduzir a entrada da rede a partir de um conjunto mínimo de *features* (ou características) extraídas da entrada original. O uso de *autoencoders* em problemas generativos se dá justamente pelo fato dela, quando bem treinada, ser capaz de reproduzir padrões aprendidos da base de treinamento.

Os *autoencoders* são compostos por dois módulos, um *encoder* (ou codificador) e um *decoder* (ou decodificador). O *encoder* é responsável por mapear a entrada para uma representação de baixa dimensão, enquanto o *decoder* é responsável por reconstruir a entrada original a partir da representação codificada. Essas duas partes são geralmente implementadas como redes neurais do tipo *feedforward*, ou que não "olham para trás", onde o *encoder* é formado por uma camada de entrada e várias camadas intermediárias, e o *decoder* é formado por várias camadas intermediárias e uma camada de saída.

Além dos problemas generativos, os algoritmos do tipo *autoencoder* também são utilizados para os problemas de detecção de anomalias e redução de dimensionalidade, uma vez que, respectivamente, ela é capaz de identificar variações no conjunto de dados de entrada caso esse se difira muito daquele usado no treinamento da rede, e por reduzir a dimensionalidade dos dados em sua etapa de codificação.

Na figura 1.1 é mostrada uma representação básica da estrutura de um *autoencoder*.

1.3.2 Termos específicos de redes que sintetizam voz

Arquitetura ponta-a-ponta

É dita da arquitetura que é capaz de treinar todos os seus módulos em conjunto, não tratando-os como modelos individuais que serão posteriormente conectados para a geração

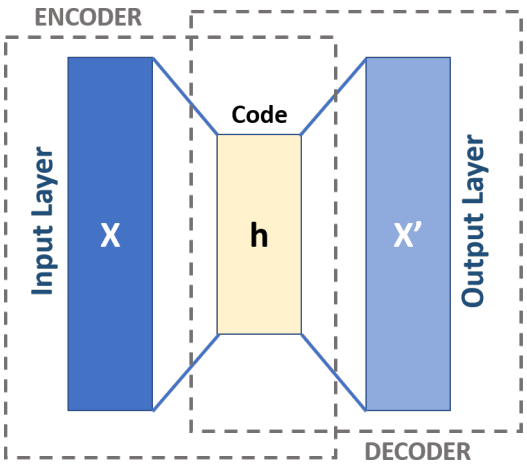


Figura 1.1: *Estrutura de um Autoencoder, onde os dados entram pelo Encoder, na camada de entrada, são simplificados como o código h, reconstruídos no Decoder e o resultado da recontrução sai pela camada de saída. Fonte: Wikimedia Commons*

final do que se pretende com a rede.

Normalmente essas estruturas são mais robustas para permitir essa integração entre partes, as vezes necessitando de ajustes internos ou adição de novos componentes.

Dessa forma as arquiteturas ponta-a-ponta tendem a acumular menos erros em sua cadeia de aprendizado, o que resulta em menos erros acumulados de treinamento, o que aumenta a precisão dos modelos. Para a síntese de voz, essa melhoria é percebida pelo aumento de naturalidade e/ou clareza da voz gerada. Em contrapartida, são redes que demandam maior poder computacional ,por necessitar que todas as usas partes sejam treinadas em conjunto, e também torna difícil a depuração dos módulos da arquitetura, pois os resultados individuais não conseguem ser analisados isoladamente.

Embedding

Etapa da rede neural para síntese que produz *word embeddings* a partir da entrada de texto da rede. *Word embeddings* são representações numéricas de palavras em um espaço multidimensional, usados para representar a semântica e o significado das palavras em um corpus de texto. Cada palavra é mapeada para um vetor único de números, onde cada dimensão representa uma característica ou propriedade da palavra.

Modelo seq2seq (*sequence-to-sequence*)

Conjunto de abordagens no ramo de aprendizado de máquinas responsável por mapear sequências em sequências, como por exemplo mapear palavras a sons, caracteres para vetor e outros. Esse tipo de modelo é utilizado, por exemplo, em traduções, geração de áudio, texto e imagens.

Mecanismos de atenção

Mecanismos de atenção, que permitem a criação de modelos baseados em atenção, são aqueles que possibilitam a identificação de relações ou alinhamentos entre diferentes tipos de estruturas, como, por exemplo, entre *frames* de fala e texto para o caso de síntese de voz. A partir da identificação dessas relações é possível aprimorar a forma com a qual o modelo prediz os fonemas pelo o que chamamos de **contexto**.

De acordo com Luong *et al.*, 2015, a forma como o contexto é calculado a partir das *features* depende do tipo de modelo de atenção, que pode ser global ou local. O modelo global considera todos os estados escondidos de uma rede recorrente para cálculo do contexto, que é a média ponderada de todos os estados. Já o modelo local considera apenas uma janela em torno do objetivo de análise do modelo a cada vez, sendo mais adequando para lidar com longos conjuntos de dados de forma mais rápida. As decisões baseadas por mecanismos de atenção são feitas de forma independente para cada trecho analisado, tanto para o modelo global quanto para o local.

Vocoder

Vocoder é dito o módulo da arquitetura de síntese de voz capaz de sintetizar voz a partir da saída da rede. Alguns dos principais Vocoders encontrados na literatura são: SampleRNN, WaveNet, Griffin-Linn e WORLD.

Espectrograma Mel

Um espectrograma é uma forma de representar uma onda (faixas de frequência) de forma gráfica, seja ela a luz, som ou calor, por exemplo, em função do tempo, amplitude de frequência, intensidade ou outra variáveis relacionada. Para que isso seja possível, todo espectrograma utiliza conceitos de decomposição de ondas para a formulação da imagem, como transformadas de Fourier, onde é possível se decompor uma onda composta em ondas simples. Ondas simples são aquelas harmônicas, que vibram em múltiplos de uma frequência fundamental.

Quando nos referimos ao espectrograma Mel, utilizamos a escala Mel para decompor a onda em questão. A escala mel é uma escala proposta por Stevens, Voklamnn e Newmann sob a afirmação de que o ouvido humano tem maior capacidade de diferenciação de distâncias no tom de um som quando esse possui baixa frequência. Desta forma, a escala Mel realiza um mapeamento das ondas de forma a ser mais compreensivo para o ser humano. Na prática de síntese de voz, quando o modelo gera um espectrograma Mel a qualidade do áudio gerado pela rede é maior, como demonstrado em Wang *et al.*, 2017.

Griffin-Lim

O Griffin-Lim é um algoritmo linear de reconstrução de fase baseado na redundância da Transformada de Fourier de curto prazo. Dado um conjunto de amplitudes e informações de um sinal sonoro, ele é capaz de criar (ou reconstruir) esse sinal. Para este trabalho é importante a compreensão de que ele é um algoritmo simples de conversão de sinal em áudio, usado como versão muitas vezes robotizada de um Vocoder, dado que ele não é

especializado na criação de voz humana.

Capítulo 2

Métodos e Procedimentos

Para definição dos métodos a serem seguidos neste trabalho é necessário relembrar qual é o seu principal objetivo:

- Realizar uma revisão de modelos generativos para síntese de voz, partindo da literatura até a prática.

O que nos leva aos objetivos secundários do trabalho, com base na proposta deste trabalho de conclusão de curso:

1. Encontrar qual e como deve ser feita a escolha e o treinamento de uma rede neural adequada às restrições empíricas de quantidade de dados e poder de processamento disponível;
2. Criar uma rede neural que sintetize falas em português a partir de entradas em inglês para que seja possível realizar testes de Interação Humano Computador para servir ao projeto ADA do USP Code Lab.
3. Avaliar a rede neural treinada.

Assim, o procedimento para realização deste trabalho envolve três etapas: revisão de literatura, aplicação de técnicas e tecnologias estudadas e avaliação do que foi desenvolvido. Sobre a revisão de literatura, ele foi feita com os textos base Abu-Mostafa, 2012, Patterson e Gibson, 2017 e Goodfellow *et al.*, 2016, e com artigos das arquiteturas para síntese de voz a seguir:

- Clarinet (Ping, Peng e J. Chen, 2018)
- Char2Wav (Sotelo *et al.*, 2017)
- Tacotron (Wang *et al.*, 2017)
- Tacotron 2 (Shen *et al.*, 2018)
- Deep Voice 3 (Ping, Peng, Gibiansky *et al.*, 2017)

Uma arquitetura-base foi escolhida com base na literatura e foi implementada com base em repositórios abertos encontrados em comunidades de compartilhamento de código. Com a rede disponível, foram realizados diferentes testes de treinamento dos pesos dessa

para que, ao fim, fosse eleita uma versão para a realização da avaliação. Para a avaliação da qualidade dos áudios sintetizados, é utilizada a escala MOS, a ser descrita com mais detalhes na seção 4.2.

Capítulo 3

Escolha de arquitetura para a síntese de voz

3.1 Arquiteturas estudadas

3.1.1 Char2wav

O grande diferencial desta arquitetura ponta a ponta é a sua composição, que consiste emum leitor ("*Reader*") e um vocoder neurais. O *Reader* comporta a estrutura *Encoder-Decoder*. O *encoder* é uma rede neural recorrente bidirecional, o que permite a utilização de informação temporal para o ajuste da rede e o *decoder* é uma rede neural recorrente com atenção. Justamente pela característica do *Reader* é que se é possível gerar áudio a partir de texto de forma eficiente, sendo ele em texto padrão ou em fonemas. O diagrama da arquitetura pode ser visto na figura 3.1.

Neste modelo o Vocoder possui ação destacada por ser uma rede SampleRNN, capaz de modelar dependências extremamente longas em dados sequenciais (Sotelo *et al.*, 2017), percebendo dinâmicas de modulação de acordo com a entrada fornecida. Desta forma, o Vocoder se torna parte essencial do modelo e não poderia ser trocado por qualquer outro Vocoder.

Conjunto de dados utilizados: VCTK (inglês) e DIMEX-100 (espanhol) (Sotelo *et al.*, 2017).

3.1.2 Multi-Speaker Clarinet

ClariNet é a primeira arquitetura neural de texto-para-onda (*Text-to-Wave – TTW*) feita para síntese de texto. No artigo de Ping, Peng e J. Chen, 2018, é proposta uma arquitetura melhorada que permite a criação de um sistema para múltiplos locutores. Cada componente da rede recebe como viés de treinamento uma incorporação treinável de locutor com poucas dimensões. Essa adição permite que a precisão de fala seja alcançada para diversos locutores diferentes, o que é reforçado pelo fato de se conseguir inferir gênero e sotaque dos mesmos após o processo do Multi-Speaker ClariNet. Por este mesmo motivo, é citado que seu uso pode ser feito com o objetivo de clonagem de voz.

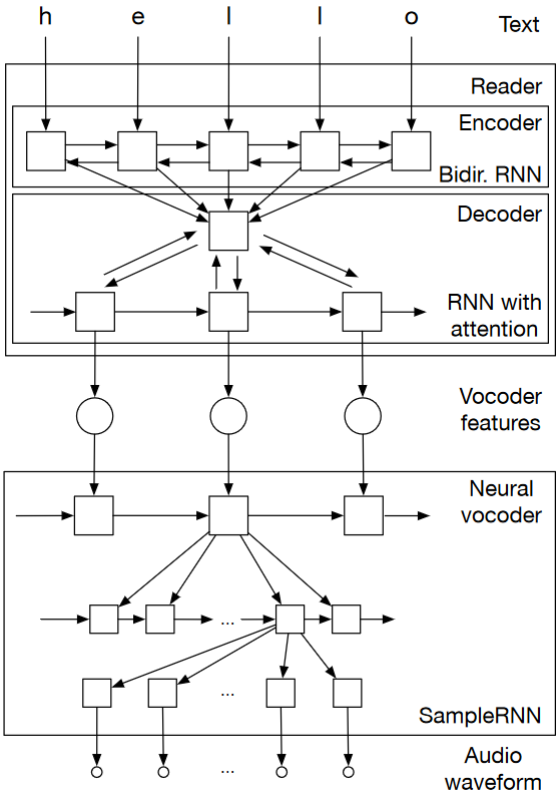
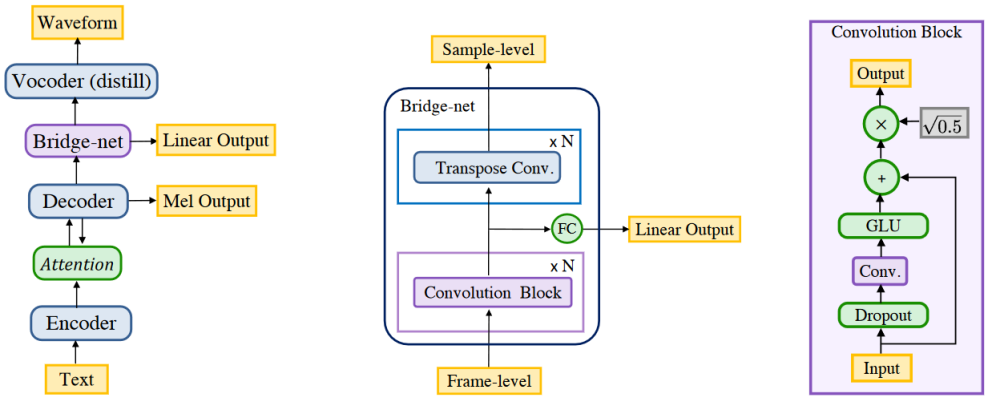


Figura 3.1: *Arquitetura Char2Wav*

A região do *Decoder-Encoder* e o Vocoder são conectadas a partir da Bridge-Net, como mostra a figura 3.2, que é um bloco de processamento convolucional não-causal que permite a geração de conteúdo em áreas desconhecidas, gerando amostras internas que auxiliam no processo de geração de voz. Possui como Vocoder o modelo generativo WaveNet autor-regressivo Gaussiano, que possui menos camadas de processamento e mesma estrutura para o par *Encoder-Decoder* utilizado no ClariNet e no Deep Voice 3.

Conjunto de dados utilizado: VCTK (inglês), sub amostrado entre as frequências de 48 a 24Hz para treino (Ping, Peng e J. Chen, 2018).



(a) Text-to-wave architecture (b) Bridge-net (c) Convolution block

Figura 3.2: *Arquitetura Multi-Speaker Clarinet*

3.1.3 Deep Voice 3

O Deep Voice 3 é uma arquitetura baseada em redes neurais seq2seq totalmente convolucionais com mecanismo de atenção de posição aumentada (*position-augmented attention mechanism*). O objetivo do Deep Voice 3 (Ping, Peng, Gibiansky *et al.*, 2017) é o de escalar o treinamento para síntese de voz para diversos locutores, permitindo treinamento rápido de diversos conjuntos de dados a partir do uso massivo de processamento paralelo, que é possível devido à sua arquitetura. Mesmo que não esteja no escopo deste trabalho, é válido mencionar que o desenvolvimento realizado com o Deep Voice 3 foi divulgado na mesma época do artigo que é tido como o marco-zero de Transformers (Li *et al.*, 2019), com poucos meses de diferença, e suas abordagens são semelhantes.

Sua arquitetura, em comparação com os modelos até então citados, evita a utilização de redes neurais recorrentes (RNN), com o objetivo de tornar o treinamento mais rápido. Em compensação, utiliza mecanismos de atenção em todas as etapas da síntese de texto, desde o codificador até a síntese de voz a partir de espectrogramas Mel ou outras formas, e necessita de uma grande quantidade de dados tratados para treino de diversos locutores. Esta etapa é essencial para a precisão do modelo.

O codificador é um modelo totalmente convolucional com vetores chave de atenção; o decodificador é um modelo totalmente convolucional causal que utiliza mecanismo de atenção convolucional de múltiplos saltos (*multi-hop convolutional attention mechanism*), e seu conversor de voz é uma rede totalmente convolucional de pós-processamento não causal e não autorregressiva, podendo-se utilizar de diversos tipos de Vocoder (Griffin-Lim, WaveNet, WORLD). Entre o codificador e decodificador também há um bloco de atenção do tipo *dot-product*, que possui uma progressão monotômica no tempo, ou seja, que é estritamente crescente ou decrescente no tempo. A arquitetura dessa rede pode ser vista na figura 3.3.

A melhor combinação para síntese de voz da arquiteturaé utilizando-se espectogramas mel e WaveNet como Vocoder.

Conjunto de dados utilizados: VCTK (inglês) e LibriSpeech (inglês) (Ping, Peng, Gibiansky *et al.*, 2017).

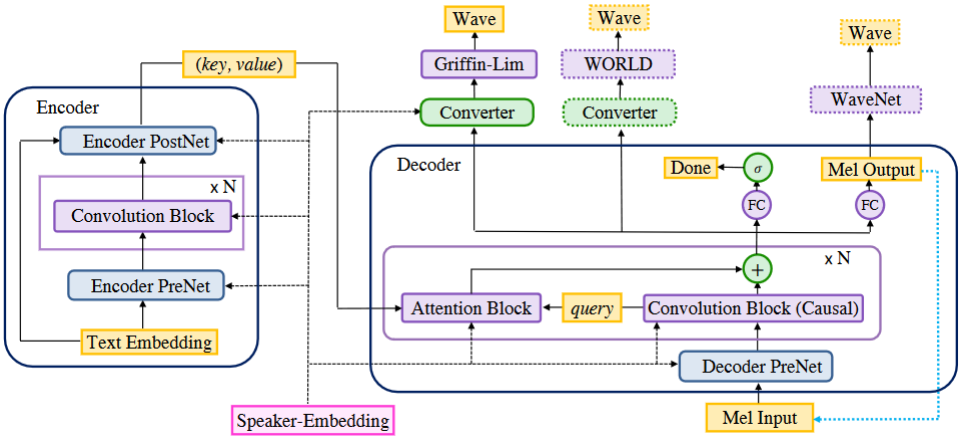


Figura 3.3: *Arquitetura Deepvoice3*

3.1.4 Tacotron

A arquitetura Tacotron é um modelo generativo ponta a ponta com paradigma de atenção que sintetiza fala a partir de caracteres e, assim como a Char2Wav, se baseia em um modelo sequência-para-sequência (seq2seq). Um dos maiores diferenciais do Tacotron é a não necessidade de tratamento dos dados muito extensa para realização do treinamento, diferente de outras que precisam realizar o pré-processamento de fonemas ou também que requerem aumento ou redução da frequência dos áudios de entrada; e também é baseado em *frames*, o que possibilita que o treinamento seja relativamente mais rápido do que métodos *autorregressivos* em nível de amostra.

destacada, por ser um modelo diferente das redes recorrentes tradicionais, como LSTMs

Sua arquitetura é composta por um codificador, um decodificador baseado em atenção e uma rede de pós-processamento que é responsável por processar a saída da rede anterior para que seja possível a sintetização de voz. Para a síntese, é utilizado o algoritmo Griffin-Lim. O módulo CBHG (*Convolutional Bank + Highway Network*) no codificador da arquitetura possui atuação destacada, por ser um modelo diferente das redes recorrentes tradicionais, como LSTMs. Composto por filtros convolucionais, redes *Highway* (um tipo de arquitetura para redes neurais profundas) e um módulo GRU RNN (módulo que funciona como uma LSTM simplificada, com menor quantidade de parâmetros), o CHBG é capaz de reduzir o *overfitting* da rede e quantidade de pronúncias incorretas.

O diagrama da arquitetura pode ser visto na figura 3.4.

Conjunto de dados utilizado: Interno da Google de inglês norte americano (Wang *et al.*, 2017).

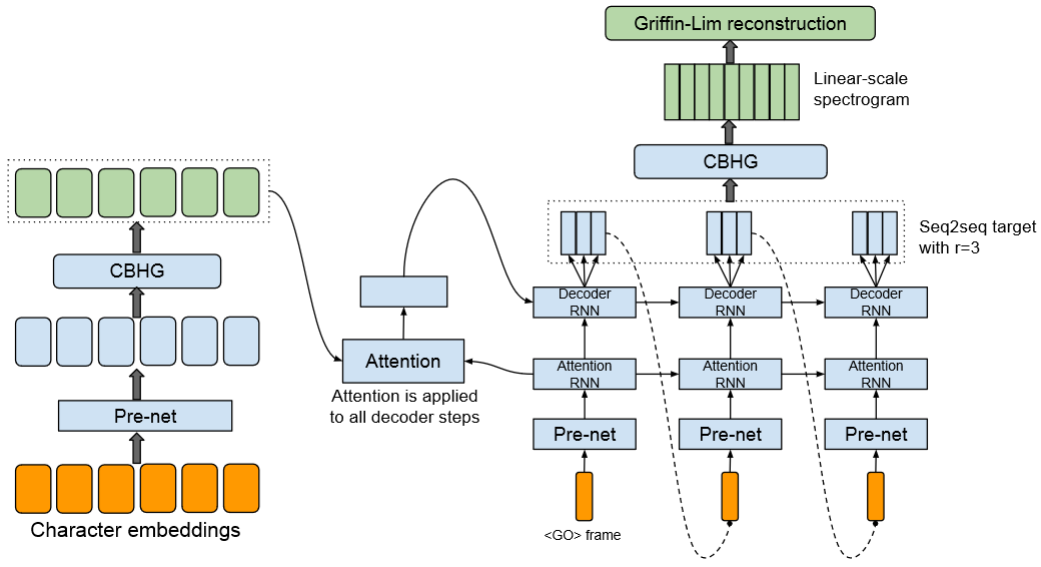


Figura 3.4: *Arquitetura Tacotron*

3.1.5 Tacotron 2

O Tacotron2 possui as mesmas características do Tacotron (Wang *et al.*, 2017) quanto a sua base, ou seja, também é um modelo composto por redes recorrentes do tipo seq2seq. As diferenças e melhorias do Tacotron 2 em relação ao seu anterior são, respectivamente: o mapeamento de agrupamentos de caracteres para espectrogramas Mel como entrada para a síntese de voz e alteração do Vocoder com base no algoritmo Griffin-Lim para a utilização do modelo WaveNet para síntese de voz. Dessa forma, Tacotron2 passa a ter uma estrutura totalmente baseada em redes neurais ponta a ponta.

Sua arquitetura é composta por uma rede recorrente do tipo seq2seq com atenção (tanto a parte do codificador quanto decodificador baseados em LSTM) responsável por gerar uma sequência de espectrogramas Mel, e por uma versão modificada de WaveNet, que é um sintetizador de áudio capaz de gerar voz a partir de diferentes entradas (linear, espectrogramas de fase ou espectrogramas Mel, por exemplo). De acordo com Shen *et al.*, 2018, os espectrogramas Mel representam a voz de forma mais suave do que amostras padrão de formas de onda, sendo também mais simples o treino da rede a partir deles. Sendo gerado a partir de uma Transformada de Fourier de curto período (STFT), os espectrogramas Mel enfatizam detalhes a curto alcance e diminuem o impacto de detalhes em alta frequência.

A arquitetura também possui uma pré-rede de processamento entre o codificador e decodificador baseada em unidades ReLU (Unidades Lineares Retificadas), que filtram as informações essenciais. Também possui uma pós-rede depois do decodificador, que é responsável por predizer resíduos adicionais aos espectrogramas Mel, o que melhora a construção geral do áudio a partir do texto. As partes da rede, de predição e síntese, são treinadas separadamente, utilizando texto normalizado, ou seja, com numerais e afins escritos por extenso.

Conjunto de dados utilizado: Interno da Google de Inglês Norte Americano, mesmo para o Tacotron (Wang *et al.*, 2017).

3.2 Comparação de arquiteturas

Foi realizado um levantamento de critérios de seleção entre as arquiteturas pesquisadas para que se pudesse, a partir de informações qualitativas, definir qual das arquiteturas seria implementada. Assim, foi realizado um estudo comparativo das arquiteturas, feito com base nas informações disponibilizadas nos artigos de cada uma.

Sugestões de parâmetros: o volume de dados utilizados e seu perfil, quantidade de máquinas utilizadas, *batch size* e tempo médio de treino (ou ainda definição de iterações ou passos do modelo)

Os parâmetros para comparação das arquiteturas são:

- Utilização de WaveNet (rede de alto desempenho) como Vocoder no treinamento da rede;
- Necessidade de ser realizado pré-processamento nos dados;

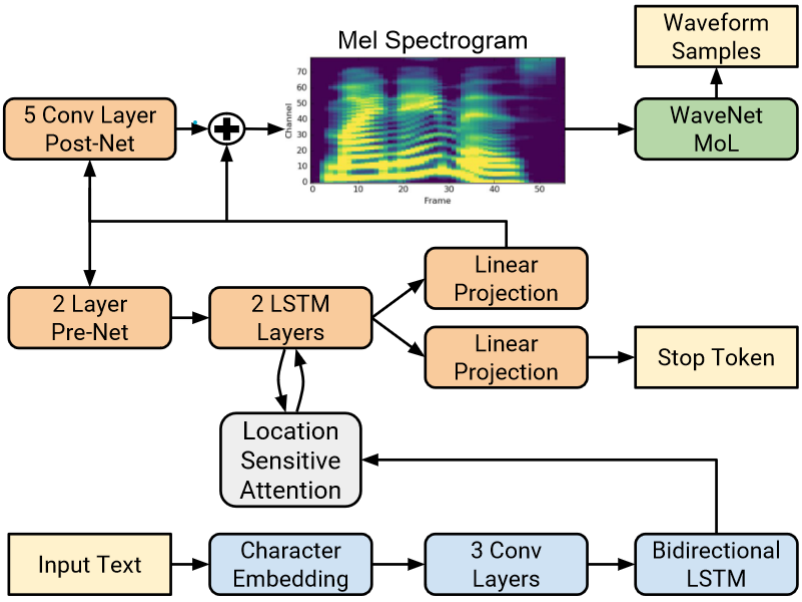


Figura 3.5: *Arquitetura Tacotron2*

- Estrutura da rede em sua disposição dos módulos de atenção;
- Avaliação na escala MOS, de naturalidade e compreensão baseada em avaliações subjetivas; e
- Existência implementações e repositórios abertos, oficiais ou não, que tenham caráter reprodutível.

A materialização do estudo comparativo se encontra na figura 3.6. Nela há também outras informações acerca das arquiteturas, além das elencadas como pontos chave para decisão, por serem encontradas em comum nos artigos.

Ao longo do estudo, encontrou-se que a utilização de espectrogramas Mel torna a síntese de voz mais eficiente (Ping, Peng, Gibiansky *et al.*, 2017 e Shen *et al.*, 2018) e que a utilização de WaveNet como Vocoder melhora consideravelmente a naturalidade da voz gerada, devido aos módulos de atenção em sua arquitetura (Sotelo *et al.*, 2017, Ping, Peng, Gibiansky *et al.*, 2017 e Shen *et al.*, 2018).

Como principais arquiteturas para a utilização no projeto foram selecionadas a Tacotron 2 e a DeepVoice 3 para comparação. Isso devido a qualidade dos áudios gerados, que podem ser verificados nos *links* presentes na tabela comparativa; pela quantidade e eficiência de uso dos módulos de atenção em suas arquiteturas; e por se mostrarem as arquiteturas de ponta mais recentes no mercado.

As arquiteturas Tacotron e DeepVoice3 foram publicadas ambas em 2018, com poucos dias de diferença. Tacotron 2 é mais rápida para treinamento em relação à outras arquiteturas, com exceção do DeepVoice, e consegue se especializar com bom desempenho para um único falante. A disposição dos seus módulos de atenção e redes de pré e pós-processamento melhoram consideravelmente a naturalidade do áudio gerado (Ping, Peng, Gibiansky *et al.*, 2017).

	Clarinet Modificado Ping, Peng e J. Chen, 2018	Char2Wav Sotelo *et al.*, 2017	Tacotron Wang *et al.*, 2017	Tacotron 2 Shen *et al.*, 2018	Deep Voice 3 Ping, Peng, Gibiansky *et al.*, 2017
Data de Lançamento	09/07/2019	15/04/2017	06/04/2017	16/02/2018	22/02/2018
Vocoder	Gaussian Autoregressive	SampleRNN	Griffin-Lim	WaveNet modificada	Griffin-Lim, WORLD & WaveNet
Pré-processamento é necessário	Sim.	-	Não.	Não.	Sim.
Normalização é necessária	Sim.	-	Sim.	Sim.	Sim.
Estrutura da rede	Encoder - Como em Ping, Peng, Gibiansky *et al.*, 2017. Decoder - Como em Ping, Peng, Gibiansky *et al.*, 2017. BridgeNet - Blocos convolucionais. Vocoder	Encoder - BRNN Decoder - Com atenção, RNN Vocoder	Encoder - PreNet & CBHG; Decoder - com atenção, Prenet & GRU RNN; Post Processing Net - CBHG Vocoder	Encoder - PreNet & bidirectional LSTM; Decoder - com atenção, Prenet & LSTM & camadas convolucionais; Post Processing Net - blocos convolucionais Vocoder	Encoder - com atenção, convolucional com speaker embedding Decoder - com atenção & PreNet, com speaker embedding, autorregressivo convolucional Converter - rede convolucional para pós processamento Vocoder
MOS (pontuação (vocoder) [Dataset - papel])	20 layers - 3.75 ± 0.42 30 layers - 3.9 ± 0.36 40 layers - 3.89 ± 0.28	-	3.82 ± 0.085 4.001 ± 0.087 Shen *et al.*, 2018 2.07 ± 0.31 [VCTK] Ping, Peng, Gibiansky *et al.*, 2017 3.78 ± 0.34 (WaveNet) Ping, Peng, Gibiansky *et al.*, 2017	4.526 ± 0.066 4.429 ± 0.071 4.148 ± 0.124 4.354 Ping, Peng, Gibiansky *et al.*, 2017	3.78 ± 0.3 [Interno] 3.44 ± 0.32 (WORLD) [VCTK] 3.01 ± 0.29 (Griffin-Lim)[VCTK]
Base de Dados	VCTK (inglês)	Dimex-100 (espanhol), VCTK (inglês), Blizzard (inglês), e Pavoque (alemão)	Interno, de inglês americano	Interno, de inglês americano	Interno, de inglês norte americano, VCTK (inglês) e LibriSpeech (inglês)
Hiperparâmetros disponíveis	Sim, no artigo.	Sim, no repositório.	Sim, no artigo.	Sim, no repositório.	Sim, no artigo.
Repositório aberto	Não.	Sim. https://github.com /sotelo/parrot	Sim. https://github.com/ keithito/ tacotron (não oficial)	Sim. https://github.com/ Rayhane-maman/ Tacotron-2 (não oficial)	Sim. https://github.com/ r9y9/deepvoice3_pytorch (não oficial)
Audio Samples disponibilizados	Sim. link1:https://multi-sp eaker-clarinet-demo.github.io	Sim. https://josesotelo .com/speechsynthesis	Sim. https://google.github .io/tacotron/publications/ tacotron/index.html	Sim. https://google.github .io/tacotron/publications/ tacotron2/	Sim. http://research.baidu. com/Blog/index-view?id=91
Ponto de atenção	Há speaker empedding em todas as partes da arquiteturas	-	Prediz espectrogramas lineares.	Prediz espectrogramas Mel. As redes são treinadas separadamente	Não possui módulos recorrentes, em compensação há muitos módulos de atenção. Precisa de grande quantidade de dados, mas é mais rápido
Treinamento	Adam Optimizer com 1.5M passos, batch size de 16.	-	Adam Optimizer 2M passos, batch size de 32.	Adam Optimizer com taxa de aprendizagem de 10^{-4}, batch size de 64.	Converge após 500k passos.

Figura 3.6: *Tabela comparativa das arquiteturas estudadas*

DeepVoice3 possui um rápido treinamento devido ao seu processamento paralelo e ausência de módulos recorrentes, podendo ser treinada para múltiplos locutores e também oferece a flexibilidade de alterar o seu Vocoder de acordo com a necessidade.

Decisão de arquitetura

A arquitetura Tacotron 2 foi escolhida de acordo com os critérios mencionados, e levando em conta também sua adequação (i.e., um gerador de voz única, com o maior índice de naturalidade possível) para futuros testes de IHC. Os pontos mais relevantes para a escolha foi a quantidade relativamente menor de dados necessários para a obtenção de um resultado satisfatório de naturalidade do áudio gerado. Por satisfatório, entenda-se do mesmo nível, ou muito próximo, do valor da escala MOS para os áudios gerados em relação às outras arquiteturas que necessitam maior quantidade de dados para treino, como mencionado pelos conjuntos de dados usados em seus respectivos artigos.

A revisão das arquiteturas para síntese de voz e estudo comparativo realizado serviram como base para desenvolvimento do artigo "Uma revisão de arquiteturas ponta a ponta para síntese de voz", apresentado no ERAMIA 2020[1]. O estudo traz a tabela comparativa

[1] *https://eradsp2020.ncc.unesp.br/aprovados* e *https://sol.sbc.org.br/index.php/eramiasp/article/view/13612*

desenvolvida.

Capítulo 4

Implementação e avaliação da arquitetura

4.1 Experimentos de treinamento da rede neural

Como forma de compreender aspectos técnicos da arquitetura Tacotron 2, foi reproduzida a implementação de um repositório aberto da mesma em TensorFlow (https://github.com/Rayhane-mamah/Tacotron-2) utilizando-se um modelo pré-treinado (https://r9y9.github.io/wavenet_vocoder/). Inicialmente, o repositório foi criado localmente no servidor interno do projeto ADA. Ao final das alterações documentadas neste trabalho o repositório foi migrado para o GitLab do projeto (https://gitlab.com/advanceddistributedassistant/ihc-tacotron-2).

Nessa estrutura inicial do repositório aberto, o Vocoder utilizado é o Griffin-Lim. Isso porque, para se utilizar o WaveNet, há a necessidade de se treinar essa rede ou de se utilizar outro modelo pré-treinado para essa parte. Assim, o modelo implementado é um híbrido entre as arquiteturas completas do Tacotron e do Tacotron 2, onde todas as melhorias do Tacotron 2 são observadas com exceção do Vocoder mais sofisticado.

A principal dificuldade para utilização do modelo pré-treinado foi a adequação da versão do TensorFlow ao projeto, que deve ser a 1.7 pois a utilização de outra versão mais atual leva à não identificação de módulos do TensorFlow.

4.1.1 Uso de transfer learning

Devido às restrições do projeto, que são a pouca quantidade de dados em português e a inexistência de modelos treinados em português que sejam de implementação aberta, optamos por aplicar a técnica de transferência de aprendizagem, denominada *Transfer Learning*. Nessa técnica utilizamos um modelo base genérico e com bom poder de generalização para aplicar em um contexto específico de treinamento. Isso é possível somente ao partirmos da hipótese de que o modelo base é mais genérico (ou seja, possui maior capacidade de generalização) do que se iniciar o treinamento da rede do zero.

Tomando como hipótese a de que o modelo em inglês é esse modelo base, optamos

por usar *Transfer Learning* tanto pela proximidade de fonemas em comum entre a língua inglesa e a língua portuguesa, quanto pela menor capacidade computacional e quantidade de dados necessária para que haja convergência do modelo.

É importante destacar também que mesmo com esses benefícios, o modelo pode não ter uma boa aprendizagem caso algumas das *features* dos novos dados sejam muito distintas do modelo base. Para o nosso caso, há essa possibilidade devido a diferença de tamanhos entre os alfabetos inglês e português, gerando alguma incompatibilidade de entradas.

Vale destacar, entretanto, que o uso de um modelo base em espanhol talvez fosse mais recomendado, mas não foi encontrado um que foi treinado com tantos dados quanto o em inglês.

4.1.2 Primeiras sintetizações

A implementação do modelo em repositório aberto por si só foi um desafio devido à incompatibilidade de versões como mencionado anteriormente. Uma vez esses problemas transpostos, foi utilizado o *checkpoint* da rede pré-treinada. O *checkpoint* foi treinado até 189.000 passos com a base de dados LJSpeech, de um único falante em inglês.

O resultado da saída da rede da estrutura anterior ao Vocoder pode ser visualizada nas figuras de 4.1 à 4.4. As figuras 4.1 e 4.3 são os espectrogramas Mel para as frases "*This is the first test on running synthesized audio for the project Ada*" e "*The first step on doing Portuguese synthesized audio*" e as figuras 4.2 e 4.4 são os diagramas de alinhamento entre os tempos gerados nos módulos codificador e decodificador (quanto mais alinhado, maior a clareza do áudio gerado).

Os áudios gerados podem ser encontrados abaixo:

- "This is the first test on running synthesized audio for the project Ada" [1];
- "The first step on doing Portuguese synthesized audio"[2].

[1] https://bit.ly/ada_first

[2] https://bit.ly/ada_second

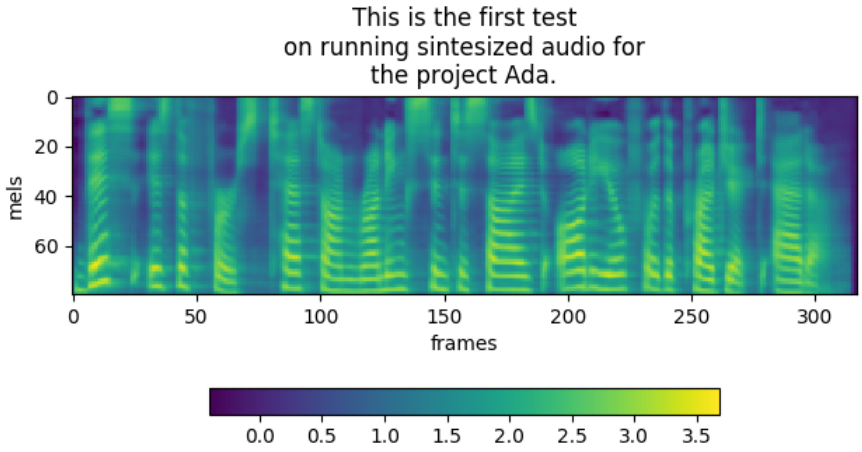


Figura 4.1: *Espectrograma Mel da frase "This is the first test on running synthesized audio for the project Ada"*

A partir deste modelo pré-treinado em inglês vamos realizar experimentos de treinamento para adaptação da rede em língua portuguesa do Brasil, na seção 4.1.5.

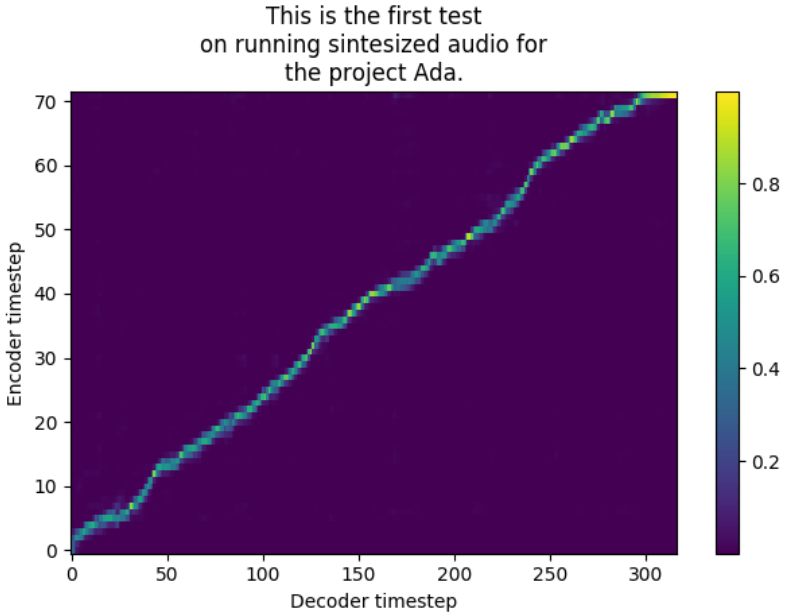


Figura 4.2: *Alinhamento entre o decodificador e codificador no áudio gerado para a frase "This is the first test on running synthesized audio for the project Ada"*

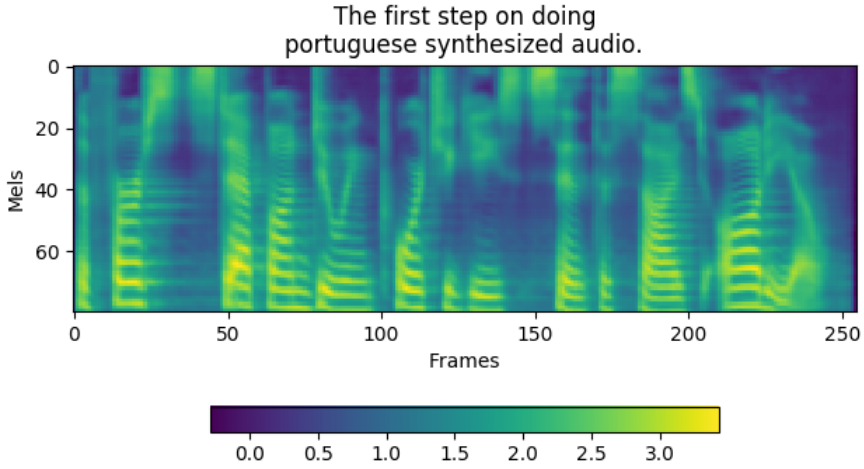


Figura 4.3: *Espectrograma Mel da frase "The first step on doing Portuguese synthesized audio"*

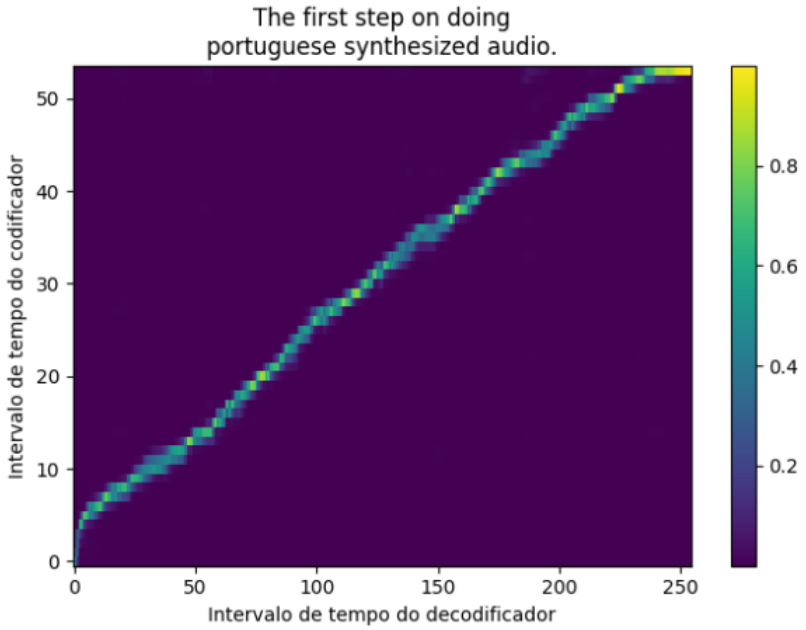


Figura 4.4: *Alinhamento entre o decodificador e codificador no áudio gerado para a frase "The first step on doing Portuguese synthesized audio"*

4.1.3 Desafios e testes de pré-processamento de entrada

Nesta seção serão explorados alguns desafios e testes de pré-processamento de entrada realizados para a síntese de áudio em português. A maior problemática encontrada ao lidar com a diferença entre os treinos para a língua inglesa e língua portuguesa é a acentuação de palavras, que é muito rica na língua portuguesa em relação à inglesa.

Foram discutidas possíveis soluções para a adequação do treino para a língua portuguesa

com base em uma mudança nos componentes da rede ou também com base no pré-processamento dos dados de entrada para a rede.

É sabido que para arquiteturas tradicionais de síntese de voz o pré-processamento ou transcrição da base de dados para o alfabeto fonético (IPA - *International Phonetical Alphabet*) é comum e muito utilizado, e importante quando se trata da transcrição fonética ou adaptação entre idiomas. Para arquiteturas ponta-a-ponta isso teoricamente não é necessário, dado que a partir da rede do tipo *Autoencoder* e dos mecanismos de atenção é possível estabelecer a interação fonética entre elementos, o que permite que diferenças de pontuação e/ou formatação do caractere (como palavras em letras maiúsculas ou minúsculas) sejam percebidas e corretamente sintetizadas para o idioma de entrada.

Até então, neste projeto, esse comportamento foi verificado com êxito para a língua inglesa, que possui consideravelmente menos acentuações do que a língua portuguesa. Para lidar com esse desafio, de adaptação do modelo treinado em inglês para a língua portuguesa, chegamos a diferentes possibilidades:

- Pré-processar os dados de entrada que possuem acentuação, para que a rede consiga identificar com maior precisão um determinado fonema resultante de acentuação que possui forma fixa, ou seja, que se mantém o mesmo de acordo com a posição na frase e/ou palavra;
- Aumento ou troca na disposição de mecanismos de atenção; maior inserção de *speaker embeddings* na rede, ou aumento na sua dimensionalidade, o que aumentaria a caracterização do falante no áudio gerado a custo de maior capacidade de processamento.

Ambas as possibilidades propostas necessitam que haja o treinamento da rede diretamente com um conjunto de dados em português para que sejam validadas e exploradas. Para um caso mais simples, considerando que podemos pensar em uma variação da primeira opção: Os dados de entrada podem ser pré-processados de forma que se adaptem ao alfabeto fonético e, a partir disso, podemos sintetizar um áudio em português utilizando-se a rede que fora treinada em inglês. Neste caso, é esperado que o áudio gerado não possua naturalidade nativa, ou seja, que seu sotaque não seja o esperado de um nativo, em detrimento de um custo-treinamento mais baixo.

4.1.4 Modificações na entrada da arquitetura escolhida

Realizou-se um pré-processamento dos dados, de forma a adaptar a entrada para o alfabeto fonético em inglês e, a partir disso, sintetizou-se áudios em português com o modelo treinado em inglês. O resultado esperado, nesse caso, era o de que as palavras sintetizadas fossem discerníveis para um nativo, embora com um sotaque não nativo.

Para cada uma das sentenças propostas abaixo foram realizados uma série de testes com diferentes mapeamentos fonéticos para o inglês. Segue abaixo a relação de sentença alvo e o melhor mapeamento em termos de naturalidade e discernimento da pronúncia:

O melhor resultado obtido foi com a sentença “Bom dia”, onde a pronúncia é natural e o texto é claro. O maior desafio encontrado foi em como lidar com o fonema em português para ‘é’, o qual não obteve uma boa representação.

Sentença	**Melhor resultado**
"Bom dia"	"Boum gee-a"[a]
"Feitiço"	"Fai-teessoo"[b]
"O céu é azul"	"O cel, eh azUl"[c]
"Áudio de sucesso"	"Oudee-oh gee suce-ssoo"[d]
"Um livro é uma porta para outro mundo"	"Uhm lIvro eh uma pohrta para oltro múndo"[e]

á ⟶ ah	é ⟶ eh	í ⟶ ih	ó ⟶ oh
ú ⟶ uh	ã ⟶ an	á ⟶ aa	à ⟶ a
ão ⟶ aun	ç ⟶ ss	ê ⟶ ee	ô ⟶ oo

Figura 4.5: *Tabela de conversão de letras com acentuação*

[a] bit.ly/ada_bd

[b] bit.ly/ada_ftc

[c] bit.ly/ada_oca

[d] bit.ly/ada_ads

[e] bit.ly/ada_ump

A realização de pré-processamento dos dados de entrada é possível e viável a curto prazo, entretanto, há o risco de haverem fonemas brasileiros que não conseguirão ser representados a partir do fonema em inglês, seja por não existir no idioma ou por outras diferenças. Além disso, a automação desse processo de pré-processamento dos dados pode não gerar resultados excelentes, devido às diferenças de pronúncia de acordo com a posição da letra na palavra.

O próximo passo é treinar a rede com dados em português, para que então se inicie outra série de testes quanto aos dados de entrada e aos módulos da arquitetura, que podem ser modificados ou não.

4.1.5 Experimentos do treinamento da rede

Diferentes formas de treinamento com base em aprendizado com e sem transferência (*transfer learnging*) foram realizadas com o modelo Tacotron 2 pré-treinado em inglês.

Base para o treinamento

Para os treinamentos exibidos a seguir são utilizados tanto o conjunto de dados em inglês LJSpeech, utilizado no estado de treinamento ("*checkpoint*") disponibilizado em comunidade de compartilhamento de código [3], e também dados em português do *Mozilla Common Voice* (MCV), base de dados comunitária desenvolvida e apoiada pela Fundação Mozilla. A base MCV possuía, no momento de treinamento da rede, 27 horas validadas de áudio em português.

[3] https://r9y9.github.io/wavenetvocoder/

Experimentos realizados

Ao todo, foram realizados os seguintes treinamentos com o modelo pré-treinado:

1. Treinamento com base reduzida do MCV (306 amostras/0.39 horas anotadas como voz feminina) a partir modelo pré-treinado até 200.500 passos;
2. Treinamento com base validada completa do MCV a partir do modelo pré-treinado até 220.000 passos;
3. Treinamento com um único falante do MCV (7.631 amostras/6.45 horas) a partir do modelo pré-treinado até 268.500 passos;
4. Treinamento com base validada completa do MCV a partir do modelo pré-treinado até 229.500 passos utilizando conversão do texto para ASCII.
5. Treinamento com um único falante do MCV (6.45 horas) a partir do modelo pré-treinado até 250.000 passos utilizando conversão do texto para ASCII.
6. Treinamento com um único falante do MCV (6.45 horas) a partir do modelo pré-treinado até 212.500 passos utilizando preprocessamento da entrada.

E como forma de averiguar a evolução da rede, além dos valores da função de perda sobre o conjunto de validação a cada época, sintetizamos as frases abaixo para verificarmos pontos onde a rede está se desenvolvendo bem e outros deficientes, em termos de compreensão fonética.

- Não é possível que isso tenha acontecido!
- Eu já sabia sobre ele.
- E então decidimos, vamos viajar neste final de semana.
- Ela é realmente incrível. Nossa!
- Eu nunca iria imaginar que isso aconteceria durante os estudos.
- Sim, para o caso dela vale a pena indicar o monitoramento médico. Isso ajudará muito.

O melhor resultado de treinamento foi obtido com *transfer learning* com seleção de dados de um único falante em português com cerca de 6.3 horas de áudios anotados, utilizando-se o modelo pré-treinado em inglês com o conjunto de dados LJSpeech.

Sobre os experimentos

O primeiro treinamento foi realizado de forma preliminar, de maneira a verificar se a rede é capaz de obter ganhos fonéticos ao se utilizar dados em português em um modelo previamente treinado para o inglês. Com o treinamento com essa pequena base de dados, o áudio sintetizado se assemelha a uma pessoa falando em português, mesmo que a compreensão do que está sendo dita seja baixa.

Com o segundo treinamento verificamos o mesmo comportamento, desta vez com maior clareza no que está sendo dito. Os áudios sintetizados possuem uma mistura de

fonemas agudos e graves, com predominância do fonema mais grave, devido à base validada ser mista de homens e mulheres, com maior prevalência da voz masculina.

O terceiro treinamento, com um único falante do MCV, foi o treinamento que retornou os melhores resultados em comparação com os anteriores, e estendemos o seu treinamento até o ponto de pouca variação dos valores da função de perda sobre o conjunto de validação. Os áudios gerados[4] demonstram uma voz em português brasileiro, porém com dificuldades em lidar com a acentuação da língua portuguesa. Esse comportamento já era esperado, como discutido anteriormente.

Como forma de lidar com a acentuação, realizamos os seguintes experimentos: para o quarto e quinto treinamento realizamos um pré-processamento na entrada para caracteres com codificação ASCII, de forma a retirar a acentuação de palavras acentuadas. Percebemos que em algumas palavras com acentuação seus caracteres tinham o seu som omitido, o que nos leva a crer que a não existência dos caracteres acentuados no alfabeto inglês resulte nesse comportamento. Dessa forma, acreditamos que a conversão para caracteres ASCII poderia melhorar a captação dessas letras pela rede, mesmo que a interpretação fonética seja debilitada devido a exclusão da acentuação. Entretanto, não foram observados resultados notavelmente diferentes dos treinamentos anteriores.

Por fim, para o sexto treinamento realizamos o pré-processamento da entrada como retrata a tabela 4.5, de forma a converter letras acentuadas em uma representação próxima à fonética utilizando-se caracteres do alfabeto inglês. O resultado obtido[5], demonstra que essa abordagem não foi efetiva em melhorar a compreensão de acentuação em palavras. Pelo contrário, chegou a gerar resultados perceptualmente menos naturais do que os resultados obtidos no terceiro treinamento.

Desta forma, o modelo com melhor desempenho foi o resultante a partir do terceiro treinamento.

4.2 Métodos de avaliação da qualidade do áudio gerado

Os resultados obtidos são condizentes com a literatura, o que demonstra uma boa capacidade de síntese de áudio para a língua brasileira. Com isso, estudos de interação humano-computador podem ser realizados.

Na literatura, a avaliação de áudios sintetizados se faz principalmente a partir da escala MOS (*Mean Opinion Score*, ou Score Médio de Opinião). Essa escala avalia a compreensão e naturalidade da fala a partir de uma pesquisa de opinião que utiliza a escala Likert de cinco pontos para cada conceito. Assim, realizamos a avaliação com o objetivo de verificarmos a qualidade percebida do áudio sintetizado com o modelo gerado

Com base no melhor treinamento da rede, a partir do terceiro treinamento, foi realizada a pesquisa de opinião de percepção do áudio gerado com 47 pessoas utilizando-se a escala

[4] *https://bit.ly/2QvIAvT*

[5] *https://bit.ly/2QtGAEv*

MOS de naturalidade e compreensão. Os resultados obtidos são exibidos nas figuras que seguem: na figura 4.6 temos o valor MOS, em 4.7 temos a avaliação de naturalidade e em 4.8 a de compreensão.

A pesquisa consistiu em 48 perguntas, sobre o total de 24 áudios originais utilizados para o treinamento da rede e 24 áudios sintetizados. Em cada conjunto de áudios há um terço de áudios repetidos para que se possa descartar dados de pessoas onde houve inconsistência nas respostas. Todas as amostras com respostas inconsistentes foram descartadas. Para cada áudio foram realizadas as seguintes perguntas:

- Q1: A fala parece natural? (1 = artificial, 5 = natural)
- Q2: Você consegue compreender o que está sendo dito? (1 = incompreensível, 5= totalmente compreensível)

A pesquisa pode ser acessada e respondida a partir do link *https://pathilink.github.io/pesquisa/*. Obtivemos 41 respostas consistentes ao questionário. Com as respostas obtidas, obtivemos os valores da escala MOS obtidos na tabela 4.6. Para fins de comparação entre os critérios de compreensão e naturalidade, mostramos as tabelas 4.7 e 4.8. Pode se verificar que os resultados obtidos são condizentes com os resultados da literatura estudada.

Geral	Original	Sintetizado
MOS	4.810 ± 0.149	3.611 ± 0.929
MOS normalizado	5.000	3.753 ± 0.965

Figura 4.6: *Tabela com valores obtidos da escala MOS.*

Q1	Original	Sintetizado
Avaliação	4.705 ± 0.133	2.720 ± 0.267
Avaliação normalizada	5.000	2.890 ± 0.284

Figura 4.7: *Tabela resultante da avaliação de naturalidade (Q1).*

Q2	Original	Sintetizado
Avaliação	4.917 ± 0.065	4.502 ± 0.189
Avaliação normalizada	5.000	4.578 ± 0.192

Figura 4.8: *Tabela resultante da avaliação de compreensão (Q2).*

Capítulo 5

Considerações e conclusões

A realização de estudos sobre arquiteturas de síntese de voz, seguida pela sua comparação e seleção, propiciou o desenvolvimento do modelo de síntese de voz em português brasileiro em sua fase inicial, o que permite o início de experimentos de interação por voz por parte do projeto ADA.

Durante o trabalho foram desenvolvidos o levantamento de critérios de seleção para arquiteturas ponta a ponta para a síntese de voz e um estudo de diferentes formas de treinamento com dados em português através da técnica denominada *Transfer Learning*. Esses dois trabalhos podem ser reutilizados e usados como base para desenvolvimentos futuros do projeto ADA ou de projetos externos.

Quanto ao estudo comparativo entre as arquiteturas, vale destacar que a falta de informações padronizadas entre os artigos restringiu o levantamento de critérios para seleção. Portanto, com a adição de mais informações sobre as arquiteturas, os critérios para seleção podem mudar, mas é esperado que isso não tenha grande impacto na diferença de qualidade e características percebidas entre as arquiteturas estudadas.

As informações disponibilizadas acerca das necessidades computacionais e de dados para o treinamento dos modelos em seus respectivos artigos são escassas e não padronizadas. A construção da tabela comparativa auxiliou na comparação geral entre arquiteturas e com isso reafirmamos como a sistematização das informações contidas nos artigos é fundamental para o processo de escolha de uma arquitetura. Também acreditamos que informações como o volume de dados utilizados e seu perfil, quantidade de máquinas utilizadas, tempo médio de treino (ou ainda definição de iterações ou passos do modelo) e *batch size* (parâmetro que pode auxiliar no treinamento por indicar um tamanho ótimo da entrada de treinamento que maximize a aprendizagem) são boas sugestões de informações que podem auxiliar no estudo comparativo das arquiteturas.

Quanto ao estudo de estratégias de treinamento que utilizaram *Transfer Learning*, o melhor resultado foi obtido ao se utilizar um conjunto de dados com 6.45 horas de áudio de um único falante sem a realização de nenhum pré-processamento na entrada da rede. Entretanto, palavras com acentuação nem sempre são pronunciadas da forma esperada devido a diferença entre os alfabetos inglês e português. Na execução dos experimentos, foi frustrante o resultado de baixa qualidade do sexto treinamento, onde houve

preprocessamento na entrada da rede com a troca de letras com acentuação para uma representação próxima à fonética que utilize caracteres ASCII. Isso porque a aplicações das trocas fonéticas foram consistentes em todo o conjunto de dados. Todavia, refletindo sobre o ocorrido, foi percebido que esse era um comportamento esperado dado que o Tacotron 2 mapeia letras para fonemas, caractere a caractere.

Os resultados obtidos a partir do valor na escala MOS de naturalidade e compreensão assemelham-se aos resultados obtidos na literatura para a avaliação do Tacotron 2. Observando-se separadamente os critérios de naturalidade e compreensão, notamos que a maior parte dos entrevistados classificaram o modelo como muito compreensível, enquanto o caráter de naturalidade da voz se mostra no meio do caminho entre um áudio totalmente robótico e um natural.

Para lidar com a necessidade de maior naturalidade à voz gerada, sugerimos como melhoria para trabalhos futuros a utilização de Vocoders especializados em voz humana ao invés do algoritmo Griffin-Lim, como por exemplo o WaveNet, Glow ou WaveGlow. Quanto à questão de acentuação no alfabeto português brasileiro, sugerimos o treinamento do zero com dados em português ou ainda o treinamento do zero com dados em inglês para posterior aplicação de *Transfer Learning* com dados em português. Nesses dois últimos casos, é necessário que haja a expansão do vocabulário simbólico da rede com a inserção de caracteres acentuados, e é imprescindível a maior quantidade de dados (na escala de 30 horas, como no conjunto de dados LJSpeech).

Mesmo com as ressalvas citadas, o modelo desenvolvido mostra-se adequado para que seja iniciada a realização de estudos na área de interação humano-computador para o projeto ADA, dado também o estado de maturidade tecnológica do mesmo. Dessa forma, todo o processo de interação, desde o reconhecimento de fala ao retorno ao usuário com a execução da tarefa solicitada, pode ser experimentado e desenvolvido entre os diferentes módulos da assistente distribuída.

Referências

[Abu-Mostafa 2012] Yaser S Abu-Mostafa. *Learning from data: a short course*. 2012 (citado nas pgs. 3, 11).

[Goodfellow *et al.* 2016] Ian Goodfellow, Yoshua Bengio e Aaron Courville. *Deep learning*. MIT press, 2016 (citado na pg. 11).

[Huang *et al.* 2017] Rui Huang, Shu Zhang, Tianyu Li e Ran He. "Beyond face rotation: global and local perception gan for photorealistic and identity preserving frontal view synthesis". Em: *Proceedings of the IEEE international conference on computer vision*. 2017, pgs. 2439–2448 (citado na pg. 6).

[Li *et al.* 2019] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao e Ming Liu. "Neural speech synthesis with transformer network". Em: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pgs. 6706–6713 (citado na pg. 15).

[Luong *et al.* 2015] Minh-Thang Luong, Hieu Pham e Christopher D Manning. "Effective approaches to attention-based neural machine translation". Em: *arXiv preprint arXiv:1508.04025* (2015) (citado na pg. 9).

[Patterson e Gibson 2017] Josh Patterson e Adam Gibson. *Deep learning: A practitioner's approach*. "O'Reilly Media, Inc.", 2017 (citado nas pgs. 4, 11).

[Ping, Peng e J. Chen 2018] Wei Ping, Kainan Peng e Jitong Chen. "Clarinet: parallel wave generation in end-to-end text-to-speech". Em: *arXiv preprint arXiv:1807.07281* (2018) (citado nas pgs. 11, 13, 14, 19).

[Ping, Peng, Gibiansky *et al.* 2017] Wei Ping, Kainan Peng, Andrew Gibiansky *et al.* "Deep voice 3: scaling text-to-speech with convolutional sequence learning". Em: *arXiv preprint arXiv:1710.07654* (2017) (citado nas pgs. 11, 15, 18, 19).

[Shen *et al.* 2018] Jonathan Shen *et al.* "Natural tts synthesis by conditioning wavenet on mel spectrogram predictions". Em: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pgs. 4779–4783 (citado nas pgs. 11, 17–19).

[Sotelo *et al.* 2017] Jose Sotelo *et al.* "Char2wav: end-to-end speech synthesis". Em: *Workshop track - ICLR 2017* (2017) (citado nas pgs. 11, 13, 18, 19).

[Wang *et al.* 2017] Yuxuan Wang *et al.* “Tacotron: towards end-to-end speech synthesis”. Em: *arXiv preprint arXiv:1703.10135* (2017) (citado nas pgs. 9, 11, 16, 17, 19).